Meikang Qiu (Ed.)

LNCS 10699

Smart Computing and Communication

Second International Conference, SmartCom 2017
Shenzhen, China, December 10–12, 2017
Proceedings

Lecture Notes in Computer Science 10699

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

More information about this series at http://www.springer.com/series/7409

Meikang Qiu (Ed.)

Smart Computing and Communication

Second International Conference, SmartCom 2017
Shenzhen, China, December 10–12, 2017
Proceedings

Springer

Editor
Meikang Qiu
Columbia University
New York, NY
USA

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-73829-1 ISBN 978-3-319-73830-7 (eBook)
https://doi.org/10.1007/978-3-319-73830-7

Library of Congress Control Number: 2017963773

LNCS Sublibrary: SL3 – Information Systems and Applications, incl. Internet/Web, and HCI

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This volume contains the papers presented at SmartCom 2017: the Second International Conference on Smart Computing and Communication held at Shenzhen University in Shenzhen, China.

There were 248 submissions. Each submission was reviewed by at least two, and on average 2.2, Program Committee members. The committee decided to accept 44 papers. The acceptance rate is less than 18%.

Recent developments in Web-based technologies and mobile applications have facilitated a dramatical growth of new technique implementations, such as cloud computing, big data, pervasive computing, Internet of Things, and social cyber-physical systems. Enabling a smart life has become a popular and timely research topic. Therefore, the International Conference on Smart Computing and Communication focuses on the fields of both smart computing and communications and aims to collect recent academic work to improve research and practical applications.

The scope of SmartCom 2017 was broad, from smart data to smart communications, from smart cloud computing to smart security. The conference gathered all high-quality research/industrial papers related to smart computing and communications and aimed at proposing a reference guideline for further research. All accepted papers are indexed by EI. SmartCom 2017 was held at Shenzhen University in China and the proceedings are published by Springer.

We would like to give special thanks to our general chairs, Guoliang Chen (Chinese Academy of Sciences), Zhong Ming (Shenzhen University), Qiang Yang (The University of Rhode Island), and Meikang Qiu (Pace University). We also appreciate our program chairs, Rui Mao (Shenzhen University), Zongming Fei (University of Kentucky), Haibo Zhang (University of Otago), and Xizhao Wang (Shenzhen University), and our operation chair, Keke Gai (Pace University), publicity chair, Hui Zhao (Henan University), and local chair, Shubin Cai (Shenzhen University). We greatly appreciate the contribution of our keynote speakers, presenters, attendees, and all other contributors.

Finally, we want to express our appreciation to our sponsors: NSF (USA), Springer LNCS, Shenzhen University, Columbia University, Pace University, and all co-sponsoring organizations.

November 2017 Meikang Qiu

Organization

Honorary General Chair

Guoliang Chen	Chinese Academy of Sciences, China

General Chairs

Zhong Ming	Shenzhen University, China
Meikang Qiu	Columbia University, USA
Qing Yang	The University of Rhode Island, USA

Program Chairs

Zongming Fei	University of Kentucky, USA
Rui Mao	Shenzhen University, China
Xizhao Wang	Shenzhen University, China
Haibo Zhang	University of Otago, New Zealand

Volume Editor

Meikang Qiu	Columbia University, USA

Operation Chair/Web Chair

Keke Gai	Pace University, USA

Local Chairs

Shuibin Cai	Shenzhen University, China
Rui Mao	Shenzhen University, China

Publicity Chair

Hui Zhao	Henan University, China

Technical Program Committee

Thomas Austin	San Jose State University, USA
Emmanuel Bernardez	IBM Research, USA
Sang-Yoon Chang	Advanced Digital Sciences Center, Singapore
Zhongming Fei	University of Kentucky, USA
Yong Guan	Lowa State University, USA

Xiaofu He	Columbia University, USA
Hao Hu	Nanjing University, China
Yue Hu	Louisiana State University, USA
Suman Kumar	Troy University, USA
Weigang Li	University of Brasilia, Brazil
Wenjia Li	New York Institute of Technology, USA
Xin Li	Carnegie Mellon University, USA
Zhiqiang Lin	University of Texas at Dallas, USA
Bo Luo	The University of Kansas, USA
Bharat Rawal	Pennsylvania State University, USA
Fuji Ren	The University of Tokushima, Japan
Ukka Riekki	University of Oulu, Finland
Art Sedighi	Global Head of Cloud Architecture and Strategy, TD Bank, USA
Lixin Tao	Pace University, USA
Zhipeng Wang	China Electronics Standardization Institute, China
Jian Xiong	Shanghai Jiao Tong University, China
Jinjun Xiong	IBM Research, USA
Yu-Dong Yao	Stevens Institute of Technology, USA
Wei Yu	Towson University, USA
Haibo Zhang	University of Otago, New Zealand
Peng Zhang	Stony Brook University, USA
Yan Zhang	University of Oslo, Norway
Yong Zhang	The University of Hong Kong, China
Hui Zhao	Henan University, China

Contents

Sphere Decoding Algorithm Based on a New Radius Definition

Ping Wang[1], DongDong Shang[2(✉)], and Zhiwei Sun[3]

[1] College of Information Engineering, Shenzhen University, Shenzhen 518060, China
wangping@szu.edu.cn
[2] College of Computer Science and Software, Shenzhen University, Shenzhen 518060, China
2150230413@email.szu.edu.cn
[3] College of Computer Engineering, Shenzhen Polytechnic, Shenzhen 518060, China
smeker@szpt.edu.cn

Abstract. The problem of integer least squares (ILSP) is playing a significant role in cryptography. ILSP is equivalent to finding the closest lattice point to a given point and is known to be NP-hard. Sphere decoding (SD) is an effective method for solving the ILSP. In this paper, we mainly solve ILSP by the sphere decoding algorithm (SDA). One of the key issues in SDA is how to implement a more effective tree pruning strategy. In this paper a new definition for sphere radius is proposed as a tree pruning strategy to reduce the computational complexity. The core idea of our algorithm (K-SE-SD) is to sacrifice a relatively small reduction in accuracy to reduce relatively more time complexity. Our experiments demonstrate that on average the proposed idea can reduce about 70.8% running time with an accuracy of 86.8% when $k = \lceil n/3 \rceil$, where n is the lattice dimension.

Keywords: Integer least squares · Cryptography
Sphere decoding algorithm · Closest lattice point
Definition for sphere radius

1 Introduction

The integer least squares problem has many important applications, such as wireless communication [1], cryptography [2], GPS [3], engineering aspect and so on. The ILSP is defined as follows:

$$\min_{x \in \mathbb{Z}^n} \|Ax - y\|_2^2, \tag{1}$$

for $A \in \mathbb{R}^{m*n}$, $m \geq n$, $y \in \mathbb{R}^m$. Assuming that A is a column full rank matrix, that is the column vectors of A are a set of basis, x is integer, Ax is the integer times linear combinations of this base, they compose a lattice space. The goal

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 1–10, 2018.
https://doi.org/10.1007/978-3-319-73830-7_1

is to find a lattice point closest to the y vector. The Shortest Vector Problem (SVP) is one of the most important computational problems on lattices. Its difficulty is closely related to the security of most lattice-based cryptographic constructions to date. In the last few decades, Lattice-based cryptography has gained wide attention from the cryptographic community, due to e.g. its presumed resistance against quantum attacks [4], average-case hardness guarantees [5], the existence of lattice-based fully homomorphic encryption schemes [6], and efficient cryptographic primitives like NTRU [7].

Sphere decoding algorithm can be divided into breadth-first algorithm and depth-first algorithm. The performance of breadth-first algorithm [8,9] is slightly less, but its computational complexity is fixed. The depth-first algorithm (DFSD) can provide optimal performance [10–12]. But transformation range of the complexity is very large. Schnorr-Euchner enumeration (SEE) based sphere decoders (SE-SD) [13] alleviate this problem by visiting the nodes at the same layer according to SEE. SE-SD is still a depth-first tree search algorithm.

SDA has two main issues. The first one is how to effectively reduce the scope of searching. Another one is that how to apply effective tree pruning strategy to reduce the number of saved nodes in the search process. There are two main approaches for implementation a more effective pruning strategy in sphere decoding algorithm; faster shrinkage of the sphere radius or a new definition for sphere radius. In this paper we follow the later and propose a new definition of sphere radius which is a function of the tree level. We use it reduce the computational complexity of SE-SD, our new algorithm (K-SE-SD) provides a tradeoff between the computational complexity and the accuracy, that is say that K-SE-SD exchanges relatively small accuracy for reducing relatively more time complexity.

The paper is organized as follows. We start in Sect. 2 with a description of SE-SD which is a variant of SDA. In Sect. 3 we describe the new proposed idea (K-SE-SD) and give our concrete implementation steps. In Sect. 4, we present our experimental results and the comparison of different algorithms. Conclusions are drawn in the last section.

2 The Sphere Decoding Algorithm with Schnorr-Euchner Enumeration

In the general, we will turn A into an upper triangular matrix with QR decomposition.

$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix},$$

where $Q \in \mathbb{R}^{n*n}$ is orthogonal and $R \in \mathbb{R}^{m*m}$ is upper triangular. Partitioning $Q = [Q_1\ Q_2]$, where Q_1 is $n \times m$ and Q_2 is $n \times (n-m)$, we get

$$\|Ax - y\|_2^2 = \|Rx - Q_1^{-1}y\|_2^2 + \|Q_2^{-1}y\|_2^2 \leq d^2. \tag{2}$$

Expanding vector norm in (2) yields

$$\sum_{i=1}^{n}\left|y_i-\sum_{j=i}^{n}R_{i,j}x_j\right|^2\leqslant d^2 \tag{3}$$

where $d^2=d^2-\|Q_2^{-1}y\|_2^2$. starting from $i=n$, (4) can be solved recursively as follows:

$$\|T_i\left(P_i\right)\|^2=\|T_{i+1}\left(P_{i+1}\right)\|^2+\|e_i\left(P_i\right)\|^2 \tag{4}$$

$$T_{n+1}\left(P_{n+1}\right)=0 \tag{5}$$

Where $T_i\left(P_i\right)>T_{i+1}\left(P_{i+1}\right)$ and $e_i\left(P_i\right)=\sum\limits_{j=i}^{n}R_{i,j}x_j-y_i$. In (6) and (7), $P_i=[x_i,x_{i+1},\ldots,x_n]^T$ is commonly known as *partial symbol vector*. It is very obvious that SDA can be recursively solved by applying an iterative tree search methodology. The cost function of the resulting optimization problem is $T_i\left(P_i\right)$ which is known as *Partial Euclidean Distance* (PED).

In general, the kth interval x_k found between LB (the lower bound) and UB (upper bound), where LB and UB can be computed as:

$$LB=\left\lceil\frac{-d_k+\hat{y}_k}{R_{k,k}}\right\rceil \text{ and } UB=\left\lfloor\frac{d_k+\hat{y}_k}{R_{k,k}}\right\rfloor,$$

where $\hat{y}_k$ and d_k are obtained as follows:

$$\hat{y}_k=\hat{y}_k-R_{k,k+1:n}x_{k+1:n} \tag{6}$$

$$d_k=\sqrt{\left(T_{k+1}\left(P_{k+1}\right)\right)^2-\left(\hat{y}_{k+1}-R_{k,k+1:n}x_{k+1:n}\right)} \tag{7}$$

DFSD followed the Phost enumeration, which searches the candidates in the order $LB, LB+1, ..., UB$. SE-SD combines the advantages of the Babai nearest plane algorithm and the Pohst strategy. Let $\hat{x}_k=\frac{1}{R_{k,k}}(\hat{y}_k-\sum\limits_{j=k+1}^{n}R_{k,j}x_j)$, if $\hat{x}_k\leqslant\lfloor\hat{x}_k\rceil$, Then the sequence $\hat{x}_k=\lfloor\hat{x}_k\rceil,\lfloor\hat{x}_k\rceil-1,\lfloor\hat{x}_k\rceil+1,\lfloor\hat{x}_k\rceil-2,...$ orders candidates nodes according to nondecreasing distance from $\hat{x}_k$ to root node. A trivial counterpart holds when $\hat{x}_k>\lfloor\hat{x}_k\rceil$. Since the volume of a layer decreases with increasing, the chance of finding the correct layer early is maximized. SE-SD updates the radius of the sphere after the first point is found. This point is known as the Babai point [14]. Hence, SE-SD allows us to fix the highest initial radius and to find the right path earlier than Phost enumeration [15].

3 The Definition of Radius

In this section, we describe how to determine the search radius so that the effect of SDA is better than the original idea. Firstly, we consider how to use the LLL lattice reduction algorithm further reduces the radius; then we use a new definition for sphere radius which is a function of the tree level to reduce the number of saved nodes in the search of tree graph based on the above. Details are as follows.

3.1 Reduce the Radius with the LLL Algorithm

In this section, we will show LLL algorithm [16] is used to further reduce the matrix R, it makes the radius smaller. The number of the lattice point in a sphere also reduces a lot. Since the integer least squares problem is reduced to the triangular ILSP. The LLL algorithm can be also applied to the upper triangular R. It can decompose R into $\widehat{Q}\widehat{R}T^{-1}$, where $\widehat{Q}$ is orthogonal, $\widehat{R}$ is upper triangular, T is unimodular, so that the columns of $\widehat{R}$ form a reduced basis according to LLL reduction algorithm.

Let A and y be the same as in introduction. After the QR decomposition for A, Q is orthogonal matrix, R is upper triangular. Then applying the LLL algorithm with $\delta = 0.75$ to the above Q and R, we can obtain $\widehat{Q}\widehat{R}T^{-1}$, where $\widehat{Q}$ is new orthogonal, $\widehat{R}$ is new upper triangular matrix, and T is unimodular. Putting the obtained values into the original expression, we can get new $\widehat{y}$:

$$\widehat{y} = \widehat{Q}^{-1}y.$$

Solving the equation of $\widehat{R}x = \widehat{y}$, we can obtain the solution of x and round the x. Then, leading to a smaller radius:

$$d = \|\widehat{R}round(x) - \widehat{y}\|.$$

After initializing radius, finding the solution by SDA, if we get a solution, pre-Multiply it by T, we will obtain the same value as the decomposition of QR.

Algorithm 1. Function of definition of sphere radius

Input: A m-dimensional vector $x \in \mathbb{Z}^n$ such that be updated, parameter: k, a n dimensions array $radiu[n]$, The upper triangular matrix: r (the above R)

Output: The updated n dimensions array $radiu[n]$

```
for i = n : 1 do
    for j = i − k + 1 : n do
        for k = j : n do
            temp += r_{j,k} x_k
        end for
        radiu[j]+=temp * temp
    end for
end for
for i = k − 1 : 1 do
    radiu[i] = radiu[k − 1]
end for
```

3.2 The K-SE-SD Algorithm

Faster shrinkage of the sphere radius and a new definition for sphere radius are two main approaches for implementation a more effective pruning strategy in SDA. In this paper we propose a new definition of sphere radius which is a function of the tree level to reduce SE-SD algorithm's time complexity. The function

of the tree level is listed in Algorithm 1. How to use the proposed function to reduce SE-SD algorithm's computational complexity is listed in Algorithm 2.

The conventional SE-SD algorithm only updates the radius when searching at leaf nodes (level one) of the search tree and finds a new result with smaller value; such that:

$$bestdist = \min\left(T_1\left(P_1\right)\right) \tag{8}$$

This updated radius will remain unchanged and be used to prune branches of tree at every level. In sphere decoding, the sphere radius test requires the following condition to be satisfied at every level:

$$T_i(x_n) < bestdist \tag{9}$$

In our proposed K-SE-SD algorithm, PEDs are compared to the associated sphere radius which is a function of the tree level instead of the global radius when searching each level of the tree. The theoretical basis of K-SE-SD is that if the PED of a parent node is growing faster than the previously detected smallest path, there is a higher risk for the children of that parent node to fail later in the search. Suppose $x = [x_1, \ldots x_n]$ is a previously detected smallest path, Independent variables of new sphere radius function of the tree level are as follows:

$$\begin{aligned} x^n &= [x_{n-k+1}, \ldots, x_{n-1}, x_n]^T \\ x^{n-1} &= [x_{n-k}, \ldots, x_{n-1}, x_n]^T \\ &\vdots \\ x^k &= [x_1, \ldots, x_{n-1}, x_n]^T \\ &\vdots \\ x^1 &= [x_1, \ldots, x_{n-1}, x_n]^T \end{aligned} \tag{10}$$

So the dependent variables of new sphere radius function of the tree level are as follows:

$$\begin{aligned} radiu[n] &= T_n\left(x^n\right) \\ radiu[n-1] &= T_{n-1}\left(x^{n-1}\right) \\ &\vdots \\ radiu[1] &= T_1\left(x^1\right) \end{aligned} \tag{11}$$

Due to definition of PED we have the following relationship for the sphere radius:

$$\text{radiu}[1] \geqslant \text{radiu}[2] \geqslant \ldots \geqslant \text{radiu}[n] \tag{12}$$

From the above, it is obvious that the number of visiting nodes and the complexity of running time significantly reduces a lot. Section 4 shows the experimental data, and explains in detail the running effect of the program.

Algorithm 2. The K-SE-SD Algorithm

Input: The upper triangular matrix: r (the above R), the center of the sphere: y (the above $\widehat{y}$), parameter: k, a n dimensions array $radiu[n]$.

Output: A m-dimensional vector $x \in \mathbb{Z}^n$ such that $r \cdot x$ is a lattice point closest to y.

1: $n :=$ the size of r, bestdist $:= \infty$, $k := n$, $dist_k := 0$, $\hat{x} := (0, \ldots, 0)$, $number := 0$
2: $\hat{x}_n := \lfloor y_n / r_{n,n} \rceil$, $value_k = r_{n,n}\hat{x}_n - y_n$, $step_k :=$ sgn*$(value_k)$
3: **while** true **do**
4: newdist $:= dist_k + {value_k}^2$
5: **if** $((newdist < bestdist)) \| ((newdist < radiu[k])$ and $number$!=0) **then**
6: **if** $k \neq 1$ **then**
7: k := k−1
8: $dist_k =$ newdist
9: $\hat{x}_k := e_k(\hat{x}_k)/r_{k,k}$
10: $value_k := e_k(\hat{x}_k)$
11: $step_k :=$ sgn*$(value_k)$
12: **else**
13: ++ $number$
14: call for Algorithm 1
15: $x := \hat{x}$
16: bestdist := newdist
17: k := k+1
18: $\hat{x}_k := \hat{x}_k + step_k$
19: $value_k := e_k(\hat{x}_k)$
20: $step_k := -step_k -$ sgn*$(value_k)$
21: **end if**
22: **end if**
23: **if** $newdist > radiu[k]$ && $number$!=0 **then**
24: **if** $k = m$ **then**
25: return x
26: **else**
27: k:=k+1
28: $\hat{x}_k := \hat{x}_k + step_k$
29: $value_k := e_k(\hat{x}_k)$
30: $step_k := -step_k -$ sgn*$(value_k)$
31: **end if**
32: **end if**
33: **end while**

4 Experiments

In this section, we compare the results under the different radius function. We performed numerous experiments to test the different algorithms. Through a large number of data test(more than ten thousand data), comparing the performance among DFSD, SE-SD, K-SE-SD (k = n/3), K-SE-SD (k = n/5) and K-SE-SD (k = n/7). Eventually, we confirm the correctness of the theoretical analysis.

Firstly, after the QR decomposition of matrix A, ordering the total number of lattice points lying in the interval into ascending order in accordance with

the $T_i\,(p_i)\,_{(1\leqslant i\leqslant n)}$. Secondly, we are considering the LLL algorithm as a method of reducing radius. At last, using K-SE-SD algorithm with different k to reduce the SE-SD algorithm's time complexity, the effect is very obvious. In the next moment, the different graphs are listed as follows.

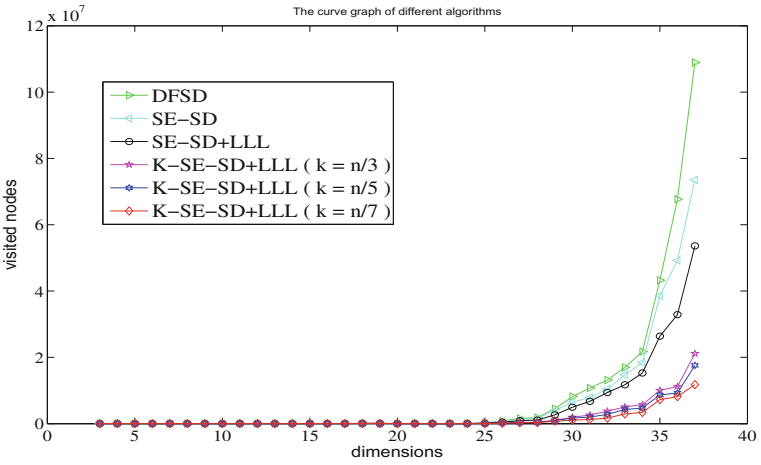


Fig. 1. The number of visited nodes under different dimensions.

Figure 1 shows the number of visited nodes under different dimensions. Usually, the bigger the dimensions are, the more the number of visited nodes are. Of course, if the nature of the different matrix is different, it also has a special phenomenon. As a whole, the graph will present a rising trend. From the graph, we know the red line is the best effect. Next, listing detailed values by a table.

Table 1. The number of visited nodes under different dimensions.

Curve\ Dimensions	4	8	12	16	20	24	28	32
$DFSD$	12	91	642	3288	29309	50330	1807199	13151570
$SE-SD$	10	76	487	2612	21843	45430	1638921	10468379
$SE-SD+LLL$	10	58	335	1570	17955	38352	1085464	9386576
$K-SE-SD+LLL(k=n/3)$	10	52	275	1187	7186	15877	423330	3765849
$K-SE-SD+LLL(k=n/5)$	10	46	210	864	6148	13740	354652	2798431
$K-SE-SD+LLL(k=n/7)$	8	41	156	628	5381	11970	293614	1862184

Table 1 lists the change condition of the different curves. The number of node for the second case is often more than the first one due to Schnorr-Euchner (SE) enumeration. The third one is usually less than the first two except for the exceptional circumstances. The second case without LLL algorithm is not so

good than the third one. We can know that the effect of LLL algorithm cannot be ignored. The last three cases are belonged to the same kind radius function with different parameter k. When the parameter k is larger, the number of visited nodes is more bigger. The effect of the proposed algorithm is very obvious and the number of nodes each dimension is minimal.

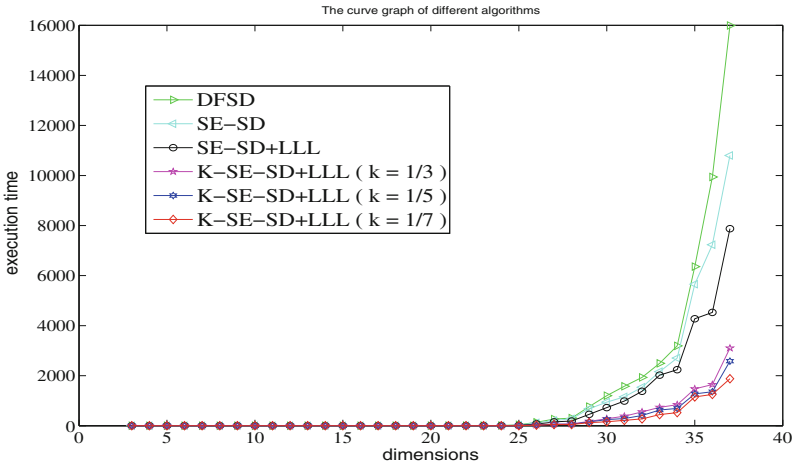


Fig. 2. The execution time of program under different dimensions.

Figure 2 shows the execution time of program under different dimensions. Usually, the more the visited nodes are, the more the execution time are. The six curves clearly show the effect of the execution time. Table 2 lists detailed execution time. There will be slight difference between each curve. Because program has recording function, so the results of running could be different every time. We can know that the last one is the best.

Table 2. The execution time of program under different dimensions.

Curve\Dimensions	4	8	12	16	20	24	28	32
$DFSD$	0.001	0.026	0.082	0.435	4.336	8.361	316.837	2039.436
$SE-SD$	0.001	0.021	0.061	0.334	3.205	6.646	278.304	1537.354
$SE-SD+LLL$	0.001	0.016	0.043	0.231	2.631	5.637	189.322	1378.483
$K-SE-SD+LLL(k=n/3)$	0.001	0.015	0.035	0.154	1.058	2.329	71.885	553.041
$K-SE-SD+LLL(k=n/5)$	0.001	0.013	0.026	0.127	0.896	1.989	60.244	410.969
$K-SE-SD+LLL(k=n/7)$	0.001	0.012	0.019	0.105	0.784	1.754	49.858	273.474

Figure 3 shows the accuracy of different dimensions. It is generally known that the first three algorithms are accurate algorithm, so their accuracy is 100%. The latter three algorithms's accuracy are improved according to the parameter k.

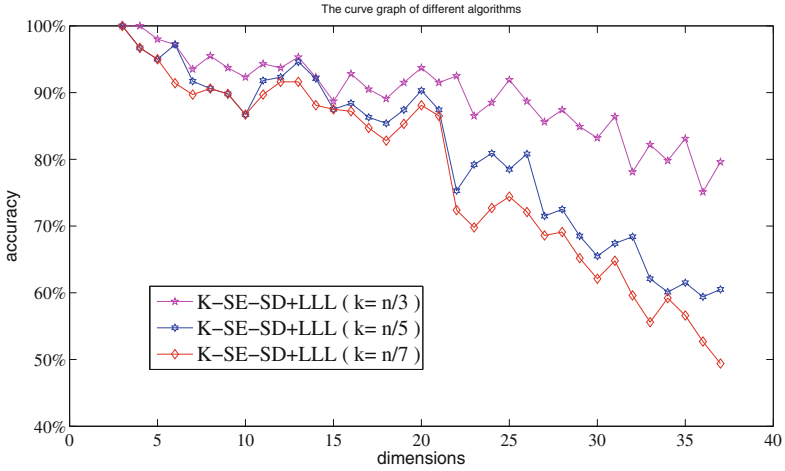


Fig. 3. The accuracy of program under different dimensions.

Table 3. The accuracy of program under different dimensions.

$Curve \diagdown Dimensions$	4	8	12	16	20	24	28	32
$K-SE-SD+LLL(k=n/3)$	100%	95.5%	93.7%	92.8%	93.7%	88.5%	87.4%	78.1%
$K-SE-SD+LLL(k=n/5)$	96.7%	90.6%	92.3%	88.4%	90.3%	80.9%	72.5%	68.4%
$K-SE-SD+LLL(k=n/7)$	96.7%	90.6%	91.6%	87.2%	88.1%	72.7%	69.1%	59.6%

Table 3 records the changes of curve. The radius functions decide the search path, the visited nodes, the running time and the complexity. Obviously, our proposed algorithm reduce relatively more time complexity at the cost of sacrificing relatively small accuracy.

We show the relationships among different graphs by three figures and three tables. When all of these algorithms are considered together, it is not surprising that the time complexity of our new algorithm is lower than the previous algorithms. On average the proposed idea can reduce about 70.8% running time with an accuracy of 86.8% when k = N/3.

5 Conclusion

In this paper, we considered the integer least squares problem is solved by sphere decoding algorithm. We addressed a key issue in SDA that how to apply a more effective tree pruning strategy to reduce the number of saved nodes. We have shown that a great execution time and a high efficiency are reachable by using our new ideas. Our experiments demonstrate that on average the proposed idea can reduce the running time by around 70.8% with an accuracy of 86.8% when k = N/3.

References

1. Hassibi, B., Vikalo, H.: On the sphere-decoding algorithm I. Expected complexity. IEEE Trans. Signal Process. **53**(8), 2806–2818 (2005)
2. Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. In: Kaliski, B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 112–131. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0052231
3. Hassibi, A., Boyd, S.: Integer parameter estimation in linear models with applications to GPS. IEEE Trans. Signal Process. **46**(11), 2938–2952 (1998)
4. Bernstein, D.J.: Introduction to Post-Quantum Cryptography. Springer, Heidelberg (2009)
5. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: Twenty-Eighth ACM Symposium on Theory of Computing, pp. 99–108 (1996)
6. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, 31 May–2 June 2009, pp. 169–178 (2009)
7. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: a ring-based public key cryptosystem. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 267–288. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0054868
8. Lai, K.C., Huang, C.C., Jia, J.J.: Variation of the fixed-complexity sphere decoder. IEEE Commun. Lett. **15**(9), 1001–1003 (2011)
9. Choi, J.W., Shim, B., Singer, A.C.: Efficient soft-input soft-output tree detection via an improved path metric. IEEE Trans. Inf. Theor. **58**(3), 1518–1533 (2012)
10. Schnorr, C., Euchner, M.: Lattice basis reduction: improved practical algorithms and solving subset sum problems. Math. Program. **66**(1–3), 181–199 (1994)
11. Fincke, U., Pohst, M.: Improved methods for calculating vectors of short length in a lattice. Math. Comput. **44**, 463–471 (1985)
12. Viterbo, E., Boutros, J.: A universal lattice code decoder for fading channels. IEEE Trans. Inf. Theor. **45**(5), 1639–1642 (1999)
13. Samuel, M., Fitz, M.: Iterative sphere detectors based on the Schnorr-Euchner enumeration. IEEE Trans. Wirel. Commun. **9**(7), 2137–2144 (2010)
14. Grtschel, M., Lovsz, L., Schrijver, A.: Geometric algorithms and combinatorial optimization. Algorithms Comb. **40**(8), 1–362 (1988)
15. Pohst, M.: On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications. ACM (1981)
16. Qiao, S.: Integer least squares: sphere decoding and the LLL algorithm. In: C 3 S 2 E Conference, pp. 23–28 (2008)

Inferring Travel Purposes for Transit Smart Card Data Using

Zhenzhen Liu[1,2], Qing-Quan Li[1], Yan Zhuang[3(✉)], Jiacheng Xiong[1], and Shuiquan Li[1,2]

[1] Shenzhen Key Laboratory of Spatial Information Smart Sensing and Services, Shenzhen, China
[2] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
[3] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China
zhuangyan@whu.edu.cn

Abstract. Understanding travel purposes is crucial for urban transportation planning and resource allocation. Conventionally, travel purposes are obtained from household travel survey, however, household travel survey is usually conducted every 5–10 years and only has 1–2% sampling size over the whole urban dwellers. Therefore, the information on travel purposes is very limited and usually biased which cause mismatch in urban and transportation planning. Meanwhile, many cities have accumulated a large amount of transit data, such as those from transit smart cards. Such data contains many individual traveling records, but has not been included to generate travel survey data because of lacking the information of travel purposes. To make fully use the data and to generate more comprehensive travel data, this study attempted to infer travel purposes for smart card data by a naïve Bayes probabilistic model. Experimental results demonstrated that proposed method could infer commuting activities with the accuracy of more than 95%, while the accuracy of predicting other activities was about 60%. This is a promising approach to integrated big data into transportation work routines.

Keywords: Travel purpose · Travel survey · Transit smart card
Bayes probabilistic model

1 Introduction

Understanding dwellers' travel purposes is one of the basis for urban and transportation planners. Conventionally, household travel survey is a major channel for obtaining such travel information [1, 2]. Since travel survey is costly and needs a lot of resource and human inputs, could only be conducted every several years, Moreover, most surveys only record a single-day travel records [3, 4]. Therefore, the information on travel purposes is very limited and usually biased which cause mismatch in urban and transportation planning.

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 11–18, 2018.
https://doi.org/10.1007/978-3-319-73830-7_2

In recent years, with the fast development of smart transport system and data collection technology, a large amount of transit smart card data has been accumulated. The transit smart card data has the advantage to detect resident's travel activities [5]. On the one hand, it has been computerized without human inputs like household travel survey. On other hand, the smart cards record travelers' travel data in a daily, monthly, and even yearly period, and thus could reflect many travel details. Given the above, as an emerging and passive way to collect data, transit smart cards has the potential to support traditional travel studies [6].

The types of travel purposes are important attributes to describe traveler's activities [7]. Some researchers use other data sources to analyze travel purposes, such as GPS [8, 16]. The travel activities could be further accurately deduced in combination of social and economic features of surveyed samples, such as family composition and model [9]. Other studies [10–12] combined geographic coding address of land use data, the work place, schools and frequently visited shops. Chen et al. clustered traveler's destination and activity sites and then predicted the travel activities in low-density regions based on the confirmed rules [13–15]. Other rules include activity duration, time and sequence [17]. While studies on travel purposes by using transit smart card data are insufficient. We first combined travel survey data and transit smart card data to analyze the relationship between travel characteristics and travel purpose, and then trained the Bayesian probabilistic model to infer travel purposes.

2 Data Description

In this study, household travel survey data were collected from Shenzhen City, P.R. China. A smart phone based survey system was developed. Respondents provided their personal information (e.g. age, gender, education, work and employer address) and family demographics information (e.g. family address and house types), and were able to complete online travel survey every day. Data were collected from November 2016 and lasted for two months. Totally, 22,095 travel records were collected. An example of the travel data are provided in Table 1.

Table 1. Example of household travel survey

Trip ID	Departure time	Arrival time	Origin	Destination	Mode	Purpose
1	7:30	8:16	A	B	Metro	Work
2	12:00	12:20	B	C	Walk	Social
3	18:03	19:06	B	A	Metro	Home

The transit smart card data used in this study has both subway and bus transit data between November 15, 2016 and December 15, 2016. The data includes IDs of the cards, type of a card, consumption amount, bus route/subway route, and time. Samples of the transit smart card data are provided in Table 2.

In our smart-phone-based travel survey, we designed a record that volunteers can provide their transit smart card number, so that their travel survey records could be

Table 2. Example of transit smart card

Card ID	Arrival time	Money	Paid amount	Time	Route
1632460	31	3	2.4	1156	306
1632461	31	2	1.6	44260	309
1632462	21	0	0	29648	Line three
1632462	22	3	2.85	30172	Line three

connected to the transit smart data. Therefore, these transit smart data were associated with individual travel destination attributes. In this study, we named this dataset as SC-HTS (smart card-household travel survey) dataset.

3 Method

3.1 Characteristics Selection and Description

SC-HTS data include public bus and subway travel records. Bus related data only contain getting on time, while subway swipe card data include both passenger getting on and off time. To more accurately distinguish travel purposes, different purposes are classified by travel characteristics, specifically, this study selects boarding time, activity durations and the frequency of visiting a place.

Boarding time. The analysis of boarding time is the first characteristic to represent different travel purposes. From statistics results, timeslot 8–10 am is mainly for work commuting. Timeslot 5–7 pm is mainly for going home from work. Apart from off work, shopping and social entertainment are major activities in the evening. This represents most people's daily activities. Since there is no public transit service between 0–4 am, data in that period is not available (Fig. 1).

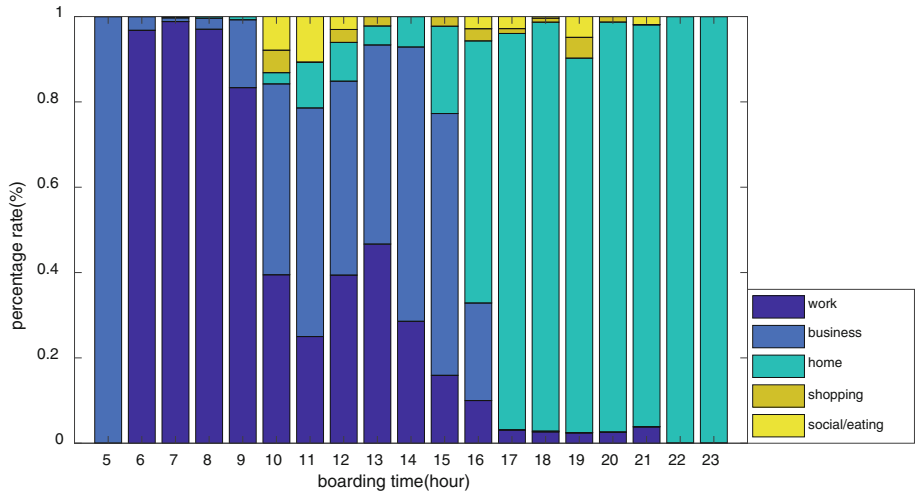


Fig. 1. Distribution of boarding time

Activity durations. Time lag between two transit smart card records is the second characteristic. Usually, transferring between bus or subway is short, while the duration between going to work and off work is normally around 8 h. It takes longer time between off work on the day and passengers go to work next morning. Figure 2 show the distribution of different activity durations in the SC-HTS dataset.

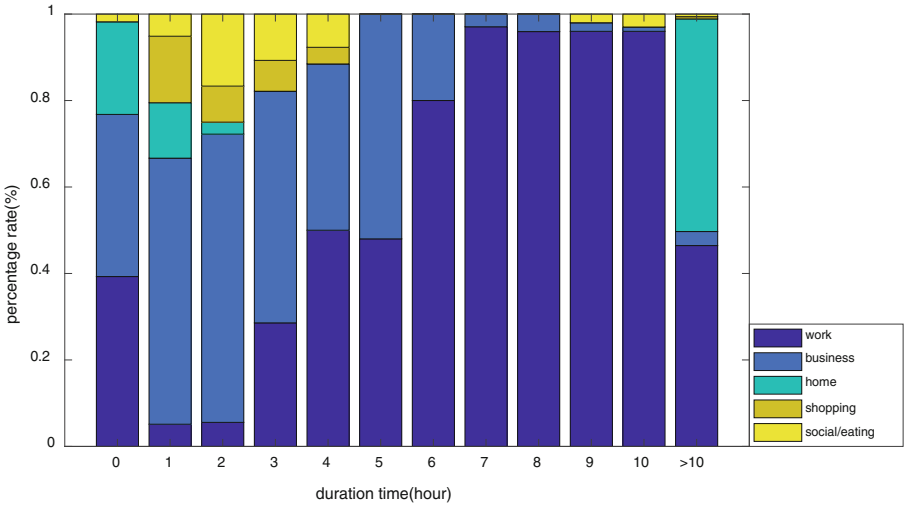


Fig. 2. Distributions of different activity durations.

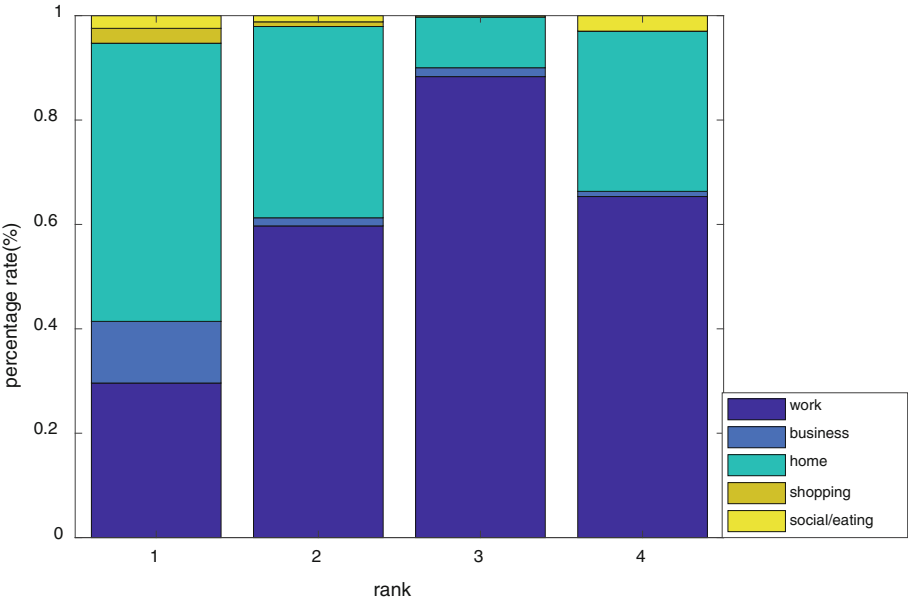


Fig. 3. Distribution of frequency of visit place

The frequency of visiting the same place. Frequency of visiting the same place is the third characteristic used in this study to depict travel purposes. In Fig. 3, The rank of 1, 2, 3 and 4 respectively representing visit the same places were 0–25%, 25%–50%, 50%–70%, 70%–100%.

3.2 Estimation of Probability Functions

Let c denote the set of travel purpose, c ∈ {'work', 'business', 'home', 'shopping', 'social/eating'}, and F denote the attributes in the SC-HTS dataset: $F = \{f_b, f_d, f_f\}$, where the attributes f_b stands for boarding time, the variable f_d is defined as an interval between the alighting and the next boarding time, f_f stands for the frequency of access to the same origin and destination.

In this study, we treated c and each element of F as discrete variables. By using Bayes' theorem, p(c|F) is described by

$$\mathrm{p(c|F)} = \frac{1}{p(F)} p(c) \prod_{k=1}^{k} p(f_k|c) \tag{1}$$

Where:

$p(c)$: the estimated travel destination probability distribution from the SC-HTS data set
$p(F)$: the probability distribution of travel characteristics estimated from the SC-HTS data set
$p(f_k|c)$: the probability distribution of various travel characteristics under different travel purposes.

When the F attribute of each trip is observed from the transit smart data, Bayesian classifier is used to estimate the purpose of each trip c. The classifier's equation is:

$$\hat{c}(F) = arg \max_{c \in C} p(c|F) \tag{2}$$

where C is a set of all the possible values of c.

By using Eq. (2), the number of trips with attribute c is described by

$$\mathrm{N(c)} = \sum\nolimits_{F \in S} \delta(c, F) N_s(F) \tag{3}$$

where

$$\delta(c, F) = \begin{cases} 1 & if\ \hat{c}(F) = c \\ 0 & otherwise \end{cases} \tag{4}$$

$N_s(F)$ is the number of trips with a vector of attributes F that is observed by SC-HTS, S is a set of all the possible values of F.

In the equation, $\delta(c, F)$ is the estimated number of trips for each purpose. When the selected classification feature F could not explain c, the travel purpose may be

misclassified because the purpose of the trip with the highest probability will be added to most of the itinerary.

4 Empirical Results

Data are classified into training set and testing set in the SC-HTS dataset. 70% data are training set by the Bayes model, and 30% of data are testing set. The training results are illustrated in Fig. 4. Among the five types of travel purposes, "work" and 'home' have the highest precision and recall rates. 97.2% of the travel purpose is go to work and the interval is usually above 8 h. 95.9% of travel is for going home from work. The Percentage of identified business about 60%. This activity often happens during working hours, and the frequency of visits is not high. shopping and social/eating recall rate is relatively low, respectively. Such travel purposes are similar and the sample of dataset is not enough.

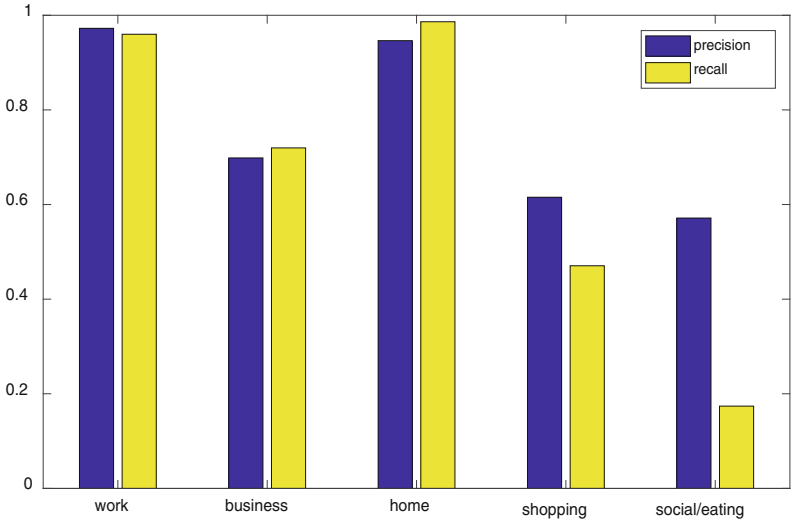


Fig. 4. The training results of Bayes model

Figure 5 shows the validation results. The accuracy of working and returning identification is still the highest, reaching more than 95%, while the precision of the other three are around 60%. The result shows that the model has good practicability.

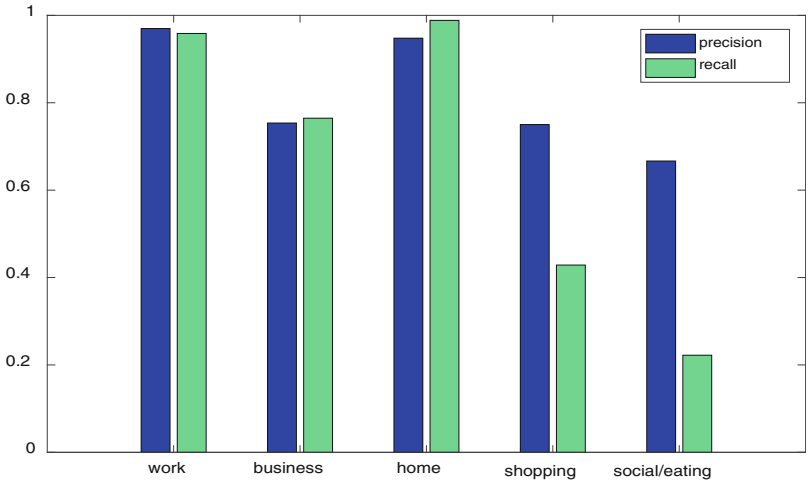


Fig. 5. The validation results of Bayes model

5 Conclusions

This study illustrates that it is feasible to use Bayes probabilistic model to infer travel purposes for transit smart card data to enhance attributes of the transport big data. The accuracy of inferring commuting purpose is quite satisfied which can reach 95%, which the accuracy of inferring other activities is about 60%. And more effort is need to improving the accuracy of non-commuting activities by considering other characteristics such as individual social-economic attributes.

References

1. Pas, E.I., Koppelman, F.S.: An examination of the determinants of day-to-day variability in individuals' urban travel behavior. Transportation **14**(1), 3–20 (1987)
2. Axhausen, K.W., Zimmermann, A., Schönfelder, S., et al.: Observing the rhythms of daily life: a six-week travel diary. Transportation **29**(2), 95–124 (2002)
3. Murakami, E., Wagner, D.P.: Can using global positioning system (GPS) improve trip reporting? Transp. Res. Part C Emerg. Technol. **7**(2), 149–165 (1999)
4. Asakura, Y., Hato, E.: Tracking survey for individual travel behaviour using mobile communication instruments. Transp. Res. Part C Emerg. Technol. **12**(3), 273–291 (2004)
5. Bagchi, M., White, P.R.: The potential of public transport smart card data. Transp. Policy **12** (5), 464–474 (2005)
6. Utsunomiya, M., Attanucci, J., Wilson, N.: Potential uses of transit smart card registration and transaction data to improve transit planning. Transp. Res. Rec. J. Transp. Res. Board **1971**, 119–126 (2006)
7. Seaborn, C., Attanucci, J., Wilson, N.: Analyzing multimodal public transport journeys in London with smart card fare payment data. Transp. Res. Rec. J. Transp. Res. Board **2121**, 55–62 (2009)

8. Wolf, J., Guensler, R., Bachman, W.: Elimination of the travel diary: experiment to derive trip purpose from global positioning system travel data. Transp. Res. Rec. J. Transp. Res. Board **1768**, 125–134 (2001)
9. Schönfelder, S., Axhausen K.W., Antille, N., et al.: Exploring the potentials of automatically collected GPS data for travel behaviour analysis (2002)
10. Stopher, P.R., Greaves, S.P., FitzGerald, C.: Developing and deploying a new wearable GPS device for transport applications. In: 28th Australian Transport Research Forum, 28 September–30 September 2005
11. Stopher, P., Clifford, E., Zhang, J., et al.: Deducing mode and purpose from GPS data. Inst. Transp. Logist. Stud. 1–13 (2008)
12. Stopher, P., FitzGerald, C., Zhang, J.: Search for a global positioning system device to measure person travel. Transp. Res. Part C Emerg. Technol. **16**(3), 350–369 (2008)
13. Chen, C., Gong, H., Lawson, C., et al.: Evaluating the feasibility of a passive travel survey collection in a complex urban environment: lessons learned from the New York City case study. Transp. Res. Part A Policy Pract. **44**(10), 830–840 (2010)
14. Chen, C., Bian, L., Ma, J.: From traces to trajectories: how well can we guess activity locations from mobile phone traces? Transp. Res. Part C Emerg. Technol. **46**, 326–337 (2014)
15. Gong, H., Chen, C.: Automating web collection and validation of GPS data for longitudinal urban travel studies (2012)
16. Bohte, W., Maat, K.: Deriving and validating trip purposes and travel modes for multi-day GPS-based travel surveys: a large-scale application in The Netherlands. Transp. Res. Part C Emerg. Technol. **17**(3), 285–297 (2009)
17. Shen, L., Stopher, P.R.: A process for trip purpose imputation from Global Positioning System data. Transp. Res. Part C Emerg. Technol. **36**, 261–267 (2013)

A Hybrid Music Recommendation System Based on Scene-State Perception Model

Zhixuan Liang(✉), Zehao Tan, Zhenyue Zhuo, and Xi Zhang

College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
jay470790400@gmail.com

Abstract. In recent years, the recommendation based on mobile users and the one based on context-aware have become popular topics in the field of the recommendation system. However, most of the music platforms need manual annotation of user scene which means if the user forgets to do that, the recommendation system may fail to work. In this paper, we propose a scene-sensing model based on Naive Bayesian classification which can be used to automatically locate the users' scene and predict their state in real time. Exactly established on the basis of user scene and life state, we propose a hybrid music recommendation system which combines the recommendation result of SVD++ collaborative filtering model and logical regression model which is used to predict the most recent popular music. Experimental results indicates that the hybrid recommendation system perform well on mobile users.

Keywords: Collaborative filtering · SVD++ · Scene perception
Regularized logistic regression · Hybrid recommender system

1 Introduction

As the consequence of the information age, users have found it difficult to search for helpful information from the huge data because the increasing number of information resources has caused the problem of information overload. Personalized recommendation system is one of the effective means to solve such problem. The traditional recommendation system mainly includes the following ones, collaborative filtering (CF) recommendation [1], content-based recommendation [2] and hybrid recommendation [3]. Collaborative filtering recommended by establishing a binary relationship between the user and the project and calculate the similarity between users which can be used to recommend according to similar useful similarity hobbies principle of personalized recommendation. Although Collaborative filtering recommendation technology has been widely used and has a good recommendation effect. However, some problems such as cold start problem and data sparse happen to it for its reliance on user behavior records. Unlike collaborative filtering recommendation, content-based recommendation learns the user's interests and preferences through feature extraction of items and analysis of user behavior records, so as to make a personalized recommendation. For content-based recommendation, it does not need a lot of user data, but

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 19–26, 2018.
https://doi.org/10.1007/978-3-319-73830-7_3

the characteristics of items are difficult to be selected automatically and the information needs to be manually entered.

In recent years, the recommendation based on mobile users and the one based on context-aware [4] have become popular topics in the field of the recommendation system. By analyzing the relationship between user context and favorite songs, it can solve the cold start problem of new users. However, most of the music platforms need manual annotation of user scene. Once the situation changes, the information about the scene has to be updated by the user. For example, the user changes from work to off duty, and the location shifts from office to living room. As for mobile users, the scene may change at any time. If each change requires the user to provide the scene information again, or if the user forgets to do that, the recommendation system may fail to work.

In this paper, we propose a scene-sensing model based on Naive Bayesian [5] classification which can be used to automatically locate the users' scene and predict their state in real time. Exactly established on the basis of user scene and life state, we design a hybrid music recommendation system which combines the advantages of SVD [6] collaborative filtering model and logical regression [7] model. Finally, we collect the data of 300 users for the experiment. The results indicate that the predictive accuracy about user scene of this algorithm can reach to 86.4%, and the accuracy of the hybrid recommendation system is also higher than that of the traditional one (Fig. 1).

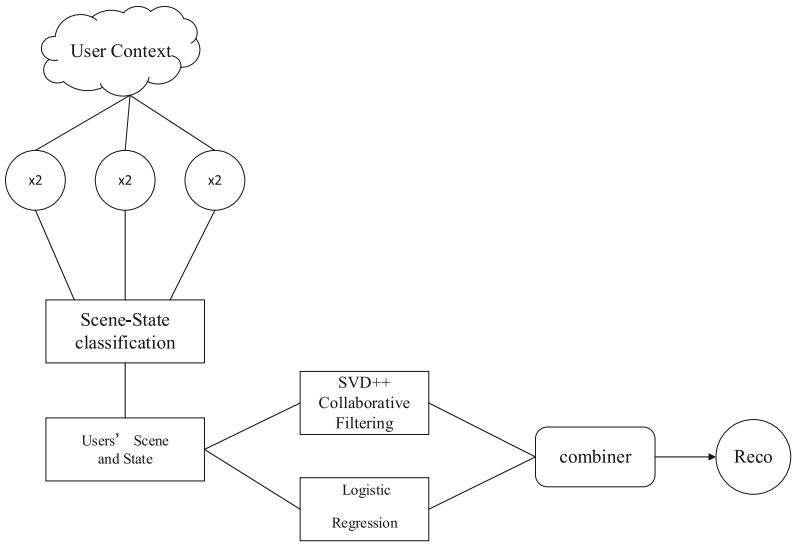


Fig. 1. System framework of scene-sensing hybrid recommendation

2 Problem Definition

Formally, given a sets of users $U = \{u = 1, \ldots, U\}$, a set of items $I = \{i = 1, \ldots, I\}$, a scenes groups $S = \{s = 1, \ldots, S\}$ and a log of users' past preferences of items

$O = \{u, i, s_{ui}, y_{ui}\}$, denoting user u has a positive feedback on item i at scene s_{ui}, our goal is to learn users' preferences from the positive feedback and recommend to each user u a list of items that will maximize her of his satisfaction. In this case, the user's feedback y_{ui} can be rated on a scale from 1 to 5. The more user enjoys the items, the higher ratings the items get.

2.1 User Scene-State Perception Model

It is well known that people's life and work is regular. For example, if one person was working in the office at last Monday then it seemed probably that he will do the same thing at this Monday if his location is in the office. Consequently, we design a Scene-State Perception model which is based on Naive Bayesian classifier to predict user's scene and state. Mathematically, the prediction can be presented as follow,

$$P(\vec{s}|f_1, f_2, \ldots, f_n) \propto P(\vec{s}) \prod_{i=1}^{n} P(f_i|\vec{s}) \tag{1}$$

$$\mathbf{S}_{\text{map}} = \arg\max_{\vec{s} \in \mathbf{S}} P(\vec{s}|f_1, f_2, \ldots, f_n) \tag{2}$$

where $\vec{s} = (s_1, s_2) \in S = \{S_1, S_2, \ldots, S_k\}$ (scene-state class label) is a vector denotes user's scene and state, $f_i \in \mathbf{F} = (\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_n)$ (assume the independence of attributes) is one of user's context feature such as user's location, time, weather and so on. $\boldsymbol{P}(\vec{s}|\, \boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_n)$ is the *class-conditional probability distribution (CPD)* which indicate the possibility of user's scene and state.

In general, when calculate the probability of a scene-state class, we often add *Laplace Smoothing* [8] parameter in order to avoid the probability of 0 cases. The modified formulation is presented as follow,

$$P(y = \boldsymbol{S_k}) = \frac{\sum_{i=1}^{N} I(y_i = \boldsymbol{S_i}) + \lambda}{N + K\lambda} \tag{3}$$

where λ is the *Laplace Smoothing* parameter (in this case its' value is 1), N is the total number of user history records and K is the total number of scene-state class.

2.2 Singular Value Decomposition

In recent years, Singular Value Decomposition (SVD) which can decompose high dimensional data into low dimensional data has been used widely in collaborative filtering technique. It decompose user-item matrix into two singular matrixes which indicate the latent relationship between user and item. Based on the traditional SVD, RSVD [9] consider the regularization constraints to avoid overfitting and SVD++ [10] make use of user's history preference which can significantly improve the recommended accuracy. In SVD++, the prediction and matrix factorization can be formulated as follow,

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T(p_u + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j) \tag{4}$$

where $\hat{r}_{ui}$ is the prediction, μ is the baseline predictors [11] of total user preference, b_i and b_u respectively denote the preference bias of index user and index item, q_i and p_u are the singular latent matrixes, $R(u)$ is the user history record and y_j is preference feedback.

The optimization equation can be presented as follow,

$$\min_{b_*, q_*, p_*} \sum (r_{ui} - \hat{r}_{ui})^2 + \lambda_2 (b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2) \tag{5}$$

Where λ_2 is the learning rate and $\lambda_2(b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2)$ is the regularization term used to avoid overfitting. The optimization can be solved by gradient descent algorithm [12].

2.3 Regularized Logistic Regression

Logistic Regression is a widely used generalized linear algorithm which can be used to predict probabilities in machine learning. In this paper, we applied regularized logistic regression to predict user's preference on songs. The model definition and objective function are formulated as follow,

$$\mathrm{h}(\theta) = \frac{1}{1 + e^{-\theta^T \cdot X}} \tag{6}$$

where X is the input which in this case are the features of songs tag artificially, θ are the weight matrix indicated user's preference, $\mathrm{h}(\theta)$ is the probability of prediction.

The loss function is presented as follow,

$$\mathrm{J}(\theta) = -[\frac{1}{m} \sum_{i=1}^{m} \left(y^{(i)} \log\left(h_\theta\left(x^{(i)} \right) \right) + \left(1 - y^{(i)} \right) \log\left(1 - h_\theta\left(x^{(i)} \right) \right) \right) + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2 \tag{7}$$

Where m is the batch size of training set, $y^{(i)}$ (1 or 0) preference and λ is the learning rate. Also, this optimization problem can be solved in gradient descent algorithm.

3 Hybrid Recommendation System Work

The framework of our music hybrid recommendation system can be summarized as follows:

Step 1: Through the mobile phone GPS positioning to obtain the user's location information and use Scene-State perception model to predict user's scene and state (e.g. driving on the way).

Step 2: According to user's scene and state, using SVD++ collaborative filtering to provide a personal ranked list of songs.
Step 3: Basic of user's scene and state, using logistic regression model to rank a list of songs which are most recent popular in such scene and state.
Step 4: Making a combine with the ranking list of songs provided by step 2 and step 3.
Step 5:According to the feedback of user's preference, adjusting the learning parameters in SVD++ and logistic regression (Fig. 2).

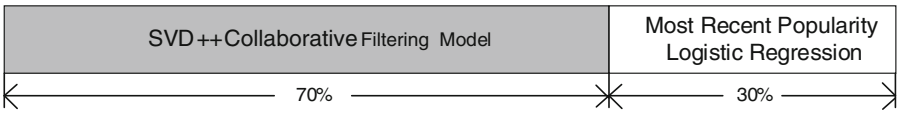


Fig. 2. The proportion of SVD++ and most recent popularity logistic regression

4 Experimental Results

Our experimental evaluation consists of three part. First, we verify of the accuracy of Scene-State classification based on Naïve Bayes. Second, we evaluate the performance of our hybrid recommendation system and compare it with other recommendation technology. Third, we study the proportion of SVD++ and most-recent-popularity logistic regression which can impact the recommendation performance.

4.1 Data Sets and Metrics

In our experiment, we use two data sets and the data are collected from our web application and mobile application. The two data sets approximately contain 40000 songs, 300 users' preference and 500 record. Firstly, we use 60% of the feedback for training, 20% of it for cross validation and the last 20% for testing. For each data set, we consider user ratings that consists of more than 4 stars as positive preference and the ones less than 3 stars as negative feedback. The metrics about our experiment is presents as follow,

$$\text{Precision@N} = \frac{|C_{N,rec} \cap C_{adopted}|}{N} \tag{8}$$

$$\text{Recall@N} = \frac{|C_{N,rec} \cap C_{adopted}|}{C_{adopted}} \tag{9}$$

4.2 Main Result

Table 1 shows the results of Scene-State recognition data set. When N is 100 and Number of scene class is 5 as well as Number of state class is 4, the accuracy of scene-state recognition model is 67.8%. If N changes from 100 to 200, the accuracy of

Table 1. Precision of scene-state recognition model

N	Number of scene class	Number of state class	Precision
100	5	4	67.8%
200	5	4	73.6%
400	5	4	77.5%
800	6	5	84.8%
1000	6	5	86.4%

it turns to 73.6% (performing bad). After increasing the capacity of data set, the accuracy of scene-state recognition model can reach up to 84.8% (N is 800). When N is 1000, the accuracy of it is as high as 86.4%, which performs better than smaller data sets. However, the growth rate of precision becomes more and more insignificant when increasing the data set.

Figure 3 presents the precision of various recommendation technology. Note that the SVD model reaches a maximum of 28% precision and RSVD model achieves 33% precision while hybrid recommendation system which combines SVD++ and logistic regression can reaches 48% precision.

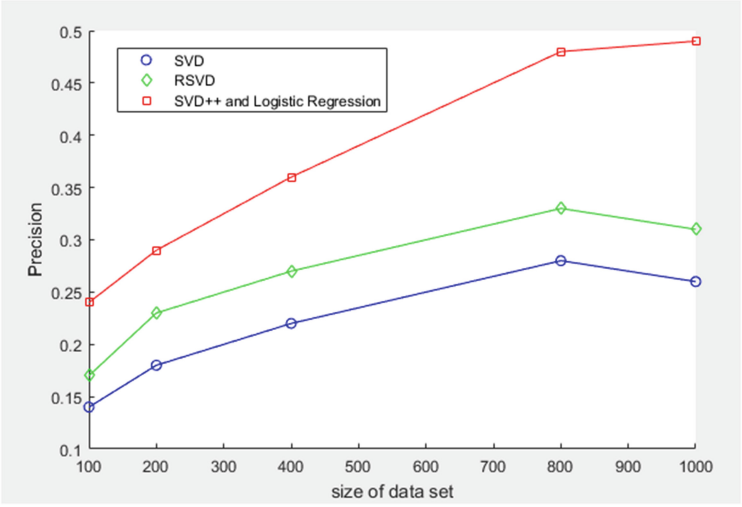


Fig. 3. The precision of various recommendation method on mobile users

Figure 4 shows the recall of various recommendation technology. According to experimental results, the hybrid recommendation system which combines SVD++ model and logistic regression model can reaches a maximum 0.26 recall, compared with SVD model which achieves 0.14 recall and RSVD model which reaches 0.19 recall. Thus, in a general, the hybrid recommendation system performs better on mobile users.

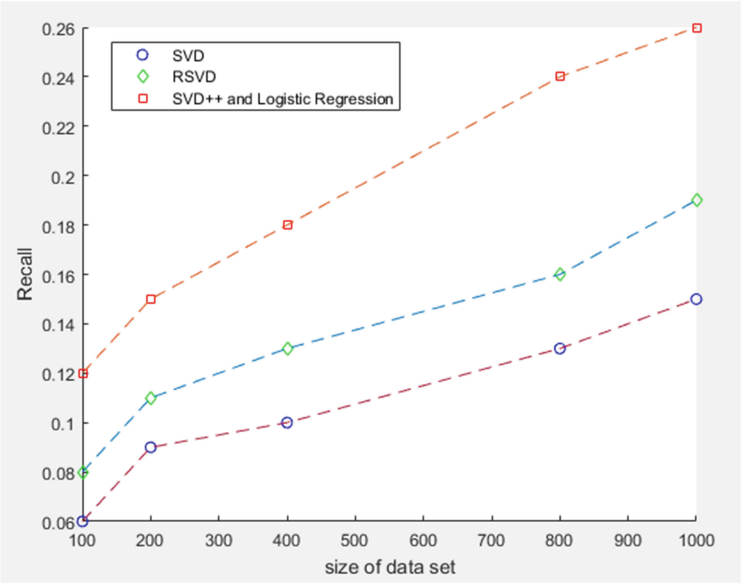


Fig. 4. The recall of various recommendation method on mobile users

5 Conclusion

In this paper, we propose a hybrid recommendation based on user scene-state recognition model, SVD++ collaborative filtering and logistic regression model. Experimental results show that scene-state precision model based on Naïve Bayes classification can accurately predict user's scene-state and the hybrid recommendation system perform efficiently on mobile user.

For future work, we are interested in scene-state clustering and plan to extend our investigation to time SVD++ collaborative filtering model and other machine learning model in order to improve recommendation quality on mobile user.

References

1. Schafer, J.B., Dan, F., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. ACM Trans. Inf. Syst. **22**(1), 5–53 (2004)
2. Popescul, A., Ungar, L.H., Pennock, D.M., Lawrence, S.: Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments, pp. 437–444. Eprint Arxiv arXiv:1301.2303 (2001)
3. Yu, L., Liu, L., Li, X.: A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-commerce. Expert Syst. Appl. **28**(1), 67–77 (2005)
4. Shin, D., Lee, J.W., Yeon, J., Lee, S.G.: Context-aware recommendation by aggregating user context. In: IEEE Conference on Commerce & Enterprise Computing, pp. 423–430 (2009)
5. Kononenko, I.: Semi-naive Bayesian classifier. In: Kodratoff, Y. (ed.) EWSL 1991. LNCS, vol. 482, pp. 206–219. Springer, Heidelberg (1991). https://doi.org/10.1007/BFb0017015

6. Kim, J.K., Cho, Y.H.: Using web usage mining and SVD to improve E-commerce recommendation quality. In: Lee, J., Barley, M. (eds.) PRIMA 2003. LNCS, vol. 2891, pp. 86–97. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39896-7_8
7. Ochmański, E., Pieckowska, J.: The logical foundations of goal-regression planning in autonomous agents. Artif. Intell. **106**(2), 267–334 (1998)
8. Field, D.A.: Laplacian smoothing and Delaunay triangulations. Commun. Appl. Numer. Methods **4**(6), 709–712 (1988)
9. Szwabe, A., Ciesielczyk, M., Janasiewicz, T.: Semantically enhanced collaborative filtering based on RSVD. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011. LNCS (LNAI), vol. 6923, pp. 10–19. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23938-0_2
10. Kumar, R., Verma, B.K., Rastogi, S.S.: Social popularity based SVD++ recommender system. Int. J. Comput. Appl. **87**(14), 33–37 (2014)
11. Norris, K.C., Greene, T., Kopple, J., Lea, J., Lewis, J.: Baseline predictors of renal disease progression in the African American study of hypertension and kidney disease. J. Am. Soc. Nephrol. **17**(10), 2928–2936 (2006)
12. Mandic, D.P.: A generalized normalized gradient descent algorithm. Signal Process. Lett. IEEE **11**(2), 115–118 (2004)

Research on Parallel Architecture of OpenCL-Based FPGA

Yi Zhang(✉), Ye Cai, and Qiuming Luo

College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
2150230429@email.szu.edu.cn

Abstract. Moore's law encounters a bottleneck today. Computing power of the general purpose processor is restricted. At the same time, new types of enterprise computing such as big data management and analysis bring more challenges to the computational performance and scalability of the data center. Research efforts have been devoted to accelerating algorithm on Field Programmable Gate Arrays (FPGAs), due to their high performance and reprogramming. In this paper, we first study the heterogeneous platform of OpenCL-based FPGA, and propose a novel multi-computing unit combined with internal hardware flow parallel acceleration framework. Then, we evaluate the influences of different number of computing units on performance and resource utilization with the high performance computing applications (AES algorithm) that implemented through the proposed framework. Meanwhile, we compare the performance with CPU implementation. The result shows that our proposed framework has advantages of high performance and scalability for the implementation of a class of algorithms suitable for parallelization, and suits for the demands of data center and high performance computing (HPC) applications.

Keywords: Heterogeneous computing · FPGA · OpenCL
High performance computing · Scalability

1 Introduction

Currently, new types of enterprise computing such as big data management, machine learning and artificial intelligence bring more requirements to the computational performance of the data center. General purpose processor has failed to meet the computational requirements by increasing the clock frequency and the number of cores, stemming mainly from the so called power consumption wall and Von Neumann bottleneck [1]. It has become an inevitable trend that heterogeneous computing platform is used to provide high performance [2–4]. Heterogeneous architectures essentially consist of a combination of CPU and a variety of hardware accelerators to speed up the execution of computationally intensive tasks [5].

Graphics Processing Unit (GPU) and FPGA are currently two mainstream hardware accelerators. GPUs offer higher floating-point peak performance than FPGAs. However, GPU-based accelerators are severely limited in the data center because it is not efficient in terms of power consumption [6]. FPGAs in comparison to GPUs can

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 27–38, 2018.
https://doi.org/10.1007/978-3-319-73830-7_4

provide reasonable processing speed while consuming only a fraction of their operating power [7]. Meanwhile, FPGA s allow for the creation of custom hardware solutions for the acceleration of algorithm, due to their reprogramming. These capabilities of FPGAs have been acknowledged by several big data companies such as Microsoft and Baidu. It is evident that they would use FPGAs rather than GPUs as accelerators in their data centers [8, 9].

However, in the aspect of using FPGA to accelerate algorithm, the traditional way based on hardware-centric maps the algorithm to FPGA in hardware pipelining [10, 11]. The way to program FPGAs has been through the use of hardware description languages (HDLs) such as Verilog and VHDL, leading to speeding a long development cycle. Moreover, the programmability issues of HDLs raise serious concerns on code maintenance and scalability. Theses limitation however can be tackled by a technique called high-level synthesis (HLS). HLS enables designers to program an FPGA using high-level languages e.g. C, C++, OpenCL [12]. The OpenCL specification defines a single programming model and a set of system-level abstractions supported by all hardware platforms conforming to the standard. For example, FPGA vendors such as Altera [13] and Xilinx [14] have started to develop OpenCL SDKs for much better programmability.

OpenCL provides a good data parallel programming model to accelerate algorithm. What we concern about is how to use this programming model to accelerate the algorithm in the FPGA. Therefore, we study the abstract architecture of OpenCL-based FPGA, and propose a novel multi-computing unit combined with internal hardware flow parallel acceleration framework.

The rest of the paper is organized as follows. In Sect. 2 we list related work. In Sect. 3 we introduce the OpenCL programming framework. In Sect. 4 we present the abstract architecture of OpenCL-based FPGA and propose a parallel acceleration framework. We elaborate on the experiment design and the results analysis in Sect. 5 and conclude in Sect. 6.

2 Related Work

This section summarizes the development of FPGA chip technology such as the increase of on-chip resources and the emergence of high-level synthesis changed the accelerated method of algorithm in the FPGA.

Previously, the researchers focused on how to use the pipeline architecture to accelerate algorithm in the FPGA. They improve the performance of algorithm by increasing the stage of pipeline and optimizing the pipeline. Hodjat et al. [10] presented the architecture of a fully pipelined AES encryption processor on a single chip FPGA. Sukhsawas et al. [11] implemented a high performance pipeline FFT on Virtex-E FPGA.

Later, with the increase of resource capacity of FPGA, researchers use multiple parallel pipeline architecture instead of single one to accelerate algorithm. This way is efficient to improve the performance of algorithm, due to parallel and pipeline execution. Yang et al. [15] presented a parallel and pipeline processing method for block cipher algorithms. Palmer et al. [16] described a parallel FFT design suitable for

FPGA implementations. It consists of multiple, parallel pipelines with a front end butterfly-like circuit to preprocess the incoming data and distribute it to the parallel pipelines.

Currently, the technology of FPGA chip has made great progress [17]. Moreover, high-level synthesis language appears, such as OpenCL [12]. These features improve the performance of FPGA and development efficiency greatly. Researchers start to use architecture of OpenCL-based FPGA to accelerate algorithm. In the field of big data management and analysis. Shan et al. [18] presented a MapReduce framework on FPGA. Hussain et al. [19] proposed a highly parallel hardware design to accelerate the K-means clustering of microarray data by implementing the K-means algorithm in FPGA. Microsoft [8] use the FPGA to accelerate the Bing search engine. In the field of deep learning. Zhang et al. [20] implemented a CNN accelerator on FPGA. Han et al. [21] implemented an efficient speech recognition engine with sparse LSTM on FPGA. Researchers at Baidu are thus considering FPGAs for accelerating their deep learning models for image search [22]. This provided us with a motivation to perform an extensive study in this regard.

3 OpenCL Programming Framework

This section gives a brief overview of the OpenCL programming framework, including its platform, memory and execution model. It also explains how the OpenCL memory model is mapped to the FPGA.

The OpenCL is a parallel programming language that addressed the challenges of programming multi-core and heterogeneous compute platform [10]. The platform mode and memory model for OpenCL are defined in Fig. 1. The platform model consists of a host connected to one or more Compute devices. A (possibly multi-core) CPU is generally considered to be the host which is responsible for setting up the environment to enable the OpenCL kernels to execute on device. Compute device (such as CPU, GPU, FPGA and so on) can be used to accelerate the compute intensive portion of an algorithm i.e. the kernels. A compute device is divided into one or more compute units (CUs), such as compute unit *1* to *N* in Fig. 1. A CU is further divided into one or more processing elements (PEs), such as PE *1* to *M* in Fig. 1.

The OpenCL memory model the behavior and hierarchy of memory that can be used by OpenCL applications. In Fig. 1, the memory is broadly divided into host (i.e. CPU) memory and device (i.e. GPU or FPGA) memory. The device memory is further divided into private memory (specific to each PE), local memory (shared by all the PEs in a CU) and a global/constant memory (shared by all the CUs). For devices using an FPGA device, the physical mapping of the OpenCL memory model is the following:

- Host memory is any memory connected to the host processor only.
- Global/constant memory is any memory that is connected to the FPGA device. These are usually memory chips (e.g. SDRAM) physically connected to the FPGA device.

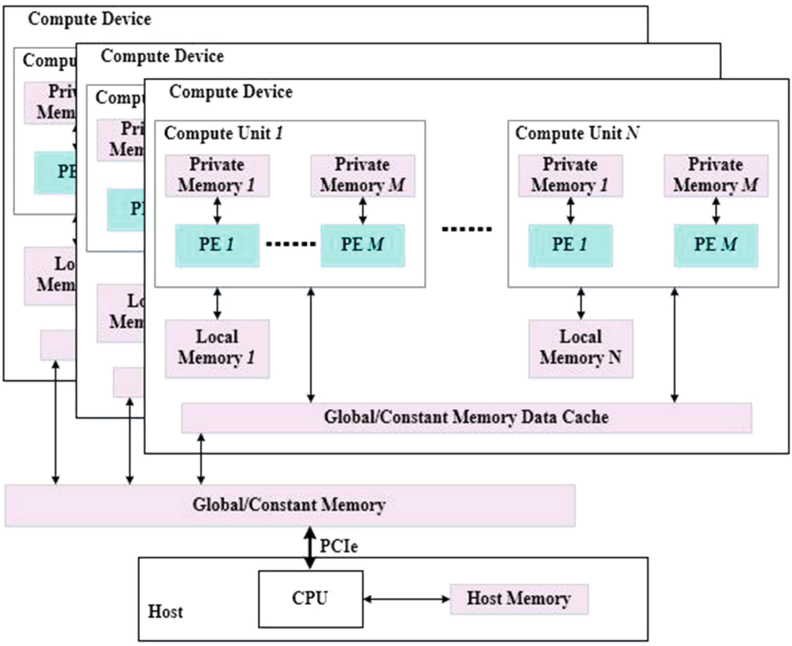


Fig. 1. The platform mode and memory model of OpenCL

- Both local memory and private memory are memory inside the FPGA device. This memory is typically implemented using registers or Block RAMs in the FPGA fabric.

The OpenCL execution model defines how kernels execute. In OpenCL a kernel is replicated and execute in parallel on multiple CUs. Each parallel run is assigned an ID which allows the kernel to work on a subset of data that is associated with it.

4 A Novel FPGA Architecture

This section introduces a parallel architecture of OpenCL-based FPGA. Under this architecture, we propose a novel multi-computing unit combined with internal hardware flow parallel acceleration framework. Meanwhile, we also explains how proposed framework accelerates the algorithm in parallel.

4.1 Overview of Architecture

The heterogeneous computing platform of OpenCL-based FPGA consists of the host-side CPU and device-side FPGA, CPU and FPGA communication through the PCIe bus, as shown in Fig. 2. We focus on FPGA in Fig. 2. The FPGA architecture is divided into static region and reconfigurable region. Static region is mainly composed of PCIe DMA controller model, Memory controller model, AXI Interconnect bus.

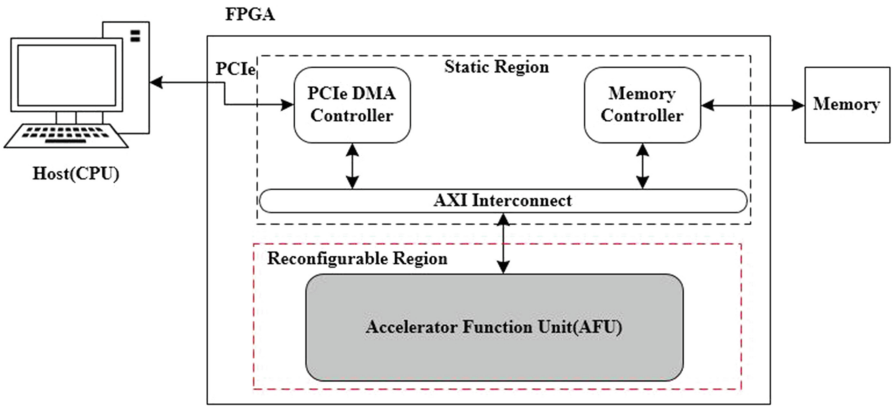


Fig. 2. Heterogeneous computing platform of OpenCL-based FPGA

Meanwhile, static region also contains all the necessary circuitry for communication between host, compute units, and off-chip global memory. This static region is pre-defined base platform which can be flashed onto an EPROM on the board. The FPGA would then be configured with this base platform upon power-up and is always there and accessible for the user. Reconfigurable region is also called accelerator function unit (AFU). The AFU contains the customized compute units which implement the user-defined accelerator kernels. In Sect. 4.2 below, we will further explain the AFU.

4.2 Parallel Acceleration Framework

The specific implementation of parallel acceleration framework is in the AFU mentioned in Sect. 4.1. Figure 3 shows the structure of the framework. The framework contains multiple customized CUs, such as CU *1* to CU *N* in Fig. 3. CUs are interconnected by AXI bus, and it implements the user-defined accelerator kernels. The

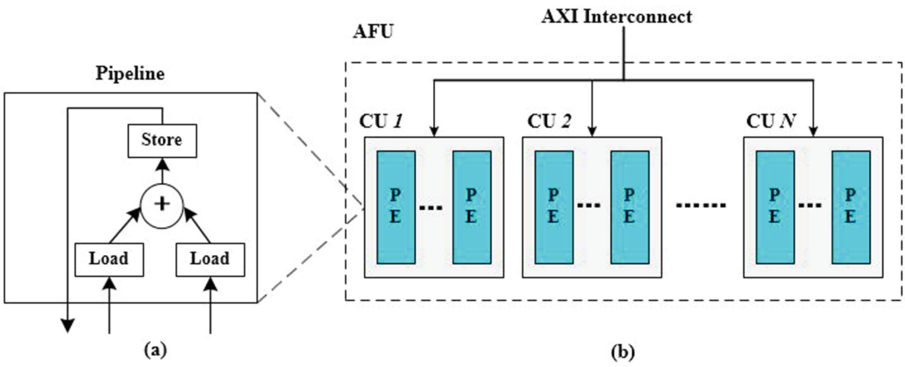


Fig. 3. Parallel acceleration framework

kernel is instantiated as a hardware executed flow (called pipeline) in the CU, as shown in Fig. 3a. Each CU contains multiple PEs that similar to hardware threads, as shown in Fig. 3b. Each PE executes an instance of the kernel in parallel. Therefore, the CUs can execute in parallel in terms of different work items (called PEs). That is, work items are assigned to CUs for executions in parallel. In this way, the massive amounts of parallelism available in the FPGA device can be customized and harnessed by proposed framework. This is different from CPU and GPU implementations of OpenCL which contain a fixed set of general purpose resources.

4.3 Acceleration of Data-Parallel

Based on the OpenCL programming framework combined with the parallel acceleration framework, we can accelerate a class of algorithms which is suitable for data-parallel. Figure 4 shows the way of data partition. When algorithm is executed in parallel, data is divided into lots of blocks, such as block*1*, block*2*, etc. Kernel of acceleration algorithm is instantiated into each CU. Each PE executed in parallel is assigned an ID which allows the PE to index a block of the data that is associated with it.

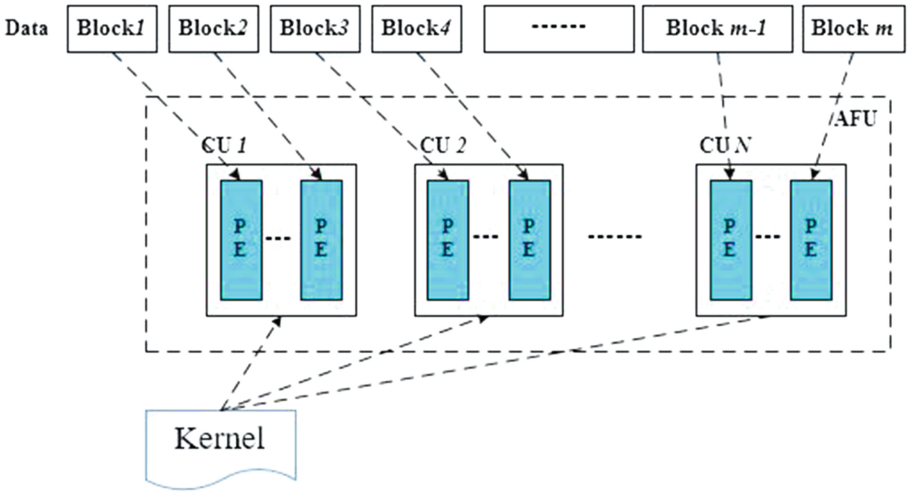


Fig. 4. Data partition

5 Experiment Design and Results Analysis

This section introduces the experiment platform. The experiment design uses a different number of CUs to accelerate AES algorithm. We evaluate the influences of different number of CU on performance and analyze the resource utilization of a CU. Meanwhile, then compare the performance with CPU implementation.

5.1 Experiment Platform

A Lenovo R680 G7 with quad Intel Xeon E7 2.67 GHz processors and 125 GB DDR3 was chosen as the host system. A Semption NSA-120 FPGA accelerator board with 500 MB DDR3 was connected through a second generation PCIe connection to the host system. The host system was running Linux Centos 6.8 and Xilinx SDAcell tool with SDK for OpenCL was installed on the host system in order to achieve communication and program the FPGA device.

5.2 Experiment Design

Firstly, in the OpenCL standard programming framework, Table 1 lists the encryption part of AES-128(ECB mode) algorithm pseudocode which is written as a kernel function executed in FPGA. After the OpenCL compiler instantiates the kernel to multiple CUs, it forms an execution flow of encryption part in each CU. The data to be encrypted is divided into 16 bytes per block. In a CU, each PE gets a block by the index of *idx* and executes the encrypt flow in parallel. Table 2 lists the host program pseudocode executed in CPU, and its execution order is as follows: (1) initial the context of between CPU and FPGA device, (2) send the data to be encrypted from the CPU to FPGA by the bus of PCIe, (3) start up each PE in FPGA to execute the encrypt flow for

Table 1. A kernel function for AES-128 algorithm

Input parameters: *in, out, keys, rounds*

// *in* points to the input buffer of the data to be encrypted as a pointer

// *out* points to the output buffer of the encrypted data as a pointer

// *keys* is the encryption key

// *rounds* is the round number for encryption

```
1:   idx = get_global_id();   // get the index number idx
2:   block = in[idx];   // get data to be encrypted by idx
3:   for (i⟵ 1 to (rounds – 1)
4:       block = SubBytes(block);          // execute the substitute bytes
5:       block = ShiftRows(block);         // execute the shift rows
6:       block = MixColumns(block)         // execute the mix columns
7:       block = Roundkey(block, keys)     // execute the add round key
8:   end for
9:   block = SubBytes(block)
10:  block = ShiftRows(block)
11:  block = Roundkey(block, keys)
12:  out[idx] = block   // store the encrypted data
```

Table 2. A host program executed in CPU

```
Input parameter: data
// data define the data to be encrypty
1: device_id = clGetDeviceIDs(); // get id number of the FPGA device
   // Create the context between CPU and FPGA
2: context = clCreateContext(device_id);
   // Create Input Buffer in FPGA and copy data from CPU to Input Buffer
3: in_buffer = clCreateBuffer(context, data);
   // Create output Buffer in FPGA and store the encrypted data
4: out_buffer = clCreateBuffer(context);
5: keys = ComputeRoundKey() ; //compute the encryption key
   // set the parameters of kernel
6: clSetKernelArg(kernl, in_buffer, out_buffer, keys, 10)
   // start up the PE to in FPGA to execute the encrypt flow for the data
7: clEnqueueNDRangeKernel(kernel);
   // read the encrypted data from the output Buffer in FPGA
8: encrypted_data = clEnqueueReadBuffer(out_buffer);
```

the data, (4) receive the encrypted data from FPGA to CPU. We use the xCU to accelerate the algorithm for 1 MB text data, where xCU means different number of CU and x = {1, 2, 4, 8, 9}. Meanwhile, we set the number of PE (PE = 1024) in each CU.

5.3 Experiment Result

(A) Throughput Analysis

Figure 5 shows the throughput ratio of accelerated algorithm with different number of CUs. From the Fig. 5 we can find that the throughput ratio is significantly improved when we increase the number of CU, because the data is divided into multiple CUs for encryption. When the number of CU is 8, the throughput ratio is the highest. When the number of CU is 9, we find that the change of throughput is very small. The main reason is that the resources of FPGA have been exhausted, and not all CUs are instantiated into the FPGA.

(B) Resource Utilization

The resource utilization for a CU is reported in Table 3. These fundamental blocks (FF, LUT, DSP, block RAM) are used to generate the custom logic for each CU in the design. We found that it consumed more blocks of FF and LUT for a CU to instantiate a kernel (AES_ECB_Encrypt) The number of each fundamental resource needed to

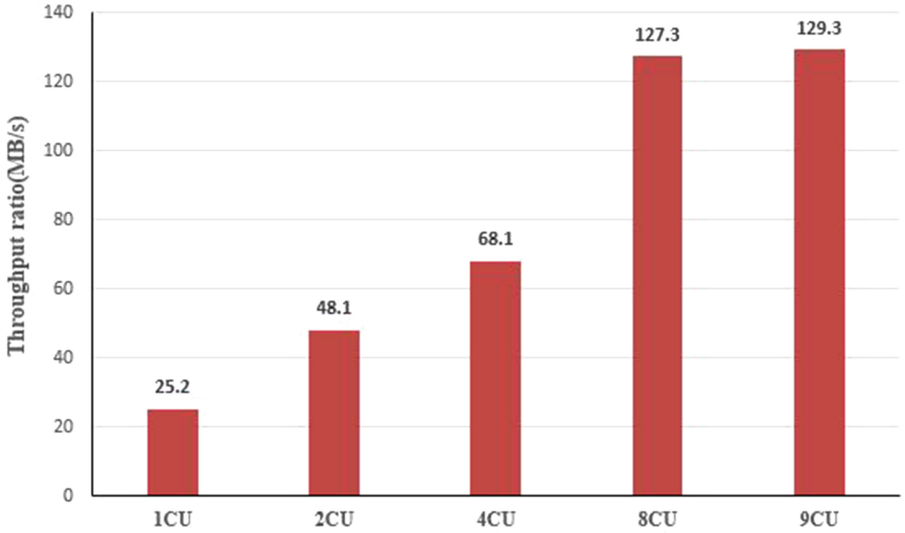


Fig. 5. Throughput ratio of accelerated algorithm with different number of CUs

Table 3. Fundamental resource for a CU

Compute unit	Kernel name	FF	LUT	DSP	BRAM
aes_instance	AES_ECB_Encrypt	4794	5424	8	14

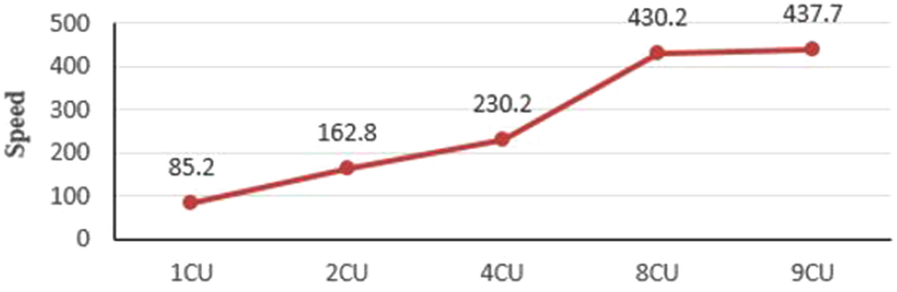
implement the custom logic in a CU determines how many CUs can be simultaneously loaded into the FPGA fabric.

(C) Performance Comparison

We run the serial AES algorithm in host-side CPU by single thread. When the algorithm executes to encrypt the 1 MB text data, the execution time of the FPGA implement and CPU implement is listed respectively in the Fig. 6a. When we use 8 CUs to accelerate the algorithm in FPGA, the execution of algorithm on FPGA is 437.7X faster than that of running on the CPU by single thread. Relative speedup can be seen in Fig. 6b.

	1CU	2CU	4CU	8CU	9CU
FPGA execution time(ms)	39.732	20.795	14.709	7.871	7.736
CPU execution time(ms)	3385.713				

(a)



(b)

Fig. 6. Execution time and Speed

6 Conclusions

The OpenCL programming framework has become a significant leap on high level synthesis of FPGAs. We researched the heterogeneous platform of OpenCL-based FPGA, and proposed a novel multi-computing unit combined with internal hardware flow parallel acceleration framework. In order to show the advantages of the proposed framework, we use the proposed framework to implement the acceleration of AES algorithm. The throughput analysis of the experiment result demonstrates that the proposed framework has good scalability and can be applied to different resources capacity of the FPGA. The more resources exit in the FPGA, the more CUs are instantiated to accelerate the algorithm in parallel. Moreover, the performance comparison of result also indicates that the algorithm implemented through proposed framework in the FPGA has higher performance than the CPU implementation. In a word, our proposed framework has advantages of high performance and scalability for the implementation of a class of algorithms suitable for parallelization, and suits for the demands of data center and high performance computing (HPC) applications.

Acknowledgement. The research was jointly supported by project grant from Shenzhen Science &Technology Foundation: JCYJ20150930105133185/JCYJ20170302153920897, National Natural Science Foundation of China: NSF/GDU1301252, and the higher education reformation project of Guangdong Provincial Department of Education: "Research on teaching reform of computer hardware series lessons based on system view", 20150819.

References

1. Esmaeilzadeh, H., Blem, E., Amant, R.S., Sankaralingam, K., Burger, D.: Dark silicon and the end of multicore scaling. IEEE Micro **32**, 122–134 (2012)
2. Vestias, M., Neto, H.: Trends of CPU, GPU and FPGA for high-performance computing. In: International Conference on Field Programmable Logic and Applications, pp. 1–6 (2014)
3. Gai, K., Qiu, M., Zhao, H., Tao, L., Zong, Z.: Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. J. Netw. Comput. Appl. **59**, 46–54 (2016)
4. Gai, K., Qiu, M., Zhao, H.: Energy-aware task assignment for mobile cyber-enabled applications in heterogeneous cloud computing. J. Parallel Distrib. Comput. **111**, 126–135 (2017)
5. Horowitz, M.: 1.1 Computing's energy problem (and what we can do about it). In: Solid-State Circuits Conference Digest of Technical Papers, pp. 10–14 (2014)
6. Nurvitadhi, E., Sheffield, D., Sim, J., Mishra, A., Venkatesh, G., Marr, D.: Accelerating binarized neural networks: comparison of FPGA, CPU, GPU, and ASIC. In: International Conference on Field-Programmable Technology, pp. 77–84 (2017)
7. Muslim, F., Liang, M., Roozmeh, M., Lavagno, L.: Efficient FPGA implementation of OpenCL high-performance computing applications via high-level synthesis. IEEE Access. **PP**, 1 (2017)
8. Putnam, A., Caulfield, A.M., Chung, E.S., Chiou, D.: A reconfigurable fabric for accelerating large-scale datacenter services. In: ACM/IEEE International Symposium on Computer Architecture, pp. 13–24 (2014)
9. Ouyang, J.: SDA: software-defined accelerator for large-scale deep learning system. In: International Symposium on VLSI Design, Automation and Test, p. 1 (2016)
10. Hodjat, A., Verbauwhede, I.: A 21.54 Gbits/s fully pipelined AES processor on FPGA. IEEE (2004)
11. Sukhsawas, S., Benkrid, K.: A high-level implementation of a high performance pipeline FFT on Virtex-E FPGAs. In: Proceedings of the IEEE Computer Society Symposium on VLSI, pp. 229–232 (2004)
12. Stone, J.E., Gohara, D., Shi, G.: OpenCL: a parallel programming standard for heterogeneous computing systems. Comput. Sci. Eng. **12**, 66–73 (2010)
13. Czajkowski, T.S., Aydonat, U., Denisenko, D., Freeman, J.: From opencl to high-performance hardware on FPGAS. In: International Conference on Field Programmable Logic and Applications, pp. 531–534 (2012)
14. Guidi, G., Reggiani, E., Di Tucci, L., Durelli, G., Blott, M., Santambrogio, M.D.: On How to improve FPGA-based systems design productivity via SDAccel. In: Proceedings of the IEEE 28th International Parallel Distributed Processing Symposium Workshops, IPDPSW 2014, pp. 247–252, August 2016
15. Yang, Y.S., Bahn, J.H., Lee, S.E., Bagherzadeh, N.: Parallel and pipeline processing for block cipher algorithms on a network-on-chip. In: Sixth International Conference on Information Technology: New Generations, pp. 849–854 (2009)
16. Palmer, J., Nelson, B.: A Parallel FFT architecture for FPGAs. In: Proceedings of the Field Programmable Logic and Application, International Conference, FPL 2004, Leuven, Belgium, 30 August–1 September, pp. 948–953 (2004)
17. Kumar, A., Verma, G., Nath, V., Choudhury, S.: IC Packaging: 3D IC Technology and Methods (2017)
18. Shan, Y., Wang, B., Yan, J., Wang, Y., Xu, N., Yang, H.: FPMR: MapReduce framework on FPGA. In: ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA 2010, Monterey, California, USA, pp. 93–102, February 2010

19. Hussain, H.M., Benkrid, K., Seker, H., Erdogan, A.T.: FPGA implementation of K-means algorithm for bioinformatics application: an accelerated approach to clustering Microarray data. In: Adaptive Hardware and Systems, pp. 248–255 (2011)
20. Zhang, C., Li, P., Sun, G., Guan, Y., Xiao, B., Cong, J.: Optimizing FPGA-based accelerator design for deep convolutional neural networks. In: Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays - FPGA 2015, pp. 161–170 (2015)
21. Han, S., Kang, J., Mao, H., Hu, Y., Li, X., Li, Y., Xie, D., Luo, H., Yao, S., Wang, Y.: ESE: efficient speech recognition engine with sparse LSTM on FPGA (2016)
22. Ouyang, J., Lin, S., Qi, W., Wang, Y., Yu, B., Jiang, S.: SDA: Software-defined accelerator for large-scale DNN systems. In: Hot Chips 26 Symposium, pp. 1–23 (2014)

Smart Resource Allocation Using Reinforcement Learning in Content-Centric Cyber-Physical Systems

Keke Gai[1], Meikang Qiu[2,3(✉)], Meiqin Liu[4], and Hui Zhao[5]

[1] School of Computer Science and Technology,
Beijing Institute of Technology, Beijing 100081, China
kekegai@yahoo.com
[2] Department of Computer Science, Pace University,
New York City, NY 10038, USA
mqiu@pace.edu
[3] Shenzhen University, Guangdong 518060, China
[4] College of Electrical Engineering,
Zhejiang University, Hangzhou 310027, China
liumeiqin@zju.edu.cn
[5] Institute of Intelligent Network System, Henan University,
Kaifeng 475000, China
zhh@henu.edu.cn

Abstract. The exponential growing rate of the networking technologies has led to a dramatical large scope of the connected computing environment. As a novel computing deployment, *Cyber-Physical Systems* (CPSs) are considered an alternative for achieving high performance by the enhanced capabilities in system controls, resource allocations, data exchanges, and flexible adoptions. However, current CPS is encountering the bottleneck concerning the resource allocation due to the mismatching networking service quality and complicated service offering environments. The concept of *Quality of Experience* (QoE) in networks further increases the demand for intensifying intelligent resource allocations to satisfy distinct user groups in a dynamic manner. This paper concentrates on the issue of resource allocations in CPS and also considers the satisfactory of QoE in content-centric computing systems. A novel approach is proposed by this work, which utilizes the mechanism of reinforcement learning to obtain high accurate QoE in resource allocations. The assessments of the proposed approach were processed by both theoretical proofs and experimental evaluations.

This project is supported by the National Natural Science Foundation of China (No. 61728303), and the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China (No. ICT170331). This project also is partially supported by National Natural Science Foundation of China (No. 61703141), the Basic and Frontier Technology Research of Henan Province Science and Technology Department (No. 162300410198) and the Henan Postdoctoral Science Found (No. 153040).

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 39–52, 2018.
https://doi.org/10.1007/978-3-319-73830-7_5

Keywords: Reinforcement learning · Resource allocation
Content-centric · Cyber-Physical System · Smart computing

1 Introduction

The dramatically increasing amount of the network-based techniques adoptions is leading to a large extent of the connected environment. Establishing a scalable and networkable system is becoming one of the mainstreams in modern industries, due to its multiple benefits, such as exchanging data, sharing infrastructure, integrating systems, or distributing workloads [1–3]. As an emerging technique in the networking era, a *Cyber-Physical System* is an efficient approach for tying up the Internet, humans, and functional objects. A functional object can be either a hardware or a software, which is expected to provision certain offerings or achieve some purposes. The advantage of interconnecting numerous devices is observable. However, developing a competent resource allocation approach is a challenging issue.

Despite many prior studies addressing resource allocation issues [4], the challenges still exist in contemporary CPSs. The vital problem in this issue is creating an adaptable mechanism for resource allocations. The mechanism needs to consider at least two aspects: the first aspect is to guarantee a high performance; the other aspect is to ensure that the strategy of the resource allocation can be created in an acceptable time period. The performance can be any examinable objective in the scenario of CPS, such as the energy cost, response time, transmission rate, or security level [5–8]. Nevertheless, these two aspects generally are incompatible with one another [9]. An algorithm demanding a short execution time can only create sub-optimal solutions [10]; an algorithm that creates optimal solutions generally needs a longer execution time for creating resource allocation strategies [11,12]. Thus, finding out an approach that distinguishes from prior research directions and effectively avoid the contradiction is a demanding job.

This paper focuses on the resource allocation issue in CPS and proposes a novel approach that uses *Reinforcement Learning* (RL) mechanism to construct the strategy of the resource allocation. Implementing a RL mechanism intends to prevent the contradiction and make the resource allocation operations smart. Term *Smart* in our model refers to intelligent operations that can output optimal solutions with using a lower-level or affordable computing resource. The proposed approach entitled *Smart Reinforcement Learning on Content-Centric Networking* (SRL-CCN) model, which applies addressable and routable contents architecture for communications. Our approach also considers *Quality of Experience* (QoE) to construct the value function used in RL. Two primary parameters are involved in the value functions, which are energy costs and response time.

The significance of this work is perceivable due to the high demand. From the technical perspective, the main contributions of this paper include:

1. This paper proposes a novel approach that uses RL techniques for resource allocations in CPS. The proposed approach can solve a core issue restricting

previous solutions, which is the contradiction between the performance and the strategy generation time. The main algorithm used our model is a dynamic programming.
2. This paper emphasizes the implementation of the content-centric network to enhance the fulfillment of the resource allocation. The value function used in RL considers both energy cost and response time, which is expandable and extendable to other applications.

The rest of this paper is organized by the following order. First, Sect. 2 presents the proposed model as well as key concepts used in our model. Next, Sect. 3 demonstrates the core algorithms that support our model. Furthermore, Sect. 4 provides a motivational example to show the major steps and mechanisms of our model. Moreover, Sect. 5 shows the configuration of the experiment and illustrates a series of experiment results. Finally, Sect. 6 gives conclusions of this work.

2 Concepts and the Proposed Model

2.1 Preliminary

We provide a brief synthesis of preliminary studies related to our work in this section. First, our model is on the basis of the fact that information transfers are controllable in CPS. This goal can be achieved by various techniques, such as utilizing *Data Chunks* (DCs) in resource allocations [13]. A DC is a term for describing a piece of information carried by a *Stream Control Transmission Protocol* (SCTP), which embraces a header showing data parameters for control purposes. For instance, *Hadoop Distributed File System* (HDFS) divides a sizable dataset into a number of DCs in order to allocate them to various processors for distinct tasks, such as *map* and *reduce* [14]. The information carried by DCs is the requirement for task allocation strategy-making.

Moreover, our previous studies have investigated the optimization of the task allocation in the connected environment. One exploration [15] was optimizing data allocations in cloud-based heterogeneous memory. This method considered the volume of *Read* and *Write* the crucial weight for the strategy generation. A heuristic checking manipulation was designed to increase the generation rate of the optimal solution. Another attempt [16] developed a heuristic algorithm to transfer a sub-optimal solution to an optimal/near-optimal solution in the context of heterogeneous cloud computing. The optimization was based on a series of operations that were called *Smart Switch*. In addition, the research [17] proposed a dynamic programming algorithm to produce optimal solutions to task allocations in the edge/fog context.

Despite many studies addressing this topic, limitations still existed in the prior proposed methods when considering stochastic input tasks. Most optimizations used relatively fixed input task tables such as the number of input tasks or the mapping table of costs. This setting normally restricted the adoptability due to the restrict requirement of the implementation scenario. Even though some

approaches [15] could process dynamic incoming tasks, a pre-stored cost mapping table was a requisite for creating allocation strategies, which restrained the methods from being ubiquitous solutions. It implied that most prior task allocation optimizations could be adopted in specific implementation contexts, albeit they might be not competent to optimizing a system with frequent updates and changeovers, involving a CPS. Thus, finding out an approach that could deal with dynamic incoming tasks without restrictions from pre-stored mapping tables was an urgent problem.

Furthermore, QoE is a general concept for increasing system management and governance by measuring the level the customers' satisfactory levels. The measurement scope covers objectives of services for quality assessments. In our model, QoE is used to describe the satisfaction level of the service requester in CPS. There are a variety of QoE factors applied in the CPS context, such as bandwidth, data package size, location, frequency of use, latency, energy costs, interruptions, or task type. These factors can be used to formulate reward parameters and value functions. The next section presents a detailed presentation about the proposed system.

2.2 System Design

In our proposed system, the core part is implementing an application-enabled intelligent agent that is designed to dynamically make decisions on task allocations based on the surrounding actions, states, and rewards. This approach derives from the theory of reinforcement learning [18–20]. Figure 1 presents the high architecture of the proposed model. In the context of CPS, data capture relies on various data sources such that multiple data streams are collected. The captured data are sent to the data manager layer, as shown in the figure, to execute a few operations for the purpose of the management, such as data preprocessing, task allocations to remote servers, or storage. The implementation of our model takes place at the component of the data task allocation.

Moreover, we formulate the content-centric resource allocations into a directed tree $G = \langle V, E \rangle$, in which nodes V refer to the set of the computation nodes and edges E refer to the set of costs corresponding to the set V. Our goal is to find out the lowest cost value by determining the computation nodes, which means that the objective is to obtain the shortest edge length by determining the participant connected nodes. A cost can be any computing resource in our model, such as energy consumption, execution time length, and hardware utilization. Each node in the tree represents a state. We assume that there are enough computation nodes for processing the computing jobs at one service request. Let the set $\{C_i\}$ denote the available computation nodes and let the set $\{T_j\}$ denote the incoming data packages or tasks. Thus, the number of the data package/task in $\{T_j\}$ is no more than the number of the computation units in $\{C_i\}$ at one resource allocation request. The incoming tasks can be divided into different requests or merged into one requests when there are exceeding incoming tasks. We also define that one node cannot be used for processing a task once it is selected by other tasks. Thus, the number of nodes having the same h depth

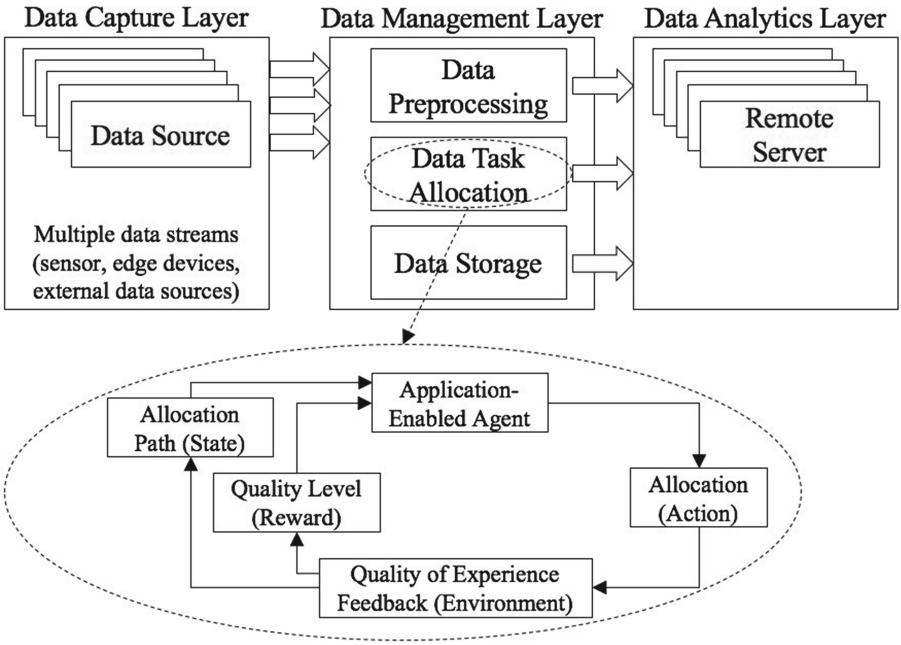


Fig. 1. Architecture of the proposed SRL-CCN model located in a CPS.

is denoted by $\mathbf{N}_h$, which can be obtained by Eq. (1). $\mathbf{N}^c$ refers to the number of the available computation nodes.

$$\mathbf{N}_h = \prod_{u=1}^{h} (\mathbf{N}^c - u + 1). \tag{1}$$

In addition, we define the root node as the determination state, which contains the key label for determining which computation node shall be taken by the first incoming task. We use the QoE mechanism to formulate the return values for each action. As shown in Fig. 1, the decision is made by an application-enabled intelligent agent. Each resource allocation is considered an action such that allocating task T_j to the computation node C_i is denoted by $a_{(T_j,C_i)}$. Corresponding with the action, the service quality levels represent various rewards and the allocation paths formulate states. The depth of the tree refers to the order of the incoming tasks. For example, at the h depth layer, all nodes at this layer are for the same task T_h.

Furthermore, we define the true value of the action a, denoted by $q(a)$, as the performance estimate after processing action a. We use a $Q_n(a)$ to denote the performance estimate at state n of $q(a)$, which is expressed in Eq. (2).

$$Q_n(a) = Q_{n-1}(a') + R_n. \tag{2}$$

Since the children nodes are resource allocation alternatives for their parent nodes, Eq. (2) implies an iteration that shows state updates due to the operation of the succeeding task over the state of the preceding tasks.

Next, in a mapping table, each row represents an incoming task and each column represents a computation node. Each element in the table refers to the cost of the resource allocation for the task. Based on this configuration, we select the lowest value at each row and subtract it from all elements. The acquisition of the mapping table is based on the implementation of QoE. We consider the differences obtained from subtraction operations the level of QoE, since the users expect the lowest cost. The level of QoE is higher when the difference value is closer to 0. The results are considered the reward values and its mathematical expression is given in Eq. (3). $R_{(T_j)}|A_{C_i} = a$ denotes the reward value when the input task T_j is allocated to the computation node C_i. We use $\mathcal{R}_{(T_j)}$ to denote the possible reward set for task T_j. In this set, all values shall be less or equal to 0. $\mathbf{C}_{(T_j)}$ denotes a set containing all costs corresponding with task T_j; $C_{(T_j,C_i)}$ refers to any element in the set $\mathbf{C}_{(T_j)}$.

$$\mathcal{R}_{(T_j)} = \{R_{(T_j)}|A_{C_i} = a\} = \{\min[\mathbf{C}_{(T_j)}] - C_{(T_j,C_i)}|A_{C_i} = a\}. \tag{3}$$

In order to obtain the optimal output, the final reward is a cumulative value. Equation (4) presents the mathematical expression of the cumulative reward value. The basic meaning of this equation is that the final reward value can be obtained by summing up all reward values from those nodes attached to the path when the action a takes place. In other words, when computing the final reward, the cumulative sum covers from the node with the final action to the root node in the tree. Every action given at one node results in a reward value so that the cumulative computation is based on a single path. Since one computation node can be only allocated once at one service request, the children nodes are not stochastic. Meanwhile, the costs of the tasks for various computation nodes are pre-mapped such that there is no need to have parameters during the cumulative computations. Equation (4) presents the expression of the cumulative reward. $R_k(a)$ refers to the reward when processing an action a over the kth incoming task. The cumulative computation only sums up the reward values from one path. $R_k|A_{C_i} = a$ means the action of the resource allocation is $A_{C_i} = a$ for this reward.

$$R_k(a) \doteq [R_k|A_{C_i} = a] = \sum_{u=1}^{k} \Big[\big[\min[\mathbf{C}_{(T_k)}] - \mathbf{C}_{(T_k,C_i)}\big]|A_{C_i} = a\Big]. \tag{4}$$

Furthermore, we define π as the policy that is an expected return label for the resource allocation strategy. Thus, $v_\pi(s)$ represents the expected value at the state s when implementing the policy π. According to these definitions, Eq. (5) shows a mathematical expression of $v_\pi(s)$. In the equation, G_{T_j} refers to the expected return cumulative reward when the jth task is allocated; $G_{T_j}|S_{T_j} = s$ means the state when task T_j is allocated to obtain the cumulative reward G_{T_j}. Similarly, Eq. (6) illustrates the value of resource allocation action a in state

s when implementing the policy π. Thus, $q_\pi(s,a)$ is the expected value of the allocation action considering both the state s and policy π.

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_{T_j}|S_{T_j} = s] = \mathbb{E}_\pi\big[\sum_{k=1}^{h} R_k|S_{T_j} = s\big]. \tag{5}$$

$$q_\pi(s,a) \doteq \mathbb{E}_\pi[G_{T_j}|S_{T_j} = s, A_{C_i} = a] = \mathbb{E}_\pi[\sum_{k=1}^{h} R_k|S_{T_j} = s, A_{C_i} = a]. \tag{6}$$

Based on the explanation above, we provide optimal value functions below in Eq. (7). In the equation, $\pi_*(s)$ and $q_*(s,a)$ represent the value of the state s when an optimal resource allocation strategy is implemented and the value of allocation action a in the state s when implementing an optimal solution strategy, respectively. A function obtaining the maximum value is applied in determining the eventual value. A back-forward path showing the policy π will be outputted once the maximum value is found by the function.

$$\begin{aligned} \pi_*(s) &\doteq \max_a q_\pi(s,a) \\ q_*(s,a) &\doteq \max_\pi q_\pi(s,a) \\ &= \max_\pi \mathbb{E}_\pi[\sum_{k=1}^{h} R_k|S_{T_h} = s, A_{C_i} = a]. \end{aligned} \tag{7}$$

In summary, this section provides a brief description about the design of the proposed reinforcement learning mechanism and its deployment in CPS. The next section will introduce the core algorithm used in this approach.

3 Algorithms

The core algorithm supporting our SRL-CCN model is called *Reinforcement Learning Resource Allocation* (RLRA) algorithm. The mechanism of this algorithm uses dynamic programming, which is designed to process unfixed input tasks. The number of the tasks that can be dealt with by implementing RLRA algorithm is equal to or less than the number of the available computation node.

Moreover, inputs of this algorithm include a set of incoming tasks, $\mathbf{T} = \{T_i\}$, and a pre-stored mapping table, $\mathcal{T}^m$. This mapping table consists costs of various types of tasks at each available computation nodes. The output is a policy $\pi_*(s,a)$ that shows the resource allocation path for producing an optimal strategy. Pseudo codes of RLRA algorithm is shown in Algorithm 3.1.

Main phases of Algorithm 3.1 are:

1. Read the information from data chunks and retrieve the cost information from the pre-stored cost mapping table. Initialize a temporary table for the purpose of the reward computation. Initialize a tree for the purpose of policy generations.

Algorithm 3.1 Reinforcement Learning Resource Allocation (RLRA) Algorithm

Require: Input tasks **T**, a mapping table $\mathcal{T}^m$
Ensure: Optimal policy $\pi_*(s, a)$

Initialize a temporary table $\emptyset \leftarrow Temp$ and a tree Tr
/* The number of the column in $Temp$ = the number of computation nodes */
for $\forall$ tasks T_i in **T do**
  Read the corresponding row from $\mathcal{T}^m$
  Operate $\min[\mathbf{C}_{(T_j)}] - C_{(T_j, C_i)}$ and update $Temp$
  for $\forall$ states s **do**
    $q_\pi(s, a) \leftarrow \mathbb{E}_\pi[\sum_{k=1}^{h} R_k | S_{T_j} = s, A_{C_i} = a]$
    /* Update Tr by cumulative computations */
  end for
end for
for $\forall$ q_π **do**
  Find $\max_\pi q_\pi(s, a)$
  $\pi_*(s, a) \leftarrow \max_\pi q_\pi(s, a)$
end for
return $\pi_*(s, a)$

2. Apply Eqs. (3) and (4) to initialize the reward table. Add tasks' rewards to the tree according to the action of the resource allocation. For each input task, a reward is a cumulative value that involves all prior nodes at the same path.
3. The manipulation of computing rewards ends once there is no input task. Pick up the maximum value from the leaves and output its policy. The policy is an optimal resource allocation strategy.

4 Motivational Example

A simple example is presented in this section in order to provide a basic picture about the implementation of our approach. We assume that there are four available computation nodes in CPS. This configuration is made for the purpose of the manipulation illustration, even though there maybe a large number of computation nodes in a real-world environment. Another assumption is that the tasks are inputted in a dynamic manner and the number of the input task is no more than the number of computation nodes. In this example, we configure four data package as the input tasks, D_1, D_2, D_3, and D_4. The input order also follows the index order, from 1 to 4. Table 1 shows a cost mapping table that is obtained from reading the information carried by data chunks and retrieving the cost information from the pre-stored cost table. Since tasks are inputted one by one, the outputs will be varied along with the various amounts of input tasks.

We present Fig. 2 to show an example of locating resource allocation actions at the same path in a tree. Figure 2(a) is a state showing the first input task. The root node is a state without any input. Nodes 2, 3, 4, and 5 are four options

Table 1. Cost mapping based on the information carried by data chunks

Data Package	C_1	C_2	C_3	C_4
D_1	5	6	5	9
D_2	5	1	5	8
D_3	8	3	5	2
D_4	3	1	10	2

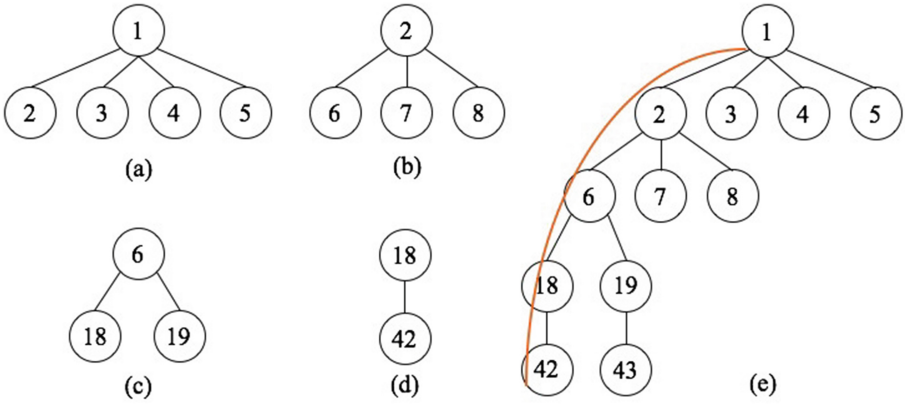


Fig. 2. An example of the resource allocation actions over one path in a tree.

of the computation node. For example, node 2 represents that C_1 is allocated. Figure 2(b) shows partial tree when the second task is inputted. Rather than four options, node 2 has three children nodes since C_1 is already taken. Nodes 6, 7, and 8 represent C_2, C_3, and C_4, respectively. Similarly, Fig. 2(c) and (d) present the succeeding nodes. The number of the children nodes decreases along with the reduction of the available computation nodes. Figure 2(e) illustrates a path (1)–(2)–(6)–(18)–(42), which represents an allocation policy as $D_1 \rightarrow C_1$, $D_2 \rightarrow C_2$, $D_3 \rightarrow C_3$, and $D_4 \rightarrow C_4$.

Moreover, we implement Algorithm 3.1 to compute reward values and obtain policies according to input tasks. States are continuously changing due to the dynamic input tasks, such that output policies will be varied. Figure 4 depicts the changes of the states when different resource allocation actions are processed. The computation nodes C_i in these tables show the first computation node that will be allocated. The policy is created by the path led by these computation nodes. It shows that a path led by C_3 is an optimal allocation policy. We provide a diagram for showing a back-forward path in Fig. 4. This output policy is optimal because the path's final reward value is -2, which is higher than any other paths' final reward values. This result's correctness can be proved by implementing a brute-force searching.

q_π

C_1	C_2	C_3	C_4
0	-1	0	-4

q_π

C_1	C_2	C_3	C_4
0	-5	0	-4

$$q_\pi(S_2|C_1) = \max[0, -4, -7] \qquad q_\pi(S_2|C_3) = \max[-4, 0, -7]$$
$$q_\pi(S_2|C_2) = \max[-5, -5, -8] \qquad q_\pi(S_2|C_4) = \max[-8, -4, -8]$$

q_π

C_1	C_2	C_3	C_4
0	-5	0	-7

$$q_\pi(S_3|C_1) = \max[-3, 0, -5, -4, -8, -10]$$
$$q_\pi(S_3|C_2) = \max[-8, -5, -11, -5, -14, -11]$$
$$q_\pi(S_3|C_3) = \max[-5, -4, -6, 0, -13, -8]$$
$$q_\pi(S_3|C_4) = \max[-9, -11, -10, -7, -14, -9]$$

q_π

C_1	C_2	C_3	C_4
-4	-7	-2	-9

$$q_\pi(S_4|C_1) = \max[-4, -9, -6, -4, -17, -10]$$
$$q_\pi(S_4|C_2) = \max[-9, -14, -12, -7, -25, -13]$$
$$q_\pi(S_4|C_3) = \max[-6, -4, -7, -2, -13, -10]$$
$$q_\pi(S_4|C_4) = \max[-18, -11, -19, -9, -14, -11]$$

Fig. 3. Changes of the states along with the incoming actions led by resource allocations.

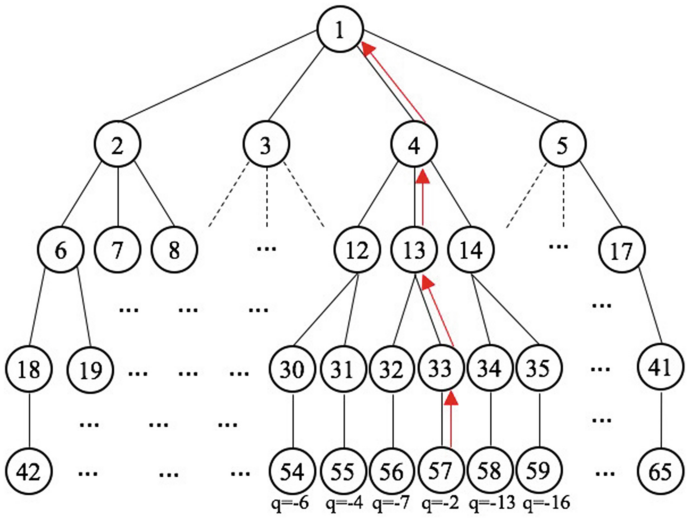


Fig. 4. A sequence of task allocation moves following a back-forward path in a tree.

5 Experiment and the Results

5.1 Experiment Configuration

We presented our experimental revaluations in this section. The evaluation mainly focused on the training time for using reinforcement learning to obtain optimal resource allocations. We configured that the number of the input task was in an incremental manner. The purpose of this configuration was to examine the impact caused by the input setting. The findings derived from this examination would guide the input configuration by ensuring the number of input tasks at one service request within an adoptable scope. Thus, inputs used in our evaluation considered two major aspects, which were the number of tasks and the number of computation nodes.

Next, the hardware configuration of the experimental environment involved a processor with a 2.4 GHz Intel Core i5, a memory with 8 GB 1600 MHz DDR3, and a graphics card with Intel Iris 1536 MB. Our simulator program was developed in Java. The program could generate random cost values for each input task according to the amount of the available computation node. Our simulator also ensured each task had at least computation node to be allocated.

Moreover, concerning the computation nodes, we grouped computation nodes depending on their computing capabilities. Computation nodes in the same group had the same cost parameters for incoming tasks. The purpose of this setting was to lower down the computing complexity. We displayed two settings aligning with the number of the computation node group in this section, which were Setting 1 with 4 groups and Setting 2 with 8 groups. The simulator had a stochastic distribution of computation nodes for groups. The total available computation nodes was always more than the number of input tasks. In order to make the results comparable, we set the scope of the input amount from 0 to 50. We ran at least 10-time rounds for an input amount under each setting.

Finally, our evaluations involved a variety of parameters for comparing and measuring data collected from various settings. The next section presented partial experiment results as well as details of parameters.

5.2 Experiment Results

We presented a few experiment results and our analysis in this section. A variety of measurement methods were applied in the study, including average and increase rate. Figure 5 depicted the training time with an incremental amount of puts in average when implementing Setting 1. The training time was counted in seconds. We observed that the time grew when the number of input tasks was increasing. According to our statistics, the average of the training time was 0.0703 s; the median value was 0.0414 s; the maximum time length was 0.3090 s that was 161.85 times longer than minimum time.

Moreover, we measured the incremental rate for each input task by implementing the following equation: $R_n^{\text{incre}} = (T_n - T_{n-1})/T_{n-1}$, where R_n^{incre} was the incremental rate from the $(n-1)th$ task to the nth task, T_n was the training

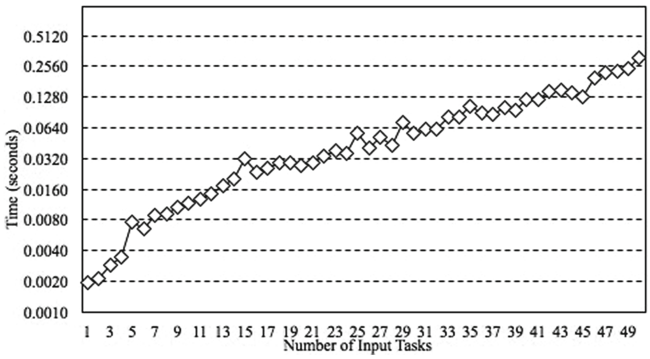


Fig. 5. Average training time with an incremental number of the input task under Setting 1.

time for n tasks and T_{n-1} was the time for $(n-1)$ tasks. The average R^{incre} was 12.97% and the scope of R^{incre} was $[-29.04\%, 121.69\%]$ under this setting.

Furthermore, Fig. 6 showed the experiment results collect from Setting 2. We noticed that the training time was dramatically increased due to the types of the computation nodes were doubled. The time obviously grew when the number of inputs was greater than 8; thus the investigation mainly considered the data collected from tasks after the 7th task the valid data in statistics. On the whole, the average of the training time was 7.17 s within a scope [0.03, 30.24]. In this scope, the median value was 2.39 s. Additionally, the incremental rate R^{incre} was 20.77%, on the average, and its scope was $[-36.18\%, 85.71\%]$.

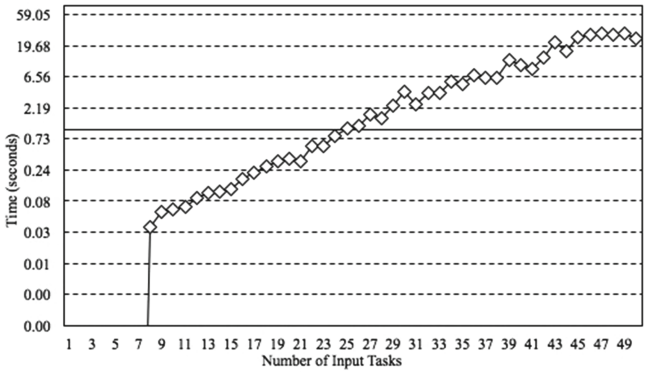


Fig. 6. Average training time with an incremental number of the input task under Setting 2.

In summary, main findings of our evaluations were summarized as follows: (1) The training time increased when the number of input tasks was rising; therefore, configuring an adoptable scope for the number of input tasks at one

service request was an important aspect in achieving real-time services. (2) The amount of the available computation nodes had a great impact on the training time. Grouping up those computation nodes with similar/close capabilities could shorten the training time period. Our future work would further explore the method of creating pre-stored table by using reinforcement learning and QoE techniques.

6 Conclusions

This paper proposed a novel approach for resource allocation in CPS, which was based on a design of the reinforcement learning mechanism. Our approach applied the method of QoE to examine the service level, which enabled a dynamic resource allocation and avoided a fixed input task table. The value function used in the proposed reinforcement learning considered the level of the QoE the reward parameter and the output of our approach was an optimal solution.

References

1. Xin, S., Guo, Q., Sun, H., Zhang, B., Wang, J., Chen, C.: Cyber-physical modeling and cyber-contingency assessment of hierarchical control systems. IEEE Trans. Smart Grid **6**(5), 2375–2385 (2015)
2. Liu, R., Vellaithurai, C., Biswas, S., Gamage, T., Srivastava, A.: Analyzing the cyber-physical impact of cyber events on the power grid. IEEE Trans. Smart Grid **6**(5), 2444–2453 (2015)
3. Gai, K., Qiu, M., Sun, X.: A survey on FinTech. J. Netw. Comput. Appl. **PP**, 1 (2017)
4. Qiu, M., Zhong, M., Li, J., Gai, K., Zong, Z.: Phase-change memory optimization for green cloud with genetic algorithm. IEEE Trans. Comput. **64**(12), 3528–3540 (2015)
5. Liu, X., Aeron, S., Aggarwal, V., Wang, X., Wu, M.: Adaptive sampling of RF fingerprints for fine-grained indoor localization. IEEE Trans. Mob. Comput. **15**(10), 2411–2423 (2016)
6. Liu, X., Zhu, Y., Kong, L., Liu, C., Gu, Y., Vasilakos, A., Wu, M.: CDC: compressive data collection for wireless sensor networks. IEEE Trans. Parallel Distrib. Syst. **26**(8), 2188–2197 (2015)
7. Gai, K., Qiu, M., Liu, M., Xiong, Z.: In-memory big data analytics under space constraints using dynamic programming. Future Gener. Comput. Syst. **PP**, 1 (2018)
8. Gai, K., Qiu, M.: Blend arithmetic operations on tensor-based fully homomorphic encryption over real numbers. IEEE Trans. Ind. Inform. **PP**(99), 1 (2018)
9. Gai, K., Qiu, M., Zhao, H., Sun, X.: Resource management in sustainable cyber-physical systems using heterogeneous cloud computing. IEEE Trans. Sustain. Comput. **PP**(99), 1–13 (2017)
10. Gai, K., Qiu, M., Zhao, H.: Privacy-preserving data encryption strategy for big data in mobile cloud computing. IEEE Trans. Big Data **PP**(99), 1–11 (2017)
11. Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G.: A survey on algorithmic approaches for solving tourist trip design problems. J. Heuristics **20**(3), 291–328 (2014)

12. Venugopalan, S., Sinnen, O.: ILP formulations for optimal task scheduling with communication delays on parallel systems. IEEE Trans. Parallel Distrib. Syst. **26**(1), 142–151 (2015)
13. Wang, W., Zhu, K., Ying, L., Tan, J., Zhang, L.: A throughput optimal algorithm for map task scheduling in MapReduce with data locality. ACM SIGMETRICS Perform. Eval. Rev. **40**(4), 33–42 (2013)
14. Ghemawat, S., Gobioff, H., Leung, S.: The Google file system. ACM SIGOPS Oper. Syst. Rev. **37**(5), 29–43 (2003)
15. Gai, K., Qiu, M., Zhao, H.: Cost-aware multimedia data allocation for heterogeneous memory using genetic algorithm in cloud computing. IEEE Trans. Cloud Comput. **PP**(99), 1–11 (2016)
16. Gai, K., Qiu, M., Zhao, H.: Energy-aware task assignment for mobile cyber-enabled applications in heterogeneous cloud computing. J. Parallel Distrib. Comput. **111**, 126–135 (2018)
17. Gai, K., Qiu, M., Zhao, H., Tao, L., Zong, Z.: Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. J. Netw. Comput. Appl. **59**, 46–54 (2015)
18. Liu, Y., Tang, L., Tong, S., Chen, C., Li, D.: Reinforcement learning design-based adaptive tracking control with less learning parameters for nonlinear discrete-time MIMO systems. IEEE Trans. Neural Netw. Learn. Syst. **26**(1), 165–176 (2015)
19. Wei, C., Zhang, Z., Qiao, W., Qu, L.: Reinforcement-learning-based intelligent maximum power point tracking control for wind energy conversion systems. IEEE Trans. Ind. Electron. **62**(10), 6360–6370 (2015)
20. Yin, J., Chen, D., Li, L.: Intelligent train operation algorithms for subway by expert system and reinforcement learning. IEEE Trans. Intell. Transp. Syst. **15**(6), 2561–2571 (2014)

Effective Malware Detection Based on Behaviour and Data Features

Zhiwu Xu(✉), Cheng Wen, Shengchao Qin, and Zhong Ming

College of Computer Science and Software Engineering,
Shenzhen University, Shenzhen, China
{xuzhiwu,sqin,mingz}@szu.edu.cn, 2150230509@email.szu.edu.cn

Abstract. Malware is one of the most serious security threats on the Internet today. Traditional detection methods become ineffective as malware continues to evolve. Recently, various machine learning approaches have been proposed for detecting malware. However, either they focused on behaviour information, leaving the data information out of consideration, or they did not consider too much about the new malware with different behaviours or new malware versions obtained by obfuscation techniques. In this paper, we propose an effective approach for malware detection using machine learning. Different from most existing work, we take into account not only the behaviour information but also the data information, namely, the opcodes, data types and system libraries used in executables. We employ various machine learning methods in our implementation. Several experiments are conducted to evaluate our approach. The results show that (1) the classifier trained by Random Forest performs best with the accuracy 0.9788 and the AUC 0.9959; (2) all the features (including data types) are effective for malware detection; (3) our classifier is capable of detecting some fresh malware; (4) our classifier has a resistance to some obfuscation techniques.

1 Introduction

Malware, or malicious software is a generic term that encompasses viruses, trojans, spywares and other intrusive codes. They are spreading all over the world through the Internet and are increasing day by day, thus becoming a serious threat. According to the recent report from McAfee [1], one of the world's leading independent cybersecurity companies, there are more than 650 million malware samples detected in Q1, 2017, in which more than 30 million ones are new. So the detection of malware is of major concern to both the anti-malware industry and researchers.

To protect legitimate users from these threats, anti-malware software products from different companies provide the major defence against malware, such as Comodo, McAfee, Kaspersky, Kingsoft, and Symantec, wherein the signature-based method is employed. However, this method can be easily evaded by malware writers through the evasion techniques such as packing, variable-renaming, and polymorphism [2]. To overcome the limitation of the signature-based method, heuristic-based approaches are proposed, aiming to identify the

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 53–66, 2018.
https://doi.org/10.1007/978-3-319-73830-7_6

malicious behaviour patterns, through either static analysis or dynamic analysis. But the increasing number of malware samples makes this method no longer effective. Recently, various machine learning approaches like Support Vector Machine, Decision Tree and Naive Bayes have been proposed for detecting malware [3]. These techniques rely on data sets that include several characteristic features for both malware and benign software to build classification models to detect (unknown) malware. Although these approaches can get a high accuracy (for the stationary data sets), it is still not enough for malware detection. On one hand, most of them focus on the behaviour features such as binary codes [4–6], opcodes [6–8] and API calls [9–11], leaving the data information out of consideration. While a few of them do consider the data information, but only simple features like strings [12,13] and file relations [14,15]. On the other hand, as malware continues to evolve, some new and unseen malware have different behaviours and features. Even the obfuscation techniques can make malware difficult to detect. Hence, more datasets and experiments are still needed to keep the detection effective.

In this paper, we propose an effective approach to detecting malware based on machine learning. Different from most existing work, we take into account not only the behaviour information but also the data information. Generally, the behaviour information reflects what the software intends to behave, while the data information indicates which datas the software intends to perform on or how data are organised. Our approach tries to learn a classifier from existing executables with known categories first, and then uses this classifier to detect new, unseen executables. In detail, we take the opcodes, data types and system libraries that are used in executables, which are collected through static analysis, as representative features. As far as we know, our approach is the first one to consider data types as features for malware detection. Moreover, in our implementation, we employ various machine learning methods, such as K-Nearest Neighbor, Native Bayes, Decision Tree, Random Forest, and Support Vector Machine to train our classifier.

Several experiments are conducted to evaluate our approach. Firstly, we conducted 10-fold cross validation experiments to see how well various machine learning methods perform. We found that the classifier trained by Random Forest performs best here, with the accuracy 0.9788 and the AUC 0.9959. Secondly, we conducted experiments to illustrate that all the features are effective for malware detection. The results also show that in some case using type information is better than using the other two. Thirdly, to test our approach's ability to detect genuinely new malware or new malware versions, we ran a time split experiment: we used our classifier to detect the malware samples which are newer than the ones in our data set. Our classifier can detect 81% of the fresh samples, which indicates that our classifier is capable of detecting some fresh malware. The results also suggest that malware classifiers should be updated often with new data or new features in order to maintain the classification accuracy. Finally, one reason that makes malware detection difficult is that obfuscation techniques can be used to evade the detection. So for that, we performed experiments to test our approach's ability to detect new malware samples that are obtained

by obfuscating the existing ones through some obfuscation tools. All the obfuscated malware samples can be detected by our classifier, demonstrating that our classifier has a resistance to some obfuscation techniques.

The remainder of this paper is organised as follows. Section 2 presents the related work. Section 3 describes our approach, and followed by the experimental results in Sect. 4. Finally, Sect. 5 concludes.

2 Related Work

There have been lots of work on malware detection. Here we focus on some recent related work based on machine learning. Interested readers can refer to the surveys [3,16] for more details.

Over the past decade, intelligent malware detection systems have been developed by applying machine learning techniques. Some approaches employ the *N-grams* method to train a classifier based on the binary code content [4–6]. Rather surprisingly, a 1-g model can distinguish malware from benign software quite well for some data set. But these approaches can be evaded easily by control-flow obfuscation. Some approaches [6–8] take the opcodes, which reflect the behaviours of the software of interest quite well, as features to classify malware and benign software. Some other approaches consider API calls, intents, permissions and commands as features [9–11], based on different classifiers. All these works focus more on the behaviour information, while our current work considers not only behaviour information but also data information.

There have been some research work that takes data information into account. Ye et al. [12] propose a malware detection approach based on interpretable strings using SVM ensemble with bagging. Islam et al. [13] consider malware classification based on integrated static and dynamic features, which includes printable strings. Both Karampatziakis et al. [14] and Tamersoy et al. [15] propose the malware detection approach based on the file-to-file relation graphs. More precisely, a file's legitimacy can be inferred by analyzing its relations (co-occurrences) with other labeled (either benign or malicious) peers. Clearly, both the string information and the file relation are quite simple and not all malware share the same information.

Some recent research work employ deep learning to help detect malware. Saxe and Berlin [17] propose a deep neural network malware classifier that achieves a usable detection rate at an extremely low false positive rate and scales to real world training example volumes on commodity hardware. Hardy et al. [18] develop a deep learning framework for intelligent malware detection based on API calls. Ye et al. [19] propose a heterogeneous deep learning framework composed of an AutoEncoder stacked up with multilayer restricted Boltzmann machines and a layer of associative memory to detect new unknown malware. In addition, concept drift is also borrowed into malware detection. Jordaney et al. [20] propose a fully tuneable classification system *Transcend* that can be tailored to be resilient against concept drift to varying degrees depending on user specifications. Our approach employ several supervised machine-learning methods.

3 Approach

In this section, we first give a definition of our malware detection problem, then present our approach.

The malware detection problem can be stated as follows: given a dataset $D = \{(e_1, c_1), \ldots, (e_m, c_m)\}$, where $e_i \in E$ is an executable file, $c_i \in C = \{benign, malicious\}$ is the corresponding actual category of e_i which may be unknown, and m is the number of executable files, the goal is to find a function $f : E \to C$ such that $\forall i \in \{1, \ldots, m\}$. $f(e_i) \approx c_i$.

Indeed, the goal is to find a classifier which classifies each executable file as precise as possible. Our approach tries to learn a classifier from existing executables with known categories first, and then uses this classifier to predict categories for new, unseen executables. Figure 1 shows the framework of our approach, which consists of two components, namely the feature extractor and the malware classifier. The feature extractor extracts the feature information (*i.e.*, opcodes, data types and system libraries) from the executables and represents them as vectors. While the malware classifier is first trained from an available dataset of executables with known categories by a supervised machine-learning method, and then can be used to detect new, unseen executables. In the following, we describe both components in more detail.

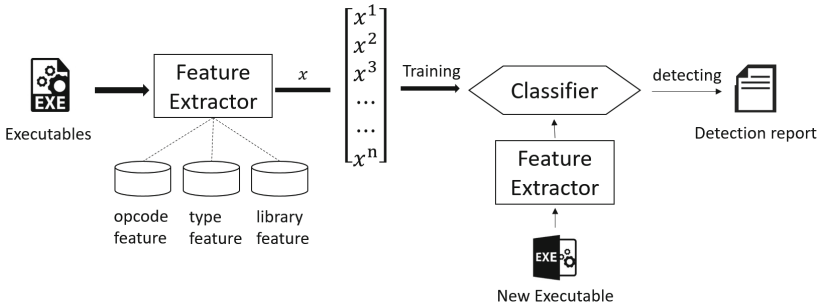


Fig. 1. The framework of our approach

3.1 Feature Extractor

This section presents the processing of the feature extractor, which consists of the following steps: (1) decompilation, (2) information extraction, and (3) feature selection and representation.

Decompilation. In this paper, we focus on the executables on Windows, namely, *exe* files and *dll* files. An instruction or a datum in an executable file can be represented as a series of binary codes, which are clearly not easy to read. So the first step is to transform the binary codes into a readable intermediate representation such as assembly codes by a decompilation tool.

Information Extraction. Next, the extractor parses the *asm* files decompiled from executables to extract the information, namely, opcodes, data types and system libraries. Generally, the opcodes used in an executable represent its intended behaviours, while the data types indicate the structures of the data it may perform on. In addition, the imported system libraries, which reflect the interaction between the executable and the system, are also considered. All such information describes the possible mission of an executable in some sense, and similar executables share the similar information.

The opcode information can be extracted either statically or dynamically. Here we just collect and count the opcodes from an asm file instruction by instruction, yielding an opcode list l_{opcode}, which records the opcodes appearing in the *asm* file and their corresponding times. For data type information, we first discover the possible variables, including local and global ones, and recover their types using BITY [21]. Then we do a statistical analysis on the recovered types, wherein we just consider their data sizes for the composed types for simplicity, yielding a type list l_{type}. The extraction of library information is similar to the one of opcodes, yielding the library list $l_{library}$.

Feature Selection and Representation. There may be too many different terms in the collected lists from all the executables, and not all of them are of interest. For example, *push* and *mov* are commonly used in both malware and benign softwares. For that, the extractor creates a dictionary to keep record of the interesting terms. Only the terms in this dictionary are reserved in the lists. Take the opcode lists for example. The extractor performs a statistical analysis on all the opcode lists from an available dataset, using TF-IDF to measure the statistical dependence. Next the extractor selects the top k weight opcodes to form an opcode dictionary, and then filters out the terms not in this dictionary for each l_{opcode}. The same applies to l_{type} and $l_{library}$.

After that, the extractor builds a *profile* for each executable file by concatenating these three lists collected from its corresponding *asm* file, namely, the executable is represented as a profile $[l_{opcode}, l_{type}, l_{library}]$. Assume that there is a fixed order for all the features. Then a profile can be simplified as a vector $(x^1, x^2, ..., x^n)$, where x^i is the value of the i-th feature and n is the total number of the feature. An example vector is shown in Fig. 2.

3.2 Malware Classifier

In this section, we present the malware classifier. As mentioned before, we first train our malware classifier from an available dataset of executables with known categories by a supervised machine-learning method, and then use it detect new, unseen executables.

Classifier Training. Now by our feature extractor, an executable e can be represented as a vector x. Let X denote the feature space for all possible vectors, D_0 represent the available dataset with known categories, and 0 and 1 represent *benign* and *malicious* respectively. Our training problem is to find a classifier $\mathsf{C} : X \to [0, 1]$ such that min $\Sigma_{(x,c)\in D_0} d(\mathsf{C}(x) - c)$, where d denotes the distance

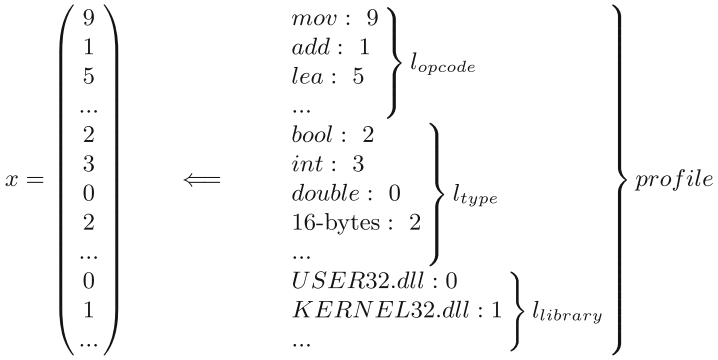


Fig. 2. Example for vector representation

function. Clearly, there are many classifier algorithms to solve this problem, such as K-Nearest Neighbor, Native Bayes, Decision Tree, Random Forest, Support Vector Machine, and SGD Classifier. In our implementation we have made all these methods available, so users can select any one they like. Moreover, we have also conducted some experiments with these algorithms and found out that the classifier trained by Random Forest performs best here (more details will be given in Sect. 4).

Malware Detection. Once it is trained, the classifier C can be used to detect malware: returns the category as close as the classifier returns. Formally, given an executable e and its feature vector x, the goal of the detection is to find $c \in C$ such that min $d(\mathsf{C}(x) - c)$.

4 Experiments

In this section, we conduct a series of experiments to evaluate our approach. Firstly, we conduct a set of cross-validation experiments to evaluate how well each classifier method performs based on the behaviour and data features. Meanwhile, we also measure the runtime for each classifier method. The runtime of the feature extractor is measured as well. Secondly, based on each kind of feature we also conduct experiments to see their effectiveness. Thirdly, to test our approach's ability to detect genuinely new malware or new malware versions, we also run a time split experiment. Finally, we conduct some experiments to test our approach's ability to detect new malware samples that are obtained by obfuscating the existing ones. In addition, the first two experiments also give a comparison of our method and most of existing methods, since most of existing methods adopt various classifier methods, which are considered here, with only behaviour features.

Our dataset consists of a malware dataset and a benign software dataset. The malware dataset consists of the samples of the BIG 2015 Challenge [22] and the samples before 2017 in theZoo aka Malware DB [23], with 11376 samples

Table 1. Performance measures in malware detection

Measure	Description
True Positive (TP)	Number of files correctly classified as malicious
True Negative (TN)	Number of files correctly classified as benign
False Positive (FP)	Number of files mistakenly classified as malicious
False Negative (FN)	Number of files mistakenly classified as benign
Precision (PPV)	TP/(TP + FP)
TP Rate (TPR)	TP/(TP + FN)
FP Rate (FPR)	FP/(TP + FN)
Accuracy (ACY)	(TP + TN)/(TP + TN + FP + FN)

in total; while the benign software dataset are collected from QIHU 360 software company, which is the biggest Internet security company in China, with 8003 samples in total.

To quantitatively validate the experimental results, we use the performance measures shown in Table 1. All our experiments are conducted in the environment: 64 Bit Windows 10 on an Intel (R) Core i5-4590 Processor (3.30 GHz) with 8 GB of RAM.

Cross Validation Experiments. To evaluate the performance of our approach, we conduct 10-*fold cross validation*experiments: in each experiment, we randomly split our dataset into 10 equal folds; then for each fold, we train the classifier model based on the data from the other folds (*i.e.*, the training set), and test the model on this fold (*i.e.*, the testing set). The learning methods we use in our experiments are listed as follows:

- K-Nearest Neighbour (KNN): experiments are performed under the range $k = 1$, $k = 3$, $k = 5$, and $k = 7$.
- Native Bayes (NB): several structural learning algorithms, that is, *Gaussian Naive Bayes* (GNB), *Multinomial Naive Bayes* (MNB), and *Bernoulli Naive Bayes* (BNB) are used in our experiments.
- Decision Trees (DT): we use decision tree with two criteria, namely, "gini" for the *Gini impurity* and "entropy" for the *information gain*. *Random Forest* (RF) with 10 trees are used as well.
- Support Vector Machines (SVM): *linear kernel*, *sigmoid kernel*, and *Radial Basis Function based kernel* (RBF) are used in our experiments, as well as the linear models with *stochastic gradient descent* (SGD).

Table 2 shows the detailed results of the experiments and Fig. 3 shows some selected *Receiver Operating Characteristic* (ROC) curve. All the KNN classifiers and DT classifiers perform quite well, with an accuracy greater than 97%, among which Random Forest with 10 entropy trees have produced the best result. Rather surprisingly, 1-KNN performs better than the other KNNs. While all the NB classifiers perform a bit worse, with the accuracy from 66.38% to 82.79%.

Table 2. Results of different methods

Classifier	PPV	TPR	FPR	ACY	Classifier	PPV	TPR	FPR	ACY
KNN K = 1	0.9609	0.9830	0.0281	0.9764	KNN K = 3	0.9572	0.9798	0.0307	0.9736
KNN K = 5	0.9556	0.9763	0.0319	0.9715	KNN K = 7	0.9556	0.9763	0.0319	0.9715
DT gini	0.9658	0.9797	0.0243	0.9773	DT entropy	0.9685	0.9787	0.0223	0.9781
RF gini	0.9678	0.9787	0.0228	0.9778	RF entropy	0.9692	0.9800	0.0218	0.9788
GNB	0.9381	0.1990	0.0092	0.6638	MNB	0.9494	0.6590	0.0247	0.8446
BNB	0.8726	0.6831	0.0701	0.8279	SVM linear	0.9581	0.9896	0.0304	0.9778
SVM rbf	0.9686	0.9119	0.0207	0.9514	SVM sigmoid	0.9671	0.7308	0.0232	0.8580
SGD	0.9582	0.9862	0.0302	0.9765					

For SVN classifiers, most of them get an accuracy greater than 95%, except for the one with sigmoid kernel whose accuracy is 85.80%. Concerning ROC curve, most classifiers can produce much better classification results than random classification results, except for Gaussian Naive Bayes with the Area Under the ROC (AUC) 0.5931, which is just a little bit bigger than 0.5. Random Forest with 10 entropy trees perform the best as well (with the AUC 0.9959). One of the main reasons for Random Forest to outperform others is that it is an ensemble learning method which may correct the Decision Trees' habit of overfitting to the training set. In the future, some other ensemble learning methods will be also considered.

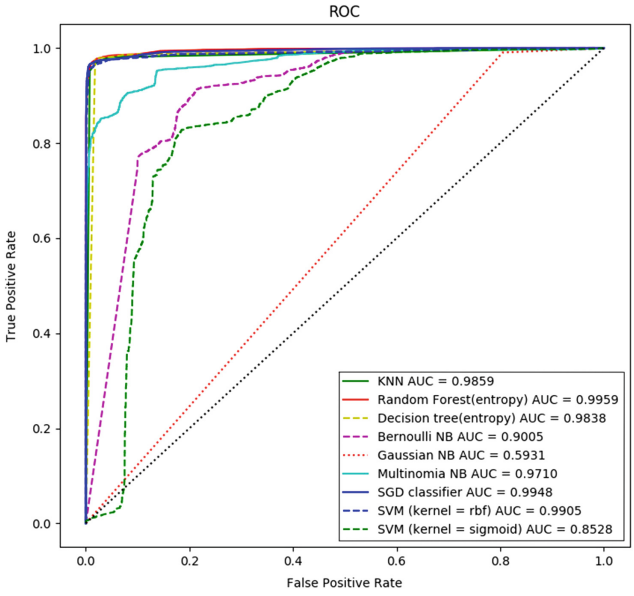


Fig. 3. ROC curve of different methods

Runtime Performance. Meanwhile, we count the training times (the runtime costed by building the classifier based on the training set) and the testing time (the time needed by evaluating the testing set) in seconds for each cross validation experiment. The results are shown in Table 3. In term of the training time, SVM has been the worst for our data set (with time ranging from 150.022 s to 1303.607 s), while Naive Bayes has been the best (with time ranging from 1.535 s to 3.093 s). Random Forest has also performed well on training, with time between 3.791 s and 4.115 s. Concerning the testing time, KNN has been the worst (due to its lazy learning), while Random Forest, Naive Bayes and SGD classifier have performed quite well, with time small than 1 s. Considering the accuracy, ROC and the runtime, we suggest users to use the Random Forest classifier.

Table 3. Runtime for various classifiers

Classifier	Training(s)	Testing(s)	Classifier	Training(s)	Testing(s)
KNN K = 1	16.477	178.789	KNN K = 3	16.369	199.474
KNN K = 5	16.517	207.052	KNN K = 7	16.238	210.557
DT gini	23.442	0.067	DT entropy	13.485	0.066
RF gini	4.115	0.086	RF entropy	3.791	0.077
GNB	3.093	0.480	MNB	1.535	0.035
BNB	1.826	0.828	SVM linear	150.022	14.494
SVM rbf	799.310	50.196	SVM sigmoid	1303.607	130.178
SGD	22.569	0.048			

We have also evaluated how the feature extractor performs. For that we first measure the total time needed by the decompilation tool IDA Pro to decompile the executables. Our data set contains 19379 executables, with the total size of 250 GB. It takes 15.6 h to decompile all the files, with an average 0.22 s/MB. IDA Pro is a heavy-weight tool, so using a lightweight one might have improved the time. Secondly, we measure the total time needed by extracting features from the decompiled asm files. The sizes of asm files range from 142 KB to 144.84 MB, with the total size of 182 GB. It takes 10.5 h to extract the features for all the asm files, with an average 0.20 s/MB. Both the decompilation time and the extracting time are considered to be acceptable.

Feature Experiments. We have also conducted experiments based on each kind of feature to see their effectiveness. For that, we conduct the same experiments as above for each kind of feature. Table 4 gives the experimental results and Fig. 4 shows the corresponding ROC curves, where only the results of one classifier are selected to present here for each kind of learning method.

From the results we can see that all the features are effective to help detect malware, and using all of the features together has produced the best results,

Table 4. Results of different features

Feature	Classifier	PPV	TPR	FPR	ACY	AUC
Opcode	Naive Bayes	0.9492	0.6128	0.0230	0.8266	0.9455
	KNN	0.9521	0.9777	0.0345	0.9705	0.9842
	Random Forest	0.9595	0.9775	0.0290	0.9736	0.9947
	SGD Classifier	0.9462	0.9701	0.0387	0.9649	0.9914
	Average	0.9518	0.8840	0.0313	0.9339	0.9790
Library	Naive Bayes	0.8508	0.7705	0.0950	0.8494	0.9311
	KNN	0.7439	0.9892	0.2395	0.8549	0.8878
	Random Forest	0.9439	0.8328	0.0348	0.9105	0.9736
	SGD Classifier	0.9401	0.8014	0.0358	0.8969	0.9661
	Average	0.8711	0.8485	0.1004	0.8785	0.9397
Type	Naive Bayes	0.8346	0.1829	0.0254	0.6476	0.9020
	KNN	0.8570	0.9735	0.1142	0.9219	0.9493
	Random Forest	0.8751	0.9790	0.0982	0.9336	0.9741
	SGD Classifier	0.8208	0.9427	0.1447	0.8913	0.9452
	Average	0.8483	0.7701	0.0945	0.8496	0.9427
All	Naive Bayes	0.9494	0.6590	0.0247	0.8446	0.9710
	KNN	0.9572	0.9798	0.0307	0.9736	0.9859
	Random Forest	0.9678	0.9787	0.0228	0.9778	0.9959
	SGD Classifier	0.9581	0.9858	0.0303	0.9763	0.9948
	Average	0.9582	0.9008	0.0272	0.9431	0.9869

with the average accuracy 0.9431 and the average AUC 0.9869. The opcode feature has performed better, with the accuracy 0.9339 and the AUC 0.9790 on average, than the other two. The reason for this is that opcodes help describe the intended behaviours of the software quite well. This also explains why most existing work on malware detection employs the opcode feature. Using the opcode feature alone seems to be enough for malware detection in most cases, but the detection may evade easily by obfuscation techniques like adding unreachable codes. In some cases it is better to use the other two features and adding them will improve the detection. So it is useful to also take the type and library features into account. On average, the type feature has performed better than the library feature in term of AUC, although the library feature has produced slightly better results than the type feature with respect to accuracy. The results also show that Random Forest has performed the best for each feature, which is consistent with the above experiments. Note that, when Random Forest is employed, using type feature can get the best TPR. The opcode and library features have been used by lots of work in practice, so we believe that the type feature, which is not considered by most of existing work, can benefit malware detection as well in practice.

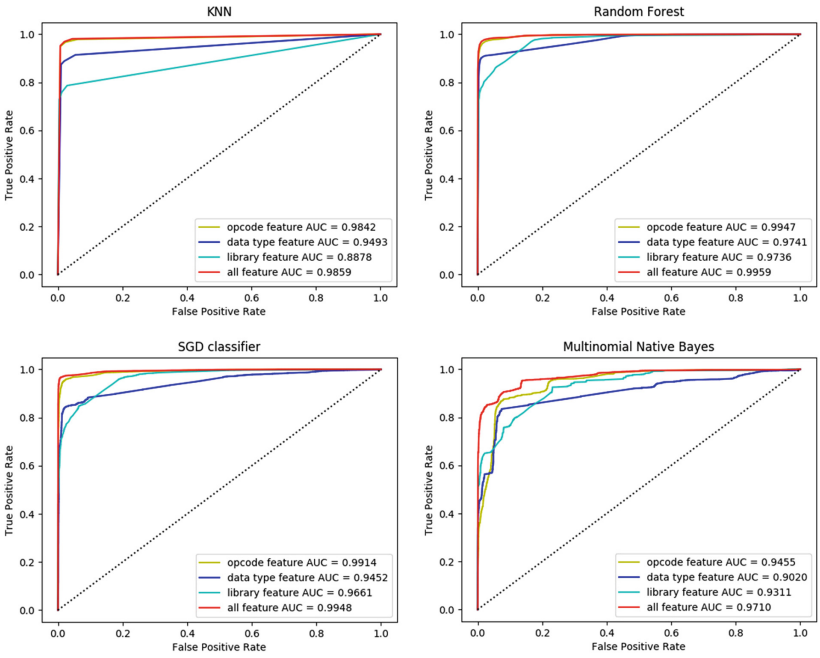


Fig. 4. ROC curve of different features

Time-Split Experiments. As Saxe and Berlin [17] point out, a shortcoming of the standard cross validation experiments is that they do not separate the ability to detect slightly modified malware from the ability to detect new malware samples and new versions of existing malware samples. In this section, to test our approach's ability to detect genuinely new malware or new malware versions, we ran a time split experiment. We first download the malware samples, which were collected dated from January 2017 to July 2017, from the DAS MALWERK website [24]. That is, all the malware samples are newer than the ones in our data set. We then train a classifier based on our data set using Random Forest. Finally, we use our classifier to detect these fresh malware samples.

The experimental results are shown in Table 5. About 81% of the samples can be detected by our classifier, which signifies that our approach can detect some malware samples and new versions of existing malware samples. However, the results also indicate that the classifier becomes ineffective as time passes. For example, the results detected by our approach for the samples collected in June and July are worse than random classification (the accuracies are not greater than 0.5). This is due to the high degree of freedom for malware writer to produce new malware samples continuously. Consequently, most malware classifiers become unsustainable in the long run, becoming rapidly antiquated as malware continues to evolve. This suggests that malware classifiers should be updated often with new data or new features in order to maintain the classification accuracy.

Table 5. Results of fresh samples

Date	Num	RF classifier	Accuracy
July, 2017	12	6	0.500
June, 2017	61	29	0.475
May, 2017	87	76	0.874
April, 2017	31	28	0.903
March, 2017	48	42	0.875
February, 2017	69	61	0.884
January, 2017	56	53	0.946
Total	364	295	0.810

Obfuscation Experiments. One thing that makes malware detection even more difficult is that malware writers may use obfuscation techniques, such as inserting NOP instructions, changing what registers to use, changing flow control with jumps, changing machine instructions to equivalent ones or reordering independent instructions, to evade the detection. In this section, we report some experiments to test our approach's ability to detect new malware samples that are obtained by obfuscating the existing ones.

The obfuscated malware samples are obtained as follows. Firstly, we use the free version of the commercial tool *Obfuscator* [25], which only supports changing code execution flow, to obfuscate 50 malware samples, which are randomly selected from our data set, yielding 50 new malware samples. Secondly, we use the open source tool *Unest* [26] to obfuscate 15 obj files, which are compiled from malware samples in C source codes through VS 2010[1], by the following four techniques, that is, (a) rewriting digital changes equivalently, (b) confusing the output string, (c) pushing the target code segment into the stack and jumping to it to confuse the target code, and (d) obfuscating the static libraries, yielding 60 new malware samples.

Table 6. Results of obfuscated malware samples

Tools	Number	RF classifier	Accuracy
Obfuscator	50	50	100%
Unest	60	60	100%

We used our classifier trained by Random Forest to detect the newly generated malware samples. The results are presented in Table 6. The results show that all the obfuscated malware samples have been detected by our classifier. This indicates that our classifier has some resistance to some obfuscation techniques. To change code execution flow, Obfuscator inserts lots of *jump* instructions. It seems

[1] We are able to obfuscate only the obj files compiled from C codes through VS 2010.

that the *jump* instructions may change the opcode feature, while the data feature and the library feature are still the same. Indeed, *jump* is commonly used in both malware and benign software. Therefore, this technique does not change the features we use and thus cannot evade our detection. Similar to Obfuscator, the techniques used in Unset make little changes on the features, thus cannot evade our detection either. Nevertheless, the techniques used here is rather simple. The integration of more (sophisticated) techniques will be considered in the future.

5 Conclusion

In this work, we have proposed a malware detection approach using various machine learning methods based on the opcodes, data types and system libraries. To evaluate the proposed approach, we have carried out some interesting experiments. Through experiments, we have found that the classifier trained by Random Forest outperforms others for our data set and all the features we have adopted are effective for malware detection. The experimental results have also demonstrated that our classifier is capable of detecting some fresh malware, and has a resistance to some obfuscation techniques.

As for future work, we may consider some other meaningful features, including static features and dynamic features, to improve the approach. We can employ the unsupervised machine learning methods or the deep learning methods to train the classifiers. More experiments on malware anti-detecting techniques are under consideration.

Acknowledgements. The authors would like to thank the anonymous reviewers for their helpful comments. This work was partially supported by the National Natural Science Foundation of China under Grants No. 61502308, 61373033 and 61672358, Science and Technology Foundation of Shenzhen City under Grant No. JCYJ20170302153712968.

References

1. McAfee Labs Threats Report, June 2017
2. Beaucamps, P., Filiol, E.: On the possibility of practically obfuscating programs towards a unified perspective of code protection. J. Comput. Virol. **3**(1), 3–21 (2007)
3. Ye, Y., Li, T., Adjeroh, D., Iyengar, S.S.: A survey on malware detection using data mining techniques. ACM Comput. Surv. **50**(3), 41 (2017)
4. Elovici, Y., Shabtai, A., Moskovitch, R., Tahan, G., Glezer, C.: Applying machine learning techniques for detection of malicious code in network traffic. In: Hertzberg, J., Beetz, M., Englert, R. (eds.) KI 2007. LNCS (LNAI), vol. 4667, pp. 44–50. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74565-5_5
5. Masud, M.M., Khan, L., Thuraisingham, B.: A scalable multi-level feature extraction technique to detect malicious executables. Inf. Syst. Front. **10**(1), 33–45 (2008)
6. Anderson, B., Storlie, C., Lane, T.: Improving malware classification: bridging the static/dynamic gap. In: ACM Workshop on Security and Artificial Intelligence, pp. 3–14 (2012)

7. Ye, Y., Li, T., Chen, Y., Jiang, Q.: Automatic malware categorization using cluster ensemble. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2010)
8. Santos, I., Brezo, F., Ugarte-Pedrero, X., Bringas, P.G.: Opcode sequences as representation of executables for data-mining-based unknown malware detection. Inf. Sci. **231**(9), 64–82 (2013)
9. Wang, T.Y., Horng, S.J., Su, M.Y., Wu, C.H.: A surveillance spyware detection system based on data mining methods. In: IEEE International Conference on Evolutionary Computation, pp. 3236–3241 (2006)
10. Ye, Y., Wang, D., Li, T., Ye, D.: IMDS: intelligent malware detection system. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1043–1047 (2007)
11. Ye, Y., Li, T., Huang, K., Jiang, Q., Chen, Y.: Hierarchical associative classifier (HAC) for malware detection from the large and imbalanced gray list. J. Intell. Inf. Syst. **35**(1), 1–20 (2009)
12. Ye, Y., Chen, L., Wang, D., Li, T., Jiang, Q., Zhao, M.: SBMDS: an interpretable string based malware detection system using SVM ensemble with bagging. J. Comput. Virol. **5**(4), 283 (2009)
13. Islam, R., Tian, R., Versteeg, S., Versteeg, S.: Classification of malware based on integrated static and dynamic features. J. Netw. Comput. Appl. **36**(2), 646–656 (2013)
14. Karampatziakis, N., Stokes, J.W., Thomas, A., Marinescu, M.: Using file relationships in malware classification. In: Flegel, U., Markatos, E., Robertson, W. (eds.) DIMVA 2012. LNCS, vol. 7591, pp. 1–20. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37300-8_1
15. Tamersoy, A., Roundy, K., Chau, D.H.: Guilt by association: large scale malware detection by mining file-relation graphs. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2014)
16. Mohamed, G.A.N., Ithnin, N.B.: Survey on representation techniques for malware detection system. Am. J. Appl. Sci. **14**(11), 1049–1069 (2017)
17. Saxe, J., Berlin, K.: Deep neural network based malware detection using two dimensional binary program features. In: 2015 10th International Conference on Malicious and Unwanted Software (MALWARE), pp. 11–20 (2015)
18. Hardy, W., Chen, L., Hou, S., Ye, Y., Li, X.: DL4MD: a deep learning framework for intelligent malware detection. In: Proceedings of the International Conference on Data Mining (2016)
19. Ye, Y., Chen, L., Hou, S., et al.: DeepAM: a heterogeneous deep learning framework for intelligent malware detection. Knowl. Inf. Syst. 1–21 (2017)
20. Jordaney, R., Sharad, K., Dash, S.K., Wang, Z., Papini, D., Nouretdinov, I., Cavallaro, L.: Transcend: detecting concept drift in malware classification models. In: 26th USENIX Security Symposium (USENIX Security 2017), pp. 625–642 (2017)
21. Xu, Z., Wen, C., Qin, S.: Learning types for binaries. In: Duan, Z., Ong, L. (eds.) ICFEM 2017. LNCS, vol. 10610, pp. 430–446. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68690-5_26
22. Microsoft Malware Classification Challenge. https://www.kaggle.com/c/malware-classification
23. theZoo aka Malware DB. http://ytisf.github.io/theZoo/
24. DAS MALWERK. http://dasmalwerk.eu/
25. Obfuscator. https://www.pelock.com/products/obfuscator
26. Unest. http://unest.org/

Load Pattern Shape Clustering Analysis for Manufacturing

Mark Junjie Li and Weiguang Liu(✉)

College of Computer Science and Software Engineering,
Shenzhen University, Shenzhen 5518060, China
jj.li@szu.edu.cn, qiantangjiang@live.com

Abstract. Manufacturing is dominant sector in electricity power consumption users. Currently an AMI infrastructure is widely deployed to collect real-time customer electricity consumption data. The knowledge of customer electricity consumption profiling can be used to smart grid dispatching, marketing and pricing based on analyzing and mining the load profiling in different industries, seasons and time periods. This paper investigates an auto fuzzy K-means clustering algorithm for load pattern shape data, which can find out the optimal number of power behaviors pattern. A data-preprocessing framework for load pattern shape retrieval is shown to reduce the dimensions efficiently. In the other hand, a validation index is applied in the algorithm, which balances the scattering within the clusters and the separation between the clusters, to discover the real electricity consumption pattern automatically based on the density of load time series data. The experimental results show this algorithm can efficiently discover the electricity consumption behavior, such as order, continual load and continual low load profiling, and different over time working pattern in weekend and public holiday. The results can predict the electricity consumption behavior for different type of industry, which will benefit for Demand Side Management smart grid dispatching and marketing.

Keywords: Smart grid · Fuzzy K-Means clustering
Load pattern shape · Load profiling · Validation index

1 Introduction

Smart grid generates a large volume data, which imply the customers power consumption profile, through Advanced Metering Infrastructure (AMI). Hidden in these golden data, the valuable knowledge, such as power consumption pattern, customers growth forecasting, etc., can be discovered with data mining approaches. The analyses of smart grid data have broad prospects for the application of the power industry such as grid resource planning, ladder price management, user power behavior grouping, power sales, stealing electricity supervision, etc., and can raise the power industry enterprise quality of management and economic benefit [1–3]. At the same time, the integration of large-scale data analysis

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 67–80, 2018.
https://doi.org/10.1007/978-3-319-73830-7_7

and economic data can indirectly predict the economic growth, energy consumption, production status of the region and provide information for the government's macroeconomic management decision-making.

Load pattern shape clustering analysis is an active area in smart grid research [4–10], the enterprise's electricity behavior analysis, for electricity pricing, project selection and other demand management to provide a wealth of user load information. Williams [1] analyzed the 250 household power consumption profile in Ireland with the average Load Pattern Shape (LPS). Zhang and Gu [10] compared the extraction of Load Pattern Shape (LPS) and its application in the demand side management.

In the field of power load pattern shape clustering analysis, fuzzy K-Means algorithm is one of the important analysis methods [11–15]. The K-Means clustering is hard clustering, data is divided into distinct clusters, where each data point can only belong to exactly one cluster. In fuzzy K-Means clustering, the difference is the membership that indicate the degree to which data object belong to each cluster. Fuzzy K-Means algorithm has lower initial clustering center sensitivity and better clustering result than K-Means algorithm when the data is not clearly defined. Song et al. [12] discussed the how to select the number of clusters and the initial clusters center using fuzzy clustering base on cloud model, and the typical power load patterns extraction and clustering. Chicco [15], the fuzzy K-Means algorithm is used to select the optimal fuzzy coefficient m for typical electrical load pattern clustering applications and compare the selection of the optimal cluster number K by different clustering validation indexes.

In this paper, the main contribution is to discover the typical power consumption pattern by typical load pattern shape clustering analysis. We use the data from the five manufacturing sectors in Dongguan, Guangdong Province, China, which include 12,426 enterprises electricity consumption data in May 2012. Based on the fuzzy k-means type clustering algorithm, this paper discovered some typical patterns of electricity consumption patterns in working days, Saturdays, Sundays and holidays. The clustering evaluation method calculates the sum of intra-cluster compactness and inter-cluster dispersion, finds the optimal fuzzy classification result and overcomes the shortcoming of traditional clustering evaluation method to find the best clustering result by find inflection point. This approach can be more effectively analyze and mining the production behavior and power consumption behavior of different manufacturing enterprises, according to the enterprise's power consumption behavior for customer clustering, compared to the current production type clustering, according to the type of customer clustering can better provide services for power supply enterprises.

The organization of this paper is as follows: Sect. 2 will introduce the background of typical load pattern shape. In Sect. 3, we describe the pre-processing method based on typical load pattern shape. Typical load pattern shape data discovering with clustering based on the fuzzy clustering methodology will be introduced in Sect. 4. Some case studies for the different sectors will be discussed in Sect. 5. And in the last part of this paper will summarize this study and prospects for future work.

2 Load Pattern Shape

The power consumption time series data can reflect the regular that the change of enterprise power consumption. The smart meter terminal will collect seven types of data, including voltage, current, active power, reactive power, apparent power, power factor and 3-phase current unbalance rate, from the smart power meters every 15 min.

Figure 1 shows a factories Load Pattern Shape (LPS) in May 2012, each window plots the power load pattern curve for 24 h. It can be found that in the working days from Monday to Friday, the daily load pattern sharp is similar, showing three shifts mode. But the load pattern has a significant change at weekends, on Saturday the mode is two shifts mode, on Sunday the mode is low-power consumption mode. May 1 (Labor Day) is a statutory holiday, so this day also low-power consumption mode. This can be illustrated the enterprise are not work on Sunday and holidays.

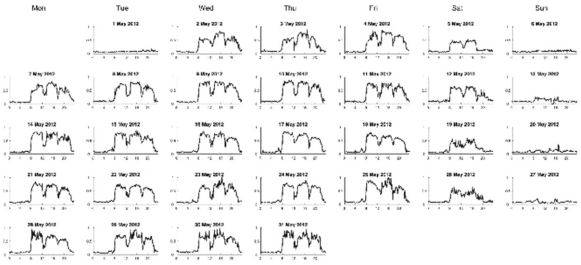


Fig. 1. A factory load pattern shape in May 2012

We observe the load pattern curve of working day from morning to night, the peak appears between 8 and 24 o'clock, that is production time. But the curves showed a significant decrease at noon and nightfall, it represents that this time for the meals and rest. And we can imagine that the production mode of enterprise is 3-shifts mode. We can observe a little the peak from the curves on Saturday, this represents that part of staff work overtime with production plan or device debugging. So, the power load curve can be divided into four typical load shape curves according to different date types, as shown in Fig. 2, clear display the enterprises power pattern shape types, the working day is 3-shifts mode, on Saturday is 2-shifts mode, the low-power consumption mode on Sunday and holiday. Through clustering analysis for the different date types, it can effectively find all kind of power consumption patterns and characteristics for enterprises.

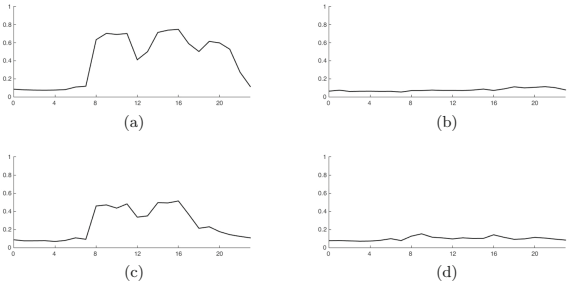


Fig. 2. Load pattern shape in different date type: (a) working day (b) holiday (c) Saturday (d) Sunday

3 Load Pattern Shape Data Pre-processing

For the initial data, we should do some pre-processing job before clustering is performed. Pre-process procedure consists of four main steps, depicted on the follows:

1. Data clean. Check the load pattern data for each enterprise, and modified or delete data that has significant errors, such as missing datanoise data, or inconsistent data.
2. Normalization. The power load data has been cleaned, should do standard normalization process, then we can draw the power load curve use the processed data. We can get the base load by computing the maximum of one users month load data, and the load data mapping the range of $[0, 1]$, eliminate some factors due to the load differences of various enterprises, and really reflects the differences between enterprises of power consumption pattern.
3. Dimensionality reduction. Power load time series data sets as $D = \{D_1, D_2, \cdots, D_m\}$, and the D_m as daily power load time series data sets, $D_m = \{d_m^1, d_m^2, \cdots, d_m^n\}$, the d_m^n is the sample points that collect by the smart power meters every 15 min, so the daily power load time series data sets include 96 data collecting points, and the month dimensionality is 2976. We use Piecewise Aggregate Approximation (PAA) for the dimensional reduction, and reduce the 15-min interval data into hourly average data.
4. Data extract. It is Strong periodical change of enterprises power consumption pattern. we can extract load sharp data for different date types. In this paper, we extract three power load pattern data of working days (Monday to Friday), weekends (Saturday to Sunday) and holidays (Labor Day).

As shown in Fig. 2, power load time series data from Fig. 1 for an entire month after pre-process, the 2976 dimensions of one month will be clustering 24 dimensions per day according to the date type of working day, holiday, Saturday and Sunday, and divided four type load pattern shape data, then clustering analysis for different date type, to find the kind of enterprises power load pattern and character.

4 Auto Fuzzy K-Means Algorithm for Load Pattern Shape

[11–15] show fuzzy K-Means algorithm is still an effective method for load pattern clustering analysis. The above papers all rely on the traditional validation index to find the "turning point" and choose the optimal clusters number. But most of the index apply to the cluster structure that have high degree of distinction, and it is difficult to find the "turning point" when the cluster structure is fuzzy. This paper based on fuzzy K-Means clustering algorithm, and use the validation index that applicable to fuzzy clustering. The index can find the optimal cluster number among the different clustering results, and can auto-clustering.

4.1 Fuzzy K-Means Algorithm

Fuzzy K-Means algorithm is widely used in practical applications. For a given set X, the each element x_i consists of m numeric attributes, that $x_i = [x_{i,1}, x_{i,2}, \cdots, x_{i,m}]^T \subset R^s$, the objective of fuzzy K-Means algorithm is minimizing the global cost function:

$$\min P(U, Z) = \sum_{j=1}^{k} \sum_{i=1}^{n} u_{i,j}^{b} \|x_i - z_j\|^2 \tag{1}$$

The membership matrix is $U = [u_{i,j}]_{n \times k}$ size of $n \times k$, $u_{i,j}$ is the membership degree from sample point i to jth cluster z_j, the matrix of cluster center is $Z = [z_1, z_2, \cdots, z_k]^T \subset R^s$ size of $k \times m$. The clustering membership function of each sample point is normalized, $\sum_{j=1}^{k} = 1$, $u_{i,j} = 1$, $u_{i,j} \in (0, 1]$, $1 \leq i \leq n$. b is a free parameter that control different cluster mixing. When $b = 0$, the objective function $P(U, Z)$ force each simple to belong to only one cluster. When $b > 0$, the objective function allows each simple to belong to multiple clusters. The default parameter of b is 2. The fuzzy K-Means algorithms objective function minimize method adopt iterative computation, the iterative function of the cluster center Z and the membership matrix U as follows:

$$U_{i,j} = \begin{cases} (\sum_{i=1}^{k} (\frac{x_i - z_j}{x_i - z_l})^{\frac{2}{b-1}})^{-1} & , \|x_i - z_l\| > 0 \\ 1 & , \|x_i - z_l\| = 0 \\ 0 & , \|x_i - z_l\| = 0, l \neq j \end{cases} \tag{2}$$

$$Z_{j,l} = \frac{\sum_{l=1}^{n} x_{i,j} x_{i,l}}{\sum_{i=1}^{n} u_{i,j}} \tag{3}$$

4.2 Validation Index for the Optimal Number of Clusters

Fuzzy K-Means clustering algorithm to set a different number to initialize the cluster center, to be different clustering results. But how to judge whether optimal clusters in clustering, the need validation index of clustering results. The clustering result evaluation rely on two contradictory indexes: the scattering within

the clusters and separation between the clusters. The separation within the clusters of single cluster smaller, the more closely of the cluster structure, clustering results better; the distance between each cluster center larger, the greater difference between clusters, and clustering results better.

The paper [16] compare the effect of several algorithms on the load curve clustering. There are five validation index functions for clustering result: CDI (clustering dispersion indicator), MSE (mean square error), SI (scatter index), DBI (Davies-Bouldin indicator), MIA (mean index adequacy), however, these indexes are still not fully determine the quality of algorithm and the number of clusters is appropriate. The paper use CDI, SI, MIA indexes and inflection point of cluster number curve to find the optimal cluster number, but it cant clearly determine the "inflection point". The "inflection point" of the clustering validation index appear if each cluster can be distinguished clearly.

In the paper [17] proposes a validation index.

$$\nu(U, Z, k) = SCATTER(k) + \frac{DISTANCE(k)}{DISTANCE(k_{max})} \tag{4}$$

The validation index is composed of two parts: the scattering within the clusters and separation between the clusters, and reflect the clustering results whether are the optimal. When the cluster structure is more compact, the clustering results of the same cluster are more similar, the compactness within the clusters smaller of the first item of the verify validation index; when the distance between clusters center larger, the higher discrimination between clusters, the separation between the clusters lower of the second item of the verify the validation index.

$$SCATTER(k) = \frac{\frac{1}{k}\sum_{i=1}^{k} \|\sigma(z_i)\|}{\|\sigma(S)\|} \tag{5}$$

The compactness within the clusters represents the average compactness of clustering result. The $\|\sigma(S)\|$ represents the compactness of all sample data, $\sigma(S) = [\sigma_1(S), \sigma_2(S), \cdots, \sigma_m(S)]^T$, $\sigma_j(S) = \frac{1}{n}\sum_{i=1}^{n}(x_{i,j} - \overline{x}_j)^2$, $\overline{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{i,j}$, $\|\sigma(Z_i)\|$ meaning the compactness within the clustering result for each cluster, and $\sigma(z_l) = [\sigma_1(z_l), \sigma_2(z_l), \cdots, \sigma_m(z_l)]^T$, $\sigma_m(z_l) = \frac{1}{n}\sum_{i=1}^{n} u_{i,j}(x_{i,j} - z_{l,j})^2$. When the K that the numbers of clustering result larger, the more compact within the clusters structure, the lower of the value $SCATTER(k)$ that meaning compactness within the cluster, and through the process of normalization, the range of $SCATTER(k)$ is $(0, 1]$.

The clustering validation index as $\nu(U, Z, k)$, and the separation between the clusters represent the separation degree meaning as follow:

$$DISTANCE(k) = \frac{d_{max}^2}{d_{min}^2} \sum_{i=1}^{k} \frac{1}{\sum_{j=1}^{k} \|z_i - z_j\|^2} \tag{6}$$

The d_{min} is the minimum distances between the clusters center, $d_{min} = \min_{i \neq j} \|z_i - z_j\|$, the d_{max} is the maximum distances between the clusters center, $d_{max} = \max_{i \neq j} \|z_i - z_j\|$. The validation index function of cluster between the

clusters is $DISTANCE(k)$, $\sum_{j=1}^{k} \|z_i - z_j\|^2$ is the sum of the distance between cluster center z_i and the other cluster centers. When the clustering center has obvious distinguish, the d_{min} and the d_{max} between the cluster centers are large, the distances sum of cluster centers is large, so the $DISTANCE(k)$ is small, and the clustering result is optimal. When the number of cluster center is too much, the d_{min} and the d_{max} between the cluster centers are small, the distances sum of cluster centers is small, so the $DISTANCE(k)$ is large. This feature can be used to effectively control the compactness within the clusters metric $SCATTER(k)$, which divides the clustering results into too many small cluster structures. In order to evaluate the clustering results of different initial cluster centers, we use the maximum of $DISTANCE(k_{max})$ do the process of normalization for the separation between the clusters.

The clustering validation index $\nu(U, Z, k)$ is a metric that balancing the compactness within the clusters and the separation between the clusters. When the clustering result are as compact as in clusters and as separate as between clusters, the clustering result are optimal and the value of clustering validation index $\nu(U, Z, k)$ is small. So, we can set the maximum of clusters K_{max} through estimate in advance, and try set the different value k as initial clusters number by decrementing each time, and combine the clustering validation index $\nu(U, Z, k)$ to find the optimal clustering number $K_{optimal}$ and clustering result. This method only set the max clustering number through estimate in advance, and the optimal clustering result can be obtained automatically.

The main steps of the auto fuzzy K-Means clustering algorithm as follows.

Input: the max clustering number K_{max} mixing parameters b, sample point sets $X = [x_1, x_2, \cdots, x_n]^T$
Output: the optimal clusters number $K_{optimal}$ cluster centers $Z = [z_1, z_2, \cdots, z_k]^T$ and membership matrix $U = [u_{i,j}]_{n \times k}$

```
Initial cluster centers (Z) ;
for k = K_max : 1 do
    Step 1: Using Fuzzy K-Means algorithm update the membership matrix (U)
    and cluster centers (Z) ;
    Step 2: Evaluate whether the clustering center point converges steadily; if
    not converge, go to the step 1;
    Step 3: Computing validation index ν(k) for this clustering results
end
Output K_optimal = arg min_{1≤k≤K_max} ν(k), Z_optimal and U_optimal
```

Algorithm 1. Auto fuzzy K-Means clustering algorithm

5 Experimental Results

In this paper, we analysis the power consumption behavior for industrial enterprises in Dongguan, May 2012. We use Piecewise Aggregate Approximation (PAA) for the dimensional reduction, and obtain the enterprises average monthly power load time series data on working days, Saturdays, Sundays and holidays. We set

the different cluster numbers for the four date type data, and applying the fuzzy K-Means clustering algorithm, combining the validation index v(U, Z, k) [17], and automatic identification of real and effective power consumption pattern. By comparing the power consumption behavior among different industries, we can analyze the power consumption patterns of major industrial in Dongguan region, and explore the characteristics of different industry power consumption patterns.

5.1 Automatic Selection for the Optimal Number of LPS Clusters

For the average monthly power load time series data on working days, this paper applied the fuzzy K-Means algorithm, and setting the different clusters number, combining the validation index $\nu(U, Z, k)$ is used to validation the compactness within the clusters and the separation between the clusters, obtain the real clustering result for enterprises user. In this paper, set the initial clusters number range of $[2, 50]$, as shown in Fig. 3, the minimum validation index function $\log(\nu(U, Z, k))$ is 33, so the optimal clustering number for average monthly working day is 33.

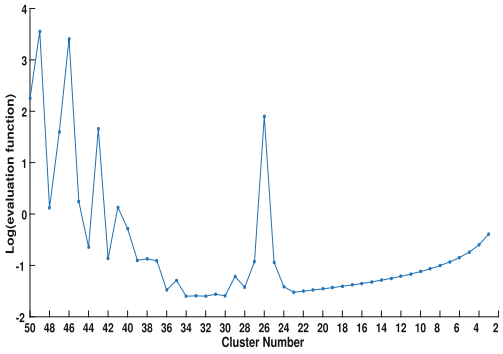


Fig. 3. Validation index result for load pattern shape clustering in working day

The optimal clustering result for power load on working day are shown in Fig. 4. The Fuzzy K-Means clustering automatic selection 33 based clusters, each cluster has a compact cluster structure within cluster, and each cluster has a distinct recognition between clusters. According to the same characters of the 33 based clusters, that be divide five power consumption patterns, as 2-shifts mode, 3-shifts mode, high-stability power consumption mode, low-stability power consumption mode, and other non-cycle power consumption mode. In the same power consumption mode, there are obvious characteristics between the different clusters. For example, 2-shifts mode can be divided into non-working time low power consumption of 2-shifts mode and non-working time continued power consumption of 2-shifts mode. In order to effective analyze the characteristics of power consumption patterns between the different industries, we divided the 33 basic cluster results into

5 kinds of common electricity patterns. For the Saturday, Sunday and statutory holidays, we use the same fuzzy K-means clustering algorithm to get the optimal basic cluster structure, and then merged into the above five kinds of electricity patterns, with a comprehensive analysis.

5.2 Knowledge Discovery for Load Pattern Shape

In this paper, for working day, Saturday, Sunday, holiday four date modes, different sectors of power load mode automatic clustering. From the optimal clustering results, we mining the power consumption characteristics for the same date mode under the different industries and the same industry under the different date modes. From the point of view of clustering, analysis the enterprises power load model.

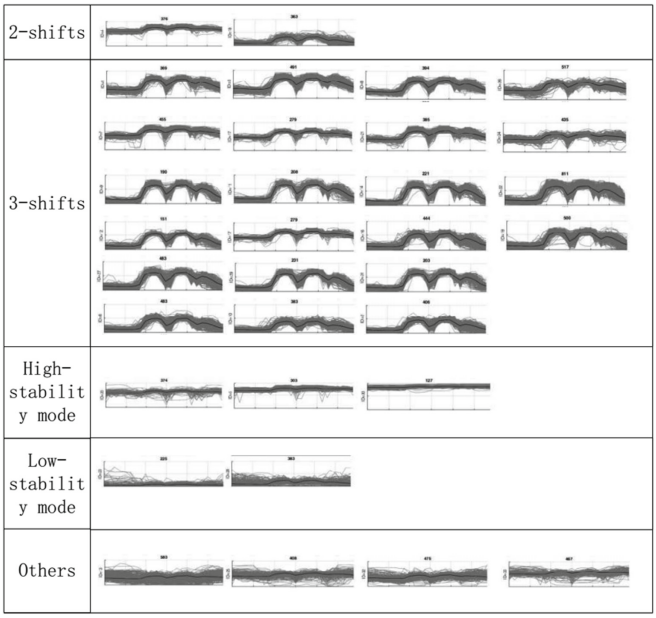


Fig. 4. Auto fuzzy K-means clustering result for load pattern shape in working day

As shown in Fig. 4, the enterprises load pattern characteristics of 3-shifts mode have three peaks, and the peak appeared at 9 am, 3 pm and 10 pm, the valley appeared at 12 am and midnight. In terms of load pattern, the 3-shifts load pattern shows the wave, and it almost no smooth power time. Most of these load pattern enterprises are labor intensive industry, but also all types of manufacturing in the main operation mode.

The load curve of 2-shifts mode is very similar to the 3-shifts mode, but the enterprises load pattern characteristics of 2-shifts mode have two peaks, and the

peak appeared around at 10 am and 4 pm, the valley appeared at 12 am, 6 pm and midnight. From 6 pm to the next morning, the vast majority of users almost no electricity, only a small part of the user has the signs of power consumption.

High-stability power consumption mode and Low-stability power consumption mode have the same load curve characteristics is the load continuously and stability, and almost no significant peak and valley of power consumption. The characteristics of this king of enterprises are production of uninterruptible and cannot power off. For example, some chemical synthesis in the continuous manufacturing industry, production line operation is continuous, the loss of power outages than the cost of night-shifts higher, so this types enterprise is low dependence on the artificial.

The other load patterns may be derived from the lighting electricity, staff quarters power consumption and some non-cycle power consumption industries like the order-driven industry. Due to some irregular power consumption data are mixed in the data collection, some models are not very obvious.

Table 1. Load pattern shape clustering result for working day

Industry type	2-shifts (%)	3-shifts (%)	High-stable (%)	Low-stable (%)	Other (%)
Electrical equipment	7.19	71.22	5.25	4.10	12.23
Textile products	2.91	81.08	2.08	4.64	9.28
Metal products	5.79	67.93	2.58	7.06	16.64
Plastic products	6.48	54.73	9.05	4.66	25.09
Communications equipment	6.74	68.58	11.50	3.12	10.06

The Table 1 summarizes the ratios of the five work patterns for all enterprise types, 3-shifts patterns is the mainly patterns of working day patterns, it proved that the regional is mainly labor-intensive pattern industry, overtime is the main mode of working. The textile products industry is a typical labor-intensive pattern industry, more than 80% of the textile products industry is 3-shifts pattern, but also proved that the textile industry is typical three-peak style enterprises. Stable power mode enterprises mainly electrical equipment industry and communications equipment industry, due to this type enterprises higher automatic level, less dependent on the artificial, and some of the industry's enterprises need to maintain long-term high-power mode. Others mode of load patterns, mainly in plastic products industry, due to some enterprises of the plastic products industry depend on customer orders, cant become the cycle power consumption pattern.

The power consumption pattern of working days reflects the electricity consumption behavior in the daily production. In view of the labor-intensive employment form of the Pearl River Delta (PRD), the enterprises in addition to electricity patterns in the working day performance of 3-shifts mode, the weekend power consumption mode for the continuous overtime of the electricity mode. Through compare between Saturday and Sunday clustering characters

Table 2. Load pattern shape clustering result for weekend

Industry type	2-shifts (%)		3-shifts (%)		High-stable (%)		Low-stable (%)		Other (%)	
	Sat	Sun	Sat	Sun	Sat	Sun	Sat	Sun	Sat	Sun
Electrical equipment	24.46	42.45	52.66	12.30	6.04	5.76	5.40	25.11	11.44	14.39
Textile products	15.82	55.15	72.04	21.95	2.80	3.15	3.21	13.98	6.13	5.77
Metal products	25.59	46.86	52.53	12.94	3.88	3.62	5.67	23.10	12.33	13.49
Plastic products	28.65	35.20	38.86	9.80	9.29	11.21	4.04	20.94	19.16	22.86
Communications equipment	27.23	37.29	46.14	10.56	11.37	11.44	3.92	21.80	11.33	18.91

of average power load, as shown in the Table 2. In the view of the distribution of power consumption patterns, the 3-shifts mode in five industries have a significant decline, and the enterprises power consumption patterns transform into 2-shifts mode and low-stability power consumption mode, this shows that the five major industries in the weekend reducing the intensity of production to varying degree. The Low-stability power consumption mode has a significant promote, while reflecting a part of the industrial enterprises working on Saturday. Compared with other models, the high-stability power mode for high automation production line jobs, the power mode performance is stable, and this classes of power consumption mode are irrelevant to date type.

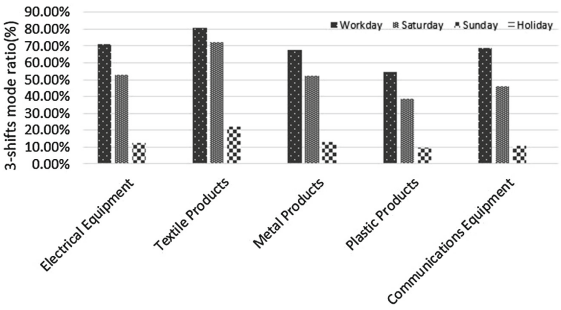


Fig. 5. Trend comparison for 3 shift term load pattern

As shown in the Fig. 5, there are four date types trend comparison for 3-shifts load pattern. The 3-shifts power consumption on Saturday than the working day decreases by an average of 16%, the 3-shifts power consumption on Sunday than the working day decreases by an average of 55%, and only about 15% enterprises are 3-shifts power consumption patterns. As shown in the Fig. 6, there are four date types trend comparison for 2-shifts load pattern. The 2-shifts power consumption on Saturday than the working day increased by an average of 19%, the 2-shifts power consumption on Sunday than the working day increased by an average of 38%. It can be inferred that a apart of enterprises is 3-shifts mode on working day, and transform into 2-shifts mode on Saturday; some enterprises

is 3-shifts mode on working day, and transform into 2-shifts or stable mode on Sunday. It can be found in the result of Figs. 5 and 6, that textile industry in the Saturday and Sunday still maintain the highest ratio in 3-shifts mode and 2-shifts mode, this phenomenon shows that the textile industry in May the period of labor intensity is higher.

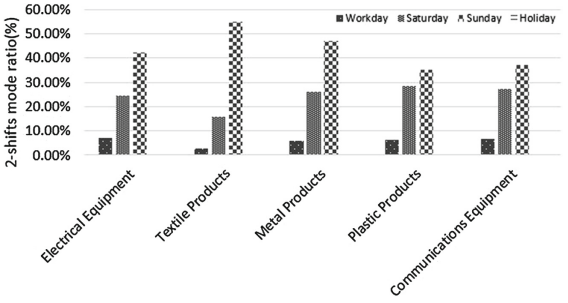


Fig. 6. Trend comparison for 2 shift term load pattern

In this paper, in addition to comparison of the five industry working days and weekend power consumption patterns, and then analyzes the characteristics of the power consumption pattern of the holidays. As shown in the Table 3, during the May 1 Labor Day national holidays, nearly 95% of the enterprises power consumption pattern for the low-stability power mode, this shows that most companies comply with national holiday requirements, stop production and the power load substantially reduced.

Table 3. Load pattern shape clustering result for holiday

Industry type	2-shifts (%)	3-shifts (%)	High-stable (%)	Low-stable (%)	Other (%)
Electrical equipment	0.00	0.50	0.79	96.26	2.45
Textile products	0.00	1.37	1.13	93.34	4.16
Metal products	0.00	0.87	0.98	95.14	3.01
Plastic products	0.00	0.58	1.34	95.34	2.74
Communications equipment	0.00	0.77	2.85	91.15	5.23

6 Conclusion

In this paper, we applied the load shape pattern method, which can be more accurately found the load characteristics of different users, it application in the demand side management is mainly reflected in the following two aspects:

1. Demand side management measure customization based on the power consumption pattern. In order to ensure customer satisfaction, we can formulate the corresponding demand-side management measures according to its power load shape characteristics. For example, to the mining power consumption patterns in this paper, like high-stability power consumption mode and low-stability power consumption mode, instead of using De-Peaking and Peak Stagger measures to optimize power consumption of the enterprises users, it is better to recommend them to be more suitable for energy-efficient demand-side management programs such as collaborative energy management. For other patterns of high volatility load, the flexibility of their production planning is relatively large, and the valley period did not schedule the production plan, this situation is not suitable the De-Peaking strategy, it should lead to the adoption of time peak shifting strategy to obtain more effective demand side management effect. 3-shifts mode with multiple peak characteristics, it can be recommended that the type of load, such as interruptible load, emergency avoidance peak demand-side management program.
2. In the framework of the ordered avoid the peak according to the user's typical load shape arranged reasonable user mix and measures to optimize the distribution of the load gap, enhance the orderly electricity effect, ease the power supply imbalance, reducing enterprise electricity cost.

In this paper, the main idea is applying the fuzzy K-Means automatic clustering approach to find the power load characteristics. The power load curve is used for prediction and feedback. The algorithm proposed in this paper combines the validity of Piecewise Aggregate Approximation (PAA) and the accuracy of the fuzzy K-means automatic clustering algorithm, and this approach is useful in the processing of complex cluster structure data. The cluster result show that the 3-shifts pattern is the main production mode in Dongguan, this supports the argument that the labor-intensive factories in China's manufacturing sector are the main industrial structure. In this paper, the fuzzy K-means clustering method proposed provides a new analysis method for studying smart grid data, and it can predict the power consumption behavior for different classes enterprises, there are significant implications for power switching, orderly power utilization and marketing.

References

1. Williams, J.: Clustering household electricity use profiles. In: Proceedings of Workshop on Machine Learning for Sensory Data Analysis, p. 19. ACM (2013)
2. Chuan, L., Rao, D.V., Ukil, A.: Load profiling of Singapore buildings for peak shaving. In: 2014 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), pp. 1–6. IEEE (2014)
3. Colley, D., Mahmoudi, N., Eghbal, D., et al.: Queensland load profiling by using clustering techniques. In: 2014 Australasian Universities Power Engineering Conference (AUPEC), pp. 1–6. IEEE (2014)

4. Lezama, F., Rodríguez, A.Y., de Cote, E.M., Sucar, L.E.: Electrical load pattern shape clustering using ant colony optimization. In: Squillero, G., Burelli, P. (eds.) EvoApplications 2016. LNCS, vol. 9597, pp. 491–506. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31204-0_32
5. Chicco, G., Ionel, O.M., Porumb, R.: Formation of load pattern clusters exploiting ant colony clustering principles. In: EUROCON 2013, pp. 1460–1467. IEEE (2013)
6. Notaristefano, A., Chicco, G., Piglione, F.: Data size reduction with symbolic aggregate approximation for electrical load pattern grouping. IET Gener. Transm. Distrib. **7**(2), 108–117 (2013)
7. Mets, K., Depuydt, F., Develder, C.: Two-stage load pattern clustering using fast wavelet transformation. IEEE Trans. Smart Grid **7**(5), 2250–2259 (2016)
8. Wang, Y., Chen, Q., Kang, C., et al.: Load profiling and its application to demand response: a review. Tsinghua Sci. Technol. **20**(2), 117–129 (2015)
9. Jianwen, H., Yuping, Y.: Consumers load pattern recognition based on cluster ensemble. Comput. Appl. Softw. **31**(12), 237–241 (2014)
10. Zhang, T., Gu, M.: Overview of electricity customer load pattern extraction technology and its application. Power Syst. Technol. **40**, 804–811 (2016)
11. Zakaria, Z., Lo, K.L.: Two-stage fuzzy clustering approach for load profiling. In: Proceedings of the 44th International Universities Power Engineering Conference (UPEC), pp. 1–5. IEEE (2009)
12. Song, Y., Li, C., Qi, Z.: Extraction of power load patterns based on cloud model and fuzzy clustering. Power Syst. Technol. **38**(12), 3378–3383 (2014)
13. Anuar, N., Zakaria, Z.: Electricity load profile determination by using fuzzy cmeans and probability neural network. Energy Procedia **14**, 1861–1869 (2012)
14. Anuar, N., Zakaria, Z.: Determination of fuzziness parameter in load profiling via Fuzzy C-Means. In: 2011 IEEE Control and System Graduate Research Colloquium (ICSGRC), pp. 139–142. IEEE (2011)
15. Chicco, G.: Overview and performance assessment of the clustering methods for electrical load pattern grouping. Energy **42**(1), 68–80 (2012)
16. Li, H.L., Guo, C.H.: Piecewise aggregate approximation method based on cloud model for time series. Control Decis. **26**(10), 1525–1529 (2011)
17. Sun, H., Wang, S., Jiang, Q.: FCM-based model selection algorithms for determining the number of clusters. Pattern Recognit. **37**(10), 2027–2037 (2004)

A New Architecture of Smart House Control System

Lianghai Yang, Feiqiao Mao(✉), and Jiaqi Tan

Shenzhen University, Shenzhen, China
i@silas.hk, feiqiao@szu.edu.cn, 451406896@qq.com

Abstract. Smart house control system is very important in practice and is one of the research focuses. This paper proposes a new architecture of smart house control system based on Ad-Hoc wireless sensor network and cloud service. The architecture is open and flexible. A smart house control system named Airome is developed based on this architecture to show the effectiveness of the architecture.

Keywords: Smart house · Cloud services · Arduino · Ad-Hoc Wireless sensor network

1 Introduction

With the development of the information technology and sensor network, various things are able to interact with the Internet, which marks the appearance of Internet of Things. A great number of household appliances are provided in markets. For example, some air conditioners can be controlled by mobile phones or regulate the temperature based on the surrounding environment. It could be years before families purchasing their new electrical appliances and it will cost the families a lot to change traditional appliances to the intelligent ones. So, this paper proposes a new architecture of smart house control system, that can accept remote commands emitted by mobile phones and computers, translate it into signal that can be comprehended by traditional household appliances, and then control the traditional appliances, like electric kettles and air conditioners. This system is an interface, playing the role of a bridge connecting the Internet and things, serving as the transition between intelligent and traditional appliances.

This new architecture of smart house control system is illustrated in aspects of the design and realization of hardware, design of wireless sensor network, and the design of software.

This study was supported by Guangdong Natural Science Foundation (2016A030313036), Shenzhen Science and Technology Foundation (JCYJ20150324140036842) and Guangdong Graduate Education Project (12JGXM-MS29).

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 81–89, 2018.
https://doi.org/10.1007/978-3-319-73830-7_8

2 Related Work

Tao et al. [1] proposed ontology-based data semantic management and application in IoT- and cloud-enabled smart homes, customizing smart home products for clients accurately. Ahmed and Kim [2] discussed the advantages and challenges in named data networking-based smart home, proposing that named data networking makes the data service of smart home more robust. Smirek et al. [3] analyzed the similarity and difference between Eclipse Smart Home (ESH) platform and Universal Remote Console (URC) platform, and proposed a concept combining URC and ESH. Zualkernan et al. [4] introduced a smart home system named Infopods, which is based on ZigBee and can control and monitor the house by multiple users in the same time.

Wang et al. [5] and his colleagues proposed a smart home control system based on ZigBee. Soliman et al. [6] and his colleagues introduced a smart home system based on cloud services and web service, of which the hardware is constructed mainly with Arduino and ZigBee. The difference between the system proposed in this paper and the work of former researchers is that the system in this paper is of both distributed wireless sensor network and cloud service.

3 The Architecture Based on Ad-Hoc and Cloud Service

3.1 The Physical Structure and Hardware Design

The proposed physical architecture as shown in Fig. 1 consists of six parts which are client, cloud server, home router, gateway, Ad-hoc wireless sensor network and household appliances.

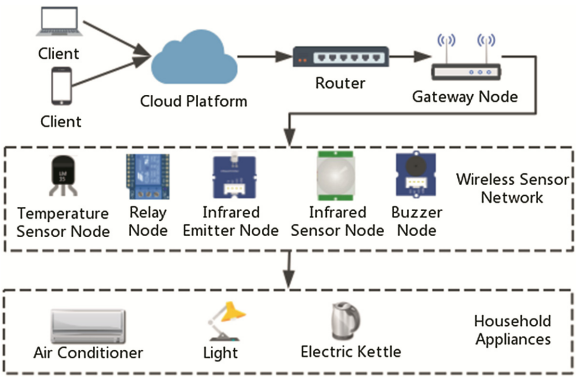


Fig. 1. Physical structure

3.2 The Hardware Designs

In the physical structure shown in Fig. 1, only the nodes forming the wireless sensor network need to be designed and realized, including gateway node, relay node, and so on.

Hardware design of gateway node. The hardware design gateway node is as shown in Fig. 2, which is based on Arduino UNO [7], Ethernet Shield and nRF24L01 module. ATMega328 is the micro controller unit, with which the program can communicate with the internet and the wireless sensor network using serial peripheral interface.

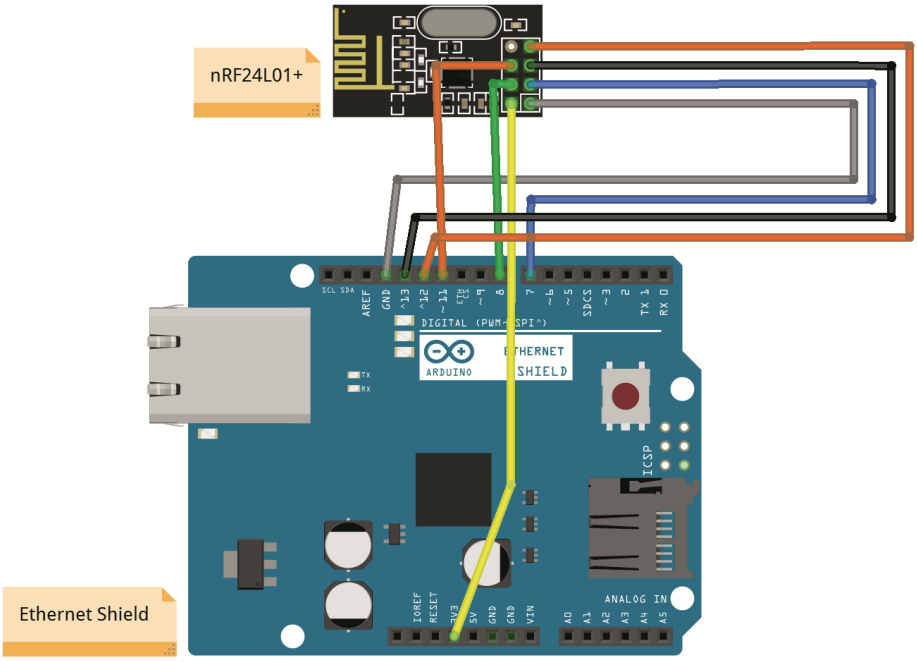


Fig. 2. The hardware reference design of home gateway node

Hardware design of relay node. As shown in Fig. 3, the hardware of relay node consists of an Arduino Pro Mini, an nRF24L01+, which is a suitable module for low cost scenario [8], an AMS 1117 and a relay module. Arduino Pro Mini reads the data from the gateway node using the wireless module, nRF24L01, and then react to the instructions in the data, like turning on the switch of a light or an electrical kettle.

As for other nodes, relay can be replaced by other modules according the node's functions, like temperature node having a temperature sensor LM35 in the place of replay.

The development of node is flexible and agile. If the node obeys the communication protocol in the system, the node can be developed with other embedded systems and join the distributed wireless sensor network.

Every single node contains an ATmega microcontroller unit, e.g. gateway node containing ATmega2560, for generally it handles larger data than other nodes'. Other nodes, like temperature monitor node, are using ATmega328P-AU, which is suitable for handling small amount of data. The data is transferred between nodes by the micro-controller unit communicating with nRF24L01P.

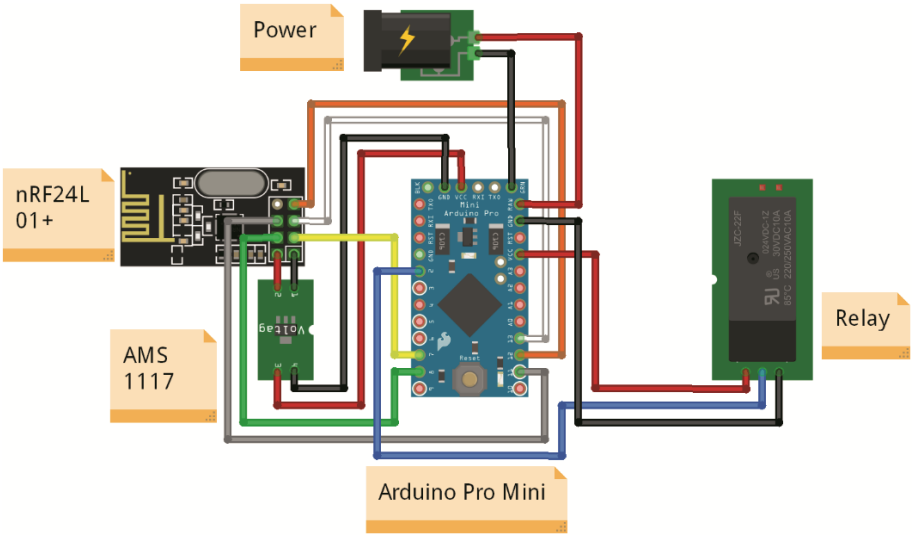


Fig. 3. The hardware reference design of relay sensor node

3.3 The Design of Ad-Hoc Wireless Sensor Network

In the proposed structure shown in Fig. 1, the nodes connecting to household appliances form an Ad-hoc wireless sensor network. As shown in Fig. 4, the nodes in this peer-to-peer network can receive data, transmit data, and net each other dynamically without a centralized node.

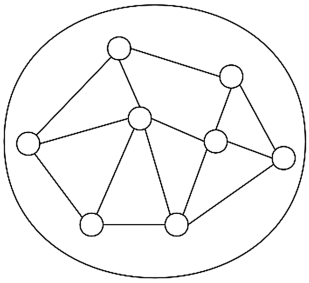


Fig. 4. The Ad-hoc Wireless Sensor network for smart house control system

Protocol of broadcast. According to flooding with self-pruning strategy, the protocol of broadcast is designed as shown in Fig. 5, for the maximum amount of data that nRF24L01P module can send at a time is 32 bytes.

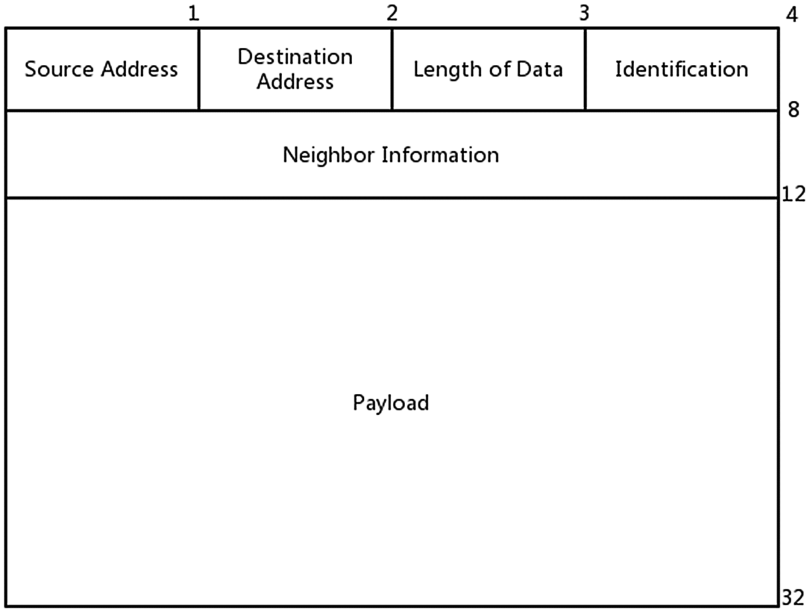


Fig. 5. Protocol of broadcast

In this protocol, the address 0x00 is set as empty, 0xFF is set as broadcast address, and the rest of address is available for nodes. The source address field records address of the initial node that sends the data. The length of the field is 1 byte, and the value varies from 0x01 to 0xFE. Destination address is the node which can receive and parse the data finally. The length of destination address is 1 byte, varying from 0x01 to 0xFE. The field, length of data, indicates the length of the payload field. The maximum value for this field is 1 byte, meaning the maximum byte of payload it can record is 255. In fact, the length of payload is limited by the maximum byte that can be sent by the wireless module. This protocol uses identification field to prevent duplicated data in a period. The length is 1 byte. Payload is the data that will be passed to upper application. The maximum length is 20 bytes, due to the limitation of nRF24L01P. If the length of data is less than 20 bytes, 0x00 will be filled into the empty bits.

Process of Ad-Hoc networks based on neighbor discovery. When a new node connects to the network, this new node will broadcast a package of which the payload is “hello”, the neighbor information is empty and the destination address is 0xFF. If nodes surrounding the new node will receive the “hello” package and response the new node with their own address, constructing a package of which the destination address is the source address of former package and sending the package in a peer-to-peer way. The new node will add the neighbors into its neighbor information table and maintain it periodically. The workflow is shown as in Fig. 6.

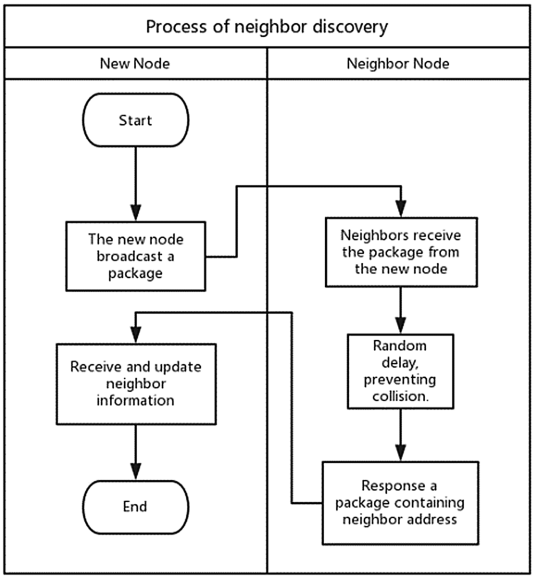


Fig. 6. Process of Ad-Hoc networks based on neighbor discovery

3.4 The Software Architecture

Software architecture. The software structure consists of three parts, which are user layer, cloud service layer and Ad-Hoc layer (Fig. 7).

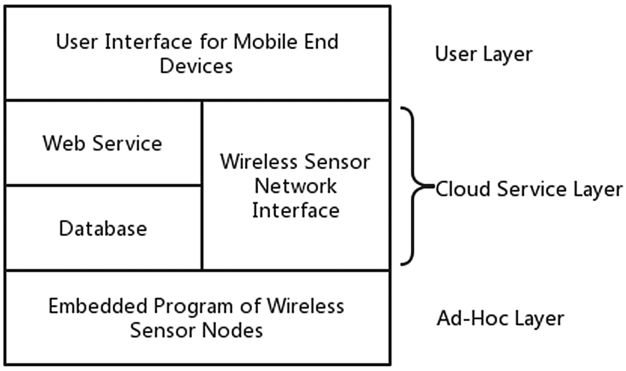


Fig. 7. Software structure

The bottom layer is embedded program of wireless sensor nodes, which is responsible for the control of electrical appliances and detecting data, like temperature, in users' house. The program calls serial peripheral interface to communicate with wireless modules, which can send data in the peer-to-peer network. Also, the procedure of

dynamic netting is handled in this layer. The networking protocol program is embedded in each sensor node.

The middle layer of the structure is cloud service layer, which is a bridge between users and the wireless sensor network. In this layer, the software runs in cloud so that users and gateway nodes can access corresponding services anytime and anywhere. The backend of web service deals with users' instructions and calls wireless sensor network interface. When wireless sensor network receives the instructions from web backend, it will encode and send the instructions to the gateway nodes of wireless sensor network.

The top layer is where the user interface lies. The web service provides user interface for mobile end devices, so that users can access the dashboard with browsers, not limited to their operating system. This part of the structure is called Browser/Server structure, which has been applied to a great multitude of applications.

4 Airome: A Demonstration System of the New Architecture

Based on the former proposed architecture, a smart home control system named Airome is realized as a demonstration. Airome is an application of Internet of Things. It is realized with LAMP in MVP [9] (Model-View-Presenter) pattern. LAMP (Linux, Apache, MySQL, PHP) is a popular web solution, providing service for millions of websites and various scenarios. How to realize Airome in cloud server is shown in Fig. 8. Gateway-worker is an open-source PHP framework, supporting distributed deployment and Transmission Control Protocol. So we choose it to realize Airome.

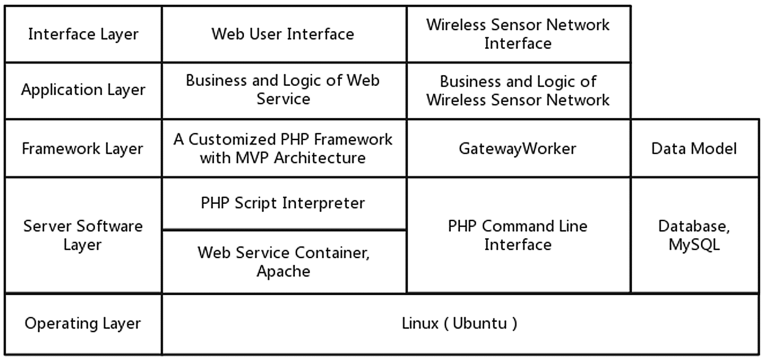


Fig. 8. Airome in cloud server

With Airome, users can access the web user interface to control their household appliances. In the meanwhile, the web service will connect to WSN (Wireless Sensor Network) server, which will deal with the commands from users and play the role of bridge connecting WSN and users.

The clients of Airome are browsers. The web interface of Airome is designed as single page application [10] for a better user experience, which means it can reduce the time that users waiting, display as a native application user interface(UI), and so on. The

UI of Airome is shown in Fig. 9. In the UI, the menu has three items, which are devices, functions and user. The appliances that are connected to the system will be displayed in the UI of "devices". Other functions like setting a scheduled task are put in the UI of "functions"; and user's information can be modified in the UI of "user".

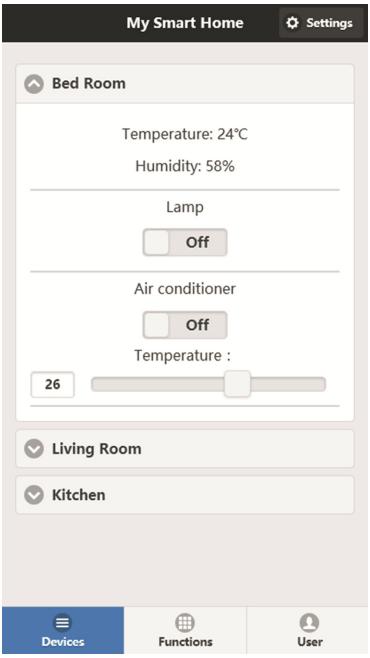


Fig. 9. User interface of Airome

The application of Airome shows that users can remotely control their household appliances anywhere anytime with their smart mobile phones. Users can also increase or decrease freely their household appliances to be controlled remotely.

5 Conclusion

We introduced a new architecture of smart house control system based on Ad-hoc WSN and cloud service in this paper. The physical structure, hardware design, Ad-hoc WSN designs and the reference software architecture have been shown and interpreted in detail. And an illustrated system called Airome based on our proposed architecture has been developed at last which had shown the effectiveness of the new architecture.

References

1. Tao, M., Ota, K., Dong, M.: Ontology-based data semantic management and application in IoT- and cloud-enabled smart homes. Futur. Gener. Comput. Syst. **76**, 528–539 (2017)
2. Ahmed, S.H., Kim, D.: Named data networking-based smart home. ICT Express **2**(3), 130–134 (2016)
3. Smirek, L., Zimmermann, G., Beigl, M.: Just a smart home or your smart home – a framework for personalized user interfaces based on eclipse smart home and universal remote console. Procedia Comput. Sci. **98**, 107–116 (2016)
4. Zualkernan, I.A., Al-Ali, A.R., Jabbar, M.A., et al.: InfoPods: Zigbee-based remote information monitoring devices for smart-homes. IEEE Trans. Consum. Electron. **55**(3), 1221–1226 (2009)
5. Wang, Z., Liu, Z., Shi, L.: The smart home controller based on zigbee. In: The 2nd International Conference on Mechanical and Electronics Engineering, vol. 2, pp. 300–302 (2010)
6. Soliman, M., Abiodun, T., Hamouda, T., Zhou, J., Lung, C.H.: Smart home: integrating internet of things with web services and cloud computing. In: IEEE Internet Conference on Cloud Computing Technology & Science, vol. 2, pp. 317–320 (2014)
7. Banzi, M., Shiloh, M.: Make: Getting Started with Arduino. Maker Media, Sebastopol (2014)
8. Khan, F.V., Teja, N.S., Parvez, A.L.A., et al.: Low-cost smart home design. Indian J. Sci. Technol. **8**(S2), 295–298 (2015)
9. Gu, M.X., Tang, K.: Comparative analysis of WebForms, MVC and MVP architecture. In: International Conference on Environmental Science & Information Application Technology, vol. 2, pp. 391–394 (2010)
10. Fink, G., Flatow, I.: Pro Single Page Application Development. Apress, New York (2014)

Big Data Analysis of TV Dramas Based on Machine Learning

Jiaqi Tan, Feiqiao Mao(✉), Lianghai Yang, and Jiahui Wang

Shenzhen University, Nanhai Ave 3688, Shenzhen 518060,
Guangdong, People's Republic of China
christinatan0704@gmail.com, feiqiao@szu.edu.com, i@silas.hk,
samwanglb@gmail.com

Abstract. Currently, large amount of TV dramas has overwhelmed the demand of TV station which had caused massive waste of resources. This article offers several practical solutions to tackle the above-mentioned problems through building model based on machine learning. Firstly, we build a TV score prediction model with regression and machine learning to rank the most welcomed TV drama. Moreover, we write an Internet worm to collect data from the internet, and build a Star popularity index prediction model by machine learning and regression. And list the much-acclaimed stars based on the popularity index. In conclusion, with the predict score of the TV drama predicted based on machine learning, it can provide a reference for TV station to manage TV programs and with the starring ranking it can help TV drama production team to produce TV dramas in a high quality.

Keywords: Machine learning · Big data · TV dramas · Predict score
Staring ranking

1 Introduction

Currently, with the fast pace development of multimedia, large quantities of TV dramas have overwhelmed the demand of TV station. And simultaneously, the oversupply of TV drama will not only put a potential threat to the development of file and television industry but also leads to massive waste of resources.

In order to lower down the risk of TV investment, improve scrip quality and forecast audience response to ensure maximum benefits. We have predicted the TV dramas' score and rank the starring by their popularity. Based on methods of machine learning (such as Classifier, Clustering, etc.) to extract the Eigen value from the data and use the Eigen value to divide the data into training data and testing data. And subsequently, build an optimized model with these data and use the optimized model to analyze relevant data in order to evaluate and customize file and television.

This study was supported by Guangdong Natural Science Foundation (2016A030313036), Shenzhen Science and Technology Foundation (JCYJ20150324140036842) and Guangdong Graduate Education Project (2015SQXX0).

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 90–95, 2018.
https://doi.org/10.1007/978-3-319-73830-7_9

2 Concepts and Related Work

Linear regression is popular in tackling some popularity prediction problem, for most of the research that had been done, it will predict data combing various method. According to [1] it made a movie recommendation considering the context sensitive and based on a time-decay model, and in research [2] it has predicted the region-specific crime rate for the future based on statistical auto regressive linear regression modeling.

There also have some related work like creating some raking model, such as a personalized ranking model [3] have make a personalized ranking based on pairwise learning. And in research [4] it has make a Social Media Content ranking based on social computing and user influence.

In this paper, considering that the factors to be use are specific date instead of fuzzy data, according to [5] we should better adapt the tolerance approach for possibilistic linear regression with fuzzy-valued inputs and/or outputs. But for the TV ranking problem in order to predict the score for TV drams, we can find out the most influential element of the model and produce an effective ranking model simply based on linear regression.

3 Score Prediction Model

This model aims for predict the score of each TV drama according to the drama's theme, production team, screen writer and starring. The modeling method is mainly based on linear regression and with this model we can predict the popularity rate of the drama.

3.1 Modeling Process

We build this TV dramas ranking model by these processes:

- **Data division**. Extract data we collected from the internet (which included the TV dramas' theme, ultimate score, director, screenwriter and starring). And then randomly divide these data into training data (at 80%) and testing data (at 20%). Training data is used to make a model to predict the score of all the TV dramas and testing data is used to calculate the standard error of the model and correct the model.
- **Modeling**. Create a model of TV ranking using regression and based on machine learning. Use the score from testing data and the predicted score to calculate the RRS (residual sum of squares) and errors of the model. Use machine learning algorithm to calculate the more accurate value of W (weight) and repeat 'Modeling' and 'Correct the model' step till getting the minimum RRS and error. Finally, we use this evaluation function to predict the score of all TV dramas:

$$\hat{y} = \hat{w}0 + \hat{w}1 \text{ director } + \hat{w}2 \text{ theme} + \hat{w}3 \text{ screenwriter } + \hat{w}4 \text{ starring} \quad (1)$$

$\hat{y}$ is the predicted score and $\hat{w}i(i >= 0,\ i \in N)$ is the weight of each features. (for example, $\hat{w}1$ is the weight of distribution enterprise.)

3.2 Proof Model Effectiveness

At first, we pick use all the possible influential factors as Eigen to build several models, after comparing the ranking result from our model to the real ranking result we find that the most influential factors are starring and director, thus using director, screenwriter, starring and drama's theme as Eigen value to build a model would be more accurate. As a result, in this model (using director, screenwriter, starring and drama's theme as Eigen value), the max error of this ranking model is 2.358532301405809 and the RRS of the model is 1.6066902 (Tables 1 and 2).

Table 1. The ranking of the latest TV dramas by predicted scores

TV drama	Predicted score	Actual score	Comment number	Theme
Battle of Changsha	9.199973382	9.2	18419	Modern revolution
All Quiet in Beiping	8.799892707	8.8	16084	Modern revolution
Operation Mekon	8.764290332	7.2	468	Modern cases
The Merchants of Qing Dynasty	8.499971847	8.5	1271	
Dating Hunter	8.29978057	8.3	5664	Modern city
Hey Daddy!	8.1997884	8.2	1138	Modern city
A Civic Yuppie in Countryside	8.100002467	8.1	3956	
My Heart is Shining	8.099951785	8.1	234	Other
Dangerous Journey	7.999915658	8	266	Modern cases
Simple World	7.899988534	7.9	5443	Modern countryside

It can be clearly seen from the form that the predicted score of "Operation Mekon" is higher than its original score to 1.5 points, but after searching on the internet, the score of "Operation Mekon" is high in reality (some of the viewer even score it to 9.3). Hence, in conclusion, using director, screenwriter, starring and drama's theme as Eigen value to build a ranking model is accurate and reliable.

Moreover, the spread of the predicted score of TV drama shows like this (see Fig. 1).

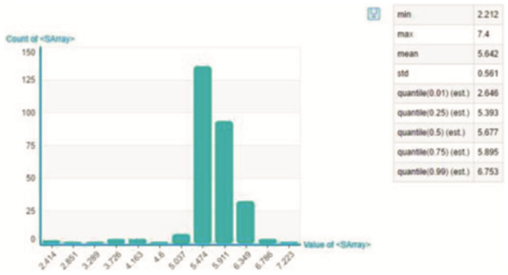


Fig. 1. The spread of all the TV dramas' scores.

Take the max score 7.4 (more accurate at 7.39978977266) as an example, the TV drama's name is "Soldier" compared to its score from "iQIYI.com" which scores 7.2, so the error of our model is less than 3%.

Lastly, we make a model to compare the predicted score with the original score (see Fig. 2), if the point is close to the middle line, means that it fits the original score well, it can be clearly seen from the graph above that most predict score is equal to its real score.

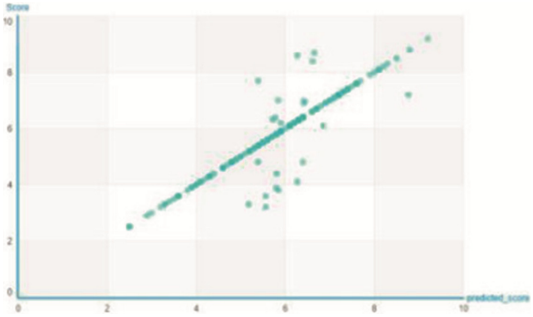


Fig. 2. Proof of consistency of predicted score and actual score

4 Star Popularity Ranking Model

The ranking model can be used to predicted the popularity of stars and rank the star popularity index.

4.1 Modeling Processes

First of all, we have written a web crawler to get data of 201 TV dramas in 2016 from iQIYI.com, which includes 351 stars and scores of relative TV drama. And subsequently, we use the same modeling method in task one (divide the data randomly in to training data (account for 80%) and testing data (account for 20%) using regression and machine learning to build a model, etc.) and the only difference is that we have changed 'star' to

target and changed 'score' to Eigen value to build a model. And finally use the model to predict the score of every star and get the star popularity index. Finally, we use this evaluation function to evaluate the star popularity index:

$$\hat{y} = \hat{w}0 + \hat{w}1 \text{ score} \tag{2}$$

$\hat{y}$ is the predicted score and $\hat{w}$ 1 is the weight of score.

After predicting the score of every star in our collected data, we can list the top ten popular TV drama star as follow:

Table 2. The ranking of star popularity index

Actor name	Count	Score
Zheng Shuang	4	9.897662804
Gan Tingting	1	9.199681232
GuLi Zhana	6	8.89894841
Zuo Jiani	1	8.700088684
Liu Wenxuan	1	8.700088684
Li Jiaxin	1	8.700088684
Han Rui	1	8.700088684
Wang Zixu	1	8.599943771
Luo Jin	2	8.599815697
Zhao Liying	2	8.540066765

4.2 Proof Model Effectiveness

After using testing data to evaluate the model, we find out the max error of the model is 1.7002398555285776 and the RRS of the model is 0.7214715971395239 which indicates that the model is conformed to the reality. Furthermore, we have compare the star popularity index that we designed to the Chinese Top Searches of Stars in the latest month (the list contains other type of stats like music stars, etc.) from iqiyi.com, ZhengShuang who has the highest popularity index in our predicted data, and she ranks the first among other stars we scored according to the list given in iqiyi.com.

5 Conclusions and Future Work

Based on the findings from TV Drama Score Prediction Model we can conclude that screenwriter, starring and drama's theme affect the score of a drama most. And using these factors to create an evaluation function (1) can predict the score of TV drama. Which can help TV station to set the timetable of TV dramas. Moreover, based on the star popularity ranking model we can find out the most popular star and produce a star popularity ranking list. These predicted data can help us to enhance the quality of TV drama and make a recommend on TV dramas to viewers.

References

1. Selva Priya, S., Gupta, L.: Predicting the future in time series using auto regressive linear regression modeling. In: 2015 Twelfth International (2015)
2. Luminto, D.: Weather analysis to predict rice cultivation time using multiple linear regression to escalate farmer's exchange rate. In: 2017 International Conference, Concepts, Theory, and Applications (ICAICTA) (2017)
3. Guo, W., Wu, S., Wang, L., Tan, T.: Personalized ranking with pairwise factorization machines. Neurocomputing **214**, 191–200 (2016)
4. Ntalianis, K., Salem, A.-B.M., El Emary, I.: Social media content ranking based on social computing and user influence. Procedia Comput. Sci. **65**, 148–157 (2015)
5. Černý, M., Hladík, M.: Possibilistic linear regression with fuzzy data: tolerance approach with prior information. Fuzzy Sets Syst. (2017)

Face Based Advertisement Recommendation with Deep Learning: A Case Study

Xiaozhe Yao, Yingying Chen, Rongjie Liao, and Shubin Cai(✉)

Shenzhen University, Shenzhen, China
xiaozhe.yaoi@gmail.com, sheeppooo.cy@gmail.com, jerryliao26@gmail.com, shubin@szu.edu.cn

Abstract. Recently, there is a massive growth of the offline advertising industries. To increase the performance of offline advertising, researchers bring out several methodologies.

However, the existing advertisement serving schemes are accustomed to focusing on traditional print media, resulting in the lack of personality and impression. Meanwhile, we find that facial features such as age, gender, can help us classify consumers intuitively and rapidly so that it can raise the accuracy in recommendation in a short time. Motivated by an original idea, we offer a Face Based Advertisement Recommendation System (FBARS). We propose that the FBARS works well in offline scenario and basically it could raise the accuracy 4 times. it performs 4 times better than the classic method using collaborative filtering.

Keywords: Computer vision · Recommendation system
Deep learning · Face recognition · Advertising

1 Introduction

Closed circuit television system has become closely, widely used and indispensably in our daily life decades ago, such as monitoring peoples' behavior in plaza, elevator , library and so on, which provides abundant faces for further analysis. With face recognition technology, monitor and television systems are not only able to collect face images but also can analyze much more implied information such as age, gender and emotion.

At the same time, with the growing of online commerce, recommendation has aroused the greatest concern. Since recommender systems could suggest recommend product to customers and help them make a purchase decision which can stimulate the potential consumption leading to great Businesses' income. greatly increase the Businesses' income.

Take the above points into account, a recommender system based on face related information is proposed in this paper. We shall first briefly address some related work in advertising recommendation and face recognition together with our system architecture. This will be followed by the implements of the adopted face recognition models and the advertising recommendation system. After that, there will be the evaluation of the whole system which is based on MovieLens dataset.

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 96–102, 2018.
https://doi.org/10.1007/978-3-319-73830-7_10

2 Related Work

2.1 Internet Advertising

Recent researchers on online advertisement recommending are gradually taking deep learning into consideration. With the explosion of data growth, far more data has been collected for the use of recommendation. For the online scenario, the data used by the recommender could be classified into two subset, which are user related data and item related data respectively. Both of them usually perform to be multi-dimensional data. Taking user data as an example, it usually includes browsing history, locations, age, gender and etc. while item data usually includes title, price, description and so on.

Classic recommender system usually adopts collaborative filtering as its basic algorithm. With the growing of computing capability, deep learning has greatly changed the traditional recommendation, such as Wide and Deep Learning for Recommendation [1]. Restricted Boltzmann machine [2] has been also used for collaborative filtering.

However, in offline scenarios, it is usually hard to collect user related data since the system cannot track customers' trace. That's the reason why there's still a lack of recommendation system in offline advertising.

2.2 Face Recognition

On the other hand, computer vision has been greatly enhanced by deep learning, especially object detection. There exists a variety of possible approaches such as support vector machine, convolutional neural networks and so on that could help us classify the age or gender of a face. Since Google's Inception V3 [3] model showed a pretty result in image classification task, it is adopted with fine-tuning as age and gender classifier.

3 System Architecture Overview

The FBARS architecture is showed as in Fig. 1 in time series. The system's camera will keep starting status and it detects faces in real time. After the face is captured, the recognition model will be called and the model will return the age and gender estimation. At the same time, a face ID which is used as a key parameter for identifying user in the recommendation system will be generated for history tracking.

In this system, the performance of recommender is evaluated by the appearing time of a user's face, that is to say, the more time a user spends on an advertisement, the better performance we think our recommender do. Meanwhile, this time cost could also become an essential parameter to determine whether this user has been attracted by this advertisement.

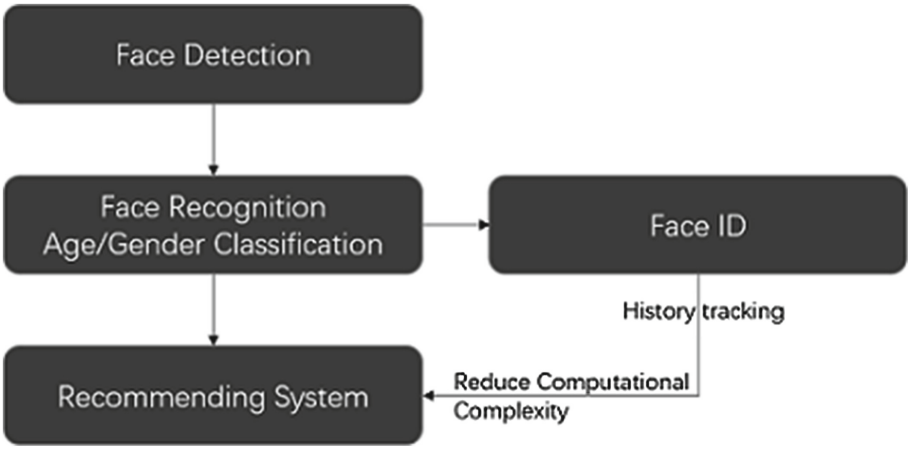


Fig. 1. FBARS architecture

4 Face Recognition for Feature Extraction

The role that face recognition module played is to extract the required feature such as age and gender. To achieve this goal, it consists of three independent parts, which are face detection module, gender classifier and age classifier.

Considering the offline scenario, the system are supposed to response as soon as possible when there stands a people in front of the camera since it might be too short for the customer to look at the advertising board. Therefore, the detection module are supposed to detect the human face at the speed of at least 24 fps.

YOLO [4], known as You Only Look Once, works really fast on object detection task. On a Titan X, the detection speed could reach 45 fps with 63.4.

Take all these into account, YOLO model is adopted in to the face detection module, which takes every frame of the camera stream and triggered recognition module as long as it detect human face.

Generally, age prediction usually point to a regression problem but technically, it is considered as a classification problem in order to be compatible with existed image classification models. More specifically, age range between 0 to 100 is separate into 8 columns, which are [0,2], [4,6], [8,12], [15,20], [25,32], [38,43], [43,60] and [60,100]. This boundary is designed for two reasons, one is that different group usually have different requirements and preferences for product while the difference among within group is far less. Like a male in [15, 20] may prefer to buy a pen instead of a shave while a 18 or 19 man may both prefer to buy a pen. And it's quite hard to identify the difference between a 18-year-old boy and a 19 one. On the hard hand, it is supposed to filter some age domains that need no advertising such as too young or too old so that it could reduce the computational complexity.

As for gender classification, it is much more simple since it is a classic two-dimensional classification problem. Therefore, we narrow both age classification task and gender classification task to general image classification problem. Inception V3, known as the third generation of Google's Inception model, works almost the best on ILSVRC 2012 in 2015 so it is adopted in the FBARS system.

In all, the face recognition module works as it is shown in Fig. 2 [6].

Fig. 2. Procedures of face recognition [6]

After the processing of face recognition, a face ID with the gender and age prediction will be added to database for further use as a three-dimensional array (Fig. 3 and Table 1).

Fig. 3. The face ID storage structure

Table 1. Performance of recommendation system

Adopted algorithm	RMSE
User-based collaborative filtering	3.1747
Item-based collaborative filtering	3.5141
Wide and deep model (200 steps)	0.6100

5 Face ID for History Tracking

The face ID module include two parts, which are face tracking and face information storage. They are some spotlight of the whole system and overcomes the shortcoming of classic offline advertising.

After the face recognition pass a face picture, age and gender information, the storage will save these data and generate a unique object id of this picture for further usage. Simultaneously, the face tracking module will generate a face-advertisement pair added by a time indicating how long the face appeared in the detection. As stated before, this time period will be used as the rank of how user will like this advertisement, that is to say the system is designed to provide users with such an advertisement which users might be interested in and will spend more time on it.

Face ID module is the so called middleware between face recognition and advertisement recommendation. It pass the triple array into recommendation stream right after the face recognition module finishes it's analysis.

6 Wide and Deep Learning for Advertisement Recommendation

As for the recommendation module, there're lots of methods to handle this problem. Followed by the wide and deep model from Google, a recommender based on deep learning is adopted in this system. The network model is show as in Fig. 4.

Except for face related information, there's still some more that could be used as recommender's parameter such as location, average income and etc. In order to make this network graph clean, they were ignored in this figure.

The collected data is separated into two different types, which are continuous features such as rank, average income and etc. The other one is categorical, which is much more in this system, such as age, gender and etc.

7 Performance Evaluation

An evaluation between the recommender system and classic user-based or item-based collaborative filtering has been conducted on MovieLens dataset. Although face images are not included in this dataset, they are collected through a widely used movie website, Douban, where there has lots of face images, age and gender of a movie start. After the collecting, the face images from website and user ID from MovieLens are paired according to the age and gender they have in common.

In other word, it is imitated that each user in movie lens has a paired face image from our collected data. By matching the user's unique id, age, gender and zip code are also selected as parameters since there's no way for the system to acquire users' occupation.

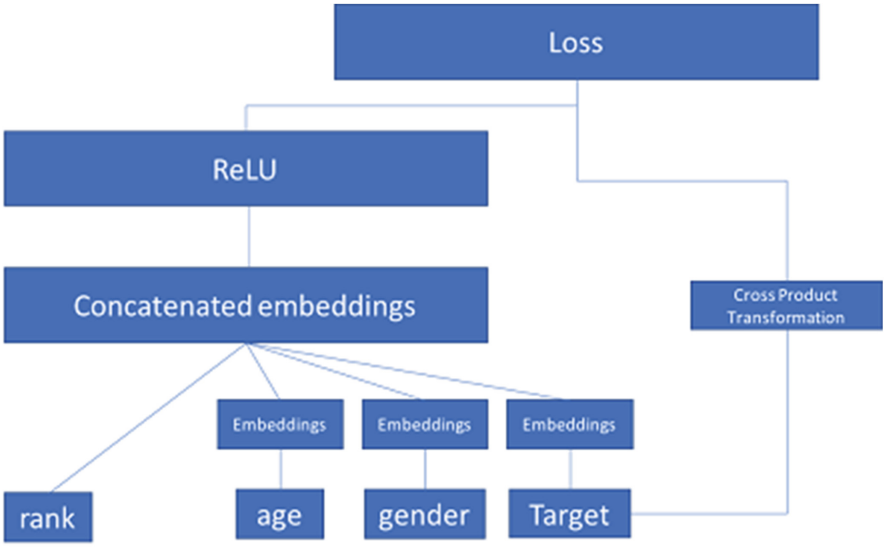


Fig. 4. Wide and deep learning network structure

After that, the original age, gender columns are hide and the system has to extract them from the image. Thus, the recommender's result are based on the original face image instead of the data in MovieLens dataset.

It showed that the wide and deep learning in 200 steps works 5 times better than classic collaborative filtering method, which is a great enhance on this problem.

8 Conclusion

An advertising system is proposed in this paper, although in experiments environment, it reached 5 times better than classic recommendation. The spotlight of the whole system is on the face ID and tracking system which enables the system to do continuous recommendation and history tracking.

However, in real life situations, it usually do much harder to detect faces when people glanced at the monitor. Therefore, if the system could connect with online advertising, it may do much better advertising effectiveness.

Acknowledgement. I would like to extend my sincere gratitude to Professor Shiqi Yu, for his instruction on face detection on this paper. I am deeply appreciate for his help.

References

1. Harper, F.M., Konstan, J.A.: The MovieLens datasets: history and context. ACM Trans. Interact. Intell. Syst. (TiiS) **5**, 19 (2016)
2. Cheng, H.T., Koc, L., Harmsen, J., et al.: Wide and deep learning for recommender systems. pp. 7–10 (2016)
3. Salakhutdinov, R., Mnih, A., Hinton, G.: Restricted boltzmann machines for collaborative filtering. In: International Conference on Machine Learning, pp. 791–798. ACM (2007)
4. Szegedy, C., Vanhoucke, V., Ioffe, S., et al.: Rethinking the inception architecture for computer vision. In: Computer Vision and Pattern Recognition, pp. 2818–2826. IEEE (2016)
5. Redmon, J., Divvala, S., Girshick, R., et al.: You only look once: unified, real-time object detection. pp. 779–788 (2015)
6. Eidinger, E., Enbar, R., Hassner, T.: Age and gender estimation of unfiltered faces. IEEE Trans. Inf. Forensics Secur. **9**(12), 2170–2179 (2014)

Ensemble Learning for Crowd Flows Prediction on Campus

Chuting Wu(✉), Tianshu Yin, Shuaijun Ge, and Ke Yu

School of Information and Communication Engineering,
Beijing University of Posts and Telecommunications, Beijing 100876, China
{wuchuting,sjge,yuke}@bupt.edu.cn, yintianshu1994@163.com

Abstract. Campus security is an increasing-attention problem in recent years. Crowd flows prediction on campus is helpful for people monitoring and can avoid potential risks. In this paper, based on distributed visiting data collection on campus, we propose a crowd flows prediction method with ensemble learning. For feature selection, we introduce more information than people visiting data, such as vocation and weather, and evaluate the feature importance as well as their combinations. For prediction model, we use stacking method with Random Forest, Gradient Boosting Tree and XGBoost for a better performance of prediction. Experimental results show that our method obtain high accuracy for crowd flows prediction with low extra cost.

Keywords: Crowd flows prediction · Ensemble learning · Stacking

1 Introduction

Nowadays more and more universities open their doors wider in order to invite more students to receive higher education. Not only the students from other universities can attend lectures and visit labs, but also teenagers can take part in academic activities on campus. It is a good policy for the young to broaden their insights. However, the administrators of universities and governments have to face more severe security risks. They tend to implement intelligent and networked monitoring system to manage and control the people, resources and environments on campus. The called Intelligent Campus comes into being [1]. The Intelligent Campus has been implemented in many universities. The data collected from multiple sources such as faculty/student e-card and campus WI-FI reflects the people's lifestyle. Big data and machine learning techniques make it possible to analyze and predict people's behavior on campus.

Crowd flows monitoring and prediction is an important part of the intelligent campus monitoring system. The visiting data of people in different places on campus is collected by mobile phones or sensors and analyzed in real-time. By analyzing the historical visiting data collected by the monitoring system of intelligent campus, we can learn the people's daily activity pattern and understand their behavior, which is helpful to guide them to move more smoothly,

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 103–113, 2018.
https://doi.org/10.1007/978-3-319-73830-7_11

safely and comfortably from one place to another. It is also helpful to prevent stampede accident or other disasters in campus.

In this paper, we attempt a new method based on the big data collected by mobile phones to analysis the change of the number of people on campus for preventing high-risk accidents. We propose a crowd flow prediction method with ensemble learning based on the historical visiting data of people. Ensemble learning [9,10], mainly including bagging, boosting and stacking schemes, is a powerful method by using multiple learning algorithms to obtain better predictive performance than constituent learning algorithms alone. There are mainly two challenges for the prediction task. The one is how to select the features about the people activities on campus. The other is how to combine multiple learning algorithms to obtain the optimal result.

Our work makes the following contributions:

(1) We introduce the crowd flows monitoring framework and distributed visiting data collection method;
(2) Based on collected visiting data, we analyze the statistical characteristics of crowd distribution;
(3) We propose crowd flows prediction method by using ensemble learning;
(4) We verify the performance of the proposed crowd flows prediction method by experiments and result analysis.

The paper is organized as follows: Sect. 2 introduces the related work. Section 3 describes the proposed crowd flows prediction method in detail. Section 4 presents the experiments and results. Section 5 concludes the paper.

2 Related Work

Nowadays with the rapid development of Smart City, Internet of Things (IoT) and big data technologies, crowd flows analysis gains more and more attentions. There are many application examples in cities and campus. For the cities, [2] analyzes the characteristics of mixed traffic flow with non-motorized vehicles and motorized vehicles at an unsignalized intersection. [3] forecasts citywide crowd flows based on big data by decomposing flows into seasonal, trend and residual flows components, which uses different mathematical models respectively. [4] proposes deep spatial-temporal residual networks for citywide crowd flows prediction. For the campus, [5] uses the records of people's consumption and WI-FI data to mine the spatial and temporal information of human activity and mobility and predicts the distribution of people on campus. [6] explores the relationship between the achievement of a student and his study partners based on the information collected from the digital campus card. Our paper emphasizes crowd flows on campus based on the visiting data other than citywide crowd flows. Since the people distribution on campus is much more related to time and space, we design a detailed and combined feature selection method for prediction. The high-dimensional class-imbalanced data may generate the cross effect and concentrate on the majority class which lead to bad results. Focusing on

these issues, [7] raises the methods of preprocessing data and [8] comes up with Ensemble Feature Selections before machine learning.

Ensemble learning [9,10] is a hot topic in the area of machine learning. It is a way by strategically generating and combining multiple models to solve a particular learning problem. There are several types of ensemble learning. Bagging (represented by Random Forest(RF) [11]), and boosting(represented by Gradient Boosting Tree(GBT) [12] and XGBoost [13,14]), are the common ensemble methods which have improved the accuracy of prediction a lot. Stacking [15] is also a common method to assemble machine learning models to enhance the ability of prediction. There are many applications of stacking. For example, [16] uses stacking method to classify imbalanced malware without unpacking process. And [17] uses stacking method for sentiment classification and verify that stacking is consistently effective over all domains, which works better than majority voting. Our paper considers to combine the traditional prediction methods, bagging method (RF) and boosting method (GBT, XGBoost), in order to explore a better stacking method for crowd flows prediction task.

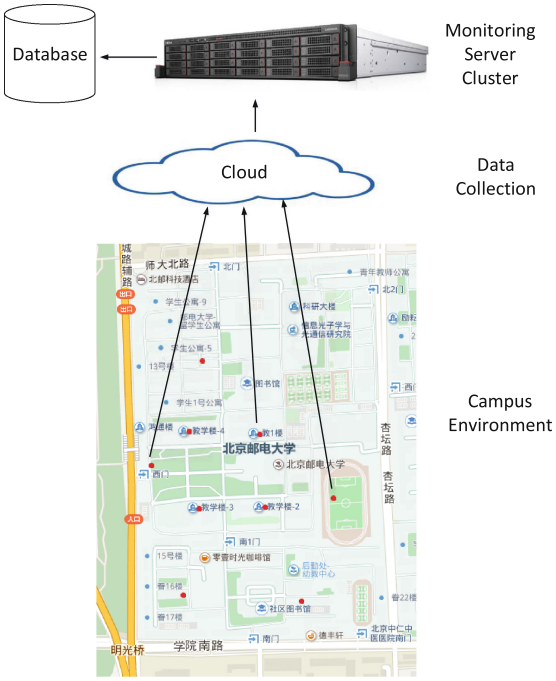


Fig. 1. Campus monitoring and prediction system

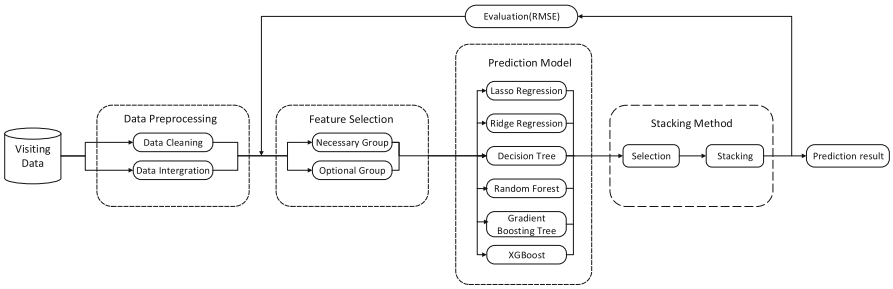


Fig. 2. Data analysis process

3 Crowd Flows Prediction Method

3.1 System Framework

Campus Monitoring and Prediction System aims at monitoring people activities and predicting the number of people at specific times and places on campus. The system framework is shown in Fig. 1. Data collection is based on distributed mobile devices and cloud. When a person arrives at one place such as classroom, library, cafeteria, and sport field, he/she will submit a request to server on Campus Network Center by mobile phone, Pad or laptop to apply Internet connection. The server's log is processed and then stored in the crowd database. Data analysis and visualization is also implemented on the server of Campus Network Center.

3.2 Algorithm Description

In this section, we present our main algorithm and Fig. 2 shows the whole process of data analysis. The description of the algorithm is given as follows:

(1) Do data cleaning and statistical analysis based on the raw visiting data in order to delete the noise data.
(2) Extract features from data and add some related features such as weather and holiday information.
(3) Divide all of the features into two groups, called necessary group and optional group respectively.
(4) Carry on crowd flows prediction by trying different features and models of machine learning.
(5) Stacking different combinations of them to obtain better results.

3.3 Data Collection and Preprocessing

3.3.1 Raw Data Collection

The raw data is collected through mobile devices of people which try to connect to the WI-FI on campus. The server of Campus Network Center receives the

request and records the detailed information about the visiting person, including time, device number, location and so on. The raw data mainly includes 3 fields as follows:

(1) person-id: each person's identifier, represented by a positive integer starting at 1.
(2) time-stamp: person's visiting time, represented by a number sequence with month, day, and hour.
(3) loc-id: id of the locations on campus, represented by a positive integer between 1 and 36.

3.3.2 Data Preprocessing

Data preprocessing includes data cleaning and data integration. The raw visiting data is transferred to structured data by deleting the incomplete and noise data. In the raw data, there are some special days in which the number of people increases or decreases sharply, for instance, the Anniversary of University and National Holiday, those points which deviate from the normal data points can't show the common rules of crowd flows and influence the accuracy of the prediction result, so we delete these noise data. It should be mentioned that although the number of people in August decreases sharply because of summer vacation, we still reserve these data points for a better result. Figure 3 shows the number of visiting people by month from July to October. It can be seen that the number of people changes with semester, vacation and special days. The visits in August are the least due to summer vacation, while the visits in October are the most due to university anniversary. These statistical analysis results should be considered in feature selection process.

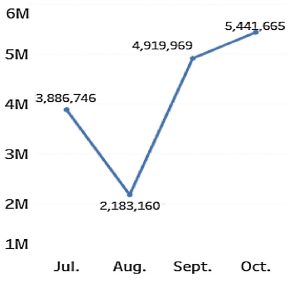


Fig. 3. Number of visiting people by month

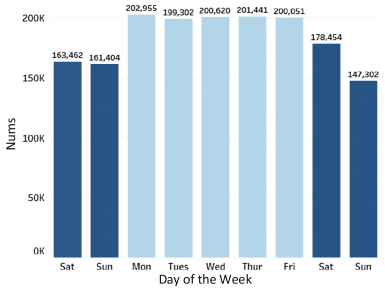


Fig. 4. Crowd distribution on weekdays and weekends

3.4 Feature Selection

3.4.1 Feature Description

The campus crowd flows prediction task is to predict the number of people in different places per hour. All of the features related to crowd should be extracted. Firstly, because we focus on the number of people on campus rather than the individual behavior, the number of people in each place per hour takes the place of person-id. Secondly, we split the time-stamp into 3 features (month, date, hour), for the reason that this way will provide more information. Thirdly, we add some new features including weather and holiday information. The crowd flows are highly related to weekdays and weekends, as shown in Fig. 4, so we add the information about Monday to Sunday. We also introduce the weather information from 2345 weather report [18] into our data, including temperature, air quality, wind level and so on. According to investigation in the campus and individual experience, we finally select 13 features and the detailed description is as Table 1.

Table 1. Feature description

Feature	Meaning	Description
month	The month of the data	Range 7 to 11
date	The date of the data	Range 1 to 31
hour	The hour of the data	Range 0 to 23
day_of_the_week	The day of the week of the date	Range 1 to 7
loc_id	The id of location	Range 1 to 36
weekend	The date is weekend or not	Weekend set 1, else 0
holiday	The date is holiday or not	Holiday set 1, else 0
air_quality	The air quality of the date	Range 1 to 6. Larger number, Worse air quality
highest_temp	The highest temperature of the date	Positive integer
lowest_temp	The lowest temperature of the date	Positive integer
wind	The wind level of the date	Range 0 to 3
PM2.5	The amount of the PM2.5 in the air of the date	Positive integer. Larger number, Higher PM2.5
rain_snow	The rain or snow level of the date	Positive integer. Larger number, Heavier rain/snow

3.4.2 Feature Ranking and Combination

Intuitively, more features can provide more information, which is helpful to increase the performance of the predict model. But some features which may interfere each other and the cross effect is likely to lead to a bad result.

And some features may not have much effect to improve the model, even damage it. For example, the date of the data is cycled and it is easy to cause misleading. So we use XGBoost to score each feature firstly, which provides the importance indication of each feature. After considering the scores, we divide all features into 2 groups, i.e. the necessary group and the optional group. Each learning iteration we select all features in necessary group and some of the features in optional group and combine them into a feature set.

3.5 Prediction Method

Model selection is another important factor to improve the performance of the prediction process. Regression analysis is a statistical process for estimating the relationships among variables. Before designing our prediction model, we test some traditional prediction models and compare their performances. The traditional prediction models include multiple linear regression, Decision Tree, and so on. Ensemble learning methods utilize multiple learning algorithms to obtain better predictive performance than any of the constituent learning algorithms alone. The ensemble learning mainly includes bagging, boosting and stacking. Random Forest is a bagging-based model by taking the majority vote in the case of single Decision Trees. XGBoost, as an upgrade version of Gradient Boosting Tree, is a very efficient scalable end-to-end tree boosting system which has been widely adopted for data analysis. Stacking is also a typical ensemble method that combines multiple machine learning models to construct a stronger one. Compared with bagging and boosting, stacking is more flexible because it combines different types of models or joints the predictive result to the features. Our proposed stacking method for crowd flows prediction is to utilize multiple predictive models' results as new features and put them into another model. Since the process of stacking needs more different kinds of models to provide diversity, we choose Lasso, Ridge, Decision Tree, Random Forest, Gradient Boosting Tree and XGBoost as optional models.

4 Experiments and Results

4.1 Experimental Dataset

The experimental dataset is provided by Campus Network Center, which contains the three fields as Sect. 3.2 from July to October, 2016. The experimental dataset has 22,600,642 items as total. Among these data, there are 16,431,540 items from July to October as train set and the others are as test set. The task of the experiment is to predict the number of people at 36 different locations per hour in November according the historical dataset. We use Root Mean Square Error(RMSE) between true and predicted values to evaluate our proposed method. The less RMSE, the better performance for the predictor.

$$X_{RMSE} = \sqrt[2]{\frac{\sum_{i=1}^{n}(x_{pred,i} - x_{true,i})^2}{n}} \tag{1}$$

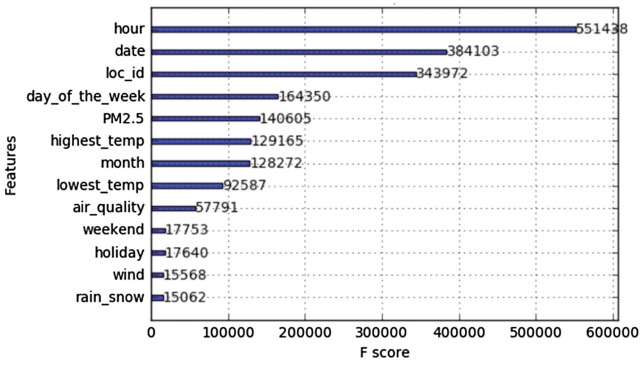


Fig. 5. Feature importance

4.2 Result Analysis

Firstly, the importance of all features calculated by XGBoost is shown in Fig. 5. We divide the features into necessary group and optional group according those computing and observation results. The necessary group including month, hour, loc_id, day_of_the_week, holiday, PM2.5, highest_temp and lowest_temp, and the optional group contains date, wind, air and weekend. It should be mentioned that from the statistic observation of raw data, we find that the crowd number change presents weekly period. However, if we reserve the two features of date and day_of_the_week, their periodic relationship leads to a negative influence to the prediction result. So we put the feature of date into optional group and calculate other features' importance again. And the feature of day_of_the_week performs well compared with the feature of weekend because the former provides more details than the latter. So different patterns of features cause different results and an appropriate pattern is important for the experiment.

Table 2 shows the RMSE of different prediction models under different feature inputs. For each model, we use 5-folds as cross-validation and Table 2 shows the best score of it. "All" means all of the necessary and optional features are used. "All-date" means all of the features are used other than date feature. From Table 2, lasso and ridge regression present worse performance and change little when using different features. Random Forest and Gradient Boosting Tree produce better RMSE. XGBoost leads to the best result and less time consuming.

According to the results, we can find that XGBoost presents it time-saving and accurate performance among the single prediction models. However, stacking is a more powerful way to take advantage of every single model and improve the prediction performance.

For stacking, we choose the best result of each prediction model to ensemble and use XGBoost as the prediction model at the second level. For each model we use 5-folds as cross-validation. The result is shown as Table 3. For each experiment, "$\surd$"means the model is used in stacking, "-" means unused. We can see from Table 3 that each stacking experiment obtains lower RMSE compared to single model in Table 2.

Table 2. Feature selection

Features	Models					
	Lasso	Ridge	DT	RF	GBT	XGBoost
All	272.66	276.80	99.88	82.36	84.00	81.14
All - date	272.66	276.65	93.94	75.61	75.42	74.31
All - date - air_quality	272.66	276.60	93.67	75.16	75.41	72.78
All - date - wind	272.66	276.43	93.94	75.15	76.52	74.24
All - date - rain_snow	272.66	276.48	91.98	75.95	76.08	75.24
All - date - wind - rain_snow	272.66	274.68	91.11	76.41	75.39	74.33

Note: Decision Tree (DT), Random Foreset (RF), Gradient Boosting Tree(GBT)

Table 3. Stacking result

Exp	Lasso	Ridge	DT	RF	GBT	XGBoost	Stacking
1	-	√	√	√	√	√	71.73
2	√	-	√	√	√	√	71.73
3	√	√	-	√	√	√	71.72
4	√	√	√	-	√	√	72.47
5	√	√	√	√	-	√	72.06
6	√	√	√	√	√	-	74.18
7	√	√	√	√	√	√	71.61

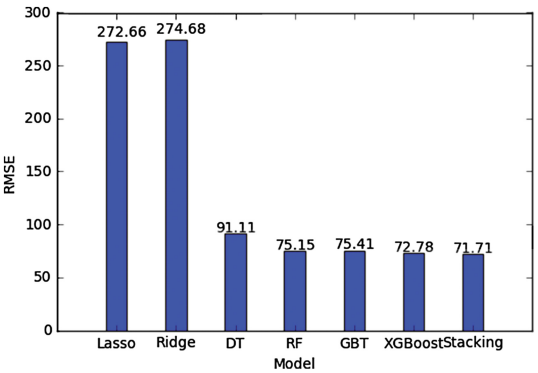


Fig. 6. Performance of different prediction models

Figure 6 shows the best results of different prediction methods, from which we can see that Stacking surpassed the others. The result verifies that the stacking method takes advantages of every prediction models and present a better performance.

5 Conclusion

In this paper, we introduce Campus Monitoring and Prediction System based on distributed people visiting data collection. We propose the feature selection and stacking-based ensemble learning for crowd flows prediction. The experimental results show that our proposed stacking method performs better for a lower RMSE. The next step is to improve the crowd flows prediction method and to implement the application system.

Acknowledgements. This work is supported by the National Natural Science Foundation of China under Grant No. 61601046 and No. 61171098, and is partially supported by the 111 Project of China under Grant No. B08004, and EU FP7 IRSES Mobile Cloud Project under Grant No. 612212.

References

1. Jackson, M.: Intelligent campus. In: International Symposium on Pervasive Computing and Applications, p. 3. IEEE (2006)
2. Xie, D.F., Gao, Z.Y., Zhao, X.M., et al.: Characteristics of mixed traffic flow with non-motorized vehicles and motorized vehicles at an unsignalized intersection. Phys. A Stat. Mech. Appl. **388**(10), 2041–2050 (2009)
3. Hoang, M.X., Zheng, Y., Singh, A.K.: FCCF: forecasting citywide crowd flows based on big data. In: The ACM Sigspatial International Conference, pp. 1–10. ACM (2016)
4. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction (2016)
5. Zhu, M., Pan, C., Yang, G., et al.: Prediction of population distribution on campus based on historical location data. In: Control and Decision Conference, pp. 2849–2854. IEEE (2016)
6. Fan, S., Li, P., Liu, T., et al.: Population behavior analysis of Chinese university students via digital campus cards. In: IEEE International Conference on Data Mining Workshop, pp. 72–77. IEEE (2016)
7. Yin, H., Gai, K.: An empirical study on preprocessing high-dimensional class-imbalanced data for classification. In: IEEE International Conference on High PERFORMANCE Computing and Communications. IEEE (2015)
8. Yin, H., Gai, K., Wang, Z.: A classification algorithm based on ensemble feature selections for imbalanced-class dataset. In: IEEE International Conference on Big Data Security on Cloud, pp. 245–249. IEEE (2016)
9. Chandra, A., Yao, X.: Ensemble learning using multi-objective evolutionary algorithms. J. Math. Model. Algorithms **5**(4), 417–445 (2006)
10. Webb, G.I., Zheng, Z.: Multistrategy Ensemble Learning: Reducing Error by Combining Ensemble Learning Techniques. IEEE Educational Activities Department (2004)
11. Breiman, L.: Random forest. Mach. Learn. **45**, 5–32 (2001)
12. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. **29**(5), 1189–1232 (2001)
13. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. pp. 785–794 (2016)

14. Chen, T., He, T., Benesty, M., et al.: XGBoost: extreme gradient boosting (2017)
15. Kaggle Ensemble Guide: http://mlwave.com/kaggle-ensembling-guide
16. Zhang, Y., Huang, Q., Ma, X., et al.: Using multi-features and ensemble learning method for imbalanced malware classification. In: Trustcom/BigDataSE/ISPA, pp. 965–973. IEEE (2017)
17. Su, Y., Zhang, Y., Ji, D., Wang, Y., Wu, H.: Ensemble learning for sentiment classification. In: Ji, D., Xiao, G. (eds.) CLSW 2012. LNCS (LNAI), vol. 7717, pp. 84–93. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36337-5_10
18. 2345 Weather Report: http://tianqi.2345.com

Impact of Probability Distribution Selection on RVFL Performance

Weipeng Cao[1], Jinzhu Gao[2], Zhong Ming[1(✉)], Shubin Cai[1], and Hua Zheng[1]

[1] Shenzhen University, Shenzhen 518060, China
mingz@szu.edu.cn
[2] University of the Pacific, Stockton, CA 95211, USA

Abstract. The initialization of input weights and hidden biases plays an important role in random vector functional link networks (RVFL). Although some optimization algorithms for initialization have been proposed in recent years, the initialization strategies of these algorithms are under the premise of the *uniform* distribution. In this paper, ten benchmark datasets are used to study the impact of different probability distributions (e.g., *Uniform*, *Gaussian*, and *Gamma* distributions) initialization on the performance of RVFL. The experimental results present some interesting observations and valuable instructions: (1) No matter whether we use *Uniform*, *Gaussian*, or *Gamma* distributions, RVFL initialized by the distribution with smaller variances always get lower training and testing RMSE; (2) Compared with the *Uniform* distribution, the *Gaussian* and *Gamma* distributions with smaller variances usually give the RVFL model better performance; (3) Regardless of the distribution, RVFL with the direct link from the input layer to the output layer has better performance than those without the link; (4) RVFL initialized by the distribution with larger variances generally needs more hidden nodes to achieve equivalent accuracy with ones having the smaller variances; (5) With the increase of distribution variances, the performance of RVFL decreases first and then remains stable.

Keywords: Random vector functional link networks
Neural networks with random weights · Random initialization
Probability distribution · Parameter selection

1 Introduction

In recent years, artificial neural networks (ANNs) have been widely applied to many applications due to its powerful learning ability [1–8]. Traditional ANNs, including deep learning, are trained by iteratively tuning all the parameters based on error back propagation. This training mechanism suffers from some notorious shortcomings such as slow convergence, local minima, time consuming, and model uncertainty. To alleviate these issues, several non-iterative neural networks with random weights (NNRWs) were proposed, such as Random vector functional link networks (RVFL) [9], Schmidt's method [10], Extreme learning machine (ELM) [11], etc. The input weights and hidden biases in NNRWs are selected randomly from a given range and

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 114–124, 2018.
https://doi.org/10.1007/978-3-319-73830-7_12

kept fixed throughout the training process while the output weights are obtained analytically. This non-iterative training mechanism allows NNRWs to achieve much faster learning speed than traditional ANNs. NNRWs are easy to implement and have been widely used in many applications. Some notable applications include embedded system [1], cluster problems [2], high-dimensional data processing [3], fuzzy nonlinear regression [7], electricity load demand forecasting [8], etc. The universal approximation capability of NNRWs has been proven in theory [12].

Since the input weights and hidden biases in NNRWs are generated randomly from a fixed scope, the model may be unstable. Many algorithms have been proposed to guide the hidden parameters selection in recent years [13–19]. In [13–15], comprehensive experiments were conducted on RVFL and some advices are given to guide the parameter selection of RVFL, such as the selection of activation function, the random range of hidden parameters, and the computation of the output weights. Several optimization algorithms were proposed in [16–19] to optimize the parameter selection of ELM. A common feature of the algorithms mentioned above is that the input weights and hidden biases are determined by assigning random numbers within an empirical range under *Uniform* distribution. Tao et al. [16] have shown that different probability distributions initialization for ELM has different influence on the ELM performance.

Although the core ideas of RVFL, Schmidt's method, and ELM are similar, each of them has its unique features [5, 6]. RVFL has the direct link between the input layer and the output layer (referred to as "the link"), while Schmidt's method and ELM do not have. In addition, the training mechanism, the types of hidden nodes, and the computation of the output weights of these algorithms are slightly different. As a result, the application scenarios of these algorithms are different, which means the parameter selection criteria applied to one algorithm may not be feasible to a different algorithm. In other words, the selection criteria of probability distribution for ELM cannot be directly applied to RVFL. To our best knowledge, there is no thorough study on the relationship between the hidden parameters' probability distribution and the performance of RVFL. Inspired by Tao's work, we conducted comprehensive experiments to study the impact of probability distribution selection on RVFL performance.

In this paper, *Uniform, Gamma,* and *Normal* distributions are used to initialize the input weights and hidden biases of RVFL. For each probability distribution, we set three different parameter pairs to control the range of random numbers. The experiments were conducted on ten UCI regression datasets [20] and the results showed that the hidden parameters' probability distribution has a significant influence on the performance of RVFL. It is noted that the influence of a probability distribution on RVFL is slightly different from ELM. In the rest of this paper, we share some interesting findings that provide a useful guidance for the parameter selection of RVFL.

The organization of this paper is as follows: Sect. 2 briefly introduces the RVFL algorithm. The details of the experiment settings and simulation results are described in Sect. 3. Section 4 provides the conclusions and our future works.

2 Review of Random Vector Functional Link Networks (RVFL)

Random vector functional link networks (RVFL), a special single hidden layer feed-forward neural network (SLFN), was proposed by Pao's research group in the 1990s [9]. The input weights and hidden biases of RVFL are selected randomly and kept fixed throughout the training process while the output weights are determined analytically. This type of training mechanism allows RVFL to have much faster learning speed than traditional SLFNs. A typical structure of RVFL is shown in Fig. 1. Remarkably, there is a direct link between the input layer and output layer. This is one of differences between RVFL and other NNRWs algorithms such as ELM [6].

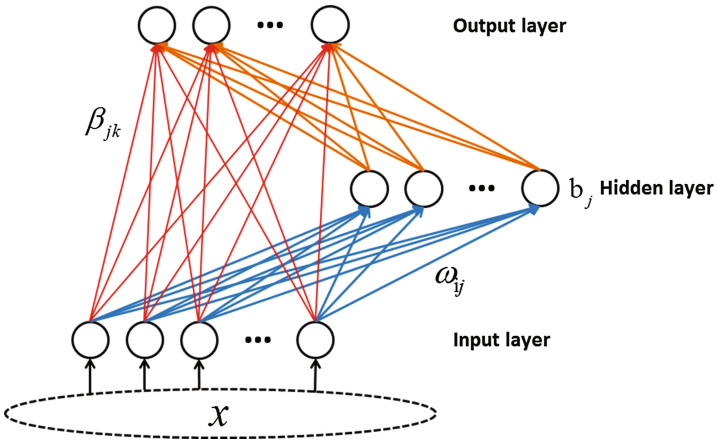


Fig. 1. The structure of RVFL

where ω denotes the input weights, β denotes the output weights including the weights between the hidden layer and output layer and the weights between the input layer and output layer, and b denotes the thresholds of hidden nodes. There is an activation function for nonlinear transformation in the hidden layer, which is usually denoted by $G(\omega \cdot x + b)$.

As discussed in [9], RVFL can be modeled as

$$D\beta = T, \tag{1}$$

where D is the matrix version of the concatenation of the original features and the random features mapping in the hidden layer, and T is the matrix version of the observation values of all data samples. The output weight β can be calculated by

$$\beta = D^{+}T \tag{2}$$

where D^{+} is the Moore–Penrose generalized inverse of D.

In most cases, the input weights ω and hidden biases b are usually determined by assigning random numbers within an empirical range (i.e., [–1, 1]) under *Uniform* distribution. [13, 14] pointed out that this randomization range cannot guarantee better performance for RVFL models, but they did not study the influence of the hidden parameters' probability distribution on RVFL. In this paper, we used three different probability distributions to initialize the input weights and hidden biases.

3 Experiment Settings and Results

In this section, we give the details of our experimental settings and results. All the simulations run in the MATLAB R2014a environment on the same Windows 10 machine with Intel Core i5-5300U 2.3 GHz CPU and 8 GB RAM. For each experiment, the average results over 50 trials are collected for RVFL. The indicators for RVFL performance testing include the root-mean-square error of training and testing (referred to as TrainRMSE and TestRMSE), standard deviation of training and testing accuracy (referred to as TrainDev and TestDev), and the convergence rate.

3.1 Data Preparation and Experimental Setup

In our experiments, ten regression datasets are used to test the performance of RVFL models. The specification of these ten datasets is summarized in Table 1. The *Sigmoid* function, $g(x) = \frac{1}{1+e^{-x}}, x \in (-\infty, +\infty)$, is chosen as the activation function of RVFL. Three mostly-used continuous probability distributions, that is, *Uniform, Gamma,* and *Normal* distributions, are used to initialize the input weights and hidden biases of RVFL. For each probability distribution, we set three different parameter pairs to study the effect of distribution variance on RVFL models.

Table 1. The details of 10 UCI datasets

Name	Training data	Testing data	Attributes
Airfoil self-noise	750	753	5
Combined cycle power plant	4500	5068	4
Concrete compressive strength	500	530	8
Energy efficiency	400	368	8
Housing	250	256	13
Red Wine quality	2000	2898	11
White Wine quality	2000	2898	11
Auto MPG	200	192	8
Parkinsons telemonitoring_motor	4000	1875	26
Parkinsons telemonitoring_total	4000	1875	26

3.2 Experimental Results and Analysis

Supported by the experiments described above, we conducted a comprehensive study to answer the following questions:

(1) Under the same network architecture, what is the effect of hidden parameters initialized with different probability distributions on the accuracy of RVFL model?
(2) What is the effect of probability distributions variance on the convergence rate of RVFL model?

To study the problem (1), we set the number of hidden nodes to 100. The experimental results are shown in Tables 2, 3, 4, 5, 6 and 7 and Fig. 2. It is noted that the extremely small values are marked as **N** and the best results are in boldface.

Table 2. Testing RMSE of RVFL with *Uniform* distribution initialization

Dataset	a = –1, b = 1		a = –10, b = 10		a = –20, b = 20	
	TestRMSE	TestDev	TestRMSE	TestDev	TestRMSE	TestDev
Airfoil	**1.21E–01**	9.92E–04	1.39E–01	**1.94E–16**	1.39E–01	2.41E–04
Combined	**2.73E–02**	**1.73E–17**	2.96E–02	6.14E–06	2.96E–02	5.97E–06
Concrete	**2.49E–02**	2.11E–04	2.74E–02	**N**	2.74E–02	2.77E–04
Energy	**2.07E–02**	**1.04E–17**	2.41E–02	2.78E–17	2.41E–02	1.04E–17
Housing	**3.69E–02**	1.39E–04	3.72E–02	**2.08E–17**	3.72E–02	1.88E–04
RedWine	**6.84E–02**	**4.16E–17**	6.94E–02	1.46E–04	6.94E–02	1.46E–04
WhiteWine	**1.94E–02**	6.94E–18	2.36E–02	**3.47E–18**	2.37E–02	3.12E–17
autoMPG	**9.63E–03**	**5.20E–18**	9.65E–03	8.43E–06	9.65E–03	8.47E–06
parkinsons_motor	**3.73E–01**	7.54E–04	3.75E–01	**1.67E–16**	3.75E–01	4.44E–16
parkinsons_total	**2.83E–01**	2.22E–16	2.84E–01	**N**	2.84E–01	5.55E–17

Table 3. Testing RMSE of RVFL with *Gaussian* distribution initialization

Dataset	mu = 0, sigma = 0.05		mu = 0, sigma = 0.2		mu = 0, sigma = 0.35	
	TestRMSE	TestDev	TestRMSE	TestDev	TestRMSE	TestDev
Airfoil	**1.07E–01**	4.19E–04	1.08E–01	**1.39E–17**	1.15E–01	8.33E–17
Combined	**2.39E–02**	6.18E–05	2.49E–02	**6.94E–18**	2.62E–02	3.47E–17
Concrete	**2.25E–02**	1.39E–17	2.30E–02	3.47E–18	2.35E–02	**N**
Energy	**1.72E–02**	**1.04E–17**	1.77E–02	1.04E–17	1.80E–02	1.39E–17
Housing	**3.29E–02**	**2.08E–17**	3.86E–02	4.16E–17	3.91E–02	2.78E–17
RedWine	**6.27E–02**	**2.78E–17**	6.81E–02	8.33E–17	6.92E–02	5.55E–17
WhiteWine	1.72E–02	1.73E–17	**1.63E–02**	**N**	1.63E–02	1.39E–17
autoMPG	**4.29E–03**	**2.60E–18**	8.49E–03	6.94E–18	9.19E–03	8.67E–18
parkinsons_motor	**3.61E–01**	3.89E–16	3.70E–01	**5.55E–17**	3.73E–01	5.55E–17
parkinsons_total	**2.64E–01**	2.78E–16	2.72E–01	**5.55E–17**	2.74E–01	9.14E–04

Table 4. Testing RMSE of RVFL with *Gamma* distribution initialization

Dataset	k = 9, b = 0.1		k = 9, b = 0.5		k = 9, b = 0.9	
	TestRMSE	TestDev	TestRMSE	TestDev	TestRMSE	TestDev
Airfoil	**9.63E–02**	4.83E–04	9.72E–02	**1.25E–16**	9.70E–02	5.71E–04
Combined	**2.53E–02**	**1.04E–17**	2.92E–02	4.34E–05	2.95E–02	4.31E–05
Concrete	**2.49E–02**	2.27E–04	2.58E–02	**2.08E–17**	2.58E–02	1.53E–04
Energy	**1.86E–02**	**1.39E–17**	1.89E–02	2.08E–17	1.89E–02	2.08E–17
Housing	**3.49E–02**	3.15E–04	3.58E–02	**1.39E–17**	3.60E–02	3.93E–04
RedWine	**6.84E–02**	**6.94E–17**	7.28E–02	1.28E–05	7.33E–02	9.50E–06
WhiteWine	**1.72E–02**	**3.47E–18**	2.06E–02	2.78E–17	2.10E–02	2.43E–17
autoMPG	**8.17E–03**	**N**	8.73E–03	3.51E–05	8.78E–03	3.47E–05
parkinsons_motor	**3.61E–01**	**1.67E–16**	3.68E–01	5.00E–16	3.68E–01	1.44E–04
parkinsons_total	**2.65E–01**	**N**	2.72E–01	**N**	2.73E–01	1.11E–16

From Tables 2, 3 and 4, we can infer that

(1) No matter what the hidden parameters initialized by *Uniform*, *Gaussian,* or *Gamma* distributions are, the testing RMSE increase with the increase of random range. In fact, the training RMSE also conforms to this rule.

One explanation is that selecting the input weights and hidden biases from a bigger random range cannot guarantee the model conforms to the principle of structural risk minimization, and thus the model has suboptimal generalization performance.

We further compared the performance of RVFL under these three probability distributions with smaller variances and the results are shown in Tables 5 and 6.

Table 5. Training RMSE of RVFL models with *Uniform, Gaussian,* and *Gamma* distributions initialization

Dataset	Uniform: a = −1, b = 1		Gaussian: mu = 0, sigma = 0.05		Gamma: k = 9, b = 0.1	
	TrainRMSE	TrainDev	TrainRMSE	TrainDev	TrainRMSE	TrainDev
Airfoil	1.23E–01	4.59E–04	9.88E–02	4.44E–04	1.04E–01	4.99E–04
Combined	2.66E–02	1.39E–17	2.33E–02	6.93E–05	2.47E–02	6.94E–18
Concrete	2.44E–02	1.49E–04	2.13E–02	6.94E–18	2.44E–02	6.73E–05
Energy	1.95E–02	1.04E–17	1.68E–02	1.39E–17	1.74E–02	N
Housing	3.68E–02	1.95E–04	2.80E–02	2.78E–17	3.76E–02	2.64E–04
RedWine	6.59E–02	5.55E–17	5.90E–02	1.39E–17	6.98E–02	1.39E–17
WhiteWine	1.84E–02	N	1.47E–02	1.21E–17	1.60E–02	1.04E–17
autoMPG	9.31E–03	3.47E–18	4.42E–03	1.73E–18	8.31E–03	1.73E–18
parkinsons_motor	3.75E–01	3.35E–04	3.60E–01	1.67E–16	3.57E–01	1.67E–16
parkinsons_total	2.83E–01	1.67E–16	2.73E–01	2.78E–16	2.62E–01	5.55E–17

Table 6. Testing RMSE of RVFL models with *Uniform, Gaussian,* and *Gamma* distributions initialization

Dataset	Uniform: a = –1, b = 1		Gaussian: mu = 0, sigma = 0.05		Gamma: k = 9, b = 0.1	
	TestRMSE	TestDev	TestRMSE	TestDev	TestRMSE	TestDev
Airfoil	1.21E–01	9.92E–04	1.07E–01	4.19E–04	9.63E–02	4.83E–04
Combined	2.73E–02	1.73E–17	2.39E–02	6.18E–05	2.53E–02	1.04E–17
Concrete	2.49E–02	2.11E–04	2.25E–02	1.39E–17	2.49E–02	2.27E–04
Energy	2.07E–02	1.04E–17	1.72E–02	1.04E–17	1.86E–02	1.39E–17
Housing	3.69E–02	1.39E–04	3.29E–02	2.08E–17	3.49E–02	3.15E–04
RedWine	6.84E–02	4.16E–17	6.27E–02	2.78E–17	6.84E–02	6.94E–17
WhiteWine	1.94E–02	6.94E–18	1.72E–02	1.73E–17	1.72E–02	3.47E–18
autoMPG	9.63E–03	5.20E–18	4.29E–03	2.60E–18	8.17E–03	N
parkinsons_motor	3.73E–01	7.54E–04	3.61E–01	3.89E–16	3.61E–01	1.67E–16
parkinsons_total	2.83E–01	2.22E–16	2.64E–01	2.78E–16	2.65E–01	N

From Tables 5 and 6, we can find the following experimental observations.

(2) *Gaussian* and *Gamma* distributions with the smaller variances initialization usually make RVFL get the lower training RMSE and testing RMSE than *Uniform* distribution.

In addition, we studied the effect of the direct link between the input layer and output layer on RVFL under the same probability distribution. The experimental results are shown in Table 7.

Table 7. Comparison of testing RMSE between RVFL with the link and without the link (under *Uniform* distribution initialization)

Dataset	a = –1, b = 1 (with the link)		a = –1, b = 1 (without the link)	
	TestRMSE	TestDev	TestRMSE	TestDev
Airfoil	**1.21E–01**	9.92E–04	1.39E–01	8.30E–04
Combined	**2.73E–02**	1.73E–17	2.96E–02	3.12E–17
Concrete	**2.49E–02**	2.11E–04	2.74E–02	2.77E–04
Energy	**2.07E–02**	1.04E–17	2.42E–02	3.47E–18
Housing	**3.69E–02**	1.39E–04	3.72E–02	1.89E–04
RedWine	**6.84E–02**	4.16E–17	6.94E–02	4.16E–17
WhiteWine	**1.94E–02**	6.94E–18	2.36E–02	1.39E–17
autoMPG	**9.63E–03**	5.20E–18	9.66E–03	6.94E–18
parkinsons_motor	**3.73E–01**	7.54E–04	3.75E–01	8.46E–04
parkinsons_total	**2.83E–01**	2.22E–16	2.84E–01	1.67E–16

From Table 7, we can find the following experimental observations and deduce the following conclusions.

(3) Initialized under the *Uniform* distribution, the RVFL models with the link have smaller testing RMSE than those without the link. The same phenomenon also occurs when using *Gaussian* and *Gamma* distributions. Therefore, we can conclude that the direct link from the input layer to the output layer plays an important role in RVFL. RVFL with the link always achieves better performance than the ones without the link. One explanation is that the link can server as a regularization for the randomization and guarantee that the model can get a better solution from the solution space. [13] makes a similar conclusion.

To study the effect of probability distributions variances on the convergence rate of RVFL model, we increase the number of hidden nodes from 2 to 100 and plot the learning curves of RVFL with different probability distribution variances. The experimental results of AutoMap are shown in Fig. 2.

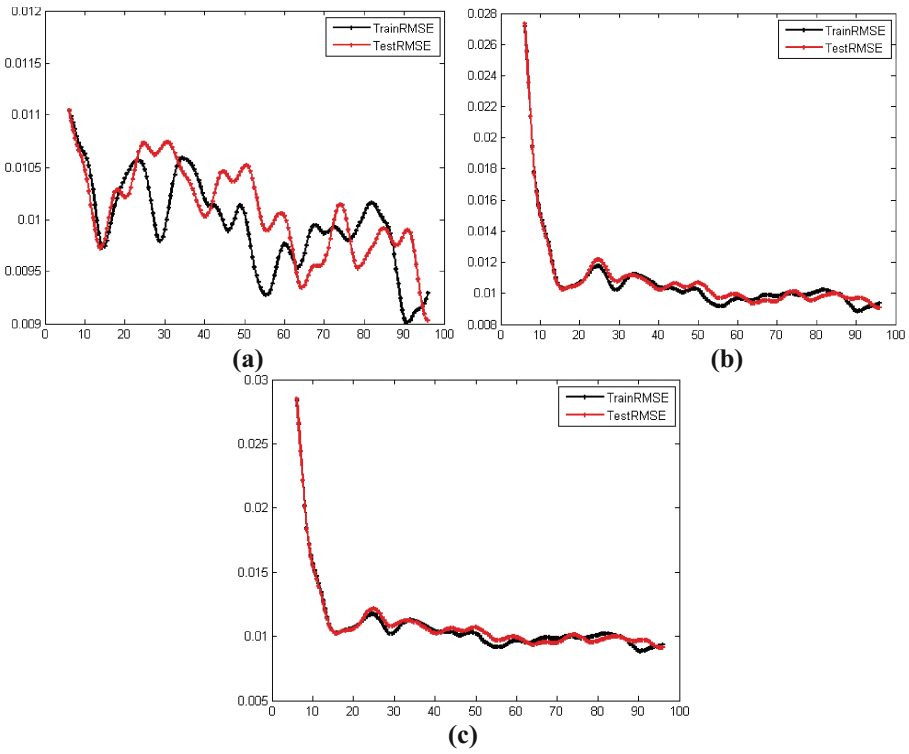


Fig. 2. (a) The learning curves of RVFL with *Uniform* distribution (a = –1, b = 1); (b) the learning curves of RVFL with *Uniform* distribution (a = –10, b = 10); (c) the learning curves of RVFL with *Uniform* distribution (a = –20, b = 20)

From Fig. 2, we have the following observations and conclusions:

(4) The *Uniform* distribution with smaller variances allows RVFL to get lower training and testing RMSE. This phenomenon can also be verified by Conclusion (1).

(5) RVFL initialized by *Uniform* distribution with larger variances generally needs more hidden nodes to achieve equivalent accuracy when compared with those having smaller variances.

Therefore, we can conclude that using *Uniform* distribution with smaller variances to initialize the hidden parameters could help RVFL achieve faster convergence rate and higher accuracy.

In addition, we find that:

(6) When the distribution variance increases, the performance of RVFL decreases first and then remains stable. As shown in Fig. 2(b) and (c), the accuracy of RVFL model does not change significantly when the random range changes from (–10, 10) to (–20, 20).

The same observations can be found for both *Gaussian* and *Gamma* distributions as well as in other datasets.

4 Conclusions and Future Works

In this paper, we conducted comprehensive experiments to validate the impact of probability distribution selection on the performance of RVFL. Based on the experimental results, we have the following interesting findings.

(1) No matter whether we use the *Uniform*, *Gaussian*, or *Gamma* distributions, the hidden parameters initialized by smaller variances allows RVFL to get lower training RMSE and testing RMSE.

(2) Compared with the *Uniform* distribution, the *Gaussian* and *Gamma* distributions with smaller variances usually improve the performance of RVFL.

(3) Regardless of the distribution, RVFL with the direct link between the input layer and the output layer can achieve better performance than ones without the link.

(4) RVFL initialized by the distribution with larger variances generally needs more hidden nodes to achieve equivalent accuracy with the ones with the smaller variances.

(5) With the increase of distribution variances, the performance of RVFL decreases first and then remains stable.

The above conclusions may provide useful guidance for RVFL initialization. In the future, we will give a complete theoretical proof for these phenomena and apply the robust RVFL to solve some complex classification problems such as class-imbalanced problems [21, 22].

Acknowledgments. This research is supported by the National Natural Science Foundation of China under Grant nos. 61672358 and the key Project of DEGP nos. 2014GKCG031.

References

1. Azad, N.L., Mozaffari, A., Fathi, A.: An optimal learning-based controller derived from Hamiltonian function combined with a cellular searching strategy for automotive coldstart emissions. Int. J. Mach. Learn. Cybern. **8**(3), 955–979 (2017)
2. Ding, S., Zhang, N., Zhang, J., Xu, X., Shi, Z.: Unsupervised extreme learning machine with representational features. Int. J. Mach. Learn. Cybern. **8**(2), 587–595 (2017)
3. Liu, P., Huang, Y., Meng, L., Gong, S., Zhang, G.: Two-stage extreme learning machine for high-dimensional data. Int. J. Mach. Learn. Cybern. **7**(5), 765–772 (2016)
4. Zhang, J., Ding, S., Zhang, N., Shi, Z.: Incremental extreme learning machine based on deep feature embedded. Int. J. Mach. Learn. Cybern. **7**(1), 111–120 (2016)
5. Zhang, L., Suganthan, P.N.: A survey of randomized algorithms for training neural networks. Inf. Sci. **364**, 146–155 (2016)
6. Cao, W.P., Wang, X.Z., Ming, Z., Gao, J.Z.: A review on neural networks with random weights. Neurocomputing **275**, 278–287 (2018). https://doi.org/10.1016/j.neucom.2017.08.040
7. He, Y.L., Wang, X.Z., Huang, J.Z.: Fuzzy nonlinear regression analysis using a random weight network. Inf. Sci. **364**, 222–240 (2016)
8. Ren, Y., Suganthan, P.N., Srikanth, N., Amaratunga, G.: Random vector functional link network for short-term electricity load demand forecasting. Inf. Sci. **367**, 1078–1093 (2016)
9. Pao, Y.H., Takefuji, Y.: Functional-link net computing: theory, system architecture, and functionalities. Computer **25**(5), 76–79 (1992)
10. Schmidt, W.F., Kraaijveld, M.A., Duin, R.P.: Feedforward neural networks with random weights. In: 11th IAPR International Conference on Pattern Recognition, pp. 1–4. IEEE (1992)
11. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: a new learning scheme of feedforward neural networks. In: 2004 IEEE International Joint Conference on Neural Networks, pp. 985–990. IEEE (2004)
12. Huang, G.B., Chen, L., Siew, C.K.: Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans. Neural Netw. **17**(4), 879–892 (2006)
13. Zhang, L., Suganthan, P.N.: A comprehensive evaluation of random vector functional link networks. Inf. Sci. **367**, 1094–1105 (2016)
14. Li, M., Wang, D.: Insights into randomized algorithms for neural networks: practical issues and common pitfalls. Inf. Sci. **382**, 170–178 (2017)
15. Wang, D., Li, M.: Robust stochastic configuration networks with kernel density estimation for uncertain data regression. Inf. Sci. **412–413**, 210–222 (2017)
16. Tao, X., Zhou, X., He, Y.L., Ashfaq, R.A.R.: Impact of variances of random weights and biases on extreme learning machine. J. Softw. **11**(5), 440–454 (2016)
17. Balasundaram, S., Gupta, D.: On optimization based extreme learning machine in primal for regression and classification by functional iterative method. Int. J. Mach. Learn. Cybern. **7**(5), 707–728 (2016)
18. Chen, Z.X., Zhu, H.Y., Wang, Y.G.: A modified extreme learning machine with sigmoidal activation functions. Neural Comput. Appl. **22**(3–4), 541–550 (2013)
19. Wang, W., Liu, X.: The selection of input weights of extreme learning machine: a sample structure preserving point of view. Neurocomputing **261**, 28–36 (2017)
20. Lichman, M.: UCI Machine Learning Repository. School of Information and Computer Science, University of California, Irvine (2013). http://archive.ics.uci.edu/ml

21. Yin, H., Gai, K., Wang, Z.: A classification algorithm based on ensemble feature selections for imbalanced-class dataset. In: The 2nd IEEE International Conference on High Performance and Smart Computing, New York, USA, pp. 245–249 (2016)
22. Yin, H., Gai, K.: An empirical study on preprocessing high-dimensional class-imbalanced data for classification. In: 2015 IEEE 17th International Conference on High Performance Computing and Communications; The IEEE International Symposium on Big Data Security on Cloud, New York, USA, pp. 1314–1319 (2015)

Joint Sparse Locality Preserving Projections

Haibiao Liu, Zhihui Lai(✉), and Yudong Chen

College of Computer Science and Software Engineering, Shenzhen University, Shen Zhen, Guang Dong, China
laizhihui@szu.edu.cn

Abstract. Manifold learning and feature selection have been widely studied in face recognition in the past two decades. This paper focuses on making use of the manifold structure of datasets for feature extraction and selection. We propose a novel method called Joint Sparse Locality Preserving Projections (JSLPP). In order to preserve the manifold structure of datasets, we first propose a manifold-based regression model by using a nearest-neighbor graph, then the $L_{2,1}$-norm regularization term is imposed on the model to perform feature selection. At last, an efficient iterative algorithm is designed to solve the sparse regression model. The convergence analysis and computational complexity analysis of the algorithm are presented. Experimental results on two face datasets indicate that JSLPP outperforms six classical and state-of-the-art dimensionality reduction algorithms.

Keywords: Manifold learning · Face recognition · Dimensionality reduction
Feature selection · Sparse feature extraction

1 Introduction

Dimensionality reduction is one of the most important topics in pattern recognition, machine learning and data mining [1–5]. Due to the curse of dimensionality, it's time-consuming to calculate the Euclidean distance between samples. In order to eliminate the redundant features and preserve meaningful features, many dimensionality reduction methods were proposed. Among them, feature extraction and feature selection are the two most important techniques. The purpose of feature extraction methods is to transform the original high-dimensional data into low-dimensional features by using a linear transformation matrix [1]. Therefore, feature extraction is also known as subspace learning. The classical subspace learning methods including Multiple Dimensional Scaling (MDS) [2], Principle Component Analysis (PCA) [3] and Linear Discriminant Analysis (LDA) [4].

MDS, PCA and LDA only consider the global information and fail to discover the underlying manifold structure of the datasets. Compared with the global Euclidean structure of the datasets, the intrinsic manifold structure embedded in the original high-dimensional space is more effective for pattern recognition [8].

Different from the KPCA and KLDA, many nonlinear manifold learning methods such as Isomap [5], Locally Linear Embedding (LLE) [6, 7], and Laplacian Eigenmap [8] can preserve the manifold structure in low-dimensional subspace with lower

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 125–133, 2018.
https://doi.org/10.1007/978-3-319-73830-7_13

computational cost. However, these non-linear manifold learning methods lack of robustness and they fail to evaluate the map on testing data. Therefore, these nonlinear manifold learning techniques might not be suitable for some pattern recognition tasks including face recognition. To overcome the drawbacks, Locality Preserving Projections (LPP) [9, 10] and Neighborhood Preserving Embedding (NPE) [11] were proposed. LPP and NPE are the linear extensions of the traditional manifold learning methods and they were widely used in many applications because of their effectiveness and efficiency.

The above methods only focus on feature extraction and thus they all lack of the ability of feature selection. It's known that feature selection is also an important way to improve the performance on pattern recognition. An effective approach to obtain feature selection is to impose a regularization term on the model. For example, Bradley and Mangasarian proposed L_1-SVM for binary classification task [12]. Wang *et al.* proposed A Hybrid Huberized SVM (HHSVM) [13] combining both L_1-norm and L_2-norm regularization term for sparse feature selection. Unlike the L_1-norm regularization, $L_{2,1}$-norm regularization can generate jointly sparse projection matrix which has better explanation for the selected features. In order to perform subspace learning and feature selection simultaneously, Gu *et al.* proposed feature selection and subspace learning (FSSL) by imposing the $L_{2,1}$-norm on the graph embedding framework [14].

Motivated by previous researches [9, 14], in this paper, we propose a novel method called Joint Sparse Locality Preserving Projections. We construct a graph based regression model and then impose $L_{2,1}$-norm regularization term for feature selection. The main contributions of this paper are as follows:

(1) We propose a novel method called Joint Sparse Locality Preserve Projections (JSLPP) which combines manifold learning and feature selection techniques. We construct a regression model and impose $L_{2,1}$-norm regularization term on the modified regression model for feature selection. In the meantime, we design an iterative algorithm to solve the problem and obtain the optimal solution.
(2) We present a comprehensive theoretical analysis for the iterative algorithm, including the convergence analysis and computational complexity analysis.
(3) Experiments show that JSLPP performs better than the existing subspace learning and feature selection methods.

The rest of this paper is organized as follows. We propose the model and its theoretical analysis in Sect. 2. Experimental results are shown in Sect. 3, and the conclusion is given in Sect. 4.

2 Joint Sparse Locality Preserving Projections

In this section, we first give the motivation of this paper and then propose the model. At last, an iterative algorithm is designed to solve the optimization problem.

2.1 The Motivations

As mentioned in the introduction section, the $L_{2,1}$-norm based on jointly sparse feature selection can greatly improve the recognition performance. Moreover, a sparse projection can also give clearer explanation for the selected features [14]. On the other hands, the manifold learning methods can preserve the local structure of the datasets which are more useful than the global structure for feature extraction in some classification tasks [9]. Therefore, it is desirable to combine the advantages of sparse feature extraction and manifold learning for improving the recognition performance. Thus, we propose a novel manifold learning model called Joint sparse locality preserving projections (JSLPP) for feature extraction and selection by imposing $L_{2,1}$-norm regularization term on the projection matrix to guarantee the joint sparsity.

2.2 Objective Function and Its Solution

In order to integrate manifold learning and sparse regression together to improve the recognition performance, we present the objective function of JSLPP as follows:

$$\min_{\mathrm{A,B}} \sum_{i=1}^{\mathrm{n}} \sum_{j=1}^{\mathrm{n}} \left\|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_j\right\|_2^2 \mathbf{W}_{ij} + \lambda\|\mathbf{B}\|_{2,1} \; s.t. \; \mathbf{A}^T\mathbf{A} = \mathbf{I} \tag{1}$$

where $\mathbf{x}$ is a d-dimensional column vector, n is the training number of the samples, $\mathbf{A} \in R^{d\times k}$ is a basic matrix and $\mathbf{B} \in R^{d\times k}(k << d)$ is a projection matrix, $\mathbf{W} \in R^{d\times k}$ is a weight graph and λ is a regularization parameter. From (1), we have

$$\begin{aligned}
&\textstyle\sum_{ij} \left\|\mathbf{x}_i - \mathbf{A}\mathbf{B}^{\mathrm{T}}\mathrm{x}_j\right\|_2^2 W_{ij} + \lambda\|\mathbf{B}\|_{2,1} \\
&= \textstyle\sum_{ij} \mathrm{tr}(\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_i - 2\mathbf{x}_i^{\mathrm{T}}\mathbf{A}\mathbf{B}^{\mathrm{T}}\mathbf{x}_j + (\mathbf{A}\mathbf{B}^{\mathrm{T}}\mathbf{x}_j)^{\mathrm{T}}\mathbf{A}\mathbf{B}^{\mathrm{T}}\mathbf{x}_j)W_{ij} + \lambda\,\mathrm{tr}(\mathbf{B}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{B}) \\
&= tr(\textstyle\sum_{ij} \mathbf{x}_i^{\mathrm{T}}W_{ij}\mathbf{x}_i - 2\sum_{ij} \mathbf{A}\mathbf{x}_i^{\mathrm{T}}W_{ij}\mathbf{x}_j\mathbf{B}^{\mathrm{T}} + \sum_{ij} (\mathbf{A}\mathbf{B}^{\mathrm{T}}\mathbf{x}_j)^{\mathrm{T}}\mathbf{A}\mathbf{B}^{\mathrm{T}}\mathbf{x}_j W_{ij}) + \lambda\,\mathrm{tr}(\mathbf{B}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{B}) \\
&= \mathrm{tr}(\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}} - 2\mathbf{A}\mathbf{X}\mathbf{W}\mathbf{X}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}} + \mathbf{B}\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}) + \lambda\,\mathrm{tr}(\mathbf{B}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{B})
\end{aligned}$$

where $\mathbf{D}$ is a diagonal matrix, that is $\mathbf{D}_{ii} = \sum_j W_{ji}$. $\mathbf{\Lambda}$ is a diagonal matrix with the i-th diagonal element defined as $\mathbf{\Lambda}_{ii} = \frac{1}{2\left\|\mathbf{B}^i\right\|_2}$, where $\mathbf{B}^i$ is the i-th row of $\mathbf{B}$. Finally, we obtain the optimization problem as follows:

$$\min_{\mathbf{A},\mathbf{B}} \; \mathrm{tr}(\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}} - 2\mathbf{A}\mathbf{X}\mathbf{W}\mathbf{X}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}} + \mathbf{B}\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}) + \lambda\,\mathrm{tr}(\mathbf{B}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{B}) \quad s.t\; \mathbf{A}^T\mathbf{A} = \mathbf{I} \tag{2}$$

To obtain the optimal solutions of the two variables in (2), we design an alternately iterative algorithm. Suppose $\mathbf{A}$ is fixed, we have:

$$l(\mathbf{A},\mathbf{B}) = \mathrm{tr}(\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}} - 2\mathbf{A}\mathbf{X}\mathbf{W}\mathbf{X}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}} + \mathbf{B}\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}) + \lambda\,\mathrm{tr}(\mathbf{B}^{\mathrm{T}}\mathbf{\Lambda}\mathbf{B}) \tag{3}$$

By taking the derivation of $l(\mathbf{A}, \mathbf{B})$ w.r.t $\mathbf{B}$ to be equal to zero, we have:

$$\mathbf{B} = (\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}} + \lambda\mathbf{\Lambda})^{-1}\mathbf{X}\mathbf{W}\mathbf{X}^{T}\mathbf{A} \tag{4}$$

For given B, discarding the constant in (2), we can obtain the following optimization problem

$$\min_{\mathbf{A}} tr(-2\mathbf{B}^{T}\mathbf{X}\mathbf{W}\mathbf{X}^{T}\mathbf{A}) \ \ s.t. \ \ \mathbf{A}^{T}\mathbf{A} = \mathbf{I}. \tag{5}$$

Then, (5) is equal to the following maximization problem

$$\max_{\mathbf{A}} tr(\mathbf{A}^{T}\mathbf{X}\mathbf{W}\mathbf{X}^{T}\mathbf{B}) \quad s.t. \ \ \mathbf{A}^{T}\mathbf{A} = \mathbf{I}. \tag{6}$$

Let SVD of $\mathbf{X}\mathbf{W}\mathbf{X}^{T}\mathbf{B} = \ \mathbf{U}\mathbf{D}\mathbf{V}^{T}$ and from Theorem 4 in [17], we have

$$\mathbf{A} = \mathbf{U}\mathbf{V}^{T}. \tag{7}$$

By alternatively updating $\mathbf{A}$ and $\mathbf{B}$ with (4) and (7) respectively, we eventually obtain the optimal projection matrix $\mathbf{B}$ and the basic matrix $\mathbf{A}$.

2.3 The Convergence

In order to prove the convergence of the proposed algorithm, we need the following Lemmas.

Lemma 1. [15] For any two nonzero-constants a and b, we have the following inequality:

$$\sqrt{\mathrm{a}} - \frac{a}{2\sqrt{b}} \leq \sqrt{b} - \frac{b}{2\sqrt{b}} \tag{8}$$

Lemma 2. [15] For any nonzero vectors $\mathbf{p}$, $\mathbf{p}_t \in R^c$, the following inequality holds:

$$\|\mathbf{p}\|_2 - \frac{\|\mathbf{p}\|_2^2}{2\|\mathbf{p}_t\|_2} \leq \|\mathbf{p}_t\|_2 - \frac{\|\mathbf{p}_t\|_2^2}{2\|\mathbf{p}_t\|_2} \tag{9}$$

With Lemmas 1 and 2, we have the following theorem.

Theorem 1. The iteration approach presented in Sect. 2.2 will monotonically decrease the objective function value in each iteration and converge to the local optimum.

Proof: For ease of representation, we denote the objective function (1) as $J(\mathbf{B}, \mathbf{A}, \mathbf{W}, \mathbf{D}, \mathbf{\Lambda}) = J(\mathbf{B}, \mathbf{A}, \mathbf{\Lambda})$. Suppose in the $(t-1)$-th iteration, we have $\mathbf{B}_{t-1}$, $\mathbf{A}_{t-1}$ and $\mathbf{\Lambda}_{t-1}$. From (4), we can find that

$$J(\mathbf{B}_{(t)}, \mathbf{A}_{(t-1)}, \mathbf{\Lambda}_{(t-1)}) \leq J(\mathbf{B}_{(t-1)}, \mathbf{A}_{(t-1)}, \mathbf{\Lambda}_{(t-1)}) \tag{10}$$

For $\mathbf{A}_t$, as its optimal value comes from the SVD decomposition value of $\mathbf{XWX}^T\mathbf{B}$ which further decreases the objective function, we have

$$J(\mathbf{B}_{(t)}, \mathbf{A}_{(t)}, \mathbf{\Lambda}_{(t-1)}) \leq J(\mathbf{B}_{(t-1)}, \mathbf{A}_{(t-1)}, \mathbf{\Lambda}_{(t-1)}) \tag{11}$$

Once the optimal $\mathbf{B}_{(t)}$ and $\mathbf{A}_{(t)}$ are obtained, we have

$$\begin{aligned}&\mathrm{tr}(-2\mathbf{A}_{(t)}\mathbf{XWX}^{\mathrm{T}}\mathbf{B}_{(t)}^{\mathrm{T}} + \mathbf{B}_{(t)}\mathbf{XDX}^{\mathrm{T}}\mathbf{B}_{(t)}^{\mathrm{T}}) + \lambda\,\mathrm{tr}(\mathbf{B}_{(t)}^{\mathrm{T}}\mathbf{\Lambda}_{(t-1)}\mathbf{B}_{(t)})\\ &\leq tr(-2\mathbf{A}_{(t-1)}\mathbf{XWX}^{\mathrm{T}}\mathbf{B}_{(t-1)}^{\mathrm{T}} + \mathbf{B}_{(t-1)}\mathbf{XDX}^{\mathrm{T}}\mathbf{B}_{(t-1)}^{\mathrm{T}}) + \lambda\,\mathrm{tr}(\mathbf{B}_{(t-1)}^{\mathrm{T}}\mathbf{\Lambda}_{(t-1)}\mathbf{B}_{(t-1)})\end{aligned}$$

That is

$$\begin{aligned}&\mathrm{tr}(-2\mathbf{A}_{(t)}\mathbf{XWX}^{\mathrm{T}}\mathbf{B}_{(t)}^{\mathrm{T}} + \mathbf{B}_{(t)}\mathbf{XDX}^{\mathrm{T}}\mathbf{B}_{(t)}^{\mathrm{T}}) + \lambda\sum_i \frac{\left\|\mathbf{B}_{(t)}^i\right\|_2^2}{\left\|\mathbf{B}_{(t-1)}^i\right\|_2}\\ &\leq tr(-2\mathbf{A}_{(t-1)}\mathbf{XWX}^{\mathrm{T}}\mathbf{B}_{(t-1)}^{\mathrm{T}} + \mathbf{B}_{(t)}\mathbf{XDX}^{\mathrm{T}}\mathbf{B}_{(t)}^{\mathrm{T}}) + \lambda\sum_i \frac{\left\|\mathbf{B}_{(t-1)}^i\right\|_2^2}{\left\|\mathbf{B}_{(t-1)}^i\right\|_2}\end{aligned} \tag{12}$$

Then, we have

$$\begin{aligned}&\mathrm{tr}(-2\mathbf{A}_{(t)}\mathbf{XWX}^{\mathrm{T}}\mathbf{B}_{(t)}^{\mathrm{T}} + \mathbf{B}_{(t)}\mathbf{XDX}^{\mathrm{T}}\mathbf{B}_{(t)}^{\mathrm{T}}) + \lambda\sum_i \left\|B_{(t)}^i\right\|_2 - (\lambda\sum_i \left\|B_{(t)}^i\right\|_2 - \lambda\sum_i \frac{\left\|\mathbf{B}_{(t)}^i\right\|_2^2}{\left\|\mathbf{B}_{(t-1)}^i\right\|_2})\\ &\leq tr(-2\mathbf{A}_{(t-1)}\mathbf{XWX}^{\mathrm{T}}\mathbf{B}_{(t-1)}^{\mathrm{T}} + \mathbf{B}_t\mathbf{XDX}^{\mathrm{T}}\mathbf{B}_{(t)}^{\mathrm{T}}) + \lambda\sum_i \left\|B_{(t-1)}^i\right\|_2 - (\lambda\sum_i \left\|B_{(t-1)}^i\right\|_2 - \lambda\sum_i \frac{\left\|\mathbf{B}_{(t-1)}^i\right\|_2^2}{\left\|\mathbf{B}_{(t-1)}^i\right\|_2})\end{aligned} \tag{13}$$

Then combining (12) and (13) and Lemma 2, we further obtain

$$\begin{aligned}&tr(-2\mathbf{A}_{(t)}\mathbf{XWX}^{\mathrm{T}}\mathbf{B}_{(t)}^{\mathrm{T}} + \mathbf{B}_{(t)}\mathbf{XDX}^{\mathrm{T}}\mathbf{B}_{(t)}^{\mathrm{T}}) + \lambda\left\|\mathbf{B}_{(t)}^i\right\|_2\\ &\leq tr(-2\mathbf{A}_{(t)}\mathbf{XWX}^{\mathrm{T}}\mathbf{B}_{(t-1)}^{\mathrm{T}} + \mathbf{B}_{(t-1)}\mathbf{XDX}^{\mathrm{T}}\mathbf{B}_{(t-1)}^{\mathrm{T}}) + \lambda\left\|\mathbf{B}_{(t-1)}^i\right\|_2\end{aligned}$$

That is

$$J(\mathbf{B}_{(t)}, \mathbf{A}_{(t)}, \mathbf{\Lambda}_{(t)}) \leq J(\mathbf{B}_{(t-1)}, \mathbf{A}_{(t-1)}, \mathbf{\Lambda}_{(t-1)}) \tag{14}$$

Therefore, the algorithm will converge to the local optimum.

2.4 Computational Complexity Analysis

The algorithm first obtain the weight matrix $\mathbf{W}$, then get the optimal projection matrix $\mathbf{B}$ and the basic matrix $\mathbf{A}$ as well as the diagonal matrix $\mathbf{\Lambda}$. The main computational cost of the iterative algorithm is to compute the projection matrix $\mathbf{B}$, the basic matrix $\mathbf{A}$ and the diagonal matrix $\mathbf{\Lambda}$. Computing the projection matrix $\mathbf{B}$ needs $O(d^3)$, the basic matrix $\mathbf{A}$ needs $O(d^3)$ and the diagonal matrix $\mathbf{\Lambda}$ needs $O(d^2)$. If the algorithm needs T iteration steps, then the total computational complexity is $O(n^2 + Tnd^3 + Tnd^3 + Td^2)$.

2.5 JSLPP Algorithm

The code of JSLPP algorithm is as follows:

BEGIN

Input: the training data $\mathbf{X}$, the weight matrix $\mathbf{W}$, the diagonal matrix $\mathbf{D}$, the dimensionality d of the sample, the desired dimensionality k of matrix $\mathbf{A}$ and $\mathbf{B}$, and the regularization parameter λ.

Program:

1: $\mathbf{W}$ = constructW($\mathbf{X}$, options)

2: $\mathbf{A}$ = rand(d, k)

3: $\mathbf{B}$ = rand(d, k)

4: $\mathbf{D}_{ii} = \sum_j W_{ji}$

5: for iter←1 to maxIter

6: $\mathbf{B} = (\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}} + \lambda\mathbf{\Lambda})^{-1}\mathbf{X}\mathbf{W}\mathbf{X}^{T}\mathbf{A}$

7: ($\mathbf{U}$, $\mathbf{V}$) = SVD($\mathbf{X}\mathbf{W}\mathbf{X}^{T}\mathbf{B}$)

8: $\mathbf{A} = \mathbf{U}\mathbf{V}^{T}$

9: Until Converge

10: End

11: $\mathbf{Y}$ = $\mathbf{B}$ * $\mathbf{X}$

Output: Low-dimensional features $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n]$

END

3 Experiments

In this section, a set of experiments are presented to evaluate the proposed JSLPP algorithm for feature extraction and selection. We compared it with PCA, LPP, L_1-norm regularized sparse subspace learning methods SPCA, the most related $L_{2,1}$-norm based Feature Selection and Subspace learning (FSSL), RFS [15] and $L_{2,1}$-norm regularized discriminative feature selection for unsupervised learning UDFS, SAIR [16]. Six methods mentioned above were compared with the JSLPP in the same experimental condition. The datesets are all divided into training sets and test sets. The number of training samples are set as 4, 6, and the rest data are used as testing sets,

respectively. In all experiments, we first performed feature extraction and selection, then used the nearest neighborhood classifier to perform classification.

3.1 Experiments on the AR Face Database

There are over 4000 color face images of 126 people in AR face database, we selected 120 images of 120 people (65 men and 55 women) from this dataset. All images are the frontal views of faces with different facial expressions, lighting conditions, and occlusions, and they are normalized to 50 × 40 pixels.

In the experiment, the number of class is 120 and each class has 20 samples. $l(l = 4, 6)$ images of each class were randomly selected and used for training and the remaining images were used for test. The optimal value of parameter γ was selected form the set $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$, Table 1 lists the average performance of different methods on the AR face database based on 10 times running, and the average recognition rates versus the dimensions of the projection are shown in the Fig. 1.

Table 1. The performance (recognition rate, standard deviation and dimension) of different methods on the AR face database

Training samples	PCA	LPP	SPCA	FSSL	UDFS	SAIR	JSLPP
4	76.20	79.81	76.20	89.84	85.44	83.55	77.76
	±4.58	±8.86	±4.58	±9.45	±8.44	±8.20	±8.20
	110	95	110	110	105	100	75
6	79.96	85.57	79.96	95.39	89.73	91.05	87.87
	4.86	±7.31	±4.85	±7.38	±5.75	±6.17	±10.38
	110	105	110	115	100	100	60

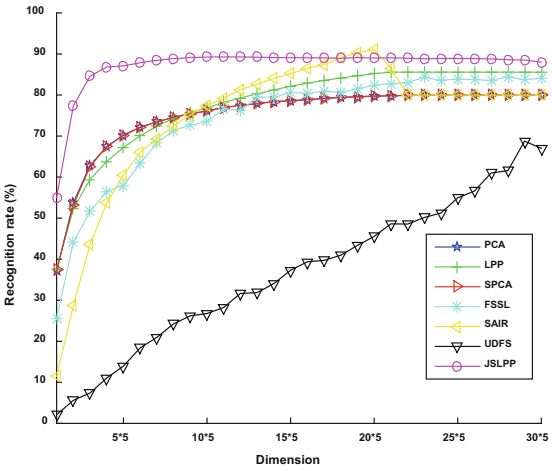


Fig. 1. The recognition rates (%) versus the dimensions of different methods on the AR face database.

3.2 Experiments on the ORL Face Database

The ORL face database have 40 people, and each person have 10 images. The images were taken at different times, varying lighting, facial expression (open or closed eyes, smiling or not smiling) and facial details (glasses or no glasses). In Table 2, we lists the average performance of different methods on the ORL face database, and the average recognition rates versus the dimensions of the projection are shown in Fig. 2.

Table 2. The performance (recognition rate, standard deviation and dimension) of different methods on the ORL face database

Training samples	PCA	LPP	SPCA	FSSL	L21R21	UDFS	SAIR	JSLPP
4	93.58	78.21	93.54	93.42	89.67	93.54	95.08	94.54
	±1.95	±2.96	±1.93	±1.70	±2.09	±2.10	±2.44	±1.72
	135	85	130	35	40	150	40	120
6	95.94	88.75	96.00	96.25	92.63	95.81	97.06	97.94
	±1.59	±2.76	±1.66	±1.85	±2.93	±1.59	±1.35	±1.41
	105	65	100	35	40	150	40	40

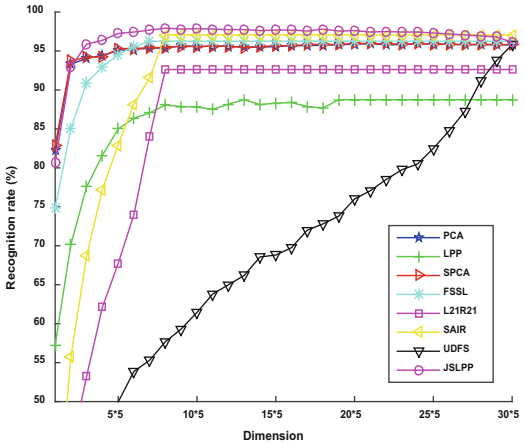


Fig. 2. The recognition rates (%) versus the dimensions of different methods on the ORL face databases.

4 Conclusion

In this paper, a novel method called Joint Sparse Locality Preserving Projection (JSLPP) is proposed for sparse subspace learning by considering manifold learning and feature selection techniques. The $L_{2,1}$-norm is introduced in the JSLPP model, an iterative algorithm is designed to solve the optimization problem. We prove the convergence of the proposed algorithm, and the computational complexity is also presented. Experiments on two well-known face datasets show that JSLPP performs better than the traditional feature extraction and linear manifold learning methods.

Acknowledgement. This work was supported in part by the Natural Science Foundation of China (Grant 61573248, Grant 61773328, Grant 61375012 and Grant 61703283), China Postdoctoral Science Foundation (Project 2016M590812 and Project 2017T100645), the Guangdong Natural Science Foundation (Project 2017A030313367 and Project 2017A030310067), and Shenzhen Municipal Science and Technology Innovation Council (No. JCYJ2017030215 3434048).

References

1. Batur, A.U., Hayes, M.H.: Linear subspace for illumination robust face recognition. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, December 2001
2. Cox, T.F., Cox, M.A.A.: Multidimensional scaling on a sphere. Commun. Stat. Theory Methods **20**(9), 2943–2953 (1991)
3. Turk, M., Pentland, A.P.: Face recognition using eigenfaces. In: IEEE Conference on Computer Vision and Pattern Recognition (1991)
4. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 711–720 (1997)
5. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**, 2319–2323 (2000)
6. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**, 2323–2326 (2000)
7. Saul, L.K., Roweis, S.T.: Think globally.: fit locally: unsupervised learning of low dimensional manifolds. J. Mach. Learn. Res. **4**, 119–155 (2003)
8. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Proceedings of Conference on Advances in Neural Information Processing System, vol. 15 (2001)
9. He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.J.: Face recognition using laplacianfaces. IEEE Trans. Pattern Anal. Mach. Intell. **27**(3), 328–340 (2005)
10. He, X., Niyogi, P.: Locality preserving projections. In: Neural Information Processing Systems, vol. 16, p. 153 (2004)
11. He, X., Cai, D., Yan, S., Zhang, H.J.: Neighborhood preserving embedding. In: ICCV, pp. 1208–1213 (2005)
12. Bradley, P., Mangasarian, O.: Feature selection via concave minimization and support vector machines
13. Wang, L., Zhu, J., Zou, H.: Hybrid huberized support vector machines for microarray classification. In: ICML (2007)
14. Gu, Q., Li, Z., Han, J.: Joint feature selection and subspace learning. In: International Joint Conference on Artificial Intelligence, pp. 1294–1299. AAAI Press (2011)
15. Nie, F., Huang, H., Cai, X., Ding, C.: Efficient and robust feature selection via joint $L_{2,1}$ norms minimization. In: Advances in Neural Information Processing Systems, vol. 23, pp. 1813–1821 (2010)
16. Ma, Z., Yang, Y., Sebe, N., Member, S., Zheng, K., Hauptmann, A.G.: Classifier-specific intermediate representation, **15**(7), 1628–1637 (2013)
17. Zou, H., Hastie, T., Tibshirani, R.: Sparse principal component analysis. J. Comput. Graph. Stat. **15**(2), 265–286 (2006)

Research on Dynamic Safe Loading Techniques in Android Application Protection System

Shubin Cai[1(✉)], Rongjie Huang[1], Ningsheng Yang[1], Jinwen Jiang[1(✉)], Zhong Ming[1(✉)], Zhengping Liang[1(✉)], and Zhiguang Shan[2,3]

[1] Shenzhen University, Shenzhen 518060, Guangdong Province, China
{shubin,mingz,liangzp}@szu.edu.cn, rongjiehuang24@163.com, ningsheng.yang@qq.com, 448012596@qq.com

[2] Engineering Research Center of Mobile Internet Application Middleware, Shenzhen, Guangdong Province, China
shanzg@cei.gov.cn

[3] State Information Center of China, Bejing, China

Abstract. Android is a widespread used embedded system. The number of Android applications has been rapidly growing. Because of Android open source policy and limited application security mechanism, Android applications are confronted with many serious security threats. By malicious reverse and illegal tampering, thousands of Android applications have been infected and millions of users have been exposed to dangers. In this paper, we proposed an improved Android applications protection system based on DEX block encryption and multi-file features checksum. Experiment results show that the proposed system is more reliable than the commonly-used Android application protection systems when facing with attack tools such as APK Tools and IDA pro.

1 Introduction

Google has always been committed to protecting users' privacy and device security. In order to cope with various forms of potential threat applications(PHAs) more efficiently, Since 2015, Google worked closely with device manufactures, system on a chip(SoC) providers and telecom carriers to release and maintenance a set of security service which provides both on-device services and remote-based services to prevent users from installing or using PHAs [1,2]. Although Google tries its best to help users protect personal privacy and data security, however, Google has not put too much effort on protecting APP intellectual property and copyright, but instead shifting this burden to open source protocols. To some extent, this is a rather irresponsible move that could directly lead to the loss of good developers and cause a dramatic deterioration in the entire Android software ecosystem environment. Therefore, it is urgent to contain unauthorized reverse software cracking and code plagiarism. Commissioning professional lawyers to draw up R & D commission contracts, clarify the ownership of intellectual property and render for breach of contract costs too much and cannot

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 134–143, 2018.
https://doi.org/10.1007/978-3-319-73830-7_14

be afforded by small and medium development teams. Hence, a strong technical support is urgently needed. In this paper, we proposed a method to protect APP based on DEX file block encryption and multi-file checksum.

2 Related Work

In view of the serious situation of Android mobile terminal security, the reinforcement of Android application software security has a very great practical significance. On the one hand, it can effectively protect the core code from plagiarism, on the other hand, it can effectively prevent malicious code injection [3–7]. At present, research on Software protection technology mainly concentrate on two aspects: One is based on the software protection technology of hardware [8–10], and another is purely based on the software protection technology. The great advantages of hardware-assisted software protection technology include robustness and strong protection intensity. But hardware-based protection also has some shortcomings like complicated deployment and expensive cost. Compared to hardware protection technology, software-based protection technology is easier to implement and the latters price is cheaper. The pure software-based protection technology can be divided into several categories: Registration verification [11,12], Software encryption [13,14], Software watermark [15,16], Software tamper proofing technology [17,18], Code confusion [19,20]. First of all, we proposed DEX file block encryption and multi-file features checksum aiming at the threats Android application faced. DEX file block encryption slices DEX file into many subfiles and encrypts each one respectively. The system decrypts and reorganizes the DEX file in memory only if the integrity verification is correct. This pattern can simply guarantee the integrity and confidentiality. However, attacker can bypass the single files digital signature and integrity verification, we are in urgent need of a more complicated verification institution. Hence we proposed a multi-file features checksum to efficiently avoid piracy. This technology verifies different DEX files in application with both local check and remote check and randomly inserts verification code into application to defend crack.

3 Concepts and the Proposed Model

3.1 DEX Block Encryption

DEX file is an executable file on Android platform. Every Android application includes many DEX files which containing its whole source code. The source code can be gathered via disassembler toolkit. Hence the protection of DEX file is the most fundamental aspect of Android application protection. DEX block encryption method will divide executable program data packet into several original files that will be encrypted to get corresponding encrypted file. By decrypting each file, it is possible to get the original executable program data packet from the index information of each file, so that the confidentiality of the executable data

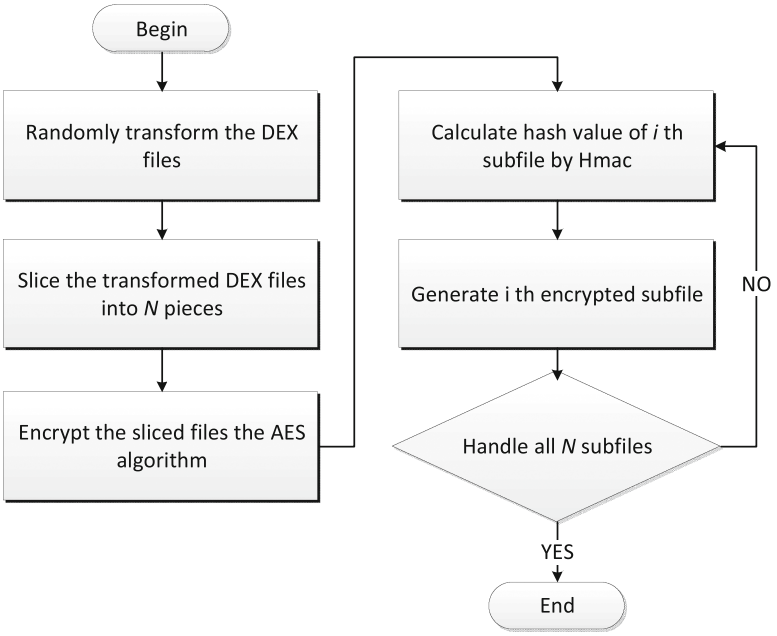


Fig. 1. Flow chart of DEX block encryption

packet can be effectively ensured, and source code of the application program can be prevented from being tampered with and stolen, greatly improving the difficulty of cracking the application. Figure 1 is the flow chart of DEX block encryption. Firstly, system handles DEX files with random transform and slices all transformed files into several pieces. After that, system generates a secret key by random number generator and encrypts all subfiles with AES algorithm. Then, system uses Hmac algorithm to generate hash value of integrity of subfiles. This hash value can be used to verify the integrity of subfiles when decrypting and reorganizing subfiles. Finally, system generates encrypted block files for each subfiles. Because the Android System native class loader can only load intact DEX file, loading shelling DEX file requires additional step and component: in-memory decryption reorganization and custom loader. We used systematic library, called libdvm.so, to implement the custom loading. When loading the encrypted Android application, system verifies application's integrity at first, then decrypts and reorganizes the data part of encrypted subfiles.

3.2 Multi-file Features Checksum

The security of the existing digital signature technology of Android system is limited. Attacker can crack it with root authority or re-signature. Through the reverse analysis, attacker can locate the verification code and bypass

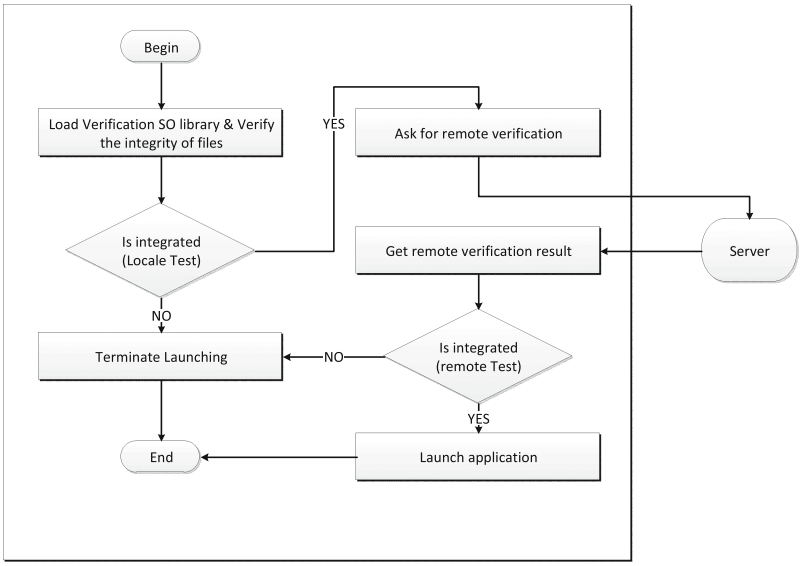


Fig. 2. Flow chart of multi-file features checksum

the verification process via modifying the verification code. Aiming at the problem that the signature verification is bypassed, a multi-file feature verification method is proposed. The following will introduce the principle of multi-file signature verification and multi-file signature verification process. The multi-file signature verification method verifies the integrity of different files in the application with local and remote double-check mechanism and inserts the verification code randomly in the application program. This approach prevents applications from privacy and falsifying and isolates tampered applications from using the services provided by the server. Figure 2 is the flow chart of multi-file integrity verification. First of all, we calculate the integrity hash value of files and use reflection mechanism to invoke remote verification on the server and compile the verification library Hmacsha.so. When system verifying the integrity of application, it uses verification function in Hmacsha.so to verify DEX files and compares the result with local integrity hash value. Only if two values are equal, the remote verification starts. If remote verification failed, then local system would refuse to launch the application. After multi-file signature processing, the application will perform file integrity verification plug-in code in the running process, which computes the integrity hash of the file by calling the bottom of file verification so library to calculate the integrity of the file hash, which compared with local stored value. When the local file is validated, the file integrity value and the APK source are submitted to the server. The server verifies the file integrity and returns the file verification result to the client.

3.3 The Overall Framework

We designed and implemented an Android application protection system based on DEX file block encryption and multi-file features checksum technology. Our proposed system contains six modules. APK processing module is mainly composed of decompression, decompilation, re-packing and re-signature functions. DEX file processing module divides and encrypts classes.dex in the original application files to generate multiple block encrypted subfiles. SO processing module encrypts core SO library. The encrypted SO library can not be loaded and called normally. System needs to check the integrity of the core library before calling it. Only if system ensured that the loaded core SO library is intact, then system would decrypt the SO library, and dynamically register the methods in the original core repository with the virtual machine. Software run-time environment detection monitors the environment of the application in order to ensure the security of the enforced application environment. Environment detection module contains simulator detection and debugger detection. Attacker can gather the contents of the registers and stacks and trace the execution flow of the program according to the intermediate results of the execution of the application [19,20]. Therefore, we can increase the real-time environment test to judge whether running in the simulator or debugger. If application was detected in a simulator environment or debugger environment, system would exit the entire application or jump to another branch. Multi-file features processing module verifies the integrity of the files in the APK, such as classes.dex, resource files, libs files, and AndroidManifest.xml. After detecting the multi-file features, system tests the file integrity the plug-in code by calling the bottom of the file validation SO library. If local verification got a correct answer, system would packet the file integrity value, APK source and other information and submit these information to the server. Then the server-side integrity verification executes and returns the result. Shell program processing module generates shell program. It constructs a custom class loader DynamicClassLoader object and replaces the original application running environment and other functions. When the program starts, the shell program prior to obtain the right to execute the program and execute a series of processing after the application program are reinforced. Figure 3 is the flow chart of reinforcing APK.

4 Experimental Results and Analysis

In order to evaluate our proposed system in all directions, we designed four different experiments. We evaluates our proposed protection system based on four indexes: anti-attack, launch time, additional volume and availability. Without loss of generality, we randomly chosen 15 application samples from f-droid (https://f-droid.org.com).

4.1 Anti-attack

In order to verify the performance of anti-reverse, anti-tamper and environment detection modules, we conducted a series of experiments to test APKs.

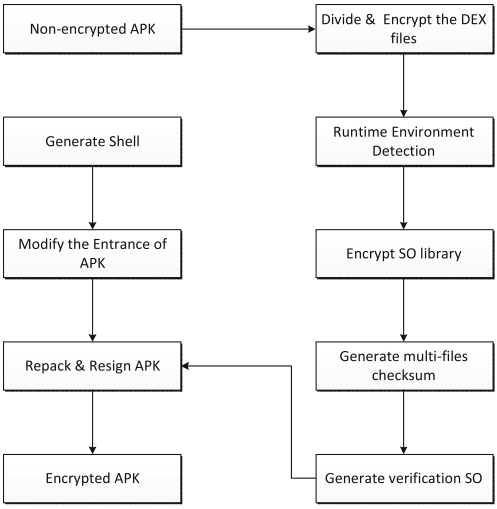


Fig. 3. Main workflow of system

Table 1. The result of anti-attack in three patterns

Attack method	appfortify	bangcle	ijiami	legu	Our method
Reserve attack by apktool	F	T	F	T	F
Reserve attack by dex2jar	T	T	T	T	F
Tampering attack by tampering files	F	F	F	T	F
Tampering attack by injection in java layer	F	F	F	F	F
Debugging attack by IDA Pro	F	F	F	F	F
Debugging attack by GDB	F	F	F	F	F

We simulated three different attack patterns on encrypted APKs and compared different encryption firmwares. The result shows in Table 1(where T: Attack succeeded, F: Attack failed). We can obviously see that our proposed system can defend all six attack tools where several commercial firmware almost failed to defend the reverse attack.

4.2 Startup Time

The reinforcement contains many additional verification steps to keep safety. Hence encrypted APKs has longer launch time than non-encrypted APKs. The shorter launch time performs better. According to Fig. 4, we can see that startup time in our pattern not performs excellent. When the application is hardened, all application startup times increase in that the time it takes for the shell to execute

during startup. In all test samples, the maximum change time of Gtalksms is 1299 ms, the minimum change time of Better Wifi onoff is 136 ms, and the average start time of APK after reinforcement increases 480.07 ms. In addition, our encryption method causes low performance fluctuation within different APKs. That indicates our proposed method may perform better on large-scale APKs. But our method provides the best security performance in anti-attack test. So the additional launch time in our system seems acceptable.

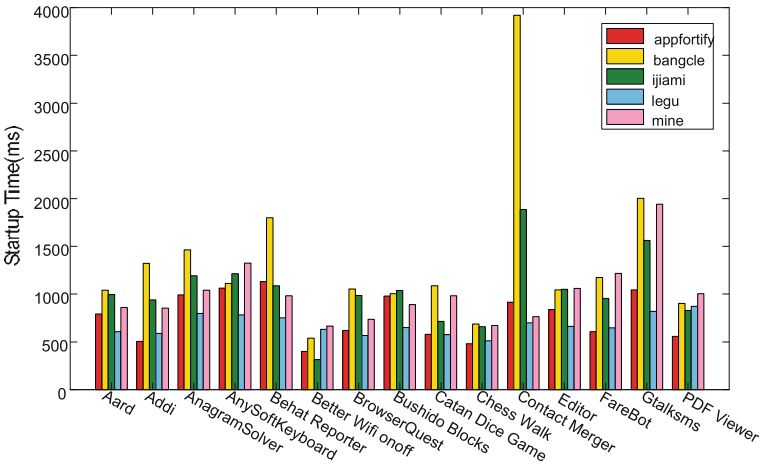


Fig. 4. Startup time of encrypted APK

4.3 Additional Volume

Additional volume of APKs occupies more space in very limited ROM in our mobile devices. So consumers prefers miniature APKs. The additional volume mainly caused by additional verification information and safety components. The volume of encrypted APKs shows in Fig. 5. Compared with other four commercial encryption firmwares, the volume of APK encrypted by our system performed mediocre in most cases.

4.4 Availability

In order to verify the availability of the application after reinforcement, the samples in experiment covered five different categories including learning & reading, system tools, multimedia, life services and social communication. We randomly selected 40 APKs in each categories and encrypted these APKs with our software protection system and tested whether the hardened application can be executed correctly. Table 2 shows the results of correctly executable applications.

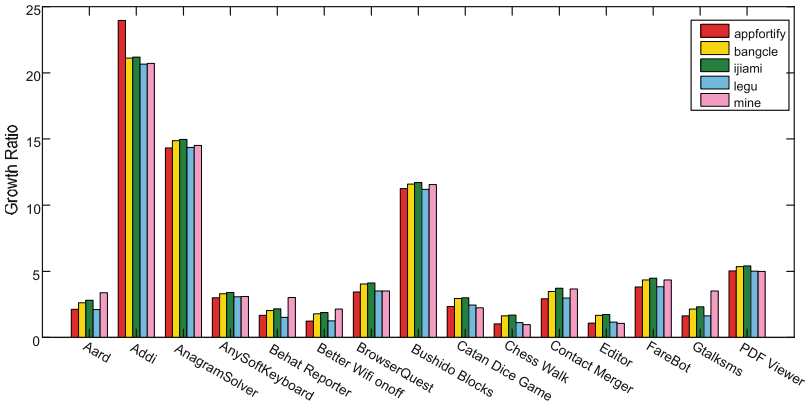


Fig. 5. Volume of encrypted APK

Table 2. The result of anti-attack in three patterns

Number of samples	Execute correctly after reinforcement	Success rate
200	183	91.5%

The results show that more then 90% of applications can be successfully and correctly implemented. Only a small part of the applications can not work properly after the reinforcement failure or reinforcement. There are two main reasons for the failure of system reinforcement: First, due to the reinforcement of the application before the application has been reinforced in the application to increase the resistance to anti-compiler tools and other functional modules, resulting in software protection system decompilation APK and re-packaged steps failure. Second, because the original APK need in the context of the specific context to be able to start normal, general software reinforcement can not be met, resulting in the failure of the program after the reinforcement load.

5 Conclusion

In this paper, we propose a software protection system based on DEX block encryption and multi-file features checksum technology. Compared with the existing methods, our proposed system has obvious improvement on anti-attack while the performance on launch time and additional volume are acceptable. In addition, our proposed method can correctly handle most Android applications except very few encrypted ones.

6 Future Work

Take into account the compatibility bug when deploying applications on different versions of the operating system. The potential bug may emerge because of

modification of system core library. Consider the utilization of distributed technology to optimize the remote verification. The verification deployed on single server may deceived by attacking server while distributed verification can defend single point of failure.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China under Grants NSFC 61672358.

References

1. Google Android Security Team: Android security 2016 year in review. Technical report (2017)
2. Google Android Security Team: Android security 2015 year in review. Technical report (2016)
3. Suarez-Tangil, G., Tapiador, J.E., Peris-Lopez, P., et al.: Evolution, detection and analysis of malware for smart devices. IEEE Commun. Surv. Tutor. **16**(2), 961–987 (2014)
4. Rastogi, V., Chen, Y., Jiang, X.: Catch me if you can: evaluating android anti-malware against transformation attacks. IEEE Trans. Inf. Forensics Secur. **9**(1), 99–108 (2013)
5. Wheeler, D.M., Conyers, A., Luo, J., et al.: Java security extensions for a Java server in a hostile environment. In: Computer Security Applications Conference, p. 64. IEEE Computer Society (2001)
6. Garber, L.: Have Java's security issues gotten out of hand? Computer **45**(12), 18–21 (2012)
7. Lin, Y.D., Huang, C.Y., Wright, M., et al.: Mobile application security. Computer **47**(6), 21–23 (2014)
8. Fernandes, E., Crispo, B., Conti, M.: FM, 99.9, radio virus: exploiting FM radio broadcasts for malware deployment. IEEE Trans. Inf. Forensics Secur. **8**(6), 1027–1037 (2013)
9. Davi, L., Koeberl, P., Sadeghi, A.R.: Hardware-assisted fine-grained control-flow integrity: towards efficient protection of embedded systems against software exploitation. IEEE (2014)
10. Bertels, K., Sima, V.M., Yankova, Y., et al.: HArtes: hardware-software codesign for heterogeneous multicore platforms. IEEE Micro **30**(5), 88–97 (2010)
11. Xi, K., Hu, J.: Dual layer structure check (DLSC) fingerprint verification scheme designed for biometric mobile template protection. In: ICIEA 2009 IEEE Conference on Industrial Electronics and Applications, pp. 630–635. IEEE (2009)
12. Huang, N., Huang, X.T., He, X.W.: A new algorithm of software copyright protection based on multi-scale triangular mapping. In: International Symposium on Information Science and Engineering, pp. 472–475 (2011)
13. Zhangjie, F., Shu, J., Sun, X., Linge, N.: Smart cloud search services: verifiable keyword-based semantic search over encrypted cloud data. IEEE Trans. Consum. Electron. **60**(4), 762–770 (2014)
14. Mller, T., Freiling, F.C.: A systematic assessment of the security of full disk encryption. IEEE Trans. Dependable Secur. Comput. **12**(5), 491–503 (2015)
15. Saha, D., Sur-Kolay, S.: Watermarking in hard intellectual property for pre-fab and post-fab verification. IEEE Trans. Very Large Scale Integr. Syst. **23**(5), 801–809 (2015)

16. Piper, A., Safavi-Naini, R.: Scalable fragile watermarking for image authentication. IET Inf. Secur. **7**(4), 300–311 (2013)
17. Kanuparthi, A.K., Zahran, M., Karri, R.: Architecture support for dynamic integrity checking. IEEE Trans. Inf. Forensics Secur. **7**(1), 321–332 (2012)
18. Kayaalp, M., Ozsoy, M., Ghazaleh, N.A., et al.: Efficiently securing systems from code reuse attacks. IEEE Trans. Comput. **63**(5), 1144–1156 (2014)
19. Collberg, C.S., Thomborson, C.D., Low, D.W.K.: Obfuscation techniques for enhancing software security: CA. US6668325 (2003)
20. Collberg, C., Thomborson, C., Low, D.: A Taxonomy of Obfuscating Transformations. Department of Computer Science the University of Auckland New Zealand (1997)

Research on Optimizing Last Level Cache Performance for Hybrid Main Memory

Hua Zheng[1], Zhong Ming[1(✉)], Meikang Qiu[2], and Xi Zhang[1]

[1] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
zhenghuamail@126.com, mingz@szu.edu.cn, zxsay@126.com
[2] Department of Computer Science, Pace University, New York, USA
qiumeikang@yahoo.com

Abstract. Hybrid main memory including DRAM and non-volatile memory (NVM) such as phase change memory (PCM) has became a perfect substitute to DRAM-based main memory. Because it has the advantage about high performance and energy-efficient in embedded systems. The effective management of last level cache is very important which can reduce cache misses and has important practical significance on the improvement of overall system performance. In last level caches, the common used cache replacement algorithm Least Recently Used (LRU) may cause cache pollution by inserting non-reusable data into the cache. In this article we research the hybrid main memory but now the existing cache policies fail to fully solve the asymmetry between the operations of NVM and DRAM. To solve these problems we mentioned above, we propose a Process-based Pollute Region Isolation (PPRI) algorithm for improving the efficiency of last level cache utilization. It is a good way to eliminate competition between reusable and nonreusable cache lines. We also propose an improved last-level cache management scheme ILRU for the hybrid main memory which improves the cache hit ratio and minimizes write-backs to PCM. Experimental results show that the proposed framework can get better performance (average improved 17.39%) and more energy saving (average decreased 12.46%) compared with the latest cache management schemes for hybrid main memory architecture.

Keywords: Hybrid main memory · Last level cache · Pollute buffer
Nonreusable pages · Write-back aware · Non-volatile memory
Embedded system

1 Introduction

DRAM-based main memory has Encountered the limitation in many aspects, such as power consumption and performance. Replacing it with non-volatile memories (NVM), such as phase change memory (PCM) [1, 2], has received widespread attention because of their ideal properties: low power consumption, high density, byte addressability, and better scalability.

However, unprotected PCM has very limited write endurance compared to DRAM. A PCM cell is only capable of 10^8 to 10^9 writes before wearing out [3]. Excessive writes

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 144–153, 2018.
https://doi.org/10.1007/978-3-319-73830-7_15

in PCM could also lead to high power consumption although read power is in the same level as DRAM [4]. In contrast, DRAM can sustain at least 10^{16} writes [5]. The time of write must be reduced in PCM for avoiding fast wear-out.

Hybrid main memory with DRAM and PCM has been proposed as a great solution for high performance and energy saving computer systems. Because we want to fully exploit the superiority of various technologies of different memory material. There are two architectures have been proposed: (i) a main memory system consisting of PCM storage coupled with a small DRAM buffer [6]. (ii) PCM and DRAM at the same level which forms the main memory [7, 8]. In this work, we focus on the second approach and refer it as the hybrid main memory architecture [9].

The most common used cache management algorithms LRU can not replace the data very well [24], It often replace the reusable data by non-reusable one, that will make some problem, which called the cache pollution. Cache pollution increases memory transactions and influence the performance of the processor because of the "memory gap". We propose the Process-based Pollute Region Isolation (PPRI) to set a part of region for pollute data which called the pollute buffer, which is dispart from the reusable data and the non-reusable data. In this way we dispart the data into two part which can avoid the fierce competition between the data.

We proposed a write-back aware shared Last Level Cache replacement scheme ILRU for getting high performance and energy saving in hybrid main memory architecture as a variant of LRU, which is fast and efficient. Experiment shows that the proposed scheme outperforms LRU and HAP [10] in both performance and energy efficiency under the environment of hybrid main memory. Our contributions can be conclude as follows.

- We introduced the concept of pollute regions which is also called the poor locality regions. We use the page coloring method to divide the last level cache into several non-overlapping areas. After use this method we can isolate the weak locality data into a special cache area in the last level cache, which is dispart from the "hot" data, So, the non-reusable data will not influence the "hot" data that the performance of memory will be improved and the energy consumption will be reduced.
- We proposed improved least recently used algorithm (ILRU) LLC management scheme which decides the position of the latest visited block according to its state, there are two kinds of block clean one and dirty one and two types of memory (PCM and DRAM), as well as the recent-history cache miss information for example which type of the memory incurs more cache misses in current program execution phase. For minimizing the cost of the memory accesses [11].
- We combined the Process-based Pollute Region Isolation (PPRI) method and improved Least Recently Used (ILRU) algorithm together used in the LLC management for hybrid main memory. We evaluate the proposed method with SPEC CPU 2006 benchmarks [12] which running on a hybrid environment which consist of gem5 [13] and NVMain [14]. Experimental results show performance improvements of up to 19% on workloads from SPEC CPU 2006 and an average of 17.39% on 7 memory intensive benchmarks from these suites and average energy saving is about 12.46%.

2 Background and Related Work

2.1 Pollute Region Isolation

The operating system can set the size of cache by using the method of page coloring. There are some common bits in the cache index and the physical page numbers. Making good use of the common bits, The page coloring method divides shared last level cache into some non-overlapping zones. The data in some pages that mapped into different color zones will help to decrease the source competition with each other.

2.2 LLC Management for the Hybrid Main Memory

The LLC management policy is a efficient and practicable way to control the number of read and write operations on different memory address spaces especially for the NVM. Zhou et al. [15] proposed the write-aware policy which can distribute the shared LLC to multiple cores by considering the write-back penalty. Wang et al. [16] They classify the LLC blocks into two class that is frequent and non-frequent write-back blocks, so that the dirty cache blocks in the LLC can be reused frequntly. The proposed policies in [17, 18] set different priority for each cache block based on block's state for example the clean block and dirty block. They prioritize clean blocks. Ferreia et al. [19] proposed a replacement policy for the "page cache" which is a small DRAM-based buffer between processor cache and PCM main memory. It tends to select clean pages as candidates in order to reduce number of PCM writes. In general, the polices which we mentioned above can't be directly applied to the cache management for hybrid memory architecture, cause they did thinking about the difference between the DRAM and NVM.

2.3 The Structure of Global Cache Pollute Buffer

The method of pollute region isolation chooses one color zone from last level cache to become the pollute buffer for putting the less used data. The aim is departing the reused data

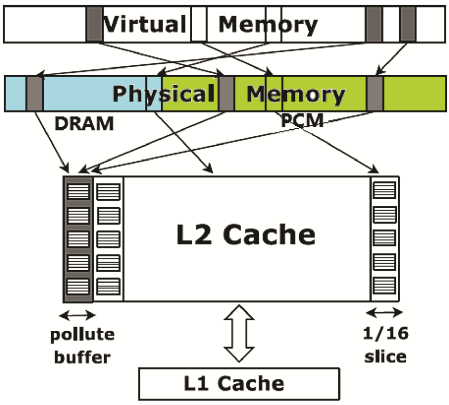


Fig. 1. The software pollute buffer

and less used data. The improvement of system performance is related to cache size for a given application. Because the cache size of a single process is not fixed in multiprocess environment, which is determined by other parameters. So we should check pollute regions of application on each cache size configuration for meeting the different demands of cache partition results. At last we can come out a best parameter of the cache size.

In our system, the pollute buffer is decided by the software-based cache partitioning system. We dedicating a single partition of the L2 to act as the pollute buffer which put the unreused data into it. Figure 1 illustrates the design of the pollute buffer using page-coloring and the hybrid memory share one pollute buffer.

2.4 Process-Based Pollute Region Isolation (PPRI)

In this paper, we exploit last level cache for identifying the weak locality data which is less reused. first we should part the region then find out the weak locality data. Our approach divide the virtual address space of a given process into several memory regions. The region consist of several continuous pages. We define two functions, *Cost(i)* and *Gain(i)*, which respectively represent incremental and decremental cache misses after remapping region i. Polluted sets of a process on a given cache size can be detected by comparing their *Cost* with *Gain*. The whole process of pollute regions detection algorithm show in Algorithm 1.

Algorithm 1. Pollute regions detection algorithm

```
1.  for i from 1 to N-W
2.  cache_size ← color_size*i
3.  j ← 16
4.  part the virtual address space into j regions
5.  execute trace-driven cache simulation
6.  PollutionSet ← Null
7.  put all regions into NormalSet
8.  rank regions in miss rate from high to low
9.  for each region k in NormalSet
10. calculate Cost (k)
11. calculate Gain (k)
12.  if Gain (k) > Cost (k) do
13. put k into PollutionSet
14. Delete k in NormalSet
15.   for each region t where t != k do
16.    for each circular sequence cseq in t do
17.     cseq.interferenceCounter[k] ← 0
18.    end for
19.   end for
20. Total_Gain ← Gain (k) - Cost (k)
21.  end if
22. end for
23. if best < Total_Gain
24. best ← Total_Gain
25. j ← j*2
26. goto Line 4
27. end if
28. cache_size[i].polluteRegionSet ← PollutionSet
29. end for
```

We suppose that the shared last level cache can be divided into N parts of color zones and the number of CPU cores is W. Therefore the color numbers change from 1 to N-W (Line 1). If the number of memory regions is too large, we cannot distinguish different regions with different cache behavior; otherwise the size is too small, it will be too expensive to make a partitioning operation. The most special thing is that the region number is a variable in our approch. The initial region number j is 16 (Line 3). For each process, we identify the pollute region in line 12 then we put it into PollutionSet in Line 13. From line 15 to line 19, our method decrease the effect of cache influence between pollute regions and normal regions. We double the value of j and repeat detecting until the value of Total_Gain no longer grows from Line 23 to Line 26. At last we can get the pollute region which was disparted from the other regions.

3 Improved Cache Replacement Policy

We apply the pollute buffer techniques by restricting cache unfriendly pages which is less reused to the pollute buffer, we eliminate competition between pages that pollute the cache and pages that benefit from caching. The cache miss rates was decline but for the hybrid main memory the traditional LRU replacement policy didn't suitable for it. Therefore, we propose a new write-back aware shared LLC replacement scheme ILRU for high performance and energy efficient hybrid main memory architecture.

3.1 Cache Behavior Analysis

In the field of high performance computing, cache replacement policy becomes a critical influence factor in high performance computer systems, for the increasing gap between the CPU and main memory. The widely-used Least-Recently Used (LRU) replacement policy is that the least recently used cache block occupies the lowest priority position, on the contrary, the newly visited block is inserted in the highest priority position (MRU). However, existing research shows that more than 65% of the blocks in LLC lead to no cache hits with LRU. Furthermore, because LRU is unaware of the asymmetric performance and energy cost between DRAM and PCM accesses, it will not be an optimal system design for hybrid main memory architecture, so we must revise the replacement policy to improve the cache hits ratio. In this work, we consider the following two design goals for LLC replacement policy with hybrid main memory architecture:

- The replacement policy must improve the cache hits, if the cache hits ratio is hight enough, the total main memory accesses will be minimized so that the energy will be saved and the main memory will be last long.
- The replacement policy should be aware of the characteristics of different memory in the hybrid main memory, for example DRAM and SPM have different cost, so we should take this into account.

The access to the LLC In a multi-level cache system can be classified by demand. For example, a CPU read/write request may results in misses in all upper-level caches. In another situation, the immediate upper-level cache evicts a dirty block that should be inserted into LLC. The hit miss of LLC leads to a main memory read which impede the execution of CPU; on the contrary, a write-back miss introduces a new dirty block in the LLC and triggers without memory reads. In general, demand misses have higher impact on the system performance due to the CPU-stalling memory read. In Fig. 2, we show the demand access behavior on LLC for a set of SPEC CPU 2006 benchmarks in the policy LRU. The results show that on average, 75.3% of all LLC demand hits across various benchmarks present to the dirty cache blocks. The result shows that the dirty cache blocks have much higher chance to be reused that very important information for us to make new cache replacement algorithms.

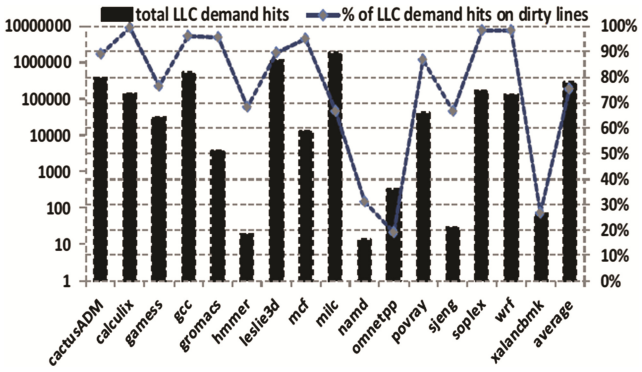


Fig. 2. The demand hit ratio of dirty cache blocks

In this work, we categorize cache blocks in LLC into 4 types: Dirty-PCM (DP), Dirty-DRAM (DD), Clean-PCM (CP), Clean-DRAM (CD), from the information above, we set the LLC priorities for different types of cache blocks as follows:

Dirty-PCM > Dirty-DRAM > Clean-PCM > Clean-DRAM.

Massive comparison experiments showed that dirty LLC blocks have higher chances to be re-referenced. As a result, we want to hold a Dirty-PCM data in LLC for longer time compared to other blocks, because the PCM read latency is longer than DRAM, if a PCM clean block is evicted from LLC and gets re-referenced again later, it leads to a higher performance cost than that of a DRAM clean block.

3.2 ILRU Scheme

There are three sub-policies about the LRU replacement policy: insertion, promotion, and eviction. We modified the insertion and promotion sub-policies of LRU to update the LRU method, for reflecting the different priority levels for various block types as we have defined before. For instance, a new cache block which just coming or a cache block hitting just now should be promoted according to its block type and the priority we defined, instead of being placed at the first position or the last position.

we used some definition and hypothesis in our proposed cache management scheme [20]. We provide each cache set with a n-bit saturating counter, where n = (log2A + 1) and A is lie on LLC. The initial value of the counter is 2n–1. We also assume that the associativity of the shared LLC is at least 10. We think the highest priority position (MRU) is at A–1 but we didn't often put some reused data to there, and an LLC eviction always chooses the LRU block at position 0 which is the lowest priority of the blocks.

Then we talk about the promotion part, if the hitting block C is a PCM block, it will be promoted to the highest position in the cache set. Otherwise if C is a DRAM block, it is promoted by (at most) A-DDP over its current position C:pos. If x lead to in a cache miss, we first evict the cache block at 0 position. If x is a write-back miss, a dirty block from upper-level cache (i.e., x:block) will to be inserted. The location of insertion lie on if it's a PCM block or DRAM block. Otherwise, x is a demand miss and x:block is a clean memory block to be read from main memory. It should be noted that the saturating counter gets decreased by 1 for a demand miss on PCM, and increased by 1 for a demand miss on DRAM. The saturating counter doesn't change with write-back misses because of that the write-back is passively which caused by inserting PCM or DRAM data in upper lever cache [21]. Overall, the more DRAM demand misses in the recent times, the more big value of the counter. Therefore, during insertion and promotion phase, the priorities of DRAM blocks should be raised reasonable and not the concrete number but a variable which will increasing more fast after hitting more than one times.

4 Evaluation

4.1 System Configuration

The experimental environment is a simulator platform which integrate the gem5 and NVMMain together for both DRAM and PCM. in the two level cache hierarchy, we use the write-back and write-allocation policy, and the classic noninclusive cache model in gem5. We use NVMain to construct the hybrid main memory modules, 128 MB DRAM and 1 GB NVM. The PCM module is based on the PCM prototype. There are a request queue in each memory module which supports the most of 64 requests at the same time.

4.2 Evaluation Results

The experimental results are as follows: Fig. 3 compares the performance of each replacement algorithms that we show in the figure include three method LRU, HAP and ILRU by using the attribute instructions per cycle (IPC). On average, ILRU improves performance by 16.46% compared with LRU, and 9.99% with compared to HAP. Our method is outstanding because of that compared with LRU, ILRU differentiate DRAM and PCM in insertion and promotion priority; and compared with HAP, ILRU further explored the disparity between dirty and clean cache blocks from both DRAM or PCM [22]. We improved the replacement method especially the promotion strategy which is more reasonable.

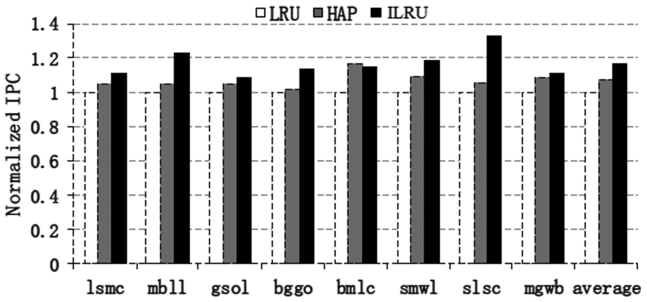


Fig. 3. Normalized IPC speedup.

Figure 4 shows the normalized write-back operations. The result in Figs. 3 and 4 shows that it will be have a significant performance improvements when the PCM write-backs reduced a lot. For example, in gsol, ILRU reduces PCM write-back by 18.91% when compared with LRU and improves the system IPC by 32.94%. The fact confirm that the impact of PCM write have a very big influence on system performance [23]. ILRU can effectively reduce the write-back requests because it sets relatively higher priority for dirty data than clean data, especially retaining PCM dirty data [24] in cache for a much long time. On average, ILRU reduces the PCM write-backs by 11.37% when compared with HAP and 18.61% when compared with LRU.

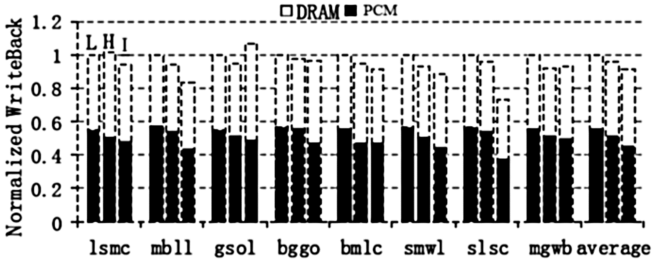


Fig. 4. Normalized write back. L = LRU, H = HAP, I = ILRU.

5 Conclusion

In this paper, we first propose a Process-based Pollute Region Isolation (PPRI) algorithm through introducing the concept of a pollute buffer to place cache blocks with little reused data before eviction which is a good way to eliminate competition between reusable and nonreusable cache lines. Then we proposed a new LLC replacement scheme ILRU to improve the last level cache hit ratio and reduce the LLC miss cost for the hybrid main memory architecture. ILRU dynamically assigns different priorities to cache blocks in LLC according to the potential cost. If a cache block will be evicted the replacement algorithm which we proposed can make a decision about which one will be swap-out according the priorities. At last we combine this two method together in

the hybrid main memory system and than A series of experiments were made. Experiment results show that PPRI+ILRU efficiently improve system performance (average 17.39%) as well as energy saving (average 12.46%).

Acknowledgments. This research is supported by the key Project of DEGP #2014GKCG031.

References

1. Ho, Y., Huang, G.M., Li, P.: Nonvolatile memristor memory: device characteristics and design implicatons. In: Proceedings of the 2009 International Conference on Computer-Aided Design, ICCAD 2009, pp. 485–490. ACM, New York (2009)
2. Qiu, M., Ming, Z., Li, J., Gai, K., Zong, Z.: Phase-change memory optimization for green cloud with genetic algorithm. IEEE Trans. Comput. **64**(12), 3528–3540 (2015)
3. Kang, D.-H., Lee, J.-H., Kong, J.H., Ha, D., Yu, J., Um, C.Y., Park, J.H., Yeung, F., Kim, J.H., Park, W.I., Jeon, Y.J., Lee, M.K., Song, Y.J., Oh, J.H., Jeong, G.T., Jeong, H.S.: Two-bit cell operation in diode-switch phase change memory cells with 90 nm technology. In: Proceedings of the 2008 Symposium on VLSI Technology, pp. 98–99 (2008)
4. Hay, A., Strauss, K., Sherwood, T., Loh, G.H., Burger, D.: Preventing PCM banks from seizing too much power. In: Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO-44 2011, pp. 186–195. ACM, New York (2011)
5. Cho, S., Lee, H.: Flip-N-Write: a simple deterministic technique to improve PRAM write performance, energy and endurance. In: Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture, MICRO-42, pp. 347–357. ACM, New York (2009)
6. Qureshi, M.K., Srinivasan, V., Rivers, J.A.: Scalable high performance main memory system using phase-change memory technology. ACM SIGARCH Comput. Archit. News **37**(3), 24–33 (2009)
7. Ramos, L.E., Gorbatov, E., Bianchini, R.: Page placement in hybrid memory systems. In: The International Conference on Supercomputing, Tucson (USA), pp. 85–95. ACM (2011)
8. Zhang, W., Li, T.: Exploring phase change memory and 3D die-stacking for power/thermal friendly, fast and durable memory architectures. In: International Conference on Parallel Architectures and Compilation Techniques, pp. 101–112. IEEE (2009)
9. Qiu, M., Chen, Z., Ming, Z., Qin, X., Niu, J.: Energy-aware data allocation with hybrid memory for mobile cloud systems. IEEE Syst. J. **11**(2), 1–10 (2014)
10. Wei, W., Jiang, D., Xiong, J., Chen, M.: HAP: hybrid-memory-aware partition in shared last-level cache. In: IEEE International Conference on Computer Design (ICCD), pp. 28–35. IEEE (2014)
11. Qiu, M., Ming, Z., Li, J., Liu, S., Wang, B., Lu, Z.: Three-phase time-aware energy minimization with DVFS and unrolling for chip multiprocessors. J. Syst. Archit. **58**(10), 439–445 (2012)
12. Henning, J.L.: SPEC CPU 2006 benchmark descriptions. ACM SIGARCH Comput. Archit. News **34**(4), 1–17 (2006)
13. Binkert, N., et al.: The gem5 simulator. ACM SIGARCH Comput. Archit. News **39**(2), 1–7 (2011)
14. Poremba, M., Xie, Y.: NVMain: an architectural-level main memory simulator for emerging non-volatile memories. In: IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 392–397. IEEE (2012)

15. Zhou, M., et al.: Writeback-aware partitioning and replacement for last-level caches in phase change main memory systems. ACM Trans. Archit. Code Optim. (TACO) **8**(4), 53 (2012)
16. Wang, Z., et al.: WADE: writeback-aware dynamic cache management for NVM-based main memory system. ACM Trans. Archit. Code Optim. (TACO) **10**(4), 51 (2013)
17. Rodríguez-Rodríguez, et al.: Write-aware replacement policies for PCM-based systems. The Comput. J. **58**(9), 2000–2005 (2014)
18. Zhang, X., et al.: A read-write aware replacement policy for phase change memory. In: Advanced Parallel Processing Technologies, pp. 31–45 (2011)
19. Ferreira, A.P., et al.: Increasing PCM main memory lifetime. In: The Conference on Design, Automation and Test in Europe, pp. 914–919 (2010)
20. Gai, K., Qiu, M., Zhao, H.: Cost-aware multimedia data allocation for heterogeneous memory using genetic algorithm in cloud computing. IEEE Trans. Cloud Comput. **PP**(99), 1 (2016)
21. Gai, K.K., Qiu, M.K., Zhao, H., Qiu, L.F.: Smart energy-aware data allocation for heterogeneous memory. In: IEEE International Conference on High Performance Computing and Communications (HPCC), pp. 136–143. IEEE (2016)
22. Gai, K., Qiu, M., Zhao, H.: Energy-aware task assignment for mobile cyber-enabled applications in heterogeneous cloud computing. J. Parallel Distrib. Comput. **111**, 126–135 (2018)
23. Gai, K., Qiu, M., Zhao, H., Tao, L., Zong, Z.: Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. J. Netw. Comput. Appl. **59**, 46–54 (2016)
24. Gai, K., Qiu, M., Sun, X.: A survey on FinTech. J. Netw. Comput. Appl. **PP**, 1 (2017)

Quality of Service (QoS) in Lan-To-Lan Environments Through Modification of Packages

Cesar Andrés Hernández[1], Gabriel Felipe Diaz[1], and Octavio José Salcedo Parra[1,2(✉)]

[1] Faculty of Engineering, Universidad Distrital "Francisco José de Caldas", Bogotá D.C., Colombia
cesar_ro11@hotmail.com, gabo.fdc@gmail.com, osalcedo@udistrital.edu.co

[2] Faculty of Engineering, Universidad Nacional de Colombia, Bogotá D.C., Colombia
ojsalcedop@unal.edu.co

Abstract. In this paper we are going to show the ability to have the libraries for python Scapy and Net Filter Queue (nfqueue), initially to capture packets from a video call on Skype, then modifying the values contained in the ToS field, evidenced where a high priority is set, and finally replacing the original package. Tests are done taking the congested channel where it is found that when changing the ToS field, both qualitative and quantitative improvement is presented in the call. Finding that by modifying the ToS field of the call packets can obtain an improvement between 5.87% and 23.5% for voice, and 9.18% and 27.55%.

Keywords: QoS · Quality of service · Bandwidth · Python · Scapy · nfqueue

1 Introduction

In the fields of packet switching networks and computer networking, the term Quality of Service traffic engineering, QoS abbreviated, refers to resources Reserve control mechanisms. Quality of service can provide different priority levels to different users or data flows, or guarantee a certain level of performance to a data flow in accordance with requests from the application program or policy provider Internet. You guarantee that the quality of service are important if the network capacity is limited, for example, in mobile communication data, especially for real - time streaming multimedia applications, such as voice over IP and television intellectual property, since these often require fixed bit rate and are sensitive to delays [1].

In the field of scapy networks is used for multiple uses where required to inspect the network, evaluating behaviors, activity in it or make changes in it. Using other tools can even be useful to verify and enhance security [2]. Also check for vulnerabilities or network restrictions [3].

Additionally, it is common to find that for networks with low bandwidth or congested, video calling applications such as Skype do not achieve adequate performance and often user gives up use. However by implementing quality of service these applications could improve the performance of these and thus allow the users continue to use regardless of network conditions.

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 154–162, 2018.
https://doi.org/10.1007/978-3-319-73830-7_16

2 Background

An alternative to implement quality of service is through the use of a model of queuing, and providing multiple differentiated services via the Internet is a very difficult problem, the strict priority queue, which is available on many products router (for example, Cisco routers is available under the label LLQ (Low Latency Queueing), offers some improvements to traffic management. a priority queue mechanism the ability to sort packets based on priority differences adds and insert them into the internal queues separated or shuffle insertion into a single queue. The forwarding algorithm always transmits packets by highest priority. If no packets highest priority level, the next queue of highest priority is served, and so on [7].

Quality of Service (QoS) has become indispensable in today's networks. Most existing quality solutions are implemented in the service layer 3 (network layer). In order to provide end to end QoS guarantees for these networks [8].

Differentiated service classifies and prioritizes traffic packages. DiffServ does not guarantee that all packets reach their destination. Marks packets with a priority level, giving the package resources available according to the priority [9].

DiffServ is currently being used daily by a large number of vendors and companies in the current market, so it is a good choice for implementing QoS on a network. Differentiated Services work by using a value in the type of service bytes (IPv4) or byte traffic class (IPv6) in an IP header, which has now redefined as byte DiffServ (VS-byte) by RFC 2474. the quality of service can be implemented in several ways, and has not been established "best way" yet to serve as an industry standard. However, as the need for QoS grows and more vendors continue to offer quality service, the real possibilities of quality end to end service continually increase [9].

Implementing QoS allows the ISP to control the amount of bandwidth that each customer is using and allows the ISP to introduce the concept of "value" in their billing systems, charging more for greater prioritization or bandwidth. QoS implementation also avoids using more bandwidth to ISPs what they are actually paying, degrading network performance for other users customers [10].

By comparing several VoIP applications, the best application in terms of quality of service was Skype; However, he got only the second place in the ranking QoE. Similarly, Dhamaka showed good results in the evaluation of quality of service, but was rated as the worst of application in terms of quality of experience [10].

DiffServ refers to the classification of packets as they enter the local network. This classification applies to traffic flow in a flow is defined by five elements; IP source address, destination IP, source port, destination port and transport protocol. A flow that has been classified or marked can then act on by other QoS mechanisms. Thus, multiple streams can be treated in a multitude of ways, depending on the requirements of each flow. Packages are first classified according to their current DSCP. Then they split in queues where a queue can be routed via a dial mechanism and other tail can be examined more closely. After further examination additional packages can be sent to mark or sent directly to the creation/dropping mechanisms where all packages finish before leaving the interface.

The IP header has a field called the Type of Service (TOS) located between the Header Length field and the total length field. (See IP datagram TOS field for a view of the type of service field in the IP header.) Traditionally, IP precedence used the first three bits of the TOS field to give 8 possible values of precedence.

000 (0) - Routine 100 (4) - Flash Override
001 (1) - Priority 101 (5) - Critical
010 (2) - Immediate 110 (6) - Internetwork Control
011 (3) - Flash 111 (7) - Network Control

DiffServ introduces the concept of point DiffServ (DSCP) code that uses the first 6 bits of the TOS field thus giving 26 = 64 different values. RFC 2474 describes the Differentiated Services Field (DS) and DiffServ code point (DSCP).

DiffServ each router handles each package differently. The concept of Per-Hop Forwarding Behavior (PHB) is introduced where classes are developed such as business, Telecommuter, Residential etc. that can be offered by an ISP as different levels of service. A Per-Hop Behaviour is effectively a mode of transmission of a particular flow or group of flows (Behavior Aggregate) traffic in a DiffServ node. A flow or flows of packets marked with a DSCP particularly in the DS field will be subject to a particular method of forwarding and rules as encapsulated in the Behavior Aggregate. This addition has three elements to it (or three colors) that determine whether the router interface (1) drops datagram, (2) Sends datagrams or (3) it is classified. This three - color marker in RFC 2697. detailed For example 5 flows can be treated as an aggregate behavior so they are similarly treated as a group in most aspects. Each stream is then characterized by a probability of further decline and forwarding behavior. Note that as the drop preference value increases, so increases the probability of being abandoned.

Set the layer three DSCP field requires a little more foresight. When we talk about DSCP values, we're usually refers only to the first six bits of a field of eight bits. This allows a range of decimal value from 0 to 63. For example, you probably recognize 46 as the DSCP Expedited Forwarding (EF) associated with the voice and other traffic in real time.

Scapy, however, requires us to provide a value of eight bits to set the field in our package. How do we become our decimal DSCP 46 (EF) to a value of eight bits? Is 46 in binary, add two zeros to the right, and convert it back to decimal.

46 = 101110
10111000 = 184

Therefore if we set DSCP EF Expedited Forwarding in a Scapy package:

=184 Tos

There are two values of quality of service we need to use QoS: IEEE 802.1Q priority identification (class of service or CoS) and differentiated point field IP control services (DSCP) or IP Precedence. Both have a default value of zero.

3 Methodology

The following tools were used for modifying packages call:

Scapy is a Python module for making packages and manipulation tool computer networks, written in Python by Philippe Biondi. Scapy was used during experiments to edit packets and to assign a desired quality of service within the TOS field of the IP datagram [4].

Netfilter is part of packet filtering framework of the Linux kernel 2.4.x, 2.6.x and later. As indicated by the netfilter home page: "Netfilter provides a set of tools within the Linux kernel that allows kernel modules to register functions with the network queue. The registered function is called for each packet traversing the network queue".

NF_QUEUE extends this ability to user space, allowing the packets are redirected using iptables rules to a program in userspace. The program can then look at the package and take action based on the contents of the package. The program has the option to decide whether to accept or reject the package. The program can also decide whether to modify the package and return it to netfilter for further processing [5]. It is this last skill which allows us to modify packets on the network to implement them appropriate protocols for quality of service.

Hping is a network tool able to send TCP/IP packets and display custom responses, similar to how the program sends ICMP ping packets. Hping3 can handle fragmentation, body and random packet sizes [11].

3.1 Description of the Context

Experiments designed to experimentally analyze the QoS provided by the network communications with Skype, were developed as follows: two personal computers called PC1 and PC2 both were installed Skype software [6] were used, on a system Linux - based operating. Additionally scapy packages are installed and nfqueue for python which allow us to make the manipulation of packages. The teams are in a DMZ within the LAN, allowing communication between the two is directly by the public respective IPs.

3.2 Prerequisites

The first thing to establish is the DMZ, this is done through the firewall or router AC gives LAN, which designate the IP address of the machine that will be in the DMZ.

3.3 Initial Tests

Then we tried a basic case, in which a single packet is sent by a specific port, and is found to be received. By using scapy build and ship the package PC1, indicating the destination IP PC2 (See Fig. 1):

```
>>> p=IP(dst="186.154.190.27")/UDP(dport=3030)
>>> p.show()
###[ IP ]###
  version= 4
  ihl= None
  tos= 0x0
  len= None
  id= 1
  flags=
  frag= 0
  ttl= 64
  proto= udp
  chksum= None
  src= 192.168.1.105
  dst= 186.154.190.27
  \options\
###[ UDP ]###
     sport= domain
     dport= 3030
     len= None
     chksum= None
>>> send(p)
.
Sent 1 packets.
>>> send(p)
```

Fig. 1. Test 1: Sending the package on PC1. Source: Authors

At the same time, in the PC2 network sniffer you have active, using the same scapy, with the appropriate filter to capture only the packets coming from the PC1.

It is evident that the package is correctly received at PC2. (See Fig. 2).

```
>>>
>>> p=sniff(filter="src 186.83.16.64", count=5)
>>> p[2]
<Ether  dst=08:00:27:87:df:13 src=00:26:b6:00:d3:b3 type=0x800 |<IP  version=4L ihl=5L tos=0x0 len=28 id=1 flags= frag=0L ttl=55 proto=udp chksum=0xf889 src=186.83.16.6
4 dst=192.168.0.11 options=[] |<UDP  sport=domain dport=arepa_cas len=8 chksum=0x688c |<Padding  load='\x00\x00\x00\x00\x00\x00\x00\x00<15\x00<\x80\x104\x97' |>>>>
>>>
>>>
```

Fig. 2. Acknowledgment of receipt in the PC2. Source: Authors

By having satisfactory results, we continue with the tests, this time sending a slightly more complex package. This package will have the precedence field 001 (priority) and other fields default.

As shown in the picture, the TOS field would be 00100000, which is equivalent to 32 decimal, and hexadecimal 20. (See Fig. 3).

```
Para la precedencia IP se maneja el campo TOS compuesto de 8 bits de la siguiente manera:
|Precedence|Delay|Throughput|Reliability|Cost| MBZ |
|   3 bits  |1 bit|  1 bit   |   1 bit   |1bit| 1bit|
las posibles alternativas para precedence son:
000 - Rutina
001 - Prioridad
010 - Inmediata
011 - Flash
100 - Anulacion de Flash
101 - Critico
110 - Control Internetwork
111 - Control de Red
ingrese estos campos para la construccion de su paquete:
Precedence: 001
Delay: 0
throughput: 0
Realiability: 0
Cost: 0
MBZ: 0
campo TOS: 00100000 Decimal: 32
```

Fig. 3. Modifications for the Test 2. Source: Authors.

We proceed to the creation and delivery of the package with this TOS, from the PC1 to the PC2 (See Fig. 4).

```
>>> p=IP(dst="186.154.190.27",tos=32)/UDP(dport=3030)
>>> p.show()
###[ IP ]###
  version= 4
  ihl= None
  tos= 0x20
  len= None
  id= 1
  flags=
  frag= 0
  ttl= 64
  proto= udp
  chksum= None
  src= 192.168.1.105
  dst= 186.154.190.27
  \options\
###[ UDP ]###
     sport= domain
     dport= 3030
     len= None
     chksum= None
>>> send(p)
.
Sent 1 packets.
```

Fig. 4. Test 2: Sending the package on PC1 with TOS. Source: Authors

We checked the sniffer destination in the PC2 (See Fig. 5).

```
>>> p=sniff(filter="src 186.83.16.64", count=8)
21:43:39.882208 IP Dynamic-IP-186831664.cable.net.co.domain > 192.168.0.11.arepa-cas: [|domain]


21:43:46.401765 IP Dynamic-IP-186831664.cable.net.co.domain > 192.168.0.11.arepa-cas: [|domain]
21:43:46.904401 IP Dynamic-IP-186831664.cable.net.co.domain > 192.168.0.11.arepa-cas: [|domain]
21:43:47.283296 IP Dynamic-IP-186831664.cable.net.co.domain > 192.168.0.11.arepa-cas: [|domain]
21:43:47.861600 IP Dynamic-IP-186831664.cable.net.co.domain > 192.168.0.11.arepa-cas: [|domain]
21:43:51.137945 IP Dynamic-IP-186831664.cable.net.co.domain > 192.168.0.11.arepa-cas: [|domain]
>>>
>>>
>>> 21:43:51.672461 IP Dynamic-IP-186831664.cable.net.co.domain > 192.168.0.11.arepa-cas: [|domain]


>>> p
<Sniffed: TCP:2 UDP:6 ICMP:0 Other:0>
>>> p[2]
<Ether  dst=58:00:27:87:df:13 src=00:26:b6:00:d3:b3 type=0x800 |<IP  version=4L ihl=5L tos=0x20 len=28 id=1 flags= frag=0L ttl=55 proto=udp chksum=0xf809 src=186.83.16.
64 dst=192.168.0.11 options=[] |<UDP  sport=domain dport=arepa_cas len=8 chksum=0x880c |<Padding  load='\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x01\x2c' |>
>>>
>>>
```

Fig. 5. Revision of Sniffer in PC2. Source: Authors.

It is evident that the packet was received and maintains the TOS field unchanged.

4 Development

Since we have successfully verified that packets with ToS arrive correctly at the destination, we replicate the same behavior for packets of a call on skype.

For this, a script that automates this task was developed, then the actions described running the script:

1. Create a queue NFQueue.
2. Create the iptables in the operating system, so that the paquet is bound to PC2 re directed to the queue created.
   a. For each packet reaches the queue created the following runs:
   b. The package is taken and the TOS field is changed to a predefined value.
   c. Due to the modification or n len packet and package chksum fields are recalculated, so that problem free.
3. The modified package, which follows its normal course accepted.
4. When the script is stopped, cleaned iptables rule created and tail NF_Queue is terminated.

Once ready the script started a common call skype and sniff packets of this call is made, it is that these packages have the ToS field to 0.

5 Results of the Experiment

Having obtained favorable qualitative results, measurements are needed to corroborate the information and we can give a quantitative result, so that in this way can indicate clearly and objectively the improvement appears to put the TOS field in the packet call skype.

For this time measurements of departure and arrival of the packages under different conditions of the call is made, and calculate the average time it took to get the package. These results are summarized in the Table 1.

Table 1. Average arrival time of packages

Service/quality	TOS 0	TOS 0x20	TOS 0x40
Number nodes	0.455731	0.453757	0.458858
Number of links	0.485541	0.489371	0.482481
Number of demands	0.504174	0.474554	0.385693
Number segments with protection	0.563675	0.511911	0.408381

6 Discussion of Results

In this paper it was shown that you can apply quality of service through modifying the TOS field in the IP datagram. Furthermore it not sets a standard for putting on the market the quality of service so there are many ways to implement it [9].

During the experiment, there was a lot of packet loss due to an attack. DiffServ does not guarantee that all packets arrive to their destination, in this technique packets with a priority level marked, and is given the available resources according to their priority. At present there is an IETF working group is developing a standard for implementing DiffServ and RFC that are currently available [9].

Having congestion of a channel, you must set the priority in the router for better performance packages are marked. The specific flow header the packet delivery by routers regarding other requests that the router receives. In other words, the flow request header set preferential treatment in the router. The priority allows a host to prioritize the packets it sends. Together, these features add greater control over delivery [9].

Although during attacks many packets are lost, this did not significantly affect the user experience during the call in Skype, which is not neither could find the time difference from packet to reach the destination because they had lost, i.e. as if it had assessed the user experience would have achieved better results. In this other job parameters obtained from service quality video calls made during the tests was analyzed QoE. The delay between packets, consumption of bandwidth, and packet loss were calculated for the four VoIP applications. We have observed that the quality of service results do not exactly match those of QoE. The best application in terms of quality of service was Skype; However, he got only the second place in the ranking QoE [10].

7 Conclusions

- It is possible the implementation of quality of service for Skype calls using external tools that allow us to modify the TOS field of the IP datagram, prioritizing packets and achieving improved quality of service, l which s note significantly or what if the channel is congested to.
- For a channel without congestion, the modification or the TOS field represents significant changes in the quality of service
- In a congested channel, modify the TOS field improves the quality of service. TOS = 0x20 using an equivalent to “Priority” an improvement of 5.87% for voice, and 9.18% for video is obtained. TOS = 0x40 using an equivalent to “Immediate” an improvement of 23.5% for voice and 27.55% for video is obtained.
- Not all service types behave similarly, some have better effectiveness in the response generated than others.
- Although it was shown or that the use of this technique achieves to improved call, it is important to note that to achieve two optimal and consistent results should be combined other techniques that complement this.
- A future work can be carried out tests to identify what techniques combine efficiently with this, to achieve high service of quality.
- Skype, changing service either way call or any other, packets should not be being altered or call generates an error and quits

References

1. Rubio, S.M., de Luque Prieto, J.J., García, J.A.C.: Análisis de la transmisión de datos multimedia mediante el protocolo IPTV, Bucaramanga (2011). http://repositorio.uis.edu.co/jspui/bitstream/123456789/2762/2/139128.pdf
2. Alcock, S., Lorier, P., Nelson, R.: Libtrace: A Packet Capture and Analysis Library
3. Basam, D., Marchany, R., Tront, J.G.: Attention: Moving Target Defense Networks. How well are you moving. Bradley Department of Electrical and Computer Engineering Virginia Tech, Blacksburg
4. Biondi, P.: Scapy (2011). http://www.secdev.org/projects/scapy
5. Amaranth, P.: Fixing broken protocols with NF_QUEUE. Linux J. Arch. **2011**(212), 7 (2011)
6. Skype. Visto el 4 de noviembre de (2015). http://www.skype.com
7. Schmitt, J., Zdarsky, F.: A case for simplicity in providing network quality of service: class-based strict priority queueing
8. Bouras, C., Kapoulas, V., Primpa, D., Papapanagiotou, V., Stamos, K., Poulopoulos, L.: Extending QoS support from Layer 3 to Layer 2
9. Wood, S., Chatterje, S.: Network Quality of Service for the Enterprise
10. Cano, M.-D., Cerdan, F.: Subjective QoE analysis of VoIP applications in a wireless campus environment
11. Hping3. Linux man page: http://linux.die.net/man/8/hping3

Heuristic Algorithm for Flexible Optical Networks OTN

Diego Fernando Aguirre Moreno[1], Octavio José Salcedo Parra[1,2(✉)], and Danilo Alfonso López Sarmiento[1]

[1] Internet Inteligente Research Group - Universidad Distrital "Francisco José de Caldas", Bogotá D.C., Colombia
dfaguirrem@correo.udistrital.edu.co, ojsalcedop@unal.edu.co, {dalopezs,osalcedo}@udistrital.edu.co
[2] Universidad Nacional de Colombia, Bogotá D.C., Colombia

Abstract. This paper presents the concept of flexible optical networks, based mainly on routing tasks and wavelength assignment ("λ") by means of a heuristic algorithm called EEM, obtaining the calculation of the optical paths according to the demands of the topologies (NSFnet and EON) with the configuration of variable transponders and 100% traffic protection via disjoint channels. With the Net2plan software, the simulations of the topologies were realized, observing the increase of 20%–80% of the traffic offered by the Network (in Tbps) before observing an increase in the blocking probability according to the routes assigned by the algorithm.

Keywords: Artificial optical networks · Optical Transport Network (OTN)
Wavelength Division Multiplexing (WDM)
Dense Wavelength Division Multiplexing (DWDM)
Dynamic Routing and Wavelength Assignment (DRWA)
Frequency Slot Unit (FSU) · Wavelength-Routed Optical Network (WRON)

1 Introduction

In the last few years optic networks and the evolution of optical switches have solved the challenge of establishing broadband services in real time such as videoconferences, SAN services, etc., seeking to take advantage of all the benefits of an elastic optical network [1]. Among the different issues in optical wavelength division multiplexing networks, the problem of wavelength assignment and routing (RWA) is crucial for the efficient operation of the network, which are known as NP-complete [2].

The first developments shows the lack of wavelength conversion for the solution of the blocking probability, Due to this most of the effort was focused on efficient routing design and wavelength selection algorithms to minimize this impact. With the capacity of wavelength conversion to 10%–20% of nodes [3, 4], and the heuristics to wavelength convertibles OXC (optical cross-connects) assignment also where proposed in [5, 6], the gain obtained was generally modest, given an apparently advantage of cost-efficiency and all the optics solutions. In order to increase the advantage obtained by the evolution of the wave conversion, it is important to study the routing and optical spectrum allocation algorithms.

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 163–172, 2018.
https://doi.org/10.1007/978-3-319-73830-7_17

DRAW is known as "Dynamic Routing and Wavelength Assignment" to the problem involving routing and the dynamically assignment of wavelengths. Helps the optic network through mechanism to stablish optic paths between destination and origin of each demand for all traffic added to a topology. These mechanisms must be controlled by some management element.

The classic DRWA problem has been extensively studied. Gerstel and Kutten [7] proposes classical techniques separating the routing problem. This technique is justified by the following reasons:

- Fault tolerance for disjoint routes to support.
- Restrictions on propagation delays solved by selecting routes with less delay through the ring.
- Computational efficient solutions to the combined problem of assigning routing and wavelengths that allocate resources optimally.

Real-time RWA algorithms Zang et al. [8] summarizes classical and heuristic techniques of both routing and wavelength assignment in Wavelength-Routed Optical Network (WRON).

Second-generation Optic Networks also called Wavelength Routing Networks or Optic Circuits Switching Networks [9], have additional functions to point-to-point transmission in the optic domain. They are based in complete optical add-drop multiplexors (OADM), optic line terminal (OLT), and optical cross connect (OXC) [10].

2 Background

For the optimization of the optical networks it is necessary to have a fully optic flexible technology. Data rate paths on wavelengths are configured with flexible granularity and spectrum switching, making use of data rate transponders with variable bandwidth and optical cross-connectors (OXCs). In this document we used the heuristic algorithm on three topologies. In each topology the traffic demands and the capacity of the transponders were configured; at the end of it the results obtained are presented with the proposed proposal.

2.1 Optic Trajectory

Now days OTN networks operates with MSP or SNCP protections where bandwidth and optic spectrum are wasted. For the establishment of optical protection paths, the approach followed by almost all studies, breaks the routing and assignment of spectrum in two independent steps [11–13]. Using the most cut-off K-path algorithm, they first calculate several off-line paths that can be used to establish the protective light path. Then, from these routes, they choose the one that has the best shared spectrum to establish optical path protection. Although this simplifies the process of establishing the protection light path, by breaking the routing and spectrum allocation steps, it is not possible to achieve a maximum spectrum distribution between the protective light paths.

2.2 Types of Algorithms

Elastic optical networks are a promising technology for future on-demand optical bandwidth services. Heterogeneous means that optical connection requests for different services requiring different data rates would coexist in the network. In this context, Routing and Spectrum Allocation algorithms face the challenge of making a fair allocation of resources, providing similar blocking performance to all services.

Heuristic algorithms classifies in two categories, depending on how they approach routing and spectrum allocation: jointly or separately. We use the term one-step algorithm for the schemes that solves both sub-problems simultaneously. There are two approaches: greedy algorithms [14] and auxiliary graphs [15], the first ones achieve a sub optimum solution in a fast way.

Two-steps algorithms decomposes the problem in to two sub-problems. In first place, the algorithm finds a pair of candidate paths. Routing calculation can be static or dynamic, depending on whether the routes are pre-calculated or may vary according to the state of the network. RSA algorithm tries to assign the request, starting at the top of the candidate paths. Some proposals stop when a feasible assignment is found, while others evaluate each, and obtain the one that is considered better.

2.3 EEM Algorithm

EEM (Routing, Spectrum and Modulation) Algorithm is designed with the objective of providing a complete, effective and realistic solution to the RWA problem, by means of heuristic algorithm. By means of the algorithm the tasks R and WA are taken care of separately, guaranteeing the operation of the photonic mesh in real time. According to the previous chapter, the proposed Heuristic algorithm is two steps solving the problem of routing, spectrum and modulation assignment in Optical network.

– Algorithm EEM (Routing, Spectrum and Modulation)
  1. Initialization (Demands, Nodes, Links, Line Capacity, Link Capacity)
  2. Repeat
  3. Generate lists of paths
  4. Generate lists of Disjoints
  5. WDM calculates the regenerators
  6. is the main path according to metrics
  7. is the secondary path according to metrics
  8. Until a new demand, nodes, links, lambdas be added.

The algorithm is based on a routing heuristic that is subject to a series of non-trivial constraints independently of what is considered in the cost function. For example, a constraint could be the total path delay. Another restriction could be the inclusion or not of certain nodes, this type of problems are NP-complete.

2.3.1 Blocking Probability

The proposed algorithm attacks the blocking probability by taking the sub-problems (routing and wavelength assignment) as a whole (elastic optical networks).

Probability of link lock: The probability of blocking Be is the same for all demands and is given by the formula of Erlang-B.

Blocking probability Be:

$$Be = E_b\left[Y_e, U_e\right] = \frac{\frac{Y_e}{U_e!}}{\sum_{k=0}^{U_e} \frac{Y_e^k}{k!}} \tag{1}$$

It is the totality of traffic offered in the link.

= Is the link capacity.

Demand and probability of network blocking: the demand on the blocking probability Bd provides the probability of rejecting a connection of a given demand, while the blocking average of the network B is weighted as the average of a fraction of traffic rejected by the network.

Network blocking probability:

$$B = \frac{1}{\sum_d hd} \sum_d hd\ Bd \tag{2}$$

2.3.2 Packet Lost Probability

To reduce the packets lost in optic networks, modulation systems must be modified or added because in conventional optical systems of a single carrier, as the transmission speed increases, the accuracy of the timing sampling becomes more and more critical. Excessive time jumps would set the sampling point far from optimal, causing serious error. On the other hand, for elastic optical networks (OFDM), time sampling with such precision is not necessary. While an appropriate "window" of sampling points containing an uncontaminated OFDM symbol is selected, it's enough to eliminate interference between symbols (ISI).

2.3.3 Delay

In circuit-switched networks the randomness of the origins reflects the speed in the packet flow observed in the links. This is the origin of the delay in storage in the nodes and discard of packages when the buffer begins to fill. If the traffic is multicast and the path is multicast tree routing the end-to-end delay may be different for different destinations nɛb (p).

$$
\begin{aligned}
T_e &= Delay\ link \\
Tp &= \sum_{e \in p} Te \\
T_p &= \max_{n \in b(p)} \left\{ \sum\nolimits_{e \in \rho_n} Te \right\}
\end{aligned}
\tag{3}
$$

2.3.4 Network Resistance

It is a term that describes the network's ability to provide and accept service levels, even in the presence of faults or attacks. This is a critic aspect in network designs. Service Level Agreements (SLA) for transport networks are 99.9% and 99.999% approximation is 5 min per year. The availability id given by the Eq. 4:

$$
\begin{aligned}
&\text{Availability:} \\
&A = MTBF - MTTRMTBF
\end{aligned}
\tag{4}
$$

MTBF is the average time between failures.

MTTR is the average time in repairs.

In the design of the EEM algorithm is implemented the technique of disjoints channels for protection of the traffic according to:

Protection Recovery: in this case the recovery is pre-planned and pre-signalized in the network where the reaction to failures is very fast. In type 1 + 1 protection the traffic is connected through the primary path ρ and the protection by the secondary, the origin sends two copies of traffic, one for each path and in case of a failure in the main path the receiver is reconfigured to receive the traffic coming from the backup path.

M(backups) : N(Main)

3 Simulations

3.1 Net2Plan Simulator

Net2Plan is an open source Java-based software available for downloading from its website [16]. It is available under the GNU General Public License (GPL). Net2Plan has its origins in September 2011 as a resource for the network planning courses of the Universidad Politécnica de Cartagena. In 2013 it was used during more than 50 h of laboratory work in two graduation courses of more than 150 students.

Net2Plan is designed with the aim of overcoming the barriers imposed by existing network planning tools for two main reasons: users are not limited to running embedded algorithms, they can integrate their own algorithms, applicable to any network instance, such as Java Run classes. Net2Plan defines a network representation, the so-called network plan, based on abstract concepts such as nodes, links, traffic demands, routes, protection segments, risk groups and network layers. Information regarding technologies can be introduced through user-defined attributes connected to nodes, links, and so on, in the network plan. The combination of an independent part of the technology and

the attributes related to it provides the flexibility required to model any network technology within Net2Plan.

3.2 Topologies

The algorithm simulations were performed on the following three topologies:

1. NSFNET (National Science Foundation Network), 14 nodes and 42 links.
2. Atlanta Network, 15 nodes and 44 links.
3. European Optical Network (EON), 18 nodes and 66 links.

3.3 Environment

The control plain executed by the RSM algorithm has been tested in simulations on topologies. For these simulations have been installed ROADM node entity with SLICE technology simulating in each of the bi-directional nodes used for that purpose (Table 1).

Table 1. OTN layer information

	Atlanta network	NSFNet	EON
Number nodes	15	14	18
Number of links	44	42	66
Number of demands	210	182	306
Number of demands multicast	15	14	18
Number of routes	224	486	306
Number multicast trees	15	14	18
Number segments with protection	224	486	306

At the spectrum level, a Flexible OTN network is used with SLICE with traffic granularity and with FSU carriers with values of 12.5 GHz, 25 GHz and 50 GHz. Traffic is protected at OTN level 1 + 1 by doubling traffic, from a demand from a source to a destination. This protection is at the node level and both routes are completely disjoint which it means that they do not share any link or any node.

This protection is 100% of the traffic. Table 2 describes the process of the simulations performed for the three topologies.

- Upon receiving the response, the node is in charge of sending the messages of the RSM algorithm to make the reservation of resources in the network by means of First-Fit.
- If the route has been set correctly, the node receives the response message from all nodes in the network to signal that the route has been set correctly.
- The simulated nodes can process requests for OTN Flexgrid routes, thanks to the fact that libraries have been implemented to choose the bandwidth that a channel occupies in a flexible way based on the RSM algorithm.

Table 2. Simulations

Simulations						
Parameter	First	Second	Third	Fourth	Fifth	First
K	5	5	5	5	5	5
Number of wavelengths per fiber	80	80	80	80	80	80
Traffic scale	×2	×2	×2	×2	×2	×2

4 Results

In Fig. 1, it can be seen that by doubling the traffic offered in the three topologies, the number of optical paths increases between 10% and 20%, having less blocking probabilities, having a lower probability of blocking in offered traffic capacity of more than 10 Tbps. With the proposed RSM heuristic algorithm and the OTN network topology with the second path as protection, it is possible to handle high flows of unicast and multicast traffic improving network performance by 60% and resistance level by 100% (all traffic is protected).

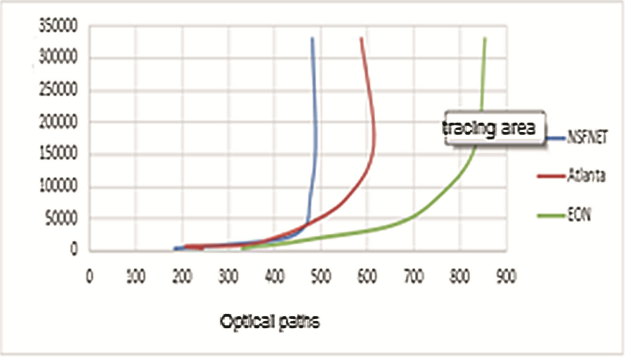


Fig. 1. Optic paths vs. Offered traffic

The matrix associated with a 100% load is obtained by calculating the maximum version of the matrix queue that has the shortest possible route (number of hops or distance in Km) of routing through the network. Different load factors were tested in each topology ranging from low traffic blocking situation where it is insignificant to a high traffic situation where blocking is unacceptable. Our metric of interest is the blocking probability (BP) in bandwidth, which is calculated with the sum of the blocking probabilities for each observed light path of each line type of 12.5, 25, 50, 75 Gbps, weighted by its rate; the ratio of the total amount of Gbps blocked to the total amount of Gbps offered to the network. In each test, the connection requests were generated for a maximum number of 860 optical paths with an average delay in the network of 9.69e+03.

5 Analysis of Results

In the blocking model, connection requests follow a Poisson process with an average rate of λ and the holding time after the negative exponential distribution, with a mean time equal to one unit of time. Time between arrivals (1/lambda) are the same for all line rates within each pair of source and destination nodes, and this value is adjusted so that the average amount of traffic offered matches the values given by the traffic matrix simulation input M.

By increasing the maximum number of paths supported at a time by a pair of input-output nodes k, the blocking probability and the amount of lambdas to perform routing remains stable (due to the modulation used), in this case is assume the amount of lambdas in 80 due to the current developments of OTN over DWDM of equipment suppliers Huawei, Alcatel, Ciena and ZTE.

The results show that at the end of the study the demand is the same for the three case studies. It is also possible to observe the same number of connections in the first two cases.

It can be observed that in the EON Topology having more ROADM nodes and links, it manages more traffic without presenting probability of Blocking (Offered traffic = 20 Tbps – BP = 0, 04). When designing a topology with MESH, the traffic management with the RSM algorithm will increase with respect to EON, maintaining a blocking probability of less than 2%; these types of solutions are expensive because of the deployment of fiber.

A large difference was observed compared to a study conducted in [17] where the routing is done from IP. Both the traffic carried and the number of wavelengths used are smaller and therefore the probability of blocking is greater. The traffic supported by the RSM algorithm is twice the traffic of the previous reference, since the maximum offered traffic is 4000 Gbps with a maximum of 350 optical paths. This demonstrates the robustness of transparent optical networks.

6 Conclusions

In order to achieve quality of service in optical networks, it is necessary to study the classical RWA optical routing problem as well as the strategies for R and WA, which considers a favorable development of the established optical paths.

The DR routing has a high impact on the blocking probability compared to the wavelength assignment WA, having fibers of 80 channels or higher depending on the modulation technique. Therefore when applying DR to a topology, better results are reflected in the level of traffic engineering.

The cost function of the R routing algorithm is adapted to the network topology and specific transmission characteristics of the topologies. Adaptive strategies allow the inclusion of link loading criteria in the cost function (No Jump, Distance in KM, Delay, etc.). This is very convenient since it avoids the congestion of the links and as a consequence the probability of blocking is reduced.

As for the first rejected request, improvements are obtained with respect to fixed cost functions, which can range from 20% to 80%, depending on the topology, by using an adaptive cost function that performs load balancing.

On the other hand, better performance is achieved by increasing the number of links or wavelengths in the "Shortest-Path" algorithm. For example, for ("λ") equal to 80, high traffic flows are obtained before the first block.

The OTN technology allowed a great flexibility in terms of network signaling, monitoring and recovery, and it seems logical that the logical evolution of the networks is to transmit IP packets directly over WDM.

For the NSFNET topology, unprotected, the filling of the most charged link occurs around request 450; down to request 250 when there is protection (1 + 1). In parallel, the first request block also undergoes an advance from request 493 to 195, that is, there is a degradation of the order of 60%.

References

1. Chaudhuri, S., Bouillet, E., Ellinas, G.: Addressing transparency in DWDM mesh survivable networks. In: Optical Fiber Communication Conference and Exhibit, OFC 2001. Technical Digest Postconference Edition (IEEE Cat. 01CH37171), vol. 2, p. TuO5-T1-3 (2001)
2. Bisbal, D., de Miguel, I., González, F., Blas, J., Aguado, J.C., Fernández, P., Durán, J., Durán, R., Lorenzo, R.M., Abril, E.J., López, M.: Dynamic routing and wavelength assignment in optical networks by means of genetic algorithms. Photonic Netw. Commun. **7**(1), 43–58 (2004)
3. Bale, K., Bouillet, E., Ellinas, G.: Benefits of "minimal" wavelength interchange in WDM rings. In: Proceedings of Optical Fiber Communication Conference, pp. 120–121 (1997)
4. Subramaniam, S., Azizoglu, M., Somani, A.: On the optimal placement of wavelength converters in wavelength-routed networks. In: Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 1998, vol. 2, pp. 902–909 (1998)
5. Subramaniam, S., Azizoglu, M., Somani, A.K.: All-optical networks with sparse wavelength conversion. IEEE/ACM Trans. Netw. **4**(4), 544–557 (1996)
6. Harai, H., Murata, M., Miyahara, H.: Heuristic algorithm for allocation of wavelength convertible nodes and routing coordination in all-optical networks. J. Light. Technol. **17**(4), 535–545 (1999)
7. Gerstel, O., Kutten, S.: Dynamic wavelength allocation in all-optical ring networks. In: Proceedings of ICC 1997 - International Conference on Communications, vol. 1, pp. 432–436 (1997)
8. Zang, H., Jue, J.P., Mukherjee, B.: A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. Opt. Netw. Mag. **1**, 47–60 (2000)
9. Alferness, R.C., Kogelnik, H., Wood, T.H.: The evolution of optical systems: optics everywhere. Bell Labs Tech. J. **5**(1), 188–202 (2002)
10. Ramaswami, R., Sivarajan, K.N., Sasak, G.H.: Optical Networks: A Practical Perspective. Elsevier/Morgan Kaufmann, Burlington (2002)
11. Walkowiak, K., Klinkowski, M.: Shared backup path protection in elastic optical networks: modeling and optimization. In: 2013 9th International Conference on the Date of Conference on Design of Reliable Communication Networks (DRCN), pp. 187–194 (2013)

12. Chen, B., Zhang, J., Zhao, Y., Liu, J., Huang, S., Gu, W., Jue, J.P.: Minimum spectrum block consumption for shared-path protection with joint failure probability in flexible bandwidth optical networks. In: Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2013, p. OW3A.8 (2013)
13. Tarhan, A., Cavdar, C.: Shared path protection for distance adaptive elastic Optical Networks under dynamic traffic. In: 2013 5th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), pp. 62–67 (2013)
14. Jinno, M., Kozicki, B., Takara, H., Watanabe, A., Sone, Y., Tanaka, T., Hirano, A.: Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network [Topics in Optical Communications]. IEEE Commun. Mag. **48**(8), 138–145 (2010)
15. Takagi, T., Hasegawa, H., Sato, K., Sone, Y., Kozicki, B., Hirano, H., Jinno, M.: Dynamic routing and frequency slot assignment for elastic optical path networks that adopt distance adaptive modulation. In: 2011 Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC) (2011)
16. Pavón, P., Izquierdo, J.L.: Net2Plan - The open-source network planner. http://www.net2plan.com/index.php. Accessed 11 June 2015
17. Izquierdo-Zaragoza, J.-L., Pedreno-Manresa, J.-J., Pavon-Marino, P.: Maximizing IP fast rerouting coverage in survivable IP-over-WDM networks. In: 2015 European Conference on Optical Communication (ECOC), pp. 1–3 (2015)

VR3DMaker: A 3D Modeling System Based on Vive

Shubin Cai[1], Jinchun Wen[1(✉)], Zhong Ming[1], and Zhiguang Shan[2]

[1] College of Computer Science and Software Engineering,
Shenzhen University, Shenzhen, China
{shubin,mingz}@szu.edu.cn, wenion@qq.com
[2] State Information Center of China, Beijing, China
shanzg@cei.gov.cn

Abstract. The capability of conventional Computer Aided Design tools can be improved with the intuitivity of virtual reality (VR for short). VR environments can provide designers with enough creative space, any desired objects and intuitive experience, which is different to the conventional way before monitor. The key technology for VR equipment is SLMA (Simultaneous localization and mapping). One of the VR Manufacturer, HTC, have provided a position and orientation tracking system to solve this problem by its product called Tracker. Therefore, we can identify our motion and track the trail. With a set of such information collected, the models in Virtual Space can be manipulated and deformed like they tend to be in the realistic world. In this paper, we present a 3D modeling platform that building a model based on the Unity3D platform by virtual reality technology. This platform includes a four module with workspace module, a base-model module, a transformation module and a model import and export module.

Keywords: Virtual reality · 3D modeling technology · Modeling flow
Structure optimization

1 Introduction

Virtual reality is one technology of vision that provides users with an immersive experience, multi view immersive 3D virtual environments. The environment that VR equipment create just like the realistic environment, users can interact with the in a virtual reality environment by means of controls such as handles, gestures, and voice as usual.

In general, the VR equipment consists of a head mounted display (HMD) and a position and orientation tracking system [1, 2]. For instance, the Vive consist of a headset, two controllers, and two infrared laser emitter units. After placement of two units, the tracking system maps both HMD and controllers as a point from space that units surround to the space that computer simulate [3]. Their position and orientation information will be gained by unites. In this way, user data such as moving trail, velocity and force can be collected by the controller. With these data, you can map actions to virtual space.

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 173–180, 2018.
https://doi.org/10.1007/978-3-319-73830-7_18

The traditional way of designing industrial models or components can be divided into several steps [4]. However there some difficulties and complicated on implementation on modelling design and product exhibition. It requires an experience worker who have spatial imagining ability in engineering graphics [5, 6]. Further on, because of the complex structures of components and limited field of vision of monitor, it hardly to implementing a coherent design and have to paging your work in several of work files.

By contrast, designers need not be constrained by devices, therefore they can fully inspire their imagination and creativity in the virtual space through the virtual equipment [7]. It not only allows the designer to really see the design object, but also can sense it and interact with it naturally.

The traditional flow process of design is shown in Fig. 1.

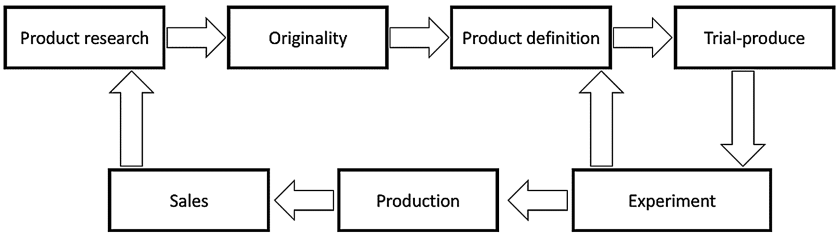


Fig. 1. The software pollute buffer

If virtual design is applied in industry design process, the stages such as sample trial producing could be reduced. Thus it can shorten design time evidently and improve efficiency of design.

Virtual technology can be used in virtual product modeling, as well as physical simulation, dynamic simulation, testing of product performance and reliability, and so on, thus reducing costs and improving efficiency [8, 9].

Our paper provides a real-time 3D model system based on Unity3D (a graphics software design engine) where users can design models or import digital files in different formats to view changes in virtual reality scenes [10]. Unity3D is currently one of the most popular 3D engine in the world, according to the official data released in the first quarter of 2016. The underlying of Unity3D is written in C++. Unity3D will provide the underlying implementation package and can be extended through plug-ins to implementation more functions.

Another software is the release of the SteamVR platform which making Unity3D a platform support developing virtual reality products conveniently. VRTK (Virtual Reality Toolkit) is a virtual reality oriented development of the excellent plug-in, most of the current virtual reality products HTC VIVE development based on the use of the plug-in.

2 Motivation

The 3Dmaker using virtual reality technology to design model in three-dimensional space. There many advantages in modeling in virtual space. For example, the immersion brought by virtual reality makes the application very intuitive, and allows people to immerse themselves in the creation and get a steady stream of inspiration. The most important thing is it is not complicated to use, simple operation for starters.

Starting with a simple set of shapes, a color palette, and an intuitive set of tools, you're able to naturally and quickly create almost anything you can imagine, that means no more mind tricks to create real, volumetric objects on a 2D surface. It's an amazing experience, very intuitive and so natural feeling—you just pick up your two controllers and start pushing, pulling, stretching, and scaling 3D objects to create solid models in virtual space, using a powerful CAD engine.

Simple interactions break the boundaries between professional designers and developers. For one thing, there are amazing creations and use cases from 3D modelers and artists. With a very natural and intuitive two-handed interface, artists achieve the layout in their mind with capability of a much more natural and intuitive way using Vive's room-scale VR.

Even people without any prior modeling experience at all, It's designed to feel more like playing with children's blocks than working with traditional 3D modeling software. It's easy to handle like doing what they'd traditionally do using a keyboard and mouse—without the burden of using a traditional CAD tool. It's as if two generations of 3D designers have been trapped behind a keyboard and now they can model and sculpt in the real world again with complete freedom of motion.

3 System Design

3.1 Architecture of VR3DMaker System

VR3DMaker is a 3D computer-aided design tool for HTC Vive that makes it easy to create, view and edit 3D objects. The file formats supported for import and export of data are STL, OBJ and OFF. The software system can be divided into the following modules:

- Workspace module: establish the coordinate system of the user role and model coordinate system. The user can freely move the workspace and observe the model.
- Basic-model module: four dimensional model of cube, sphere, cylinder and ring.
- Model transformation module: provides the translation, rotation, scale operations and so on.
- Import and export modules: import the corresponding format model in the model coordinate system or export the model in the model coordinate system to the local file.

The relationships among them showed in Fig. 2.

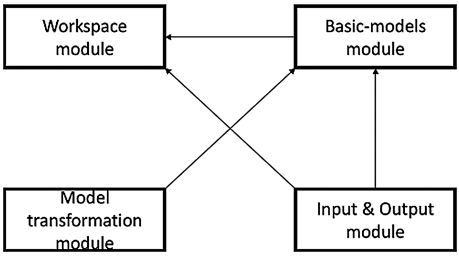


Fig. 2. Relationships among modules

Simple models will be spawned by Base-model module and handle in Workspace module, therefore the Workspace module is the basis of the software. Model transformations actually correspond to model entities in order to model transformations. The model you create will be export by Import and export modules finally.

3.2 Workspace Module

The working space is different from world space (also called user space). As we know, all the objects (model) which need to be transformed belong to the working space. Working space like a container that all objects can be put inside and do some the manipulation.

In order to indicate the working space, we need to set a new coordinate along the Y-axis. First, we create an empty GameObject (Unity3D units) which represents working space named coordinate. Since the user is standing on the $Y = 0$ plane of the world coordinate system in the virtual reality scene, the initial position of the workspace is at the $Y = 1$ plane, and the X axis Z axis is the same as that of the X axis Z axis of the world coordinate axis. As shown in Fig. 3.

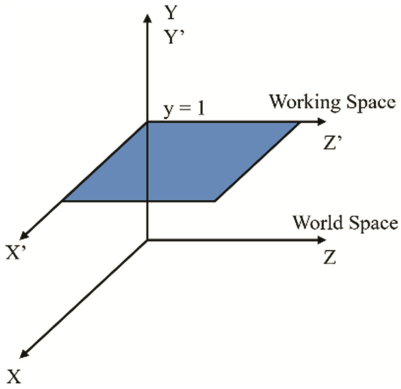


Fig. 3. The software pollute buffer

The GameObject with three linear coordinate indicator in the same coordinate object, therefore any displacement and rotation operation will not affect the integrity of the coordinate system of the Coordinate indicator.

All model objects need to be transformed in the workspace, a new GameObject object named workstation is created under the coordinate object, and all models are generated as child objects of that object. There are need to be considers two cases in following circumstances: when all the models of the working space need to be translation and rotation, we only need to translate the coordinate object, in order to ensure the coordinate indicator and the workspace is not separated; when objects are scales in different size, due to the need to ensure coordinate department of indicator accuracy, so only the object of the Workstation zoom.

3.3 Basic-Models Module

The complex objects can be made up of simple models. There are four base model which named cube, sphere, cylinder and ring are provided by this software system.

When the user create a 3D model, in addition to the way the directly import model from outside, users chose base models which provided by software.

According to the difference of the generative model, the data of the vertices and triangles of the corresponding 3D model are generated. The pre generated vertex data is independent of the workspace coordinate system, and the user adjusts the workspace coordinate parameter without affecting the preset 3D model parameters. The generated 3D models have different names according to the type, and when the same type of underlying model exists, it will have the added number after naming to distinguish the different models.

The model is generated at the origin of the workspace coordinate system, and the length is controlled at about 1 units per length depending on the four models Fig. 4.

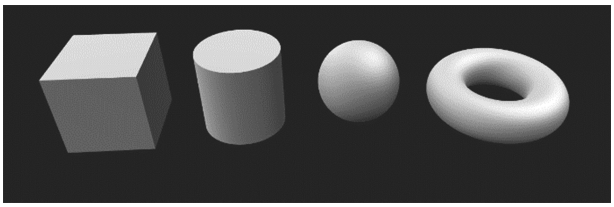


Fig. 4. The samples of basic-models

3.4 Model Transformation Module

For models generated from simple objects or imported objects, some operations need to be processed. The model transformation module is divided into several operations, including moving, scaling, rotating, copying, and deleting. Transform mode and work area, model transformation affects only the operating model is a change in the local coordinate model relative to the workspace, and then converted by area coordinates in the world coordinate position, finally obtains in the world coordinate.

There some public methods named Move, Scale, and Rotate are corresponding to translate, scale, rotate three operations of GameObject. When choosing objects, you need to select a point that represents the position of the object, which is closest to the point of interaction.

Algorithm 1 VecticesSelected: returns the smallest distance from the specified point

```
Procedure VerticesSelected(vectorPosition, vertices)
    Pos -= BaseObj.transform.position;
    While index < vertices.size() do
        posDistance = Distance(Pos, vertices[i]);
        if minDistance >posDistance
                minDistance = posDistance;
        end if
    return index;
end procedure
```

Traversing all vertices of an object, finding the minimum by calculating the difference, thus finding the point of choice.

3.5 Input and Output Module

Due to the formation of model in different coordinate system and zoom ratio of different sizes, some 3D modeling software to create the model appears to be great in the software project, so the need for each point was normalized.

Save the coordinates in the above three file when the absolute value of each coordinate value of the contrast value, then all the vertex coordinates divided by the maximum value taken through all coordinates in the coordinate point, the absolute value of the maximum coordinate value is 1.

But the operation, because the floating point precision problem, two floating-point division will wait for a not accurate floating-point. In this environment, the model files imported will tear.

Therefore, the local scaling of the import model is scaled according to the maximum coordinate value. Since the length of the 10 unit is a moderate model length in this project system, the final local zoom size should be 10 times the default scaling value 1, divided by the absolute value of the maximum coordinate value.

4 Result

The current version of our program runs on a consumer grade PC equipped with a GeForce GTX 1080 graphic card. The paragraph shows results of the models. The appearance of the coordinate axis in Fig. 5 indicator in the virtual reality environment. The light blue frame is the range of activity set by HTC Vive, but it is only marked, and the user can exceed the border activity in the case of space permission.

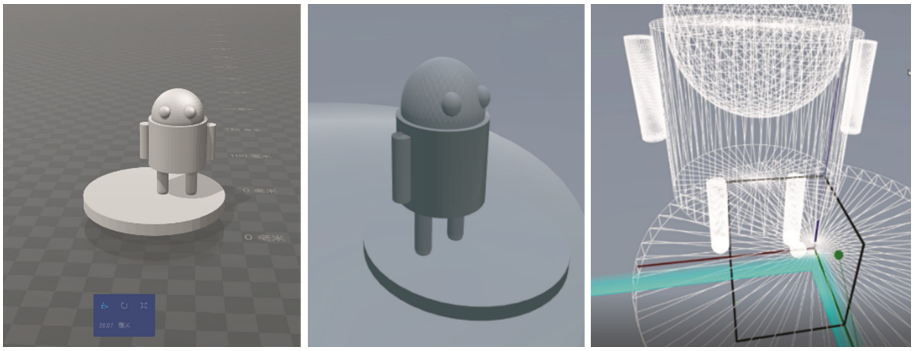


Fig. 5. Exhibition of works (Color figure online)

We designs a model with different shapes consist of sphere, cylinder and spheroid based on appearance of android and exported by the IO model of the system as show in Fig. 5.

We also invited 20 testers to do draw a number of models. All of them have no experience in model design and 3D software before. The experimental procedures are as follows: Besides tell them how to operate the function of software (except operations of basis button or hardware), we ask them to explore in a bit more detail where to find that information using. Models from simple to complex: cuboid, ellipsoid, complex combination. We record the average time of all testers from start drawing to finish work.

Time (s)	Cuboid	Ellipsoid	Complex combination
CAD 2016	10.11	16.15	190.42
VR3DMaker	9.47	12.43	132.14

As you can see from the previous table, when the model is simple, it takes almost the same time, but as the complexity of the model increases, the time spent is increased and the gap becomes larger. The cost of using software is less than that of not using software, so it can be seen that the software is more intuitive and the operation mode is more suitable for people to operate.

5 Conclusion

This system allows users to merge, intersect, or subtract objects from each other, or slice them into pieces. Users can build any models, observe the three-dimensional model, introduce the 3D model of their own design and export and experience the charm of virtual reality technology. Although there is a distinction that must be made between precision and dimensioning, the application of virtual technology can realize the complete digitalization of the product in industrial design, because the virtual product

is a computer model that can have all the information and functions of the product close to reality.

Acknowledgments. Supported by the Project of Department of Education of Guangdong Province. (NO. 2015SQXX0)

References

1. Niehorster, D.C., Li, L., Lappe, M.: The accuracy and precision of position and orientation tracking in the HTC Vive virtual reality system for scientific research. i-Perception **8**(3), 01 May 2017. https://doi.org/10.1177/2041669517708205
2. Kryger, G., Gurocak, H.: Characterization of Nintendo wii remote for 3D tracking in virtual reality. In: ASME 2010 World Conference on Innovative Virtual Reality, pp. 229–235 (2010)
3. Lorenz, M., Spranger, M., Riedel, T., Pürzel, F., Wittstock, V., Klimant, P.: CAD to VR – a methodology for the automated conversion of kinematic CAD models to virtual reality. Procedia Cirp **41**, 358–363 (2016)
4. Chulvi, V., Mulet, E., Felip, F., García-García, C.: The effect of information and communication technologies on creativity in collaborative design. Res. Eng. Des. **28**(1), 1–17 (2016)
5. Raposo, A., Corseuil, E.T.L., Wagner, G.N., dos Santos, I.H.F., Gattass, M.: Towards the use of CAD models in VR applications. In: ACM International Conference on Virtual Reality Continuum and its Applications, pp. 67–74. ACM (2006)
6. Sun, H.: Proceedings of the 2006 ACM International Conference on Virtual Reality Continuum and Its Applications. ACM (2006)
7. Corseuil, E.T.L., Raposo, A.B., Silva, R.J.M.D., Pinto, M.H.G., Wagner, G.N., Gattass, M.: A VR tool for the visualization of CAD models (2004)
8. Zhang, Y., Shi, X.: Research on the three-dimensional displaying of STL ASCII and binary file. Adv. Mater. Res. **940**, 433–436 (2014)
9. Lavoué, G., Dupont, F.: Semi-sharp subdivision surface fitting based on feature lines approximation. Comput. Graph. **33**(2), 151–161 (2009)
10. Jezernik, A., Hren, G.: A solution to integrate computer-aided design (CAD) and virtual reality (VR) databases in design and manufacturing processes. Int. J. Adv. Manuf. Technol. **22**(11–12), 768–774 (2003)

A Knowledge Graph Based Solution for Entity Discovery and Linking in Open-Domain Questions

Kai Lei[1], Bing Zhang[1], Yong Liu[2], Yang Deng[1], Dongyu Zhang[3], and Ying Shen[1,2(✉)]

[1] Institute of Big Data Technologies, Shenzhen Key Lab for Cloud Computing Technology and Applications, School of Electronic and Computer Engineering (SECE), Peking University, Shenzhen 518055, People's Republic of China
{leik,shenying}@pkusz.edu.cn, {zhang_bing,ydeng}@pku.edu.cn

[2] IER Business Development Center, Shenzhen, China
13312962646@189.cn

[3] Computer Science and Engineering Department, Harbin Institute of Technology, Harbin, China
dongyu.z@hit.edu.cn

Abstract. Named entity discovery and linking is the fundamental and core component of question answering. In Question Entity Discovery and Linking (QEDL) problem, traditional methods are challenged because multiple entities in one short question are difficult to be discovered entirely and the incomplete information in short text makes entity linking hard to implement. To overcome these difficulties, we proposed a knowledge graph based solution for QEDL and developed a system consists of Question Entity Discovery (QED) module and Entity Linking (EL) module. The method of QED module is a tradeoff and ensemble of two methods. One is the method based on knowledge graph retrieval, which could extract more entities in questions and guarantee the recall rate, the other is the method based on Conditional Random Field (CRF), which improves the precision rate. The EL module is treated as a ranking problem and Learning to Rank (LTR) method with features such as semantic similarity, text similarity and entity popularity is utilized to extract and make full use of the information in short texts. On the official dataset of a shared QEDL evaluation task, our approach could obtain 64.44% F1 score of QED and 64.86% accuracy of EL, which ranks the 2nd place and indicates its practical use for QEDL problem.

Keywords: Question answering · Data mining · Entity discovery · Entity linking Knowledge graph

1 Introduction

Question Answering (QA) is a popular research direction in Artificial Intelligence, aiming at building a system which can answer natural language questions automatically. Discovering entities in questions and linking them to the corresponding entries in the existing Knowledge Graph (KG) is the first step of QA because rich sources of facts from KG lays the foundation for answering the questions.

Specifically, Named Entity Discovery (or Recognition) (NED) is to discover and extract named entities from texts, which is critical technology of QA, information

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 181–190, 2018.
https://doi.org/10.1007/978-3-319-73830-7_19

extraction, machine translation and other applications. The concept of named entity was firstly proposed at Message Understanding Conference (MUC) [1], referring to the proper names or other meaningful quantity phrases. In order to meet the needs of different applications, the meaning of named entities could be expanded. Entities such as product names, movie names etc. could also be included. Entity Linking (EL) [2] is to resolve named entities to corresponding entries in a structured KG. It can make full use of the semantic information of the rich text in knowledge graph, which has important significance in QA, information retrieval and knowledge graph construction.

To accelerate the development of related research, the China Conference on Knowledge Graph and Semantic Computing (CCKS) organized a shared evaluation task on Question Entity Discovery and Linking (QEDL) in 2017. QEDL is more difficult than traditional NED and EL tasks. Firstly, one short question may contains multiple entities, discovering all of them is a challenge. Secondly, it is difficult to obtain enough context information when linking entities to KG because questions are usually short texts. Moreover, only small amount of manual annotation training data is available sometimes, which requires the efficient method could converge quickly.

To address the challenges mentioned above, we proposed a knowledge graph based solution for QEDL problem and developed a system consists of QED module and EL module. In QED module, the method based on KG retrieval was firstly employed, it could extract more entities in questions and guarantee the recall rate. Then the method based on Conditional Random Field (CRF) is utilized, which could improve the precision rate of entity discovery. Afterwards, two methods were merged together, which is a tradeoff of the precision and recall rate. Furthermore, the ensemble method could converge quickly to obtain ideal performance even if only small training corpus is available. EL module was treated as a ranking problem and Learning to Rank (LTR) method with features such as semantic similarity, entity popularity and text similarity is employed to make full use of the information in short texts.

The rest of this paper is structured as follows: Sect. 2 describes the related work. Section 3 introduces the details of the proposed methods. Experimental results and evaluations are presented in Sect. 4. Finally, we conclude this paper in Sect. 5.

2 Related Work

Lots of works have been involved in the research of NED and EL. The main technical methods of NED include rules and dictionary-based method, statistical method and the emerging method based on deep learning. Rules and dictionary-based method [3] is the earliest method used to NER task. But it has disadvantages such as long system construction period, time-consuming, poor portability and so on. Statistical method for NED uses machine learning models such as Hidden Markov Model (HMM) [4], Maximum Entropy (ME) [5], Support Vector Machine (SVM) [6] and CRF [7], etc. trained by manually annotated corpus. Thus the linguistic knowledge is not required. It can be completed in a short time and change less when transplanted into new domains. The method based on deep learning have been recently proposed, which include bidirectional Long Short-Term Memory with a CRF layer (BiLSTM-CRF) [8], BiLSTM and

convolutional neural networks architecture (BiLSTM-CNN) [9] and other neural network models. Deep learning method doesn't need feature engineering, doesn't use any hand-crafted features or domain specific knowledge, thus it's portable. But it requires large amounts of manual annotation data and long training time. The evaluation of NED has been actively promoted the research. At present, the most influential evaluation meetings include Message Understanding Conference (MUC), Multilingual Entity Task Evaluation (MET), Automatic Content Extraction (ACE), Document Understanding Conference (DUC), etc.

Many of the entity linking systems use supervised machine learning methods, including LTR methods [10], graph-based methods [11] and model integration methods [12]. Vector Space Model (VSM) [13], as an unsupervised learning method, is also widely used in EL systems. In addition, many international meetings organized the evaluation of EL task, such as the "Link the Wiki" task in the EX meeting, the KBP task of the TAC meeting, the KBA tasks of the TREC meeting and the ERD'14 task at SIGIR. Although many researches have been carried out on the general domain, few studies focused on the question entity linking, which is more difficult because the information in questions is incomplete and has a lot of errors.

3 Methods

The QEDL task consists of two subtasks: QED and EL. Because of the small amount of training data, using joint learning method of the two subtasks is difficult to iterate until convergence and is prone to make mistakes. So we designed a pipeline system separating two subtasks with two independency modules.

Figure 1 shows the overview of our system and details are described in this section.

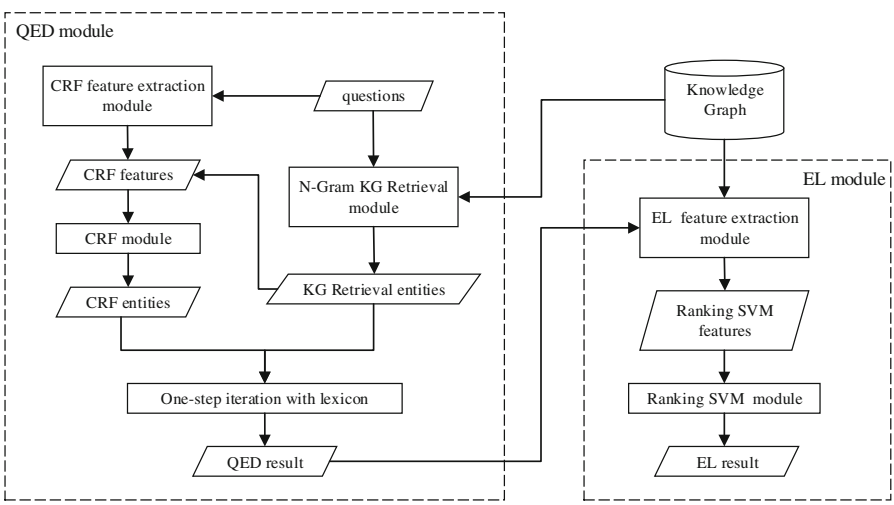


Fig. 1. Overview of our system

3.1 Question Entity Discovery

The QED module is an ensemble of two methods and details are as follows.

Knowledge Graph Retrieval. KG retrieval is widely used to generate candidate entities in question answering over knowledge graph [14]. The core idea is to search n-grams of words of the question to match the entity in the given knowledge graph and select a set of matching entities. We conduct this approach as following steps.

a. Generate all possible n-grams from the question, and tag parts of speech (POS);
b. Replace space and other meaningless symbols with a special mark "_";
c. Remove 1-g that contains only one character;
d. Remove n-grams without any noun, verb, character or number;
e. Keep all the n-grams left which can match a certain entity in knowledge graph.

For example, a given question "孕妇吃方便面好吗?" generates a set of n-grams that match the entity in knowledge graph, like "孕妇/吃/方便/方便面/好/吗". After the procedure, three 1-grams, "吃/好/吗", are removed for containing only one character, while "方便" is discarded due to its POS tags with only adjective. Finally, we remain "孕妇/方便面" as question entities.

The KG retrieval method doesn't need feature extraction, training and testing and can obtain high recall rate, but the precision rate is relatively low.

CRF. CRF method regards QED as a sequence labeling problem. We utilized BIOES tagging rules in the sequence labeling system.

The CRF feature extraction module extracts features presented in Table 1.

Table 1. Feature list of QED module

Feature name	Feature description
Character	A single character, and N-grams (N = 1, 2, 3, 4)
Word boundary (WB)	The boundary of the word where the character is located
Part of speech (POS)	The part of speech of the word where the character is located
Stop words (SW)	The word where the character is located is stop words or not
Document frequency (DF)	The DF value of the word where the character is located

Although features are simple, this method is effective, especially in terms of the precision rate.

CRF Based on KG Retrieval. In consideration of the high recall and low precision rate of KG retrieval method as well as the high precision and low recall rate of CRF method, we proposed a new ensemble method, CRF based on KG retrieval, which merge the two methods mentioned above together. More specifically, we tag the entities discovered in KG retrieval method with BIOES tagging rules and then take it as a feature of CRF.

The ensemble method is a tradeoff of the precision and recall rate and thus improved the system performance. On the one hand, it could improve the precision rate without

much losses of recall rate comparing with the KG method, on the other hand, it could discover more entities with higher precision rate than traditional CRF method. The ensemble method can also obtain good result even though using less training data due to fast convergence.

One-Step Iteration with Lexicon. As our system is a pipeline system of QED and EL, EL module uses the output of QED as input and the performance could be affected by QED results. We hope to improve the recall rate of QED to make sure more entities can be discovered and come into EL model. Thus one-step iteration using the result of KG retrieval and lexicon is added to our system. Those candidate entities discovered by KG retrieval method but ignored by CRF method would be matched to the lexicon. If the candidate entity is in the lexicon, then add it to the final discovery result. The lexicon we used is THUOCL [15] constructed by Tsinghua University.

Recall rate of QED is therefore improved by the iteration, which lays a solid foundation for the next step, EL module.

3.2 Entity Linking

Traditional EL module can be broken into two steps: candidate entity generation and candidate entity ranking. In this task, candidate entities can be generated using the provided API of CN-DBpedia, which is the knowledge graph constructed by Knowledge Works[1]. Therefore, our work mainly focused on candidate entity ranking. Ranking SVM is utilized to rank candidate entities and find the most matching one. To make full use of the information in short questions, rich features are employed and details are described below.

Semantic Similarity. The method we utilized to calculate semantic similarity between the question and candidate entity is Saliency-weighted semantic network proposed by [16]. The function for calculating semantic similarity is:

$$f_{ss} = \sum_{w \in q} IDF(w) \cdot \frac{sem(w,e) \cdot (k_1+1)}{sem(w,e) + k_1 \cdot (1-b+b \cdot \frac{|e|}{avge})} \tag{1}$$

$$sem(w,\ e) = \max_{w \in e} f_{sem}(w,\ w') \tag{2}$$

Here, q is the question, w is the term in q, e is the candidate entity and $avge$ is the average length of candidate entities. $IDF(w)$ calculated from large amount of unlabeled Wiki corpus is used to weight the words in questions based on the idea that common terms (like determiners) do not contribute as much to the meaning of a text as less frequent words do. In formula (2), $sem(w,\ e)$ is the semantic similarity of term w with respect to the candidate entity e. The function f_{sem} returns the semantic similarity between two terms. As terms are represented as vectors using word embeddings trained by Wiki corpus, f_{sem} could be calculated by the distance between two vectors, which reflects

[1] http://kw.fudan.edu.cn/.

semantic similarity information. Cosine similarity is used to calculate distance between vectors in our system. The parameters k1 and b have a smoothing effect and we default set k1 = 1.5 and b = 0.75.

Formula (1) looks similar to the famous BM25 formula [17], but original BM25 formula only captures the lexical similarity between two texts, while we implement the formula with TF-IDF weighting scheme and word embeddings to measure both lexical and semantic similarity between two texts.

Text Similarity. Term Frequency-Inverse Document Frequency (TF-IDF) model [18], Latent Semantic Indexing (LSI) model [19] and Latent Dirichlet Allocation (LDA) model [20] are effective and frequently used methods for text similarity calculation.

TF-IDF model convert text into fixed-length vector space and spatial similarity is used to approximate text similarity. Words in the text are weighted by the number of occurrences in the text and the importance to the text. LSI uses Singular value decomposition (SVD) technique to word-document matrix to reduce the dimension of TF-IDF model. LDA is the topic model, the word vectors of texts after remove stop words are mapped to the topic distribution and cosine similarity is calculated to represent the text similarity. Gensim[2] is used to build TF-IDF, LSI and LDA model.

The three methods above are exploited to calculate text similarity between questions and the name of candidate entities and the values were put together as a feature set of learning to rank model. In addition to the text similarity between questions and entity name (TS_QEN for short), text similarity between questions and the attributes of candidate entities obtained by API (TS_QEA for short) is also calculated.

Entity Popularity. The popularity of an entity indicates the possibility of the entity being mentioned in a question. We use the number of results returned by search engine when searching the entity to represent entity popularity. The popularity feature is defined as follows:

$$P(e) = logN \tag{3}$$

Given an entity e, N is the hit number returned by Baidu. For example, the entity mention "方便面" corresponds to two candidate entities in CN-DBpedia, "方便面（快餐类面制食品）" and "方便面（中国大陆歌手肖飞演唱歌曲）". When we search them respectively in Baidu, the former retrieves about 1,370 relevant results while the latter retrieves about 447 relevant results. The popularity of candidate entities proved to be a distinguishable feature to EL task.

4 Experiments and Evaluation

Experiments and evaluation have been carried out based on the training set which contains about 1400 manually annotated questions and the test set contains about 800 questions without labels published by CCKS2017 QEDL task. The knowledge graph

[2] http://radimrehurek.com/gensim/index.html.

this task uses is CN-DBpedia, which contains hundreds of millions of entities and could be accessed through API. The evaluation results are as follows.

4.1 Question Entity Discovery Results

QED is treated as a sequence labeling problem in our system and different methods with different features described in Sect. 3.1 are exploited. To evaluate the results, Precision rate, Recall rate and F1 Score are used as evaluation indicators in QED module. The results of the experiment are shown in Table 2.

Table 2. Performance of QED module

Methods	Features	Precision (%)	Recall (%)	F1 (%)
KG retrieval	/	28.63	72.60	41.06
CRF	Character	44.66	43.28	43.96
	Character + WB	46.95	50.11	48.48
	Character + WB + POS	46.46	53.84	49.88
	Character + WB + POS + SW	47.42	53.94	50.47
	Character + WB + POS + SW + DF	47.88	54.26	50.87
CRF based on KG retrieval	Character + WB + POS + SW + DF + KG information	**55.90**	67.16	61.02
One-step iteration with lexicon	Character + WB + POS + SW + DF + KG information	55.36	**77.08**	**64.44**

As is presented in Table 2, KG retrieval method could obtain high recall rate, but the precision rate is low. Traditional CRF method with features such as character, word boundary, part of speech, stop words and document frequency of terms obtained higher precision rate and F1 score compare with KG Retrieval method, but the recall rate is relatively low. The CRF based on KG retrieval method is really effective, it has positive effect on both precision and recall rate and increases 10.15% point of the F1 score on the foundation of traditional CRF method. At last, although one-step iteration with lexicon couldn't increase the precision rate, it has greatly improved the recall rate and thus improved the F1 score, which also lays a solid foundation for the next step, EL module.

In addition to the quality of entity discovery, convergence speed of methods should also be concerned about, especially when only a small amount of labeled training data is available. To evaluate the convergence speed, we developed an experiment utilizing different size of training sets and different methods. The methods being evaluated in this experiment include traditional CRF, CRF based on KG Retrieval proposed in this paper and BiLSTM-CRF, the emerging and outstanding deep learning method for entity discovery. The result is shown in Fig. 2.

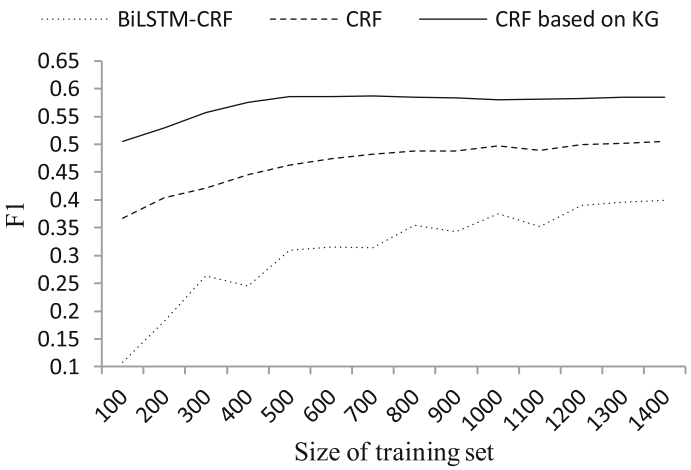


Fig. 2. Comparisons of convergence speed in different methods

Figure 2 illustrates that traditional CRF method could converge when the size of training set is about 1000 while the method we proposed, CRF based on KG Retrieval, could converge utilizing only about 600 training data, which is very efficient. As for BiLSTM-CRF method, it is hard to converge on small training set thus the performance on this task is unsatisfactory, it requires much larger size of manually annotated training data.

4.2 Entity Linking Results

In the EL module, ranking SVM with features described in Sect. 3.2 is employed to rank the candidate entities. In order to evaluate the performance of EL without being disturbed by the result of QED, we only evaluate the EL performance of those correctly recognized entities. Obviously, the Precision = Recall = F1 = Accuracy of EL under the premise that the correct entity mention is given. Table 3 shows the experimental results of EL module.

Table 3. Performance of EL module

Features	Accuracy (%)
Semantic similarity	60.71
Semantic similarity + TS_QEN	61.55
Semantic similarity + TS_QEN + TS_QEA	62.37
Semantic similarity + TS_QEN + TS_QEA + Entity popularity	**64.86**

From Table 3 we can see, performance of EL module is improved with the increase of the feature sets, each of them can capture different aspects of information in questions and candidate entities and thus contribute to the result.

In addition to evaluating the EL module, it is interesting to see how different sets of features affect the performance, thus the ablation study is carried out. To analyze the importance of one feature set, we leave out it and use the rest of the feature sets to calculate the result. The results of the ablation study are shown in Table 4, sorted by accuracy.

Table 4. Effect of each feature set

Omitted feature set	Accuracy (%)
Semantic similarity	60.44
TS_QEN	61.13
TS_QEA	61.55
Entity popularity	62.79

Table 4 shows that leaving out the semantic similarity feature has the most dramatic effect on performance. So the semantic similarity feature has the most significant contribution to the ranking model, next is TS_QEN and TS_QEA feature sets, the entity popularity feature contributes least.

4.3 Overall Results

At last, we evaluate the overall performance of the entire QEDL system. The evaluation results are shown in Table 5.

Table 5. Overall performance of QEDL

NED (%)			EL (%)	Overall (%)		
Precision	Recall	F1	Accuracy	Precision	Recall	F1
55.36	77.08	64.44	64.86	38.96	54.05	45.28

The overall performance of our method ranks the 2nd place in CCKS2017 QEDL task, indicating its practical use for QEDL problem.

5 Conclusion

This paper introduces a knowledge graph based solution of QEDL problem consists of QED and EL module. In the QED module, CRF based on knowledge graph retrieval with one-step iteration method is utilized, which could discover high-density entities in short questions and guarantee the recall rate without losses of precision rate. The method also converge quickly and the advantage is more obvious especially when the training set is relatively small. EL module is treated as a ranking problem and ranking SVM with semantic similarity, text similarity and popularity features is employed to make full use of the information in short texts. The results of evaluation show that our approach could converge faster than BiLSTM-CRF method in QED and obtain higher F1 score up to 64.44% while the accuracy of EL is 64.44%, which ranks the 2nd place in the QEDL evaluation task. According to the result, our solution for QEDL is valuable. In the future

we want to extend our system to more NED and EL problems not only in questions but also in other short texts.

Acknowledgement. This work was financially supported by the National Natural Science Foundation of China (No. 61602013), and the Shenzhen Key Fundamental Research Projects (Grant Nos. JCYJ20160330095313861, JCYJ20151030154330711 and JCYJ20151014093505032).

References

1. Chinchor, N.: MUC7 named entity task definition. In: MUC (1997)
2. Han, X., Sun, L.: A generative entity-mention model for linking entities with knowledge base. In: ACL, pp. 945–954 (2011)
3. Humphreys, R.G., Azzam, S., Huyck, C., Mitchell, B., Cunningham, H., Wilks, Y.: Description of the LaSIE-II System as Used for MUC7, pp. 127–140 (1998)
4. Fu, G., Luke, K.K.: Chinese named entity recognition using lexicalized HMMs. ACM SIGKDD Explor. Newsl. **7**(1), 19–25 (2005)
5. Hai, L.C., Ng, H.T.: Named entity recognition: a maximum entropy approach using global information. In: COLING, pp. 1–7 (2002)
6. Li, L., Mao, T., Huang, D., Yang, Y.: Hybrid models for Chinese named entity recognition. In: Proceedings of SIGHAN Workshop, pp. 72–78 (2006)
7. Chen, A., Peng, F., Shan, R., Sun, G.: Chinese named entity recognition with conditional probabilistic models, pp. 173–176 (2006)
8. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. Comput. Sci. (2015)
9. Chiu, J. P. C., Nichols, E.: Named entity recognition with bidirectional LSTM-CNNs. Comput. Sci. (2015)
10. Zheng, Z., Li, F., Huang, M., Zhu, X.: Learning to link entities with knowledge base. In: NAACL, pp. 483–491 (2010)
11. Hoffart, J., Yosef, M.A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M.: Robust disambiguation of named entities in text. In: EMNLP, pp. 782–792 (2011)
12. Mihalcea, R., Csomai, A.: Wikify! linking documents to encyclopedic knowledge. In: CIKM, pp. 233–242 (2007)
13. Cucerzan, S.: Large-scale named entity disambiguation based on wikipedia data. In: EMNLP-CoNLL, pp. 708–716 (2007)
14. Golub, D., He, X.: Character-level question answering with attention. In: Proceedings of EMNLP, pp. 1598–1607 (2016)
15. Han, S., Zhang, Y., Ma, Y.: THUOCL: Tsinghua open Chinese lexicon (2016)
16. Kenter, T., Rijke, M.D.: Short text similarity with word embeddings. In: CIKM, pp. 1411–1420 (2015)
17. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Found. Trends® Inf. Retr. **3**(4), 333–389 (2009)
18. Xu, W.: A Chinese keyword extraction algorithm based on TFIDF method. Inf. Stud. Theory Appl. (2008)
19. Mirzal, A.: Similarity-based matrix completion algorithm for latent semantic indexing. In: IEEE ICCSCE, pp. 79–84 (2014)
20. Celikyilmaz, A., Hakkani-Tur, D., Tur, G.: LDA based similarity modeling for question answering. In: Proceedings of the NAACL HLT Workshop, pp. 1–9 (2010)

Predicting App Usage Based on Link Prediction in User-App Bipartite Network

Yaowen Tan(✉), Ke Yu, Xiaofei Wu, Di Pan, and Yang Liu

Beijing University of Posts and Telecommunications, Beijing, China
tanyaowen@bupt.edu.cn

Abstract. Nowadays the explosion of smartphone Apps has created a fertile ground to study behavior of mobile users. In this paper, we utilize network footprint data (NFP) which consists of DPI data from ISPs and Crawler data from Web for predicting App usage. We construct User-App Bipartite network based on the network footprint data and propose the App Usage prediction method based on link prediction. We extract three-category features calculated from the bipartite network, the projection network and the original NFP data respectively and apply supervised machine learning models to the proposed features. We compare the results of App usage prediction model with different features combination in our experiments. It can be seen that the proposed link-prediction-based method is very effective for App usage prediction.

Keywords: App usage · Link prediction · Bipartite network

1 Introduction

With the development of smart phones and the speed upgrade of the wireless network, people can ubiquitously access to the Internet, and surf the Internet anywhere at anytime. At the same time, the explosion of number of smartphone Apps and their diversity has created a fertile ground to study behavior of smartphone users. Understanding user behavior of App usage, is important and valuable not only for the App developer but also for Internet Service Providers (ISPs), for better network management and service provisioning.

Nowadays lots of researches on App usage are based on App usage log from App servers or sensor log from smart phones. These datasets are usually related to a specific App or a small group of users, due to the limitation of data scale. The heterogeneous data from ISP's network as well as Web related to a specific mobile user is called network footprint (NFP). Like a person's footprint, network footprint data records the mobile user's action, location, interest etc. for a long term on mobile Internet. Actually, the network footprint data can be used to analysis the user's behavior of App usage. The NFP data consists of the Deep Packet Inspection (DPI) data from ISPs and the App's information from Web Crawler. The DPI data provides detailed traffic interaction between every user and App server, at the same time the App's information clarifies App's

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 191–205, 2018.
https://doi.org/10.1007/978-3-319-73830-7_20

domain name, category, and other characteristics. The NFP data is a complete presentation of the behavior of a user which is just like the user's footprint when surfing on Internet. So the NFP data provides us a wide perspective to analyze user's behavior of App usage.

In this paper we focus on App usage prediction based on link prediction in bipartite network for the NFP data. Our main task is to predict whether a user will use a App based on the historical NFP data. The detailed record in NFP data includes a user's id that can uniquely identify a user, accurate time (in hour or minute), the App's id that the user visited in the time, the Page View (PV) number that represents the user visiting frequency in the time, the App's category id, etc. We construct User-App bipartite network, and transform the App usage prediction to a link prediction problem in the complex network which can focus on extracting missing information. Then the link prediction problem is reformulated in terms of a two-class discrimination problem: linking (user uses this App) and not-linking (user doesn't use this App), hence allowing to apply classical supervised machine learning approach for learning prediction models. We collect 4-day NFP data from ISP's operational network, and experimental results illustrate the effectiveness of our proposed method.

The rest of paper is organized as follows. In Sect. 2, we discuss related work on App usage prediction and link prediction in bipartite network. In Sect. 3, the App usage prediction method are proposed. In Sect. 4, we describe the experimental results. In Sect. 5, conclusions are presented.

2 Related Work

There are a lot of Apps emerging on Internet everyday because of the rapidly increasing needs of users. Since the analysis of App usage plays a more and more important role for the App developer, many researchers are focusing on it. The author of [6,7] designed an AppNow widget that is able to dynamically predict users' Apps usage through mining temporal profiles from the users' previous usage behavior. [8] adopted the concept of minimum description (MDL) to select personalized features for different users and proposed a KNN-based App Prediction framework (KAP) to predict App usage from the App usage log joint with the Hardware Sensors data. In [14], the authors proposed a hybrid model combining the advantages of neighborhood models and latent factor models based on collaborative filtering to recommend Apps by predicting users' behavior of App usage. [9] proposed a Temporal-based Apps Predictor (TAP) to dynamically predict the Apps which are mostly to be used. In [16], the authors developed an App usage prediction model that leverages three key everyday factors that affect App usage decisions (i.e. intrinsic user App preferences and user historical patterns; user activities and the environment as observed through sensor-based contextual signals; the shared aggregate patterns of App behavior that appear in various user communities). [17] studied the mobile Apps usage pattern from user perspective using the collected data such as time, location, last used App, network type, network speed on mobile devices. However, those researches on

App usage prediction only considered the visible features which will lose many features that are hidden in a relatively small dataset. In this paper, in order to utilize the global information about App usage of mobile users and mine the hidden relations between users and Apps, we use the NFP data that contains traffic data and crawler data to construct the bipartite network, and then predict the user's behavior of App usage based on the link prediction in bipartite network.

Link prediction in complex networks has attracted increasing attention from both physical and computer science communities. The algorithms can be used to extract missing information, identify spurious interactions, evaluate network evolving mechanisms etc. [10]. In [13], the author focused on more practical link prediction with finding the latent correlation among the contacting common neighbors and alleviating the strong independency hypothesis among shared neighbors in five representative datasets. [5] implemented Navie Bayes approach by considering it as a binary classification problem to identify the missing links in an ego network. In [12], an innovative clustering-based method was proposed to solve two problems: the one is that clustering characteristics of nodes are utilized to dig potential links, and the other is that the link prediction algorithm is modified for specific networks to achieve better results.

The bipartite network is a special case of the traditional complex networks, which contains edges between two types of nodes so that two nodes belonging to the same type do not have any edges, and therefore the local methods mentioned in [10] cannot be used directly without any specialization. [11] constructed a drug-side effect network based on the SIDER2 database and introduced link prediction methods in the network. The authors of [15] proposed a few implementations of structural holes which are then validated with extended neighborhood-based methods on a real dataset derived from IMDb network. In [3], the authors created a bipartite graph from the Yelp Challenge dataset by connecting uses to businesses by the reviews they wrote and then attempted to predict new edges (i.e. reviews) at a later time. [4] defined the concept of candidate node pair (CNP) based on the projected graph which maps the bipartite network on to a unipartite one, and then performed the link prediction only within the CNPs. In this paper, we construct the User-App bipartite network and project the graph to User side as well as the App side and then use some link prediction algorithms to extract features from the original graph as well as the projection map to build a supervised machine learning approach to predict the user's behavior of App usage.

3 Propsed App Usage Prediction Method

3.1 NFP Dataset

In our study, the NFP data consists of two parts. The first part of data called Deep Packet Inspection (DPI) comes from an ISP in a province of eastern China which contains millions of users' online records such as the domains that users visit every time. Another part called the Apps' information comes from the Web

Crawler which contains every App's information and the category of the App. As shown in Fig. 1, we use wireshark to grab the App rules that represent the URL format when a user downloads or initiates an App, then we use these App rules to extract the information of user and App interaction in DPI data on Hadoop distributed platform. At the same time, we get the information of the App itself by the Web Crawler. Finally, we join the DPI data and the Web Crawler data together by the App name, and store the NFP data in distributed database such as GreenPlum. Table 1 shows the detailed fields of NFP data.

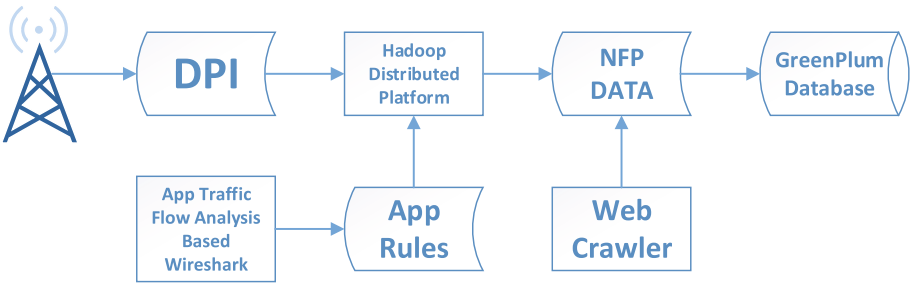


Fig. 1. The NFP data collection process

Table 1. The detailed fields in the NFP data

Fields	Description
User_id	Uniquely identify the user
App_id	Uniquely identify the App
App_category	The category to which this App belongs
Time (hour)	When the user used the App
PV	The number of times users use the App at this hour

3.2 User-App Bipartite Network Construction

We denote the bipartite network with the bipartite graph $G(U, A, E)$ where U and A represent the two types of nodes in G that U contains all users and A contains all Apps in our NFP data. And E represent the edges between users and Apps, and there is no edge within U as well as within A, every edge $(u, a) \in E$ satisfies $u \in U, a \in A$. As shown in Fig. 2, the red color node indicates users, green means Apps.

Our goal is to predict whether a User-App link will exist in the future, so we build a supervised machine learning model of classifier and train it with the features extracted from data in the past period of the User-App bipartite

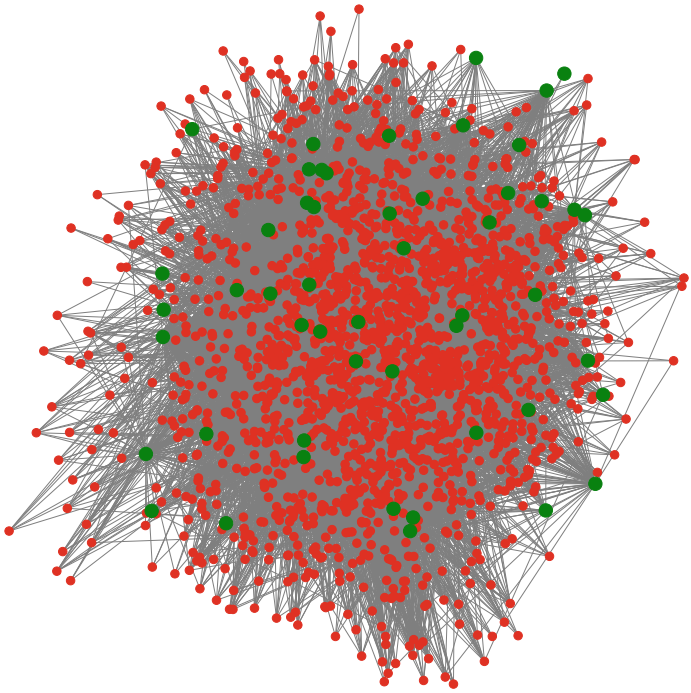


Fig. 2. The User-App bipartite network (Color figure online)

network, and evaluate the model on the test data in the time period following the train data (i.e. use the data of the previous day to predict the day following it). We define G_n as a bipartite network constructed based on App usage at day n, and consider G_{n+1} as the labeling graph that represents the App usage at the day after day n. Let the time sequence be $\{t_n, t_{n+1}, t_{n+2}\}$. We use the sliding window to build the model, and then partition the data into two parts. We use the data of t_n day to extract features as the train features and the data of t_{n+1} day as the train target. At the same time, we use data of t_{n+1} day to extract features as the test features and the data of t_{n+2} day as the test target. For example, in training samples, the positive examples are the User-App having an edge in t_{n+1}, meaning that the user used the App in t_{n+1}. The negative examples are the User-App not having edge in t_{n+1}, meaning that the user had not used the App in t_{n+1}. We train our models with samples from $[t_n, t_{n+1}]$, and we make predictions with our train model on User-App pairs in t_{n+1}. Finally, we evaluate our predictions by examing t_{n+2}. We choose Logistic Regression (LR) and XGBoost (XGB) as our classifier model. The Fig. 3 shows the framework of our whole approach.

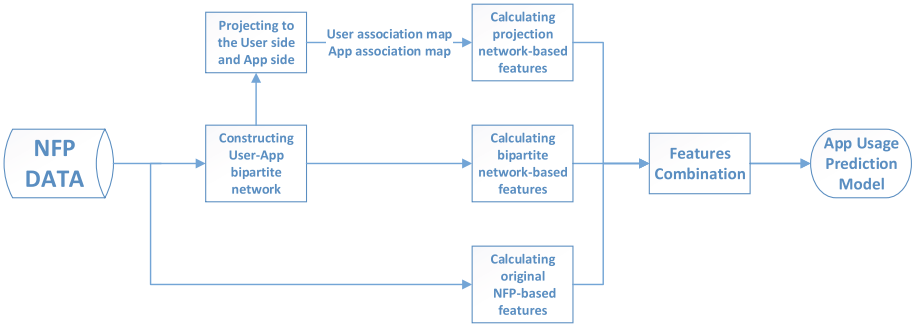


Fig. 3. The framework of our approach

3.3 Feature Extraction

We define three-category features based on the NFP data, i.e. Bipartite Network-based features, Projection Network-based features, Original NFP-based features.

3.3.1 Bipartite Network-Based Features

From the User-App bipartie network $G(U, A, E)$, let b belonging to U or A represent a user or an App, and the corresponding x belonging to A or U represent an App or a user. We define

$$\Gamma(b) = \{x|(b, x) \in E, b \in U \ or \ A\} \tag{1}$$

to signify the set of neighbors of node b (a user or an App) in G,

$$\Gamma'(b) = \cap_{y \in \Gamma(b)} \Gamma(y) \tag{2}$$

to signify the set of b's neighbors' neighbors.

Some of the features from the User-App bipartite network are defined based on the methods proposed in [2], and the other can be defined in the similar way. Table 2 shows these Bipartite Network-based features. As can be seen from the table, every feature has two descriptions because we not only calculated it from User side to App side, but also from App side to User side in the User-App bipartite network. For a User-App pair (u, a), $\Gamma(u)$ is the set of neighbors of a user (i.e. a user uses a collection of all Apps), $\Gamma'(a)$ indicates the intersection of the set of Apps used by each user of App a (i.e. who used this App also used the other App), $\Gamma(a)$ is the set of neighbors of a App (i.e. an App is used by a collection of users), $\Gamma'(u)$ indicates the intersection of the set of users who used the same App as user u (i.e. user used Apps' other users). Meanwhile, the k_u and k_a is the degree of $\Gamma(u)$ and $\Gamma(a)$, k'_u and k'_a is the degree of $\Gamma'(u)$ and $\Gamma'(a)$, d is the distance between a user and an App in the graph.

All the above features can be calculated directly from the bipartite graph, which indicate the relations between a user and an App.

Table 2. Bipartite network-based features

Bipartite network-based feature	Description
Common Neighbors (CN)	$Score_{u,a}^{CN} = \left\|\Gamma\left(u\right) \cap \Gamma'\left(a\right)\right\|$ $Score_{a,u}^{CN} = \left\|\Gamma\left(a\right) \cap \Gamma'\left(u\right)\right\|$
Salton Index (Salton)	$Score_{u,a}^{Salton} = \frac{\left\|\Gamma(u) \cap \Gamma'(a)\right\|}{\sqrt{k_u * k'_a}}$ $Score_{a,u}^{Salton} = \frac{\left\|\Gamma(a) \cap \Gamma'(u)\right\|}{\sqrt{k_a * k'_u}}$
Jaccard Index (Jaccard)	$Score_{u,a}^{Jaccard} = \frac{\left\|\Gamma(u) \cap \Gamma'(a)\right\|}{\left\|\Gamma(u) \cup \Gamma'(a)\right\|}$ $Score_{a,u}^{Jaccard} = \frac{\left\|\Gamma(a) \cap \Gamma'(u)\right\|}{\left\|\Gamma(a) \cup \Gamma'(u)\right\|}$
Sorensen Index (SI)	$Score_{u,a}^{Sorensen} = \frac{2\left\|\Gamma(u) \cap \Gamma'(a)\right\|}{k_u + k'_a}$ $Score_{a,u}^{Sorensen} = \frac{2\left\|\Gamma(a) \cap \Gamma'(u)\right\|}{k_a + k'_u}$
Hub Promoted Index (HPI)	$Score_{u,a}^{HPI} = \frac{\left\|\Gamma(u) \cap \Gamma'(a)\right\|}{\min\{k_u, k'_a\}}$ $Score_{a,u}^{HPI} = \frac{\left\|\Gamma(a) \cap \Gamma'(u)\right\|}{\min\{k_a, k'_u\}}$
Hub Depressed Index (HDI)	$Score_{u,a}^{HDI} = \frac{\left\|\Gamma(u) \cap \Gamma'(a)\right\|}{max\{k_u, k'_a\}}$ $Score_{a,u}^{HDI} = \frac{\left\|\Gamma(a) \cap \Gamma'(u)\right\|}{max\{k_a, k'_u\}}$
Leicht-Holme-Newman Index (LHN1)	$Score_{u,a}^{LHN1} = \frac{\left\|\Gamma(u) \cap \Gamma'(a)\right\|}{k_u * k'_a}$ $Score_{a,u}^{LHN1} = \frac{\left\|\Gamma(a) \cap \Gamma'(u)\right\|}{k_a * k'_u}$
Preferential Attachment Index (PA)	$Score_{u,a}^{PA} = k_u * k'_a$ $Score_{a,u}^{PA} = k_a * k'_u$
Adamic-Adar Index (AA)	$Score_{u,a}^{AA} = \sum_{x \in \Gamma(u) \cap \Gamma'(a)} \frac{1}{\log k_x}$ $Score_{a,u}^{AA} = \sum_{x \in \Gamma(a) \cap \Gamma'(u)} \frac{1}{\log k_x}$
Resource Allocation Index (RA)	$Score_{u,a}^{RA} = \sum_{x \in \Gamma(u) \cap \Gamma'(a)} \frac{1}{k_x}$ $Score_{a,u}^{RA} = \sum_{x \in \Gamma(a) \cap \Gamma'(u)} \frac{1}{k_x}$
Shortest Distance (SD)	$Score_{u,a}^{SD} = Score_{a,u}^{SD} = \min_{d \in paths} d$

3.3.2 Projection Network-Based Features

We projected the User-App bipartite network to a User association map as well as an App association map. As illustrated in Fig. 4, two user nodes are connected in the User association map if they use the same App and two Apps are connected in the App association map if they are used by same user. The User association map and App association map are defined as follows:

$$G_{\mathrm{U}}^{n} = \langle V_U \subseteq U, E = \{(u_1, u_2) : u_1, u_2 \in U, \left|\Gamma_G\left(u_1\right) \cap \Gamma_G\left(u_2\right)\right| \geq n\}\rangle \quad (3)$$

$$G_{\mathrm{A}}^{m} = \langle V_A \subseteq A, E = \{(a_1, a_2) : a_1, a_2 \in A, \left|\Gamma_G\left(a_1\right) \cap \Gamma_G\left(a_2\right)\right| \geq m\}\rangle \quad (4)$$

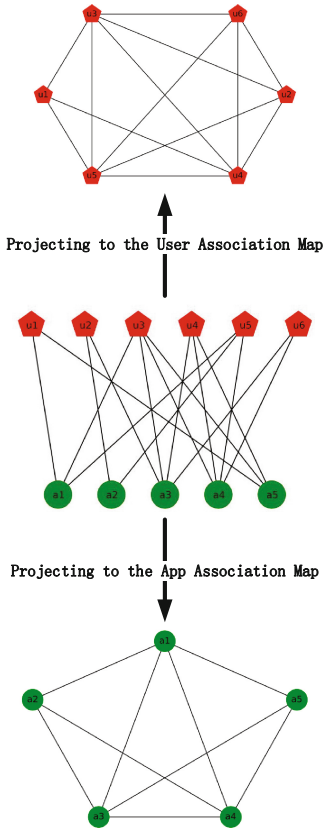


Fig. 4. Projecting the User-App network to the User association map and App association map

The projection network-based features we use are similar to the approach proposed in [1].

$$f^i_{G^n_U}(u,a) = \varphi_{x\in\Gamma_{G_u}(u),y\in\Gamma_G(a)}\left(f^i_{G^n_U}(x,y)\right) \tag{5}$$

$$f^i_{G^m_A}(u,a) = \varphi_{x\in\Gamma_{G_a}(a),y\in\Gamma_G(u)}\left(f^i_{G^m_A}(x,y)\right) \tag{6}$$

Let $f^i_G(u,a)$ be a feature that f represents the algorithm mentioned in [10], An aggregate function φ selects the most Appropriate value from the set{min,max}, depending on the feature. For instance, we calculate the projection network-based feature of a pair of User-App of CN, f represent CN at this time and φ will be the max() function. Note that both projection parameter n and m can take different values.

Table 3 shows the projection network-based features. $\Gamma_{G_u}(u)$ represent the set of u's neighbors in the User association map and $\Gamma_{G_a}(a)$ represent the set of a's neighbors in the App association map, and $\Gamma_G(b)$ represent the set of b's neighbors in the User-App bipartite network.

Table 3. Projection Network-based Features

Projection graph	Feature	Description
User association map	CN	$CN_{G_U^n}(u,a) = \max_{x\in\Gamma_{G_u}(u),y\in\Gamma_G(a)}\left(CN_{G_U^n}(x,y)\right)$ $CN_{G_U^n}(x,y) = \left\lvert\Gamma_{G_U^n}(x)\cap\Gamma_{G_U^n}(y)\right\rvert$
	Jaccard	$Jaccard_{G_U^n}(u,a) = \max_{x\in\Gamma_{G_u}(u),y\in\Gamma_G(a)}\left(Jaccard_{G_U^n}(x,y)\right)$ $Jaccard_{G_U^n}(x,y) = \frac{\left\lvert\Gamma_{G_U^n}(x)\cap\Gamma_{G_U^n}(y)\right\rvert}{\left\lvert\Gamma_{G_U^n}(x)\cup\Gamma_{G_U^n}(y)\right\rvert}$
App association map	CN	$CN_{G_A^m}(u,a) = \max_{x\in\Gamma_{G_a}(a),y\in\Gamma_G(u)}\left(CN_{G_A^m}(x,y)\right)$ $CN_{G_A^m}(x,y) = \left\lvert\Gamma_{G_A^m}(x)\cap\Gamma_{G_A^m}(y)\right\rvert$
	Salton	$Salton_{G_A^m}(u,a) = \max_{x\in\Gamma_{G_a}(a),y\in\Gamma_G(u)}\left(Salton_{G_A^m}(x,y)\right)$ $Salton_{G_A^m}(x,y) = \frac{\left\lvert\Gamma_{G_A^m}(x)\cap\Gamma_{G_A^m}(y)\right\rvert}{\sqrt{k_x * k_y}}$
	Jaccard	$Jaccard_{G_A^m}(u,a) = \max_{x\in\Gamma_{G_a}(a),y\in\Gamma_G(u)}\left(Jaccard_{G_A^m}(x,y)\right)$ $Jaccard_{G_A^m}(x,y) = \frac{\left\lvert\Gamma_{G_A^m}(x)\cap\Gamma_{G_A^m}(y)\right\rvert}{\left\lvert\Gamma_{G_A^m}(x)\cup\Gamma_{G_A^m}(y)\right\rvert}$
	SI	$SI_{G_A^m}(u,a) = \max_{x\in\Gamma_{G_a}(a),y\in\Gamma_G(u)}\left(SI_{G_A^m}(x,y)\right)$ $SI_{G_A^m}(x,y) = \frac{2\left\lvert\Gamma_{G_A^m}(x)\cap\Gamma_{G_A^m}(y)\right\rvert}{k_x+k_y}$
	HPI	$HPI_{G_A^m}(u,a) = \max_{x\in\Gamma_{G_a}(a),y\in\Gamma_G(u)}\left(HPI_{G_A^m}(x,y)\right)$ $HPI_{G_A^m}(x,y) = \frac{\left\lvert\Gamma_{G_A^m}(x)\cap\Gamma_{G_A^m}(y)\right\rvert}{\min\{k_x,k_y\}}$
	HDI	$HDI_{G_A^m}(u,a) = \max_{x\in\Gamma_{G_a}(a),y\in\Gamma_G(u)}\left(HDI_{G_A^m}(x,y)\right)$ $HDI_{G_A^m}(x,y) = \frac{\left\lvert\Gamma_{G_A^m}(x)\cap\Gamma_{G_A^m}(y)\right\rvert}{\max\{k_x,k_y\}}$
	LHN1	$LHN1_{G_A^m}(u,a) = \max_{x\in\Gamma_{G_a}(a),y\in\Gamma_G(u)}\left(LHN1_{G_A^m}(x,y)\right)$ $LHN1_{G_A^m}(x,y) = \frac{\left\lvert\Gamma_{G_A^m}(x)\cap\Gamma_{G_A^m}(y)\right\rvert}{k_x * k_y}$
	PA	$PA_{G_A^m}(u,a) = \max_{x\in\Gamma_{G_a}(a),y\in\Gamma_G(u)}\left(PA_{G_A^m}(x,y)\right)$ $PA_{G_A^m}(x,y) = k_x * k_y$
	AA	$AA_{G_A^m}(u,a) = \max_{x\in\Gamma_{G_a}(a),y\in\Gamma_G(u)}\left(AA_{G_A^m}(x,y)\right)$ $AA_{G_A^m}(x,y) = \sum_{z\in\Gamma_{G_A^m}(x)\cap\Gamma_{G_A^m}(y)}\frac{1}{\log k_z}$
	RA	$RA_{G_A^m}(u,a) = \max_{x\in\Gamma_{G_a}(a),y\in\Gamma_G(u)}\left(RA_{G_A^m}(x,y)\right)$ $RA_{G_A^m}(x,y) = \sum_{z\in\Gamma_{G_A^m}(x)\cap\Gamma_{G_A^m}(y)}\frac{1}{k_z}$
	SD	$SD_{G_A^m}(u,a) = \min_{x\in\Gamma_{G_a}(a),y\in\Gamma_G(u)}\left(SD_{G_A^m}(x,y)\right)$ $SD_{G_A^m}(x,y) = \min_{d\in paths}$

Table 4. Original NFP-based features

	Feature	Description
User related	ActiveHourRatio	The number of hours the user online accounts for the proportion of whole day
	SumApps	The number of Apps that the user use per day
	SumCategories	The number of categories of App that the user use per day
App related	PVProportionOfAllApps	The number of PVs of this App accounts for the, proportion of the total number of PVs of all Apps
	AppUVs	The number of this App's users
	UVProportionOfAllApps	The number of UVs of this App accounts for the, proportion of the total number of UVs of all Apps
	PVProportionOfThisCategory	The number of PVs of this App accounts for the, proportion of the total number of PVs of this category that this App belongs to
	UVProportionOfThisCategory	The number of UVs of this App accounts for the, proportion of the total number of UVs of this category that this App belongs to
	CategoryPVProportionOfAllApps	The number of PVs of this category that this App belongs to accounts for the, proportion of the total number of PVs of all Apps
	CategoryUVProportionOfAllApps	The number of UVs of this category that this App belongs to accounts for the, proportion of the total number of UVs of all Apps
User-App related	UserUseThisAppActiveHourRatio	The number of hours that the user use this App accounts for the, proportion of the total hours of the user online
	UserUseThisAppActiveHourAccountForThisCategory	The number of hours that the user use this App accounts for the, proportion of the total hours of the user use this category that this App belongs to
	UserUseThisAppPVsRatio	The number of PVs that the user use this App accounts for the, proportion of the total PVs of the user online
	UserUseThisAppPVsAccountForThisCategory	The number of PVs that the user use this App accounts for the, proportion of the total PVs of the user use this category that this App belongs to
User-category related	UserUseThisCategoryActiveHourRatio	The number of hours that the user use this category accounts for the, proportion of the total hours of the user online
	UserUseThisCategoryPVsRatio	The number of PVs that the user use this category accounts for the, proportion of the total hours of the user online

3.3.3 Original NFP-based Features

In addition to the above features, we have extracted some other features that are related to neither bipartite graph nor projected graph. As shown in Table 4, in order not to lose time and visiting information in the original NFP data, we use this information to calculate some other features based the original NFP data. These features contain four parts: the first part is for the user, reflecting the user's entire behavior, regardless of the specific App; the second part is for App, which reflects the characteristics of App itself and the overall use of it, regardless of the specific user; the third part takes a pair of User-App as the statistical object, displays the related characteristics between user and App; the fourth part takes a pair of User-Category as the statistical object, displays the related characteristics between user and App category.

4 Experiments and Results

4.1 Experiment Dataset

We choose the NFP data for 4 consecutive days in January 2017 as our experiment data. There are more than 25000 users using 54 Apps belonging to 19 categories every day in this data shown in Table 5.

Table 5. The dataset

Time	day1	day2	day3	day4
Users	25413			
Apps	54			
Links	193045	190013	193617	190033

Table 6. Confusion Matrix

Real	Prediction	
	Link	Not link
Link	TP	FN
Not link	FP	TN

We constructed the user-App bipartite network and calculated the three-category features for each day. The day1's data is used as training features and the day2's data is used as training labels. Besides, both the day2's data and day3's data are used as test features for better evaluate our model, corresponding to the day3's data and the day4's data as a test labels.

Figure 5 shows the degree distribution of the users and the Apps of each day. It can be seen that most people use about 10 Apps a day on average and a small number of people use only a small amount of App or a lot of App. The degree distribution of App follows a power law.

Figure 6 shows the Apps' PV, UV distribution of a day. We can see that the two curves have the similar trends during a 24 h duration, i.e. the PV/UV approaches the lowest point at 2–5 am while remaining at the peak point from 10 am to 8 pm. This is reasonable and it confirms that App usage fluctuates in a similar way as people's daily activity patterns. The number of the daytime was superfluous at night that conform to people's daily routines. There is a low

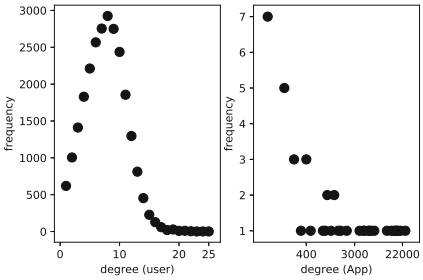


Fig. 5. Degree distribution of users and Apps

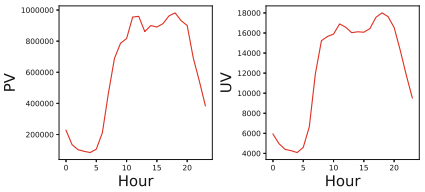


Fig. 6. Daily PV/UV distribution of Apps

in the time of day at noon which may be people will break at noon that caused the decrease of App usage.

Figure 7 shows the PV ratio of all categories, the category of *Communication and chat* takes up a big part in the diagram that means there are a lot of people use this category's App such as *WeChat*. The category of *Personal tools* has the smallest proportion because this category's App such as *Perpetual Calendar* is not so popular and people may not use it many times a day. It is interesting that the *music audio* has a relatively small ratio which does not seem to be in line with people's perception. This is because listening to music is a long process, and people don't click on it many times.

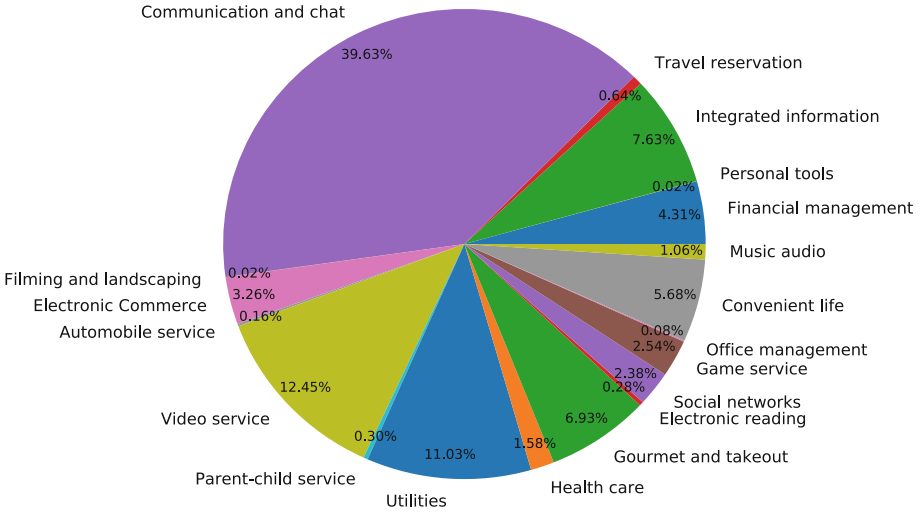


Fig. 7. The PV ratio of App categories

4.2 Evaluation Methods and Matrics

We mainly train two models (i.e. LR, XGB) through a combination of different features. Then we evaluate each model with the test data. We can divide the combination of real results and the predict results into four cases: true positive (TP), false positive (FP), true negative (TN), false negative (FN). The Confusion matrix is defined in Table 6. We use the accuracy, precision, recall and F1-value to evaluate our prediction models.

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{7}$$

$$precision = \frac{TP}{TP + FP} \tag{8}$$

$$recall = \frac{TP}{TP + FN} \tag{9}$$

$$F1\text{-}value = \frac{2 * TP}{2 * TP + FN + FP} \tag{10}$$

To simplify, we represent Bipartite Network-based features as BN, Projection Network-based features as PN, Original NFP-based features as ONFP. BN means using all the features defined in Table 2, while BN with only $Score^{f}_{u,a}$ and BN with only $Score^{f}_{a,u}$ means only using part of features in Table 2. PN can modify the values of different n and m defined in Eqs. 3 and 4.

4.3 Result Analysis

The prediction results of different models with different feature combinations are shown in Table 7. It can be seen that the combination of three categories of features (i.e. bipartite network-based, projection graph-based, and original

Table 7. Results

Feature used	Model	Accuracy	Precision	Recall	F1	Feature used	Model	Accuracy	Precision	Recall	F1
BN (only $Score^{f}_{u,a}$)	LR	0.8762	0.9520	0.7923	0.8649	BN + ONFP + PN (n = 10, m = 1)	LR	**0.9134**	0.9238	0.9011	**0.9123**
	XGB	0.8807	0.9597	0.7948	0.8695		XGB	0.9069	0.9065	0.9073	0.9069
BN (only $Score^{f}_{a,u}$)	LR	0.8849	0.8912	0.8769	0.8840	BN + PN (n = 10, m = 4)	LR	0.9087	0.9099	0.9073	0.9086
	XGB	0.8825	0.8918	0.8707	0.8811		XGB	0.8965	0.9462	0.8408	0.8904
BN	LR	0.9041	0.9212	0.8837	0.9021	BN + ONFP + PN (n = 10, m = 4)	LR	0.9129	0.9221	0.9021	0.9120
	XGB	0.8896	0.9158	0.8581	0.8860		XGB	0.8970	0.9329	0.8555	0.8925
BN + ONFP	LR	0.9134	0.9255	0.8993	0.9122	BN + PN (n = 10, m = 10)	LR	0.9113	0.9172	0.9043	0.9107
	XGB	0.9051	0.9374	0.8682	0.9015		XGB	0.9113	0.9232	0.8973	0.9101
BN + PN (n = 10, m = 1)	LR	0.9061	0.9105	0.9009	0.9057	BN + ONFP + PN (n = 10, m = 10)	LR	0.9126	0.9198	0.9040	0.9118
	XGB	0.9057	0.9040	0.9079	0.9060		XGB	0.9111	0.9228	0.8973	0.9100

NFP-based) leads to a relative good prediction performance. For App usage prediction task, LR model is slightly better than XGB model. It is interesting that the combination of BN + ONFP and the combination of BN + ONFP + PN (n = 10, m = 1) have the same highest accuracy in LR. At the same time, the F1-value reaches the maximum in our experiment when we use the combination of BN + ONFP + PN (n = 10, m = 1) on LR. In general, the features that we calculated with various combinations have achieved a relatively good results in predicting App usage.

5 Conclusion

This paper aims at predicting App usage, i.e., whether a mobile user use an App in the future based on historical information of App usage of this user. We propose the App Usage prediction method based on link prediction. We construct the User-App bipartite network based on the network footprint data to extract some features from it as well as from the projection graphs. Meanwhile, we consider some other features including time and visiting information in the original NFP data. Finally, we combine these features in a variety ways to train our prediction models. The experimental results prove that the proposed feature combination and prediction method has good performance for App usage prediction.

For the next step, we want to add the weights to the bipartite network and consider the location information of the user as the other feature. We also plan to use these methods in a variety of scenarios, for instance, we can predict whether a user will use Game App or Video App at afternoon.

Acknowledgements. This work is supported by the National Natural Science Foundation of China under Grant No. 61601046 and No. 61171098, and is partially supported by the 111 Project of China under Grant No. B08004, and EU FP7 IRSES Mobile Cloud Project under Grant No. 612212.

References

1. Benchettara, N., Kanawati, R., Rouveirol, C.: Supervised machine learning applied to link prediction in bipartite social networks. In: 2010 International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 326–330. IEEE (2010)
2. Chang, Y.-J., Kao, H.-Y.: Link prediction in a bipartite network using Wikipedia revision information. In: 2012 Conference on Technologies and Applications of Artificial Intelligence (TAAI), pp. 50–55. IEEE (2012)
3. Chinta, K.K., Clark, K., Mani, A.: Cs224w project final report supervised link prediction in bipartite networks (2014)
4. Gao, M., Chen, L.: A projection based algorithm for link prediction in bipartite network. In: 2016 International Conference on Information System and Artificial Intelligence (ISAI), pp. 56–61. IEEE (2016)

5. Gupta, A.K., Sardana, N.: Naïve Bayes approach for predicting missing links in ego networks. In: 2016 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS), pp. 161–165. IEEE (2016)
6. Liao, Z.-X., Lei, P.-R., Shen, T.-J., Li, S.-C., Peng, W.-C.: AppNow: predicting usages of mobile applications on smart phones. In: 2012 Conference on Technologies and Applications of Artificial Intelligence (TAAI), pp. 300–303. IEEE (2012)
7. Liao, Z.-X., Lei, P.-R., Shen, T.-J., Li, S.-C., Peng, W.-C.: Mining temporal profiles of mobile applications for usage prediction. In: 2012 IEEE 12th International Conference on Data Mining Workshops (ICDMW), pp. 890–893. IEEE (2012)
8. Liao, Z.-X., Li, S.-C., Peng, W.-C., Philip, S.Y., Liu, T.-C.: On the feature discovery for app usage prediction in smartphones. In: 2013 IEEE 13th International Conference on Data Mining (ICDM), pp. 1127–1132. IEEE (2013)
9. Liao, Z.-X., Pan, Y.-C., Peng, W.-C., Lei, P.-R.: On mining mobile apps usage behavior for predicting apps usage in smartphones. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, pp. 609–618. ACM (2013)
10. Lü, L., Zhou, T.: Link prediction in complex networks: a survey. Phys. A Stat. Mech. Appl. **390**(6), 1150–1170 (2011)
11. Luo, Y., Liu, Q., Wu, W., Li, F., Bo, X.: Predicting drug side effects based on link prediction in bipartite network. In: 2014 7th International Conference on Biomedical Engineering and Informatics (BMEI), pp. 729–733. IEEE (2014)
12. Ma, Y., Cheng, G., Liu, Z., Liang, X.: Link prediction based on clustering information in scientific coauthorship networks. In: IEEE International Conference on Data Science in Cyberspace (DSC), pp. 668–672. IEEE (2016)
13. Wu, J.-H., Zhang, G.-J., Ren, Y.-Z., Zhang, X.-Y., et al.: Exploiting neighbors' latent correlation for link prediction in complex network. In: 2013 International Conference on Machine Learning and Cybernetics (ICMLC), vol. 3, pp. 1077–1082. IEEE (2013)
14. Wu, X., Zhu, Y.: A hybrid approach based on collaborative filtering to recommending mobile apps. In: 2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS), pp. 8–15. IEEE (2016)
15. Xia, S., Dai, B.T., Lim, E.-P., Zhang, Y., Xing, C.: Link prediction for bipartite social networks: the role of structural holes. In: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 153–157. IEEE (2012)
16. Xu, Y., Lin, M., Lu, H., Cardone, G., Lane, N., Chen, Z., Campbell, A., Choudhury, T.: Preference, context and communities: a multi-faceted approach to predicting smartphone app usage patterns. In: Proceedings of the 2013 International Symposium on Wearable Computers, pp. 69–76. ACM (2013)
17. Zhu, K., Zhang, X., Xiang, B., Zhang, L.: Exploiting user context and network information for mobile application usage prediction. In: Proceedings of the 7th International Workshop on Hot Topics in Planet-Scale MObile Computing and Online Social NeTworking, pp. 25–30. ACM (2015)

Sentiment Classification of Reviews on Automobile Websites by Combining Word2Vec and Dependency Parsing

Feifei Liu(✉), Fang Wei, Ke Yu, and Xiaofei Wu

Beijing University of Posts and Telecommunications, Beijing, China
liufeifei@bupt.edu.cn

Abstract. The online product reviews become one of the most useful and vast information sources for guiding customers' decisions and helping the companies improve the quality of the products and services. Therefore, It is valuable to automatically identify sentiments from comment texts, which is concerned with the Sentiment Classification. In this paper, we propose a novel machine learning-based method called ADSSR to classify the sentiments of reviews on popular automobile websites in China. We extract the features based on dependency parsing which can reveal the syntactic structure of the sentence, to avoid obtaining the same vectors for sentences that contain the same words but a different grammatical structure. In order to reduce the dimensionality of the feature vectors and keep the contributions of the low-frequency words, we obtain the distributed vectors learned by Word2Vec and group the semantic similar words in a cluster through the K-means to obtain the pairs of each word and its corresponding cluster, and then replace every word with its corresponding cluster label. Experiments show the efficiency of the proposed sentiment classification method.

Keywords: Sentiment classification · Word2vec
Dependency parsing · K-means

1 Introduction

With the increasing popularity of the e-commerce, in order to improve the quality of service and attract more consumers, many e-commerce websites allow consumers to comment on their products. Users generally want to learn knowledge about the quality and reputation of the product to decide whether to buy. For companies, it is important to improve their products and discover new opportunities in the market. The product reviews become one of the most useful and vast information sources for guiding customers' decision and helping the companies improve the quality of the products and services. Therefore, it is valuable to automatically identify sentiments from review texts, which is concerned with the Sentiment Classification. There are many automobile websites, such

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 206–221, 2018.
https://doi.org/10.1007/978-3-319-73830-7_21

as Autohome[1], Yiche[2], PCauto[3], Xcar[4] in China, providing information about automobiles, dealers and automobile reviews, with the increasingly prosperous automobile industry of China. In this paper, we identify sentiments from review texts on automobile websites based on Sentiment Classification to help users and companies know all aspects of the product more intuitively.

Sentiment classification is a process that associates a given text with one or more classes based on the characteristics (contents or attributes) of the text under a predefined classification system [1]. There are two general approaches of sentiment classification: the method based on sentiment lexicon and the method based on machine learning [8]. The former calculates the sum of the number of sentiment words in the text to determine the sentiment polarity according to the labeled positive and negative words in the lexicon. The latter uses feature vectors to represent texts and classify the vectors by classifier. The purpose of our paper is to propose a novel machine learning-based method to classify the sentiments of reviews on popular automobile websites in China. Features play a fundamental role in sentiment classification. Currently, in order to improve the effect of the method based on machine learning, people are committed to proposing more efficient classifiers and extracting more useful features. In the papers which are focusing on feature extraction, the Bag-of-Words is often used as baseline method, but in sometimes can not properly capture more complex linguistic phenomena.

In this paper, we crawled down all automobile reviews from automobile websites in China and identify sentiments from review texts. To avoid obtaining the same vectors for sentences that contain the same words but a different grammatical structure, we extract the features based on dependency parsing which can reveal the syntactic structure of the sentence.

In order to reduce the dimensionality of the feature vectors and keep the contributions of the low-frequency words, we propose a method called ADSSR based on Word2Vec and K-means. Word2Vec can learn the distributed vectors of words in the high dimensional vector space and calculate the cosine distances between words, so the similarity of the words can be described by the distance of the vectors in the vector space. People always use it to find the relative words of the basic dictionary words. In our paper, We group the semantic similar words in a cluster by using K-means to cluster the similar distributed vectors learned by Word2Vec.

The rest of this paper is organized as follows. Section 2 reviews related works about sentiment classification. Our method based on dependency paring and Wor2Vec and key steps are described in detail in Sect. 3. The experimental results are illustrated in Sect. 4. Finally, Sect. 5 summarizes this paper.

[1] http://www.autohome.com.cn.
[2] http://www.bitauto.com.
[3] http://www.pcauto.com.cn/.
[4] http://www.xcar.com.cn/.

2 Related Work

Sentiment classification plays an important role in automatically identifying sentiments from review texts. There are two general approaches of sentiment classification: the method based on sentiment lexicon and the method based on machine learning. Several studies of the two general approaches of sentiment classification have already been published. In the former direction, Takamura proposed a method for extracting semantic orientations of words and a criterion for parameter selection on the basis of magnetization [20]. Taboada presented a lexicon-based approach to extract sentiment from text using dictionaries of words annotated with their semantic orientation (polarity and strength) [21]. In the latter direction, Pang applied the method based on machine learning, such as Naive Bayes (NB), Support Vector Machine (SVM) and Maximum Entropy (ME) to sentiment classification task [4]. He did experiments with features based on unigrams, bigrams, adjectives and so on. Li proposed a novel active learning approach, named co-selecting, to reduce the annotation cost for imbalanced sentiment classification [5]. Turney presented a simple unsupervised learning algorithm for classifying reviews [6].

Focusing on feature analysis of the machine learning-based method, Apoorv Agarwal introduced POS-specific prior polarity features and presented 3 models for sentiment classification: unigram model, feature based model and tree kernel based model [7]. Zhong Zhai extracted sentiment words, substring, substring-groups and key-substring-groups as features, in order to effectively select different types of features to333 improve sentiment classification performance [9]. Considering semantic information and leveraging the syntactic relationships between n-gram features, Abbasi proposed a rule-based multivariate text feature selection method called Feature Relation Network (FRN) [10].

Dependency parsing, which was presented by French linguist, Teensier in 1959 [2], use the grammar system including the phrase structure grammar and dependency grammar, to generate the dependency grammar for a sentence. Changqin Quan proposed a method, which combines sentiment lexicon and dependency parsing to determine the sentiment orientation and the positive or negative attitudes of the topic [11]. Guo Fu-liang proposed a new Improved Dependency Parsing to analyze micro-blog sentiment orientation on the basis of analyzing the peculiarity of micro-blog texts [12]. The dependency parsing is used to get the objects and sentiment words. Lin Li proposed a new informative review identification method based on dependency parsing and sentiment analysis [13]. He used the vector of the dependency relations parsed in the sentence as the features.

Word2Vec released by Google in 2013 is a tool based on deep learning [19]. It can learn the distributed vectors of words in the high dimensional vector space which can be used for many applications, particularly word prediction and translation. There are several studies applying the distributed vectors to sentiment analysis. Xue presented a novel model to build a Sentiment Dictionary using Word2Vec tool [14]. He used the method based on semantic similarity measuring the semantic distance between Weibo words and basic dictionary

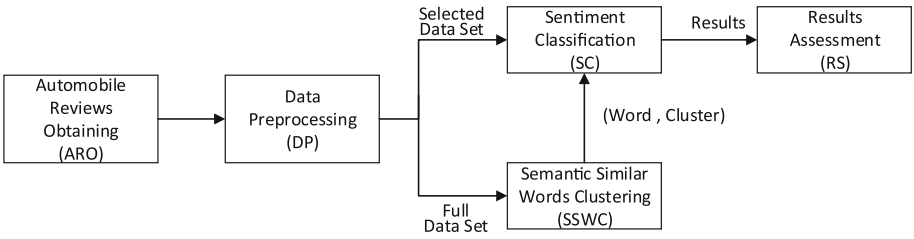


Fig. 1. The framework of method ADSSR.

words to calculate the semantic orientation. Su clustered the similar features together and learn the word representations as candidate feature vectors by using Word2Vec [15]. Lu extended the sentiment dictionaries from NTU and HowNet by finding the relative words of the basic dictionary words based on the Word2Vec tool [16]. Zharmagambetov proposed a modern approach to the task of sentiment analysis of movie reviews by using deep learning recurrent neural networks and decision trees [17]. His paper focuses on using Word2Vec model for text classification. However he did not consider the syntax structure of the texts.

There are few works considering to combine the Word2Vec and Dependency Parsing. In our paper, we extracted the features based on Dependency parsing to reveal the syntactic structure of the sentence. People usually use Dependency parsing and Word2Vec for lexicon-based method to find new sentiment words to expand the sentiment lexicon, however, we extracted the dependent word pairs based on dependency parsing in the sentence as the features of the binary classifier. In order to reduce the dimensionality of the feature vectors, we replaced every word with its corresponding cluster. To obtain the pairs of each word and its corresponding cluster label, we trained the word vectors with Word2Vec tool and grouped the semantic similar words in a cluster by K-means. Our method ADSSR will be described in detail in next section.

3 ADSSR Method

3.1 Framework of ADSSR

To help users and companies know all aspects of the product more intuitively, in this paper, we crawled down the product reviews from the automobile website and identified sentiments from review texts based on Sentiment Classification. Figure 1 illustrates the general framework of our work. The framework can be divided into five modules: Automobile Reviews Obtaining (ARO), Data Preprocessing (DP), Sentiment Classification (SC), Semantic Similar Words Clustering (SSWC) and Results Assessment (RA). In module ARO, we crawled the reviews form the autohome website. To get the data set for machine learning, we labeled the positive and negative samples, and then we defined the Full Data Set and the

Selected Data Set in module DP. In module SC, we mainly constructed the feature vectors, trained binary classifier and classified the texts into positive class or negative class. In order to obtain the pair of each word and its corresponding cluster, in module SSWC, we trained the word vectors with Word2Vec tool on the Full Data Set and grouped the semantic similar words in a cluster. Finally, to verify the effectiveness of our method, we compared our method with others and evaluated the experimental results with accuracy, precision, recall and F1 averaged from 5 runs in module Results Assessment. We explained the modules respectively in the following sections.

3.2 Automobile Reviews Obtaining

Among mangy automobile websites, Autohome is the most popular and valuable automobile website in China, so we crawled down all product reviews, more than 600,000 reviews, from the Autohome website, by using web crawler in Python[5] language. The web pages of automobile reviews shown in Fig. 2 contain the information about the user ids, automobile and the automobile reviews. Most of the consumers write the review texts and give scores from the following 8 aspects: Space, Power, Control, Fuel Consumption, Cost-effective, Interior, Appearance and Comfort. Each review consists of the following fields in each aspects:

1. Review text: Text of the review content in this aspect on the right of the web page.
2. Grade score: The grade score in this aspect using stars by the consumer on the left of the web page.

We intend to identify sentiments from review text and label the review text to positive or negative class according to its corresponding grade score, so we crawled down the review texts and the grade scores of each product review. According to the aspects of review texts and grade scores, each review is stored as follows (Fig. 3).

3.3 Data Preprocesing

Considering the structure of the reviews, we divided the reviews crawled down in the previous section into 8 data sets according to the aspects of the review texts. Each entry in every data set consists of the grade score given by the consumer and the review text. Figure 4 shows the entries in Space data set. What need to be emphasized is our following work is implemented in each data set. We labeled the review text to positive or negative class according to its corresponding grade score. The Full Data Set is made up of all positive samples and negative samples. The number of positive samples is much larger than the number of negative samples, so we randomly selected positive samples and negative samples with the same number, and then made up the Selected Data Set with them.

[5] https://www.python.org/.

Fig. 2. The web page of the automobile reviews on Autohome.

{"grade scores": {"空间": "4", "动力": "5", "操控": "5", "油耗": "5", "舒适性": "4", "外观": "5", "内饰": "5", "性价比": "4"}, "review texts": "【空间】我微胖一米七三，坐前排调好座椅空间够用，后排坐两人妥妥的，做三个就挤了，只能说还行。小车在我们这里小区停车也有一定的便利性，总体还可以。【动力】舍得给油的话动力还是很足的，家用够了。起步油给小了感觉特别肉，有说是一种保护，不然新手有风险，大点油门就好。有帖子说S档起步，新手基本上没用过S档，以后有机会试试。不开赛车动力很满意。【操控】新手开的车少，感觉很好开，开着舒服。【油耗】新手开车这个油耗很满意了。就是搞不懂车载电脑油耗表2为什么显示会有时候十五六升。实际300块油跑了近600公里续航里程显示还能跑七十公里。【舒适性】座椅的舒适性很好，朋友、同事坐都说挺舒服的。就是空间上身材高大或者常载人多的朋友不适合。【外观】最满意的一点【内饰】精致，买车的时候看过不少车，小3的内设的最满意的。简洁的操控台看上去很舒服【性价比】对比不少，懂的也不多，反正自己觉得还好，有句话叫有钱难买我乐意，跟找媳妇一样的，看上了就不要太斤斤计较，不然活得太累。"}

Fig. 3. The storage format of the review texts.

```
root@ant-dell:/home/lff# head -n 10 Space.txt
4##*##我相信选择A3的车友不会冲着它的空间来的，可能会是第二辆第三辆车，所以在这里我就简单说一下吧，空间足够，后排空间两边座位挺舒服空间尚可，但是中间座位基本坐不了成年人，太不舒服。后备箱空间不大，但是两厢车的好处就是放倒后排座椅后超大的装备空间，这也是我选择两厢最重要的地方，没事和心爱的人周末来个小短途旅行是不是很惬意哦？OK
4##*##驾驶室的空间基本上就是我不满意的地方，坐在上面竟然还能碰着头，难道是4S店里的加装造成车座的过分上移？后排空间优秀，后排空间坐过三个人，他们也没抱怨说空间小，挤不开，说明后面的乘坐空间还是可以的。后备箱很大，我至今没放满过。
4##*##空间上只能说比较充足地满足二人世界或者三小之家使用。前排驾驶位空间很到位，可调节空间也选择多；副驾与主驾差不多等级空间，两者的头部高度与脚踩部等都能保障（本人172，60，女友也是娇小身材，两人驾驶与乘坐都能有足够大小变化调整空间）；后排座位只能有一般两厢车空间，甚至只能在中等程度，两人坐很充足，加上扶手放下还比较可以，如若三个成年人就比较拥挤，高度与宽度很紧凑，中间位的起拱比较高，脚部不会舒适；后备箱小家庭使用还是基本保障，空间切割布局也很平整，甚至下部备胎处也有可以备万不得已的小空间可以塞放，其余的杂物储物等空间保持这款车给我的简洁感觉，数量不多，但每个的空间也只是基本到位，小巧仅此。
4##*##一汽大众奥迪a3这款车的空间，比较一般没有特别大的空间，前排空间比较满意，坐里面满是起码没有压抑感，后排空间比较小，不过我后面经常不载人，说说这款车的后备箱吧，空间吧，有一次我跟老婆去旅游的时候，把东西都放倒后备箱，不过后背箱刚好够用，说不上来大，也不算太小，如果家用的话，应该是没问题的，能全部放进去。
```

Fig. 4. The top 10 entries in space.

3.4 Sentiment Classification

The module Sentiment Classification (SC) and Semantic Similar Words Clustering (SSWC) are shown in Fig. 5.

In module Sentiment Classification, we did the following steps for every review text in Selected Data Set: segment each review text in the Selected Data

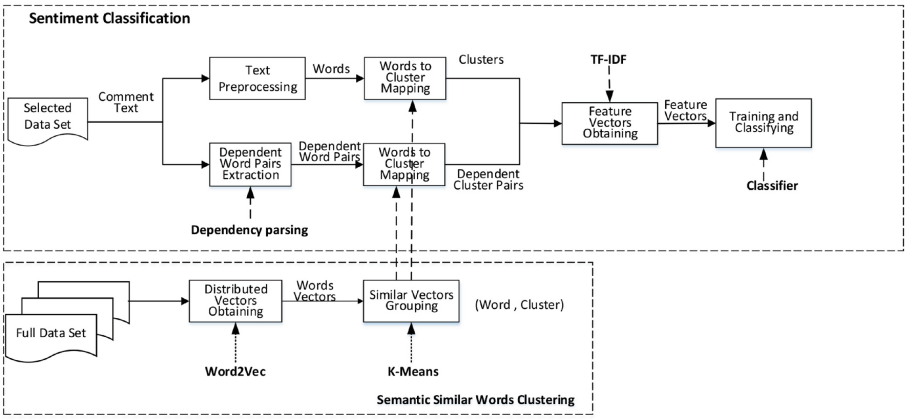


Fig. 5. The module SC and SSWC.

Set into words and delete the stop words, extract the dependent words pairs based on dependency parsing, replace every word with its corresponding cluster obtained through the module Semantic Similar Words Clustering. The next step is applying TF-IDF on clusters and dependent clusters pairs of all review texts to generate weight factor to obtain the feature vector for every review text. Finally, we trained the binary classifier and classify every review text into positive class or negative class.

In module Semantic Similar Words Clustering, we trained the word vectors with Word2Vec tool on the Full Data Set, grouped the semantic similar words in a cluster to obtain the pair of each word and its corresponding cluster.

The detail of our modules will be introduced in the following subsections.

3.4.1 Text Preprocessing

The Language Technology Platform (LTP)[6], is an open-sourced Chinese natural language processing system developed in the Research Center for Social Computing and Information Retrieval, HIT. We used *Pyltp* library which is the Python package for the LTP to preprocess the review texts (segment every review text into words and delete the stop words).

3.4.2 Dependent Word Pairs Extraction

Dependency Parsing (DP) reveals the syntactic structure by analyzing the dependencies between language components of the sentence [18]. It analyze the grammatical elements in the sentence and establish the dependency relation between the language components. The *Pyltp* library can return the dependency

[6] https://github.com/HIT-SCIR/ltp.

relation between two words. We extracted the dependent word pairs of every review text based on dependency parsing. And then, we expressed the original sentence with the words and dependent word pairs in the sentence.

An example of Dependency Parsing obtained by *Pyltp* is shown in Fig. 6. As we can see from the figure, the arrows, from the control word to subordinate word, label the dependency relation by dependency relation symbol. 15 dependency relations in Chinese established by LTP[7] are shown in Table 1.

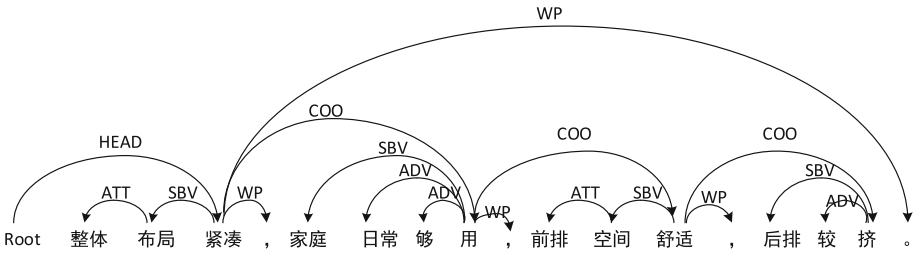


Fig. 6. An example of dependency parsing obtained by *Pyltp*.

Table 1. The dependency relations in Chinese established by LTP

Tag	Relation
SBV	subject-verb
VOB	verb-object
IOB	indirect-object
FOB	fronting-object
DBL	double
ATT	attribute
ADV	adverbial
CMP	complement
COO	coordinate
POB	preposition-object
LAD	left adjunct
RAD	right adjunct
IS	independent structure
WP	punctuation
HED	head

[7] http://www.ltp-cloud.com/intro/.

In order to reveal the syntactic structure, we want to extract the dependent word pairs (the control word and the subordinate word) in the sentence as the features instead of the traditional Bag-of-Words, so the useful word pairs should be selected. Firstly, the symbol `WP` representing the punctuation is not considered. By observing a large number of cases, we find the symbol `HED` always marks the first verb in the sentence having a lot of verbs without the core verb. The symbol `COO` always marks the verbs, that can increase the frequencies of verbs. So the symbol `HED` and `COO` should not be considered. An example with the dependency relations after selected are shown in Fig. 7.

Fig. 7. An example with the useful dependency relations after selected.

We express the original sentences of the review texts with the words and dependent word pairs in the sentence as the following vector:

$$\mathbf{S} = (w_1, w_2, \ldots, w_n, (w_{d_{11}}, w_{d_{12}}), (w_{d_{21}}, w_{d_{22}}), \ldots, (w_{d_{m1}}, w_{d_{m2}}))$$

Among the parameters, $w_i, 0 < i \leq n$, refers to the word in the sentence. The parameter $n \in \mathbf{R}$, refers to the number of the words. $(w_{d_{i1}}, w_{d_{i2}}), 0 < i \leq m$, refers to the dependent word pairs in the sentence. The parameter $m \in \mathbf{R}$, refers to the number of the dependent word pairs.

3.4.3 Words to Cluster Mapping

We used the traditional vector space model (VSM) to represent each review text and used the words and the dependent word pairs of all sentences in the review text as the features. Even if we used the high-frequency words and high-frequency dependent word pairs as the features, it will lead to the high dimension of the feature vectors because of the diversity of words. In order to reduce the dimensionality of the feature vectors and keep the contributions of the low-frequency words, we grouped the semantic similar words in a cluster to obtain the pair of each word and its corresponding cluster in the module Semantic Similar Words Clustering and then replaced each word in the sentence with its corresponding cluster label. The sentences of the review texts after word to cluster mapping is shown as the following vector:

$$\mathbf{S} = (c_{w_1}, c_{w_2}, \ldots, c_{w_n}, (c_{w_{d_{11}}}, c_{w_{d_{12}}}), (c_{w_{d_{21}}}, c_{w_{d_{22}}}), \ldots, (c_{w_{d_{m1}}}, c_{w_{d_{m1}}}))$$

Among the parameters, $c_{w_i}, 0 < i \leq n$, refers to the cluster of the w_i in the sentence. The parameter $n \in \mathbf{R}$, refers to the number of the words.

$(c_{w_{d_{i1}}}, c_{,w_{d_{i2}}}), 0 < i \leq m$, refers to the dependent cluster pairs in the sentence. The parameter $m \in \mathbf{R}$, refers to the number of the dependent word pairs.

3.4.4 Feature Vectors Obtaining

We used the traditional vector space model (VSM) to represent each review text and used the clusters and the dependent cluster pairs of all sentences in the review text as the features. We ignore all clusters and the dependent cluster pairs with total frequency in the Selected Data Set lower than 3. We used the tf-idf value as the feature weight and defined the feature vector of each review text as the following vector:

$$\mathbf{D} = (\omega_1, \omega_2, \omega_3, \ldots, \omega_{d-1}, \omega_d)$$

The parameter $\omega_i \in \mathbf{R}, 0 < i \leq d$, refers to the tf-idf value of the feature in the review text, and the parameter $d \in \mathbf{R}$ refers to the number of the features of the feature vectors.

$$\omega_i = tf_i \times \log \frac{N}{n_i} \tag{1}$$

In the formula of calculating the tf-idf value ω_i to the feature t_i in the review text D above, tf_i refers to the term frequency of the feature t_i in the review text D, N refers to the total number of review texts in the training samples, n_i refers to the number of review texts that contain the feature t_i.

3.4.5 Training and Classifying

We randomly selected 80% of the positive samples and 80% of the negative samples in the Selected Data Set as the training set. The rest of the review texts are selected as the testing set.

After we getting the feature vectors of the training set, we used them as the input of the binary classifier to train the model. And then we obtained the feature vectors of the testing set and classify them into positive class or negative class using the model. In our experiments, we respectively trained support vector machines (SVM) with linear kernel, naive bayesian (NB), decision trees (DT), random forests (RF) with 10 trees, in module Sentiment Classification to classify the review texts, all using the feature vectors obtained from Sect. 3.4.4 as inputs.

3.5 Semantic Similar Words Clustering

The core of this module is to obtain the pair of each word and its corresponding cluster. In order to obtain the pairs we used Word2Vec on the Full Data Set to get the distributed word vectors of all the words followed by using K-means clustering algorithm to group similar vectors. The distributed word vector of the word trained by the Word2Vec model can describe the similarity of the words by the distance of the vectors in the vector space, so we can group the semantic similar words in a cluster through the K-means clustering the vectors by distance.

3.5.1 Distributed Vectors Obtaining

Word2Vec is a deep learning implementation that learns the distributed vectors of the words in the data set by learn the co-present words, published by Google in 2013. There are two model architectures [3] of Word2Vec: The continuous bag of words (CBOW) architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word.

In our experiments, the `gensim`[8] python library which includes in Word2Vec model was used to learn the distributed vectors of the words on the Full Data Set of each aspect. We ignored all words with total frequency in the Full Data Set lower than 20 by setting the parameter *min-count* as *20*. The parameter *sg* was set as *1*, which means the Skip-gram architecture shown as Fig. 8 was used.

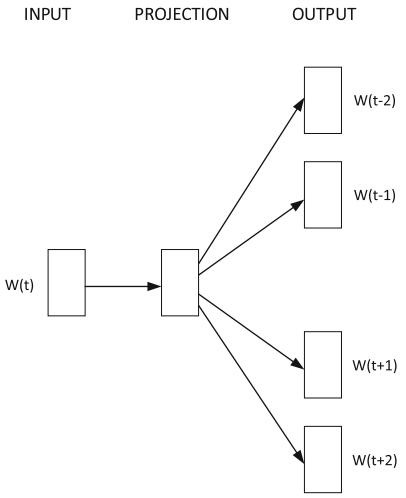


Fig. 8. The Skip-gram architecture of Word2Vec.

The objective of the Skip-gram [19] model is to maximize the average log probability shown as following:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \tag{2}$$

The parameter c is the size of the training review text, and T is the number of the words in the training review text. The parameter j (the parameter *window* of the Word2Vec model) is set as *5* by default, which means the maximum distance between the current and predicted word within a sentence is five. The basic Skip-gram formulation defines $p(w_{t+j}|w_t)$ using the softmax function and uses Noise Contrastive Estimation (NCE) to approximate the full softmax by

[8] http://radimrehurek.com/gensim/.

setting the parameter *hs* as *0*. The rest parameter of the Word2Vec model is set by default e.x. the parameter *size* is *100* which means the dimensionality of the word vectors is 100.

3.5.2 Similar Vectors Grouping

After obtaining the vectors of the words in the Full Data Set, we used them as the inputs of the K-Means model built in *sklearn*[9] library of python to cluster the similar vectors. After a series of experiments, we observed if the parameter $n\_clusters$ was set $n_{cluster}$ as n_{words} mod 10 can get the better result, where n_{word} is the number of the words with total frequency more than 20 in the Full Data Set. When the clustering have converged, we can obtain the pair of each word and its corresponding cluster according the pair of each word vector and its corresponding cluster.

3.6 Results Assessment

We compared our method with other methods to verify the effects of the dependent word pairs obtained based on Dependency parsing and the words to clusters mapping. We conducted our experiments 5 runs on each aspect and evaluated the experimental results with accuracy, precision, recall and F1 averaged from 5 runs.

4 Experiments Results

4.1 Experimental Dataset

We crawled down all product reviews, more than 600,000 reviews, from the Autohome website, and divided the review texts and grade scores into different data sets according to the aspects. We labeled the review text which grade score is 5 as the positive sample and the review text which grade score ≤ 2 as negative sample. We show the results of 3 data sets in 3 aspects: Space, Interior and Comfort. There are 8,900 positive samples and 8,900 negative samples in the Selected Data Set of each data set. We randomly selected 80% of the positive samples and 80% of the negative samples in the Selected Data Set as the training set. The rest of the review texts were selected as the testing set. The Table 2 shows the scale of data sets, positive samples, negative samples and the Full Set Data in different aspects.

4.2 Methods and Metrics

In order to verify the effects of the dependent word pairs obtained based on Dependency parsing and the words to cluster mapping, we chose the method

[9] http://scikit-learn.org/stable/.

Table 2. The scale of data sets in different aspects

	Space	Power	Interior	Appearance
Data set	554953	554860	546454	554752
Positive samples	300461	168225	174043	327769
Negative samples	8975	14519	20921	4429
Full Data Set	309436	182744	194964	332198
	Comfort	Control	Cost-effective	Fuel consumption
Data set	554830	554889	546041	31091
Positive samples	160170	283990	287414	11648
Negative samples	20666	6265	8967	1282
Full Data Set1	180836	290255	296381	12930

called ADSSR-BOW which uses feature vectors based on Bag-of-Words without dependent cluster pairs and the method called ADSR (method ADSSR without model Semantic Similar Word Clustering) which uses feature vectors without words to cluster mapping as inputs to compare with our method ADSSR. In our experiments, we respectively trained support vector machines (SVM) with linear kernel, naive bayesian (NB), decision trees (DT), random forests (RF) with 10 trees, in module Sentiment Classification to classify the review texts, all respectively using the feature vectors constructed by ADSSR-BOW, ADSR and ADSSR as inputs. We conducted our experiments 5 runs on each aspect and evaluate the experimental results with accuracy, precision, recall and F1 averaged from 5 runs.

4.3 Results and Analysis

We show the results of 3 data sets in 3 aspects: Space, Interior and Comfort in Tables 3 and 4. As we can see from Table 3, ADSSR can reduce the dimensionality of the feature vectors compared with ADSR. In Table 4, the accuracy value of ADSSR in Space aspect with SVM is **0.89**, however the accuracy values of other methods are just 0.87 and 0.88. Summarizing all the results in the tables, we can see the values of accuracy, precision, recall and F1 of ADSSR in each aspect with each classifier are higher than the values of other methods, except the precision in Space aspect with NB. We can conclude that the ADSSR is more effective than the other methods. We add the dependent word pairs to the traditional bag-of-words which can avoid the same vectors for sentences that contain the same words but a different grammatical structure. It can reduce the dimensionality of the feature vectors and keep the contributions of the low-frequency words by grouping the semantic similar words and word to cluster mapping.

Table 3. The dimensionality for different method on a collection of classifiers and aspects

	Space	Interior	Comfort
ADSSR-BOW	800	500	500
ADSR	15379	14001	15205
ADSSR	13555	12101	13213

Table 4. The results for different method and classifiers in different aspects

		Space				Interior				Comfort			
		SVM	NB	DT	RF	SVM	NB	DT	RF	SVM	NB	DT	RF
Accuracy	ADSSR-BOW	0.87	0.85	0.78	0.83	0.83	0.83	0.76	0.81	0.83	0.81	0.75	0.79
	ADSR	0.88	0.88	0.80	0.84	0.85	0.85	0.76	0.81	0.85	0.84	0.76	0.81
	ADSSR	**0.89**	**0.88**	**0.80**	**0.84**	**0.86**	**0.86**	**0.78**	**0.82**	**0.85**	**0.85**	**0.77**	**0.81**
Precision	ADSSR-BOW	0.86	0.85	0.78	0.85	0.84	0.83	0.77	0.83	0.83	0.82	0.75	0.82
	ADSR	0.88	**0.88**	0.79	0.86	0.86	0.85	0.78	0.83	0.85	0.83	0.77	0.83
	ADSSR	**0.89**	0.87	**0.80**	**0.86**	**0.87**	**0.87**	**0.79**	**0.84**	**0.85**	**0.84**	**0.77**	**0.83**
Recall	ADSSR-BOW	0.87	0.85	0.78	0.79	0.82	0.82	0.76	0.79	0.82	0.81	0.74	0.75
	ADSR	0.89	0.88	0.80	0.81	0.83	0.84	0.77	0.79	0.84	0.85	0.76	0.78
	ADSSR	**0.89**	**0.88**	**0.80**	**0.81**	**0.85**	**0.85**	**0.78**	**0.79**	**0.85**	**0.85**	**0.77**	**0.80**
F1	ADSSR-BOW	0.87	0.85	0.78	0.82	0.83	0.83	0.76	0.81	0.82	0.81	0.75	0.78
	ADSR	0.89	0.88	0.80	0.83	0.84	0.84	0.77	0.81	0.85	0.84	0.76	0.80
	ADSSR	**0.89**	**0.88**	**0.80**	**0.84**	**0.86**	**0.86**	**0.78**	**0.82**	**0.85**	**0.85**	**0.77**	**0.81**

5 Conclusions

In this paper, we proposed the method ADSSR combining Word2Vec and Dependency Parsing to classify the sentiments of reviews on popular automobile website in China. Firstly, we extracted the words and the dependent words pairs obtained based on Dependency parsing in the sentence as the features. In order to reduce the dimensionality of the feature vectors, we replaced every word with its corresponding cluster. To obtain the pairs of each word and its corresponding cluster, we trained the word vectors with Word2Vec tool and grouped the semantic similar words in a cluster. The TF-IDF was applied on clusters and dependent clusters pairs of all review texts to generate weight factor to obtain the feature vector for every review text. Finally, we respectively trained SVM, NB, DT and RF to classify the review texts.

By comparing our results with other methods, we can see that our method is effective. The characteristics of our method is: revealing the syntactic structure of the sentence, reducing the dimensionality of the feature vectors and keeping the contributions of the low-frequency words.

Our future works will be related to proposing a more effective method than K-means to cluster the semantic similar words. We intend to apply ADSSR on solving the problem that the grade score can not reflect the sentiment of the review text correctly.

Acknowledgements. This work is supported by the National Natural Science Foundation of China under Grant No. 61601046 and No. 61171098, and is partially supported by the 111 Project of China under Grant No. B08004, and EU FP7 IRSES Mobile Cloud Project under Grant No. 612212.

References

1. Zong, C.: Statistics Natural Language Processing. Tsinghua University Press, Beijing (2008)
2. Tesnière, L.: Eléments de syntaxe structurale. Librairie C. Klincksieck (1959)
3. Mikolov, T., et al.: Efficient estimation of word representations in vector space. Computer Science (2013)
4. Bo, P., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Acl-02 Conference on Empirical Methods in Natural Language Processing, pp. 79–86. Association for Computational Linguistics (2002)
5. Li, S., et al.: Active learning for imbalanced sentiment classification. In: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 139–148. Association for Computational Linguistics (2012)
6. Turney, P.D.: Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Meeting on Association for Computational Linguistics, pp. 417–424. Association for Computational Linguistics (2002)
7. Agarwal, A., et al.: Sentiment analysis of Twitter data. In: The Workshop on Languages in Social Media, pp. 30–38. Association for Computational Linguistics (2011)
8. Dai, X., Prout, B.: Unlock big data emotions: weighted word embeddings for sentiment classification. In: IEEE International Conference on Big Data, pp. 3833–3838. IEEE (2017)
9. Zhai, Z., et al.: Exploiting effective features for Chinese sentiment classification. Expert Syst. Appl. **38**(8), 9139–9146 (2011)
10. Abbasi, A., et al.: Selecting attributes for sentiment classification using feature relation networks. IEEE Trans. Knowl. Data Eng. **23**(3), 447–462 (2011)
11. Quan, C., Wei, X., Ren, F.: Combine sentiment lexicon and dependency parsing for sentiment classification. In: IEEE/SICE International Symposium on System Integration, pp. 100–104. IEEE (2013)
12. Guo, F.L., Zhou, G.: Research on micro-blog sentiment orientation analysis based on improved dependency parsing. In: International Conference on Consumer Electronics, Communications and Networks, pp. 546–550. IEEE (2014)
13. Li, L.: Identification of informative reviews enhanced by dependency parsing and sentiment analysis. In: IEEE International Conference on Computer Communication and the Internet. IEEE (2016)
14. Xue, B., Fu, C., Zhan, S.: A study on sentiment computing and classification of Sina Weibo with Word2vec. In: IEEE International Congress on Big Data, pp. 358–363. IEEE (2014)
15. Su, Z., et al.: Chinese sentiment classification using a neural network tool Word2vec. In: International Conference on Multisensor Fusion and Information Integration for Intelligent Systems, pp. 1–6. IEEE (2014)
16. Lu, X., et al.: An approach to sentiment analysis of short Chinese texts based on SVMs. In: Control Conference, pp. 9115–9120. IEEE (2015)

17. Zharmagambetov, A.S., Pak, A.A.: Sentiment analysis of a document using deep learning approach and decision trees. In: Twelve International Conference on Electronics Computer and Computation, pp. 1–4. IEEE (2016)
18. Wang, C., Huo, L.: Research on sentence sensitivity analysis based on dependency parsing. World Academic Publishing
19. Mikolov, T., et al.: Distributed representations of words and phrases and their compositionality. In: International Conference on Neural Information Processing Systems, pp. 3111–3119. Curran Associates Inc. (2013)
20. Takamura, H., Inui, T., Okumura, M.: Extracting semantic orientations of words using spin model.In: Meeting on Association for Computational Linguistics, pp. 133–140. Association for Computational Linguistics (2005)
21. Taboada, M., et al.: Lexicon-based methods for sentiment analysis. Comput. Linguist. **37**(2), 267–307 (2011)

Data Quality Evaluation: Methodology and Key Factors

Ying Yang[1], Yuan Yuan[2], and Bo Li[2(✉)]

[1] Standardization Department, China Aero-Polytechnology Establishment, Aviation Industry of China, Beijing, China
yangyingpfy@163.com
[2] School of Computer Science and Engineering, Beihang University, Beijing, China
libo@act.buaa.edu.cn

Abstract. Data Quality Evaluation is becoming an institutionalized stage in data quality lifecycle. More and more practice is promoted by data management and user organization in specific fields especially in better informationalized application circumstance.

In order to improve the ability of data quality evaluation, the paper presents the key factors for data quality assessment and measurement. On the base of analyzing the main methodologies and standards on data quality management, the key factors includes objectives, general principles, characteristics, measurement function etc.

Keywords: Data quality evaluation · Measures · Standards · Characteristics

1 Introduction

Data Quality Evaluation is becoming an institutionalized stage in data quality lifecycle. More and more practice is promoted by data management and user organization in specific fields especially in better informationalized application circumstance.

Recently, the data quality assessment and measurement are utilized to improve the efficiency systematically. Methodology and key factors of data quality assessment and measurement related to the general concepts, terminology, objectives, procedure, model, principles, characteristics and measures.

Based on the achievement from the practice in authors' investigation project, the relative national or international standards focus on geographic, environmental and software field, the paper provide useful guidance for data quality evaluation scenarios over data lifecycle, and provide the fundamental input to Big Data service etc.

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 222–230, 2018.
https://doi.org/10.1007/978-3-319-73830-7_22

2 Data Quality Management Methodology

2.1 Total Data Quality Management (TDQM) in Department of Defense (DoD)

DoD TDQM methodology [1] is intended to validate data quality problems, indentify root causes, and improve data quality and utility. TDQM is a process to support database migration, promote the use of data standards, and improve in conformance to business rules. TDQM approach conforms to TQM methodologies, integrates management techniques, improvement efforts, and technical tools to create and sustain a culture that is committed to continuous improvement. To attain TDQM objectives, data quality work includes four essential tasks: definition, measurement, analysis and improvement.

Definition phase: data quality problem are defined by establishment of the scope and objectives of the data quality management project, and by judging whether criteria conform to relative standards.

Measurement phase: data quality measurements should present qualitative indexes and quantitative indexes, judging whether conformance to standards or not, and flag exceptions or suspicious datum.

Analysis phase: identification, priority and validation of quality issues are the common analysis procedure. Providing relative recommendations to solve the issues of data quality problem.

Improvement phase: improvement projects are defined and choosed the opportunities to implement them. Improving data quality may lead to changing data entry procedures, updating data validation rules, using standards that prescribe a uniform representation throughout the DoD.

2.2 ISO 8000 Data Quality Management (DQM)

ISO 8000 DQM methodology is a family standards focus on systematic solution. The series standards include part 001-99 general principle, part 100-199 master data, part 200-299 business data quality, part 300-399 product data quality. Under ISO 8000, part 150 A Framework for Data Quality Management is very important which presents three principles. Firstly, data quality management is not merely a technology implementation. Secondly, effective management is based upon a number of key processes. Lastly, achieving continuous improvement of the data quality is the key objective. The methodology is summarized as nine box model [2] describing the roles, process and responsibility.

Figure 1 illustrates the roles, responsibilities and functions of data quality management reflected stakeholders of data quality.

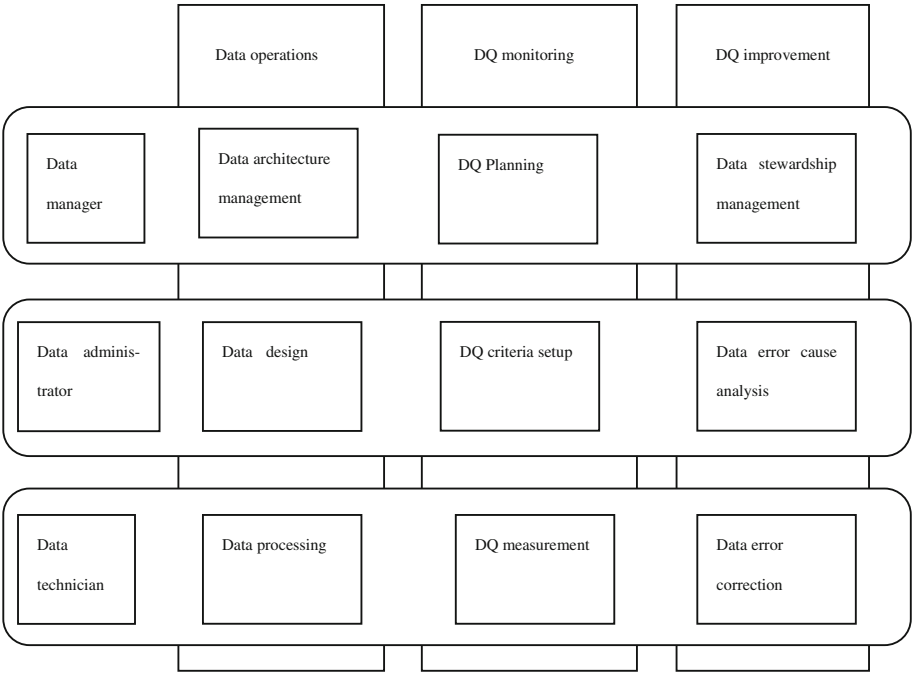


Fig. 1. The model of data quality management [2]

This model framework is divided into three processes and three roles. Three generic roles consist of data manager, data administrator and data technician. Three key processes and tasks are as follows:

Data operations consist of data architecture management, data design, data processing, focusing on the application.
Data quality monitoring consist of data quality planning, data quality criteria setup, data quality measurement, focusing on the assessment.
Data quality improvement consist of focus on data stewardship and flow management, data error cause analysis, data error correction, eliminating causes and correcting errors of data.

3 Key Factors of Data Quality Evaluation

3.1 The Objectives of Data Quality Evaluation

3.1.1 Compliance to Organizational Management Requirements

Satisfaction degree evaluation of data quality is provided according to measure the compliance to requirements. Baseline criteria are established within organizational information systems, then inspected periodically by specific regulations. The result of comparing with baseline is depended on attaining appropriate data or data sets.

3.1.2 Root Causes of Problems Within Organizational Environment

The evaluation offers opportunity to identify reoccurring issues damaging data quality. Some key questions can be answered just like: Do certain types of errors occur more frequently? What efforts are concentrated so as to get tgreater improvement in data quality? Determining root causes, there might be analysis from several points of view, including:

Process Problem: Data errors can be attributed to process problems. Checking the existing processes supported data entry, assignment and implementation of data quality responsibilities are suggested. These actions are recommend to correct deficiencies.
System Problem: Data problems frequently originate from system design deficiencies by poorly documented modifications, imperfect user training, or system beyond their original intent.
Policy and Procedure Problem: Data errors reveal lack of appropriate guidance, conflicting in existing directives, instructions and standards.
Data Design Problem: Data problem can be attributed to incomplete designation of database, errors of data values, incomplete specification of technical and business rules. For example, the inappropriate application of primary key constraints, referential integrity specifications, metadata specifications, null and not null data criterion etc.

3.1.3 Supporting Persistent Improvement of Data Quality

The recommendations of data quality improvement are categorized by four tasks reflecting multi-dimensions viewpoint of stakeholders. The recommendations are as follows:

Process Improvement: Concentration on the functional processes can change centralized data entry and data collection, in order to eliminate non value activities.
System Improvement: Data environment is depended on system. Software, hardware, and telecommunication changes can be utilized for improvement of data quality.
Policy and Procedure Improvement: Development of appropriate guidance for roles and responsibilities can be increase quality. Institutionalization such behaviors, for instance adding the periodic data quality examination into operating procedure, can promote business efficiency and data quality.
Data Design Improvement: Establishment and implementation of data standards aid in enhancing the overall data design capacity.

3.2 The General Principle of Data Quality Evaluation

3.2.1 The General Evaluation Principles

The general data quality evaluation principles include scientificity, intelligibility, objectivity and operationality.

Scientificity: Under the guidance of theories of quality management, data management, and standardization management, based on actual data collection, data acquisition, data storage and data exchange, the evaluation scheme shall be rational, well-arranged and operational.
Intelligibility: The evaluation scheme shall be concise, pertinent, plain and understandable, assuring that all concerned stakeholders and relevant personnels precisely comprehend, organize, and regulate the data quality assessment.
Objectivity: The evaluation scheme is supposed to be objective and reliable, authentically reflecting the fulfillment of need and the quality of data instead of blind enlargement and reduction of the evaluation scope.
Operationality: The evaluation scheme shall be specific, feasible for fear of unnecessary interferences. Data needed in assessment shall be easy to acquire and the evaluation process shall be objective, concise and convenient, maximizing the aid of automation tools and vehicles.

3.2.2 The Design Principles of Characteristics and Measures

The designation of data quality characteristics and measures conforms to the principle of integrity, openness, conciseness, and Orthogonality.

Integrity: In accordance with the characteristics and requirements of evaluation object, qualitative and quantitative meters shall be combined to form a comprehensive, systemic, and integral meter system.
Openness: Based on the actual situation, if accepted by the concerned stakeholders, the evaluation meter and weight can be adjusted, added and deleted and the evaluation process and its subprocess can be iterated.
Conciseness: Key points shall be simplified and highlighted, guaranteeing the selection of evaluation meter is sufficient and necessary.
Orthogonality: The design of meter and its corresponding data collection shall avoid ambiguity and overlap.

3.2.3 The Common Steps of Evaluation Procedure

The data quality evaluation tasks consist of four common steps [3]: definition of evaluation requirement, implementation of measurement, analysis result of assessment, implementation of improvement. It should be point out that iteration anyone step is permitted whenever needed, and implementation of improvement are performed by the data user organization after the prior evaluation according to current requirement and environment. Implementation of improvement response the effectiveness of the task of data quality evaluation to some extent.

Figure 2: illustrates the obligatory steps of data quality evaluation

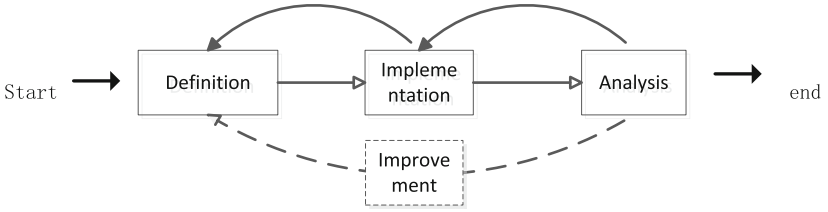


Fig. 2. The obligatory steps of data quality evaluation

3.3 The Data Quality Evaluation Dimensions

There are many methodologies and dimensions in data quality evaluation among specific field, especially in better informationalized circumstance. Abstraction from different data quality measurement practice, there are four categories characteristics: process, inherent, system dependent and user' satisfaction.

Figure 3: illustrates the relationships of process quality measures, data quality measures and quality in use measures.

Quality measures from inherent point of view: inherent characteristics may be applied to data itself, especially to data domain values and possible restrictions, for instance business rules governing the quality required for the characteristic, relationships of data values, metadata.

Quality measures from system dependent point of view: system dependent characteristics may be applied to quantify the influence on information technology applied in systems including software, hardware, network etc.

Relationship between types of quality measures from process and efficiency point of views: Data are expected to be correlated with other quality measures. High quality of the development and maintenance process is able to realize high quality of data, and data quality influences quality in use which represents the effect perceived by the final user [4].

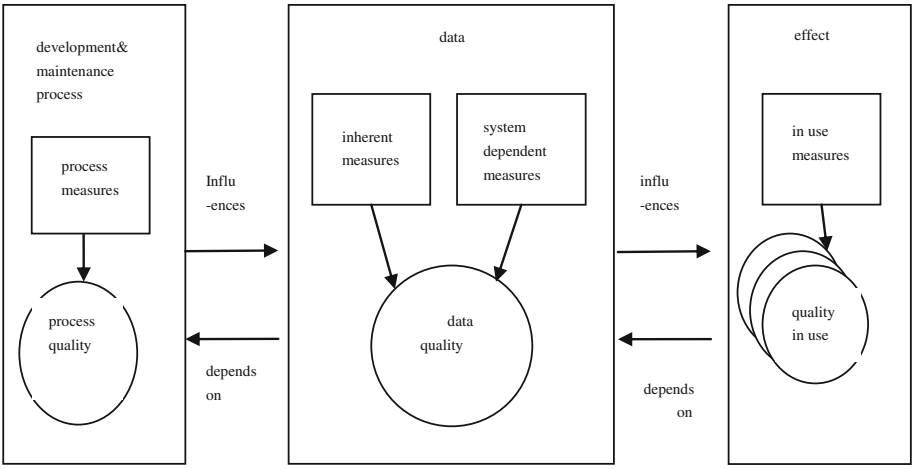


Fig. 3. Relationship between types of quality measures [4]

3.4 The Measures of Data Quality Characteristics [5]

According to the general principles of data quality evaluation, the data quality characteristics in this document are focus on 10 characteristics generally choosed to measurement in accordance with the inherent, system dependent points of view.

Completeness: Completeness measures provide the degree to which data associated with a target entity has values for all expected attributes in a specific context of use. Completeness includes attribute completeness, record completeness, data value completeness, data file completeness, metadata completeness etc.

Example: completeness of attribute for metadata [4] X
measurement function X = A/B
A = number of attributes with complete metadata within the data dictionary
B = number of attributes for which metadata are expected within the data dictionary

Consistency: Consistency measures provide the degree to which data attributes that are free from contradiction and coherent with other data. Consistency includes data format consistency, referential integrity, architecture consistency etc.

Accuracy: Accuracy measures provide the degree to which data has attribute that correctly represent the true value of the intended attributes. Accuracy includes data model accuracy, metadata accuracy, semantic data accuracy, syntactic data accuracy etc.

Example: metadata accuracy [4] X
measurement function X = A/B
A = number of metadata that provides the requested information
B = number of metadata defined within the specification of data

Compliance: Compliance measures provide the degree to which data has attributes that adhere to standards, regulations and conventions. Compliance includes regulatory compliance of value or format etc.
Currentness: Currentness measures provide the degree to which data has attributes that are of right stage. Currentness includes update frequency, timeliness of update etc.
Credibility: Credibility measures provide the degree to which data has attributes which considered as true and believable. Credibility includes values credibility, source credibility etc.

Example: source credibility [4] X
measurement function X = A/B
A = number of data values certified by a qualified organization.
B = number of data values for which source credibility can be defined.

Confidentiality: Confidentiality measures provide the degree to which data has attributes that only accessible by authorized users. Confidentiality includes encryption usage, privacy protection etc.
Traceability: Traceability measures provide the degree to which data has attributes that support an audit trail access of changes made to data. Traceability includes users access traceability etc.
Efficiency: Efficiency measures provide the degree to which data has attributes that can be processed the expected levels of performance. Efficiency includes usable efficiency etc.

Example: s usable efficiency [4] X
measurement function X = A/B
A = number of data values that intended users evaluate as "easily used"
B = number of data values evaluated by users.

4 Conclusion

This paper provides the fundamental solution of data quality evaluation that focus on the methodology, principle, procedure, model, characteristics and measures. The data quality assessment and measurement are becoming more and more popular in data management and quality management in order to improve the efficiency systematically. The concerning factors in this paper are abstracted from the practice in author's investigation project, and on the basis of achievement proposed in industrial standard s, national standards, international standards, that focus on geographi, environmental and software field etc. The information in this paper will provide useful guidance for data quality evaluation scenarios over data lifecycle, and provide the fundamental input to Big Data service etc.

References

1. DOD Guidelines on Data Quality Management (Summary), 31 July 2003
2. ISO/TS 8000:150 – A Framework for Data Quality Management (2011). http://www.dpadvantage.co.uk
3. McGilvray, D.: Executing Data Quality Project, Ten Steps to Quality Data and Trusted Information (2008)
4. ISO/IEC DIS 25024 – Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation - Measurement of Data Quality (2015)
5. Loshin, D.: The Practitioner's Guide to Data Quality Improvement (2011)

SecTube: SGX-Based Trusted Transmission System

Jian Chen[1], Bo Dai[1], Yanbo Wang[1], Yiyang Yao[1(✉)], and Bo Li[2]

[1] Information and Telecommunication Branch, State Grid Zhejiang Electric Power Company, Hangzhou, China
{chen_jian,dai_bo,wang_yanbo,yao_yiyang}@zj.sgcc.com.cn
[2] School of Computer Science and Engineering, Beihang University, Beijing, China
libo@act.buaa.edu.cn

Abstract. Trusted communication is a key component in trusted computing paradigm. Sensitive data usually has to be migrated between two applications or platforms in the environment of open network. In this case, not only file integrity monitor tools but also trusted transmission is needed. However, existing trusted transmission solutions run on the user's application platform or operating system. The lack of the isolation makes such security software easy to be subverted. In this paper, we present a novel approach called SecTube to protect the data safety in transmission. It utilizes Intel's new security technology SGX to give user application a safer execution environment. We also present the design and implementation of enclave socket in this paper. We realize the SecTube in Ubuntu 14.04 and several experiments are conducted. The experimental results show the effectiveness of SecTube and demonstrate that the average performance overhead of SecTube is only about 15%.

Keywords: Trusted computing · Encryption · Transmission · SGX

1 Introduction

For software developers, security is an inevitable problem to face, particularly when applications need to keep and process confidential data. Aiming at this problem trusted computing has been proposed. The core of trusted computing is that the computers will consistently behave in expected ways, and those behaviors will be enforced by computer hardware and software.

To protect sensitive data, file integrity monitor technique such as tripwire [1] is proposed in recent years. However, data usually has to be migrated between two applications or platforms in the environment of open network. In this case, not only file integrity monitor tools but also trusted transmission is needed. Trusted transmission [2, 3] is a technique that protects the data's confidentiality during the data transmission process by means of encrypting, accessing control and auditing. And it is always used for outgiving and accessing sensitive files safely. Nowadays, companies are increasingly concerned about the security of data. Hence the trusted transmission becomes research hotspot.

Existing trusted transmission solutions such as encryption run on the user's application platform or operating system, which brings a series of security issues [4, 5]. For

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 231–238, 2018.
https://doi.org/10.1007/978-3-319-73830-7_23

example, malwares e.g. viruses and hacking programs which threaten the confidentiality of the data may also threaten the security solutions. This is likely to result in a loss of sensitive data. Operation Aurora [6] attack is a typical example. What's worse, even if the above mentioned treat is solved, sophisticated attacker can still obtain the sensitive data by controlling the hardware. Memory Snooping [7] and ColdBoot Style Attack [8] are such attacks. In another word, in extreme case that even the hardware and operating system are untrusted, to guarantee the trusted transmission is very difficult.

Aiming at this problem, Intel has developed innovative new technology SGX to enable software developers to develop and deploy secure applications on untrusted platforms [9–13]. The technology enables applications to execute with confidentiality and integrity in the native OS environment. In general, this technology allows software developers control of the security of sensitive code and data by creating trusted domains with in applications to protect critical information during execution and at rest eventually.

In this paper, we present a novel approach called SecTube to protect the data safety in transmission. SecTube utilizes Intel's new security technology SGX. In SecTube, user applications are running in a protection mode: It leverages SGX to provide applications with cleartext of their data and the OS and underlying hardware a encrypt view. So the confidentiality will be promised even the underlying hardware or Operating system is compromised.

We describe the background and motivation first, and then present the design and implementation details of SecTube. Then we conduct several experiments to demonstrate our approach's effectiveness and efficiency. Finally, we close with the conclusion.

2 Design and Implementations

2.1 Design Goals

SecTube offers a last line of defense in the event of an OS compromise or a side channel attack. In this subsection, we discuss the design goals of SecTube including the treats it attempt to address.

Trusted transmission. Since it is a trusted transmission system, our key design goals are protecting the system from the following threats:

1. The contents of the file or the key used for encryption are stolen.
2. Platform and application identity spoofing, that is, SecTube should prevent attackers from posing as a legitimate application.
3. The tampering of application logs and access constraint.

Efficiency. SecTube should not perturb the proper functioning of the transmission and the overhead should be as little as possible.

Ease of adoption. Our solution should be compatible to most mainstream platforms.

2.2 Architecture

This subsection will introduce the overall design of the SecTube. As is shown in Fig. 1, the system is a typical client-server architecture.

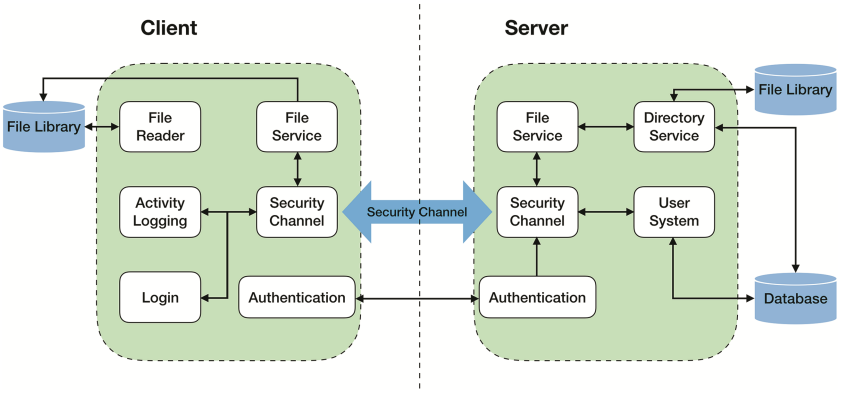


Fig. 1. The architecture of Sectube

The trusted part of client, which is hosted on SGX in protected enclave, performs file operations that requires confidentiality. The following will describe how is the architecture and the entire system running to achieve security in a variety of file management scenarios.

The authentication module in client verifies the legitimacy of the platform and the user of the corresponding server. Using SGX's authentication function, this module generates a verifiable report about the client entity, that is, the identity information bound to the platform on which the client resides, which is bound by the CPU. The server-side inspects the report to determine that the machine communicates with itself at the time supports the SGX function and that the client's identity is legal. Client and server side implement a one-off supply protocol to make the confidential data is sealed to the platform the application is on, using SGX's sealing function. And this encrypted confidential data can only be decrypted and operated by the application. Therefore, we can seal the data for the subsequent establishment of a session with the server conveniently, because this way we do not have to provide identity information of the platform after each time the clients start.

In the architecture of the system, the access rights and encryption keys of the files are saved to the server's database. The database administrator can modify the corresponding access rights and group the users for the client to manage the permissions. After the client has been authenticated, a session is established between the server, and the server confirms the security of the client and its platform. Moreover, if the file is simply transferred from the server to the client, it greatly reduces the security of the system, but also makes the client's encrypted storage lost its meaning. To do this, it's necessary to establish a separate secure channel between the server and the client, with each channel corresponding to the unique client. After the protected file is encrypted in

the enclave, the client sends a specific file to the client according to the client's file request, which is then distributed to the user who has obtained the authentication and authorization to view or perform other operations.

Once the authenticated user receives the encrypted file, he can use the security file reading component running in the enclave of the client platform for file viewing. At the same time, the client's permission check component needs to check whether the user has permission to view or modify the file. Once the file permissions check fails, the file decryption component will not work.

Similarly running in the enclave, the security activity record component will record the user's activity as well as the operation of the file and so on. This allows the system to have the characteristics of the audit and approval of the user's operation, thus enhancing the security while making the system itself more robust.

For the untrusted parts of the application, such as the use of the system itself, which mostly is file IO operations, the system uses well-defined interfaces between the SGX host and enclave a good interface to read. The interfaces are used to ensure that the data is not from the safe part, that is, what we call enclave, leak to the untrustworthy area. In addition, SGX hardware and software protection can also ensure that the application data in the enclave and the confidentiality and integrity of the code is difficult to be destroyed.

As for the server part, since the above has basically described the entire operation process, so here introduce only a certain part of the components. The server consists of several modules: authentication and session management module, file transfer key generation module, and a database that stores users and related data. All communication between the server and the client is encrypted, and in a variety of scenarios the system provides guarantees such as end-to-end integrity and playback protection and so on.

2.3 Implementation of Enclave Socket

To send files from the server to the client, the data must be transferred from the secure channel during the entire process.

Although enclave can legally access the shared memory in the host outside the enclave, there is still a problem in this way, because a malicious host or operating system may potentially modify non-enclave memory. Therefore, in order to avoid this situation, SecTube uses a more stringent form of communication protocol, that is, to use the shared code and data area, namely Trampoline and Stub. This area defines a strict interface to interact with the enclave, making the associated security attributes easy to control.

The communication is unidirectional and is all dominated by the enclave. As shown in Fig. 2, enclave requests a socket instance for the network. First, it set the corresponding parameters in the stub (for example, assign fcode to FSCOKET), and then call the predefined handler, Trampoline, and exit the enclave mode (call EEXIT). When the host program or the operating system has processed the enclave request, the result or return value will be stored in the Stub, and then SecTube calls the ERESUME command to restore the operation of the enclave. After restoring the control of the program to the location before the enclave, enclave can read the value in the stub and get the socket instance through the in_arg0 in the stub.

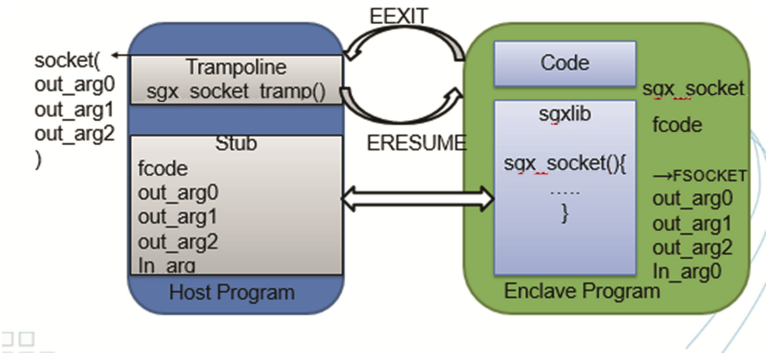


Fig. 2. The enclave socket

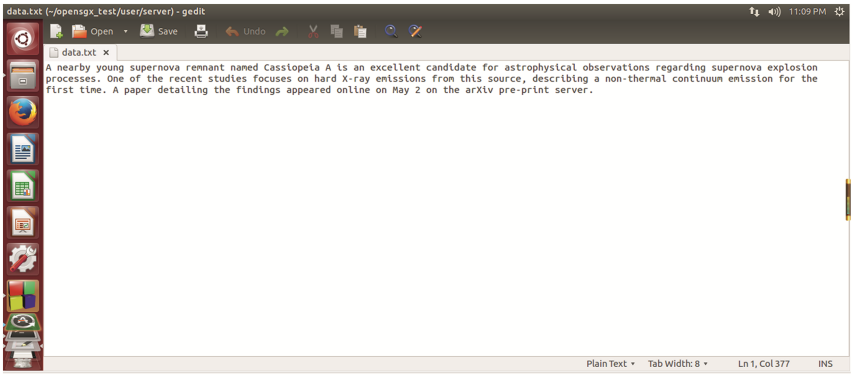


Fig. 3. The content of the text file

3 Experiment

To verify the effectiveness and efficiency of SecTube, we conduct several comprehensive experiments. Before describing the experimental results, we first give the experimental setup.

Experimental Setup. The experiments were conducted on two physical servers of Intel Core i7 CPU with 16 GB memory. One is the client. The other is a remote storage server. The two machines are connected with Giga bytes network. The OS of the two machines is Ubuntu 14.04 with Linux kernel 3.4. To evaluate the effectiveness and performance overhead introduced by SecTube, we design two experiments. One is to evaluate the effectiveness of SecTube, the other is to evaluate the transmission performance overhead of SecTube. Remote file copy is used to test the transmission performance. By comparing the transmission efficiency before and during the backup process, we could evaluate the performance of SecTube.

Effectiveness. To first evaluate the effectiveness of SecTube, we simulate malicious that tries to steal the transferred file in operating system and transmission channel. In the experiment, we first edit a text file and then send the file from server to the client. Then we use side channel attack and tries to obtain the private key by illegally scan the host's memory. But all we can get is the encrypted text, as is shown is Fig. 4 (the original content of the text is in Fig. 3). The result shows the effectiveness of SecTube.

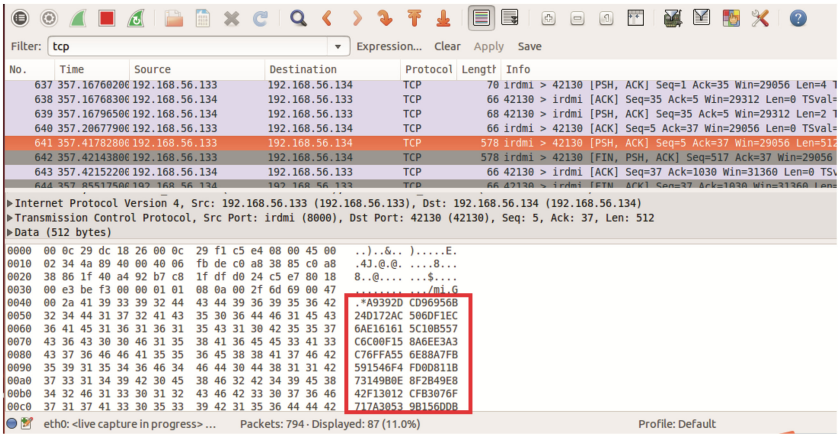


Fig. 4. The text is encrypted

Performance. We use online file transmission to test the performance of SecTube. We test five different file sizes: 64 KB, 128 KB, 256 KB, 512 KB and 1024 KB. As shown in Fig. 5, the average performance overhead of SecTube is about 15%. Performance is

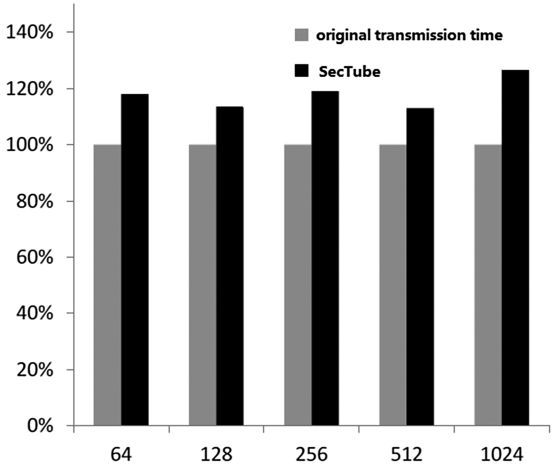


Fig. 5. The result of performance experiment

worse when file size is 1024 KB, and performance overhead is small when file sizes are 128 KB and 512 KB. In general, the performance overhead of SecTube during backup is acceptable.

4 Conclusion

This paper proposed a novel approach called SecTube to protect the data safety in transmission. It leverages SGX to provide applications with cleartext of their data and the OS and underlying hardware a encrypt view. And we also present the design and implementation of enclave socket in this paper. We realize the SecTube in Ubuntu 14.04 and several experiments are conducted. The experiment results show the effectiveness of SecTube and demonstrate that SecTube only incurs acceptable overhead.

Acknowledgement. The authors gratefully acknowledge the anonymous reviewers for their helpful suggestions.

References

1. Kim, G.H., Spafford, E.H.: Experiences with Tripwire: using integrity checkers for intrusion detection. In: System Administration, Networking and Security Conference (1994)
2. Chang, X., et al.: ZRTP-based trusted transmission of VoIP traffic and formal verification. In: IEEE International Conference on Multimedia Information Networking and Security, pp. 560–563 (2012)
3. Minmin, L., Liu, J.: A trusted transmission protocol based on trusted computing technology. In: International Conference on Computational Problem-Solving IEEE, pp. 473–476 (2012)
4. Chen, X., Garfinkel, T., Lewis, E.C., Subrahmanyam, P., Waldspurger, C.A., Boneh, D., Dwoskin, J., Ports, D.R.K.: Overshadow: a virtualization-based approach to retrofitting protection in commodity operating systems. SIGPLAN Not. **43**(3), 2–13 (2008)
5. Hofmann, O.S., Kim, S., Dunn, A.M., Lee, M.Z., Witchel, E.: InkTag: secure applications on an untrusted operating system. SIGPLAN Not. **48**(4), 265–278 (2013)
6. McAfee Labs and McAfee Foundstone Professional Services: Protecting Your Critical Assets: Lessons Learned from "Operation Aurora". http://www.mcafee.com/us/resources/white-papers/wpprotecting-critical-assets.pdf. Accessed 17 June 2013
7. Clarke, D., et al.: Checking the integrity of memory in a snooping-based symmetric multiprocessor (SMP) system. MIT CSAIL CSG-TR-470 **42**(1–3), 335–346 (2004)
8. Haldermen, J.A., Schoen, S.D., Heninger, N., et al.: Lest We Remember: Cold Boot Attacks on Encryption Keys. https://citp.princeton.edu/research/memory/. Accessed 17 June 2013
9. Winter, J.: Trusted computing building blocks for embedded linux-based ARM trustzone platforms. In: Computer and Communications Security, pp. 21–30 (2008)
10. Anati, I., Gueron, S., Johnson, S., et al.: Innovative technology for CPU based attestation and sealing. In: Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy, vol. 13 (2013)
11. McKeen, F., Alexandrovich, I., Berenzon, A., et al.: Innovative instructions and software model for isolated execution. In: HASP@ ISCA, p. 10 (2013)

12. Hoekstra, M., Lal, R., Pappachan, P., et al.: Using innovative instructions to create trustworthy software solutions. In: HASP@ ISCA, p. 11 (2013)
13. Schuster, F., Costa, M., Fournet, C., et al.: VC3: trustworthy data analytics in the cloud using SGX. In: 2015 IEEE Symposium on Security and Privacy (SP), p. 38. IEEE (2015)

The Research How to Judge Social Vehicles Driving into ART

Weihu Wang[1], Zenggang Xiong[1(✉)], Yanshen Liu[2], Fang Xu[1], and Conghuan Ye[1]

[1] School of Computer and Information Science, Hubei Engineering University, Xiaogan 432000, China
wangweihu80@163.com, xzg@hbeu.edu.cn

[2] National Engineering Research Center for E-Learning, Central China Normal University, Wuhan 430079, China

Abstract. This paper presents a new algorithm for avoiding ambulances being stuck for a long time in first aid, which uses advanced the Internet of Things (IoT) and location technology. Firstly, referring to BRT at home and aboard, ART is presented in the paper, which will be a kind of public service program in many important cities in China. Meanwhile; it is a very detailed research about the design idea of ART. Secondly, it also presents an algorithm how to judge social vehicles driving into ART, in order to avoid ambulances blocked, and increase times to rescue patients. Finally, the feasibility of the proposed algorithm is illustrated by some simulation experiments.

Keywords: ART (Ambulance Rapid Transit) · Social vehicles · Location Logistic Regression

1 Introduction

In China, the ageing of populations is now universal trends. Thus, the demand of the ambulance in modern society is increasing day by day [1]. In a modern city, there is a universal phenomenon that an ambulance in rescue processing, often has meet traffic jam or has been trapped etc., so that the ambulance is unable to arrive in time to rescue emergence, and treatment of the best time is delayed. Even there is a serious fact that the ambulance is the hope of their lives for many critical patients [2]. However, the emergency situation that the ambulance along the transit of life often has encountered some kinds of difficulties has not been solved through traffic regulations at present [3].

So In that way, how to solve the problems that the ambulance is also blocked, the life channel is unblocked in the city, has become an important research topic. In this papers, referring to BRT (Bus Rapid Transit) in some cities, such as Beijing, Changzhou, Tokyo, and Seoul [4], I suggested that ART (Ambulance Rapid Transit) is established by government as soon as possible, in order to locate some social vehicles. Meanwhile, I propose also a new algorithm, which is how to judge the social vehicles driving into the ART channel, so as to avoid the traffic jam and improve the time for rescuing patients.

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 239–248, 2018.
https://doi.org/10.1007/978-3-319-73830-7_24

The rest of the paper is organized as follows. Section 2 describes the design idea of ART; Sect. 3 presents an algorithm how to judge the social vehicles driving into the ART.

2 Ambulance Rapid Transit (ART)

2.1 The Design Idea of ART

ART (Ambulance Rapid Transit), is a fast channel. It has been specially designed for these emergency vehicles in the existing road of the city, for instance, ambulances, fire trucks or police cars. But it also can be reused with existing BRT channel. For example, ART programming model is shown in Fig. 1.

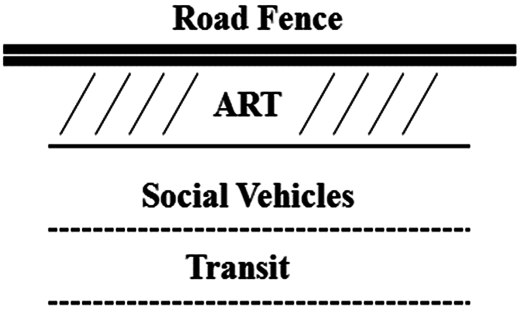


Fig. 1. ART planning model (one side of the road)

ART fully embodies development idea- the people-oriented, constructing a harmonious society. Through the closed special road, it can achieve the fast, punctual, comfort and security of services. Meanwhile it can also avoid the phenomenon of traffic jams in ambulances or other emergency vehicles, in order to improve the efficiency of public services.

2.2 The Feature of ART

From the design idea, ART has three features, as following:

(1) Keeping life channel open, Saving waiting time
ART is a customized channel, also is a life channel. There is the most important guarantee to keep it open. So the ambulance arrives at the scene in time, saving waiting time of the patients.
(2) Improving survival hope
The ambulance timely arrives, so emergency medical persons can undertake an emergency mission at the scene, and improve the hope of the patient life.
(3) Making the traffic situation better

It can improve original phenomenon in city that the social vehicles and emergency vehicles confusedly occupied lanes, to bring traffic jams. It can also balance the urban traffic development, make the quality of public life environment better.

3 How to Judge Social Vehicles Approaching to the ART

In Sect. 2, it has described the design idea about ART, but how to monitor social vehicles entering the ART, avoiding causing traffic jam. Therefore, there is necessary to reaches an efficient algorithm how to judge in this papers. In fact, it is a kind of comparison algorithm between the actual driving path of social vehicles and the preset ART. In the process of matching route, according to GPS terminal equipment in the social vehicle, it will compare obtaining the position with planned ART.

3.1 Idea About Algorithm

The basic principle how to judge social vehicles entering ART algorithm, is as following: Through comparing the coordination of social vehicle and some preliminary selected each key points of ART, and computing the distance between two points [5], it is necessary to judge the society vehicles whether entering the ART, as shown in Fig. 2.

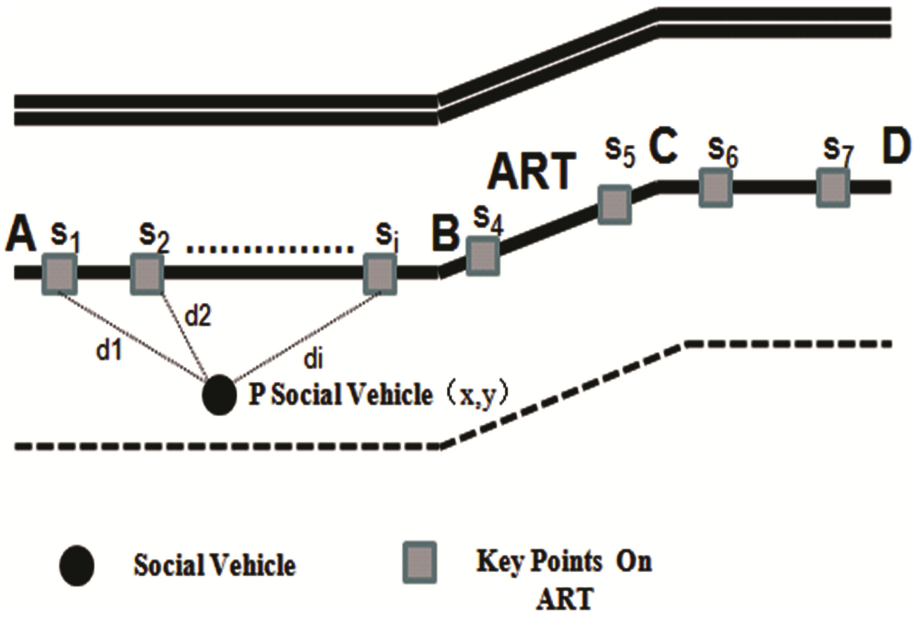


Fig. 2. The basic idea of algorithm

The algorithm mainly includes the following five steps:

(1) According to the characters of the ART, producing some approximate straight line segments, such as line segments AB, BC and CD [6, 7].
(2) In the each line segment of the ART, setting random a number of key points. Such as on the line AB, seeing expression 3.1 below.

$$S = \{S_1, S_2, S_3, \ldots., S_i\} (i = 1, 2, 3 \ldots \ldots n) \tag{3.1}$$

(3) According to the formula (3.2) and (3.3), computing the Euclidean distance d_i (i = 1, 2, 3… N) between position of the social vehicle and each point in set S in the line segment [8–10], finally finding the minimum distance.

$$d_i = PS_i = \sqrt{\left(x_p - x_{si}\right)^2 \div \left(y_p - y_{si}\right)^2} \quad (i = 1, 2, 3 \ldots \ldots n) \tag{3.2}$$

$$d_{\min} = |PS|_{\min} = \min(|PS_i|) \quad (i = 1, 2, 3 \ldots \ldots n) \tag{3.3}$$

(4) Determining if social vehicle is running into the ART. R represents the shortest distance threshold between social vehicles and ART.

$$\begin{cases} d_{\min} = PS_{\min} \le R \text{ Driving into the ART} \\ d_{\min} = PS_{\min} > R \text{ on the contrary, Far from the ART} \end{cases}$$

(5) Here, a straight line segment has completed, then repeat from step (2) to (4), until the whole line alignments on the ART has completed.

Social vehicle approaches into the ART judgment algorithm, as shown in Fig. 3.

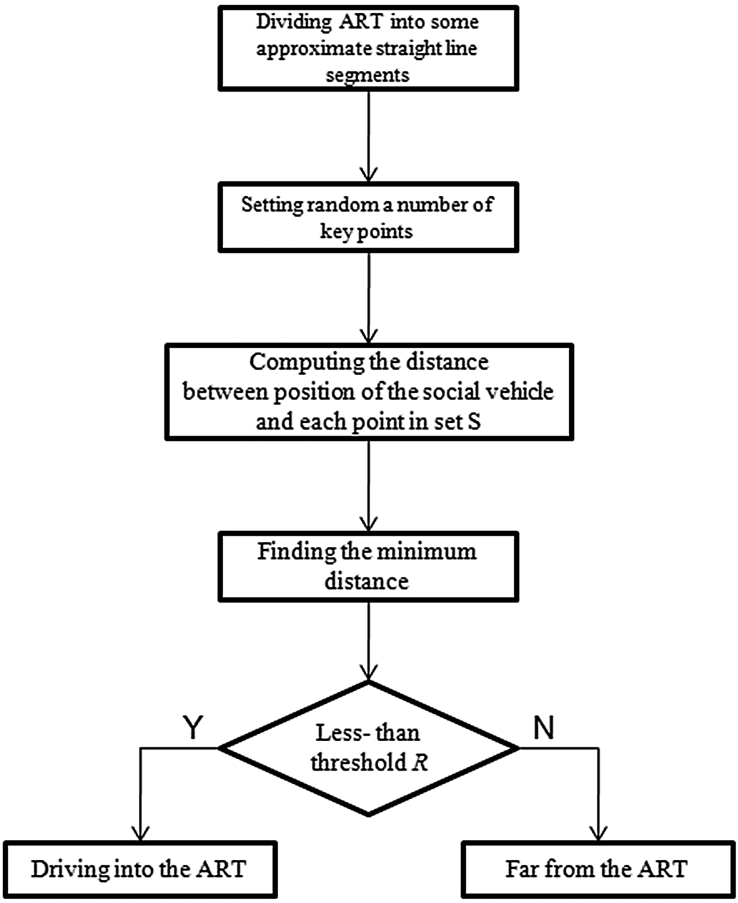


Fig. 3. The Flow Chart of Algorithm

3.2 The Logistic Regression of the Threshold R

It is very necessary to compare between distance d and the Threshold in Step (3) and (4) of algorithm, so to judge if social vehicles driving into ART. In fact, this is a typical two classification problem. because distance d is a continuous function, which maps from f ∈ R to f ∈ {0, 1}, it can be classified by Logistic Regression.

X stands for n-dimensional feature vectors (Suppose only discuss distance d and shape of the social vehicles), n-dimensional feature vectors W is used to represent the corresponding weights, and b stands for intercept [11, 12]. It presents a linear relation equation for the algorithm, such expressions (3.4).

$$\begin{aligned} f(w, x, b) &= w_1 x_1 + w_2 x_2 + b \\ &= W^T X + b \end{aligned} \tag{3.4}$$

When substituting (1) into Logistic Function, it is easy to obtain logistic Regression model, such expressions (3.5).

$$\begin{aligned} h(w,b) &= g(f(w,x,b)) \\ &= \frac{1}{(1+e^{-z})} \\ &= \frac{1}{(1+e^{-(f(w,x,b))})} \end{aligned} \tag{3.5}$$

According to the fact that the weight of distance d is larger than Shape in two dimensional feature vectors, so it is mainly to research by dimension reduction method here. It is quick to gain conditions driving into ART according to the character of Logistic Function, as shown in Fig. 4.

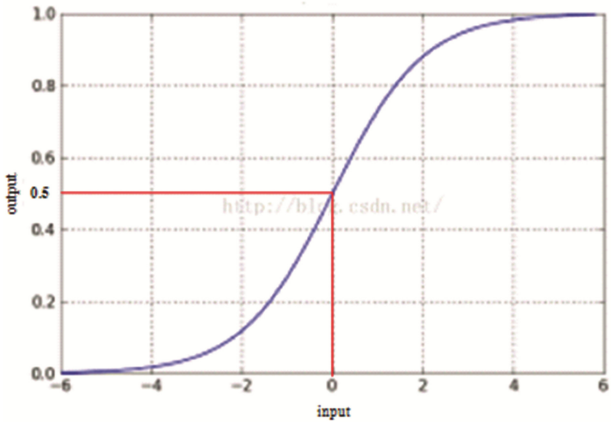


Fig. 4. Logistic Function

If z = 0, $f(w,x,b) = 0$, g = 0.5, x1 = d = 0, obtained position of the social vehicle (x_p, y_p) now, is either coincide with the key points in ART or inside ART.

If Z < 0, $f(w,x,b) < 0$, g < 0.5, obtained position of the social vehicle (x_p, y_p) now is inside ART.

If Z > 0, $f(w,x,b) > 0$, g > 0.5, obtained position of the social vehicle (x_p, y_p) now is outside ART.

In conclusion, When Z takes different values, the classification is as follows.

$$\begin{cases} \mathrm{Z} = 0 & \mathrm{class} = 1 \\ \mathrm{Z} < 0 & \mathrm{class} = 1 \\ \mathrm{Z} > 0 & \mathrm{class} = 0 \end{cases}$$

Class∈{0, 1}, which stands for quantitative criteria of classification. 1 and 0 represent driving or not driving respectively.

4 Simulation Experiments

4.1 Optimizing the Threshold R

After removing the feature vector shape. it gives the shortest distance threshold R by Linear Classifiers Model. The collecting data is organized in accordance with distance R increasing by 0.1 m, as shown in Table 1.

Table 1. Collecting data

Sample number	Distance R(m)	Shape	Class	Sample number	Distance R(m)	Shape	Class
1	0	A1	1	26	0.5	A1	0
2	0	A2	1	27	0.5	A2	0
3	0	B1	1	28	0.5	B1	0
4	0	B2	1	29	0.5	B2	0
5	0	C1	1	30	0.5	C1	0
6	0.1	A1	1	31	0.6	A1	0
7	0.1	A2	1	32	0.6	A2	0
8	0.1	B1	1	33	0.6	B1	0
9	0.1	B2	1	34	0.6	B2	0
10	0.1	C1	1	35	0.6	C1	0
11	0.2	A1	1	36	0.7	A1	0
12	0.2	A2	1	37	0.7	A2	0
13	0.2	B1	1	38	0.7	B1	0
14	0.2	B2	1	39	0.7	B2	0
15	0.2	C1	1	40	0.7	C1	0
16	0.3	A1	1	41	0.8	A1	0
17	0.3	A2	1	42	0.8	A2	0
18	0.3	B1	1	43	0.8	B1	0
19	0.3	B2	1	44	0.8	B2	0
20	0.3	C1	1	45	0.8	C1	0
21	0.4	A1	1	46	0.9	A1	0
22	0.4	A2	1	47	0.9	A2	0
23	0.4	B1	1	48	0.9	B1	0
24	0.4	B2	1	49	0.9	B2	0
25	0.4	C1	1	50	0.9	C1	0

Note: Class 1 represents Driving into the ART; Class 0 represents Far from the ART

The core code about python is as follows.

//25% and 75% data in the above table is respectively training data and test data as learning model by Random algorithm.

```
  >>train_X,test_X, train_y, test_y = train_test_split(train, target,
 test_size = 0.25, random_state = 0)

  //Initializing Logistic Regression.

  >>lr=logisticRegression()

  //Calling fit function in Logistic Regression, which is used to
train model parameters.

>>Lr.fit(train_X,train_y)

// Predicting test_X by using trained model Lr, the result is
  stored in the variable lr_y_predict.

>>lr_y_predict=lr.predict(test_X)
```

Accuracy of determining driving into ART based on Linear Classifiers Model is 0.989. Finally, based on the characteristic of the learning model and actual ART situation, here it is reasonable that the threshold of R is selected as 0–0.5 in this algorithm.

4.2 The Simulation of the Algorithm

In order to verify the algorithm, we have tested ten groups of data in the experiment, and statistics is shown in Table 2. The comparison chart of anomalous line segments is shown in Fig. 4.

Table 2. Calculating the number of line segments in the algorithm

The number of key points on the ART	Experiment results	
	The number of calculated line segments	The number of anomalous line segments
10	10	0
50	50	0
80	80	0
100	100	1
150	150	4
200	200	6
250	250	11
300	300	16
400	400	14
500	500	17

From Table 1, in the algorithm, it needs calculate the distances between the social vehicles and all social segments on the ART. Because the preset points on the ATR is relatively little, monitoring efficiency is quite good. When there have a large number of the preset points on it, it is inefficient to monitor social vehicles because of large amount of calculation.

In Fig. 5, we can find that when line segment number is between 300 and 400, it is very obvious to calculate abnormal, it is possible that the line is too long. Therefore, the complexity of this algorithm is relatively high. It need to be improved in the future.

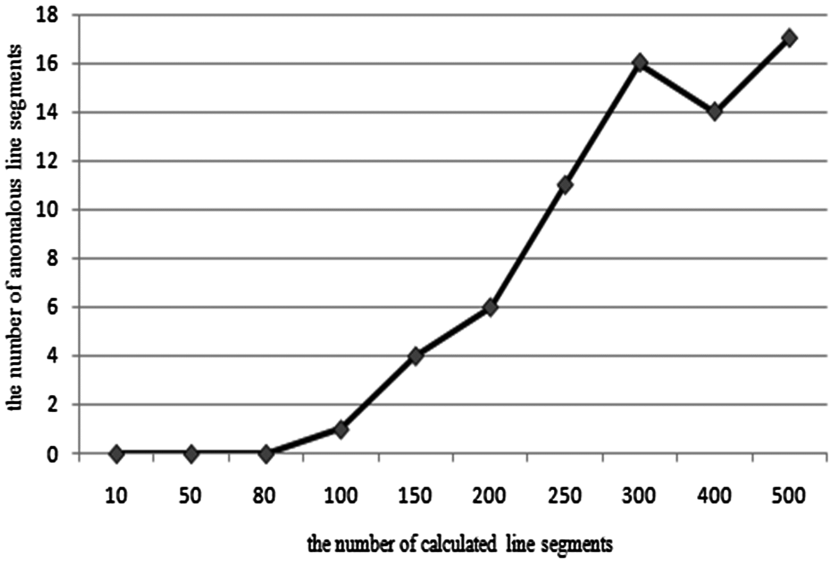


Fig. 5. The comparison of anomalous line segments

5 Conclusion

In order to solve the blocked ambulances, which are unable to arrive in the scene in time in the city, the paper puts forward to build a ART, referring to BRT in the domestic and foreign cities, combining with several cities in China. Firstly, I describe the design ideas about ART in detail, secondly, I propose a algorithm how to judge social vehicles enter the ART, Through comparing the coordination of social vehicle and some preliminary selected each key points of ART, and computing the distance between two points, it is necessary to judge the society vehicles whether entering the ART. It is very important how to optimize the threshold R by logistic Regression. Finally, it is effective to avoid the traffic jam, and improve the time for rescuing patients by Simulation Experiments. There is a further study how to judge driving into ART by multiple factors in the future.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (No. 61370092), Hubei Provincial Department of Education Outstanding Youth Scientific Innovation Team Support Foundation (T201410).

References

1. Hu, Q.: Avoiding ambulances: releasing for life. Ningbo Economy (Finance View), (06), 48 (2013)
2. Yao, B.: Ambulances blocked, reflecting public service. Law Life (01), 22–23 (2013)
3. Wang, W., Liu, Y.: Research of intelligence ambulance based on analysis of ambulance industry in China. Softw. Guide (01), 14–15 (2015)
4. Guo, Z., Wu, H., Zhu, H.: Choice of bus Rapid Transit mode and network planning: a case study of Wuhan. Transp. Res. (01), 33–37 (2015)
5. Gao, H., Jia, K., He, J.: On algorithms of path matching and trace playback and their implementation. Comput. Appl. Softw. (04), 26–28 (2010)
6. Cao, Z.: Research and Implementation of Geo-Fence for Dangerous Goods Transportation. Zhejiang University (2014)
7. Zhang, Y.: Research on GPRS-Based real time vehicle positioning and surveillance system. Beijing University of Posts and Telecommunications (2009)
8. Che, L., Yan, Y., Jiang, L., Zhou, J., Zhu, C.: Traffic organization design and research of BRT system in Wuhan City. Urban Roads Bridges Flood Control (03), 17–21 (2014)
9. Yang, H., Wang, D., Zhang, J.: Research on the shortest path of congested traffic based on ant colony algorithm. Comput. Simul. (03), 186–191 (2015)
10. Zhang, Z., Lv, M., Sun, G., Chen, G.: An efficient lower-bounding approach to point-to-point shortest path problem. J. Univ. Sci. Technol. China **10**, 874–880 (2014)
11. Gai, K., Qiu, L., Chen, M., Zhao, H., Qiu, M.: SA-EAST: security-aware efficient data transmission for ITS in mobile heterogeneous cloud computing. ACM Trans. Embed. Comput. Syst. **16**(2), 1–22 (2017)
12. Gai, K., Qiu, M., Sun, X., Zhao, H.: Security and privacy issues: a survey on FinTech. In: International Conference on Smart Computing & Communication, pp. 236–247 (2016)

PSVA: A Content-Based Publish/Subscribe Video Advertising Framework

Feiyang Wang[1], Dongyu Zhang[2], Yuming Lu[3], and Kai Lei[1(✉)]

[1] Institute of Big Data Technologies, Shenzhen Key Lab for Cloud Computing Technology and Applications, School of Electronic and Computer Engineering (SECE), Peking University, Shenzhen 518055, People's Republic of China
Wangfy16@pku.edu.cn, leik@pkusz.edu.cn
[2] Harbin Institute of Technology, Harbin, China
dongyu.z@hit.edu.cn
[3] Shenzhen Key Lab for Visual Media Processing and Streaming Media, Shenzhen Institute of Information Technology, Shenzhen 518172, People's Republic of China
luyuming@sziit.edu.cn

Abstract. Online video technology rapidly developed in the past few decades, amounts all applications, video advertising is a uprising approach to improve commercial advertisement revenue due to its diverse presentations, interactivity GUI, tailored media publishing and measurable subscription features etc. However to accurately match commercial adverts from the publisher's media source, with consumer desire contents, as well as effective real-time video streaming transmission in a media processing system remains a problem. This paper proposes a content-based Publish/Subscribe Video Advertising (PSVA) framework. In this framework, we construct a high-performance scalable communication subsystem to carry out real-time video streaming transmission, and the handle event matching by a stream computation subsystem. The experimental results show that the PSVA framework is effective.

Keywords: Online video · Publish/subscribe system · Video advertising Event matching

1 Introduction

Online video has occupied more than 85% of internet traffic and rising, and Publish/Subscribe (P/B) system is an important commercial application that allows advertising video sources to "publish" advertisements (Ads) onto the internet, so that consumers (Users) can chose contents of interest, known as "subscribe". Previous video-oriented sites such as YouTube, MSN Video have tried to provide effective video advertising services. However, most of the Ads are matched based on text information, and the positions of Ads insertion are usually fixed, such as at the beginning or the end of a video. This form will not only make the number of embedded ads limited, but also more intrusive ads within video content [1]. We believe ads should be contextually relevant to online video content in terms of multimodal relevance (i.e., textual, visual, and aural relevance). For example,

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 249–258, 2018.
https://doi.org/10.1007/978-3-319-73830-7_25

when viewing an online football video, users may prefer a relevant ad with the similar football types to the video, which cannot be measured just by textual information. This capability will enable delivering the ads with much higher relevance.

Motivated by the above observations, We try to use the P/B system to build a more comprehensive system for online video advertising. The proposed system, called PSVA, supports more effective video advertising in terms of contextual relevance and less intrusiveness by Big data related technology techniques (e.g., streaming computing, message queuing, column database). In PSVA, given an online source video defined by video content, relevant video ads will be matched to the original video via the P/S system and seamlessly inserted into the video at appropriate positions. we construct a high-performance scalable communication subsystem to carry out real-time video streaming transmission, and the handle event matching by a stream computation subsystem. The main contributions of this paper are as follows:

- We propose a framework PSVA to apply the P/B system to video ad scenes.
- We construct a high-performance scalable communication subsystem.
- We designed a series of experiments to verify the performance of the PSVA framework.

The rest of this paper is arranged as follows. Section 2 reviews related work. Section 3 describes the P/B system. Section 4 introduces the design and implementation of PSVA. Section 5 tests PSVA performance and Sect. 6 concludes this paper.

2 Related Work

Many factors need to be considered in the design of the video ad systems, such as video streaming, advertising content, user experience. Especially, we needs to take into account the large-scale high-complexity features of video streams and the distribution control of media streaming. In the prior art, streaming computing is a way to handle real-time data streams. For the distribution of media streaming control, the use of non-ip content center network distribution of streaming media NDN-Hippo system framework is an effective reference form [2].

Employing mobile cloud computing (MCC) to enable mobile users to acquire benefits of cloud computing by an environmental friendly method is an efficient strategy for meeting current industrial demands. However, the restrictions of wireless bandwidth and device capacity have brought various obstacles, [3] propose a dynamic energy-aware cloudlet-based mobile cloud computing model (DECM) focusing on solving the additional energy consumptions during the wireless communications by leveraging dynamic cloudlets (DCL)-based model. [4] propose a novel approach for reducing the computation energy costs for heterogeneous MES in cloud systems. the proposed model is called Energy-Aware Heterogeneous Cloud Management (EA-HCM) model. It aims to reduce the total energy cost of the mobile heterogeneous embedded systems by a novel task assignment to heterogeneous cores and mobile clouds. the approach is effective to save energy when deploying heterogeneous embedded systems in mobile cloud systems.

3 Publish/Subscribe System

This section describes the content of the Publish/subscribe, this paper mainly uses content-based Publish/subscribe model.

3.1 Publish/Subscribe System

The Publish/subscribe technology was first presented by Frank Schmuck, which was presented at the 1987 Symposium on Operating System Principles that is sponsored by the American Computer Society (ACM) [5].

P/B system consists of three types of Entity: publisher, subscriber, and the middleware. The publisher passes the generated event to the middleware, and the subscriber submits the subscription condition to the middleware. Middleware consists of multiple proxy nodes, which are connected to each other to form a coverage network to complete the routing, matching and delivery tasks. P/B system is a message paradigm that the publishers do not need to know which subscribers subscribe to the message, and the subscribers do not need to know the publisher of the message. The decoupling between the publisher and the subscriber makes the P/B system more scalability. In a P/B system, the publisher publishes a message to a middleware that consists of a proxy node, at the same time the subscriber submits the subscription to the middleware. The next step, Middleware passes the message from the publisher to the subscriber through storage-match-forward mechanism. The main advantages of the P/B system include loosely coupled and scalability shown in Fig. 1.

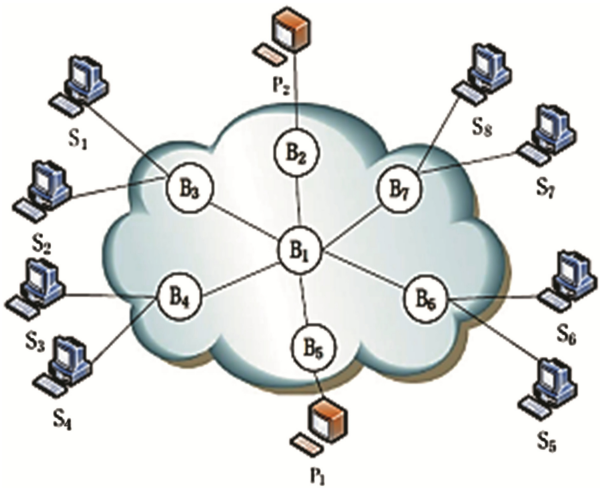


Fig. 1. A distributed publish/subscribe system

3.2 Content-Based Publish/Subscribe System

In a P/B system, Subscribers are usually only interested in a particular event or event type. The P/B system is divided into: subject-based, content-based, type-based. This paper focus on content-based publish/subscribe systems, subject-based and type-based publish/subscribe systems can refer to [6, 7].

Content-based publish/subscribe system was proposed by Rosenblum and Wolf In 1997 to improve the expressiveness of subscriptions [8]. In this model, the event is no longer classified according to the pre-defined theme, but by the properties of the event itself to decide. The attributes can be the metadata user to describe the event.

Content-based publish/subscribe systems greatly improve the ability to express subscriptions, that subscribers do not need to know a set of pre-defined topics. The matching task burden of the content-based publish/subscribe system is aggravated compared to the topic-based, because it is necessary to check the contents of the attributes in the event to determine whether the event meets the subscription criteria. Therefore, it is necessary to improve the efficiency of matching in order to meet the needs of large-scale distributed systems.

Typical content-based publish/subscribe systems are Grphon [9], SIENA [10], and JEDI [11]. When the user subscribes, the PSVA system, presented by this paper, first analyzes whether the message matches the subscription content and then decides whether to send the message to the user. The system can meet the needs of the user message in the actual network scenario.

3.3 Publish/Subscribe System Architecture

The early P/S system mainly uses a centralized architecture, which is responsible for messaging by a proxy server. the publisher sends the event to the proxy server, the subscribes sends the subscribes to the proxy server, server is responsible for event matching and event forwarding. As the size of the P/S system continues to expand, centralized proxy server has become a bottleneck in system performance. Large-scale P/S system usually uses a distributed architecture to improve the scalability of the system, a number of proxy nodes distributed in different locations, connected to form a different topology. each agent node responsible for subscribing to routes and events matching and passing.

4 Design and Implementation

4.1 Design

Design a complete, real-time, scalable video ads system, and under the background of data growth, the relative stability of the system is maintained in order to deal with the mass data of the user, the core algorithm of the PSVA is PADEM [12] and the FlatMap-Filter calculation mode. In addition, because the user's interest and preference will change with time, the historical log information will increase, so the need to re calculate and update the data to maintain the freshness of the user's historical data. Finally, the

results need to be returned in real time and provide high quality recommendation; the logic structure of the system is divided into 4 levels: as shown in Fig. 2, the storage layer, the computing layer, the Router layer and the access layer.

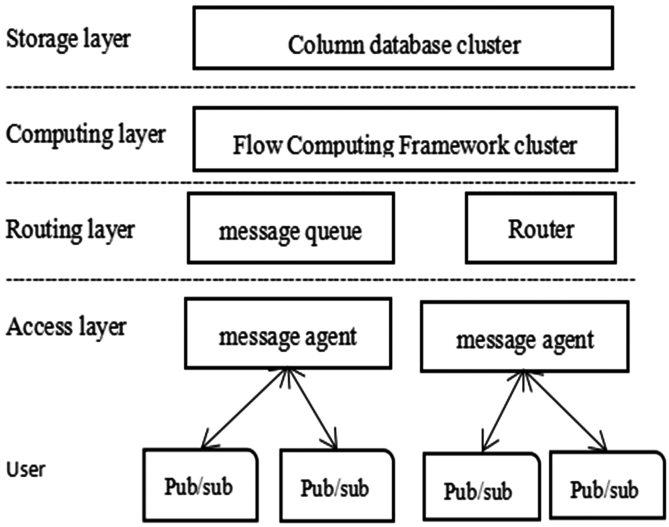


Fig. 2. System hierarchical architecture

The system consists of message distribution, message subscription, message matching, message transmission, abnormal monitoring (not discussed) modules, Respectively. The modules are implemented by the system access layer, routing layer, computing layer and storage layer respectively. the cooperation between the modules is shown in Fig. 3.

The Storage Layer. The storage layer consists of a database client and a column database. The database client is responsible for reading messages from the message queue and writing to the column database. The column database is a database of data storage in a column-dependent storage architecture, which is mainly suitable for batch data processing and real-time queries.

The Computing Layer. The computing layer uses the Storm compute framework cluster to match the event and subscription attributes. When the message distribution module (i.e. the message agent) receives the message, the message is forwarded to the computing layer through the message queue for subscription matching. After the matching succeeds, the system passes the subscriber and its user information to the Router module as parameters. And then sent to the matching subscribers. The message matching module consists of two steps: filtering and matching.

The Router Layer. The routing layer is mainly composed of message queues and Router modules. The message queue is used to store the asynchronous communication

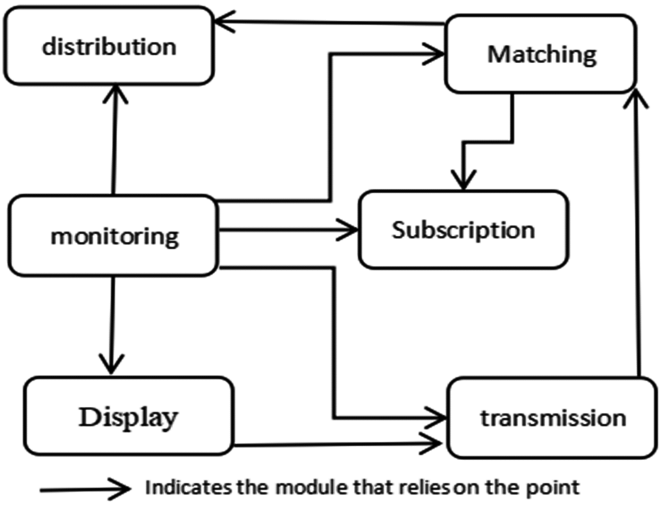


Fig. 3. The relationship between the modules

system between the modules. The Router module is responsible for receiving event matching results from the computing layer.

The Access Layer. The access layer is responsible for the reliable distribution of messages and consists of a message agent cluster. Each message agent maintains a TCP connection with the user through the user's heartbeat information, in order to achieve the message push.

4.2 Implementation

Message Distribution Module. The module receives information from publishers and subscribers for reliable distribution of messages, which is implemented by the access layer. we have use a rate-based, explicit and hop-by-hop congestion control algorithm, NDN Hop-by-Hop RCP, namely NHBH-RCP, to achieve high link utilization and increase overall network throughput [13]. the message agent is implemented using the Vert.x framework, Vert.x is an event-driven network programming framework.

Message Subscription Module. The module Responsible for subscription management, which is divided into message cache and persistent storage. The module is implemented by the system storage layer. In this paper, HBase as a storage system to achieve data storage and real-time query. There are three reasons for choosing HBase: (1) HBase supports the dynamic addition of columns. In the P/B system, the event and subscription information attributes can be dynamically modified, the use of HBase can save space and improve read efficiency. (2) HBase has a mass storage capacity. HBase stores all historical subscriptions, events, and matching results. (3) HBase cache mechanism can store the event message, which has not confirmed yet, to ensure the system's read and write throughput. HBase mainly contains four tables: event table, subscription relationship

table, matching results table and unconfirmed messages table. When the message agent receives a confirmation of a message, it will delete the record from HBase's unconfirmed message table. In order to ensure reliable transmission of messages, router periodically obtains unacknowledged messages from online subscribers for retransmission.

Message Matching Module. The module used to connect publishers and subscribers, It use the matching algorithm PADEM to find a matching successful subscriber. It is implemented by the computing layer, This paper uses the FlatMap-Filter calculation mode to further improve the parallelism and throughput of the computing layer. FlatMap is an one-to-many mapping operation and filter is a filtering operation. In this system, FlatMap will subscribe to only a few (usually one) nodes, not all nodes, and then it will find the matching subscribers by these matching nodes on the event Filter operation.

Message Transmission Module. The module responsible for the message transmission, which pushes the message to the subscriber based on the subscriber and user information that the matching module queries. This module provides reliable transmission service according to network quality. It is implemented by the routing layer. the composition of the routing layer is shown in Fig. 4.

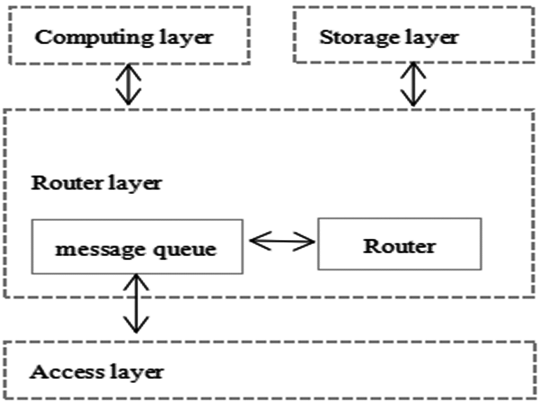


Fig. 4. The routing layer consists of modules

5 Experimental Results

In this section we will show the experimental results of the PSVA Framework. First, we build a Storm computer cluster, the cluster consists of 7 desktop computers, One server is deployed as a Message Agent, one server is used to deploy Message Queuing Kafka and ZooKeeper, and five machines are used to deploy the test client. The client is also implemented using the same Vert.x framework, but the running load is much lower than the Message Agent, thus ensuring the accuracy of the test results. each computer collocation Dual-core Intel 2.6 CPU GHz, 4G memory. The operating system is Ubuntu16.04LTS, JDK is Oracle-JDK8u121, Storm version is 1.0.2, Kafka version is 0.10.1. The NTP

protocol synchronization time between servers is less than 1 ms. This paper Use the LoadRunner12.53 to test the PSVA system, mainly to observe hits per Second, throughput, and Average Transaction Response Time.

Hits per second displays the number of hits made on the Web server by Vusers during each second of the load test. i.e. the number of requests initiated by the client to the server. throughput displays the amount of throughput (in bytes) on the Web server during the load test. Throughput represents the amount of data that the Vusers received from the server at any given second. This graph helps you to evaluate the amount of load Vusers generate, in terms of server throughput. Average Transaction Response Time displays the average time taken to perform transactions during each second of the load test.

Figure 5 shows a composite graph of hits per second and throughput. It can evaluate the amount of load generated by the virtual user. If the number of requests sent by the client is greater, the larger the throughput should be, the more requests will affect the average transaction response time. From the composite graph, the two graphics curves are normal and basically consistent. The experiment proved that the server can accept the client's request, and return the results, the system is stable.

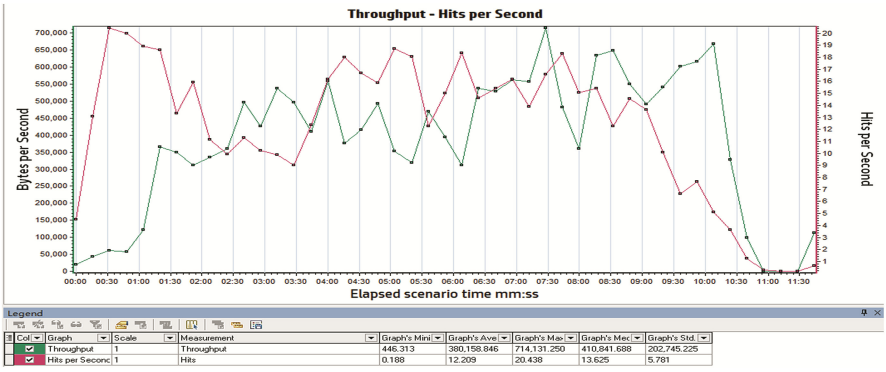
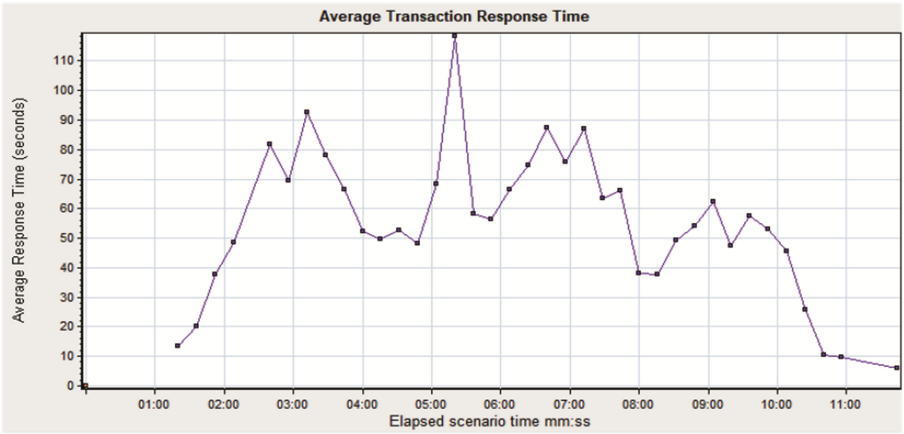


Fig. 5. Hits per second-throughput

Figure 6 shows a Average Transaction Response Time graph, It helps determine whether the performance of the server is within acceptable minimum and maximum transaction performance time ranges defined for your system.

In this experiment, we can see that the PSVA framework has a good degree of scalability through a test and analysis of hits per second, throughput, and Average Transaction Response Time. the experimental results show that the system is stable.



Color	Scale	Measurement	Graph's Minimum	Graph's Average	Graph's Maximum	Graph's Median	Graph's Std. Deviation
	1	Action_Transaction	6.002	54.882	118.339	54.320	24.333
	1	vuser_init_Transaction	0.013	0.013	0.013	0.013	0.000

Fig. 6. Average transaction response time

6 Conclusion

This paper first analyzes the existing problems of online advertising, then choose the P/B system to build a more comprehensive system for online video advertising, And we propose PSVA framework. The following describes the design and implementation of the PSVA framework. Finally, the effectiveness of PSVA is verified by experiments. the experimental results show that with the increase of the number of users in the cluster, the throughput can be increased, which shows that the system has good performance and scalability. The PSVA framework designed and implemented in this paper is preliminary. In Future, there are still a lot of work to be done on system availability and data center synchronization strategy design.

Acknowledgement. This work has been financially supported by Shenzhen Key Fundamental Research Project (Grant No. JCYJ20170412151008290, JCYJ20170306091556329 and JCYJ 20170412150946024).

References

1. Mei, T., Hua, X.S., Li, S.: VideoSense: a contextual in-video advertising system. IEEE Trans. Circuits Syst. Video Technol. **19**(12), 1866–1879 (2009)
2. Lei, K., Yu, L., Wei, J.: Scalable control panel for media streaming in NDN. In: 1st ACM ICN-2014, Paris, France, pp. 207–208 (2014)
3. Gai, K., Qiu, M., Zhao, H., et al.: Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. J. Netw. Comput. Appl. **59**(C), 46–54 (2016)
4. Gai, K., Qiu, M., Zhao, H.: Energy-aware task assignment for mobile cyber-enabled applications in heterogeneous cloud computing. J. Parallel Distrib. Comput. (2017)

5. Birman, K., Joseph, T.: Exploiting virtual synchrony in distributed systems. ACM SIGOPS Oper. Syst. Rev. **21**(5), 123–138 (1987)
6. Milo, T., Zur, T., Verbin, E.: Boosting topic-based publish-subscribe systems with dynamic clustering. In: ACM SIGMOD International Conference on Management of Data, pp. 749–760. ACM (2007)
7. Eugster, P.T., Guerraoui, R., Sventek, J.: Type-based publish/subscribe. Technical report, EPFL, Lausanne, Switzerland, June 2000
8. Rosenblum, D.S., Wolf, A.L.: A design framework for Internet-scale event observation and notification. In: ACM, pp. 344–360 (1997)
9. Banavar, G., Chandra, T., Mukherjee, B., et al.: An efficient multicast protocol for content-based publish-subscribe systems. In: Proceedings of the IEEE International Conference on Distributed Computing Systems, pp. 262–272. IEEE (1999)
10. Carzaniga, A., Rosenblum, D.S., Wolf, A.L.: Achieving scalability and expressiveness in an Internet-scale event notification service. In: Nineteenth ACM Symposium on Principles of Distributed Computing, pp. 219–227 (2000)
11. Cugola, G., Nitto, E.D., Fuggetta, A.: The JEDI event-based infrastructure and its application to the development of the OPSS WFMS. IEEE Trans. Softw. Eng. **27**(9), 827–850 (2001)
12. Yang, J., Fan, J., Li, C., et al.: A novel index structure to efficiently match events in large-scale publish/subscribe systems. Comput. Commun. **99**, 24–36 (2017)
13. Lei, K., Hou, C., Li, L., et al.: A RCP-based congestion control protocol in named data networking. In: International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, pp. 538–541. IEEE (2015)

Practice and Research on Private Cloud Platform Based on OpenStack

Zhe Diao(✉) and Youwei Zhu

University of International Relations, Beijing, China
diaozhe@uir.cn

Abstract. In recent years, cloud computing develops rapidly, and it has become increasingly popular research field in the IT field. This paper elaborates the concept and structure of cloud computing, cloud storage and private cloud, and discusses how to implement private cloud. This paper introduces the important role that OpenStack plays in private cloud construction, and puts OpenStack as an example to introduce the construction of private cloud platform.

Keywords: OpenStack · Cloud computing · Private cloud

1 Introduction

Cloud computing is a comprehensive reflection based on parallel computing, distributed computing, utility computing, grid computing and the integration of virtualization, network storage: load balancing and other technologies. With the rise and development of cloud computing technology, as an extension of the concept of cloud computing, cloud storage technology has gradually become a popular research topic. Cloud storage is a system that covers several important parts of servers, network devices, access networks, application software, public access interfaces, and client programs. Cloud storage is a system that covers several important parts of servers, network devices, access networks, application software, public access interfaces, and client programs.

At present, the financial, education and other areas are the most concentrated areas of network users, network, computing and storage and other resources are very rich, but the actual utilization of resources is not high, mainly reflected in: software utilization is low, the use of hardware resources Unbalanced, security problems happened frequently and so on. With the industry's own change and the arrival of large data age, the software system gradually tend to distributed, high stability, high availability architecture. Software testing is no longer just like a traditional system test in the past, but tends to be highly automated, fast feedback, real-world, and non-functional testing.

This paper intends to build a private cloud platform suitable for the enterprise in the experimental environment. It is suitable for data security, resource acquisition and so on. It is better to plan resources such as computing, storage and network, to explore the unified hardware and software resources, virtual management method. Based on the OpenStack open source tool package, this paper builds the private cloud computing

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 259–267, 2018.
https://doi.org/10.1007/978-3-319-73830-7_26

platform in the experimental environment, so that the resources can be distributed flexibly and ensure the data security.

2 Overview of Private Clouds

2.1 Definition of Private Cloud

The so-called private cloud, refers to the hardware resources to be virtualized for the internal staff to use, for the outside of the enterprise is a transparent cloud computing system. Providers and users of private cloud platform services are the same company or organization, so the private cloud resources for data security and service stability is much more effective compared to the control of public clouds. For businesses that create private clouds, it has all the infrastructure of a private cloud, so you can freely control projects and deployed applications. Enterprises can improve the utilization of resources through the private cloud, thereby achieving lower costs, improving enterprise information data security and enterprise core business competitiveness. The creation of a private cloud can be done either by the enterprise IT department or by the cloud computing service provider to create a good solution, then deliver and deploy it in the user's work environment.

2.2 The Core Technology of Private Cloud

Virtualization technology. Virtualization technology can achieve the separation between operating system, application software and the underlying physical equipment, so that the computing resources are transparent, scalable and efficient features. Virtualization has several different categories, in which full virtualization is a virtualization of all hardware devices, providing complete hardware conditions for each virtual machine; paravirtualizing the underlying basic hardware that achieves sharing and access to through virtual machine manager; operating system virtualization is the multiple servers arranging on virtualized operating system.

Data distribution and storage technology. The distributed storage technology can achieve massive storage and other operations, redundant storage can guarantee the high reliability of the data, and can make up for the unreliability of hardware facilities. Hardware load balancing in load balancing is to install load balancers directly between the server and the external network. The specialized tasks are done independently of the operating system, and the load balancing strategy is diversified. The traffic management is intelligent and the load balancing demands reach best. Software load balancing cluster can be implemented according to the master control machine loading global mobilization, the difficulty lies in the strength of schedule is harder to grasp through. Or by using the DHT method to achieve, the difficulty lies in the need for data and traffic estimation. In order to ensure reliability, needing to copy the redundant data, there are three replication modes: asynchronous, strong synchronization and semi-synchronization.

Platform management technology. The work of the server can achieve rapid deployment and operation of the business, and can quickly find the system failure and repair, to achieve large-scale system management.

Application of private cloud. At present, the application of private cloud mainly exists in the enterprise, the enterprise connects the computer to form an internal independent computer resource pool through a certain way. Set up a management node, so you can achieve centralized management and scheduling of computer resources within the enterprise, and using virtualization to provide users with services, this approach not only improves the efficiency of management and maintenance, but also increase the internal data security and make the internal resources to be shared, division of labor more reasonable and transparent.

3 OpenStack Open Source Project Profile

OpenStack is co-developed by an American National Aeronautics and Space Administration and Rack space, it is designed to provide open source projects for public and private clouds construction and management software. Its community has more than 130 companies and 1350 developers, these organizations and individuals will see OpenStack as the general front end of infrastructure as a Service (IaaS) resources. The first task of the OpenStack project is to simplify the cloud deployment process and bring good scalability.

Currently, OpenStack's core projects include: Nova, Swift, Glance, Keystone, Neutron, Cinder, Horizon, Metering: Ceilometer, Heat. OpenStack projects can be combined to work together to provide a complete cloud infrastructure services; and can work independently to provide services respectively.

OpenStack has a very good development trend and application space, its advantages are the following:

High standardization. OpenStack began with the support of many enterprises and open source organizations, and have been applied by more than 60 IT companies in the world. In the process, Dell, HP, and Cisco are all actively involved in its development of standards, which makes OpenStack have characteristics of a high standardization and a wide range of application.

Flexible management. OpenStack is fully open source, which allows anyone to participate in the development and maintenance of it, and modular design can be compatible with third-party technology to meet the changing needs of users.

Compatibility. The OpenStack project is compatible with other private and public cloud software, which allows it to migrate to other cloud computing platforms under specific conditions.

Flexible architecture. OpenStack provides multi-tier architecture design, the independent level of each other, to maintain loose coupling and its scalable performance is also good.

Resources fully utilized. Rapid access to a variety of resource nodes, support the deployment of resources in the existing work environment, integrate physically dispersed resources, while make full use of local resources.
The middleware platform supports secondary development. Because the technology the platform uses include HTTP services and FTP services, it is a separate package, with good versatility and supports the characteristics of secondary development.
Ease of operation. OpenStack supports WEB way to provide services, whether the user or administrator, the operation is very convenient, and can be completed by submitting a one-time processing of a job, the job offers download links to achieve ease of operation.

4 Realization of Private Cloud Platform

4.1 Hardware and Software Environment of Private Cloud Platform

Hardware equipment. Taking into account the service capacity of the private cloud platform, the hardware devices are shown in Fig. 1 below

设备名称	指标	数量	备注
主机设备	DL580 G8服务器，每台CPU:4*15C；内存：512G、硬盘：5*900G。	2	1）CPU必须支持VT技术。 2）为了提高系统的稳定性，我们用4块硬盘做了raid
网络设备	48口千兆三层交换机	3	提供内外网连接
防火墙	吞吐量2048Mbps,4个千兆端口	1	隔离内外网设备，负责与其他网络互联

Fig. 1. Private cloud platform hardware

Software system. Private cloud platform is a set of open source system architecture, we use the framework of all open source software. As shown in Fig. 2.

软件名称	功能	版本	备注
OpenStack	为云测试平台提供基础设施服务。	Liberty	目前社区最新版本为M版，我们决定落后社区一个版本，以保证稳定性。
Zabbix	为云测试平台提供系统级监控。	2.4.5	弥补OpenStack Celimeter不能对硬件资源进行监控。
ELK	日志分析平台	2.x	ELK由ElasticSearch、Logstash、Kibana以及Nigix组成，分布式日志收集平台。

Fig. 2. Private cloud platform software system

4.2 Private Cloud Platform Deployment Program

Private cloud platform mainly uses two node deployment plan. One of the nodes will be served as the management, network, computing and storage functions, another node will act as a network dual live and computational and storage functions. The main deployment structure shown in Fig. 3.

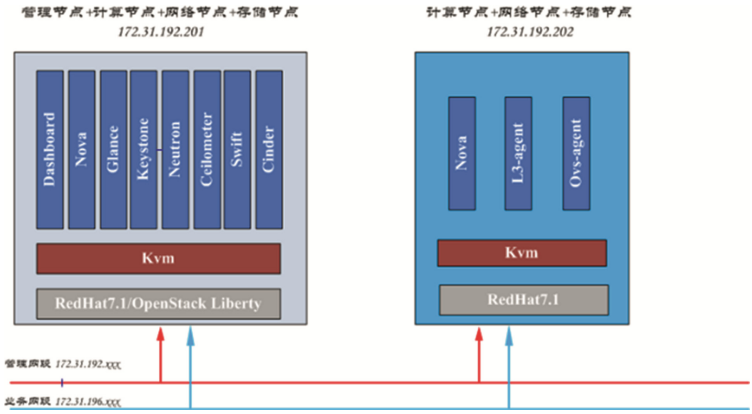


Fig. 3. Two-node deployment plan of private cloud platform

4.3 Logical Architecture of Private Cloud Platform

Since the OpenStack provide infrastructure services in a private cloud platform. Which itself consists of multiple components, in the implementation of OpenStack process can be based on their own business needs to select the appropriate components. It consists of multiple components and can select the appropriate components based on their own

business needs to select the appropriate components in the implementation of OpenStack process. Figure 4 is the logical structure of the private cloud platform.

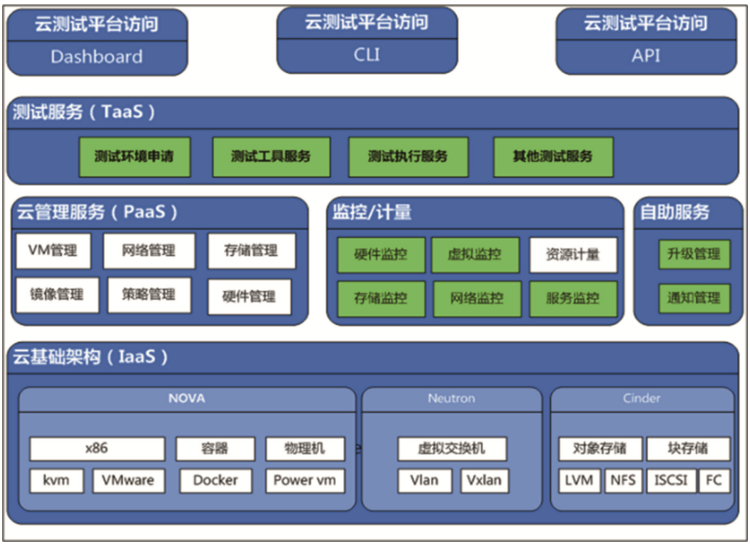


Fig. 4. Private cloud platform logical architecture

In the IaaS layer, we mainly selected NOVA as a computing resource management function, Neutron for network virtualization, Cinder for block storage. The three core components pooled the hardware computing resources, network resources and storage resources.

At the PaaS layer, OpenStack's Celimeter component provides VMM-level resource monitoring, and in order to make it impossible to monitor the underlying hardware resources, we use Zabbix as enterprise-class system resource monitoring. We have done some secondary development at the same time in the OpenStack component upgrades, notification services, etc.

At the TaaS layer, we offer testing services, we invoke its APIs through OpenStack's Heat component to integrate test tools, such as test management tools TestLink, CI tool Jenkins, and automated test tool Fitnesse.

Users can use the private cloud platform in three ways, they can access to Horizon components that OpenStack provide, the Dashboard, the command line and the way of API.

4.4 Plan of Private Cloud Platform Network Implementation

Network implementation is the core technology of private cloud platform and is also one of the difficulties, here is a brief introduction to the private cloud platform network

implementation. OpenStack supports four kinds of network virtualization implementation, namely FlatDHCP, GRE, VLAN, and VXLAN. The private cloud platform mainly uses OVS VXLAN protocol.

VXLAN is a tunnel forwarding mode that encapsulates Ethernet packets on the UDP transport layer. It uses 24-bit bits to identify Layer 2 network segments. Using VXLAN can overcome the limitations of VLANs with only 4000 available VLAN IDs. Of course, for small businesses private cloud VLAN can meet the needs of the network. Using VXLAN can overcome the limitations of VLANs with only 4000 available VLAN IDs. Of course, for small businesses private cloud VLAN can meet the needs of the network.

How the private cloud platform to achieve the main network is shown in Fig. 5. It mainly describes how the virtual machine on the compute node communicates with the external network. The calculation node mainly consists of two bridges: the integrated bridge br-int and the tunnel bridge br-tun. The rule of integrated bridge br-int is relatively simple and it is used as a normal Layer 2 switch. Br-tun acts as a virtual bridge, the rules of it are a bit more complicated. Screening the internal network packet reasonably will be carried with the corresponding internal vlan tag, thrown out from the correct tunnel; will change the corresponding network packet outside with the correct tunnel number to the corresponding internal vlan tag.

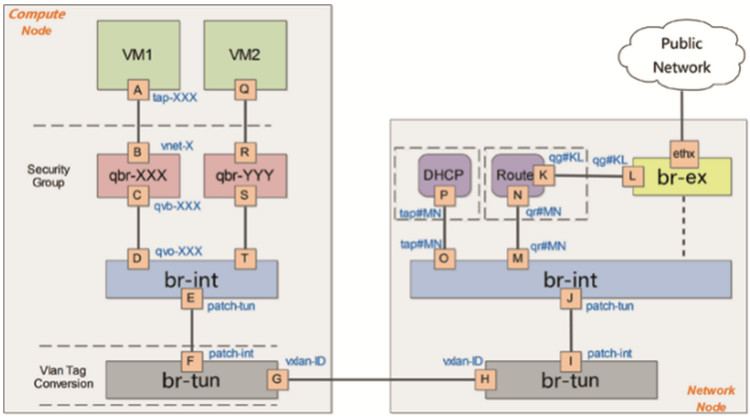


Fig. 5. Private cloud platform network implementation

The network node is responsible for the tasks of the network service, including DHCP, routing and advanced network services. It generally includes three bridges: br-tun, br-int and br-ex. Br-int and br-tun are similar to the two bridge functions on the compute node. Br-ex mainly has two core interfaces, one is mounted on the physical interface, such as eth0, the network packet will be sent from this interface to the external network. The other is such interface like qg-xxx, is connected to the router service network space, which binds a router's external IP, as NAT address, in addition, is also on the network space.

5 Concluding Remarks

In recent years, cloud computing has been in rapid development and wide application, for enterprises and peer education industry, to build a private cloud platform can effectively reduce the cost of updating equipment, improve equipment utilization and resource utilization. By analyzing the definition and characteristics of private cloud and OpenStack open source projects, this paper details the need for private cloud systems building and major application environments, then, with OpenStack, for example, from both physical and logical architecture describes the architecture of private cloud platform for the cloud computing platform to build the accumulation of experience.

Acknowledgment. This work was supported by the National Key Research and Development Program of China (No. 2016QY04W0805) and the Program (No. Z16051).

References

1. Xiong, M., Fang, B., Zhang, Y., Wu, J., Li, S.F.: OpenStack authentication security problem research. Post Telecommun. Des. Technol. (2014)
2. Xu, L., Wang, L.: Based on OpenStack's private cloud building research. Inf. Commun. (2014)
3. Liang, Y., Yang, H., Li, H., Lan, G.: Based on OpenStack resource monitoring system. Comput. Syst. Appl. (2014)
4. Zhang, X., Zheng, X., Li, X.: OpenStack cloud virtual machine security strategy research. Inf. Technol. (2014)
5. Zhang, X., Zheng, X., Li, X.: Based on GlusterFSOpenStack platform design. Microcomput. Appl. (2014)
6. Zhao, S., Li, L., Ling, X., Xu, C., Yang, J.: Based on OpenStack Tsinghua cloud platform construction and scheduling design. Comput. Appl. (2013)
7. Zhang, Y., Wu, Q., Tan, Y.: Based on the software injection of on-demand virtual machine instance system. Comput. Technol. Dev. (2013)
8. Jiang, Y., Wang, W., Cao, L., Liu, K., Chen, G.: Based on open source software, the construction of private cloud computing platform. Telecommun. Sci. (2013)
9. Li, Z., Zhao, J.: OpenStack open source cloud computing platform. Softw. Guide (2012)
10. Rao, H., Wang, Y., Qi, J., Yin, Y.: Cloud service matching method for time-varying demand. China Manag. Sci. (S1) (2012)
11. Wang, H.: Cloud computing resource management subsystem research and implementation. Beijing Jiaotong University (2012)
12. Liu, F.: Cloud computing virtual machine deployment mechanism. Taiyuan University of Technology (2012)
13. Wang, X.: Based on OpenStack to build a private cloud computing platform. South China University of Technology (2012)
14. Gao, G.: Based on OpenStack computing cloud research and implementation. Chengdu University of Technology (2012)
15. Chen, Z.: Based on the cloud computing virtual desktop platform design and implementation. Fudan University (2012)
16. Cao, F.: Hadoop-based cloud computing model research and application. Chongqing University (2011)

17. Shi, Y.: Cloud computing research and Hadoop application development and testing. Beijing University of Posts and Telecommunications (2011)
18. Wu, J.: Cloud computing automation software installation system design and implementation. Hebei University of Technology (2011)
19. Wu, G.: MapReduce parallel programming model in cloud computing. Henan Polytechnic University (2010)

Approach for Semi-automatic Construction of Anti-infective Drug Ontology Based on Entity Linking

Ying Shen[1], Yang Deng[1], Kaiqi Yuan[1], Li Liu[2], and Yong Liu[3(✉)]

[1] School of Electronics and Computer Engineering (SECE), PKU Shenzhen Graduate School, Shenzhen 518055, People's Republic of China
shenying@pkusz.edu.cn, {Ydeng,kqyuan}@pku.edu.cn

[2] Institute of Education, Tsinghua University, Beijing 100084, People's Republic of China
li-liu16@mails.tsinghua.edu.cn

[3] IER Business Development Center, Shenzhen, People's Republic of China
13312962646@189.cn

Abstract. Ontology can be used for the interpretation of natural language. To construct an anti-infective drug ontology, one needs to design and deploy a methodological step to carry out the entity discovery and linking. Medical synonym resources have been an important part of medical natural language processing (NLP). However, there are problems such as low precision and low recall rate. In this study, an NLP approach is adopted to generate candidate entities. Open ontology is analyzed to extract semantic relations. Six-word vector features and word-level features are selected to perform the entity linking. The extraction results of synonyms with a single feature and different combinations of features are studied. Experiments show that our selected features have achieved a precision rate of 86.77%, a recall rate of 89.03% and an F1 score of 87.89%. This paper finally presents the structure of the proposed ontology and its relevant statistical data.

Keywords: Data mining · Ontology construction · Entity discovery
Entity linking

1 Introduction

It is widely pointed that the classical ontologies cannot sufficiently handle imprecise and vague knowledge for some real world applications, but domain ontology can effectively resolve data and knowledge problems with uncertainty.

Ontology construction can be performed automatically, semi-automatically and manually. Automatic construction of ontology, being the research focus and development trend of search engines, has many difficulties, for example, entity discovery and linking based on analysis of unstructured data and relationship study on taxonomy based on web text or Wikipedia [1] The above research issues are more complicated in the medical field:

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 268–277, 2018.
https://doi.org/10.1007/978-3-319-73830-7_27

- Medical entities are partly in disordered word form, with missing heuristic information and a diversified demonstrative appearance of technical terms. The medical data are characterized by large quantity, diversified data type, and high redundancy [2]. The medical data not only include diagnosis and treatment data such as medical orders, nursing records, hospitalization condition, and medical condition but also contain all data produced in the medical system by the doctor, patient, medical facility, management and service personnel. The storage format of the data is diverse, mainly caused by diversification of symptoms, miscellaneous categories of department and cross-field problems and diversified treatment methods.
- Complicated and uncertain diagnosis and therapy reasoning but strict requirements on decision accuracy. A typical clinical cycle includes diagnosis, prognosis, treatment and therapy follow-up. Specific syndromes and signs lead to intricacy of diagnosis and treatment flow. In addition, one has to take into account the laboratory examination data, family history, genes, epidemiologic data, existing medical literature, etc. [3].

This study mainly involves medical ontology construction. For the medical data collected from electronic medical records (EMRs) and the internet, a machine learning approach is adopted to carry out the entity discovery and recognition. Clinical information such as the disease and its syndrome and patient conditions are extracted. The cancer ontology is constructed based on the schema proposed by doctors.

The main contributions of this work can be summed as followed:

- Methods of the entity/relation extraction and entity linking is proposed to construct a domain ontology, which can effectively improve the knowledge representation.
- For medical entity linking, word vector features and word-level features are used for English medical synonym identification. The result of synonyms with single feature and different combinations of features are studied.
- An anti-infective drug ontology is semi-automatic constructed. The structure of the proposed ontology is visible to effectively provide clinical information.

The rest of our paper is structured as follows: Sect. 2 discusses the related work, Sect. 3 gives a detailed description of the overall framework of our method, Sect. 4 presents experimental setup, results and analysis, and Sect. 5 summarizes this work and the future direction.

2 Related Work

Entity discovery and recognition are fundamental tasks of NLP. In recent years, the entity discovery technique has been widely used in the biomedical domain, which recognizes the names of entities such as protein, DNA, RNA and cell line [4]. Clinicians and healthcare providers often find filtering and retrieving useful knowledge from clinical documents such as EMRs to be difficult. Therefore, entity discovery, which has been actively employed to unlock information from free-text, could be used to address this problem and consequently to improve clinical care [5].

Several studies have been conducted to apply rule- and/or ML-based approaches to EMRs to assist scientists in the extraction of valuable information. For example, deBruijn et al. [6] developed a discriminative semi-Markov hidden Markov model (HMM) based on a wide range of features generated from both training texts and external knowledge sources to identify clinical concepts in discharge summaries and progress reports. They observed that projecting textual features onto a high-dimensional feature space and the utilization of external sources for semantic and syntactic tagging are beneficial.

Jiang et al. [7] proposed a hybrid clinical entity extraction system combining an ML-based named entity recognizer with rule-based methods for post-processing. Two ML algorithms, conditional random fields (CRF) and SVM, were applied. Their results suggest that CRF and SVM can significantly enhance the performance of NER. For the entity linking between data of different encyclopedias and articles of different titles [8], SVMs have exhibited high performance in various classification tasks [9]. Meanwhile, many algorithms have been proposed to optimize the loss function of SVM, such as sequential minimal optimization [10] and cutting plane [11].

Instead of developing new ML- or rule-based methods, some researchers have focused on combining existing approaches. Kang et al. [12] integrated six named entity recognizers and chunkers, including both ML- and thesaurus-based methods, to annotate clinical records.

Based on entity linking, we proposed an approach for semi-automatic construction of anti-infective drug ontology, which is beneficial to deepen the work like improving data governance in large organizations through ontology and linked data [13], preventing electronic health record error using ontology in big data [14], and realizing drug side effects data representation and full spectrum inference using ontology in intelligent telehealth [15].

3 Methods

3.1 Entity and Relation Discovery

Based on EMR, an NLP approach is employed to generate candidate entities to construct an ontology. To process the English language, morphological analyses and tagging are performed to identify the concepts that characterize the domain and constitute the arguments of relationships. The proposed morphological analyzer referring to an existing dictionary can recognize the variant forms of a given word and simplify the word by identifying its stem. The running results of POS tagging tag the given words as different types, such as nouns (identifying the objects' classes), adjectives (describing the objects' attributes or properties), verbs or adverbs (presenting the actions and associations).

To extract the relations, an open ontology - disease ontology (DO) [16] is analyzed. This high-quality data resource contains reliable semantic relations, including hyponymy, synonym, whole-partial relation and property-body relation. We also manually define the hyponymy of important entities according to medical structurization information. Relations can identify the properties of concepts and instances and clarify the semantic structure. To construct an ontology, several relations must be taken into

account, including is-a (generalization/specialization), is-part-of (composition/decomposition), has-a (possession/dispossession), and the causal and temporal relations.

In ontology, classes are placed in an inheritance hierarchy. Graphical modeling languages such as UML [17] can represent the concepts and their relations. This study adopts some rules for the determination and extension of hierarchies, such as transitivity of hierarchical relation, deductive reasoning, and analogy reasoning.

A bottom-up approach is adopted to instantiate objects and unite object classes to their super classes, while a top-down approach is adopted to verify hierarchies and carry out the domain axiomatization with objects, object types, relations and interactions.

3.2 Entity Linking

SVM is used to determine whether a pair of words is synonymous by means of the selected features, and the precision rate, the recall rate and the F1 score are calculated. SVM is used as the classifier with the radial basis kernel function. In total, 6 features are selected: 2 word vector features (cosine similarity and cosine similarity of two sets) and 4 English word-level features (the duplicate word feature, the subsequence feature, the first character feature, and the abbreviation feature).

Acquisition of word vectors requires a large-scale corpus for training the model. Except for the DO ontology mentioned above, the English corpus is derived from DailyMed, WikiDisease, WebMD, and MayoClinic. A total of 1327 synonym pairs are used as positive samples, and several terms are selected from the Cambridge Dictionary together with medical terms to randomly form term pairs as negative samples. Thus training and test datasets are constructed, and the ratio of positive to negative is 1:1.

Word Vector Calculation. The process of analyzing textual content can be simplified into vector operations in K-dimensional vector space through training. Distributed representation is adopted here by using a three-layer neural network consisting of an input layer, a hidden layer and an output layer. The neural network models take as their input a large corpus of text and produce a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space.

Cosine Similarity. The similarity in the word vector space can be used to represent the semantic similarity of text. Cosine similarity measures the similarity between two non-zero vectors by computing the cosine of the angle between them, namely, the inner product. The cosine similarity of two multidimensional word vectors A and B is defined as follows:

$$CosSim(A, B) = \frac{\mathrm{A} \cdot \mathrm{B}}{\|A\|\|B\|} = \frac{\sum_{\mathrm{i}=1}^{n} A_i B_i}{\sqrt{\sum_{\mathrm{i}=1}^{n} A_i^2}\sqrt{\sum_{\mathrm{i}=1}^{n} B_i^2}} \tag{1}$$

It can be seen that the cosine of 0° is 1, and it is not greater than 1 for any other angle. It is thus a judgment of orientation but not scale: two vectors with the same orientation

have a cosine similarity of 1; two vectors at 90° have a similarity of 0; and two vectors diametrically opposed have a similarity of –1, independent of their scale. The larger the value is, the more similar the two words are.

Referring to [18], in our work, given two translated terms sets EnList1 and EnList2, we estimate their similarity by using the average vector of EnList:

$$AvgVec = \frac{1}{n} \sum_{i=1}^{n} term_i, term_i \in EnList \tag{2}$$

Cosine similarity of two sets, represented by two average vectors, can be calculated using formula (1).

English Morphological Feature. In linguistics, morphology is the study of the structure of words, the rules by which words are formed, and the relationships between words. Morphology also considers parts of speech, intonation and stress, and the ways context can change a word's pronunciation and meaning. Several features are adopted, including the following:

a. **Duplicate word feature**: Returns 1 if there are duplicate words in two translated sets; otherwise, 0.
b. **Subsequence feature**: If one term is the subsequent to another term, returns 1; otherwise, 0.
c. **First character feature**: If all the first characters in each word from a and b match each other, returns 1; otherwise, 0 [19]. For example, "liver cancer" and "liver carcinoma", sharing the same first characters "lc", returns 1.
d. **Abbreviation feature**: If all the upper case characters from a and b match each other, returns 1; otherwise, 0. For example, m4 = 1 for "USA" and "United States of America".

3.3 Ontological Construction

With the knowledge elements (e.g., class, relation, hierarchy), the process of ontological construction (Fig. 1) is illustrated by the UML sequence diagram, which is an object-oriented language representing the development and deployment of the system. The initial ontology can be expanded through acquiring knowledge. The entities and relations of new input clinical texts are first analyzed and extracted and then placed in the existing hierarchies.

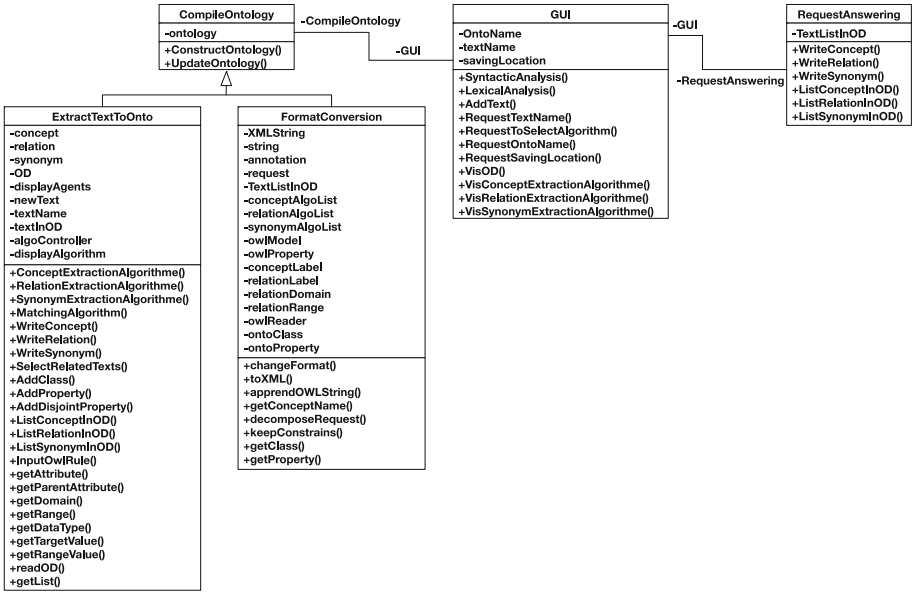


Fig. 1. UML class diagram: prototype of ontology construction and query answering

4 Experiments and Evaluation

4.1 Entity Discovery and Linking

In order to observe the contribution of each feature, and identify which combinations of features are more effective, we performed 2 ^ 6 experiments with or without one or more specific features. In each case, precision rate, recall rate and F1 score are calculated respectively. The results for the single feature are shown in Table 1. Among the six features, cosine similarity brings about the best results. The English feature like the number of abbreviations, first characters, and the duplicate words can bring help in a certain range.

Table 1. Precision, recall and F1 of single feature

Features	Precision (%)	Recall (%)	F1(%)
Cosine similarity	**85.79**	**82.16**	**83.94%**
Cosine similarity of two sets	61.56	39.63	48.22%
Duplicate word feature	48.62	36.47	41.68%
Subsequence feature	52.78	49.81	51.25%
First character feature	59.29	45.23	51.31%
Abbreviation feature	54.66	47.57	50.87%

As for the combinations of features, Table 2 shows the best results in 2 ^ 6 feature combinations, sorted by precision. The best combination is {cosine similarity, duplicate

word, subsequence feature, first character feature, abbreviation feature}, where the accuracy is 86.77%, the recall rate is 89.03%, the F1 score is 87.89%.

Table 2. Performance of feature combination

Features	Precision (%)	Recall (%)	F1(%)
Cosine + first character	71.54	74.47	72.98
Cosine + subsequence + abbreviation	78.35	75.48	76.89
Cosine + duplicate + subsequence + abbreviation	82.92	83.50	83.21
Cosine + duplicate + subsequence + abbreviation + first character	**86.77**	**89.03**	**87.89**

4.2 Ontology Knowledge

Take the ontology construction in the field of anti-infective drugs as an example. We studied 39 types of antibiotics, 19,308 types of relation between drugs (e.g. 952 "subClassOf" hyponymy relation), and 8 types of property (Table 3).

Table 3. Anti-infective drug ontology

	Class	Axiom	Logical axiom	Annotation axiom	SubClassOf	Object property	Hidden GCI
Ontology	39	9178	952	8226	952	8	0

As shown in Fig. 2, one of the anti-infective drug ontology class "Amoxycillin" and its relevant relations is indicated as follows: the green nodes represent the contraindication between Amoxycillin and other drugs (relation_contraindication_drug); the red nodes point to the bacteria which can be treated by Amoxycillin (relation_antibiotics_bacteria); the pink nodes indicate the relationship between infection site and bacteria (relation_infs_bacteria); the gray nodes specify the relationship between disease and bacteria (relation_disease_bacteria); and the yellow nodes show the relationship between disease and complication (relation_disease_complication).

For the proposed ontology, since new clinical knowledge cannot be applied automatically without clinical verification, the hierarchy expansion is implemented with new input data, which should be supported by clinical evidence and supervised by a knowledge engineer.

Fig. 2. Antibiotics ontology-Amoxycillin (Color figure online)

5 Conclusion

This article presents research about knowledge processing to realize the domain ontology construction. Experimental results show that the selected features can improve the precision rate, recall rate and F1 score of entity linking. Full use is made of the English characteristics of abbreviation, first character, subsequence, and duplicate words, which have good classification effects. Word vectors are used to fully capture semantic information.

In future work, experiments will be focused on more specific clinical fields such as the extraction of symptoms, complications and disease names [20, 21] to further enhance the accuracy of synonym identification.

Acknowledgement. This work was financially supported by the National Natural Science Foundation of China (No. 61602013), and the Shenzhen Key Fundamental Research Projects (Grant No. JCYJ20160330095313861, and JCYJ20151030154330711).

References

1. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. In: AAAI, no. 6, pp. 775–780 (2006)
2. Zhou, G., Zhang, J., Su, J., Shen, D., Tan, C.: Recognizing names in biomedical texts: a machine learning approach. Bioinformatics **20**(7), 1178–1190 (2004)
3. Zhou, L., Hripcsak, G.: Temporal reasoning with medical data—a review with emphasis on medical natural language processing. J. Biomed. Inform. **40**(2), 183–202 (2007)
4. Denny, J.C., Peterson, J.F., Choma, N.N., Xu, H., Miller, R.A., Bastarache, L., Peterson, N.B.: Extracting timing and status descriptors for colonoscopy testing from electronic medical records. J. Am. Med. Inform. Assoc. **17**(4), 383–388 (2010)
5. Krallinger, M., Erhardt, R.A.A., Valencia, A.: Text-mining approaches in molecular biology and biomedicine. Drug Discov. Today **10**(6), 439–445 (2005)
6. de Bruijn, B., Cherry, C., Kiritchenko, S., Martin, J., Zhu, X.: Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. J. Am. Med. Inform. Assoc. **18**(5), 557–562 (2011)
7. Jiang, M., Chen, Y., Liu, M., Rosenbloom, S.T., Mani, S., Denny, J.C., Xu, H.: A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries. J. Am. Med. Inform. Assoc. **18**(5), 601–606 (2011)
8. Chang, K.W., Samdani, R., Rozovskaya, A., Rizzolo, N., Sammons, M., Roth, D.: Inference protocols for coreference resolution. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, pp. 40–44 (2011)
9. Kudo, T., Matsumoto, Y.: Chunking with support vector machines. In: Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies, pp. 1–8 (2001)
10. Cao, L.J., Keerthi, S.S., Ong, C.J., Zhang, J.Q., Periyathamby, U., Fu, X.J., Lee, H.P.: Parallel sequential minimal optimization for the training of support vector machines. IEEE Trans. Neural Netw. **17**(4), 1039–1049 (2006)
11. Franc, V., Sonnenburg, S., Werner, T.: Cutting plane methods in machine learning. In: Sra, S., Nowozin, S., Wright, S.J. (eds.) Optimization for Machine Learning, pp. 185–218. MIT Press, Cambridge (2011)
12. Kang, N., Barendse, R.J., Afzal, Z., Singh, B., Schuemie, M.J., van Mulligen, E.M., Kors, J.A.: Erasmus MC approaches to the i2b2 challenge. In: Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data, Boston, MA, USA. i2b2 (2010)
13. DeStefano, R.J., Tao, L., Gai, K.: Improving data governance in large organizations through ontology and linked data. In: 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud), pp. 279–284 (2016)
14. Gai, K., Qiu, M., Chen, L.C., Liu, M.: Electronic health record error prevention approach using ontology in big data. In: High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conference on Embedded Software and Systems (ICESS), pp. 752–757 (2015)
15. Jayaraman, S., Tao, L., Gai, K., Jiang, N.: Drug side effects data representation and full spectrum inferencing using knowledge graphs in intelligent telehealth. In: 2016 IEEE 3rd International Conference Cyber Security and Cloud Computing (CSCloud), pp. 289–294 (2016)

16. Schriml, L.M., Arze, C., Nadendla, S., Chang, Y.W.W., Mazaitis, M., Felix, V., Feng, G., Kibbe, W.A.: Disease ontology: a backbone for disease semantic integration. Nucleic Acids Res. **40**(D1), D940–D946 (2011)
17. Shao, Y., Lei, K., Chen, L., Huang, Z., Cui, B., Liu, Z., Tong, Y., Xu, J.: Fast parallel path concatenation for graph extraction. IEEE Trans. Knowl. Data Eng. **29**(10), 2210–2222 (2017)
18. Dumas, M., ter Hofstede, A.H.M.: UML activity diagrams as a workflow specification language. In: Gogolla, M., Kobryn, C. (eds.) UML 2001. LNCS, vol. 2185, pp. 76–90. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45441-1_7
19. Henriksson, A., Skeppstedt, M., Kvist, M., Duneld, M., Conway, M.: Corpus-driven terminology development: populating Swedish SNOMED CT with synonyms extracted from electronic health records. In: Proceedings of the 2013 Workshop on Biomedical Natural Language Processing (BioNLP), pp. 36–44. Association for Computational Linguistics (2013)
20. Henriksson, A., Moen, H., Skeppstedt, M., Eklund, A.M., Daudaravicius, V., Hassel, M.: Synonym extraction of medical terms from clinical text using combinations of word space models. In: Proceedings of the 5th International Symposium on Semantic Mining in Biomedicine (2012)
21. Doğan, R.I., Leaman, R., Lu, Z.: NCBI disease corpus: a resource for disease name recognition and concept normalization. J. Biomed. Inform. **47**, 1–10 (2014)

Constructing Ontology-Based Cancer Treatment Decision Support System with Case-Based Reasoning

Ying Shen[1], Joël Colloc[2], Armelle Jacquet-Andrieu[3], Ziyi Guo[1], and Yong Liu[4(✉)]

[1] School of Electronics and Computer Engineering (SECE), Peking University Shenzhen Graduate School, Shenzhen 518055, People's Republic of China
shenying@pkusz.edu.cn, guoziyi@pku.edu.cn

[2] CIRTAI - Université du Havre, 25 Rue Philippe Lebon, 76086 Le Havre Cedex, France
joel.colloc@univ-lehavre.fr

[3] MoDyCo - Université Paris Ouest (UMR CNRS 7114), 200 avenue de la République, 92000 Nanterre, France
armelle.jacquet@u-paris10.fr

[4] IER Business Development Center, Shenzhen, People's Republic of China
13312962646@189.cn

Abstract. Decision support is a probabilistic and quantitative method designed for modeling problems in situations with ambiguity. Computer technology can be employed to provide clinical decision support and treatment recommendations. The problem of natural language applications is that they lack formality and the interpretation is not consistent. Conversely, ontologies can capture the intended meaning and specify modeling primitives. Disease Ontology (DO) that pertains to cancer's clinical stages and their corresponding information components is utilized to improve the reasoning ability of a decision support system (DSS). The proposed DSS uses Case-Based Reasoning (CBR) to consider disease manifestations and provides physicians with treatment solutions from similar previous cases for reference. The proposed DSS supports natural language processing (NLP) queries. The DSS obtained 84.63% accuracy in disease classification with the help of the ontology.

Keywords: Clinical decision support · Data mining · Ontology construction
Decision support system · Case-Based Reasoning

1 Introduction

Clinicians cannot obtain knowledge regarding clinical processes from the existing, inadequate medical databases. In current decision support systems (DSSs) [1, 2], databases and computer records work together to facilitate decision-making by improving access to relevant data through defined interfaces. We focus on researching the problem of classification and diagnosis. After investigating the uncertainty about the actual situation of the study object (patient, organ, population), it is necessary to distinguish the possible symptoms and diseases from the impossible ones to determine effective measures [3].

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 278–288, 2018.
https://doi.org/10.1007/978-3-319-73830-7_28

This study proposes an ontology-based DSS to aid in cancer diagnosis and therapy. Several existing ontologies and high-quality data resources are available for use, including the Disease Ontology (DO) [4], the NCBI organismal classification ontology (NCBI Taxonomy) [5], the Human Phenotype Ontology (HPO) [6], and DrugBank [7], as well as some useful websites such as the U.S. National Library of Medicine (NLM) [8] and Wikipedia [9]. We utilize ontology triples to improve the reasoning ability of the DSS. Afterwards, Case-Based Reasoning (CBR) is employed to provide physicians with treatment solutions from similar previous cases for reference. A case study concerning the answering of clinical queries about gastric cancer (diagnosis, prognosis and treatment) is developed to illustrate the implementation of DSS with an ontology.

The main contributions of this work can be summed as followed:

- Base the medical reasoning framework which defines rules and interfaces, the combined operation of DSS, CBR and ontology can aide clinical decision-making.
- To simplify the semantic representation of medicines, existing biomedical ontology is reused in DSS.
- The comparison between DSS with or without ontology presents that our model can fully utilize the ontology information and provides a stable performance in diagnosis and disease classification.

The rest of this paper is structured as follows: Sect. 2 describes the related work. Section 3 introduces the details of the proposed methods; Experimental results and evaluations are presented in Sect. 4. Finally, we conclude the paper in Sect. 5.

2 Related Work

The application of DSS for the cooperation of ontology has been investigated in many works shown below. Farion et al. [10] used ontology-driven design to represent components of a CDSS. A prototype of the system was implemented for two clinical decision problems and settings (triage of acute pain in the emergency department and postoperative management of radical prostatectomy on the hospital ward). Haghighi et al. [11] have designed a knowledge acquisition tool to facilitate the creation and maintenance of a knowledge base by the domain expert and its sharing and reuse by other institutions. They used the Unified Medical Language System (UMLS) which contains the domain entities and constitutes the relations repository. Lee and Wang [12] presented a novel fuzzy expert system for diabetes decision support application. The proposed fuzzy expert system can work effectively for diabetes decision support application. Jayaraman et al. [13] realized the drug side effects data representation and full spectrum inference using ontology in Intelligent Telehealth. The proposed model allows the doctors and caregivers to derive dynamic information about side effects avoiding costly errors caused by human interpretation. Ontology is used to prevent electronic health record error approach [14] etc. Besides, ontology can be used in other domain for decision supports. Ontology is adopted to control the intercrossed access for secure financial services on multimedia big data in cloud systems [15], and to classify cyber incident for cybersecurity insurance in financial industry [16].

In the development of a DSS for medical decision-making, an approach using a quantitative model analysis has increasingly gained attention. Some tools, technologies and concepts, such as decision trees and Bayes [17] and probability theories, can improve the operation of medical decision-making. CBR computes component similarities by exploring indexed knowledge. CBR is a qualitative and quantitative mixed model of experience storage and retrieval. This method is analogous to problem-solving methods that compare new cases with previously indexed cases [18]. CBR involves semantic distances developed using different approaches, including algorithms of structural similarity, statistical learning, digital approaches from neural networks, and fuzzy logic. The research on semantic distances often combines symbolic and numeric aspects [19].

3 Methods

3.1 Architecture of DSS

The architecture of DSS is presented in Fig. 1 and consists of the knowledge acquisition model, the ontology model and the interconnection model. Each part can be considered as an independent computing environment. The knowledge acquisition module provides medical data to the "Interpreter module" and the "Inference engine". The DSS repository recognizes, processes and stores medical data. The "Interpreter module" operates as an algorithm controller to perform query analysis and knowledge extraction. The "Inference engine" mainly normalizes the interrogation flow and formulates the relationships among the diagnosis, prognosis and treatment. It uses a case-based inference rule to analyze the likelihood and symptoms of complications of the current diseases so that medical care personnel can determine a prognosis or implement preventive measures.

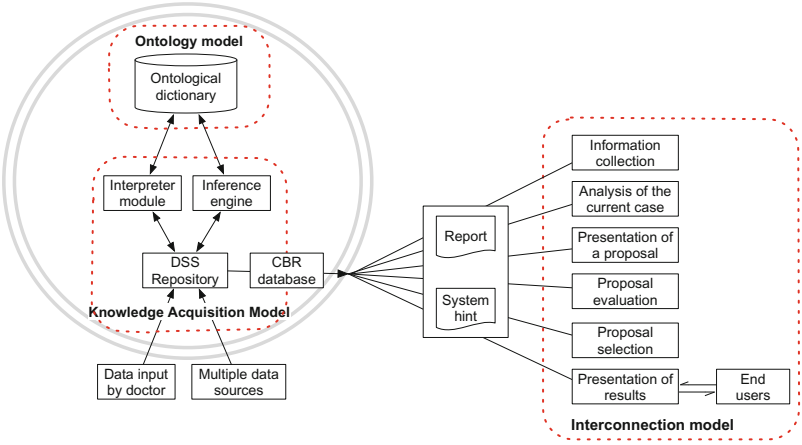


Fig. 1. Architecture of DSS

3.2 Medical Reasoning Framework

Based on our previous work [3], Fig. 2 presents the relations between diagnosis (Δ), prognosis (Π), and treatment (Θ) and describes a series of medical inference rules using a simulated treatment approach.

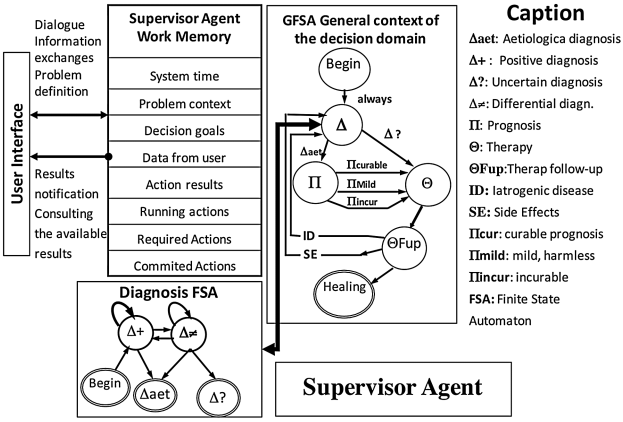


Fig. 2. Reasoning between diagnosis, prognosis and treatment

CBR successively stores and indexes clinical cases and knowledge in different directories (Δ, Π, Θ, SΘ) by identifying keywords related to the problem of the case (PB), the environment (patient record) (E) and the result (R) (Fig. 3).

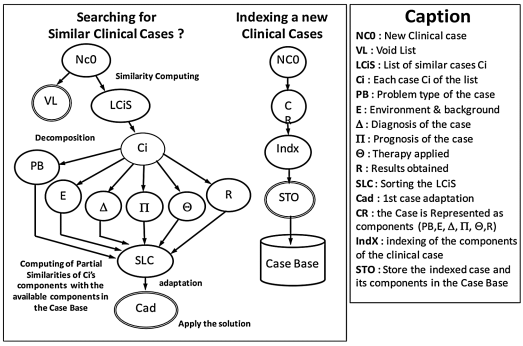


Fig. 3. Case-base reasoning

The supervisor monitors and triggers all necessary steps of the clinical decision process through finite state automaton (FSA) and ensures the dialogue between the computer and the end-user. It controls the management and execution of clinical tasks with predefined available models. A model is instructed to give up a task if the situation or the operating environment is amended.

In the DSS, each model uses reflexive knowledge to determine whether it should contribute to the task requested by the supervisor. If the model determines that it should contribute, then the supervisor assigns its part of the on-going actions, and the necessary models become active. Answers are extracted from the DSS repository and the CBR database and listed for query answering through a matching approach by consulting the indexed corpus.

3.3 Ontology Utilization

In the DO, each ontology triple is established based on an inference rule. The DO triple is extracted and utilized in this study to clarify the medical knowledge and improve the reasoning ability of the DSS. For example, the triple (radical surgery, cure, cancer) refers to the possibility that radical surgery can help to cure cancer. By means of inheritance and matching, the cancer-relevant knowledge contains the following information.

1. Symptoms of each type of cancer.
2. Classification, biological features and cell proliferation dynamics of cancer.
3. Cancer therapies: surgical treatment (radical surgery, palliative surgery, surgical exploration), radiotherapy, chemotherapy (systemic chemotherapy, adjuvant chemotherapy, neoadjuvant chemotherapy, special approach chemotherapy for advanced stage or disseminated tumors) and food therapy.
4. Basic therapeutic principles for tumors, including therapeutic regimens for head and neck neoplasms, thoracic neoplasms, abdominal malignant tumors, tumors of the urinary system, tumors of the female genital system, CNS tumors, malignant tumors of the hematopoietic system, soft tissue tumors, primary malignant bone tumors, and metastatic tumors. Consider breast cancer as an example:
   Phase 0 and I: Breast-conserving conservative surgery + postoperative radical radiotherapy or modified radical mastectomy.
   Early Phase II: Same as Phase 0 and I. Chemotherapy or endocrine therapy will be performed according to pathology and receptor conditions.
   Phase II: Modified radical mastectomy ± radiotherapy ± chemotherapy ± endocrine therapy.
   Phase III: Neoadjuvant chemotherapy ± radiotherapy + modified radical mastectomy (or radical resection) + postoperative radiotherapy + chemotherapy ± endocrine therapy.
   Phase IV: Mainly chemotherapy and endocrine therapy ± local radiotherapy ± local operation.
5. Diagnosis and treatment for cancer pain: Mechanism and classification of cancer pain, evaluation (conventional evaluation principle, quantitative evaluation principle, comprehensive evaluation principle and dynamic evaluation principle) of cancer pain, and therapeutic principles and methods (etiological treatment, drug analgesia therapy and non-drug therapy) for cancer pain.
6. Pharmacological action and pharmacokinetics of drugs.
7. Drug options and doses, dose intensity, relative dose intensity, dose density, course of treatment, intervals, etc.

Since new clinical knowledge cannot be applied automatically without clinical verification, a hierarchy expansion is implemented for incorporating new input data, which should be supported by clinical evidence and supervised by a knowledge engineer.

4 Experiments and Evaluation

4.1 Example: Diagnosis, Prognosis and Treatment of Gastric Cancer

The output interface of the DSS is mainly used to generate query results. Figures 4, 5 and 6 illustrate the diagnosis, prognosis and treatment of gastric cancer. In Fig. 4, SAT Δ concerns gastric cancer diagnosis. Its inference graph refers to the evolution of existing gastric cancer cases, corresponding to the structure of SAT Δ detailed in Fig. 2. In the inference graph, Δ0 indicates a positive diagnosis, Δ1 specifies a differential diagnosis, and Δ2 to Δ6 identify etiological diagnoses. The initial presence of gastric cancer (SO1) can lead to diagnoses Δ1 to Δ6 according to clinical signs SE1 to SE6.

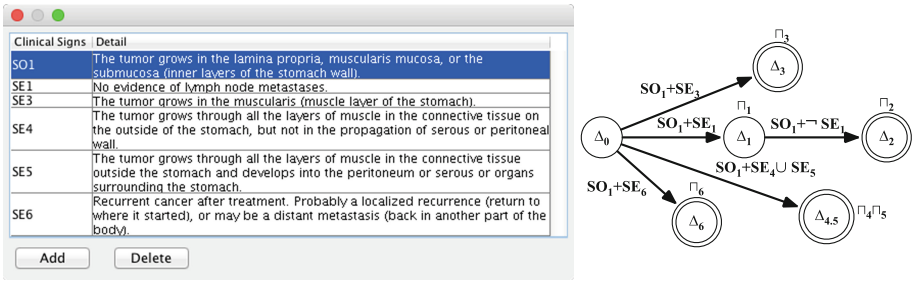


Fig. 4. Clinical signs to determine gastric cancer

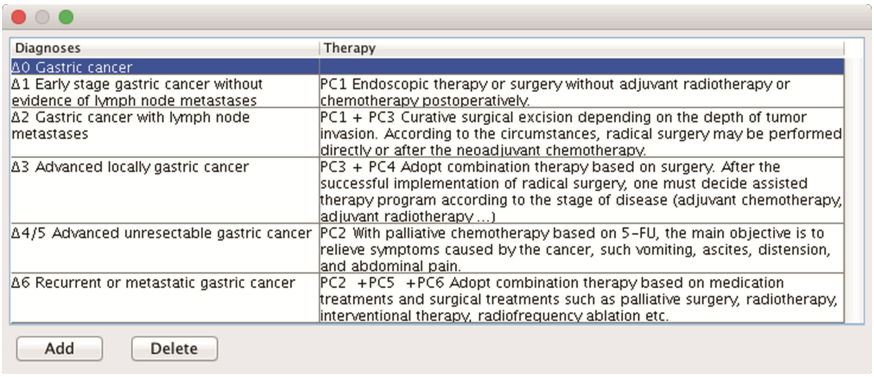


Fig. 5. Diagnoses and therapies for curing gastric cancer

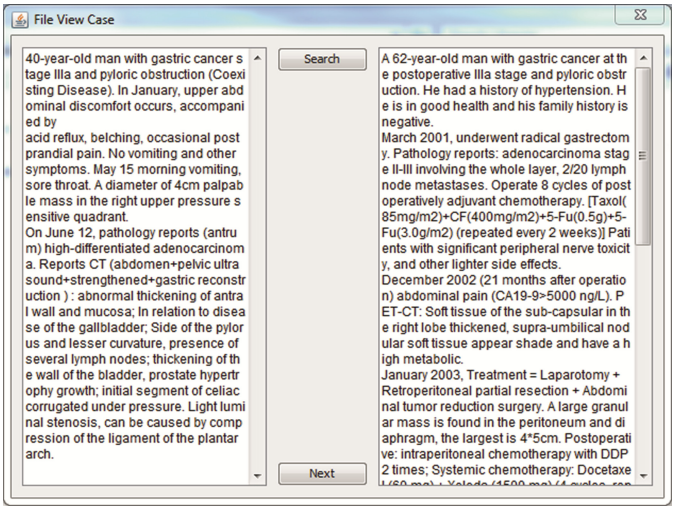


Fig. 6. Extraction of similar clinical cases with CBR

SAT Θ indicates various therapeutic strategies (Θ1 to Θ6) (Fig. 5) corresponding to different diagnoses and prognoses. The knowledge encapsulated in these several steps is stored in the DSS and can be extracted by CBR.

Figure 6 shows the similar clinical cases extracted by CBR. After the patient's condition is input into the DSS, the CBR base of object cases serves as a corpus to search for optimal treatments for a 40-year-old man with gastric cancer at the postoperative stage IIIa and pyloric obstruction. After separating the new clinical case and the corresponding health record, CBR computes the similarities of related components by exploring the indexed knowledge. Matching approach is launched to carry out knowledge representation and similar clinical cases selection. CBR seeks to identify clinical cases with important words such as terminologies (e.g. syndrome: acid reflux, belching, vomiting…) and corresponding synonyms, as well as adjectives (e.g. upper abdominal discomfort, occasional postprandial pain…) and verbs (occur, accompany, cause…) that make phrase more accurate.

In Fig. 6, with the information of current patient such as Δ (gastric cancer stage IIIa) and PB (problem of current case: high-differentiated adenocarcinoma, pyloric obstruction, palpable mass, abnormal thickening of antral wall and mucosa…), DSS works with CBR to search and provide a list of clinical treatment suggestions to patients and physicians.

Concerning the case of Fig. 6 top panel (clinical case in need of help), one of the optimal clinical treatment suggestions (Fig. 6 bottom panel) is detailed in this article with key elements:

– 1st round Θ: underwent radical gastrectomy;
  2nd round Θ: laparotomy + retroperitoneal partial resection + abdominal tumor reduction surgery;
  3rd round Θ: chemotherapy: CPT-11 (120 mg) + 5-Fu (300 mg);

4th round Θ: nodular partial excision + side-by-side ascending-ilea colon.
– 1st round SΘ: postoperatively adjuvant chemotherapy;
2nd round SΘ: intraperitoneal chemotherapy with DDP 2;
4th round SΘ: chemotherapy with DDP + 5-FU.

And R (result: death; survival time: 4 years 11 months), as well as some additional information like date and duration. Other extracted similar cases are not detailed for the sake of brevity. Physicians can therefore make clinical decisions by referencing to similar cases provided by DSS.

4.2 ROC Graphs for the Performance Evaluation

Based on the positive and negative cases of gastric cancer, we compare the classification effectiveness of the DSS with and without ontology.

The 5-fold cross-validation method was used to randomly divide the experimental dataset into five sub-datasets of the same size. Each was used separately as the test set and the other four as the training set. These five experiments were repeated using the DSS with and without ontology, and the average precision of the classification results was calculated. In a binary classification, a sample is judged to belong to a certain class when its posterior probability of belonging to that class is greater than 0.5. The threshold was therefore set to 0.5 for the precision calculation. The results are presented in Tables 1 and 2. As shown in these two tables, the average precision of the results increased by 11% (from 73.68% to 84.63%).

Table 1. Experimental results on DSS without ontology

No. of test samples	TP	FP	TN	FN	FP/(FP + TN)	TP/(TP + FN)	Accuracy
78	39	17	20	2	0.46	0.95	75.64%
77	38	17	18	3	0.48	0.93	72.72%
77	37	15	19	6	0.44	0.86	72.72%
76	37	16	18	5	0.47	0.88	72.36%
76	37	15	20	4	0.43	0.90	75.00%
Average accuracy	73.68%						

Table 2. Experimental results on DSS with ontology

No. of test samples	TP	FP	TN	FN	FP/(FP + TN)	TP/(TP + FN)	Accuracy
78	43	10	23	2	0.30	0.95	84.61%
77	43	7	24	3	0.22	0.94	87.01%
77	42	11	22	2	0.33	0.95	83.11%
76	44	8	21	3	0.27	0.94	85.52%
76	41	12	22	1	0.35	0.97	82.89%
Average accuracy	84.63%						

On the basis of the above experimental results, ROC curves were used to evaluate the classification performance of the DSS with and without ontology (Fig. 7). The red

and black curves represent the classification performance of DSS with and without ontology, respectively. The diagonal navy gray line shows a random model. The curves climb quickly upward initially, indicating that the model correctly predicted the cases. Given the same test dataset, the classification performance of the DSS with ontology is significantly better than that of the DSS without ontology. Moreover, the AUC of the DSS with ontology is 0.846, which means that in 84.6% of cases, a randomly selected case from the group where the target equals 1 has a higher score than that of a randomly chosen case from the group where the target equals 0. This approach clearly outperforms the DSS without ontology, which achieved an AUC of only 73%.

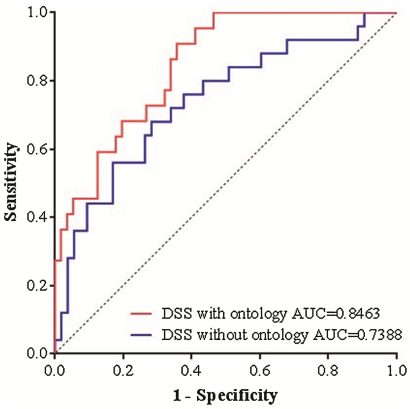


Fig. 7. ROC chart and AUC for classifier evaluations (Color figure online)

5 Conclusion

This study has developed a DSS for diagnosing and treating cancers. To strengthen the knowledge and interlinking of data, our system is based on the reuse of an existing disease ontology, which improves the reasoning ability of the DSS. The patient's condition is analyzed to estimate the stage of the cancer. Based on previous cases indexed in the CBR database, the search results will be returned to the doctors for use as references in diagnosis.

In our future research, we will adopt the ontology enrichment method to reuse other existing biomedical ontologies, leading to a large domain ontology. In this case, several modules can be added to the DSS. For example, the treatment of cancer patients is conducted in cycles; therefore, a warning function shall be added to the system to remind the medical staff to carry out a new cycle of treatment. When a therapeutic regimen is given by the system according the patient's conditions, an introduction (including pharmaceuticals, price and major efficacy) to the recommended medications [19] will be presented below the window to help doctors use drugs rationally according to the patient's financial situation.

Acknowledgement. This work was financially supported by the National Natural Science Foundation of China (No. 61602013), and the Shenzhen Key Fundamental Research Projects (Grant No. JCYJ20160330095313861, and JCYJ20151030154330711).

References

1. Musen, M.A., Middleton, B., Greenes, R.A.: Clinical decision-support systems. In: Shortliffe, E., Cimino, J. (eds.) Biomedical Informatics, pp. 643–674. Springer, London (2014). https://doi.org/10.1007/978-1-4471-4474-8_22
2. Holstiege, J., Mathes, T.: Effects of computer-aided clinical decision support systems in improving antibiotic prescribing by primary care providers: a systematic review. J. Am. Med. Inform. Assoc. **22**(1), 236–242 (2014)
3. Shen, Y., Colloc, J., Jacquet-Andrieu, A., Lei, K.: Emerging medical informatics with case-based reasoning for aiding clinical decision in multi-agent system. J. Biomed. Inform. **56**, 307–317 (2015)
4. Schriml, L.M., Arze, C., Nadendla, S., Chang, Y.W.W., Mazaitis, M., Felix, V., Kibbe, W.A.: Disease ontology: a backbone for disease semantic integration. Nucleic Acids Res. **40**(D1), 940–946 (2011)
5. Federhen, S.: The NCBI taxonomy database. Nucleic Acids Res. **40**(1), 136–143 (2011)
6. Köhler, S., Doelken, S.C., Mungall, C.J., Bauer, S., Firth, H.V.: The human phenotype ontology project: linking molecular biology and disease through phenotype data. Nucleic Acids Res. **42**(D1), 966–974 (2013)
7. Wishart, D.S., Knox, C., Guo, A.C., Shrivastava, S., Hassanali, M., Stothard, P., Chang, Z., Woolsey, J.: DrugBank: a comprehensive resource for in silico drug discovery and exploration. Nucleic acids Res. **34**(suppl_1), 668–672 (2006)
8. Bodenreider, O.: The unified medical language system (UMLS): integrating biomedical terminology. Nucleic acids Res. **32**(suppl_1), 267–270 (2004)
9. Medicine Wikipedia: Wikipedia (2017). https://en.wikipedia.org/wiki/Medicine. Accessed 10 Oct 2017
10. Farion, K., Michalowski, W., Wilk, S., O'sullivan, D., Rubin, S., Weiss, D.: Clinical decision support system for point of care use–ontology-driven design and software implementation. Methods Inf. Med. **48**(4), 381–390 (2009)
11. Haghighi, P.D., Burstein, F., Zaslavsky, A., Arbon, P.: Development and evaluation of ontology for intelligent decision support in medical emergency management for mass gatherings. Decis. Support Syst. **54**(2), 1192–1204 (2013)
12. Lee, C.S., Wang, M.H.: A fuzzy expert system for diabetes decision support application. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **41**(1), 139–153 (2011)
13. Jayaraman, S., Tao, L., Gai, K., Jiang, N.: Drug side effects data representation and full spectrum inferencing using knowledge graphs in intelligent telehealth. In: 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing, pp. 289–294 (2016)
14. Gai, K., Qiu, M., Chen, L.C., Liu, M.: Electronic health record error prevention approach using ontology in big data. In: 2015 IEEE 17th International Symposium on High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium Cyberspace Safety and Security (CSS), and 2015 IEEE 12th International Conference on Embedded Software and Systems (ICESS), pp. 752–757 (2015)
15. Li, Y., Gai, K., Ming, Z., Zhao, H., Qiu, M.: Intercrossed access controls for secure financial services on multimedia big data in cloud systems. ACM Trans. Multimed. Comput. Commun. Appl. (TOMM) **12**, 67 (2016)

16. Elnagdy, S.A., Qiu, M., Gai, K.: Cyber incident classifications using ontology-based knowledge representation for cybersecurity insurance in financial industry. In: 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing, pp. 301–306 (2016)
17. Efron, B.: Bayes' theorem in the 21st century. Science **340**(6137), 1177–1178 (2013)
18. Kolaczyk, E.D., Csárdi, G.: Statistical Analysis of Network Data with R. Use R! 65. Springer, New York (2014). https://doi.org/10.1007/978-1-4939-0983-4
19. Yan, W., Zanni-Merk, C., Rousselot, F.: Ontology matching for facilitating inventive design based on semantic similarity and case-based reasoning. Int. J. Knowl. Based Intell. Eng. Syst. **17**(3), 243–256 (2013)

Using Virtualization for Blockchain Testing

Chen Chen, Zhuyun Qi, Yirui Liu, and Kai Lei(✉)

Shenzhen Key Lab for Cloud Computing Technology and Applications,
School of Electronic and Computer Engineering (SECE), Institute of Big Data Technologies,
Peking University, Shenzhen 518055, People's Republic of China
tocc@sz.pku.edu.cn, {qizy,leik}@pkusz.edu.cn, 993518823@qq.com

Abstract. Blockchain technology is experiencing prosperity. A wide variety of open source blockchains emerge recent years, which are different in architecture design or protocol parameters. When a developer wants to compare runtime performance of different blockchains, he could conduct a simulation with event simulator, or run application on physical machines. Either way will result in problems of unconvincing or costly. Container, a lightweight virtualization technique, is suitable for solving this dilemma. Tens of containers could run simultaneously in a single-core CPU computer, with each container having a running blockchain client. A simulated blockchain network with hundreds of nodes could be easily established in this way. In this paper, we firstly overview blockchain architecture choices which will significantly affect performance. Then we introduce our framework on testing blockchains using containerization. Authenticity and high cost of P2P application testing would be balanced in this framework. Finally, we implement our framework to run a demo testing how bitcoin's network parameters will affect system reliability.

Keywords: Blockchain testing · P2P testing · Containers

1 Introduction

As blockchain technology blooming, various organizations have released codebases on blockchain system. Communities like Bitcoin and Ethereum have rapid product iteration. The Hyperledger project, hosted by The Linux Foundation, is developing several frameworks simultaneously. Such like Fabric, Sawtooth, Iroha and so on. Institution like Ripple Lab released their blockchain system as a solution of foreign exchange settlement. More than that, traditional enterprises are also excited about blockchain. Many of them form coalitions like R3, which make great efforts on their framework Corda.

These frameworks are designed for different scenarios. It makes their architecture emphasizing on different aspects and having a unique tradeoff between security and performance. An important usage scenario of blockchain is crypto currency. Bitcoin is a pioneer is this field. It realized a decentralized digital currency circulation mechanism which participants are not required to trust any third party. Participants only have to trust that economic incentive and algorithm will make this decentralized system act in honest, and guarantee value of their digital currency. Systems like Ethereum go further than

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 289–299, 2018.
https://doi.org/10.1007/978-3-319-73830-7_29

crypto currency. Not only does it allow crypto currency circulation, but it can also execute any decentralized application, which is known as smart contract. Bitcoin and Ethereum are permissionless blockchain, which means anyone can join or leave - act honestly or dishonestly - as they want. So the primary problem of permissionless blockchains is fault tolerance. Permissionless blockchain should tolerate not only crash fault when nodes are crashed, but also Byzantine fault [1] which nodes have malicious behavior.

Another application area of blockchain technology is in business. When several companies set up a consortium to cooperate with others, they have demand for a global ledger to share data. Because of the independent interest of each company, it would be best if this ledger is maintained by every participant. Framework for this scenario is called permissioned blockchain since participants can be identified in the system. Malicious behavior in such system can be recorded and punished in reality, so nodes have weak trust between each other.

Primary problem of business blockchain framework is performance improvement. Blockchain is costly in computation and networking comparing to traditional database. Performance is the bottleneck when deploy blockchain in real world business.

From the perspective of blockchain developers, they have to select a codebase which best fits their business scenes and performance requirements. Considering blockchain testing, usually we have two ways: (1) Implement blockchain's protocol logic in an event simulator, then conduct a simulation. (2) Deploy blockchain application on physical machines.

One example [2] of the simulation method is tuning bitcoin protocol's parameter with ns-3, a discrete-event network simulator. Testing with simulator has to abstract protocol reaction of different event. It requires extra work to reimplement protocol logic, and more importantly, it is not testing a real application. Testing with physical machines [3] could be costly. It can only deploy application in a small scale scene such as private blockchain. The cost would be extremely high if testing with thousands of nodes. The testing authenticity with physical machines and low cost with simulator could be blended if we introduce virtualization into blockchain testing. We can simulate several logical nodes in a single CPU, and all logical nodes are running real application simultaneously.

The remaining part of this paper is structured as follows. In Sect. 2, we overview blockchain architecture choices which will significantly affect performance. In Sect. 3, we introduce technical background of our testing. In Sect. 4, we describe conception and workflow of our framework. We conduct a demo testing how bitcoin's network parameters will affect system reliability in Sect. 5, and conclude this paper in Sect. 6.

2 Variety of Blockchain Architecture

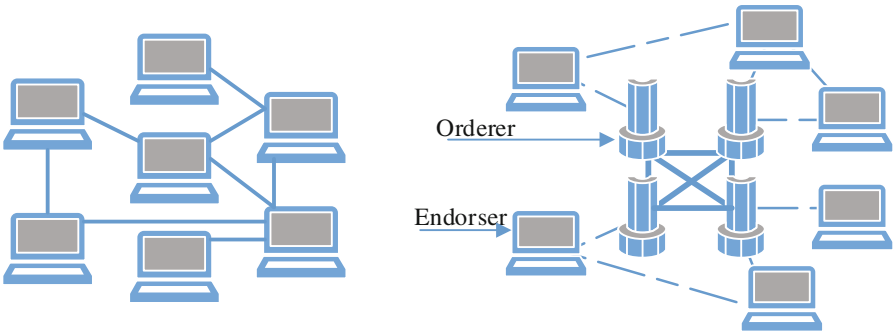
There are a wild variety of blockchain projects. In this section, we summarize blockchain architecture choices which will significantly affect performance, and we have a brief comparison over them.

2.1 System Architecture

System architecture should be selected according to the business model and transaction process. Mainstream blockchain projects take either fully decentralized P2P architecture or collaborative architecture which nodes are interdependent.

P2P architecture. In P2P architecture, every single node in the network is the full node with all features needed and complete all work independently. A participant has the weakest trust in others. It maintains a complete replica of global ledger and checks every transaction inside. Redundant storage and complex computing result in low throughput of the system. This architecture fits decentralized crypto currencies well which little trust between nodes.

Collaborative architecture. The throughput of P2P architecture is too low that it can hardly meet the requirements of large scale business. Throughput can be enhanced if decouple full node functions into several parts. Each node has part of functions and work collaborative in order to decrease redundancy in computation or storage. One example is decoupling transaction verifying and sorting. Figure 1b describes the architecture of Fabric [4]. Fabric decouples functions into endorsement service and ordering service. Endorser nodes will endorse transactions which verified by them. If a transaction gets enough endorsement, it will send to orderer nodes to decide the order with other transactions. Collaborative architecture is suitable for permissioned blockchain in large scale business scene.



(a) P2P architecture (b) Collaborative architecture

Fig. 1. Blockchain system architecture comparison

2.2 Consensus Algorithm

Blockchain is a distributed system. Nodes should reach a consensus of global ledger states. And system has to tolerate some nodes behaving maliciously in the meantime. Many kinds of consensus algorithm have been proposed to adapt different consistency requirements.

BFT Consensus. Byzantine fault tolerance (BFT) consensus can tolerate failed nodes generating arbitrary data and pretending to be the correct ones. Traditional BFT algorithms, like PBFT, require nodes communicate with each other for several rounds. It can only fit a small scale consensus due to its communication bound. But PBFT is a strong consistency algorithm reaching consensus any point in time. Fabric applies PBFT algorithm between orderer nodes by default.

Probabilistic based consensus. Probabilistic based consensus like proof of work (POW) [5] can solve the scalability problem of traditional BFT. Participants compete with each other to solve a difficult computation problem. Possibility to win the competition for a node is based on probabilistic, and the winner has the right to commit transactions into ledger. POW consensus is high scalability since difficulty of computation problem will adjust over time. Nevertheless, several nodes could solve computation problem almost at the same time, which means having a fork, so it can only achieve eventual consistency. Also, POW is computation bound and waste lots of energy.

Non-BFT consensus. Tolerance Byzantine fault in system could be costly due to communication bound or computation bound in consensus algorithm. In some permissioned blockchain scenes, nodes have strong trust with others. It can apply non-Byzantine fault consensus like Three-phase commit. Which will save computation power and network bandwidth although it can't defend nodes with malicious.

2.3 Protocol Parameters

As distributed applications, blockchain systems have many parameters need to be set. For instance Bitcoin has multiple variants in different parameterization. Block interval time in Bitcoin is 10 min, while Litecoin is 2.5 min, Dogecoin is 1 min. Block size is limited at 1 MB in Bitcoin, while proposals like BIP-101 (Bitcoin Improvement Proposal) suggests to increase block size to 8 MB as first step, and BIP-102 suggests increase to 2 MB.

Protocol parameters of blockchain system should be set according to business requirement, network condition and system scale. Tuning parameters could be difficult because the relationships between parameters are complicated. And there is not a convincing mathematical model to judge which blue print is the best.

3 Technical Background

In this section, we introduce some technologies used in our testing framework. Firstly we give a brief introduction to containerization. Then we present Docker platform used in our experiments, and discuss a detail of process scheduler which is significant component in testing framework.

3.1 Containerization

Containerization, also known as operating system level virtualization, is a light weight virtualization technique. It uses functions support by kernel to isolate user-space instances. Each instance, or called container, has its independent resources like CPU power, storage and network. A program in a container seems running in a unique real computer. However containers can share same operation system and appropriate libraries, which make container launches fast and saves storage.

3.2 Docker

Docker is a platform helping developers automatically assemble their containers from source code, and deploy containers across machines. It was built on several low-level kernel features, such as namespaces and control groups in Linux.

When containers are running in a host, they have to compete for CPU power with each other. Docker uses default Linux kernel scheduler called Completely Fair Scheduler (CFS) to allocate resource for executing process. Process scheduler is important in our framework and must be properly tuned. It brings additional context switching overhead if CPU time period is too short. But it results in containers froze for a long time if CPU time period is too long.

Two CFS parameters can be set in Docker. One is period of CPUs, the other is CPU quota of container. Both parameters should be set between 1 ms and 1 s. We conduct three experiments to explore differences between parameters. Each experiment has 10 containers keep outputting container ID and current Unix timestamp accurate to nanoseconds. Each container shares 10% of CPU power. CPU period is set from 10 ms to 1 s.

If a CPU slice is 1 ms and we have 10 containers, in every 10 ms each container is supposed to execute for 1 ms and suspend for 9 ms. In order to figure out the mechanism for CFS, we do a simple experiment to analyze CPU slides size.

In our 2.5 GHz single core CPU computer, it will take nearly 0.7 ms to print a system time log in nanosecond. We let 10 containers compete to print system time log. If a container is continuous taking CPU resource, its container ID will appear consecutive in log. We could figure out the size of CPU time slices of different configuration.

Figure 2 shows that context switch would be frequently if CPU quota set at 1 ms. But there are little differences between 10 ms CPU quota and 100 ms. So we prefer set CPU quota at 10 ms in our experiments.

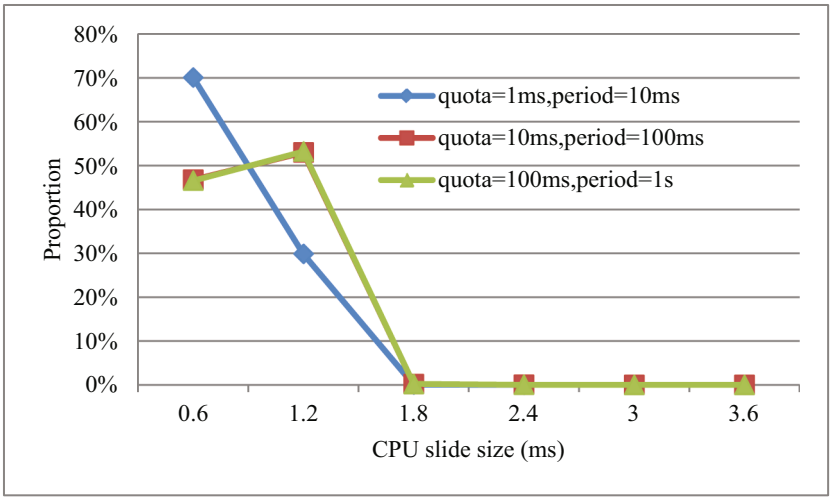


Fig. 2. CPU slide size of different CFS configurations.

4 Testing Framework

There is variety of blockchain architecture. It would be helpful if we could test blockchains in an inexpensive way. We propose our framework on blockchain testing with virtualization.

4.1 Conception of Framework

Figure 3 demonstrates architecture of our framework. In single host mode, containers in the host machine can be connected by a virtual network device like bridge container. In multi host mode, physical machines can be connected by Ethernet, and we could set multiple virtual network devices to imitate a large scale P2P network. Ideally, a physical machine could simulate tens of logical nodes running real P2P application, and several router nodes simulating routing process. We can abstract a real-world network in this way, with all virtual nodes running real applications.

There are various popular virtualization technologies, such as (1) Hypervisor Virtualization and (2) Containerization. We prefer using container as our virtualization basis. Containerization is a low cost virtualization technology which allows more virtual nodes in a physical machine.

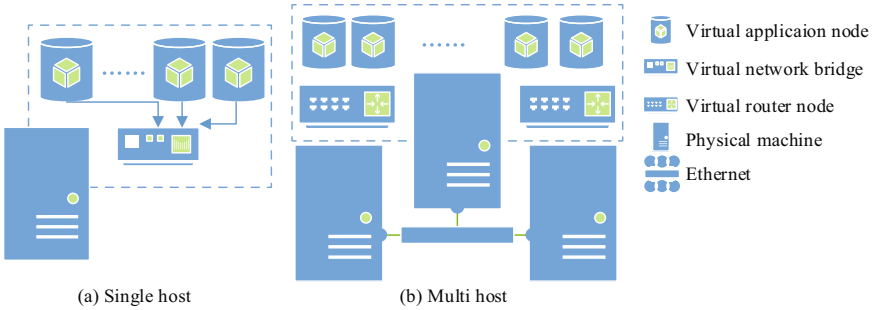


Fig. 3. Testing framework

4.2 Testing Flow

Docker image of blockchain application should be built first. It would be better if configurations are given by runtime parameters, so we only have to build image once. Our testing flow is described at Fig. 4. First we give configuration of experiment, and then containers run and build connection with others. We have to check if the blockchain system has stabilized before conduct observations. For instance if we study on performance of POW blockchains, at least two preconditions must be met: (1) all containers are connected; (2) mining rate is steady in a period of time. After that we think the system is running stable, then we can start system status recording or fault injection.

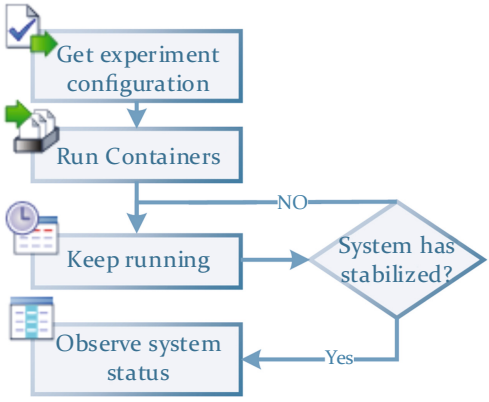


Fig. 4. Testing flow

5 Demo Experiment

5.1 Architecture of Demo

We intend to figure out three questions about small-scale bitcoin system in our demo:

- How bitcoin protocol's parameters, like block interval, affect system reliability?
- How network conditions, like network latency, affect system reliability?
- Will system reliability be affected if the number of nodes increased?

We use Docker platform to deploy our customized bitcoin application into containers, which act as real bitcoin miners. All of bitcoin containers are connected by a bridge container, which is a virtualized ethernet bridge device.

Our demo runs on a cheap cloud server, which has a single 2.5 GHz CPU with 2 GB RAM. We launch tens of bitcoin containers in this host machine. Each container is limited with 20 MB memory without swap, and share same proportion of host machine's CPU cycles.

The first bitcoin container launched will be set as a seed node in this bridge network. All containers launching later will connect to seed node. As implemented in default bitcoin application, each miner will initiate maximum 8 outbound connections to remote nodes, and accept hundreds of inbound connection made by others. Message can flow in both directions with any peer. After each miner has numerous connections with others, we think this small-scale P2P network has been established.

5.2 Experiments

We modified bitcoin source code [6] of version 0.3.24 to conduct our demo. Our evaluation of system reliability is stale block rate. A block being abandoned after a blockchain fork occurred named stale block. When a fork occurs, there are more than two different blocks valid at the same time. Participants don't know which block will win out or which blocks will be discarded later. The system becomes unavailable at this time. Stale block rate demonstrates if we find a new block, how likely this block is going to be a stale block. A higher stale block rate means more unavailable time of the system.

$$stale\ block\ rate = \frac{total\ blocks\ generated - hight\ increased\ of\ highest\ chain}{total\ blocks\ generated} \tag{1}$$

Three experiments groups have been made to study on different topics.

Number of nodes. In Fig. 5, we found stale block rate positive correlation with number of nodes. In POW blockchain, any miner node has probability to find a new block. It takes more time for message propagate throughout the network if number of nodes increased, therefore possibility that two miners respectively generate a block increased.

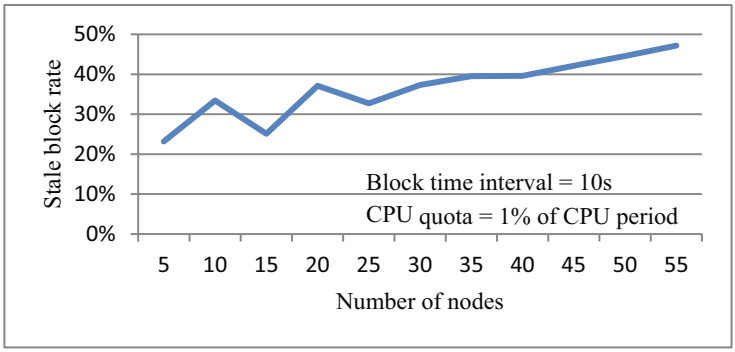


Fig. 5. Stale block rate on number of nodes

Block time interval. As in Fig. 6, stale block rate is negative correlation with block time interval. If block time interval is far greater than message propagation delay, stale block rate will close to 0 because message has enough time to transfer before someone find a new block.

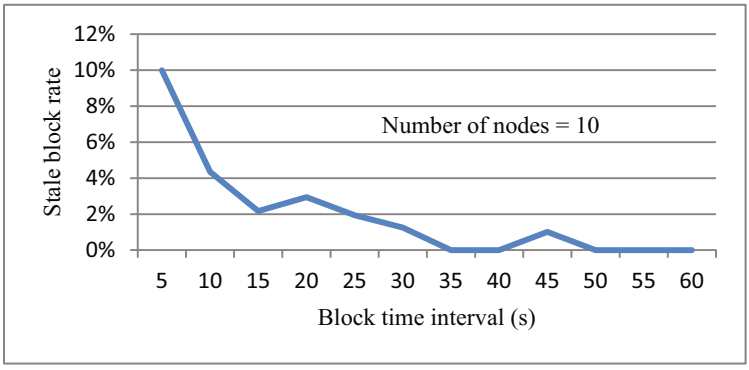


Fig. 6. Stale block rate on block time interval

Network delay. Figure 7 shows that stale block rate is positive correlation with network delay. Messages take more time to propagate if network delay increased, and every node is more likely to generate its new block respectively.

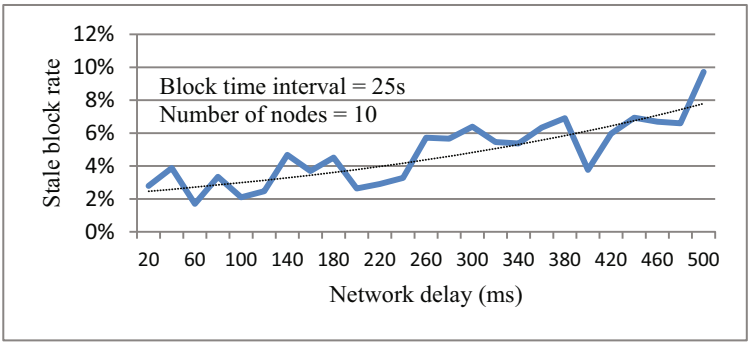


Fig. 7. Stale block rate on network delay

6 Conclusion and Future Work

We have found these advantages of using containerization for blockchain testing while building this demo.

- Ease of deployment. With the help of container platform like Docker, we can build once and run anywhere.
- Low cost. It's easy to virtualize tens of containers in a single core CPU.
- Authenticity. All containers are running an integral application.

However, execution delay is inevitable. Only one program can be executed at any point of time. Containers must line up to execute their application discretely. We have tried running hundreds of containers in a single CPU. However execution delay is too long that some nodes cannot establish a TCP connection with any other node. We think limitations of virtualization in software testing mainly lie in execution delay.

We will carry out more in-depth study on mathematical model of blockchain system, and blockchain performance in a complex future network environment, such like congested Named Data Networking scene [7].

Acknowledgement. This work has been financially supported by Shenzhen Key Fundamental Research Projects (Grant No. JCYJ20170412151008290, JCYJ20170306091556329 and JCYJ20170412150946024).

References

1. Castro, M., Liskov, B.: Practical Byzantine fault tolerance. In: OSDI, vol. 99, pp. 173–186 (1999)
2. Gervais, A., Karame, G.O., Wüst, K., et al.: On the security and performance of proof of work blockchains. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 3–16. ACM 2016

3. Dinh, T.T.A., Wang, J., Chen, G., et al.: BLOCKBENCH: a framework for analyzing private blockchains. In: Proceedings of the 2017 ACM International Conference on Management of Data, pp. 1085–1100. ACM (2017)
4. Fabric Documentation. https://hyperledger-fabric.readthedocs.io. Accessed 10 Nov 2017
5. Vukolić, M.: The quest for scalable blockchain fabric: proof-of-work vs. BFT replication. In: Camenisch, J., Kesdoğan, D. (eds.) iNetSec 2015. LNCS, vol. 9591, pp. 112–125. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39028-4_9
6. Our Modified Bitcoin Client. https://github.com/ccen/bitcoinL. Accessed 10 Nov 2017
7. Lei, K., Hou, C., Li, L., et al.: A rcp-based congestion control protocol in named data networking. In: 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 538–541. IEEE (2015)

DiPot: A Distributed Industrial Honeypot System

Jianhong Cao[1], Wei Li[1], Jianjun Li[1], and Bo Li[2(✉)]

[1] China Tobacco Zhejiang Industrial Co., Ltd., Zhejiang, China
{caojh,lwei,lijj}@zjtobacco.com
[2] School of Computer Science and Engineering, Beihang University, Beijing, China
libo@act.buaa.edu.cn

Abstract. Recent years witness the prosperous of Internet and Cyber Physical Systems (CPS). More and more industrial devices and systems are connected to the Internet and thus become the target for attackers. This paper proposed a distributed industrial honeypot system called DiPot to monitor Internet scanning and attacking behaviors against industrial control systems. DiPot offers attack clustering and visualization services to users and could help users to be aware of current ICS security situation. Different from existing Honeypot systems, DiPot has two advantages: high-degree simulation and deep data analysis. DiPot is also equipped with an advanced visualization frontend and could provide users with good experience. Through 6 months running, DiPot has obtained plenty of data and captured some real-world attack samples from Internet. The experimental results demonstrate the effectiveness and efficiency of DiPot.

Keywords: Industrial control system · Honeypot · Distributed

1 Introduction

From Stuxnet, the security of industrial control systems (ICS) has gained the attention from both academy and industry. On the other side, ICS also attracts the attention from attackers since ICS is much vulnerable to and could be easily accessed from the Internet. Traditional industrial control systems were designed for close environments in which industrial devices and systems are connected using Local Area Network (LAN) and cannot accessed be from outside world. However, with the development of Internet and CPS, more and more industrial control devices are connected to the Internet for remote control and monitoring. Besides, the security of industrial protocols is seldom considered. Facing Internet threats against ICS devices and systems, researchers put great emphasis on protecting ICS infrastructures and preventing them from being subverted by attackers [1–5]. On the other hand, it is more effective to study the attack patterns and motivations from the view of attackers. And to study the attackers requires the researchers to understand the attacker's behavior.

Honeypot is a very effective approach to study the behaviors of attackers. By disguising as normal machines or devices, Honeypot could record the behaviors of attackers and scanners, and then for further analysis. There are some open-source honeypot systems [6–8, 10, 11] which are popular among researchers. Recently, Conpot,

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 300–309, 2018.
https://doi.org/10.1007/978-3-319-73830-7_30

an industrial honeypot, has been proposed and widely used to analyze the ICS attack behaviors and patterns. However, Conpot has some limitations.

Firstly, Conpot is a low-interaction honeypot, which means that it could only accept connection passively and reply with simple response messages. Therefore, Conpot could be easily identified by attackers [9]. Secondly, Conpot is designed to only collect access traces from outsiders and therefore lack of data analysis capabilities. Third and last, Conpot is hard to use and requires considerably more efforts for users to learn.

To address the above issues, we present a distributed honeypot system called DiPot to capture and analyze the suspicious behaviors against industrial control systems from Internet. DiPot outperforms existing ICS honeypots in two aspects: (1) DiPot achieves high-fidelity simulation of ICS protocols and devices, therefore is hard to be identified by scanners and attackers. (2) DiPot supports deep data analysis on the captured suspicious behaviors and leverages clustering approaches to further differentiate attackers and infer the motivation behind. Until now, DiPot has been successfully running online for more than 6 months and has captured various access and attack sequences against ICS devices and systems.

The following paper is organized as follows. We describe the related work first, and then present the design and implementation details of DiPot. Then we demonstrate our results and evaluate the effectiveness and efficiency of DiPot. Finally, we close with the conclusion.

2 Related Work

Honeyd [6] is an open source virtual honeypot framework which introduced the simulation of various types of system in the network protocol stack layers. It can simulate many OS fingerprint, thereby could cheat network scanning tools such as NMap/ZMap. However, Honeyd is no longer applicable to industrial honeypot today, and it has inherent defects: (1) the port simulation of Honeyd is too simple and cannot be used for simulating ICS devices. (2) Unlike traditional TCP three-handshake mechanism, Honeyd uses two-stage handshake and could be easily identified by attackers [6]. Honeynet Project [7] follows the idea of Honeyd, but makes improvement and constructs a network of honeypots. Compared with single honeypot node, Honeynet is more stable and could collect more data than Honeyd. Besides, Honeynet also provides the ability to observe the attacker's motivation, methods, tactics and follow-up behaviors. However, the aim of Honeynet is not for industrial control systems, and cannot be used as an industrial honeypot directly. A similar system to our work is Conpot. Conpot [8] adds industrial control protocols based on Honeynet. The advantage of Conpot is the default support for industrial equipment simulation. Users can quickly deploy an ICS honeypot through simple configuration and disguised as PLCs or DCS waiting for the attackers to connect. However, Conpot is a low-interaction honeypot, which means that it could only accept connection passively and reply with simple response messages. Therefore, Conpot could be easily identified by attackers. In additional, the user experience of Conpot is not so good since it could only record the attack behaviors and lack of the analysis and visualization of collected data.

3 Design and Implementation

In this section, we introduce the design and implementation of DiPot. We will first present the design goal of DiPot, and then present the design and implementation.

3.1 Design Goals

DiPot aims at capturing and recording suspicious and malicious behaviors to industrial control systems. Unlike existing ICS honeypots such as Conpot, DiPot puts emphasis on the coverage and diversity for ICS devices. Besides, DiPot should have the deep-analysis ability such as attack behavior classification and attacker fingerprint identification. Since DiPot is an industrial honeypot system, the key design goals are collecting more useful data without being identified by attacks or scanners.

The detailed design goals are summarized as follows:

Effectiveness. DiPot should cover basic network protocols such as TCP, HTTP, FTP. And should cover most open industrial protocol such as *Modbus, S7comm, BACnet, SNMP, ipmi, Guardian_Ast and Kamstrup*.

High-fidelity. DiPot should behavior like a real ICS devices without being identified by attackers or scanners. In this paper, we consider four aspects: software information, response time, physical fingerprint and interaction depth.

Not perturb the proper functioning of the transmission and the overhead should be as little as possible.

Security. DiPot should prevent itself and the hosted environment from being subverted, and protect the logs from being tampered.

Ease of adoption. Our system should be easily deployed and compatible to mainstream platforms.

3.2 Architecture

In this subsection, we will present the overall design of the Dipot. As shown in Fig. 1, DiPot is a distributed system and is composed of three components: Honeypot Node (HN), Data Processing Node (DPN) and Management Node (MN). HNs are deployed in a distributed manner to ensure the collection of diverse attacks from Internet. DPN is responsible for formatting, cleaning, storing and analyzing the data collected by HNs and sends the results in JSON format to MN. MN is in charge of user interaction and data visualization. MN provides Web interfaces and API to interact with users.

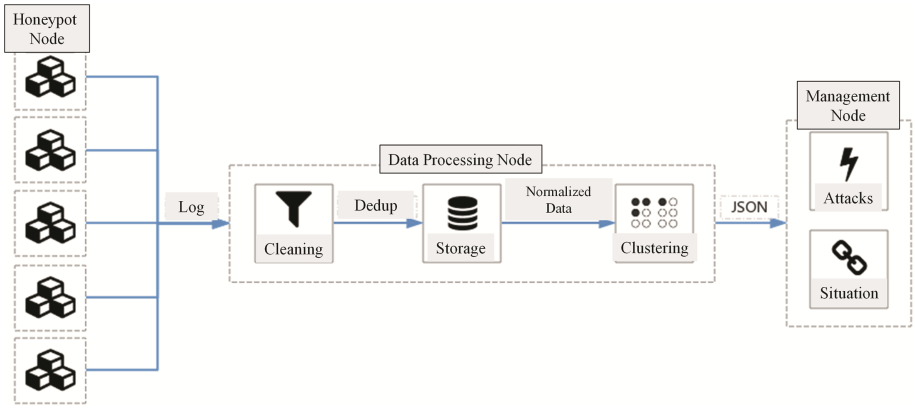


Fig. 1. The architecture of DiPot

(1) ***Honeypot Node (HN)***
HN is implemented based on Conpot, because Conpot has supported several industrial devices and protocols. We extend Conpot to support more devices and protocols, and we also add anti-detection capability in HN. Besides, to record more useful data, we add some triggers, such as logging, formatting etc.; Anti-detection refers to the capability to be identified as honeypot since some ICS search engines such as Shodan have the function to identify Honeypot. We implement our anti-detection function mainly by changing configuration item and response time settings. Our approach successfully "cheats" Shodan that our HN nodes are recognized as real ICS device by Shodan.

(2) ***Data Processing Node (DPN)***
The main function of DPN is to cluster and analyze the information gathered by HNs, and forecast the trend. By analyzing the historical records of attack behaviors, DPN establish attacker behavior models to summarize the attacker behavior patterns. DPN also provide the ability to predict the future behavior of attackers.

Before going deep into the data, some preliminary work is done by DPN such as data cleaning, formatting and saving the log information to the database. Then the preprocessed data is transformed to a sequence of vectors. Each vector contains some feature such as IP, port, protocol, function code. The sequence of vectors is obtained within a certain time interval. Then we use clustering algorithms to classify the vectors into several clusters. Each cluster may represent a type of attacks. The detailed process is described as follows:

a. DPN collects the data from the same IP address within a certain period of time. The data includes both benign network traffic and malicious commands (such as "CPU Control request STOP"). We extract some features from the network flows: such as source IP, destination IP, destination port, timestamp, packet length, malicious command data. The features are combined as a vector to represent the behaviors of network traffic captured by HN.

b. We use k-center clustering algorithm to obtain multiple clusters.

c. Each instance in the same cluster may contain multiple command sequences at the same time. We extracted association rules from the cluster which contains malicious commands. And the extracted rules are further refined as the attack features.

(3) ***Management Node (MN)***
MN is responsible for the display of statistical information, clustering results and manual analysis results. The detailed process of MN is described as follows:

a. Real-time situation display. The page displays the access sequence received by the honeypot and shows the results on the map view in a real-time manner. This page is designed to help users understand the current security situation in an intuitive manner. Besides, some statistics are also show on this page such as attack frequency, the most vulnerable area.

b. Clustering results display. The page shows the clustering results (e.g. IP, protocol), and the access sequence in a time interval are shown in graphs, which is convenient for users to observe the occurrence of attack behaviors.

c. Manual analysis results display. The page shows the specific access sequence analyzed and checked by human. These sequences generally have specific functions, such as locating all slave nodes, and then send control commands to control the honeypot.

3.3 Implementation

(1) ***Honeypot Node (HN)***
We improve Conpot to support fine-grained protocol and device simulation. Take Modbus as an example, we not only support basic information configuration, but also support coil block configuration and analog input and output simulation. In this section, we show the simulation configurations of a Modbus PLC with standard input, coil and registers. The basic configuration of Mobus is described in Table 1.

Table 1. Modbus basic configuration

Configuration name	Value	Description
VendorName	Siemens	Provider name
ProductCode	SIMATIC	Product name
MajorMinorRevision	S7-200	Type
Mode	Serial	Serial port
Delay	100	delay

We set the Modbus slave number to 3. The first two slaves are identical, and their function is to set the coil. Each slave are configured to have 2 blocks. The block configuration is shown in Table 2.

Table 2. Modbus coil block configuration

	Configuration name	Value	Description
Block	Type	COILS	Set the block as the coil
	Starting_address	1	Set the block address to 1
	Size	128	Set the space of block to 128
Block	Type	DISCRETE_INPUTS	Set the block to digital input
	Starting_address	10001	Set the block starting address to 10001
	Size	32	Set the block space to 32

The third slave of Modbus is analog input. The block configuration is shown in Table 3.

(2) ***Data Processing Node (DPN)***

First, DPN periodically pulls the raw log file from HNs. The honeypots log in string format. And the log information differs with different industrial protocols. We observe that handshake packets in industrial control protocols are in a large proportion among the whole network traffic. However, handshake packets contain many duplicated information. Therefore, before doing clustering, we first clean and format the data. For example, Modbus handshake process will trigger HNs four times and records twice. The content of the log record is completely identical. Thus, we only store one data record in the database. We follows the log format of Conpot, however we add exception/error handling mechanism to ensure that only effective information will be saved in the database.

Table 3. Modbus analog input configuration

	Configuration name	Value	Description
Block	Type	ANALOG_INPUTS	Set the block to analog input
	Starting_address	30001	Set block address to 30001
	Size	8	Set block space to 8
Block	Type	HOLDING_REGISTERS	Set the block to holding Reg
	Starting_address	40001	Set block address to 40001
	Size	8	Set the block space to 8

We use kmeans to do log clustering. First we select date, time, source_IP, protocol, function_ID, slave_ID as the features. Clustering analysis is only carried out within the data collected from one HN. Each field is mapped to a value. Date and time combination of the field into the time stamp; source_IP is converted into digital number. According to different categories of the protocols, HTTP is mapped to 0, the rest of the industrial protocols are mapped to 1. Function_ID and slave_ID are the two fields only existing in Modbus, thus we preserve its original values. After the mapping is done, all the fields are normalized to the values between 0 and 1. Finally, we use Euclidean distance as the metric to run kmeans.

$$p = \sum_{n=1}^{k} \sum_{i=1}^{num_i} \sum_{j=1}^{dim} \sqrt{\left(x_{ij} - \overline{x_i}\right)^2}$$

(3) ***Management Node (MN)***

MN is in charge of user interaction and data visualization. As shown in Fig. 2, there are five HNs (A HN is shown as a Yellow dot.) around the world. The HNs are hosted in cloud VMs and distributed to different countries. Each yellow line represents an access sequence to the target HN. The access time, source IP, location etc. are recorded and shown in the bottom part. The left side and right side show the statistics information.

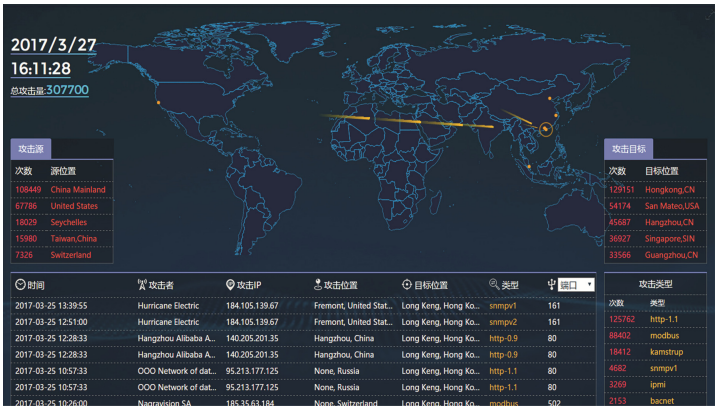


Fig. 2. Data visualization page

Figure 3 shows the clustering results page. Users can choose which HN to show at the upper box. The left Figure presents the protocol distribution of the current HN. The right Figure shows the IP access distribution of the current HN.

Fig. 3. Data clustering page

4 Results

4.1 Statistics Information

DiPot are online running for more than 6 months. We have collect 317,484 access sequence and capture 4,827 suspicious IPs. The protocol distribution is as follows (Table 4):

Table 4. Protocol distribution

Protocol	Access sequence number
HTTP	163698
Modbus	88475
Kamstrup	24545
SNMPv1	4766
ipmi	3296
BACnet	2182
Guardian_Ast	950
SNMPv2	907
Kamstrup manage	130
S7comm	53
Total	289002

4.2 Captured Samples

We present some suspicious samples captured by Dipot.

(1) Modbus Scan
It is a low-risk access sequence. Its function is to obtain the Modbus device and the firmware version information, using the Modbus protocol in 17 (0X11) function code, its function is to read equipment identification code, which is the basic information of the device code. Table 5 shows some scan samples suspected launching by Shodan.

(2) Modbus Over-length Attack
It is a high-risk access sequence with the function of sending packets beyond the Modbus protocol packet length and causing a denial of service. This is achieved by using the weakness of the Modbus protocol exception mechanism, whose command itself does not contain any valid information, but simply uses the super long sequence to cause the Modbus service master node to crash.

(3) Kamstrup Device Scan
This is a low-risk access sequence, its function is to read Kamstrup electric power equipment, the operation directly through the Kamstrup protocol to achieve command by sending request message 0104xx (XX controller number), respectively to Kamstrup power generation equipment and other basic information by scanning and reading, the detailed information of the current state of charge.

Table 5. Modbus scan samples

Request	Slave ID	Fingerprints
0000000000020011	0	a626c8be-375d-42d4-97fb-e0cc3aaa52e3
0000000000020111	1	a626c8be-375d-42d4-97fb-e0cc3aaa52e3
000000000005012b0e0100	1	a626c8be-375d-42d4-97fb-e0cc3aaa52e3
0000000000020211	2	a626c8be-375d-42d4-97fb-e0cc3aaa52e3
000000000005022b0e0100	2	a626c8be-375d-42d4-97fb-e0cc3aaa52e3
…	…	…
000000000002f711	247	a626c8be-375d-42d4-97fb-e0cc3aaa52e3
000000000005f72b0e0100	247	a626c8be-375d-42d4-97fb-e0cc3aaa52e3
000000000002f811	248	a626c8be-375d-42d4-97fb-e0cc3aaa52e3
000000000002f911	249	a626c8be-375d-42d4-97fb-e0cc3aaa52e3
000000000002fa11	250	a626c8be-375d-42d4-97fb-e0cc3aaa52e3
000000000002fb11	251	a626c8be-375d-42d4-97fb-e0cc3aaa52e3
000000000002fc11	252	a626c8be-375d-42d4-97fb-e0cc3aaa52e3
000000000002fd11	253	a626c8be-375d-42d4-97fb-e0cc3aaa52e3
000000000002fe11	254	a626c8be-375d-42d4-97fb-e0cc3aaa52e3
000000000002ff11	255	a626c8be-375d-42d4-97fb-e0cc3aaa52e3

5 Conclusion

This paper proposed a distributed industrial honeypot system called DiPot to monitor Internet scanning and attacking behaviors against industrial control systems. DiPot offers attack clustering and visualization services to users and could help users to be aware of current ICS security situation. Different from existing Honeypot systems, DiPot has two advantages: high-degree simulation and deep data analysis. DiPot is also equipped with an advanced visualization frontend and could provide users with good experience. Through 6 months running, DiPot has obtained plenty of data and captured some attack samples from Internet. The experimental results demonstrate the effectiveness of DiPot.

Acknowledgement. The authors gratefully acknowledge the anonymous reviewers for their helpful suggestions.

References

1. Mo, Y., Chabukswar, R., Sinopoli, B.: Detecting integrity attacks on SCADA systems. IEEE Trans. Control Syst. Technol. **22**(4), 1396–1407 (2014)
2. Kleinmann, A., Wool, A.: Automatic construction of statechart-based anomaly detection models for multi-threaded SCADA via spectral analysis. In: ACM Workshop on Cyber-Physical Systems Security and Privacy. ACM, pp. 1–12 (2016)

3. Zhou, C., Huang, S., Xiong, N., et al.: Design and analysis of multimodel-based anomaly intrusion detection systems in industrial process automation. IEEE Trans. Syst. Man Cybern. Syst. **45**(10), 1345–1360 (2015)
4. Tomlin Jr., L., Farnam, M.R.: A clustering approach to industrial network intrusion detection [EB/OL]. http://insurehub.org/sites/default/files/reports/CyberSecurity_Final_Research_Report_LTomlin_MFarnam%20(1).pdf
5. Goldenberg, N., Wool, A.: Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems. Int. J. Crit. Infrastruct. Prot. **6**(2), 63–75 (2013)
6. Bodenheim, R., Butts, J., Dunlap, S., et al.: Evaluation of the ability of the Shodan search engine to identify Internet-facing industrial control devices. Int. J. Crit. Infrastruct. Prot. **7**(2), 114–123 (2014)
7. Serbanescu, A.V., Obermeier, S., Yu, D.Y.: A flexible architecture for industrial control system honeypots. In: 2015 12th International Joint Conference on e-Business and Telecommunications (ICETE). IEEE, vol. 4, pp. 16–26 (2015)
8. Formby, D., Srinivasan, P., Leonard, A., et al.: Who's in control of your control system? Device fingerprinting for cyber-physical systems. In: Network and Distributed System Security Symposium (NDSS) (2016)
9. Krawetz, N.: Anti-honeypot technology. IEEE Secur. Priv. **2**(1), 76–79 (2004)
10. Bodenheim, R.C.: Impact of the Shodan computer search engine on internet-facing industrial control system devices. Air Force Institute of Technology Wright-Patterson AFB OH Graduate School of Engineering and Management (2014)
11. Serbanescu, A.V., Obermeier, S., Yu, D.Y.: ICS threat analysis using a large-scale honeynet. In: Proceedings of the 3rd International Symposium for ICS & SCADA Cyber Security Research. British Computer Society, pp. 20–30 (2015)

MSA vs. MVC: Future Trends for Big Data Processing Platforms

Yuming Lu[1], Wei Liu[1(✉)], and Haoxiang Cui[2(✉)]

[1] Shenzhen Key Lab for Visual Media Processing and Streaming Media, Shenzhen Institute of Information Technology, Shenzhen 518172, People's Republic of China
{luyuming,liuwei}@sziit.edu.cn
[2] College of Information Engineering, Shenzhen University, Shenzhen 518060, People's Republic of China
Jumeng_hulk@126.com

Abstract. Big data processing systems design is highly prioritized concern for both academia and industry. The conventional MVC architecture exposes limitations on system scalability and consistency. The task of integrating new services into an existing commercial application platform has become an impossible task and torturing nightmare for the system development team. The innovative MSA architecture is aimed to solve such a problem. The main contribution of this paper is comparison between the MSA and MVC system design and development architectures, summaries future research and development issues and challenges. This paper first discusses the problems and challenges of big data management, compares and discusses the characteristics of MVC and MSA patterned big data processing (BDP) platforms. Then we verify the MSA big data management systems, distributed data storage and the progress of the large data storage architecture utilize an experimental BDP platform. Finally list future research and development direction to provide valuable reference for further work.

Keywords: Micro-service architecture · Big data processing platform

1 Introduction

In the past three decades, Internet of Things (IoT) has drawn worldwide attention, and focus industry onto three key technology topics: Big Data, Cloud Computing, and Artificial Intelligent. In order to serve the purpose of processing Big Data, the Cloud Computing act as the brain of the IoT, responsible of gathering information of data/content that scattered all over the network then plan/compute utilizes Artificial Intelligent algorithms for best distributed storage and optimum transmission/distribution paths. The extensibility of relational database management system (RDMS) developed use the structured query language (SQL) has encountered unprecedented obstacles in the exponentially growing Internet environment and cannot be qualified for big data analysis [4]. The RDMS pursues high consistency and correctness. The vertical scaling system, by increasing or replacing the CPU, memory, and hard disk to extend the capabilities of a single node, will eventually run into a "bottleneck" [5].

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 310–320, 2018.
https://doi.org/10.1007/978-3-319-73830-7_31

The big data technology solves four core problems [9]: Storage, how can a huge amount of data be stored efficiently? Mainly including HDFS and Kafka; Computation, how quickly can you calculate a huge amount of data? Mainly include MapReduce, Spark, Flink, etc. Query, how to quickly query large amounts of data? Mainly for NoSQL and Olap, Nosql mainly includes Hbase, Cassandra, etc. Olap includes Kylin, Impala, etc. Nosql mainly solves random query, and Olap technology mainly solves related queries. Data mining, How to excavate a huge amount of data to excavate hidden knowledge? The current hot machine learning and deep learning techniques are included TensorFlow, Caffe, Mahout, etc. Google's various big data applications is build base on its own set of cloud computing technologies and tools for big data processing, namely mapReduce and bigtable [6, 7]. The Hadoop and Spark is the most popular BDP these days amounts industry.

2 Big Data Processing System

There are many immerging BDP systems, this paper investigates representative systems from two angles of views: load processing and data type. The application scope of various systems is clarified, however many areas of big data processing platform technology still require further in-depth research.

From the load processing perspective of view, the current system can be classified into three categories: batch processing, stream processing and interactive processing. The batch system emphasizes high throughput of the system, such as the Hadoop system. Streaming processing is an increasingly popular research field in recent years. Many applications with high computational timeliness requirements are expected to be completed in as short a delay as possible, such as online advertising recommendations, intrusion detection and identification, and so on. Current representative Streaming computing systems include yahoo's S4, Twitter's Storm, Google's MillWheel, Microsoft's TimeStream, and DStream, etc. Interactive query analysis is mainly used in large-scale data warehouses, from early Hive to Dremel, Impala and Shark, and various Sql-on-Hadoop and massively parallel processing (MPP) systems appear continuously.

From the perspective of data types, current systems provide a variety of data abstractions such as collections, tables, graphs, and matrices. Usually a programming framework is suitable for solving certain kinds of problems and not applicable to all kinds of problem areas. For example, MapReduce is suitable for solving the collection of data that is independent of the record; Piccolo utilizes distributed memory storage table data to speed up the operation efficiency of iterative computation. MadLINQ provides a large-scale matrix operation that simplifies the development of a large class of machine learning and data mining algorithms.

2.1 Streaming Big Data Processing Frameworks: Storm, Spark, and Samza

Storm, is a graph structure originally designed for real-time computing. The distribution topology will be submitted to the cluster, which will distribute the code from the master node (master node) in the cluster and assign the task to the working node. One topology

includes spout and bolt nodes. Spout sends the message as well as responses for sending the data stream to the tuple. Bolt is responsible for converting these streams, so that the bolt can perform calculations, filters, and so on. The bolt can also send the data to other bolt randomly. The tuple emitted by spout is an immutable array that corresponds to a fixed key value pair.

Spark Streaming is the core of Spark API extensions, do not like Storm processing data flow one at a time, before processing, spark will advance its segmentation for a period of a batch job according to the time interval. Spark's abstraction for persistent data streams is called discretized stream (DStream), and a DStream is a micro-batching (micro-batching) RDD (Resilient distributed dataset); RDD is a distributed data set that can be operated in parallel in two ways, namely, the conversion of arbitrary functions and sliding window data [10].

Samza handles the data flow by processes each received message separately. Samza's streaming unit is neither tuple nor Dstream, but messages. In Samza, data streams are separated and each part consists of an ordered array of read-only messages, each with a specific ID (offset). The system also supports batch processing, which process multiple messages for the same data stream partition. Samza's execution and data flow modules are pluggable, although Samza features a Yarn that relies on Hadoop (another resource scheduler) and Apache Kafka.

The above three kinds of real-time computing systems are open-source systems, with the advantage of low latency, scalability, and fault tolerance. Their common characteristic is allows users to run code, data flow and task allocation in parallel with a wide range of computing device, as well as high capability of fault tolerance. In addition, they can all simplify the complexity of the underlying implementation. The terms of the three frameworks are different, but the principle concept is similar.

2.2 Programming Framework Based on Data Flow Model

Many programming frameworks boil down to data flow models such as MapReduce, Dryad, Spark, Flume Java, and Storm. The data flow model utilize a directed acyclic graph (DAG) to express a calculation, and the vertices in the graph represent the computation task, while the side represents the data dependency.

MapReduce computes two simple programming interfaces for programmers, namely Map functions and Reduce functions. The input and output data of the task are organized in a key value manner and the framework automatically aggregates the records of the same key values in the intermediate results, as the losing Value of the Reduce function. MapReduce has great advantages. Firstly it is highly scalable and can be executed concurrently on thousands of machines; secondly, it is fault-tolerant, and even if the cluster fails, it usually does not affect the normal operation of the task [10, 11]; finally the simplicity, the user only needs to complete the map and reduce functions to complete the parallel processing of large-scale data. MapReduce also has some limitations, such as the long startup time of map and reduce tasks, and unsuitable application scenarios with high timeliness requirements. Its Graphs model has several disk read and write and network transmission, for iterative machine learning applications, often require the same graphs tasks performed repeatedly, this brings a lot of disk read and

write and network transmission overhead, causes low efficiency. Graphs, including two stages of Map and Reduce tasks, the Map after the phase of the mission to Reduce phase of the mission, unable to express complex topology between tasks, essentially graphs model can be thought of as a special case of the DAG calculation.

Dryad adopts the general DAG calculation model, which can flexibly express the topology of tasks, which also brings a certain programming burden due to the need for the programmer to explicitly construct the topology. A Dryad provides more mechanism in data transmission, including Shared memory, TCP pipes, temporary files and distributed file system, etc., and graphs only provides temporary file transmission way between the maps and reduce stage intermediate results.

Spark use DAG, a data set of conversion to represent a complete data processing, DAG in the vertex is elastic resilient distributed data (resilient distributed dataset, RDD). RDD is an abstraction of an intermediate data set repeatedly used in iterative computation, representing a collection of read-only records. RDD created after its contents cannot be modified, but can undertake various transformation operations to RDD, the content of the original RDD after transformation to form the new RDD, Spark will maintain the dependencies between RDD, called Lineage (Lineage). This chain operation not only simplifies application development, but also makes failover easier. Spark sets the RDD fragmentation (partition) to multiple machines while performing coarse-grained conversion operations, namely data parallelism. A record entry into Spark will undergo multiple conversion operations output, or pipelining. Spark provides RDD. As memory gets cheaper, memory based computing becomes more and more a choice in system design [11].

2.3 Machine Learning and Deep Learning

Machine learning refers to any type of computer program that can "learn" by itself without being programmed. The term was first coined by Alan Turing in a famous essay in 1950, "computational mechanics and artificial intelligence," and came up with "can a robot think?". Most of the "brains" that are used for predictive planning (including spam filtering, product recommendation and fraud detection) are machine learning algorithms, e.g. Linear classification algorithm as shown in Fig. 1.

Data processing scientists can write machine learning algorithms using a range of techniques and languages, including Java, Python, Scala, and so on. They can also use pre-built machine-learning frameworks to speed up the process; Mahout is a popular machine learning framework on Apache Hadoop, and Apache Spark's MLlib library has become a standard.

Deep learning is a form of machine learning that can take advantage of either supervised or unsupervised algorithms, or both. While not necessarily new, but the deep learning in recent popularity soared, as to accelerate the solutions to some difficult types of computer problems, computer vision and is the most significant application field of natural language processing (NLP) [12]. Deep learning is based on the representation learning (or feature learning) of machine learning theory. Through hierarchical learning process extraction of advanced, complex abstractions as data representation, deep learning model produces results faster than standard machine learning methods.

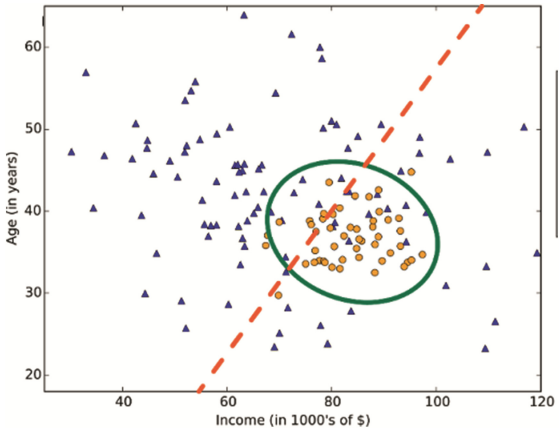


Fig. 1. Linear classification algorithm

Deep learning consists of many hidden layers, each of which take input from the previous layer, processes it, and outputs it to the next level in a Daisy chain. Such a convolution neural network (CNN) is shown in Fig. 2. AlphaGo, developed by Google's DeepMind team, is popular for two main reasons. First, CNN was running faster on GPUs, such as Nvidia's Tesla K80. Secondly, data scientists realized that with a larger database to train CNN can greatly improve the performance and the accuracy in the computer vision applications that utilizes NLP algorithm.

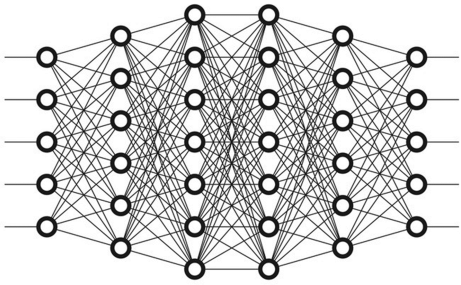


Fig. 2. Hidden layers of neural networks

3 MVC Architecture Introduction

The model-view-controller (MVC) architecture is a widely used software design pattern.

The application systems developed under the MVC pattern are divided into three tiers, namely model layer, view layer, and control layer. With different layers of the multi-layer system architecture level classification can be unified, corresponding multi-layer system architecture model of data service layer and business logic layer, view and

controller corresponding to the performance of the multi-layer system architecture layer. There may be overlaps between the layers. MVC-basic architecture is shown in Fig. 3.

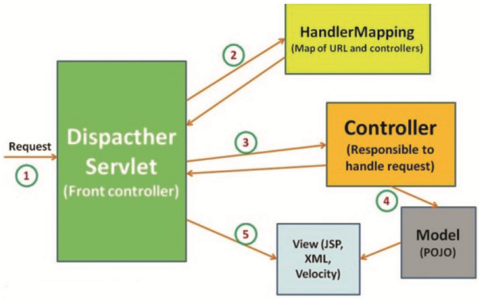


Fig. 3. MVC-basic architecture

3.1 Models Views and Controllers

Model represents the processing of business logic and the development of business rules. The processing of the business logic is transparent to other layers, the model accepts data from the view request and returns the final processing result. The core of MVC is the design of business logic. MVC does not provide a design approach to the model, but defines and interprets how the models should be organized to facilitate model refactoring and reusability. In addition to the business logic, there is a very important part of the model is a data model, i.e. entities object data (persistence), to save a data table in the database, for example, to get the data from a database. All the operations related to the database are limited to this model. In the case, the storage of the data is defined in the entity Bean and then implemented by the method provided by the Bean Entity Manager interface (such as adding, deleting, querying, modifying, etc.).

View represents the user interface. For Web applications, it can be composed of static HTML, or it can be composed of XHTML, XML, ASP, PHP, and JSP. For j2ee-based applications, which are typically made up of JSP, HTML, XML, and JSF, the specific choices can be selected based on the requirements and the developer's technical reserves. As the complexity of the application increases and the scale of the system expand, the processing of the interface becomes complex, and an application system usually contains several different views. The business process is handled by the Model. For example, a view of an order is responsible for passing the input data and requests of the user interface to the control and model, as well as accepting data from the model and displaying it to the user.

Controller act as a distributor, in order to select a specific model and view for a specific user request, the control layer does not process any data. For example, users click on a link, the controller match the user's information to a model, and view returns the choice to meet the user requirements. As a result, the controller associates several models and views as a middleware to establish multiple-to-multiple relationships. The processing of the system increases the complexity of the system and influences the performance of the system to a certain extent. But for the development of complex

business logic and user interface for the very large number of large enterprise application system, using the MVC can improve the robustness of the system and code reusability, thus greatly improving the efficiency of software development. Combining with specific techniques, the implementation of the MVC pattern can be made relatively simple.

For implementing MVC design pattern to develop a traditional j2ee-based enterprise application system, there are many options. The Model can choose standard EJB technology, or with lightweight object persistence techniques such as Hibernate and JDO. The View can be implemented using only JSP technology, as well as JSF and JSP. The Controller can choose Servlet technology to implement, or choose Struts, spring, Jobs Seam and other development frameworks. The MVC design pattern has many advantages: first, improve the reusability: multiple views can share the same model, makes the model component reuse can be different view, improve the reusability of the code. Second, to build loosely coupled systems: MVC is independent at all levels, one of which cannot affect the other two levels, which can help build a loosely coupled application system [13]. Third, high degree of flexibility: connect different views and models with the controller, realize different functions, and make the system configuration and implementation highly flexible. Fourth, the system development process of manageability.

However MVC architecture BDPs exposes limitations on system scalability and consistency. The query execution time grows linearly with the amount of data growth. The task of integrating new services into an existing commercial application platform has become an impossible task and torturing nightmare for the system development team.

3.2 MSA Architecture

The micro-service architecture (MSA) pattern is a term that has emerged in the field of software architecture patterns in the last two years. Although its birth time is not long, but its various speeches, articles, books on the frequency of the appearance has made many people aware of its impact on the software field. The birth of MSA is not an accidental, but the result of the rapid development of the Internet, virtualization technology. The developers and operation IT professionals (DevOps) culture of continuously develop agile and lean methodology and fast development are limited by the traditional monolithic architecture, which it cannot adapt to the rapidly changes and increasing services.

The MSA is an architectural pattern that advocates the partitioning of a single application into a small set of services that provide the ultimate value to the user by coordinating and cooperating with each other. Each service runs in its own process, with lightweight communication mechanisms between services and services (usually the RESTful API based on the HTTP protocol). Each service is built around the specific business and can be deployed independently to production environments, class production environments, and so on. In addition, should try to avoid a unified, centralized service management mechanism, for a specific service, according to the business context, select the appropriate language, tools to build it. The core benefits of the MSA includes: small and focused services implementation; independent processes; lightweight communication

mechanisms; loosely coupled distributed data management, independent deployments tasks.

With the rapid development of the market and the expansion of business, the application of single block architecture is facing more challenges, and its reconstruction and is imperative. The emergence of the MSA, is the rapid development of the Internet, virtualization technology applications and sustainable delivery, DevOps a comprehensive product. With the personalization of user demand and the shortening of product life cycle, the MSA is the inevitable direction for the development of software architecture in the future, such as flexibility, expansibility, scalability and high availability. At the same time, the popularity of container virtualization technology represented by Docker will greatly reduce the cost of service implementation and provide a solid foundation and guarantee for the service landing and large-scale use. Figure 4 shows MSA architecture.

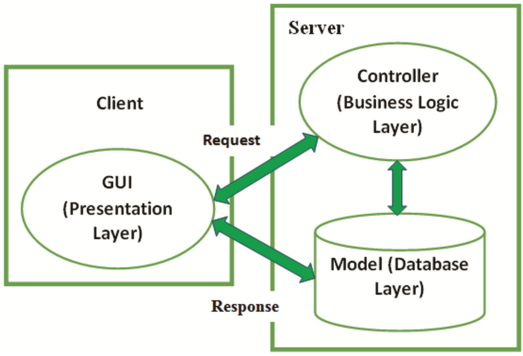


Fig. 4. MSA architecture

4 MSA-BDP Establishment

4.1 Big Data Processing Platform Experiment

In the analysis of big data, different analytical platforms need to be used to analyze objects and targets. Based on network traffic for the huge amounts of metadata, regular import Hadoop platform HDFS file system for storage, and use a HIVE for the first time digging to get orders of magnitude significantly reduced safety related data. It is also widely used depending on the content of the storage.

PostgreSQL, MySQL, and other traditional RDMS are used to classify and store the security events and associated information for the first analysis. RDMS can satisfy the regular data high real-time requirements of large concurrent queries, Hadoop can cope with the huge amounts of data of low real-time demand small concurrent queries, both strengths and cannot replace each other, can only meet the demand of actual business complement each other. In addition, flexible use of Python scripts and Linux Shell commands can also greatly improve the efficiency of your system. For the sake of security analysis, a 24 physical machine node is used to analyze the experimental

platform. Each node with 64 GB of memory, two Xeon E5-2670 CPU, two pieces of 240 GB SSD and 12 pieces of 2 TB SATA hard disk, all done through High Speed Ethernet interconnection between nodes, the total capacity of HDFS is 333 TB (due to replication can be set up to 3, the real effective capacity of 111 TB). The Hadoop software was deployed using the Cloud-era Standard 4.8.0 release, using CDH4.5.0+IMPALA 1.2.1+SOLAR 1.1.0. The most commonly used queries are still using HIVE to quickly implement simple MapReduce statistics using SQL statements. The discovery is based on clear during actual use the GUI query is still not stable enough, and from the automation considerations, the actual queries are ultimately performed under the command line interface (CLI). Since the overall hardware performance of the current Hadoop platform is still available, the actual application demand response time is also acceptable, so there is no compression or partition column to establish index optimization. In order to further improve the efficiency of the query, there is much work to be done in this area.

4.2 Metadata File Storage

This study utilizes the high-speed document's search engine module, and adopts the open source search engine **solar**. The metadata extraction module is responsible for converting the source data format and extracting the required information from the source data to form metadata in an XML file format [14]. The data migration module has migrated these source data and metadata to the storage module in a matching form. Search speed and efficiency of the current storage device, high-speed sol search engine is used in designing the metadata timing crawl module stored in the latest metadata, and upload it to the search engine in the module. Metadata file storage performance is shown in Fig. 5.

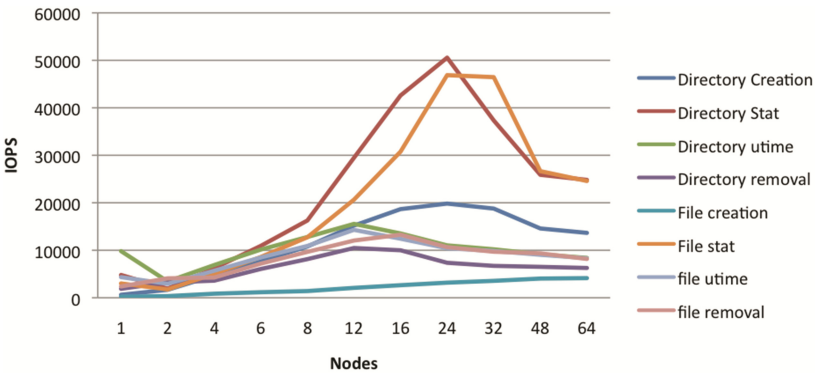


Fig. 5. Metadata file storage

All data is stored in an inherent format and uploaded to the high speed search engine, with the following advantages and characteristics compared with traditional data processing methods. The traditional data processing method is that when the storage data reaches a certain scale, the retrieval response speed will be significantly slower, so it needs to be tuned continuously. And this medical big data platform USES the high

speed search engine and establishes index, the retrieval performance will not decrease obviously, and need not tune. In the experiment, the retrieval of the medical data of a patient in different systems including image information took less than 1 s, which was more than 10 times faster than the traditional data processing method. This system is flexible, demand for new services only need to add new data storage and the development of new demand function according to these data; second, compared with the traditional ETL + BI data analysis tool, based on the characteristics of files are stored under the trend of big data, but also has the traditional relational database ETL + BI tool has the incomparable advantage, not only can deal with structured data, unstructured data can be retrieved statistics and display; When amounts of data increased, the traditional relational database library cause enormous pressure and danger, and the platform horizontal simply increasing storage can, currently support extended to petabytes.

5 Conclusion

Big data contains great value, mining the value behind the massive scale of data set require more agile and lean big data processing (BDP) platform. However many challenges and limitations lies ahead with the existing BDP platforms implemented with monolithic architecture. Major internet services providers like Google, Amazon, LinkedIn, Twitter, etc., has occupied the forefront of technological innovation with the quick upgrading iteration, and they have accumulated experience that is worth learning from other industries. We reviewed some open source software tools, which plays unique and important role in terms of spreading big data technology in the industry, and should be given sufficient attention in technological innovation. Big data management platforms optimization with distributed data management and centralized cloud computing engines will become a new research and development hotspot in the future. This paper briefly discusses the problems and challenges of big data management, compares and discusses the characteristics of MVC and MSA patterned big data processing platforms. The current big data management systems, new distributed data storage and the progress of the current large data storage architecture and management algorithm are summarized. Finally, we listed future research and development direction to provide valuable reference for future research in this field.

Acknowledgements. We would like to present our appreciation for the support from the National Science Foundation of China project: NSFC-Guangdong project U1301252, Science and Technology Innovation Commission Foundation of Shenzhen project: JCYJ20160608151239996 and JCYJ20170307114301790.

References

1. Russakovsky, O., Deng, J., Su, H., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
2. Anagnostopoulos, I., Zeadally, S., Exposito, E.: Handling big data: research challenges and future directions. J. Supercomput. **72**(4), 1494–1516 (2016)

3. Qiu, M., Gai, K., Xiong, Z.: Privacy-preserving wireless communications using bipartite matching in social big data. Future Gener. Comput. Syst. (2017)
4. Gai, K., Qiu, M., Zhao, H., et al.: Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. J. Netw. Comput. Appl. **59**(C), 46–54 (2016)
5. Gai, K., Qiu, M., Zhao, H.: Energy-aware task assignment for mobile cyber-enabled applications in heterogeneous cloud computing. J. Parallel Distrib. Comput. **111**, 126–135 (2017)
6. Sun, D., Zhang, G., Yang, S., et al.: Re-stream: real-time and energy-efficient resource scheduling in big data stream computing environments. Inf. Sci. **319**(32), 92–112 (2015)
7. Stepnowsky, C., Sarmiento, K.F., Amdur, A.: Weaving the internet of sleep: the future of patient-centric collaborative sleep health management using web-based platforms. Sleep **38**(8), 1157–1165 (2015)
8. Jordan, A.J., Huitema, D., Hildén, M., et al.: Emergence of polycentric climate governance and its future prospects. Nat. Clim. Change **5**(11), 34–54 (2015)
9. Bajaber, F., Elshawi, R., Batarfi, O., et al.: Big Data 2.0 processing systems: taxonomy and open challenges. J. Grid Comput. **14**(3), 1–27 (2016)
10. Wolfert, S., Ge, L., Verdouw, C., et al.: Big Data in smart farming – a review. Agric. Syst. **153**(12), 69–80 (2017)
11. Greene, A.C., Giffin, K.A., Greene, C.S., et al.: Adapting bioinformatics curricula for big data. Brief. Bioinform. **17**(1), 43–50 (2016)
12. Shao, Y., Kai, L., Lei, C., et al.: Fast parallel path concatenation for graph extraction. IEEE Trans. Knowl. Data Eng. **PP**(99), 1 (2017)

Attention-Aware Path-Based Relation Extraction for Medical Knowledge Graph

Desi Wen[1], Yong Liu[2], Kaiqi Yuan[1], Shangchun Si[1], and Ying Shen[1(✉)]

[1] Shenzhen Key Lab for Cloud Computing Technology and Applications, School of Electronic and Computer Engineering (SECE), Institute of Big Data Technologies, Peking University, Shenzhen 518055, People's Republic of China
{wendesi, shangchunsi}@sz.pku.edu.cn, kqyuan@pku.edu.cn, shenying@pkusz.edu.cn

[2] IER Business Development Center, Shenzhen, People's Republic of China
l3312962646@l89.cn

Abstract. The task of entity relation extraction discovers new relation facts and enables broader applications of knowledge graph. Distant supervision is widely adopted for relation extraction, which requires large amounts of texts containing entity pairs as training data. However, in some specific domains such as medical-related applications, entity pairs that have certain relations might not appear together, thus it is difficult to meet the requirement for distantly supervised relation extraction. In the light of this challenge, we propose a novel path-based model to discover new entity relation facts. Instead of finding texts for relation extraction, the proposed method extracts path-only information for entity pairs from the current knowledge graph. For each pair of entities, multiple paths can be extracted, and some of them are more useful for relation extraction than others. In order to capture this observation, we employ attention mechanism to assign different weights for different paths, which highlights the useful paths for entity relation extraction. To demonstrate the effectiveness of the proposed method, we conduct various experiments on a large-scale medical knowledge graph. Compared with the state-of-the-art relation extraction methods using the structure of knowledge graph, the proposed method significantly improves the accuracy of extracted relation facts and achieves the best performance.

Keywords: Relation extraction · Path attention · Knowledge graph

1 Introduction

In recent years deep learning has been one of the most influential and representative technologies in the field of artificial intelligence. The unprecedented breakthroughs in application of this technology lead to a new wave of development both in academia and industry. If intelligent machine has a brain in the future, deep learning will be learning mechanism of the machine brain, and knowledge graph will be knowledge base of it. Knowledge graph, crucial for big data intelligence, will also impact on

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 321–331, 2018.
https://doi.org/10.1007/978-3-319-73830-7_32

areas such as natural language processing, information retrieval, and artificial intelligence profoundly.

Knowledge graph is essentially a semantic network composed of entities and the relationship between entities. Nowadays, knowledge graph has already been widely used in various applications, such as question answering [1] and recommender system [2].

There are many open source knowledge graph projects, such as freebase, YAGO, Dbpedia, etc., but knowledge graph is still far from complete. Therefore, relation extraction supplements knowledge graph extracting semantic relations between entities. Distant supervision [3] is the most widely adopted method for relation extraction. However, the distant supervised relation extraction method requires a massive amount of sentences containing two entities, which is strict restriction for many entity pairs; furthermore, most of the existing relation extraction models using external in formation rather than abundant implied information within knowledge graph.

To address the above issues, we propose a path-based strategy to infer relations from the structure of knowledge graph rather than text. For an entity pair that has a potential relation, we first calculate the path between entity pairs from the existing knowledge graph, treat path as a sequence, and then encode the sequence using recurrent neural network. However, path has its corresponding establishment likelihood. Inspired by this observation we add attention model to put different weights on different paths, With attention weights embodied in path vector, relations are thus extracted.

The contributions of our work can be summarized as follows:

- Compared with other text-based relational extraction models, our model uses path information in the knowledge graph to substantially reduce the difficulty of training data acquisition;
- Take path attention model to assign corresponding weights for different paths, which reduces noise from inadequate paths;
- We construct a medical knowledge graph to evaluate our model. The experimental results demonstrate our model achieves the highest precision over other structure-based models.

2 Related Work

Relation extraction has been an important branch of knowledge graph completion, emerging many excellent research models. Lin et al. [4] propose a multi-sentence relation extraction model. For an entity pair, relation classification achieves by calculating eigenvector of the sentence containing the entity pair through using Convolutional Neural Network (CNN) and adding sentence attention model to assign sentence weights. Miwa and Bansal [5] propose a relation extraction model based on word sequence and tree structure.

However, distant supervised model requires a large number of sentences containing two entities as training sets. In some specific domains, such as medical field, are hard to meet the above conditions. To address this issue, Zeng et al. [6] propose a path-based relation extraction model that uses the CNN to extract eigenvectors of sentences

containing a single entity and constructs middleware between the two target entities for reasoning to extract relations. Nevertheless, entities may belong to multiple classes, causing ambiguity when applying single sentence.

Besides extracting relations from text, another way is from the structure of knowledge graph, which includes knowledge representation learning. Knowledge representation learning mainly suggests representation learning for entities and relations in knowledge graph, transforming entities and relation vectors into the low-dimensional dense vector space, and carrying out corresponding calculation and reasoning.

TransE [7] is a simple but efficient model proposes by Bordes et al. For triple (h, r, t), transE considers h + r = t. Compared with the previous knowledge representation learning model, parameters are relatively few in transE. The model is simple and intuitive, with small calculation, especially good at dealing with one-to-one relations. However, one-to-many, many-to-one and many-to-many relations are too difficult for transE model to deal with.

Thus, Feng [8] propose the transH model. It maps relations to another hyperplane in the same space and designs complicated sampling method for training. However, Ji et al. put forward the transD [9] model, and believe that entity is a complex of multiple attributes, and different relations concern with different attributes of the entity, so entity and relation should be in different spaces.

In the knowledge graph, some of the entity relations connect a large amount of entities, whereas some entity relations are quite simple. If one model is used for all cases, it may lead to inadequate training for complicated relations and overfitting for simple relations. Therefore, Ji et al. [10] propose the tranSparse model, using relatively dense matrices for complex relations and sparsely matrices for simple relations via SparseMatrix.

Knowledge representation models above utilize directly connected triples as features, but path [11] in the knowledge graph contains numerous implied information. Das et al. [12] use triple path as a sequence and that entities might belong to multiple classes is taken into account. So they add class information to triple vector representation, and put sequence into Recurrent Neural Network(RNN) to extract relations. However, the model has two obvious weaknesses: (1) ignore multiple paths; (2) ignore soft reasoning, as the establishment probability of paths is not always equal to 1 or 0. Since in medical field, relations for symptoms corresponding to diseases and appropriate drugs corresponding to symptoms establish only to some extent [13].

3 Methodology

Given a set of entity pairs (head, tail), our model calculates path among entity pairs and computes the likelihood of each relations r based on the path. In this section, we will introduce our model as follows:

Calculate Path: For a given set of entity pairs (h, t), we find a set of paths $\{x_1, x_2, \ldots, x_n\}$ from the knowledge graph, where x_i $(\mathrm{i} = 1, 2, \ldots, \mathrm{n})$ is the acyclic path taking node h as start and node t as end.

Path Encode: Given a path x, use Gated Recurrent Unit (GRU) to compute its distributed representation.

Path Attention: After learning distributed representation of all paths, attention model assigns different weights to paths, from which relations among entity pairs are calculated.

3.1 Calculate Path

For a group of entity pairs (h, t), we calculate acyclic path that satisfies conditions (source, target, minLen, maxLen, maxPaths) from the knowledge graph G, where G is directed graph, source is the starting of path, target is the ending of path, minLen is the lower limit of path length, maxLen is the upper limit of path length, maxPaths is the upper limit of the number of paths.

We adopt the breadth-first search to determine whether there exists a path to satisfy the (source, target, minLen, maxLen) condition in G, and if so, use the depth-first search to find all the paths satisfying the (source, target, minLen, maxLen, maxPaths) condition.

Finally, we can get a set of head-to-tail paths $\{x_1, x_2, \ldots, x_n\}$, the structure of path x is $((h_1, r_1, t_1), (h_2, r_2, t_2), \ldots, (h_m, r_m, t_m))$, where $h_1 = h$, $t_m = \mathrm{t}$, $t_{j-1} = h_j \, (i \leq j < m)$.

3.2 Path Encoding

Triple Representation: After Sect. 3.1 we get a set of paths, and each path x contains a number of triples, each triple $(\mathrm{h}, \mathrm{r}, \mathrm{t})$ contains two entities and one relation. Entities and relations have different representations. We derive idea from the transE model that entities and relations are in the same dimension space, so they are mapped into a d-dimensional space.

Entities and relations are represented by column vector of the same embedded matrix V, $\mathrm{V} \in R^{d \times (e+r)}$, where e indicates the total number of entities and r indicates the total number of relations.

We concatenate vector representation of two entities with entity representation of relation, to form a triple representation t, $\mathrm{t} \in R^{3d}$.

At last, we transform the triple path into a set of vector sequence $\mathrm{x} = \{t_1, t_2, \ldots t_m\}$ and input it to GRU.

GRU: Gated Recurrent Unit (GRU) proposed by Cho et al. [14] shared parameters in time series and thus associates connected input. It consists of reset gate r, update gate z and a memory cell s, calculated as follows:

$$\mathrm{z} = \sigma(t_i U_z + s_{i-1} W_z + b_z) \tag{1}$$

$$\mathrm{r} = \sigma(t_i U_r + s_{i-1} W_r + b_r) \tag{2}$$

$$\mathrm{h} = \tanh(t_i U_h + (s_{i-1} \cdot \mathrm{r}) W_h + b_h) \tag{3}$$

$$s_i = (1 - z) \cdot \mathrm{h} - \mathrm{z} \cdot s_{i-1} \tag{4}$$

Where t_i is the input vector, representation vector of triple t in our task, h is the output vector, z is the update gate, r is the reset gate, U_z, U_r, U_h, W_z, W_r, $W_h R^{3d \times 3d}$ are the weight matrix, b_z, b_r, b_h are the offset, σ is the sigmoid function, $\cdot$ is the Hadamard product.

We use vector sequence $\mathrm{x} = \{t_1, t_2, \ldots t_m\}$ obtained by Sect. 3.2 as the input of GRU, and select the final output vector h_m as the final encoding representation of current triple path p, $\mathrm{p} = h_m$.

Path Attention: After the previous steps, we will encode path with head entity as start and tail entity as end to form a path matrix $\mathrm{S} \in R^{3d \times m}$, which consists of encoded path $[p_1, p_2, \ldots p_m]$ generated by GRU.

Obviously, next step should use all the path information in matrix S to extract relations of relation pairs (h, t). However, not all the paths are correct. In medical field, each path has its own establishment probability. That is the reason we introduce the attention model to give different weights α_i for each path p_i, and calculate the vector representation pr in path matrix S.

$$\mathrm{pr} = \sum\nolimits_i \alpha_i p_i \tag{5}$$

According to the different settings of α, our model is divided into the following three categories.

One: We randomly select a path as representative from path set, which means α is a one hot vector. This approach is a naive baseline of path attention.

Average (AVE): We assume that each path in the path set has the same contribution to pr. Consequently, we assign the same weights for each path. Where pr equals to average of each path vector in path set.

Path Attention (PATT): We are supposed to calculate different weights α_i for each path p_i due to its different contribution.

$$\mathrm{M} = \tanh(W_s \mathrm{S}) \tag{6}$$

$$\alpha = \mathrm{softmax}\left(w^T \mathrm{M}\right) \tag{7}$$

$$\mathrm{pr} = \mathrm{S}\alpha^T \tag{8}$$

Where $\mathrm{M} \in R^{3d \times m}$ is the mapping matrix of path matrix, $\alpha \in R^{3d}$ is the attention model weight, $\mathrm{pr} \in R^{3d}$ is path representation of the attention model weight, $W_s \in R^{3d \times 3d}$, $\mathrm{w} \in R^{3d}$ is projection parameters.

pr is the final path matrix representation, transformed to vector e with dimension equal to the number of relation categories r by a fully connected layer, and converted into conditional probability distribution y through softmax layer ultimately.

$$y = \text{softmax}(W_o pr + b_o) \tag{9}$$

Where $W_o \in R^{r \times 3d}$ is the mapping matrix of fully connected layer, $b_o \in R^r$ is the offset vector of fully connected layer.

4 Experiments

Experiments will prove that relation extraction in our model may take full advantage of path information in the knowledge graph for relation extractions and path attention could reduce negative effect of unreasonable paths. To start with, we will introduce the datasets in the experiment, one of the approach of building negative samples, and parameter settings in our model. Then verify the affect of path embedding in comparison with other triple embedding model. Last but not least, compare different path attention weight settings to prove the affect of path attention model.

4.1 Experiment Setup

Dataset: We have constructed a Chinese medical knowledge graph that covers information on diseases, symptoms, drugs, food, surgery and so on in the medical field. This knowledge graph has a total of 45427 entities, 26 relations and 396,172 triples. The experiment divides triples into 27 relations, where the redundant relations are unrelated, since most entities do not necessarily have relations among each other. We construct negative samples with unrelated entity pairs and choose negative samples relations as the 27th relation - unrelated relation. The transH model proposes a strategy of building negative samples which randomly replaces a head or tail entity for an entity pairs (h, t). However, negative samples constructed that way are quite rough, and whether or not entity pairs have relations is not for sure. Particularly, there are plenty of entities with relations but not directly connected. We design an algorithm for generating entity pairs of no relations. For the given entity pair (h, t), entities are randomly selected from the knowledge graph to replace the head entity and tail entity, forming $(\mathrm{h_w}, \mathrm{t})$ and $(\mathrm{h}, \mathrm{t_w})$ triples to make $(\mathrm{h_w}, \mathrm{t})$ and $(\mathrm{h}, \mathrm{t_w})$ not include in the knowledge graph, and $\text{Neighbors}(\mathrm{h_w}) \cap \text{Neighbors}(\mathrm{t}) = \emptyset$, $\text{Neighbors}(\mathrm{h}) \cap \text{Neighbors}(\mathrm{t_w}) = \emptyset$, where $\text{Neighbors}(\text{Entity})$ is an entity set that directly connected to Entity in current knowledge graph.

Comparative Method: We will use the knowledge graph representation model, transE, transH, transR to do comparative tests, because the structure information of knowledge graph applied in these models. The thoughts of knowledge graph representation models above are rather close, so relation extraction task could be described as follows: given a triple $(\mathrm{h}, \mathrm{r}, \mathrm{t})$, calculate $\|h^* + r^* - t^*\|$ and choose the smallest score relation r as its predicted relation, where h^*, r^*, t^* is different mapping of h, r, t in different models. The knowledge representation learning has a triple classification task, which is specifically used to determine whether a triple $(\mathrm{h}, \mathrm{r}, \mathrm{t})$ is a correct fact. Relevant algorithm in the task is not practical. Since it intends to find a value seg as

dividing line: $\|h^* + r^* - t^*\| < seg$ for the correct fact, $\|h^* + r^* - t^*\| \geq$ seg for the wrong fact, with select the highest correct rate seg by validation set. Whereas, the algorithm does not apply in our experiments because t it tends to judge all triples incorrect with the increase in the proportion of negative samples. Therefore, the negative samples relation is treated as class 27 in our experiment.

Parameter Settings: We employ three-fold cross validation method to verify our experiments. The path length lower limit minLen is generally set 2, the path length upper limit maxLen $\in \{4, 5, 6\}$, and the upper limit of the number of paths maxPaths is set as needed. Entity embedding size and relation embedding size d_e, $d_r \in \{30, 50, 100\}$, batch size B $\in \{64, 128, 256, 512\}$, dropout probability p $\in \{0.4, 0.5, 0.6\}$, path embedding size $d_p \in \{128, 256, 512\}$. In experiment, we set minLen $= 2$, maxLen $= 5$, maxPath $= 100$, $d_e = d_r = 50$, B $= 128$, p $= 0.5$, $d_p = 256$, and we choose 20 as the number of iterations in training.

4.2 Performance Comparison

This part we will verify the effect of path encoding. We take the GRU + ONE model in comparison with transE, transH, transD and tranSparse models in our experiment.

Table 1 shows our experimental results. The GRU + ONE model outperforms others remarkably since path information of more abundant information is taken into account, compared to those models that use only source and target information in the path, such as the TransE. It is proved that path encoding is more efficient than triple encoding in the task of relation extraction.

The proposed model has a comparatively substantial increase to other models. The reason may be more path information taken into account. On the contrary to knowledge representation model with single triples, path information makes decision taking advantage of all the triple information in paths. The more information we use, the higher accuracy we get.

Table 1. Comparison among GRU + ONE and trans series.

Model	Dataset 1	Dataset 2	Dataset 3
transE	0.5154	0.5157	0.5452
transH	0.5329	0.5532	0.5325
transD	0.5617	0.5345	0.5792
tranSparse	0.5711	0.5731	0.5882
GRU + ONE	0.7490	**0.7528**	**0.7628**

4.3 Effect on Path Attention

Now we will prove the effect of path attention model. There may be tens of paths between two entities in the knowledge graph, but we can not use all the paths due to computational ability restriction. Therefore, we divide test set into the following three categories to verify the effect of model with randomly selected paths.

One. For each group of entity pairs, we randomly choose a path that satisfies the (minLen, maxLen) condition and connects two entities, and use it to extract relations;
Two. For each group of entity pairs, we randomly choose two paths that satisfy the (minLen, maxLen) condition and connect two entities, and them to extract relations;
All. For each group of entity pairs, we choose path that satisfy the (minLen, maxLen, maxPaths) condition and connect two entities, and use it to extract relations.

Table 2 shows our experimental results. The accuracy of GRU + AVE and GRU + PATT model are lower than GRU + ONE model while selecting one path randomly. The GRU + ONE model takes a single path for training, so it could utilize characteristics of one path better. However, when it comes to all paths, the accuracy of GRU + AVE and GRU + PATT model outperform 0.02 higher than the GRU + ONE model, which covers too little information. In contrast to the GRU + AVE and GRU + PATT model, GRU + ONE model solely inputs in a single path involving small quantity of data. Particularly in deep learning of millions of parameters for training, little training data may lead to overfitting problem, making its generalization ability weak. This could also explain the reason why the accuracy of GRU + ONE model is relatively low.

Table 2. Performance of relation extraction with different number of sentences.

Model	Dataset 1			Dataset 2			Dataset 3		
Path	One	Two	ALL	One	Two	ALL	One	Two	ALL
GRU + ONE	**0.7528**		0.7529	**0.7528**		0.7528	**0.7628**		0.7628
GRU + AVE	0.7216	0.7016	0.7660	0.7201	0.7076	0.7718	0.7276	0.7184	0.7733
GRU + PATT	0.7490	**0.7318**	**0.7769**	0.7463	**0.7380**	**0.7733**	0.7546	**0.7480**	**0.7824**

Consider the GRU + AVE and GRU + PATT models. It can be inferred from Table 2 that the accuracy of GRU + PATT model is about 0.01 higher than that of GRU + AVE model while using all paths. Nevertheless, when using a single path or two paths, the accuracy of GRU + PATT model is approximately 0.03 higher than GRU + AVE model. Path attention model acts effectively even with incomplete information, since all paths are treated equally and rearranged the same weight in the GRU + AVE model when it occurs to path information shortage. Therefore, the accuracy is relevant to the proportion of unreasonable paths in path collection. The GRU + PATT model could increase reasonable path weights by reducing the unreasonable path weights in the way of adding path attention model to perform better, even if the part of unreasonable paths is still large.

In conclusion, path attention model could guarantee high level of accuracy although there are too many paths to acquire all the path information.

4.4 Case Study

Table 3 demonstrates two example of path attention selected from test data. For each triple, we select its paths with the highest and the lowest attention weight. By the use of path attention, our model could arrange larger weight for paths having higher establishment probability and smaller weight for paths having lower establishment probability. As the first triple illustrates, the two entities connected by relation "alias of the disease" are basically different expressions of the same entity, so this path is logical. And the path with lower score has multi-level "complication" relations, which could not make the two diseases "infectious shock" and "abdominal pain" treated in the same department definitely.

Table 3. Example of path attention.

Triple	Score	Path
(infectious shock, medical department, emergency department)	Max: 0.3298	(infectious shock, disease alias, septic shock), (septic shock, medical department, emergency department)
	Min: 0.0565	(infectious shock, complication, disseminated intravascular coagulation), (disseminated intravascular coagulation, complication, abdominal pain), (abdominal pain, complication, electrolyte disturbance), (electrolyte disturbance, medical department, emergency department)
(beryllium poisoning, complication, pulmonary edema)	Max: 0.9938	(beryllium poisoning, complication, pneumonia), (pneumonia, disease alias, lower respiratory infections), (lower respiratory infections, complication, pulmonary edema)
	Min: 0.0062	(beryllium poisoning, disease examination, urinary calcium), (urinary calcium, possible disease with higher score, hypercalcemic nephropathy), (hypercalcemic nephropathy, complication, uremia), (uremia, complication, pulmonary edema)

In the second triple, reasoning path with higher score derives relation between two diseases as "complication" from two "complication" relations, which is also reasonable even accompanied by some problems. The path with lower score speculates from disease "beryllium poisoning" to disease "pulmonary edema", through disease "uremia". Beryllium and its compounds against lungs cause disease "Beryllium poisoning", whose incidence site lies in lungs. "Uremia" is a kind of kidney disease of little connection with "Beryllium poisoning", which provide more rational explanation for path with high score. Consequently, the two paths are endowed with quite different weights by path attention model.

5 Conclusions

In this paper, we propose a model to explore relations based on path information instead of text information, which is supposed to reduce requirements of dataset. Besides, we employ GRU + path attention to assign different weights for paths to alleviate the negative effect of unreasonable paths. In experimental part, we compare with other models based on knowledge graph structure, and experiments demonstrate that our model is obviously superior to other models.

Next step we will expand our research from the following two aspects:

1. Our model relies on path information, and there are some key entities connecting thousands of entities in current knowledge graph. These will at exponential level increase the number of paths in the algorithm we construct paths. Therefore, we will consider a more effective way of building paths.
2. The knowledge graph contains not only structure information, but also plenty of text information, such as entity descriptions in general knowledge graph and drug instruction descriptions in medical knowledge graph. Next research will concentrate on how to align text information in the way of path.

Acknowledgement. This work was financially supported by the National Natural Science Foundation of China (No. 61602013), and the Shenzhen Key Fundamental Research Projects (Grant No. JCYJ20160330095313861, and JCYJ20151030154330711).

References

1. Yin, J., Jiang, X., Lu, Z., Shang, L., Li, H., Li, X.: Neural generative question answering, vol. 27, pp. 2972–2978 (2015)
2. Zhang, F., Yuan, N.J., Lian, D., Xie, X., Ma, W.Y.: Collaborative knowledge base embedding for recommender systems. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 353–362 (2016)
3. Mintz, M., Bills, S., Snow, R., Dan, J.: Distant supervision for relation extraction without labeled data. In: Joint Conference of the, Meeting of the ACL and the, International Joint Conference on Natural Language Processing of the AFNLP: Volume 2, pp. 1003–1011 (2009)
4. Lin, Y., Shen, S., Liu, Z., Luan, H., Sun, M.: Neural relation extraction with selective attention over instances. In: Meeting of the Association for Computational Linguistics, pp. 2124–2133 (2016)
5. Miwa, M., Bansal, M.: End-to-end relation extraction using LSTMs on sequences and tree structures (2016)
6. Zeng, W., Lin, Y., Liu, Z., Sun, M.: Incorporating relation paths in neural relation extraction (2016)
7. Bordes, A., Usunier, N., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: International Conference on Neural Information Processing Systems, pp. 2787–2795 (2013)
8. Feng, J.: Knowledge graph embedding by translating on hyperplanes. In: AAAI (2014)

9. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 687–696 (2015)
10. Ji, G., Liu, K., He, S., Zhao, J.: Knowledge graph completion with adaptive sparse transfer matrix. In: Thirtieth AAAI Conference on Artificial Intelligence, pp. 985–991 (2016)
11. Shao, Y., Kai, L., Lei, C., Zi, H., Cui, B., Liu, Z., et al.: Fast parallel path concatenation for graph extraction. IEEE Trans. Knowl. Data Eng. 99 (2017)
12. Das, R., Neelakantan, A., Belanger, D., Mccallum, A.: Incorporating selectional preferences in multi-hop relation extraction. In: The Workshop on Automated Knowledge Base Construction, pp. 18–23 (2016)
13. Shen, Y., Colloc, J., Jacquet-Andrieu, A., Lei, K.: Emerging medical informatics with case-based reasoning for aiding clinical decision in multi-agent system. J. Biomed. Inform. **56**(C), 307–317 (2015)
14. Cho, K., Merrienboer, B.V., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Computer Science (2014)

Information Centric Networking Media Streaming Experiment Platform Design

Yuming Lu(✉), Tao Hu, and Xiaojun Wang(✉)

Shenzhen Key Lab for Visual Media Processing and Streaming Media, Shenzhen Institute of Information Technology, Shenzhen 518172, People's Republic of China
luyuming@sziit.edu.cn, happy.hut@163.com, burandanxin@gmail.com

Abstract. Information Centric Networking (ICN) is a revolutionary concept that considers data content interconnected networking instead of equipment interconnected data transmission. Designing and optimizing such a system require not only a software simulator environment but also an experimental platform to test new type of big-data processing architecture. This paper introduce and contribute on put forward the idea of a new distributed publish-subscribe media big data processing platforms base on the ICN architecture. First part discusses the component that included in a ICN media streaming platform, and then describes the design of a distributed publish-subscribe (P/S) system, which manages the subscription query, storage, transmission and content scheduling that complies with the named data networking (NDN) framework. The simulation utilizes software defined network (SDN) controller to carry out artificial intelligence supported computing in the cloud to generate global view of the distributed media content.

Keywords: Streaming media · Information center networking
Publish-subscribe systems · Software defined networks · Named data networks

1 Introduction

In the Internet of Things (IoT) era, wireless communication technology developed so advanced that allows information society carry out revolutionary changes. According to Cisco, from 2013 to 2018, global Internet users will increase from 2.5 billion to nearly 4 billion (more than 51% of the world's population), global network equipment and the number of connections will increase from 12 to 210 billion, and video media traffic is growing in proportion in the overall traffic. Can contemporary internet IP architectures support such an exponentially growing traffic?

In IP networks, information exchange is usually based on uniform resource locators (URL). In the case of host outage or the intermediate router fails, the content will become unavailable and it is especially common in the mobile ad-hoc network (MANET) scenario [2, 3]. Internet-based applications gradually transforming end-to-end communication tasks to content acquisition tasks. In terms of transmission, streaming technology is highly efficient, packs multimedia contents in the format of streams, with one-tenth of startup latency reduction and requires not much cache capacity for webcasts.

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 332–343, 2018.
https://doi.org/10.1007/978-3-319-73830-7_33

TCP/IP architecture is proven incapable of supporting such fast changes to provide good network scalability, consequently ICN type of network architecture occurred [2, 3].

ICN refers to a series of new mechanisms to enable content name centric addressing, which differs from conventional equipment centric addressing, aim to carry out Artificial Intelligent supported media content streaming. The future ICN media streaming network compute a global node and content topology view for abstract and simplify the contents and cache distribution, in order to analyze, manage resources to provide sufficient user quality of experience (QoE). Designing computational platform for such a caching mechanism is the key for future ICN media transmission.

ICN assuming every node in the network capable of respond to a request and routing based on the requested content name, networks act not only as transportation platforms but also a content catching and distribution system. ICN allows content to be stored anywhere, so a copy of the content may be retrieved via singular or multiple paths from the nearest node. This not only improves response time but also alleviates the impact of network failure. Named data networking (NDN) is a US NFS funded project that studies the possible solution for ICN type of future Internet architecture [1]. ICN utilizes the name of the content as a routing index to separate the resource from its host, and caches forwarded data, allow the user to cache from the network without having to get the data from the source, thus reducing the amount of network traffic and response time. ICN has a natural advantage for supporting mobility. If mobile users move away from its original position and requesting the data, multicast interest packet will propagate and find the new location that is nearest to obtain these data, without rerouting back to the original data source. This solves the problems of connection brake down due to node movement.

Host-centric networks give each interface a network address, and the connections between applications are bound to a particular interface, making it difficult to easily switch between them. ICN separates the content from the host interface, making it easy to reuse requests on multiple interfaces, enables mobile devices to make better use of multiple interfaces such as blue tooth, UMTS, WIFI and other interfaces. NDN investigate technological evolution from today's host-centric to data-centric network architecture, which has its roots in an earlier project, Content-Centric Networking (CCN, first publicly presented by Van Jacobson in 2006).

Future ICN utilizes publish/subscribe (P/B) system to transmit information in the format of stream data, the user only cares about the content that they interested and publish their requests, not the location of the content [9], and implementing such system remains a industry challenge.

2 Streaming Technology

Streaming media refers to the continuous time media transmit over the Internet/Intranet using streaming technology, such as audio, video, or multimedia files. Streaming media technology allows users viewing or listening initial part of video and audio, only to wait a short period of buffering time, when download is incomplete, the rest of the multimedia streaming files that have not been buffered will continue to be sent by the server to the

local computer. In contrast, traditional technology must wait until the entire multimedia file has been downloaded. It divides multimedia files into compressed sub-stream packages.

2.1 Principle of Streaming Media Technology

The Internet is an intermittent asynchronous network based on packet transmissions, files are decomposed into many packets as they are transmitted. The dynamic network topology changes require caching along the transmit path and dynamic routing to reduce the impact of network transmission latency, jitter, miss match reception, and congestion caused by control overhead traffic.

Figure 1 demonstrates the basic modules of streaming transmission system and the streaming procedure in general: When a user requests an audio/video, the Web browser transmits control information to the Web server through the HTTP/TCP protocol, and the web server retrieves the data resource requested by the user and passes it to the Web browser. The Web browser on the client then launches A/V player and initializes media play out. A/V player submits control information to A/V server through a real-time streaming protocol (RTSP), which provides the ability to manipulate playback, fast forward, fast, pause, and record commands. Server uses the RTSP protocol to transmit A/V data to clients.

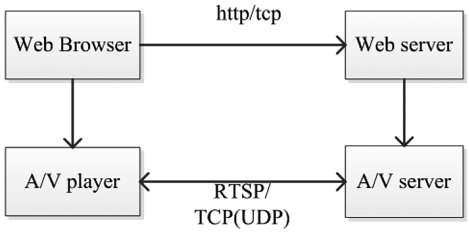


Fig. 1. The basic components of a streaming system

2.2 Characteristics of Media Streaming

The media streaming system includes the following five aspects:

1. Coding tool: Convert multimedia data to streaming media.
2. Streaming data.
3. Server: Storage and control of streaming media data.
4. Network: suitable for multimedia transmission protocols and real-time transmission protocols.
5. Player: for the client to browse the streaming media files.

The main feature of streaming is the transmission of media in the form of data streams, such as streaming TV programs to PCs via the Internet. Different industry systems might implement above modules on the client or server side with different streaming standards in some ways. The two common methods for streaming

transmission are: sequential non-real-time streaming transmission and real-time streaming transmission. Firstly, sequential streaming is transmitted in sequential order, and users can watch online media but not real time. This type of streaming transmission is suitable for short segments, the browser uses standard FTP to communicate with the server through firewall. Secondly, real-time streaming transmission requires stable and sufficient network connection bandwidth, allowing users to view media content in real time. In terms of the playback quality, real-time streaming is often poor due to bandwidth limitation of the connection, the slow transmission rate cause by the congestion. Real-time media transmission requires special transmission protocol, such as real-time streaming protocol (RTSP), sometimes content distribution network (CDN) servers, which allows nearest-to-user media caching. However Real-time Streaming will be affected by firewalls, some real-time content cannot be viewed with a firewall working in action.

2.3 Streaming Media Technology Trends

The streaming transmission have the following advantages over general downloads and playback mechanism. First, the start time delay greatly shortens, the user doesn't have to wait for content download to the hard disk completely, however, fast forwarding will cause pausing or freeze on the browser, users still need to wait. Secondly, the reduced local cache capacity occupation due to the asynchronous transmissions. The limited cache size in the client device might cause unacceptable latency and jitter. To ensure the correct packet transfer sequence, the media data can be kept in a continuous output queue, and will not be randomly abandoned when network congestion occurs. Finally, the latest research and industrial development tend to adopt streaming technology with publish-subscribe based big data processing platforms utilize Hadoop, Spark, Graph Computing and SDN in order to improve transmit efficiency.

3 NDN Networks

3.1 NDN Network Architecture

NDN is evolved from contemporary host-centric network IP architecture to data-centric network architecture, and this conceptually simple shift will have great influence for future design, develop, deploy, and utilization of networks or applications. IP addressing has nature disadvantages in mobile networking scenarios such as mobile ad-hoc networks (MANET), because mobile node enter or leave the coverage area can lead to network outages or intermittent connection. In fact, the application usually only needs to get the correct data and doesn't care which node the data is coming from. IP networking requires infrastructure to support address definition and management, which results in great overhead traffic. As the number of users and number of media contents exponentially increases, total network throughput and system capacity decreases dramatically, eventually paralyses the whole data transmission system, poor in scalability. Previous researches shows the impact of mobility on network performance [2, 3].

NDN architecture is similar to the traditional hourglass IP networking architecture. The major difference is the networking layer as shown in Fig. 2. NDN is data content centric, utilize the data/content name addressing rather than the host IP addressing, each node has a content repository, content store (CS), to manage distributed content caching, reduce time delay and the network overhead traffic. This results in higher efficiency, flexibility and scalability.

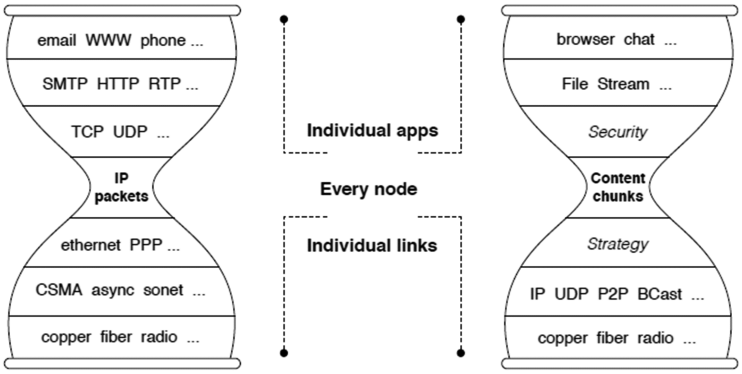


Fig. 2. TCP/IP and NDN architecture [1]

3.2 NDN Packet Types

NDN traffic consists of two types of packets: interest and content packets. Communication in the NDN network is mainly through these two packets as shown in Fig. 3.

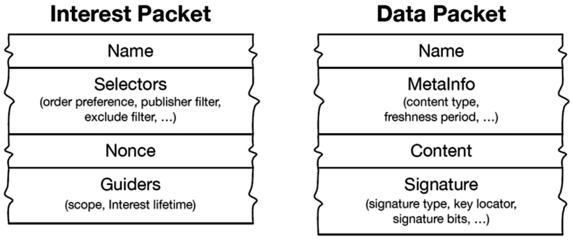


Fig. 3. NDN packets [1]

Interest packet is the users originated request and neighboring nodes forward it hop by hop until the intermediate nodes found in the local content store a matching content packet, or has the information about where requested content is stored. If no matching, intermediate nodes will add hop-count in the interest packet header, then either forward it on or drop the packet if it is exceed its maximum lifetime that measured in hops. Interest package includes a unique name, request data preferences, lifetime, priority sequence, publisher filter, send range, and so on. Contents packet is the data packet that actually carries multimedia data, together includes a name of the data being carried, and

data signature information such as authentication information, a publisher ID that indicates the source of the data, key location, and expiration date etc.

3.3 NDN Routing and Forwarding Mechanism

Each node in the NDN network is capable of forwarding interest packets and maintains three sets of data structures for NDN routing and forwarding mechanism:

(1) content store (CS) is a content repository, which is used to cache content package after the node; (2) forwarding information base (FIB) table is a forward the information library, similar to the routing table in IP network and is used to record and maintain the information of the surrounding nodes; (3) pending information table (PIT) is the backlog request table, which is used to record the information of interest packets forwarded by the node, and the content name in the interest package is the unique id, followed by a series of interface set information shown in Fig. 4.

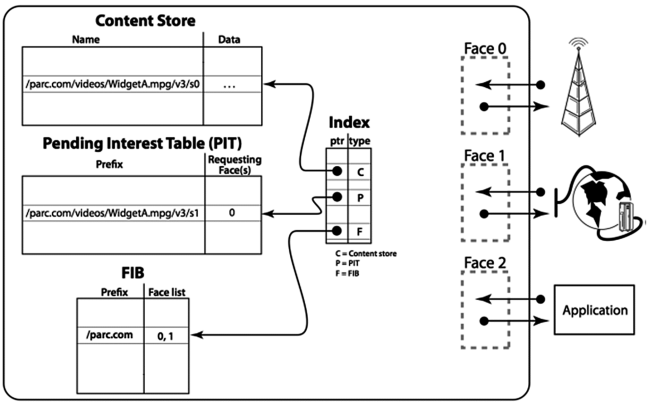


Fig. 4. NDN communication process [1]

The communication process of NDN is shown in Fig. 5. User requests data by sending an interest packet. Interest package contains only the name identity of the data, while users does not need to know what the data address is. When the intermediate router receives the request, it records that the request source information in the FIB table. When the content package is found, the content package is returned to the request node with data naming, signature information, and so on. The same content package does not contain any interface address information, and the interface information that is established in the routing node is returned based on the interest packet forwarding. When the network has multiple requests for the same content package, routing nodes process the first arrived interest requests. If any interest request finds a matching content package which stores information in any routing node, reverse route information will be sent back to the requesting source. Meanwhile, a transmission path is then set for transmission. When the content package is received, the routing node will issue the content package to each node identified by the interface set, known as the Index, and then remove

the interface set record and cache of the content package based on the policy. After the completion of the communication, cache memory of router nodes will be released.

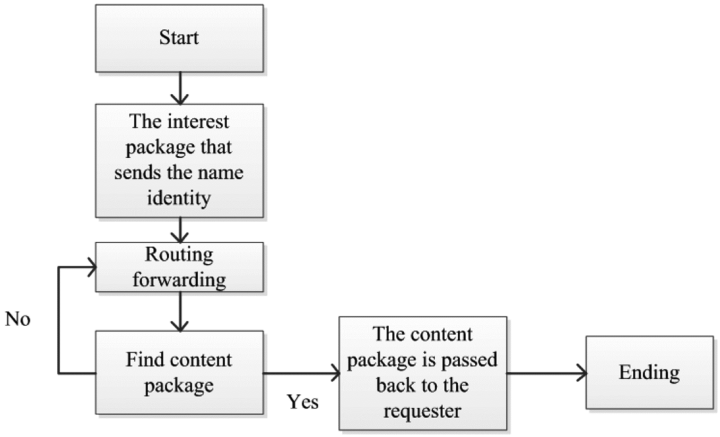


Fig. 5. NDN communication process

In NDN network, a content repository is used for increasing the degree of data sharing in the network and reducing the network traffic. Hence, NDN CS does not clear the cache memory at the end of a reply. In other words, NDN doesn't think that the content package has lost the value, but rather considering possible future demands from the other nodes. This design is a great improvement to the existing network mechanism, which is capable of improving the value of the content package, and reducing network bandwidth consumption at the same time.

4 NDN Based Streaming Experiment for MANET

A mobile ad-hoc network (MANET) network is a self-organized wireless network which can operate without infrastructure support. We design and build an experimental media streaming platform base on the ICN-NDN framework. In the experimental platform, we constructed a simple publish-subscribe (P/S) system to simulate the named data networking mechanism under the MANET scenario. Each interest-query matching event is measured with end-to-end respond time on the user side. Results of three data content catching algorithms: catching along the path, interval catching and least frequently catching. The ICN media streaming system experiment is designed base on NDN mechanism due to the advantages of NDN in a mobile network scenario, is called the NDN streaming media system. This architecture can avoid problems such as the frequent changes network topology, avalanche effects of overhead packet etc. that might affect the performance of media transmission.

4.1 NDN Media Streaming Experiment Frameworks

The NDN mechanism, as shown in Fig. 7, is introduced in the underlying network layer to simulate the communication, distribution and forwarding of media files in a MANET network. The naming mechanism of NDN plays an important supporting role in the acquisition of streaming media, which make easy for sharing of streaming media files. Using the optimized caching algorithm to improve the effectiveness of network cache can greatly improve the search and forward efficiency of network files (Fig. 6).

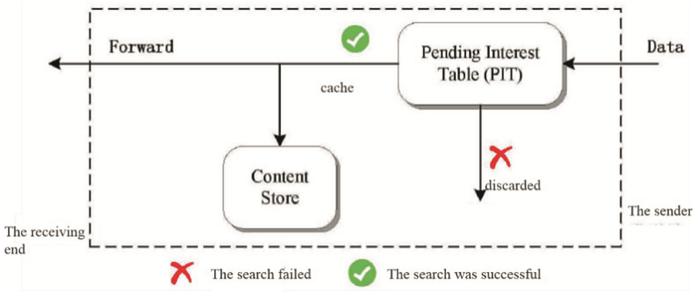


Fig. 6. NDN interest packet forwarding model

The system architecture focus mainly on two layers of construction, the bottom layer is the network layer based on NDN mechanism, and the upper layer is the streaming media player. The network layer and the upper streaming media playback layer communicate through the buffer of the system to realize the loosely coupled system architecture.

The streaming media broadcast layer is divided into the network interaction, framing, decoding, and play modules as shown in Figs. 7 and 8.

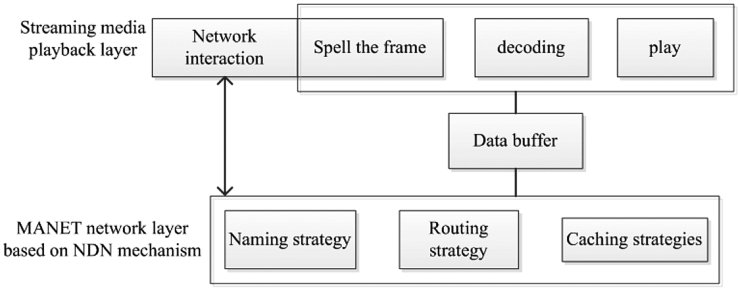


Fig. 7. System structure diagram

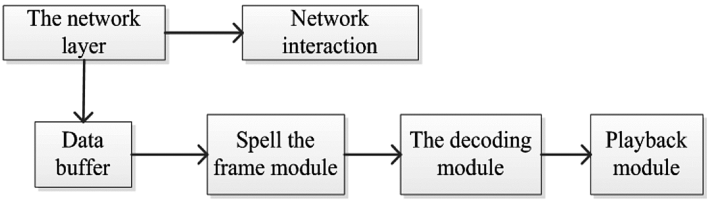


Fig. 8. Data processing procedure of streaming media layer

Network interaction module interacts with the network layer, is responsible for collaborative network streaming media file access, local stream media file sharing, file search, framing, decoding. Frame module, receives the data from the data buffer, which is responsible for the splicing of a certain number of streaming data pieces. Decoding module is provided decoding to the play module for media playing; Buffer module is adopted to store the arrival of data slice information and temporary save streaming media files. This design is for better users play out quality of experience (QoE), improves the continuity of the stream media file and the synchronization of data. A buffer threshold in defined in the system. When the number of files within the buffer reaches this threshold, the media data will be encapsulated to construct frame or decomposed for playback. When a file in the buffer reaches a certain number, buffer data in the buffer is extracted to play media content. The playing speed should be always greater than the network downloading speed otherwise it is considered network congestion, and a re-routing should take place. This can provide certain degree of transmission QoE guarantees for effective and stable media playing.

4.2 NDN Media Streaming Framework Evaluation

In order to verify the effectiveness of the proposed experimental system framework, three kinds of caching algorithms were simulated utilize ndnSIM simulation tool operates on an Ubuntu systems. In this experiment, there are 64 NDN nodes in the MANET

Table 1. Parameter setting of the experiment

Parameters	Example
Total number of nodes	64
The area size	1000 m * 100 m
Data stream type	CBR
The communication distance	250 m
Packet size	512 kbytes
Mobile model	Random waypoint
Maximum mobile speed of nodes	30 m/s
Node duration	8 s
MAC type	Ad-hoc Wifi Mac
The simulation time	100 s

scenario, the communication distance is 250 m between nodes, the simulation time is within 100 s, and the specific parameters are shown in Table 1. The end-to-end response time is measured to indicate the time difference between send out the interest package and receives matching content package. Three cache algorithms are simulated: along path cache, interval caching and least frequently (LF) caching, to compare utilizes the measure of average response times as shown in Fig. 9.

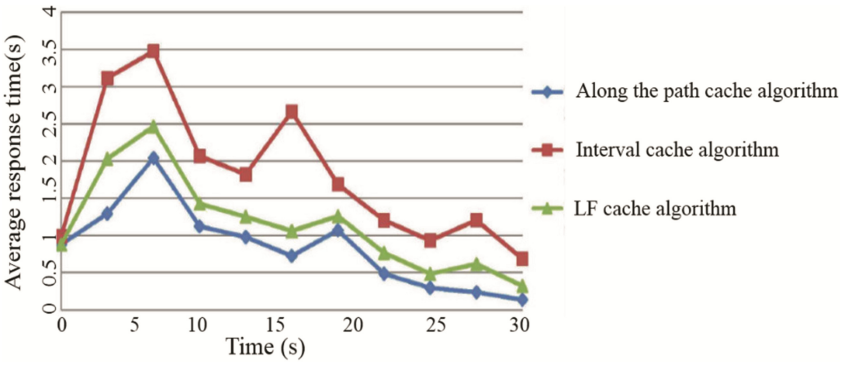


Fig. 9. Average response time

The horizontal axis is simulation duration in time, and the vertical axis is the average response time.

The average response time of all the three cache algorithms is higher at the initial stage of operation, but along the path cache algorithm has the most prominent advantage in response time. The overall response times of the three algorithms are shown in Fig. 10. The along path cache algorithm has the shortest response time compares to the other two algorithms. Both interval caching algorithm and LF have a longer response time. In contrary LF is performing better, the interval caching algorithm has higher average response time.

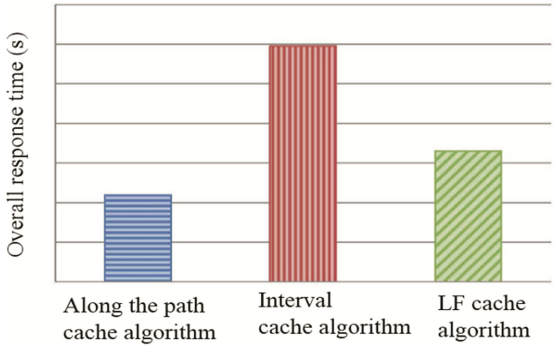


Fig. 10. The overall response time

Result shows that the along the path cache algorithm has the minimum overall response time, the LF caching algorithm takes longer, the interval caching algorithm has the longest response time. The developed prototype system based on the NDN mechanism of streaming media can simulate the basic communication functions. Users can play local video as well as sharing local video with other users. Results in Figs. 9 and 10 demonstrate that simulation model and algorithms meet the hypothesis, and the network assumption and cache environment is perfectly functional. Users can find other media contents from other user nodes in the network. This study laid a foundation for future research on ICN/NDN media streaming in a mobile communication scenario. Further experiment is needed in terms of: improve the cloud computation paradigm by compares global contents graph computing and tree computing algorithms; investigate methods to construct universal streaming big data store and transmit architecture that combines both the conventional batch and stream data processing paradigm into resilient distributed data (RDD) universal streaming process paradigm; how to quantitatively measure BDP efficiency in heterogeneous network environment to improve quality of experience (QoE) etc.

5 Conclusion

This paper is the first stage experimental attempt for media streaming big data processing (BDP) base on the information centric networking (ICN) framework. In the experimental platform, we constructed a simple P/S system to simulate the NDN mechanism under the mobile ad-hoc network (MANET) scenario. Each interest-query matching event is measured with end-to-end responds time on the user side. Results of three data content catching algorithms: catching along the path, interval catching and least frequently catching (LF), verified hypothesis that the cache algorithm has strong impact on content matching efficiency, maintain optimized catching can introduce smaller end-to-end response time as well as reduce network overhead at the same time. Further experiment is needed in terms of: improve the cloud computation paradigm by compares global contents graph computing and tree computing algorithms; investigate methods to construct universal streaming big data store and transmit architecture that combines both the conventional batch and stream data processing paradigm into resilient distributed data (RDD) universal streaming process paradigm; how to quantitatively measure BDP efficiency in heterogeneous network environment to improve quality of experience (QoE) etc.

Acknowledgements. We would like to present our appreciation for the support from the National Science Foundation of China Guang dong project U1301252, Science and Technology Innovation Commission Foundation of Shen Zhen project JCYJ20160608151239996 and JCYJ 20170307114301790.

References

1. Zhang, L., Estrin, D., Burke, J., et al.: Named data networking (NDN) project. Transp. Res. Rec. J. Transp. Res. Board **1892**(1), 227–234 (2014)
2. Lu, Y., Mitchell, P.D., Grace, D., Pearce, D.A.J.: Performance evaluation of minimum impact routing for multi-hop wireless ad hoc network. In: Wireless Personal Multimedia Communications Conference, pp. 22–26 (2004)
3. Lu, Y., Grace, D., Mitchell, P.D.: Capacity evaluation of a multi-hop wireless ad hoc network using minimum impact routing. In: International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1–4. IEEE (2006)
4. Gibbens, M., Gniady, C., Ye, L., et al.: Hadoop on named data networking: experience and results. Proc. ACM Meas. Anal. Comput. Syst. **1**(1), 2 (2017)
5. Zhang, W.-W., He, J.-F.: The application of streaming media technology in transmission of foreign language teaching resources. J. Changchun Norm. Univ. **34**(10), 45–49 (2015)
6. Wang, J.-J.: Design of national music video management system based on streaming media technology. Electron. Des. Eng. **24**(18), 149–151 (2016)
7. Li, X., Wang, K.-F.: A flow media transmission method based on Mesh network structure. Mod. Electron. Tech. **40**(6), 62–64 (2017)
8. Xu, W.: Design and implementation of QoE analysis model based on HLS streaming video live. J. Xi'an Aeronaut. Univ. **35**(3), 66–68 (2017)
9. You, Z.-G.: Potential based streaming media with content caching for ICN. Comput. Knowl. Technol. **11**(6), 213–215 (2015)
10. Ming, T., Sen, Y.: Hybrid mechanism based on edge caching and tracing route supporting streaming media services in information centric networks. J. Inf. Eng. Univ. **17**(6), 735–742 (2016)
11. Qiu, M., Zhong, M., Li, J., et al.: Phase-change memory optimization for green cloud with genetic algorithm. IEEE Trans. Comput. **64**(12), 3528–3540 (2015)
12. Li, Y., Dai, W., Ming, Z., et al.: Privacy protection for preventing data over-collection in smart city. IEEE Trans. Comput. **65**(5), 1339–1350 (2016)
13. Zhu, X., Qin, X., Qiu, M.: QoS-aware fault-tolerant scheduling for real-time tasks on heterogeneous clusters. IEEE Trans. Comput. **60**(6), 800–812 (2011)
14. Qiu, M., Chen, Z., Ming, Z., et al.: Energy-aware data allocation with hybrid memory for mobile cloud systems. IEEE Syst. J. **PP**(99), 1–10 (2014)
15. Gai, K., Qiu, M., Zhao, H.: Energy-aware task assignment for mobile cyber-enabled applications in heterogeneous cloud computing. J. Parallel Distrib. Comput. **111**, 126–135 (2017)
16. Gai, K., Qiu, M., Zhao, H., et al.: Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. J. Netw. Comput. Appl. **59**(C), 46–54 (2016)

An Implementation of Content-Based Pub/Sub System via Stream Computation

Lei Huang[1], Li Liu[2], Jiayu Chen[1], and Kai Lei[1(✉)]

[1] Shenzhen Key Lab for Cloud Computing Technology and Applications, School of Electronic and Computer Engineering (SECE), Institute of Big Data Technologies, Peking University, Shenzhen 518055, People's Republic of China
lhuang@pku.edu.cn, qxlong_chen@163.com, leik@pkusz.edu.cn

[2] Institute of Education, Tsinghua University, Beijing 100084, People's Republic of China
li-liu16@mails.tsinghua.edu.cn

Abstract. The sheer volume of data delivered via the Internet requires a more flexible and powerful communication model. As an expressive loosely-coupled asynchronous messaging model, Publish-Subscribe (Pub/Sub) system has been widely used. Traditional topic-based Pub/Sub system fails to understand the information of messages delivered, all messages must be previously classified into a set of topics. Content-based Pub/Sub system can dynamically choose subscribers for each message by its metadata. Existing distributed Pub/Sub systems are built on the overlay network consists of message brokers, which can adapt to heterogeneous network but inevitably impairs performance. In this paper, we designed a novel centralized tiered content-based Pub/Sub system with a four-layer architecture. In access layer, a customized naming strategy is proposed to achieve high availability. Internal message routing is finished in routing layer and sharding scheme is used to lower routing overhead. In computation layer, a two-step streaming computation model is used to boost the performance. In storage layer we adopt column-oriented database HBase for persistence. A set of comprehensive experiments were conduct to verify that our system achieve excellent performance, linear scalability and high availability.

Keywords: Content-based Pub/Sub · Messaging system · Stream computation

1 Introduction

With the explosion of Internet applications, the data delivered via the Internet is drastically increasing, including social network notifications, E-Commercial transactions and sensor data accumulated by IoT devices. The requirement for efficient message delivery is becoming urgent. The communication characteristics of these applications can be summarized as asynchronous, anonymous, one-to-many communication.

This work has been financially supported by Shenzhen Key Fundamental Research Projects (Grant No. JCYJ20170412151008290, JCYJ20170306091556329, JCYJ20170412150946024).

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 344–353, 2018.
https://doi.org/10.1007/978-3-319-73830-7_34

Publish/Subscribe (Pub/Sub) system is such a dynamic, loosely-coupled, flexible communication system that interconnects information producers and consumers in a distributed environment.

Depends on their filtering methods, Pub/Sub systems can be classified into two categories: topic-based and content-based. In topic-based Pub/Sub systems, subscribers must previously subscribe to some topic(s) which is usually a string, when publishers publish a message or event of that topic, the system will multicast that message to all subscribers who subscribed that topic. Topic-based Pub/Sub system is simple and intuitive, but it has only one dimension-the topic, which greatly limits its expressiveness. To address the challenges mentioned above, content-based Pub/Sub system was designed [1]. These systems support an attribute space of multiple dimensions and an event is defined as a tuple containing one or several attributes. So subscribers can describe their subscriptions as a predicate over attributes.

In this paper, we implemented a tiered content-based Pub/Sub system and optimized the bottleneck of each layer using a bunch of techniques including caching, sharding, etc. The rest of this paper is structured as follows: Sect. 2 describes related work. Section 3 introduces the implementation and optimization of our system. Experiments and evaluations are presented in Sect. 4. Finally, we conclude this paper in Sect. 5.

2 Related Work

Existing research on Pub/Sub systems are mainly about broker-based Pub/Sub systems, where message brokers are a group of dedicated servers handling subscriptions. Gryphon [2] and MEDYM [3] are early Pub/Sub systems based on message brokers. Every brokers should store all subscriptions, which limits their scalability. Other systems, like SIENA [4], organize those message brokers into an overlay network, all events and subscriptions are forwarded in this overlay network, which enhances scalability but undermines performance, especially when system grows, a large routing table must be maintained to decide to which broker the message should be forwarded.

3 Implementation and Optimizations

3.1 Overview

The architecture of our system is hierarchical and consists of four layers that are access layer, routing layer, computing layer and storage layer (Fig. 1).

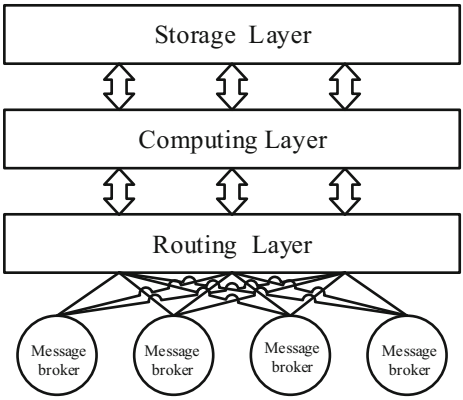


Fig. 1. System architecture

Access layer is made up of a cluster of message brokers and maintains TCP long connections with clients including publishers and subscribers. Routing layer is responsible for retrieving event matching result from computing layer and forwarding event to corresponding subscribers. Computing layer decides to which subscriber the event should be forwarded and then notify that subscriber via routing layer. We choose Apache Storm [5] as the stream computing framework. In the rest of this section, we will introduce the implementation and optimization details of our system.

3.2 Access Layer

Access layer handles three kinds of messages that are CONNECT, SUBSCRIBE and PUBLISH. CONNECT is used by clients (publishers and subscribers) to establish a session with server and periodically keep-alive with server via heart-beat messages.

Figure 2 gives a detailed explanation of how CONNECT message is handled. The processing of SUBSCRIBE message has no difference with CONNECT message, so we may just skip that.

Protocol
We compared HTTP and MQTT and we finally chose MQTT as the communication protocol. HTTP is simple but the string-based nature makes it less efficient. MQTT [6] is a lightweight machine-to-machine (M2M) protocol that perfectly supports publish/subscribe semantic.

Message Brokers
In order to accommodate a large number of message brokers in one host, the message broker is designed to be fast and dedicated to single responsibility. It simply accepts connection requests, decodes messages and submits matching tasks to computing layer. It does not care how events are matched. We choose Netty [7], an asynchronous C/S framework as the network communication framework. In order to handle a high volume

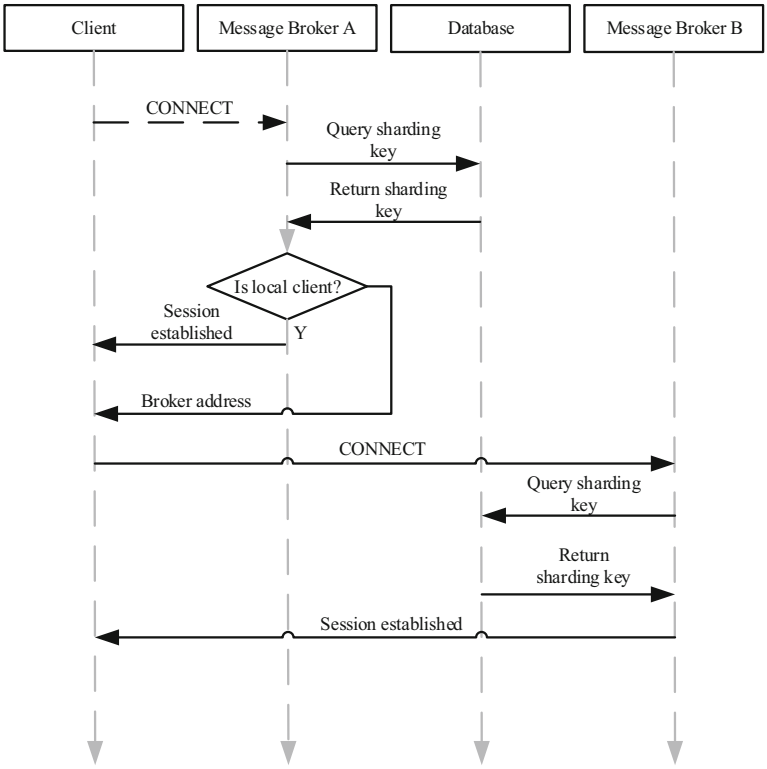


Fig. 2. CONNECT handle process

of connections, we employ an RCP-based strategy to handle inter-broker congestion control [8].

3.3 Fast Event Matching

Computing layer is responsible for event matching, deciding which subscriber will be interested in the incoming events. In order to achieve fast event matching, we adopt a two-phase computing model called FlatMap-Filter.

FlatMap-Filter

From we can observe, it's very common that subscriptions may overlap with each other. For example, subscription $S_1 = \{0 < x < 10\}$ and $S_2 = \{-2 < x < 10\}$ can be merged into $S' = \{-2 < x < 10\}$. Using that kind of aggregation can reduce the scale of event matching.

There are three kinds of nodes in FlatMap-Filter model: matcher decides whether the event matches its criterion, dispatcher dispatches matching tasks to matchers and collector collects matching results from matchers (Fig. 3).

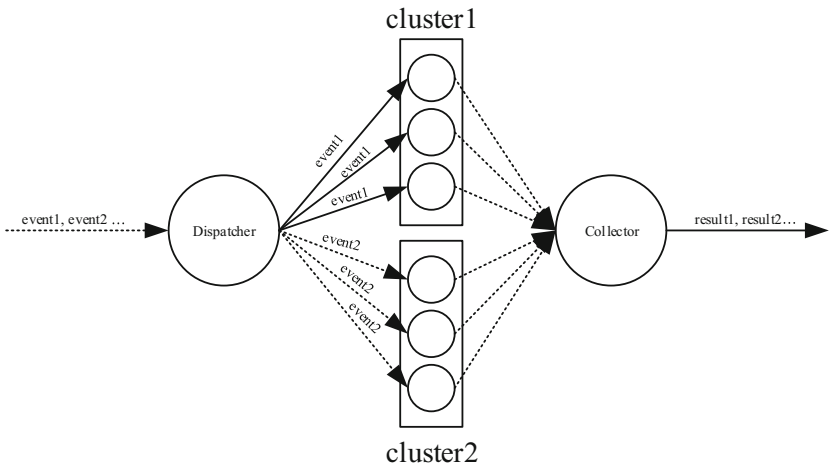


Fig. 3. FlatMap-Filter model

Like Google MapReduce, FlatMap-Filter first dispatches the event to be matched to a group of aggregated matcher nodes (called cluster) that may (or may not completely) be interested. It's called FlatMap, inspired by functional programming. That process can downsize the matching scale to a large extent. After FlatMap, matchers directly compute the attributes of incoming events against its own criterion, which is called *filtering*.

FlatMap Algorithm

The algorithm of FlatMap phase is hot-pluggable, that is it can be substitute according to events distribution. Many algorithms have been proposed to handle such clustering problem, like K-means [9], attribute-space partitioning [10] and R-Tree [11]. In our system we simply choose attribute space partitioning for FlatMap process to aggregate events and subscriptions.

3.4 Message Routing

After event matching, the result should be forwarded to corresponding subscribers. Finding subscriber of interest rapidly is what routing layer addresses. If we simply store the relationship of subscribers and their message brokers into database, the system needs to query database for every event. The cost is unacceptable even if caching is employed to bypass some queries.

We use sharding mechanism combined with consistent hashing to bypass database query while avoid data distribution hotspot and keep the mapping relationship consistent as system scales. Here we borrow the word *sharding* from database area to indicate that the brokers are logically split into partitions. If we want to know the broker of some client, we first calculate the partition that client belongs to. When clients connect for the first time, the system calculates a P-Key (partition key) for it.

3.5 Storage Engine

Storage is usually the performance bottleneck in most systems because it involves IO operations.

Considering the scenario, we use Apache HBase as the storage engine. HBase is a column-based database widely used in Hadoop ecosystem. Compared to RMDBS like MySQL/PostgreSQL/Oracle DB and in-memory storage engine like Redis, HBase can add or remove columns dynamically. That is very important because in our system subscriptions and events may have different number of attributes, which will result into waste of storage space if we employ RMDBS. More importantly, based on HDFS, HBase supports sharding by nature. When system scales, HBase can automatically scale out to a large database cluster and distribute data evenly to HDFS nodes.

In order to record the complete status of system, four tables should be stored inside HBase: incoming events tables, subscription table, matching result table and unacknowledged events table. The last one unacknowledged events table stores those events who have been matched but not yet pushed to subscribers because they somehow have lost connection with server.

Table Schema

We pick subscription table to demonstrate the design of table schema. Table 1 shows the schema of subscription table in HBase.

Table 1. Subscription table schema

Row key	Column families					
	Integer			Float		
md5(uid):timestamp:sid	Attr_1	Attr_2	Attr_2	Attr_1	Attr_2	Attr_3

As can be seen in the table, subscription table has 2 column families called *Integer* and *Float*, inside each column family is the attribute name. In the row key, *md5(uid)* refers to the digest value of subscriber id and *sid* refers to the session id of that subscriber.

4 Experiments

In this section we will demonstrate the experiment results of our system. The experiment involves two part: standalone mode performance test and distributed deployment performance test in cloud computing environment. Both tests are carried out on public cloud instances with 4 Intel Xeon E5-2600 v4 cores, 4 GB RAM. Instances are connected with each other through 2 Gb switch. The operating system is Ubuntu 16.04 LTS, with OpenJDK 8u144 and Apache Storm 1.0.5 installed.

4.1 Standalone Mode

In standalone mode we want to get the squeeze the last bit of performance, so we have to tune some TCP/IP protocol options inside Linux kernel. Table 2 lists the detailed tuning options and their values.

Table 2. Kernel tuning options

Option	Value
net.ipv4.tcp_keepalive_intvl	15
net.ipv4.tcp_keepalive_time	600
net.ipv4.tcp_keepalive_probes	3
net.ipv4.ip_local_port_range	1024 65535
net.ipv4.tcp_syn_retries	1
net.ipv4.tcp_synack_retries	1
net.ipv4.tcp_retries2	5
net.ipv4.tcp_fin_timeout	2
net.ipv4.tcp_max_tw_buckets	36000
net.ipv4.tcp_tw_recycle	1
net.ipv4.tcp_tw_reuse	1
net.ipv4.tcp_max_orphans	32768
net.ipv4.tcp_syncookies	1
net.ipv4.tcp_max_syn_backlog	16348
net.ipv4.tcp_wmem	8192 131072 16777216
net.ipv4.tcp_rmem	32768 131072 16777216
net.ipv4.tcp_mem	786432 1048576 1572864
net.core.somaxconn	16384
net.core.netdev_max_backlog	16384

In standalone mode, message brokers, computing layer nodes and HBase storage engine all run on the same server and the test clients, including subscribers and publishers are distributed in 5 different servers.

Figure 4 shows system throughput and latency as the request per second (QPS) grows. When QPS reaches 15000, the latency drastically grows so we assume that 15000 is the appropriate workload. As Fig. 5 shows, at that workload, most requests can be finished within 600 ms, and the average process time is 342 ms.

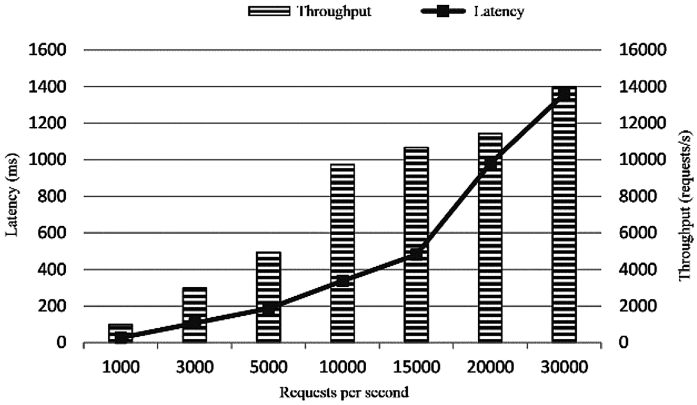


Fig. 4. Latency and throughput as request per second grows

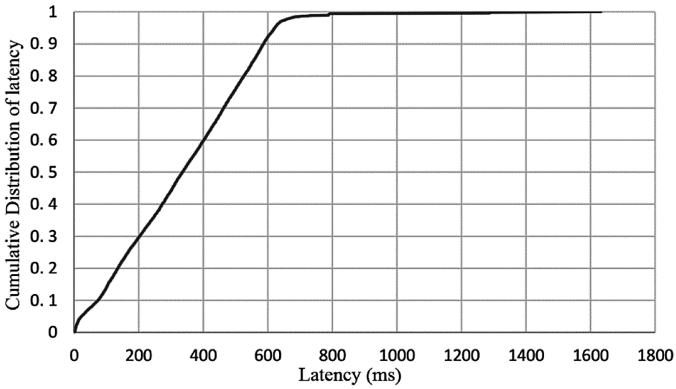


Fig. 5. Cumulative distribution of latency when 15000 requests arrive per second

4.2 Distributed Cluster on Cloud

When we test in distributed deployment mode, we care more about the scalability of system, that is, whether our system can achieve a linear performance growth as we add more nodes into the system. All tests have been conducted when there are 1,000,000 subscriptions registered.

Figure 6 shows that as we add nodes into system, the throughput has an almost linear growth while the latency increases only a bit.

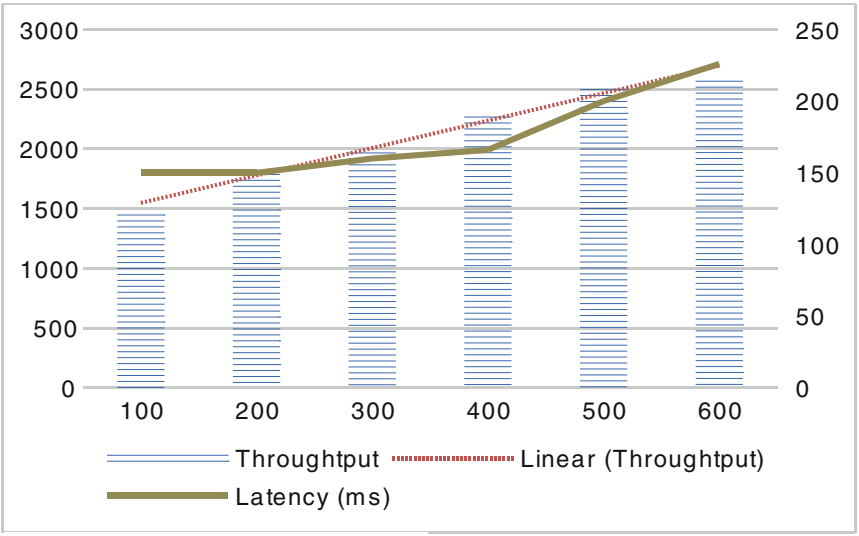


Fig. 6. System latency and throughput change as we add nodes

5 Conclusion

In this paper, we designed and implemented a tiered content-based Pub/Sub system that is meant to deal with a high volume of connections. We use asynchronous framework in the access layer to achieve high performance and maintain a large number of connections simultaneously. In computing layer, we employ Apache Storm to implement a 2-phase computing model which can speed the matching process up while keep great scalability. In storage layer we use column-based HBase as the storage engine, which has sharding ability by nature. In routing layer, we introduce consistent hashing mechanism to assign partition key to clients while as the system scales the mapping relation remains consistent. With a set of experiments, we can see that our system has a good performance when deployed standalone, while as we add more nodes (message brokers and computing nodes) into it, the throughput grows almost linearly and latency remains steady.

References

1. Banavar, G., Chandra, T., Mukherjee, B., Nagarajarao, J.: An efficient multicast protocol for content-based publish-subscribe systems. In: Proceedings of IEEE International Conference on Distributed Computing Systems, 1999, pp. 262–272 (1999)
2. Strom, R., Banavar, G., Chandra, T., Kaplan, M., Miller, K., Mukherjee, B., et al.: Gryphon: an information flow based approach to message brokering. Computer Science arXiv:cs/9810019 (1998)

3. Cao, F., Singh, J.P.: MEDYM: match-early with dynamic multicast for content-based publish-subscribe networks. In: Alonso, G. (ed.) Middleware 2005. LNCS, vol. 3790, pp. 292–313. Springer, Heidelberg (2005). https://doi.org/10.1007/11587552_15
4. Carzaniga, A., Rosenblum, D.S., Wolf, A.L.: Design and evaluation of a wide-area event notification service. ACM Trans. Comput. Syst. **19**(3), 283–334 (2003)
5. Apache Storm. http://storm.apache.org. Accessed 08 Oct 2017
6. MQTT. http://mqtt.org. Accessed 08 Oct 2017
7. Netty. https://netty.io. Accessed 08 Oct 2017
8. Lei, K., Hou, C., Li, L., Xu, K.: A RCP-based congestion control protocol in named data networking. In: 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 538–541. IEEE (2015)
9. Wong, T., Katz, R., Mccanne, S.: An evaluation of preference clustering in large-scale multicast applications. In: Proceedings of IEEE Nineteenth Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2000, vol. 2, pp. 451–460. IEEE (2000)
10. Wang, Y.M., Qiu, L., Achlioptas, D., Das, G., Larson, P., Wang, H.J.: Subscription partitioning and routing in content-based publish/subscribe systems. In: Disc Proceedings of International Symposium on Distributed Computing (2002)
11. Beckmann, N., Kriegel, H.P., Schneider, R., Seeger, B.: The R*-tree: an efficient and robust access method for points and rectangles. ACM Sigmod Rec. **19**(2), 22–331 (1990)

Security Message Broadcast Mechanism Research in Vehicular Network

Yanlin Zhao, Kena Dong, and Xiumei Fan(✉)

Faculty of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710048, China
xmfan@xaut.edu.cn

Abstract. Due to the rapid movement of the vehicle as a network node in the vehicle ad hoc network, the network topology changes frequently, and the communication link between the nodes is unstable, which leads directly to the fact that the security message cannot be transmitted in a timely and reliable manner. Based on these problems and practical requirements, this paper proposes a security message broadcast mechanism SMLP based on location prediction. The algorithm is divided into three steps. First, the future location information is predicted according to the location and direction of the neighbor node, and then the optimal relay node is selected. Finally, the relay node broadcasts the security message. In the stage of selecting the optimal relay node, the algorithm overcomes the shortcoming of the nearest neighbor node of the destination node as the next hop forwarding node, and further considers the continuous connection time between the nodes, direction coefficient and distance coefficient of node, and the stability coefficient is defined by the combination of these factors. SMLP chooses the neighbor node with the largest stability coefficient as the next hop relay node. The simulation results show that the SMLP routing protocol can reduce the packet loss rate and end-to-end delay in the packet transmission process to a certain extent compared with the GPSR routing protocol. It can be well applied to the transmission of security messages in the vehicle ad hoc network.

Keywords: IOV · Security message · Relay node · SMLP routing protocol

1 Introduction

Vehicles not only bring travel convenience, but also a lot of trouble to people, such as traffic congestion and even traffic accidents. Even if the relevant departments have adopted many measures, such as widening the road, the construction of viaducts, but the effect is not significant1. And thus road congestion, and even traffic accidents happened frequently, such as rear-end, positive collision, speeding and so on.

X. Fan, Ph.D. Prof., Main research interesting: Vehicular Networks, Mobile Internet, Delay tolerant networks.

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 354–364, 2018.
https://doi.org/10.1007/978-3-319-73830-7_35

IOV is a dynamic mobile communication system, including information exchange between the vehicles, the roadside equipment and people, and communicate with the public network2. It can share information between vehicles and vehicles, vehicles and roadside units, vehicles and people, and process information in the network information platform, and provide relevant information services according to the different needs. There are two kinds of messages in IOV, security message and non-security message. The security messages include collision warning messages, overtaking lane messages, emergency braking messages and so on. Non-security messages refer to some entertainment messages that are provided to improve the comfort of the drivers and the passengers.

If the destination node is not within the communication range of the source node, it needs to select the relay node to relay and send the message to the further destination area. By investigating the broadcast mechanism of security messages in IOV, it is possible to effectively reduce the occurrence of accidents or avoid secondary accidents by sending road conditions or emergency messages through the relay nodes as quickly and reliably as possible to the affected vehicles, so there is a far-reaching significance.

2 Research on Security Message Broadcasting Mechanism

With the further demand of the security emergency broadcast mechanism in IOV, many domestic and foreign experts and scholars have put forward their own ideas and algorithms for this topic. In [3], a broadcast scheme is proposed to improve the transmission efficiency. The main idea is to determine the transmission slot of the emergency message according to the SNR and the node location. This algorithm shortens the channel access time and improves the propagation efficiency of the messages. Randomly selects a time slot to send the security message in the contention window, which reduces the probability of collision. In [4], a theoretical model framework based on the density of the vehicle is chosen, and no specific algorithm flow is designed. However, a new idea is provided for the broadcast algorithm of the security message. For the vehicle density and network topology in different circumstances, with different frequencies and power to send security messages, will not cause broadcast storm. However, in this paper, there is no specific network congestion standard, so it is necessary to find the appropriate congestion detection standard and power control algorithm, combined with the actual algorithm and theory simulation experiments to verify.

For the intersection and the straight road, the vehicle distribution structure is different, the algorithm used for message transmission also varies according to the regional characteristics. In [5], a routing protocol based on adaptive delay is proposed in urban scene. Its main characteristic is that the forwarding strategy is divided into greedy mode and intersection forwarding mode according to the road condition, and the message is preferentially transmitted to the vehicle nodes which in the intersection. The literature [6, 7] also divides the broadcast into two modes, the directed broadcast protocol on the straight road and the multi-hop broadcasting protocol based on the phase information at the intersection, compared to the traditional broadcast protocol, this paper

selects the relay node considering the distance and speed in addition to the transmission power and transmission rate.

In addition to the above methods, there are many experts and scholars based on the existing routing protocol, especially with the rapid development of global positioning technology, making the location-based routing protocol has been widely used, the most typical of which is the GPSR routing protocol [8, 9], this protocol has two forwarding mode, greedy forwarding mode and peripheral forwarding mode. The two forwarding modes complement each other to complete the forwarding process of the message.

When the source node needs to send the packet to the destination node and the destination node is not within the transmission range of the source node, the GPSR first adopts the greedy forwarding mode, that is, the node closest to the destination node is selected as the next hop in the neighbor node of the source node, and the process is executed until the data packet is transmitted to the destination node. But when all the neighbor nodes of the source node are far from the destination node, the greedy forwarding will fall into an infinite loop, which we call the local optimal situation [10].

The advantage of GPSR protocol is that it can avoid the establishment, maintenance and storage of routing tables in nodes, and rely on neighbor nodes to make path decisions. However, due to the high mobility of the vehicle nodes, there are the following problems: (1) the link is unstable, because the GPSR routing protocol selects the nearest node from the destination node as forwarding node, which is the border node of source node, if the border node moves faster, the link will be disconnected, resulting in packet loss and relatively large delay. (2) It does not consider the speed and direction of the relay node, GPSR routing protocol only considers the linear distance factor, and does not take into account the direction and speed of the node.

3 Security Message Broadcast Mechanism Based on the Location Prediction

The purpose of this paper is to find out how the source vehicle sends the security message to all nodes in the affected area in time when an emergency occurs on the road. This process is carried out in two steps. First, the source vehicle broadcasts emergency messages and then selects the appropriate relay node to continue forwarding the security emergency message to the farther vehicle node group. The selection process of relay nodes is the main content of this research. Aiming at the shortcomings of GPSR routing protocol, this paper proposes a new message broadcast protocol based on GPSR protocol, which is security message broadcast mechanism based on the location prediction (SMLP), and through the simulation platform to verify in conclusion.

3.1 Protocol Ideas

The SMLP protocol is based on the position prediction. Therefore, before the relay node is selected, the first step is to predict the position of the neighbor node t by the speed and direction information of the neighbor node. This can prevent the selected relay node

from deviating destination node, or increased packet loss and other issues because of the relay out of coverage of the current node.

SMLP is the improvement of greedy mode, so the forwarding mode should be selected, the current node judges whether have neighbor nodes which closer to the destination node than itself in the neighbor nodes. If any, then enter the greedy forwarding mode, otherwise it is the peripheral forwarding mode.

After determining that the forwarding mode is greedy forwarding mode, the algorithm in this mode is executed as follows: First, the selection range of the relay node is limited to a certain area according to the distance coefficient, which will exclude a series of non-compliant neighbor node, these nodes will increase the protocol delay and hops. Then the node connection time and direction coefficient are weighted in proportion to form a stable coefficient, and the optimal relay node is selected according to the stability coefficient.

3.2 Position Prediction

Predicting the motion path or position of each neighbor node can save a lot of time for faster and more accurate selection of the optimal relay node. Assuming that the position coordinates of the neighbor node i are (x_i, y_i) at time t_S, and (X_{ni}, y_{ni}) at time ($t_S + T$). After calculating the velocity vi and direction θ_i information of the node i according to the formulas (1) and (2), the new coordinate information is updated to the neighbor list.

$$\frac{\sqrt{\left(y_{ni} - y_i\right)^2 + \left(x_{ni} - x_i\right)^2}}{T} \tag{1}$$

$$\tan^{-1} \frac{y_{\mathrm{ni}} - y_{\mathrm{i}}}{x_{\mathrm{ni}} - x_{\mathrm{i}}}, x_{\mathrm{ni}} \neq x_{\mathrm{i}} \tag{2}$$

$$\begin{cases} x_{ti} = x_{ni} + v_i \cdot t \cdot \cos\theta_i \\ y_{ti} = y_{ni} + v_i \cdot t \cdot \sin\theta_i \end{cases} \tag{3}$$

After obtaining the velocity and direction information, the node S can use formula (3) to get the position information (X_{ti}, y_{ti}) of all its neighbors after t milliseconds. Node S must have above calculation for each neighbor, with future location information, we can lay the foundation for the next step relay selection.

3.3 Relay Selection

In order to improve the lack of GPSR routing protocol, we redefine the selection algorithm of relay nodes from the following aspects.

3.3.1 Node Connectivity Time

In the step of relay selection, we should select the neighbor node which in the source node communication within a long time as a relay node. It is considered that the node

connection time Δt as one of the factors for selecting the next hop node. If a vehicle node accumulating in the communication range of the source node is longer, it indicates that the vehicle node can maintain a good communication with the source node for a long time, so the communication situation between source node and the node is relatively stable, then the node can be selected as a relay node a candidate node.

Assuming that the location of the source node $S(X_S, Y_S)$ and the neighbor node $F(X_F, Y_F)$ are shown in Fig. 1, the position of the source node S and the neighbor node F is located at $S'(X'_S, Y'_S)$ and $F'(X'_F, Y'_F)$ after Δt millisecond, and then F will leave the communication range of the source node S, that is, the distance between S′ and F′ is the communication range R of the source node S at this time. In this process, Δt is the connection time between the source node and the neighbor node.

$$\begin{aligned} R^2 &= D(S', F') = \left(X_{S'} - X_{F'}\right)^2 + \left(Y_{S'} - Y_{F'}\right)^2 \\ &= \left(\left(X_S + V_{SX} \cdot \Delta t\right) - \left(X_F + V_{FX} \cdot \Delta t\right)\right)^2 + \left(\left(Y_S + V_{SY} \cdot \Delta t\right) - \left(Y_F + V_{FY} \cdot \Delta t\right)\right)^2 \\ &= a \cdot \Delta t^2 + b \cdot \Delta t + c \end{aligned} \tag{4}$$

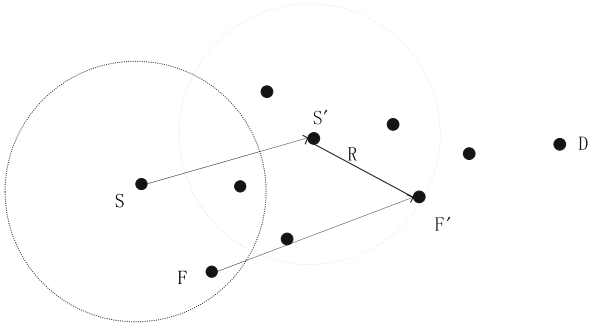


Fig. 1. Node connection time

Where the values of a, b, and c are:

$$\begin{aligned} \mathrm{a} &= \left(V_{SX} - V_{FX}\right)^2 + \left(V_{SY} - V_{FY}\right)^2 \\ b &= 2\left[\left(V_{SX} - V_{FX}\right)\left(X_S - X_F\right) + \left(V_{SY} - V_{FY}\right)\left(Y_S - Y_F\right)\right] \\ c &= \left(X_S - X_F\right)^2 + \left(Y_S - Y_F\right)^2 \end{aligned}$$

The value of the node connection time Δt can be obtained by solving Eq. (4).

3.3.2 Distance Coefficient

In order to minimize the number of hops from source node to the destination node and take into account the characteristics of the algorithm in the original GPSR protocol, a distance coefficient μ is set. The range of this coefficient is [0, 1], the relay node selection range is limited to μR and R (called stable area), as shown in the shaded part of Fig. 2.

If there is no node between μR and R, the value of μ is gradually decreased until a node satisfying the condition.

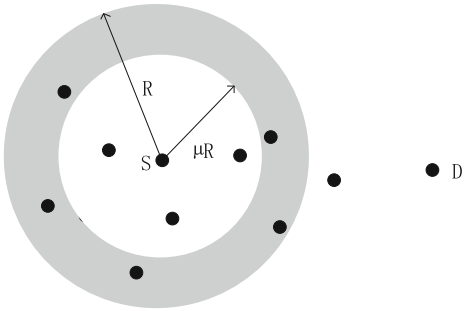


Fig. 2. Defines the selection range of the relay node

3.3.3 Direction Factor

In order to the packet can reach the destination node more reliably, we need to transfer the packet along the direction of the destination node. In the process of node location prediction, we have calculated the motion direction of the neighbor node, and we also need to calculate the direction of the source node $S(X_S, Y_S)$ to the destination node $D(X_D, Y_D)$, as shown in Eq. (5). And the angle θ between the two directions is calculated, as shown in Eq. (6). The smaller the value of θ, the greater the probability that the neighbor node will forward the security message to the destination node. So θ can as another factor that we consider to select the relay node.

$$\theta_{SD} = \tan^{-1} \frac{y_D - y_S}{x_D - x_S} \tag{5}$$

$$\theta = \left| \theta_i - \theta_{SD} \right| \tag{6}$$

3.3.4 Stability Coefficient

In order to make the connectivity time between source node and the relay node for a long time and satisfy the minimum number of hops of the entire route, and the packets reach the destination node reliably, we need to consider the three factors of node connectivity time, distance coefficient and direction coefficient, the specific process is as follows:

1. Calculate the value of μR to determine the stable region [μR, R];
2. Calculate the distance set FD for all neighbor nodes and destination nodes;
3. Find node set SD in the FD which is in the stable area;
4. Calculate the node connectivity time Δt and the direction coefficient θ of each neighbor node, and apply them to weighting processing in Eq. (7).

$$P_{SF} = \alpha \cdot \Delta t + \beta(1 - \frac{\theta}{2\pi}) \tag{7}$$

Where α and β are the weight coefficients of node connectivity time and direction factors, respectively. PSF is called the stability coefficient, the larger the value, indicating that the neighbor node can link with the source node in a longer time, and has a closer direction with destination node. It can effectively improve the performance of the entire network.

4 Simulation Experiments and Result Analysis

4.1 Simulation Modeling

In this paper, the traffic simulator VanetMobiSim and the network performance simulation software NS2 are used to build the joint simulation platform, and the proposed SMLP protocol is simulated. Finally, the simulation results are analyzed. The parameter settings are shown in Tables 1 and 2, respectively.

Table 1. Parameter settings in VanetMobiSim

Content	Value
Simulation area	2.1 km * 2.1 km
Simulation time	200 s
The number of vehicle nodes	30–70
The speed of vehicle nodes	5 m/s–30 m/s
Traffic lights	10
Road speed limit	30 m/s

Table 2. NS2 simulation parameters

Content	Value
Simulation area	2.1 km * 2.1 km
Vehicle movement mode	VanetMobiSim
Traffic model	CBR
Number of vehicles	30–70
Vehicle communication range	250 m
MAC layer protocol	802.11

4.2 Simulation Results Analysis

4.2.1 Performance Indicators

After simulation, all the simulation results are obtained, including data such as the number of packet sent and received and the time of delivery, and the simulation results

are presented by drawing the data graph with GNUPLOT software. The main validation of this paper is as follows.

(1) Packet Delivery Loss Ratio
Packet loss rate is the percentage of the sum of the packets lost by the destination vehicle and the sum of the packet sent by the source vehicle within a certain period of time, usually expressed as a percentile. The calculation method is shown in Eq. (8):

$$P_{PDLR} = 1 - \frac{\sum_{\Delta t} P_r}{\sum_{\Delta t} P_s} \times 100\% \tag{8}$$

Where P_{PDLR} is the packet loss rate, $\sum_{\Delta t} P_r$ is the total number of packets received by the destination vehicle within the time period of Δt, $\sum_{\Delta t} P_s$ indicating the total number of packets sent by the source vehicle within Δt time.

(2) Average end to end delay
The average end to end delay is the ratio of the time it takes for the packets transfer from source vehicle to the destination vehicle and the total amount of packets successfully received. As shown in Eq. (9):

$$\mathrm{D_{AD}} = \frac{\sum_{\Delta t} \left(t_{p\_r} - t_{p\_s}\right)}{\sum_{\Delta t} P_r} \tag{9}$$

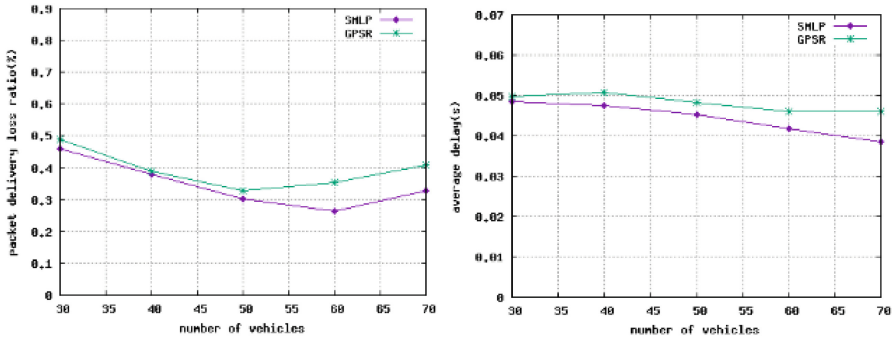
Here, D_{AD} is the average end to end delay, t_{p_r} indicating the packet reception time, t_{p_s} is the time when the packet transmitted, and $\sum_{\Delta t} \left(t_{p\_r} - t_{p\_s}\right)$ is the total time interval in the time period of Δt, $\sum_{\Delta t} P_r$ indicating the total amount of data packets received in Δt millisecond.

4.2.2 Validation Analysis

(1) Simulation scene one
The influence of the number of vehicle nodes on packet loss rate and average delay of different network protocols is analyzed under the same condition of vehicle node speed and other parameters. The speed of the vehicle node is set to a fixed value of 10 m/s, the number of vehicle nodes are set to 30, 40, 50, 60, 70 respectively, and the simulation experiments of these five groups are carried out in turn.

Figure 3(a) and (b) show the packet loss rate and the average end to end delay change as the number of vehicles increases. In the process of increasing the number of vehicles, although the number of nodes increased, but because the overall number of vehicles in the network is relatively small, the vehicle connectivity is not high

enough, so the packet loss rate of GPSR and SMLP show a downward trend but are relatively large, the average end to end delay also increased slightly. The number of vehicles continues to increase, making the network connected to strengthen and the packet loss rate and delay of two protocols are further reduced, because the SMLP protocol have considered the instability of the boundary node when selecting the forwarding node, making the selected relay can receive the security message in time, which reduces the packet loss and delay further than the GPSR protocol. But on the other hand, when the number of vehicle nodes in the network increases to a certain extent, the collision of the packets in the network increases, and there is a slight broadcast storm phenomenon, so the packet loss rate also increases.



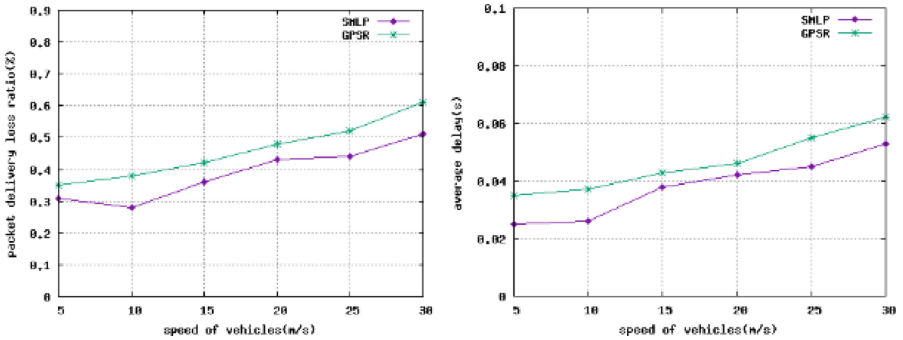
（a）Packet delivery loss ratio （b）Average end to end delay

Fig. 3. Impact of the number of vehicles on the packet loss rate and average delay

(2) Simulation scene two

The speed of the vehicle will affect the transmission efficiency of the packet, in order to verify the performance of the SMLP protocol, we set the number of vehicle nodes to 50, the only difference is the speed of the vehicle, set to 5 m/s, 10 m/s, 15 m/s, 20 m/s, 25 m/s, 30 m/s, respectively.

The greater the speed of the vehicle, the greater the packet loss rate and the delay, most of the current routing protocol is this feature, as shown in Fig. 4(a) and (b). This is because when the vehicle speed is too large, increasing the dynamic topology of the network, the connection between vehicles may break at any time, resulting in packet loss rate increases, while the delay increases, which is the overall trend of the two protocols. In the next-hop node selection algorithm of SMLP routing protocol, the connection time between nodes is considered as a consideration. Therefore, the selected route is a more stable path. So the rising trend of packet loss rate is smaller than that of GPSR which considers distance only. The reason why the delay of SMLP is smaller than GPSR is because the former has predicted the position of the nodes at the next time, and it will not choose a sub-optimal or even a “Pseudo neighbor node” which have out of the current node coverage because of the large difference between the relative velocity of the neighbor node and the current node.



（a）Packet delivery loss ratio （b）Average end to end delay

Fig. 4. Impact of the number of vehicles on the packet loss rate and average delay

5 Summary and Prospect

By analyzing the shortcomings of GPSR routing protocol, this paper proposes a new routing protocol - a security message broadcasting mechanism based on location prediction, which is abbreviated as SMLP. SMLP emphasizes the connectivity time and direction coefficient to the stability coefficient, and selects the neighbor node with the largest stability coefficient as the next hop forwarding node, rather than the boundary node in GPSR. However, there is no classification of security messages, and different priorities should be set according to the urgency. When the message in the network is large, it is necessary to use the scheduling algorithm to queue the messages according to the priority, so as to avoid the best transmission period of the emergency messages.

Acknowledgement. This work was supported by Shaanxi Province Hundred Talents Program, The key research and development plan of Shaanxi province (2017ZDCXL-GY-05-01); The natural science foundation of Shaanxi province (2016JM6058).

References

1. Zhao, H.: Study on the analysis of urban road traffic congestion and its countermeasures - taking Jinan City as an example. Value Eng. **09**, 22–23 (2016)
2. Mostafa, A.: Packet delivery delay and throughput optimization for vehicular networks. Dissertations & Theses-Gradworks (2013)
3. Douzi, Z., Dong, H.: A broadcasting scheme for distributing emergency messages in VANETs of vehicle ad hoc networks. Electronic Technique Application (7) (2015)
4. Lei, H.-M., Ye, X., An, L., Wang, Y.: Study on secure transmission strategy of vehicle ad hoc network security. J. Sichuan Ordnance Trade (8) (2011)
5. Li, M., Cao, J.: Geocast routing protocol based on adaptive delay in VANETs city scene. Telev. Technol. (5) (2015)
6. Yin, Y.: Study on multi-hop broadcasting protocol for emergency security message of vehicle self-organizing network. Dalian University of Technology (2013)

7. Brahmi, N., Boussedjra, M., Mouzna, J.: Mobility support and improving GPSR routing approach in vehicular adhoc networks. In: New Technologies, Mobility and Security (2008)
8. Clausen, T., Jacquet, P.: Optimized Link State Routing protocol. ED RFC3626, October 2003
9. Jianghong, H., Chao, L., Xing, W., et al.: Multi senders broadcast protocol based on distance in internet of vehicles. Comput. Eng. **41**(1), 6–11 (2015)
10. Karp, B., Kung, H.T.: GPSR: Greedy Perimeter Stateless Routing for wireless networks. In: Proceedings of the Annual International Conference on Mobile Computing and Networking, vol. 11, no. 2, pp. 243–254 (2000)

Efficient Algorithm for Traffic Engineering in Multi-domain Networks

Jian Sun[1], Siyu Sun[1], Ke Li[2], Dan Liao[1,3(✉)], and Victor Chang[4]

[1] Key Lab of Optical Fiber Sensing and Communications, University of Electronic Science and Technology of China, Chengdu, China
dliao.uest@gmail.com
[2] School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China
[3] Guangdong Institute of Electronic and Information Engineering, UESTC, Chengdu, China
[4] Xi'an Jiaotong-Liverpool University, Suzhou, China

Abstract. Current communication and network infrastructures have created billions of petabytes of data on the network every second. This imposes challenging traffic demands as a major research problem. This paper proposes a scheme for Cloud-of-Things and Edge Computing (CoTEC) traffic management in multi-domain networks. In order to direct the traffic flows through the service nodes in multi-domain networks, we assign the critical egress point for each traffic flow in the CoTEC network with multiple egress routers to optimize CoTEC traffic flows known as Egress-Topology (ET). Therefore, the proposed ET topology incorporates traditional Multi-Topology Routing (MTR) in the CoTEC network to address the inconsistencies between service overlay routing and the Border Gateway Protocol (BGP) policies. Furthermore, the proposed ET introduces a number of programmable nodes which can be configured to ease of ongoing traffic on the network, and re-align services among the other nodes in multi-domain networks. Results show that our optimization algorithm has lower execution time and better QoS than without using optimization algorithm, thus allowing us to meet the demand of flexibility and efficiency of multi-domain networks in comparisons to justify our research contributions.

Keywords: Multi-topology routing for CoTEC · CoTEC traffic engineering
Multi-domain networks for CoTEC

1 Introduction

The scale concluding the complexity of the Internet keeps growing with a huge trend in the last few years, particularly the development of Cloud-of-Things and Edge Computing (CoTEC), which are aimed to improve traffic and process millions and billions of data in the network [1–3]. Combining these technologies of Cloud-of-Things and Edge Computing have the following advantages. First, more sources, dimensions and availability of data can be collected and then processed by more resources, services and service providers. Second, boundaries between different service providers, APIs, standards can be thinner and less restricted, since real interoperability and integration of

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 365–374, 2018.
https://doi.org/10.1007/978-3-319-73830-7_36

shared resources, outputs and analysis can be offered [4–7]. However, demands to process and analyze millions and billions of data, and the abilities to manage billions of traffics, have become increasing important. Therefore, we need to address interoperability, performance, and security as well as the traffic monitoring and re-distributing for CoTEC. The internet technology for CoTEC is not only a medium of communication between machines and people, it is also providing an infrastructure for developing a range of applications and services that can run on the globally distributed internet technologies combining Cloud, IoT and Edge Computing. Service overlay networks (SON) for CoTEC have been proposed as a solution to provide the desirable service applications such as VoIP, video communication and surveillance, streaming services, which are essential for future network services [8–10].

Recently, under the background of SON for CoTEC, many researchers pay their curious attention to active networks [11, 12] consist of router nodes that can be programmed. Figure 1 illustrates a simple example. Deployed on individual nodes (i.e., active/programmable router), two kinds of services whose type names are s1 and s2 are assumed here. The active programmable routers are represented by filled solid circles on a node, while the common routers where there are no special service are indicated by the empty circles. In Fig. 1(a), we consider that traffic flow D (1, 8) owns service demand set {s1} and traffic requirement D (1, 4) owns {s2}. In the example, two routes are available for the traffic flow D (1, 8) to meet service requirement in SON. With the given position of service nodes, the traffic flow D (1, 8) would choose the route along the red line through border router node 3 to meet the service requirement. As illustrated in the example, there exists an influence or relationship between the service node placement strategy and traffic flow routing in CoTEC. Furthermore, the traffic flow should choose the border router 3 as an egress node of AS 1. However, this may lead to inconsistency between service overlay routing and Border Gateway Protocol BGP [12, 13] routing policy used between ASes when multiple border routers exist in the CoTEC network.

In this context, we propose the CoTEC network traffic engineering problem in multi-domain networks in this paper. A kind of deployment scenario of this problem is displayed in Fig. 1.

2 Problem Statement and Formulation

For the inter-AS overlay network, the network model is able to be denoted by a directed graph G = (V, E), in which V is the overlay nodes set in each intra-AS, while E is the set of links which are bidirectional and in the AS. For the local network with multiple egress routers, it is represented as graph G' = (V', E') directed in which V is the nodes set and E is bidirectional links set. Moreover, we use T to denote the logical topology set of local network.

We have a set of source-destination (OD) traffic demands and a set of services (S) in CoTEC. Every traffic flow generates its service requirements different from others, which is represented as, $(i,j) \in \mathrm{OD}$, $\mathrm{s} \in \mathrm{S}$.

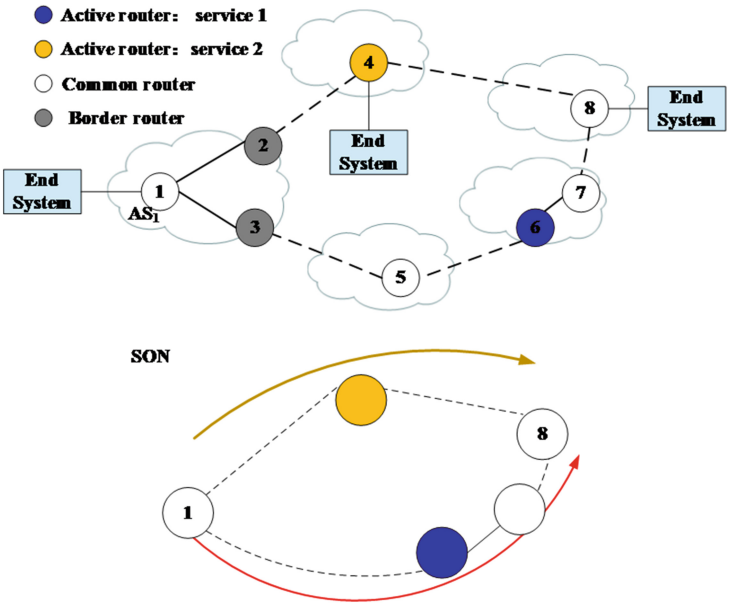


Fig. 1. An Examples of SON for CoTEC in Multi-Domain Networks

Let's assume that the possible paths of each traffic demand are given beforehand. We are trying to determine an optimal traffic routing, how to place programmable router nodes, and how to deploy network services through these nodes.

The objective function is adopting in [11]. Three parameters are needed in this function, i.e., T_{base} (the basic time delay of routing in an ordinary router), T_{active} (the delay of process in a programmable router), and T_s (the service delay of service types attached to the packet on programmable router). Hence, the time that the packet receives services over the routers in CoTEC that can be programmed during the entire process is to add those three parameters together: $T_{base} + T_{active} + T_s$.

Input:

$P(i,j)$: **the pair set of the paths that is from every source *I* to destination *j***

$f_{ij}(p)$: $i \in V, j \in V, p \in P(I, j)$, the traffic requirement of path p

β_p^k : $i \in V, j \in V, p \in P(I, j)$, node k belongs to path p for traffic requirement from node I to j.

D_{ij} : $(I, j) \in OD$, the vector of traffic requirement.

d_{ij}^s : $(I, j) \in OD$, $s \in S$,=1,in case that traffic flow from node I to j requires service type s; if not, =0.

c_e: $e \in E$, the capability of link e

Variables:

v: the maximum percentage of average packet processing time versus average packet arrival time at all nodes

A_k: $k \in V$, binary variables, = 1, if deployment of a router that can programmed is over node k; if not, = 0

B_k^s : $k \in V$, $s \in S$, binary variables, = 1, if deployment of service type s is over node k; if not, = 0.

y_{ij}^{ks} : $I, j, k \in V$, $s \in S$, binary variables, = 1, if service type s is provided by traffic demand pair (I, j) on node k; if not, = 0

Objective:

$$Minimize\ v \tag{1}$$

Constraints:

(1) *Multiple path routing*

$$\sum_{p \in P(i,j)} f_{ij}(p) = 1 \quad i,j \in V \tag{2}$$

$$f_{ij}(p) \cdot \beta_p^k = p_{ij}^k \quad i,j,k \in V, \mathrm{p} \in \mathrm{P(i,\ j)} \tag{3}$$

$$\sum_{i \in V} \sum_{j \in V} f_{ij}(p) \cdot \delta_{ij}^p(e) \cdot D_{ij} \le c_e \quad e \in E \tag{4}$$

Constraints (2) compels each traffic flow to travel through one of the given paths. Constraints (3) in CoTEC declares whether routing traffic requirement pair (i, j) is through node k. Constraint (4) declares that the sum of traffic through each link is no more than the capacity of link c_e.

(2) *Deployment of programmable nodes and services*

$$\sum_{t \in T} y_{ij}^{tks} \le p_{ij}^k, \quad i,j,k \in V, s \in S \tag{5}$$

$$\sum_{t \in T} y_{ij}^{tks} \le A_k, \quad i,j,k \in V, s \in S \tag{6}$$

$$\sum_{t \in T} \sum_{k \in N} y_{ij}^{tks} = d_{ij}^s, \quad i,j \in V, s \in S \tag{7}$$

$$B_k^s \ge \sum_{t \in T} y_{ij}^{tks}, \quad i,j \in V, s \in S \tag{8}$$

$$B_k^s \ge \sum_{t \in T} \sum_{(i,j) \in OD} y_{ij}^{tks}, \quad i,j \in V, s \in S \tag{9}$$

Constraints (5)–(7) relate to how to place programmable routers and how to deploy services among these nodes. Constraint (5) presents that the node k in routed requirement *(i, j)* cannot receive service s when the node is not on the shortest path in CoTEC. Constraint (6) assures that placing a programmable router can guarantee the service requirement. Constraint (7) shows that the service requirements of every traffic flow should be met. Constraints (8) and (9) avoid distributing repetitively services among programmable nodes in CoTEC.

(3) *Packet processing time.*

$$\sum_{(i,j)\in OD} D_{ij} p_{ij}^{k} T_{base} \le v, \quad k \in V \tag{10}$$

$$\sum_{(i,j)\in OD} D_{ij}(T_{base} + T_{active} + \sum_{s\in S} T_s) = C \tag{11}$$

$$\begin{aligned} &\sum_{(i,j)\in OD} D_{ij} p_{ij}^{k}(T_{base} + T_{active}) \\ &+ \sum_{(i,j)\in OD} D_{ij} \sum_{t} \sum_{s\in S} T_s y_{ij}^{tks} \le v + C(1 - A_k), \quad k \in V \end{aligned} \tag{12}$$

Constraints (10)–(12) form the stability conditions of the packet processing time. Constraint (10) states delay of packet processing on ordinary router nodes. Constraint (1 k) define the equation about C. Therefore, if a programmable router is on node k *(A_k = 1)*, the constraint (11) is correct, or is redundant when there is not a programmable router on node k *(A_k = 0)*. Constraint (12) provides the delay of packet processing for programmable router nodes. If $A_k = 1$, that is to say, node k is programmable router, delay contains T_{base}, T_{active}, and T_s. If not, the redundancy of constraint is seen.

3 Heuristic Algorithm Design for CoTEC

For making the sub-problem manageable, we propose a new heuristic optimization algorithm for CoTEC in this section (Fig. 2).

3.1 Logical Topology Design

In the paper, we design the logical topology allowing for distance constraint.

Heuristic optimization algorithm *for CoTEC*

Step-1: Construct logical topologies according to egress router nodes in CoTEC.
Step-2: For each logical topology, we adopt **Variable Neighbourhood Search (VNS)** algorithm for CoTEC to **set link weight.** We define the number of iterations for each logical topology to be 500.
Step-3: Re-compute link load for each link obtained from each logical topology in CoTEC.
Step-4: Output link weight $w^k(e)$ of each logical topology and maximum link utilization θ.

Fig. 2. The main process of the Heuristic optimization algorithm

3.2 Variable Neighborhood Search Algorithm

We present a Variable Neighbourhood Search (VNS) algorithm to determine link weight for each logical topology.

A. **Neighborhood structure for CoTEC**

The steps for VNS to generate a neighborhood needs two steps as follow: First, every link is sorted according to link utilization. Second, the link weight is changed based on the utilization of link for gaining the neighborhood.

B. **Diversification for CoTEC**

We introduce a new diversification operation to VNS in CoTEC. We modify each neighborhood by increasing link weights of recorded links. Accordingly, in order to avoid the currently-explored region, the re-evaluation is needed by these revised neighborhoods.

C. **Neighborhood evaluation for CoTEC**

It is challenging to completely explore the neighborhood because of the limitation of the number of neighbors allowed in the CoTEC solution. For the purpose of avoiding the shortcoming, only the neighbors set that satisfy the distance constrains can be evaluated here.

4 Experimental Results for CoTEC

4.1 Performance Metric

The metrics used for the evaluation of our solutions are showed as follow.

v: the maximum percentage of packet processing time and the average inter-arriving time in CoTEC.

MLU (Maximum Link Utilization) in CoTEC: The maximum utilization under all the links in a network. ed. This metric should be higher as possible.

4.2 Performance Results

A. Performance of PDR Optimization

We make comparison of PDR optimization performance with PD-only (PD optimization with given paths) and Routing-only (routing optimization with given position of programmable routers) adopted in *CoTEC*. We generate random networks in *CoTEC* (i.e., Net1, Net2, Net3).

From Table 1, we can see that the feasibility and efficiency of proposal PDR optimization scheme can be indicated through the experimental results.

Table 1. Comparison of different schemes adopted in CoTEC (T_{base} = 11.5629 μs, T_{active} = 1.9382 μs, T_s = 15 μs)

Scheme / Topology	PD-only Optimization	Routing-only Optimization	PDR Optimization
Net 1	0.0648	0.0714	0.0536
Net 2	0.1454	0.1551	0.1145
Net 3	0.2382	0.2190	0.1873

B. Performance Benefit of MTR in CoTEC

Then the evaluation of the effect of the MTR technique introduced in LW-MTR is given here. From analysis of results, the excess overhead of logical topologies can be experienced, as well as the overhead of computation and storage (Table 2).

Table 2. MLU using various amount of logical topologies in CoTEC (Ebone)

Quantity of logical topologies	Traffic matrix				
	TM1 α_i = 0.0125	TM2 α_i = 0.025	TM3 α_i = 0.05	TM4 α_i = 0.1	TM5 α_i = 0.2
2	0.38	0.49	0.53	0.65	0.74
3	0.35	0.46	0.51	0.63	0.74
4	0.35	0.44	0.49	0.62	0.74

C. Effect of Heuristic Algorithm in CoTEC

As shown in Table 3, our algorithm for CoTEC performs as near-optimal level.

Table 3. Results of different calculation schemes

Topology	Heuristic MLU	Time(s)	Optimal solution MLU	Time(s)	LPR solution MLU	Time(s)
Telstra	0.57	52	0.57	120	0.41	9
Ebone	0.63	168	N/A	N/A	0.42	105

In Table 4, we can make a statement that our heuristic algorithm is superior to FT algorithm in terms of both the quality of the solution and the running time.

Table 4. Results of different local search algorithms in CoTEC

Topology	Heuristic algorithm MLU	Time(s)	FT MLU	Time(s)
Telstra	0.57	48	0.57	86
Ebone	0.63	280	0.77	1,936

4.3 Validating Our Proposal with a Real Service for CoTEC

To validate our approach, we used two services to compare performance in CoTEC, one used the proposal discussed in this paper and the other did not use the proposal.

Results in Fig. 3 show the linear relationship between the approach with our proposal and without proposal.

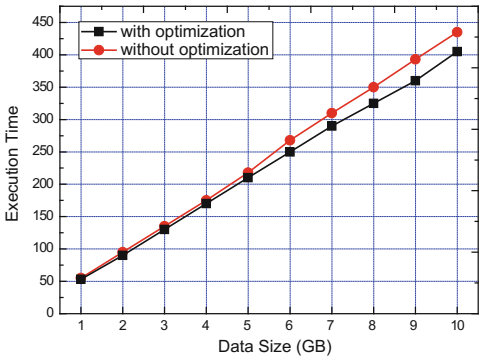


Fig. 3. Execution time for identifying network problems

Figure 4 shows the execution time for optimizing network problems while transferring and storing data up to 10 GB in size in the heterogeneous CoTEC environment.

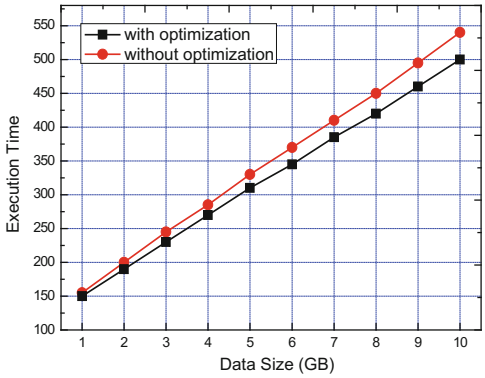


Fig. 4. Execution time for optimizing network problems

5 Conclusion

We research the network traffic engineering problem in multi-domain networks in CoTEC and two novel optimization models are proposed in our paper. The first model optimizes the deployment of routers that can be programmed, as well as distribution of services among these routers, while taking account of traffic flow routing (i.e., overlay routing). The second one resolves complex consistency issue between overlay routing and BGP policy aimed at every traffic flow. In addition, the assign of egress router meets the service demands while putting forward optimized resource utilization of the in CoTEC network. We introduce the multi-topology routing (MTR) in order to ensure the performance of assigned egress routers with near-optimal network. In such a manner, we can use MTR by the method of optimizing link weight to make local network dimensioned in the purpose of achieving path diversity (multiple shortest paths) among multiple logical topologies. Furthermore, a heuristic optimization algorithm is used to make the issue of optimizing link weight using MTR tractable in CoTEC. We validate our research contributions and explain how our research work is highly relevant. Experiential results show that our research is beneficial to both service providers and network operators by identifying network problems and optimizing them, thus it can maintain a better QoS in CoTEC.

Acknowledgement. This research was partially supported by the National Natural Science Foundation of China (61571098), Fundamental Research Funds for Central Universities (ZYGX2016J217), Guangdong Science and Technology Foundation (2013A040600001, 2013B090200004, 2014B090901007, 2015A040404001, 2013B040300001).

References

1. Dastjerdi, A.V., Gupta, H., Calheiros, R.N., et al.: Fog computing: principles, architectures, and applications. arXiv preprint arXiv:1601.02752 (2016)
2. Díaz, M., Martín, C., Rubio, B.: State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. J. Netw. Comput. Appl. **67**, 99–117 (2016)
3. Bandyopadhyay, D., Sen, J.: Internet of things: applications and challenges in technology and standardization. Wirel. Pers. Commun. **58**(1), 49–69 (2011)
4. Sun, G., Chang, V., Ramachandran, M., et al.: Efficient location privacy algorithm for internet of things services and applications. J. Netw. Comput. Appl. **89**, 3–13 (2017)
5. Sohal, A.S., Sandhu, R., Sood, S.K., et al.: A cybersecurity framework to identify malicious edge device in fog computing and cloud-of-things environments. Comput. Secur. (2017)
6. Chang, V.: Towards data analysis for weather cloud computing. Knowl.-Based Syst. **127**, 29–45 (2017)
7. Sun, G., Xie, Y., Liao, D., et al.: User-defined privacy location-sharing system in mobile online social networks. J. Netw. Comput. Appl. **86**, 34–45 (2017)
8. Sun, G., Yu, H., Anand, V., et al.: Optimal provisioning for virtual network request in cloud-based data centers. Photon. Netw. Commun. **24**(2), 118–131 (2012)
9. Al Ridhawi, Y., Karmouch, A.: QoS-based composition of service specific overlay networks. IEEE Trans. Comput. **64**(3), 832–846 (2015)
10. Chen, Y., Radhakrishnan, S., Dhall, S., et al.: The service overlay network design problem for interactive internet applications. Comput. Oper. Res. **57**, 73–82 (2015)
11. Liu, R., Liu, H., Kwak, D., et al.: Balanced traffic routing: design, implementation, and evaluation. Ad Hoc Netw. **37**, 14–28 (2016)
12. Sun, G., Yu, H., Anand, V., et al.: A cost efficient framework and algorithm for embedding dynamic virtual network requests. Future Gener. Comput. Syst. **29**(5), 1265–1277 (2013)
13. Caesar, M., Rexford, J.: BGP routing policies in ISP networks. IEEE Netw. **19**, 5–11 (2005)
14. Sermpezis, P., Dimitropoulos, X.: Analysing the Effects of Routing Centralization on BGP Convergence Time. arXiv preprint arXiv:1605.08864 (2016)

Reliable and Efficient Deployment for Virtual Network Functions

Jian Sun[1], Gang Sun[1,2], Dan Liao[1,3(✉)], Yao Li[1], Muthu Ramachandran[4], and Victor Chang[5]

[1] Key Lab of Optical Fiber Sensing and Communications, Ministry of Education, University of Electronic Science and Technology of China, Chengdu 611731, China
dliao.uestc@gmail.com
[2] Center for Cyber Security, University of Electronic Science and Technology of China, Chengdu 611731, China
[3] Guangdong Institute of Electronic and Information Engineering, UESTC, Chengdu 523808, China
[4] Leeds Beckett University, Leeds, UK
[5] Xi'an Jiaotong-Liverpool University, Suzhou 215123, China

Abstract. Network function virtualization (NFV) is a promising technique aimed at reducing capital expenditures (CAPEX) and operating expenditures (OPEX), and improving the flexibility and scalability of an entire network. However, this emerging technique has some challenges. A major problem is reliability, which involves ensuring the availability of deployed SFCs, namely, the probability of successfully chaining a series of big-data-based virtual network functions (VNFs) while considering both the feasibility and the specific requirements of clients, because the substrate network remains vulnerable to earthquakes, floods and other natural disasters. Based on the premise of users' demands for SFC requirements, we present an Ensure Reliability Cost Saving (ER_CS) algorithm to reduce the CAPEX and OPEX of telecommunication service providers (TSPs) by reducing the reliability of the SFC deployments. We employ big-data-based arbitrary topologies as the substrate network. The results of extensive experiments indicate that the proposed algorithms perform efficiently in terms of the blocking ratio, resource consumption and time consumption.

Keywords: Network function virtual · Service function chains · Reliability
Economical big data networking

1 Introduction

Telecommunication service providers (TSPs) desire flexible and cost-efficient methods for dispatching network services as market demands increase. NFV provides an opportunity to efficiently and dynamically deploy service function chains (SFCs) [1–3] without modifying dedicated infrastructure, which is costly and has become complex over time. The basic idea behind NFV is to decouple these network functions (e.g., firewall, WAN optimizers, and proxies) from the underlying customized devices and

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 375–384, 2018.
https://doi.org/10.1007/978-3-319-73830-7_37

accomplish equivalent network functions via software-based functions running in virtual machines deployed on devices.

Because the reliability of NFV is critical and is a prerequisite for successfully executing SFCs and satisfying service level agreements (SLAs), improving reliability while reducing the cost of network providers is a research objective in academic and industrial arenas.

In this paper, we propose an ER algorithm to solve this problem. High reliability requires TSPs to increase CAPEX and OPEX. If we can properly reduce the reliability, we can also reduce CAPEX and OPEX. We first propose the algorithm ER_CS (based on ER) that works in conjunction with the load balancing of the substrate network. However, by analyzing the deployment scheme in ER_CS, we discover that it does not appear to be the best scheme. Therefore, we further propose the ER_CS_ADJ algorithm to adjust the deployment scheme by minimizing SFC resource consumption in the physical network. We conduct massive simulations on arbitrary topologies to verify the effectiveness of these algorithms. From the simulations and results, we determine that our algorithms are profitable in terms of resource costs, block ratio and deployment time.

The remainder of this paper is organized as follows. In Sect. 2, we analyze related studies. In Sect. 3, we describe the problem in this research with some formulations. In Sect. 4, we propose our heuristic algorithm and provide line-by-line details. A performance evaluation of the network algorithm is presented in Sect. 5, and Sect. 6 concludes this work.

2 Related Work

To satisfy various requests from users, service providers are eager to seek a flexible, scalable, agile, effective, resource efficient and energy efficient scheme for placing VNFs. Ensuring service reliability while finding an economical and resource-efficient solution to the problem of big-data-based VNF deployment is the goal of this work.

Numerous studies are relevant to big-data-based NFV, including how to determine and place network functions. Bouten et al. [2] presented a set of affinity and anti-affinity constraints that can be used by TSPs to define big-data-based placement constraints. They proposed a semantic conflict mechanism to evaluate SFC requests that filters invalid mechanisms to reduce the mapping time.

The performance of big-data-based NFVs with regard to resource allocation or consumption and the acceptance ratio when mapping big-data-based VNFs has been investigated for years. Rankothge et al. [11] proposed a genetic algorithm to optimize resource allocation. They demonstrated its efficiency in optimizing resource allocation via three network function centers proposed by the authors.

Other research projects have focused on issues such as the availability of big-data-based NFV. Due to potential failures (such as node or link failures) that can be caused by earthquakes, floods, or malfunctions such as power outages, many researchers have expressed interest in the field of high availability (HA) to protect data or network functions. Unlike some schemes, which aim to solve general big-data-based VN mapping problems for unicast services (which includes two procedures: virtual node and link

mapping) such as [4]. The authors of [13] proposed an efficient framework for evaluating the reliability of NFV deployments; however, they did not investigate how to adjust NFV deployments based on their framework. The proposed framework can be used only to evaluate deployment schemes but was not intended to improve the schemes based on its results.

Although numerous studies have considered the reliability of deployed SFCs, few studies have considered the needs of users while also considering the TSP revenues. In other words, few studies have focused on building an economical network environment. Therefore, we propose the ER_CS algorithm to reduce reliability under the premise of guaranteeing users' demands while also considering economical VNF deployments.

3 Problem Statement

As described in Fig. 1, a SFC request consists of several VNFs, a source node s and a destination node t. Each of these VNFs represents a network function, as described above. The thick blue dashed line represents another scheme whose reliability is 0.94 and resource consumption is 202, called service function forwarding path 1 (SFP1). The thick red dotted line represents one deployment scheme for the request whose reliability is 0.97 and resource consumption is 232, called SFP2. We assume that the demand reliability of users is 0.90. The thin blue dashed line, which represents a VNF in SFC, is deployed on a substrate network in SFP1. The red line will yield the best experience for the users, whereas the blue line will generate a better balance for the network providers because the network can hold more requests, which allows greater potential

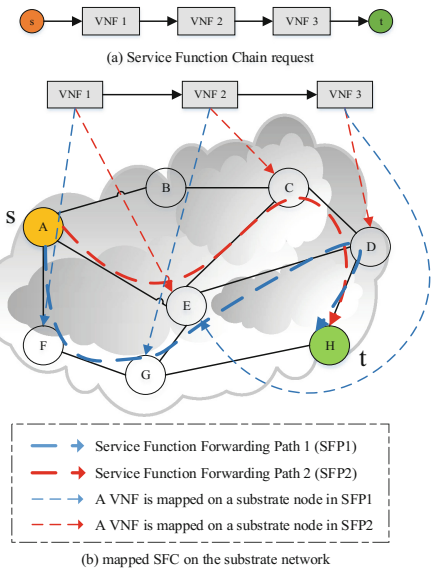


Fig. 1. Example of mapped VNFs (Color figure online)

profits for TSPs. The goal of this paper is to find a deployment scheme that both satisfies users' reliability demands and minimizes resource consumption to reduce costs.

To achieve effective and reliable network services while deploying SFC requests, we need to deploy VNFs to more reliable nodes and attempt to maximize the total availability of the deployment of SFC. This goal can be notated as formula (1).

$$\begin{aligned} &\max\left\{R^S = \prod_{v_p \in V_N^S} r_{v_p} \times \prod_{e_p \in E_L^S} r_{e_p}\right\} \\ &\forall v_p \in V_P, 0 < r_{v_p} < 1.0 \\ &\forall e_p \in E_P, 0 < \mathrm{r}_{e_p} < 1.0 \end{aligned} \tag{1}$$

Where r_{vp} and r_{ep} represent the reliability of the nodes and links deployed for SFC requests, respectively. The reliability of each node and link in the underlying network is denoted by a positive number less than 1 according to the constraint behind the optimization objective. This paper estimates the total reliability of SFC by calculating the product of the reliability of each substrate node and link involved in a SFC deployment scheme.

Due to limited resources, considering only the reliability of SFC may cause enormous resource consumption and reduce the mapping success rate. Therefore, the paper aims to solve the contradiction between the reliability and the bandwidth consumption, maintaining a balance between resource consumption and service reliability to ensure the effective use of resources.

4 Algorithm Design

4.1 Ensure-Reliability Heuristic Algorithm ER

In a NFV environment, many virtual networks share one substrate network; consequently, the failure of one substrate link or substrate node may cause massive failures in virtual networks, have a large-scale impact, and reduce network stability. Therefore, we propose a heuristic algorithm referred to as ER based on the reliability-aware SFC mapping problem. The pseudo-code is presented in Algorithm 1.

***Algorithm* 1.** Ensure big-data-based reliability (ER)
Input: 1. Substrate network $G_P = (V_P, E_P)$;
2. SFC request $SR = (N_S, L_S, s, t)$.
Output: SFC deployment scheme P^S.
1: Initialization: **let** $V_{remain} = V_P$;
2: **for** all VNF n_f in SR, **do**
3: **if** n_f is not the last VNF of SFC, **then**
4: initiateAllVertex() and **let** $r_{v_{so}}^{v_{so}} = r_{v_{so}}$;
5: Call URSO procedure 1 to update the information;
6: **let** $\kappa_r = -\infty$ and $v_{temp} = v_\infty$;
7: **for** each vertex v in V_P, **do**
8: **if** $v \neq v_{si}$ and $w_v^r \geq w_{n_f}$ and $\kappa_r < r_v^{v_{so}}$, **then**
9: $\kappa_r = r_v^{v_{so}}$, $v_{temp} = v$;
10: **end if**
11: **end for**
12: **if** $\kappa_r = -\infty$, **then return** null;
13: generateScheme(κ_r, n_f), let $v_{so} = v_{temp}$;
14: **else**
15: repeat line 4 and 5, $r_{v_{si}}^{v_{si}} = r_{v_{si}}$
16: call URSI procedure 2 to update the information;
17: **for** each vertex v in V_P, **do**
18: **if** $w_v^r \geq w_{n_f}$ and $\kappa_r < r_v^{v_{si}} \times r_v^{v_{so}} / r_v$, **then**
19: $\kappa_r = r_v^{v_{si}} \times r_v^{v_{so}} / r_v$, $v_{temp} = v$;
20: **end if**
21: **end for**
22: repeat line 12 to line 13;
23: **end for**

When receiving a big-data-based SFC request, we firstly let $v_{so} = s$, $v_{si} = t$. When one VNF has been deployed, we will change the value of v_{so} just like line 13.

For all the VNFs other than the last one, the ER algorithm initializes the reliability of all vertexes to the source to be negative infinity and the reliability of the source vertex to be its vertex's reliability. Then, it initializes their prior vertex on the path to the source to be an inaccessible node. Next, it calls procedure 1—update all reliability to source (URSO)—to update the reliabilities of all nodes to the SFC source. In lines 6 to 11, we initialize the maximum reliability variable and the substrate node that has the maximum reliability to map the VNF, and traverse all the nodes to find the variable defined in line 6, which cannot be the sink vertex. We generate the mapping scheme and map the VNF onto the vertex v_{temp} with the reliability calculated in the previous procedure. If the reliability variable remains negative infinity, we are unable to find a mapping vertex that satisfies the demands for mapping VNF.

To map the last VNF in an SFC we must not only consider the mapping vertex's reliability to the previous VNF mapping vertex but also its accessibility and reliability at the destination node of the SFC. Similar to the previously described algorithm, we update the reliabilities of all nodes to the SFC's destination after updating the reliabilities to the SFC's source.

URSO will compute all the path's (from one underlying node in V_{remain} to the source node v_{so}) reliability, choosing and saving a path which has the max reliability. This procedure will traverse the node in V_{remain} and find a node (this node must satisfy computing resource requirement of the current VNF, and the edges in the path (from source node to it) also need to satisfy bandwidth resource requirement of the virtual link (from prior VNF to current VNF)) that has max reliability.

Procedure 2 (i.e., update all reliability to sink (URSI)) is similar to URSO; the only difference is that rather than computing the reliability to the source, it computes the reliability to the destination.

4.2 Big-Data-Based Ensure-Reliability Cost Saving Heuristic Algorithm ER_CS Based on Load Balancing

To maximize the reliability, SFC functions should be deployed on vertexes with high reliability, which may cause imbalanced loading in the network. Based on the algorithm ER, we introduce the idea of load balance and present the reliability-guarantee heuristic algorithm ER_CS, which is based on load balance.

In this thesis, the objective of load balance is to assign service flow transport to links with lighter loads to reduce the possibility of congestion caused by load imbalance. The following mathematical model describes load improvement:

$$\delta = \frac{1}{w^r_{v_i}} + \sum_{e^o_{v_i} \in e^o_i} \frac{1}{m^r_{e^o_{v_i}}} + m^{v_{so}}_{v_i}, \forall v_i \in V_P \tag{2}$$

Where e^o_i denotes the set of the out-degree edge of vertex v_i, the denominator in the second fraction denotes the remaining bandwidth resource of the out-degree edge of vertex v_i, and the last symbol denotes the sum of the bandwidth cost of the path from vertex v_i to the source v_{so}. As expressed by the formula, the smaller the load factor is, the larger the vertex's remaining computing resource is, and the larger the remaining bandwidth resource of the out-degree is, the smaller the total bandwidth cost of the vertex to the source is.

Therefore, we adjust the ER algorithm to compute the δ of all the vertexes that satisfy the criteria based on satisfying R^U, the node's computing resource demands and the link's bandwidth resource demands. We add a comparison of the values of δ in URSO to find the vertexes with smaller δ values to host VNFs.

4.3 Bandwidth Optimizing Algorithm ER_CS_ADJ

We improve the ER_CS algorithm through bandwidth cost reduction, and we propose the bandwidth optimizing algorithm ER_CS_ADJ. We skillfully adjust the VNFs'

mapping position based on the mapping scheme generated by ER_CS to lengthen the mapping paths of virtual links with low bandwidth demands and shorten it with high bandwidth demands; consequently, we reduce the bandwidth cost.

Algorithm 2. Big-data-based ER_CS adjust (ER_CS_ADJ)
Input: SFC deployment scheme P^S.
Output: Adjusted SFC deployment scheme P^S.
1: **let** χ_{move} = findMinLink(SR);
2: **if** χ_{move} = 0, **then return;**
3: **while** $\chi_{move} > 0$
4: **for** all n_f need to be removed, **do**
5: **for** all forwarding vertex v between two related function vertex, **do**
6: **if** $w_v^r \geq w_{n_f}$ and $B_{remain}^{\min} \geq B_{request}$, **then**
7: deploy n_f on vertex v;
8: **end if**
9: **end for**
10: **end for**
11: χ_{move} --;
12: **end while**

The function findMinLink(SR) finds the virtual link with the minimum bandwidth request in the SFC. The VNFs behind this link are the VNFs that must be moved; we denote the number of these as χ_{move}. When moving these VNFs, we need to traverse the VNFs in reverse order. When we adjust the mapping position of one VNF, we traverse all the forwarding vertexes on the path between this VNF and the updated VNF in reverse order. For example, when moving the last VNF, we traverse forward from the first forwarding vertex prior to the destination of the SFC. When moving the penultimate VNF, the deployment position of the last VNF is determined; thus, we traverse forward from the deployment position of the last VNF. The remaining steps can be performed in the same manner.

5 Simulation Results

5.1 Simulation Environment

To evaluate the schemes described in Sect. 4, we implemented an event simulation in Java. To demonstrate the applicability of the algorithm for all circumstances, we employ the Waxman 2 model from GT-ITM [15] to randomly generate small and large network instances as substrate networks. The small substrate network includes 20 nodes and the large substrate network contains 100 nodes.

During the simulation process, to compare and evaluate the performance of the three algorithms, we modified Compute followed by Network Load Balance (CNLB) [9] to

the Link Mapping First (LMF) algorithm [10] without changing its core concept to be the compared algorithm in this paper.

5.2 Simulation Results and Analysis

All the left figures below was simulated in a small simulation topology, and the right one was simulated in a big simulation topology.

Figure 2 shows the simulation results of the SFC block rate when deploying SFC requests for these four algorithms. We vary the number of functions of each SFC from 3 to 12 and randomly generate 10,000 SFC requests for each number of functions. The block rate denotes the proportion of the failed SFC deployment requests in all 10,000 SFC requests. The comparisons shown in left and right indicate that the three algorithms have a distinct advantage in block rate as the network size increases.

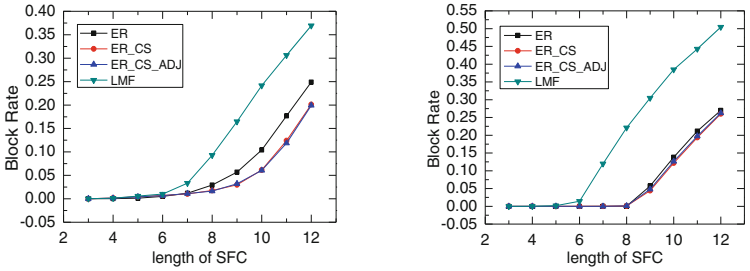


Fig. 2. Block rates of SFCs in different topology

The results of the bandwidth overhead for SFC requests, shown in Fig. 3, reveal that the three algorithms proposed in this paper have an advantage over the LMF scheme in terms of bandwidth consumption, and that the ER_CS_ADJ algorithm performs the best.

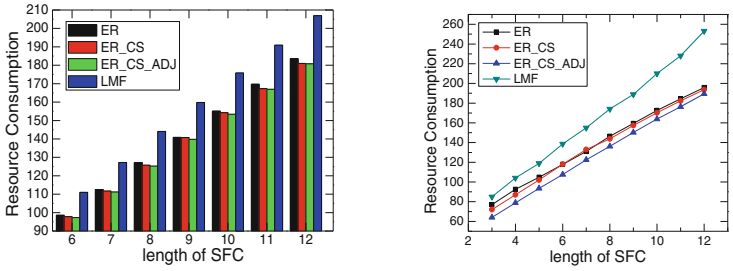


Fig. 3. Average resource consumption (i.e., computing resource and bandwidth resource) of SFCs in different topologies

The time consumption of each SFC mapping algorithm was evaluated by gradually increasing the number of service function chain requests, as shown in Fig. 4. The average time overhead of the SFC requests deployed by the three algorithms proposed in this paper is substantially lower than the average time overhead of the LMF algorithm.

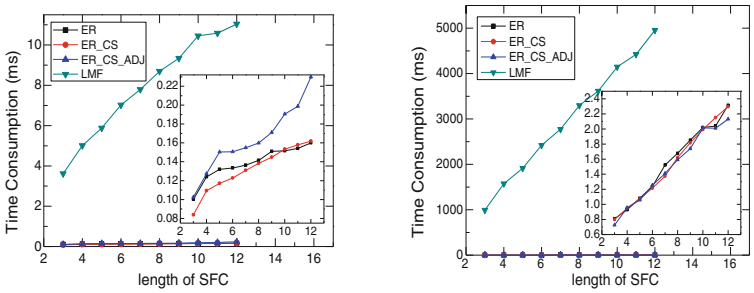


Fig. 4. Average time consumed when SFCs are deployed

6 Conclusions and Future Work

In this paper, we identified a problem: the high reliability requests of users reduce the CAPEX and OPEX of TSPs. Thus, we proposed ER to guarantee the basic reliability needs of users. However, considering the revenue of the TSPs, we discover that network imbalances will influence the request success rate and the resource utilization rate. Therefore, we proposed ER_CS, which is based on ER and considers the load balance factor. Although this algorithm achieved substantial progress, we discovered that the scheme used for ER_CS can be improved. Thus, we proposed ER_CS_ADJ. The simulation results indicate that ER_CS_ADJ achieves the objectives of this study. We demonstrated that our network algorithms can successfully work in a range of test environments and satisfy user demands.

Acknowledgement. This research was partially supported by the National Natural Science Foundation of China (61571098), Fundamental Research Funds for the Central Universities (ZYGX2016J217), Guangdong Science and Technology Foundation (2013A040600001, 2013B090200004, 2014B090901007, 2015A040404001, 2013B040300001).

References

1. Mechtri, M., Ghribi, C., Zeghlache, D.: A scalable algorithm for the placement of service function chains. IEEE Trans. Netw. Serv. Manag. **13**(3), 533–546 (2016)
2. Bouten, N., Mijumbi, R., Serrat, J., et al.: semantically enhanced mapping algorithm for affinity-constrained service function chain requests. IEEE Trans. Netw. Serv. Manag. **14**(2), 317–331 (2017)
3. Beck, M.T., Botero, J.F.: Scalable and coordinated allocation of service function chains. Comput. Commun. **102**, 78–88 (2017)

4. Chowdhury, N.M.M.K., Rahman, M.R., Boutaba, R.: Virtual network embedding with coordinated node and link mapping. In: INFOCOM, pp. 783–791 (2009)
5. Gao, X., Ye, Z., et al.: Virtual network mapping for multicast services with max-min fairness of reliability. IEEE/OSA J. Opt. Commun. Netw. **7**(9), 942–951 (2015)
6. Cziva, R., Pezaros, D.P.: Container network functions: bringing NFV to the network edge. IEEE Commun. Mag. **55**(6), 24–31 (2017)
7. Sun, G., Liao, D., Bu, S., et al.: The efficient framework and algorithm for provisioning evolving VDC in federated data centers. Future Gener. Comput. Syst. **73**, 79–89 (2017)
8. Sun, G., Anand, V., Liao, D., Lu, C., et al.: Power-efficient provisioning for online virtual network requests in cloud-based datacenters. IEEE Syst. J. **9**(2), 427–441 (2015)
9. Ye, Z., Patel, A.N., Ji, P.N., et al.: Virtual infrastructure embedding over software-defined flex-grid optical networks. In: GLOBECOM, pp. 1204–1209 (2013)
10. Ye, Z., Cao, X., Wang, J., et al.: Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization. IEEE Netw. **30**(3), 81–87 (2016)
11. Rankothge, W., Le, F., Russo, A., et al.: Optimizing resource allocation for virtualized network functions in a cloud center using genetic algorithms. IEEE Trans. Netw. Serv. Manag. **14**(2), 343–356 (2017)
12. Luizelli, M.C., Cordeiro, W.L.D.C., Buriol, L.S., et al.: A fix-and-optimize approach for efficient and large scale virtual network function placement and chaining. Comput. Commun. **102**, 67–77 (2017)
13. Liu, J., Jiang, Z., Kato, N., et al.: Reliability evaluation for NFV deployment of future mobile broadband networks. IEEE Wirel. Commun. **23**(3), 90–96 (2016)
14. Knuth, D.E.: A generalization of Dijkstra's algorithm. Inf. Process. Lett. **6**(1), 1–5 (1977)
15. Calvert, K.L., Zegura, E.: GT-ITM: Georgia tech internetwork topology models (Software). Georgia Tech. http://www.cc.gatech.edu/fac/Ellen.Zegura/gt-itm/gt-itm.tar.gz

Empirical Study of Data Allocation in Heterogeneous Memory

Hui Zhao[1], Meikang Qiu[2,3], and Keke Gai[4(✉)]

[1] Institute of Intelligent Network System, Henan University, Kaifeng 475000, China
zhh@henu.edu.cn
[2] Department of Computer Science, Pace University, New York City, NY 10038, USA
mqiu@pace.edu
[3] Shenzhen University, Shenzhen 518060, Guangdong, China
[4] School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
kekegai@yahoo.com

Abstract. With the rapid development of data-driven technologies, implementing heterogeneous memories is an alternative for processing large-size data tasks or efficient computations while considering economic factors. Many previous studies have addressed the exploration of adopting heterogeneous memories in the field of the algorithm design. One of the vital components of using the heterogeneous memory is creating effective data allocation plans. However, it is challenge to discern the superiority of each method for generating data allocation plans due to various application scenarios and constraints. In this work, we have completed an empirical study focusing recent advanced data allocation mechanisms for heterogeneous memories. We use experimental evaluations to examine a number of representative strategies and the main findings of this work also include analyses and syntheses deriving from our evaluations.

Keywords: Data allocation · Heterogeneous memory
High performance · Empirical study

1 Introduction

With the swift development of the memory design and production, the financial cost of the memory has been cutting down while the hardware performance is continuously increasing. This trend has enabled a flexible deployment of the memory for fitting in various users' demands [1–3]. Meanwhile, the blooming adoption of big data requires enhancements in both data processing and data storage. The connected environment further grows the workload of the data

This work is supported by the Basic and Frontier Technology Research of Henan Province Science and Technology Department (No. 162300410198).

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 385–395, 2018.
https://doi.org/10.1007/978-3-319-73830-7_38

exchange and task distributions [4–7]. As a core component of data processing, increasing the usage and efficiency of the memory is urgent needs in a big data era. Thus, designing heterogeneous memories has become a popular research domain that attracts numerous researchers over years. A variety of data allocation algorithms have been proposed.

Moreover, heterogeneous memory also has a potential value for in-memory big data analytics. The study [8] applied the technique of the file virtual address to the in-memory file system in order to optimize memory usage and the approach was called SIMFS. Another study [9] also addressed similar technique and proposed a data storage approach, called SIM-DAM, which enhanced the efficiency of the memory space by optimally distributing data storage between in-memory storage and disk storage.

Despite many prior studies addressing the design of the heterogeneous memory, it is hard to discern the performance of various data allocation algorithms. Most optimization algorithms are developed to specific implementations. Inputs and outputs maybe varied due to different application assumptions and system configurations. The availability of memory types also introduces difficulties in understanding the variety of different data allocation algorithms. Lack of unified configuration standard results in the situation that memory producers can hardly understand the differences between distinct algorithms, which restricts the adoption of the heterogeneous memory in practice. Therefore, there is an urgent demand to perceive the prior data allocation approaches' performance throughout a comprehensive intercross evaluation.

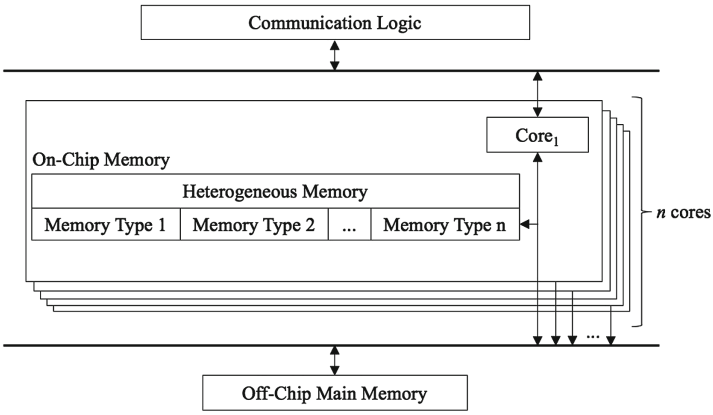


Fig. 1. The typical architecture of the heterogeneous memory.

This paper concentrates on evaluating the performance of the data allocation algorithms for heterogeneous memory, which mainly covers the application scenarios, input and output configurations, computing resource consumptions, energy costs, and strategy generation time. Figure 1 illustrates the typical

architecture of the heterogeneous memory. A data allocation scheduling mainly address the task distribution to different memory types including both on-chip memory and off-chip main memory. An empirical study has been accomplished, which are aligned with the concerns mentioned above. The significance of this study is observable due to both urgent demands in practice and contributions to the body of knowledge.

The main contributions of this paper include:

1. This empirical study has evaluated a few typical data allocation approaches for heterogeneous memory. The study provides a review on the evaluated approaches, which include heuristic algorithm, dynamic programming, and a few active resource scheduling algorithms.
2. The synthesis of main findings can provide memory producers and heterogeneous memory researchers with a quantitative support. We also provide recommendations for distinct algorithms' implementations.

The rest of this paper is organized by the order below. First, Sect. 2 presents preliminary work about previous studies of data allocations, which focuses on a number of representative approaches in our empirical study. Next, Sect. 3 illustrates the evaluation configuration, experiment results, and analyses. Finally, Sect. 4 gives a conclusion of our empirical study.

2 Preliminary

We reviewed a number of prior characteristic data allocation algorithms throughout a qualitative study in this section. The number of the reviewed approaches was limited due to the page length restriction so that only representative data allocation mechanisms were selected.

First, contemporarily active approaches generally required a dramatic short time for generating data allocation strategies. *Round-Robin* (RR) algorithm was a classic scheduling algorithm that was widely adopted in practice due to its easy deployment [10]. The disadvantage of RR was that the optimization level of the output was low. The other commonly deployed algorithm is a greedy algorithm [11]. This type of the algorithm had a priority setting so that the output generally was a sub-optimal solution while ensuring a short strategy generation time. The main reason for broadly deploying these two types of algorithms was its extremely short time period for creating data allocation plans. This phenomenon also was related to the implementation characteristics of the memory, since most in-memory data processing tasks take a short moment. A long execution time for creating a data allocation strategy could result in a latency.

Moreover, the heuristic algorithm was a typical approach in data allocations. The main advantage of a heuristic algorithm was that the data allocation strategy generation was relatively efficient. However, most heuristic algorithms could not produce optimal solutions. An expected output of the heuristic algorithm was a near-optimal solution. For example, Qiu et al. [12] proposed a genetic

algorithm to optimize SLC/MLC PCM memory for the purpose of green computing. In addition, SDAM [13] was a heuristic algorithm that was designed for a multi-dimensional heterogeneous memory setting. The algorithm used a cliff-climbing mechanism that transferred a sub-optimal solution to a near-optimal or an optimal solution. The crucial component of this algorithm was called a *Smart Switch*, in which the enhancement of the strategy took place. Meanwhile, considering the connected environment, CAHCM is another heuristic algorithm [14] that emphasized the implementation of cloud computing. The number of the reads and writes for the input data was used as the priority for creating an original strategy. An optimization was operated over the original strategy such that the output was a near-optimal solution. This algorithm also considered the performance cliff of the memory.

Furthermore, dynamic programming was an approach that could produce optimal solutions that was a major advantage. In general, the main weakness of dynamic programming was that it required a long execution time period for creating outputs. Hu et al. [15] proposed DAHS that was an optimization solution to hybrid scratch pad memory combining SRAM and nonvolatile memory. Qiu et al. [16] further developed a dynamic programming (MDPDA) that could allocate data to three types of memory for hybrid scratch-pad memory. This algorithm was a representative dynamic programming designed for heterogeneous memory. The number of the memory type was fixed according to the algorithm setting. Zhao et al. [17] further proposed an approach entitled OMDDA for multi-dimensional heterogeneous memory. The number of the memory type was unfixed so that it could be applied to various heterogeneous memory. Its back-forward table was also improved such that the strategy generation time could be shortened, considering the perspective of the computation complexity.

The next section will illustrate some experiment results collected from the evaluation of the selected algorithms.

3 Empirical Studies

3.1 Experiment Configuration

We illustrated partial experiment results and main findings that are associated with Sect. 2. The evaluations were completed on our simulator that offered the standard and unified input settings. Comparisons were completed based on a number of representative scheduling algorithms, which included RR [10], Greedy [11], SDAM [13], CAHCM [14], MDPDA [16], DAHS [15], and OMDDA [17]. These algorithms involved greedy, genetic, heuristic, and dynamic programming algorithms, which could embody the mainstream of techniques in data allocations for heterogeneous memory. The number of evaluation rounds was between 1×10^3–1×10^4, which could avoid noises and ensure the correctness of the assessment.

Moreover, the comparisons mainly concerned two aspects, which were costs and strategy generation time. The first aspect would consider all possible computing resources utilized or required in data allocation manipulations the cost,

Table 1. Input configuration

	Memory #	Input data number						
		(a)	(b)	(c)	(d)	(e)	(f)	(g)
Setting 1	2	10	20	30	40	50	75	100
Setting 2	3	10	20	30	40	50	75	100
Setting 3	4	10	20	30	40	50	75	100
Setting 4	5	10	20	30	40	50	75	100
Setting 5	6	10	20	30	40	50	75	100

such as the energy consumption, latency, memory space, hardware availability, and probability. Therefore, the cost was counted in units for the purpose of comparisons in our evaluation. The other aspect examined the execution time of the algorithm, which could assess whether the algorithm was adoptable in practice. In most situations, an optimal algorithm required a longer execution time than sub-optimal/near-optimal algorithms. However, in some situations, an optimal algorithm might be adoptable due to the specific configuration, such as a low-dimensional memory setting or less input task amount. In our evaluations, the time length was counted in milliseconds.

Table 1 displayed a series of experiment configuration used in the evaluations. There were five settings given by the table, in which the configuration depended on the number of the memory type. This configuration fitted in various data allocation algorithms in distinct memory settings. For example, algorithm DAHS [15] mainly supported two heterogeneous memories, while the algorithm MDPDA [16] was designed for a three-heterogeneous memory setting. Thus, it implied that not all algorithms participated in all comparison evaluations. For instance, the DAHS algorithm might be involved in Setting 1 only; the OMDDA algorithm could be examined throughout all settings due to its multi-dimensional design.

In addition, under each setting, there were a few sub-settings configured by the number of input data packages/tasks. As shown in the table, sub-settings were given from (a) to (g), in which provided distinct input amounts as 10, 20, 30, 40, 50, 75, and 100, respectively. The reason for having these settings was to further simulate diverse application scenarios. In line with the comparison aspects, the condition of the inputs could impact on the performance of the algorithm. For instance, we assumed some near-optimal solutions could have a better performance within a specific scope of input task volume than other scopes, since the rate of obtaining optimal solutions might be higher. Another assumption could be that a dynamic programming could be more adoptable in a range of input tasks than other ranges due to the balance between the latency caused by the strategy generations and the cost saving due to the optimal solution. All these issues would be addressed in the empirical study.

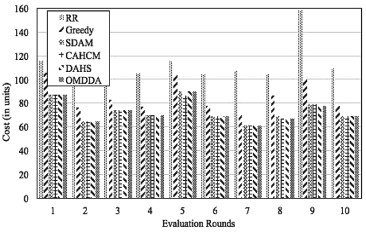
In summary, the experiment configuration in this empirical study highlighted the adaptabilities for various algorithms. The investigations would focus on a few featured algorithms throughout a series of comparisons and syntheses. The next

section would display partial results and present our main findings derived from our experiments.

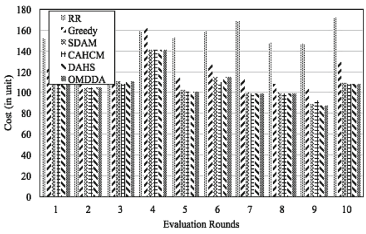
3.2 Experiment Results

We presented partial experiment results in this section, which aligned with settings given in Sect. 3.1. Figure 2 illustrated a number of figures that depicted partial results collected from settings 1-(a) and 1-(b). Under this setting, heterogeneous memory consisted of two memories. Figure 2a and b depicted cost comparisons based on ten-round evaluations that were randomly picked from 1×10^3 rounds. DAHS and DMDDA were two dynamic programming algorithms, which could output optimal solutions. SDAM had a high rate of obtaining optimal solutions under Setting 1. CAHCM had around 80% of gaining optimal solutions.

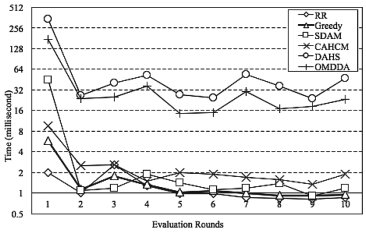
Figure 2c and d showed comparisons of the strategy generation time using logarithmic scale at 2. The results provided an observable difference between dynamic programming and other algorithms. Under Setting 1-(a), The average generation time lengths for DAHS and OMDDA were 67.53 and 37.99 ms, which were much longer than two heuristic algorithms, SDAM and CAHCM. The average generation time of SDAM and CAHCM was 5.56 ms and 2.64 ms, respectively. With the increase of the input data packages, the generation time of dynamic programming would be highly impacted while other algorithms had



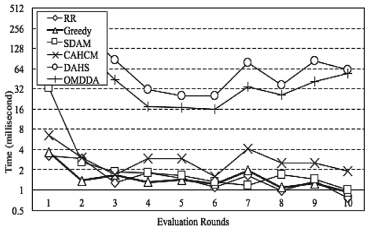
(a) Partial experiment results comparing cost levels collected from Setting 1-(a).



(b) Partial experiment results comparing cost levels collected from Setting 1-(b).



(c) Partial experiment results comparing strategy generation time collected from Setting 1-(a). (Logarithmic scale: 2)



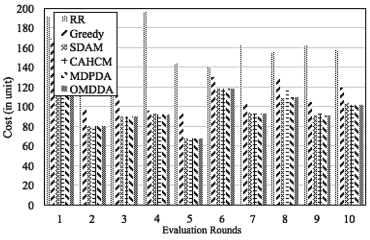
(d) Partial experiment results comparing strategy generation time collected from Setting 1-(b). (Logarithmic scale: 2)

Fig. 2. Partial experiment results collected from Setting 1.

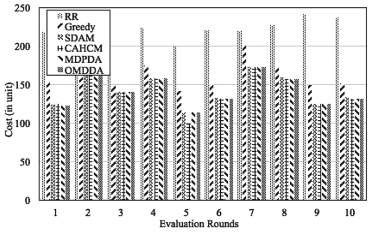
a limited impact. Under Setting 1-(b), DAHS and OMDDA's average strategy generation time became 93.57 ms and 62.68 ms. Increase rates were 38.59% and 64.97%, respectively. Having this phenomenon was that the computation complexity could be remarkably increased for dynamic programming but other algorithms had few impacts when the memory-dimension was two.

Next, Fig. 3 depicted some experiment results collected from Setting 2. Similar to result displays of Setting 1, we provided some results from two sub-settings to show the assessment trends. Two heuristic algorithms had a stable performance under this setting. We also observed that there was a great impact on the generation time of dynamic programming due to the increase of the memory types. There were two dynamic programming algorithms involved in this setting, namely, MDPDA and OMDDA. According to our statistics, MDPDA's average generation time was 90.50 ms and 201.99 ms for Setting 2-(c) and Setting 2-(d), respectively. OMDDA's average generation time was 62.62 ms and 133.13 ms under the same settings.

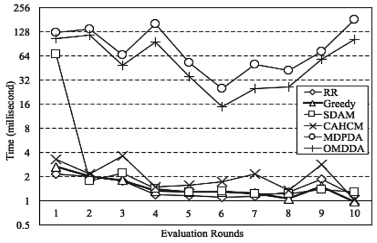
Furthermore, Fig. 4 provided some results obtained from Setting 3. Two subsettings were settings 3-(e) and 3-(g), which respectively had 75 and 100 input data packages. There was only one dynamic programming (OMDDA) due to its four-memory setting. The main finding under this setting was that implementing dynamic programming could result in a great latency. We observed that the average strategy generation time lengths for OMDDA were 0.33 s and 2.60 s



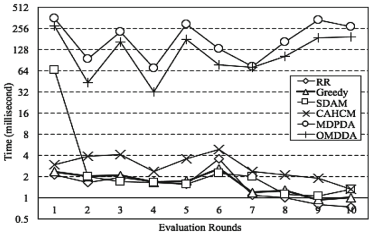
(a) Partial experiment results comparing cost levels collected from Setting 2-(c).



(b) Partial experiment results comparing cost levels collected from Setting 2-(d).

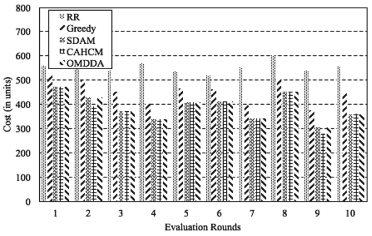


(c) Partial experiment results comparing strategy generation time collected from Setting 2-(c). (Logarithmic scale: 2)

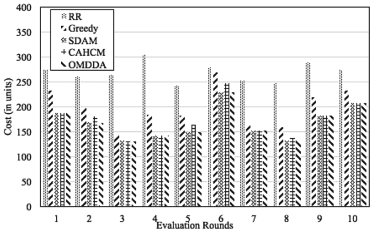


(d) Partial experiment results comparing strategy generation time collected from Setting 2-(d). (Logarithmic scale: 2)

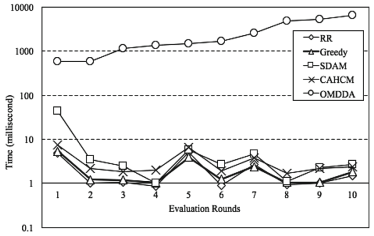
Fig. 3. Partial experiment results collected from Setting 2.



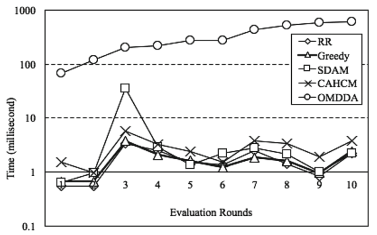
(a) Partial experiment results comparing cost levels collected from Setting 3-(e).



(b) Partial experiment results comparing cost levels collected from Setting 3-(g).



(c) Partial experiment results comparing strategy generation time collected from Setting 3-(e). (Logarithmic scale: 10)



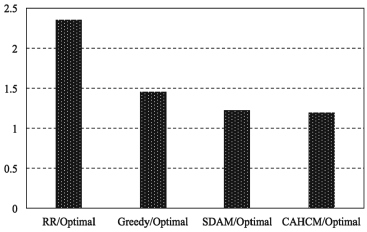
(d) Partial experiment results comparing strategy generation time collected from Setting 3-(g). (Logarithmic scale: 10)

Fig. 4. Partial experiment results collected from Setting 3.

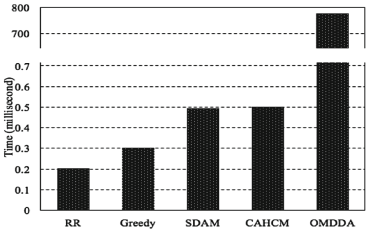
under Setting 3-(e) and Setting 3-(g), respectively. This result implied that using dynamic programming would have a great cost in creating strategies.

Moreover, Fig. 5 illustrated partial results that were based on a 1×10^4-round evaluation under Setting 4-(e). We used a rate deriving from "X/optimal" to examine the performance of saving costs by comparing a sub-/near-optimal solution to an optimal solution. Figure 5a depicted rate values on average for four algorithms. It was observable that two heuristic algorithms had a similar performance. The average of SDAM/Optimal was 1.23 and the average of CAHCM/Optimal was 1.20. Figure 5b depicted the strategy generation time for different algorithms on average. It showed that OMDDA was not suitable for small-sized data package due to its big cost in efficiency. The generation time for SDAM, CAHCM, and OMDDA on average were 0.496 ms, 0.500 ms, and 777.648 ms, according to our statistics. Figures 5c–f showed trend lines of the strategy generation time for four algorithms.

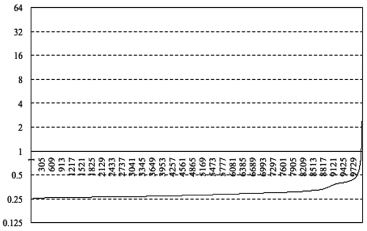
In addition, Fig. 6 depicted some results that focused on the strategy generation time, which were collected from settings 5(a), 5(c), 5(f), and 5(g). SDAM's generation time lengths on average for these settings were 6.72, 8.73, 45.09, and 51.78 ms, according to these 10-round evaluations. Under the same setting, CAHCM's average time lengths were 2.33, 2.46, 17.14, and 18.73 ms; OMDDA's were 38.49, 431.91, 10979.34, and 7141.49 ms. The reason for these results, based on our analysis, was that input data packages also had impact on individual strategy generation.



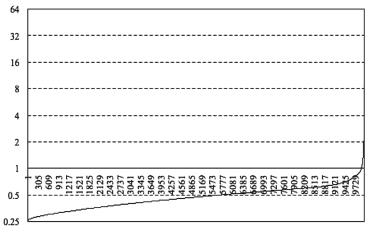
(a) Average comparisons between sub-optimal/near-optimal and optimal solutions.



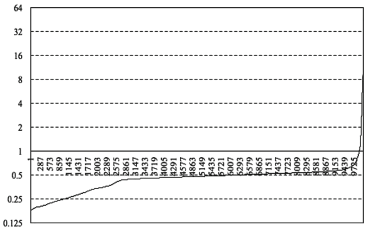
(b) Average strategy generation time comparisons.



(c) Trend line of the strategy generation time for greedy algorithm, counted in milliseconds. (Logarithmic scale: 2)



(d) Trend line of the strategy generation time for CAHCM algorithm, counted in milliseconds. (Logarithmic scale: 2)

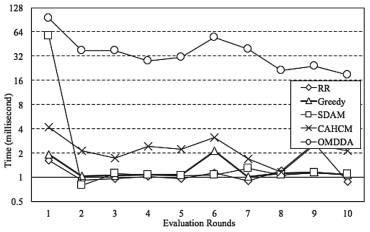


(e) Trend line of the strategy generation time for SDAM algorithm, counted in milliseconds. (Logarithmic scale: 2)

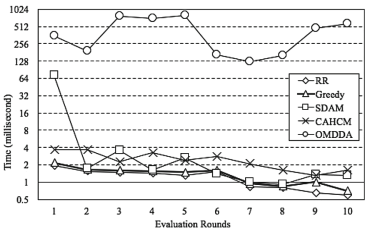
(f) Trend line of the strategy generation time for OMDDA algorithm, counted in milliseconds. (Logarithmic scale: 2)

Fig. 5. Partial experiment results from Setting 4-(e) with 1×10^4 rounds evaluations.

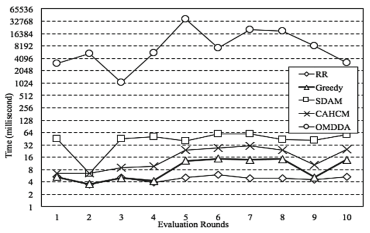
In summary, our findings included: (1) dynamic programming would be restricted when the number of input data packages was more than 50, based on our evaluation configuration; (2) both heuristic algorithms tested in our empirical study had stable performance in cost saving and required an acceptable strategy generation time; (3) dynamic programming was more suitable for those large-sized input tasks when considering the trade-off between the cost-saving advantage and the latency caused by the strategy generation.



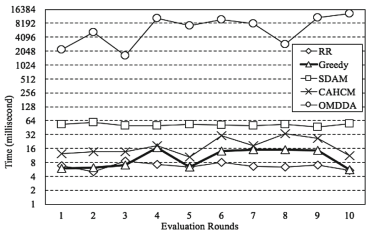
(a) Comparisons of partial evaluations' strategy generation time under Setting 5-(a). (Logarithmic scale: 2)



(b) Comparisons of partial evaluations' strategy generation time under Setting 5-(c). (Logarithmic scale: 2)



(c) Comparisons of partial evaluations' strategy generation time under Setting 5-(f). (Logarithmic scale: 2).



(d) Comparisons of partial evaluations' strategy generation time under Setting 5-(g). (Logarithmic scale: 2)

Fig. 6. Partial experiment results collected from Setting 5.

4 Conclusions

In this paper, we reviewed a few representative data allocation approaches for heterogeneous memory and completed an empirical study via a series of comparisons. The study gave future research a theoretical guidance about the adaptability. Our main findings showed that dynamic programming was restricted by the volume of input tasks as well as the amount of memory types. Two heuristic algorithms in our investigations had a stable performance, which had a higher-level adaptability.

References

1. Meswani, M., Blagodurov, S., Roberts, D., Slice, J., Ignatowski, M., Loh, G.: Heterogeneous memory architectures: a HW/SW approach for mixing die-stacked and off-package memories. In: IEEE 21st International Symposium on High Performance Computer Architecture, Burlingame, CA, USA, pp. 126–136. IEEE (2015)
2. Agarwal, N., Nellans, D., Stephenson, M., O'Connor, M., Keckler, S.: Page placement strategies for GPUs within heterogeneous memory systems. ACM SIGPLAN Not. **50**(4), 607–618 (2015)
3. Gai, K., Qiu, M., Sun, X.: A survey on fintech. J. Netw. Comput. Appl. **PP**, 1 (2017)

4. Gai, K., Qiu, M., Zhao, H.: Energy-aware task assignment for mobile cyber-enabled applications in heterogeneous cloud computing. J. Parallel Distrib. Comput. **111**, 126–135 (2018)
5. Gai, K., Qiu, M., Zhao, H., Sun, X.: Resource management in sustainable cyber-physical systems using heterogeneous cloud computing. IEEE Trans. Sustain. Comput. **PP**(99), 1–13 (2017)
6. Gai, K., Qiu, M., Zhao, H., Tao, L., Zong, Z.: Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. J. Netw. Comput. Appl. **59**, 46–54 (2015)
7. Gai, K., Qiu, M.: Blend arithmetic operations on tensor-based fully homomorphic encryption over real numbers. IEEE Trans. Industrial Inf. **PP**(99), 1 (2018)
8. Sha, E., Chen, X., Zhuge, Q., Shi, L., Jiang, W.: A new design of in-memory file system based on file virtual address framework. IEEE Trans. Comput. **65**(10), 2959–2972 (2016)
9. Gai, K., Qiu, M., Liu, M., Xiong, Z.: In-memory big data analytics under space constraints using dynamic programming. Future Gener. Comput. Syst. **PP**, 1 (2018)
10. Hahne, E.: Round-robin scheduling for max-min fairness in data networks. IEEE J. Sel. Areas Commun. **9**(7), 1024–1039 (1991)
11. Cully, B., Wires, J., Meyer, D., Jamieson, K., Fraser, K., et al.: Strata: high-performance scalable storage on virtualized non-volatile memory. In: Proceedings of the 12th USENIX Conference on File and Storage Technology, San Jose, CA, USA, pp. 17–31 (2014)
12. Qiu, M., Zhong, M., Li, J., Gai, K., Zong, Z.: Phase-change memory optimization for green cloud with genetic algorithm. IEEE Trans. Comput. **64**(12), 3528–3540 (2015)
13. Gai, K., Qiu, M., Zhao, H., Qiu, L.: Smart energy-aware data allocation for heterogeneous memory. In: IEEE 18th International Conference on High Performance Computing and Communications, Sydney, NSW, Australia, pp. 136–143. IEEE (2016)
14. Gai, K., Qiu, M., Zhao, H.: Cost-aware multimedia data allocation for heterogeneous memory using genetic algorithm in cloud computing. IEEE Trans. Cloud Comput. **PP**(99), 1–11 (2016)
15. Hu, J., Xue, C., Zhuge, Q., Tseng, W., Sha, E.: Data allocation optimization for hybrid scratch pad memory with SRAM and nonvolatile memory. IEEE Trans. VLSIS **21**(6), 1094–1102 (2013)
16. Qiu, M., Chen, Z., Liu, M.: Low-power low-latency data allocation for hybrid scratch-pad memory. IEEE Embed. Syst. Lett. **6**, 69–72 (2014)
17. Zhao, H., Qiu, M., Chen, M., Gai, K.: Cost-aware optimal data allocations for multiple dimensional heterogeneous memories using dynamic programming in big data. J. Comp. Sci. **PP**, 1 (2016)

NEM: A NEW In-VM Monitoring with High Efficiency and Strong Isolation

Jingjie Qin(✉), Bin Shi, and Bo Li

School of Computer Science and Engineering,
Beihang University, Beijing, China
{qinjj,shibin,libo}@act.buaa.edu.com

Abstract. VMI technology is proposed to protect virtual machine and prevent it from attacking by malware. Although VMI technology can provide out-of-VM isolation to ensure the security of monitors, the overhead of context switching between the guest VMs and the hypervisor for each monitor point makes this approach wasteful in many application scenarios. On the other hand, semantic gap of extracting meaningful information from the guest is a problem need to be optimized for the VMI technology. In this paper, we present None-Exit Monitoring (NEM), a framework that can do the monitoring inside the guest to avoid overhead of VM-exit and VM-entry switching, and it can also provide strong isolation between the guest and the monitor tools. In NEM, we use two new hardware virtualization assistant features: Intel VT VMFUNC and #VE. NEM can provide isolated memory views and strict limits of privileges while using EPTP-switching to realize world-switches instead of root/non-root switching, which can reduce overhead of invocation of the monitor. On the other hand, IN-VM monitoring can achieve richer information on a virtual machine, which can enhance the capability of the monitor. To support EPTP-switching function of VMFUNC and #VE exception, we patch the open source KVM. We also implement NEM in KVM and evaluate its functionality and efficiency. Experimental result has shown that NEM can satisfy the security requirement of a virtual machine monitor and can greatly improve the efficiency.

Keywords: Virtual machine monitor · Virtualization security
VMFUNC · EPTP-switching · Virtual exception

1 Introduction

Problems. Virtualization technology has been widely used in cloud computing, and can make full use of resources in the data center improving efficiency of operation and reduce cost. In the cloud platform especially the public cloud platform, a virtual machine may support many services for different users. And if the virtual machine has some security vulnerability, the vulnerability will spread rapidly and will influence the subsequent use. Besides, the virtual machines exist in many physical machines, without strong isolation the vulnerability will

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 396–405, 2018.
https://doi.org/10.1007/978-3-319-73830-7_39

spread more easily. The lack of intra-process isolation will allow attackers to bypass the state-of-the-art security defenses [11]. The virtual machine monitor deployed outside the virtual machine can provide strong isolation between guests to ensure that malware can't influence the security tools. VMI (Virtual Machine Introspection) [1,2], which can introspect guest VMs internal states from the VMM (Virtual machine monitor), has been widely used for monitoring [3] and instruction detection [4]. The widespread adoption of VMI has been hampered by the semantic gap: the gap of extracting high-level semantic information from low-level data sources. To solve the semantic gap, researchers have proposed many different VMI technologies, such as Virtuso [5], Min-c [6], XenAccess [2] and so on.

However, the VMI tools and the guest VM have competitions in the running process of guest, some introspection systems are intrusive in stopping the guest for tens of seconds to do the introspection eagerly [5]. Thus, the VMI tools will be ineffective and strongly influence the guest performance. And some security approaches require to monitor executing events frequently, such as LSM [12] and SELinux that hook into a large number of kernel events enforce specific security policies. So the users need to do a tradeoff between performance overhead and security protection, as the frequent introspection will cause service disruption and performance overhead, while infrequent introspection will miss some undetected security attack. On the other hand, VMI technology has been limited by the semantic gap: extracting high-level meaningful information from low-level data sources, which is a barrier to the development and deployment of VMI technology.

Solution. In this paper, we introduced a new monitoring mechanism called NEM (None-Exit Monitoring), which can do the security monitoring without exit the guest virtual machine and isolate the VM and the monitor strongly. NEM uses the Intel hardware extension called VMFUNC [10] and the new exception type called Virtualization Exception (#VE). #VE can be used to catch EPT exceptions and handle them in guest OS directly without introducing extra EPT violations VM-Exit overhead. And VMFUNC can provide VM functions in guest OS (VMX non-root mode). This instruction allows software in VMX non-root operation to invoke a VM function, which is processor functionality enabled and configured by software in VMX root operation (host). Currently, Intel defines only one VM function called EPTP-switching: which can allows software in guest to load a new EPT pointer then using a new EPT paging-structure hierarchy. The EPT Pointer is selected from a potential EPTP values configured in advance. By using #VE and EPTP-switching, we can intercept a running application without VM-Exit and do the security analysis in an isolated view.

Contributions. In summary, this paper makes the following contributions:

- A monitor inside the virtual machine with strong physical isolation by two views of EPT with different privileges. The isolation can do the detection in separate pages and prevent the VM from attacking by malwares to keep safety.

- NEM can do the monitoring inside the guest to avoid overhead of VM-exit and VM-entry switching, which can improve 60% overhead than the out-of-VM monitoring. On the other hand, IN-VM monitoring can achieve richer information of a virtual machine, which can enhance the capability of the monitor.
- We patch the open source KVM to support EPTP-switching of VMFUNC and #VE exception.
- An implementation of virtual machine monitor based on the mechanism of NEM and experiment the performance of the new monitor comparing with the traditional VMI.

The rest of the paper is organized as follows: the second section briefly describe the related work of EPTP-switching how it applied in monitoring. Sections 3 and 4 illustrate the design and detailed implementation of EPTP-switching and #VE in KVM and the isolated monitoring of NEM. Experimental evaluation of our solution and result analysis is presented in Sect. 5. Section 6 draws a conclusion and discusses future directions.

2 Related Work

2.1 Xen Altp2m

Xen has added the feature called altp2m to make stealthy monitoring [13]. Altp2m allows Xen to create more than one EPT for each guest with the support of Intel hardware. The Xen based DRAKVUF Dynamic Malware Analysis [7] used this new feature in combination with an additional; technique to maximum effect. DRAKVUF can write breakpoint instructions into the guests memory at code-locations of interest by enabling the trapping of debugging instructions into the hypervisor without breaking other uninterested functions using the same page. By this way, we will only get an event for precisely the code-location we are interested in this method effectively reduces the overhead. Altp2m can change the page mapping to solve this problem. The same guest physical memory can be setup to be backed by different pages in the different views, which can hide the presence of the breakpoints specific to each vCPU. All without having paused any of the other vCPUs or having emulated.

2.2 SeCage

SeCage [8] retrofits commodity hardware virtualization extensions to support efficient isolation of sensitive code manipulating critical secrets from the remaining code. SeCage separate control and data plane using VMFUNC mechanism and nested paging in Intel processors to transparently provide different memory views for different compartments, and allow low-cost and transparent invocation across domains without hypervisor intervention. SeCage inits two EPTs called EPT-N and EPT-S. EPT-N is used for the entire guest VM and EPT-S is used

for each protected secret compartment. SeCage classifies the memory into two parts: data and code. EPT-S maps all data including secrets while EPT-N has data other than the secrets. And for the code section, the trampoline code is mapped to these two EPTs as read-only. The EPT-S contains the sensitive function code, while the EPT-N maps code other than the sensitive functions. When code invokes a function in a secret compartment, instead of directly calling the sensitive function, it calls into the corresponding trampoline code, which at first executes the VMFUNC instruction to load memory of secret compartment, then the stack pointer is modified to point to the secure stack page. After the functions returns, the trampoline code will restore the ESP to the previous stack frame location, executes VMFUNC and returns the result to the caller.

3 Design

3.1 Solution Overview

Our solution in Fig. 1 is to use #VE (virtual exception) to avoid VM-exit when handle intercept events and do the monitoring in guest to achieve rich semantic information. After using #VE to catch EPT violation, NEM will use the No.0 instruction of VMFUNC to switch the EPT view to correct the violation to resume the application.

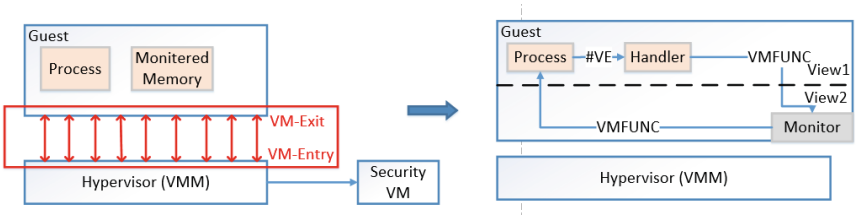


Fig. 1. The traditional VMI and the monitoring mechanism of NEM

3.2 EPTP-Switching

EPTP switching is a new VM Functions to execute VMFUNC instruction in non-root mode without VM-Exit. The RAX register will save the number of VM function and EPTP switching is the number 0 VM Function. EPTP switching will allow software in guest VM to load a new EPT pointer (EPTP), thereby establishes a different EPT paging-structure view. The EPTP is selected from an EPTP list prepared before by the hypervisor and different EPTP point to different EPT view, by which the same GPA (guest physical address) are translated to different HPA (host physical address). Specifically, the value of ECX is used to select an entry from the EPTP list, the 4-KByte structure referenced by the EPTP-list address.

3.3 Virtual Exception

Virtual Exception is an EPT-violation exceptions which can handle EPT violations instead of VM exits. A virtualization exception can occur only in VMX non-root operation. When the processor encounters a virtualization exception, it saves information about the exception to the virtualization-exception information area. After saving virtualization-exception information, the processor delivers a virtualization exception as it did to any other exception.

3.4 Monitoring

Figure 2 shows the process of intercept, switching and monitoring. NEM uses #VE to intercept the program and then switch the EPT to the View2 and do the monitoring.

Firstly, the application is running in view1. To intercept the process, NEM modified the first instruction to be incorrect in a section of code, which will cause a #VE exception without VM-exit. Handler1 handles #VE exception and using EPTP switching to switch to another EPT View2. View2 is a secure view which is isolated from View1 and has the code of monitoring. After switching, NEM will run the monitoring code to check the guest operation. After finishing monitoring, NEM will correct the first instruction and return to continue execute it in the View2. The second instruction is also modified to be incorrect in order to switch to View1 after monitoring. After finishing the first instruction, the second instruction will cause a #VE exception and enter into Handler3 to execute EPTP switching to switch back to handler1 of View1 and then return to continue the process.

After switching to the secure EPT view, NEM will check the program operations saved in the register to see whether it has some sensitive operations such as some sensitive system call, I/O file and high privilege request. The monitoring code is invoked by specific secure strategy. For example, when a process is created or switched, the monitor should scan and analysis the process chain of the guest. When NEM detects some malware operation, it will return to the View1 and stop the process before the execution of first instruction. Specifically, monitoring strategy is beyond the scope of this article and will not be introduced in detail.

3.5 EPTP-Switching

To isolate View1 from View2, NEM will set different privileges on the two views. Figure 3 shows the privileges in the two views:

- When program is running in View1, the code in View1 doesn't have the access of read, write or execute for the content of View2. And the modification of the first instruction (#VE hook) is write-protect.
- When view has switched to View2, the monitor code has the write, read and execute privilege to the code and data in View1. As the program in View1 is non-executable, its codes don't have the read, write or execute privilege to the content of View2.

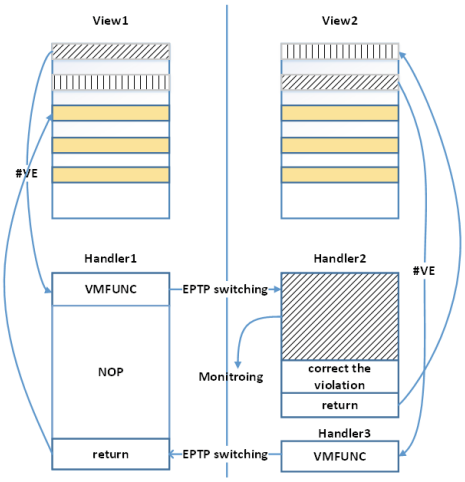


Fig. 2. The process of monitoring in guest using #VE and VMFUNC

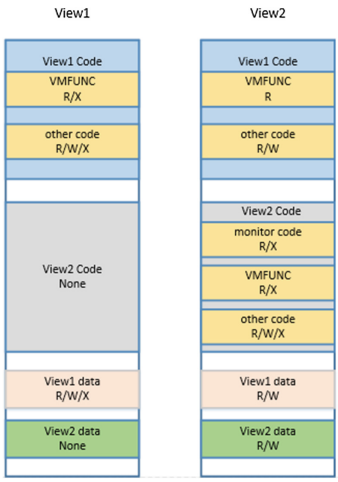


Fig. 3. Memory privileges of View1 and View2

- The VMFUNC instruction in View1 and View2 is write-protect to keep safe from malwares malicious modification.

4 Implementation

NEM uses KVM as the hypervisor, deployed in Ubuntu 14.04 64-bit LTS, and the kernel version is 4.10.2. Currently, KVM can't support EPTP-switching and #VE, so we modified KVM to support these two functions. And then realize a monitor prototype system in the guest virtual machine.

To enable EPTP-switching, KVM should configure the VMCS field. The VM functions and EPTP switching bit should be set 1 and store the configured EPTP in the VMCS field. EPTP list is prepared in advance. To achieve multiple EPTs, we modified the structure of page table and use a pointer array ept_root_hpa_list to represent multi-EPTP. The page table of number 0 (default) EPT is filled in and other EPTPs point to different tables which will be filled in by a page fault later on. EPTP switching support at most 512 EPTP entries, thus NEM can create 512 compartments for the guest.

The EPT-violation #VE VM-execution control is set 1. Virtualization exceptions save data into the virtualization-exception information area. And the 16-bit value EPTP index is saved in the Virtualization-Exception information and then saved into the VMCS. In the guest, we need to handle #VE by cloning the IDT and hook the vector #20.

When start a virtual machine, NEM will be deployed in the guest. The monitoring code is compiled in the secure EPT in advance. When launching an

application, the application will wake up NEM to do the monitoring. Firstly, the start virtual address and the information of the application will be passed to the NEM and NEM will invoke NEM_INIT to inject hook code in the application in EPT View1 and hook the first instruction of View1 and the second instruction of View2 by #VE exception.

At runtime, the code of View1 and View2 is execute sequentially. When meeting a hook in the application, #VE exception will be handled in Handler1 of View1. Handler1 has only one instruction VMFUNC and then the EPT will be switched to Handler2. Handler2 is configured in View2 and has privilege of read and write to the code of the application. And if the code of the application try to read or write the code in View2, a VM-exit will be occurred due to EPT violation and reported to the VMM. In this case, the hypervisor should stop the application execution and inform the user of this unsafe access request. And if the monitor codes find some sensitive operation, the application should also be paused and inform the user about the sensitive part.

After checking the application, it can continue running until stopped by the hook in View2. This hook is used for the EPT to switch to View1. To ensure the consistency of View1 and View2, the blank codes in View1 is filled in NOP instructions.

5 Evaluation

5.1 Secure Evaluation

NEM can satisfy the security requirement of a virtual machine monitor, which can be described in two aspects:

1. NEM can keep its own safety from the attack of the malwares and isolate itself from the program. Due to the multi-EPTs, the monitor has the high privilege to the program while the program has the low privilege to the EPT view of monitor. If the program wants to read or write the page of View2, there will occur a VM-exit and report to the VMM about the unauthorized access. On the other hand, NEM can save the process of extracting high-level semantic information from low-level data sources, by which can achieve richer information of the guest.
2. NEM can support effective monitoring to prevent malware from attacking the guest virtual machine. When #VE exception is triggered, NEM will check the legality of the hook and provide write-protect to the hook in case of malicious modification to the hook. NEM will also check the program of sensitive system call or high privilege request. However, some malwares can modify the original guest address of hook or VMFUNC to other addresses by modifying the mapping of the kernel page, which will bypass the monitor. To solve this problem, we should monitor the modification of kernel pages by ensuring the address to keep same in a period of time using a timer monitor. If malwares invoke some sensitive system calls or modify kernel page of the hook, NEM should kill the program in secure EPT and report to the user about the attack.

5.2 Performance Evaluation

NEM can do the monitoring inside guest without VM-exit, which can reduce substantial overhead. We test the overhead of an integrated monitoring process and compare the overhead of using monitor "out-of-VM" with NEM. All experiments are done on a machine with 4 Intel Core running at 3.2 GHz and with 8 GB memory. The host kernel for KVM is Linux 4.10.2 and the guest kernel is Linux 4.4.0. Both the OS of host and guest are Ubuntu 14.04 64-bit LTS.

5.2.1 Microbenchmarks

NEM uses EPTP-switching to switch from guest to the monitor, which can reduce the overhead of world-switched between the guest VMs for each invocation of the monitor through VM-exit. We compare the overhead of EPTP-switching and the switching through VM-exit and VM-entry.

Table 1. Overhead of the two switching mechanism (μs)

Times	EPTP-switching	VM-exit/VM-entry
1	0.1995	1.573
2	0.2885	1.248
3	0.3807	1.352
4	0.2838	1.375
5	0.2890	1.414
Average	0.2883	1.392

Table 1 shows the overhead of EPTP-switching and VM-exit to VM-entry. We can see that the context switch between root and non-root mode will cost more time than EPTP-switching. Thus, NEM saves the process of switching between root and non-root mode by using EPTP-switching without VM-exit, which will greatly improve the efficiency of monitoring.

5.2.2 Real-World Performance

To get a more reliable analysis, we run NEM on a virtual machine and experiment the cost of different system calls comparing with the native Linux and the out-of-VM monitoring. We use Lmbench [9] to benchmark the systems by invoking some system calls to test the monitoring efficiency.

Table 2 shows the common system calls in Lmbench benchmark system. "null call" means a getpid call, "open/close" shows the process of open and close a file. These operations are all invoked in guest without the trap into the hypervisor root mode. As a result, in NEM and Out-of-VM monitoring, these system calls is close to Native Linux. Fork will create new process, which needs to map memory frequently. And exec will replace the old process with a new process, which

Table 2. Overhead of system calls in different system by Lmbench (μs)

Benchmarks	Native Linux	NEM	Out-of-VM monitoring
null call (getppid)	0.13	0.13	0.14
open/close	0.74	0.74	0.74
fork	71.9	112.3	287.3
exec	221.0	429.0	1123.0
sh	516.0	823.1	2511.2

will modify the data and code sections. These system calls need to trap into root mode to do memory virtualization. Using NEM can use some instructions inside the guest without VM-exit and VM-entry switching between guest and hypervisor, which can improve efficiency dramatically.

6 Conclusion and Future Work

In this paper, we presented NEM, a novel approach that can satisfy the security of a virtual machine monitor and reduce the overhead of root/non-root switching when intercepting the processes. NEM can monitor the processes inside the guest to hook system call without VM-exit, which can greatly reduce the overhead of monitoring. On the other hand, NEM can save the process of transfer low level semantic information to high level information, and can achieve richer information of a guest virtual machine. NEM was shown to be useful by protecting virtual machine from real software attack, and can improve monitoring performance dramatically.

We plan to improve NEM to be higher available by offering programmable interface to users to develop his own monitor by modifying rules of monitoring. And we plan to do more works about the isolation safety of a monitor.

Acknowledgements. This work is supported by the 863 project 2015AA01A202.

References

1. Garfinkel, T., Rosenblum, M.: A virtual machine introspection based architecture for intrusion detection. In: NDSS, vol. 3, pp. 191–206, February 2003
2. Payne, B.D., Martim, D.D.A., Lee, W.: Secure and flexible monitoring of virtual machines. In: Twenty-Third Annual Computer Security Applications Conference, ACSAC 2007, pp. 385–397. IEEE, December 2007
3. Srinivasan, D., Wang, Z., Jiang, X., Xu, D.: Process out-grafting: an efficient out-of-vm approach for fine-grained process execution monitoring. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, pp. 363–374. ACM, October 2011
4. Payne, B.D., Carbone, M., Sharif, M., Lee, W.: Lares: an architecture for secure active monitoring using virtualization. In: IEEE Symposium on Security and Privacy, SP 2008, pp. 233–247. IEEE, May 2008

5. Dolan-Gavitt, B., Leek, T., Zhivich, M., Giffin, J., Lee, W.: Virtuoso: narrowing the semantic gap in virtual machine introspection. In: 2011 IEEE Symposium on Security and Privacy (SP), pp. 297–312. IEEE, May 2011
6. Inoue, H., Adelstein, F., Donovan, M., Brueckner, S.: Automatically bridging the semantic gap using C interpreter. In: 6th Annual Symposium on Information Assurance (ASIA11), p. 51, June 2011
7. DRAKVUF Malware Analysis System. https://drakvuf.com/
8. Liu, Y., Zhou, T., Chen, K., Chen, H., Xia, Y.: Thwarting memory disclosure with efficient hypervisor-enforced intra-domain isolation. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1607–1619. ACM, October 2015
9. McVoy, L.W., Staelin, C.: lmbench: portable tools for performance analysis. In: USENIX Annual Technical Conference, pp. 279–294, January 1996
10. Guide, P.: Intel® 64 and IA-32 Architectures Software Developers Manual, vol. 3B. System programming Guide, Part 2 (2011)
11. Lu, K., Song, C., Lee, B., Chung, S.P., Kim, T., Lee, W.: ASLR-Guard: stopping address space leakage for code reuse attacks. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 280–291. ACM, October 2015
12. Morris, J., Smalley, S., Kroah-Hartman, G.: General security support for the linux kernel. In: USENIX Security Symposium, Linux Security Modules, August 2002
13. Xen altp2m. https://blog.xenproject.org/2016/04/13/stealthy-monitoring-with-xen-altp2m/

k-CoFi: Modeling *k*-Granularity Preference Context in Collaborative Filtering

Yunfeng Huang[1], Zixiang Chen[1], Lin Li[1], Weike Pan[1(✉)], Zhiguang Shan[2], and Zhong Ming[1(✉)]

[1] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
{huangyunfeng2017,chenzixiang2016,lilin20171}@email.szu.edu.cn, {panweike,mingz}@szu.edu.cn
[2] Information Research Department, State Information Center, Beijing, China
shanzhiguang@263.com

Abstract. Collaborative filtering (CF) is a highly applicable technology for predicting a user's rating to a certain item. Recently, some works have gradually switched from modeling users' rating behaviors alone to modeling both users' behaviors and preference context beneath rating behaviors such as the set of other items rated by user u. In this paper, we go one step beyond and propose a novel perspective, i.e., k-granularity preference context, which is able to absorb existing preference context as special cases. Based on this new perspective, we further develop a novel and a generic recommendation method called k-CoFi that models k-granularity preference context in collaborative filtering in a principled way. Empirically, we study the effectiveness of factorization with coarse granularity, fine granularity and smooth granularity, and their complementarity, by applying k-CoFi to three real-world datasets. We also obtain some interesting and promising results and useful guidance for practitioners from the experiments.

Keywords: k-granularity preference context · Matrix factorization Collaborative filtering

1 Introduction

Collaborative filtering (CF) or (user, item) rating prediction is one of the most well-known and well-studied technology in recommender systems largely because of its high applicability to different applications and domains. Users' preference modeling is one of the most fundamental issues in developing advanced CF methods such as factorization machine [4] and deep learning [1].

Recently, some works have switched to modeling both users' preferences and their preference context contained in the rating data, instead of modeling users' preferences alone. For example, in SVD++ [2], in order to predict the rating of a (user, item) pair (u, i), besides the latent representation of user u and item i,

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 406–416, 2018.
https://doi.org/10.1007/978-3-319-73830-7_40

the latent representation of other items rated by user u are also included as the preference context. The justification is that two users with similar sets of rated items are likely to have similar preference patterns. As another instance, in MF-MPC (matrix factorization with multiclass preference context) [3], the preference context of rated items in SVD++ [2] is further refined according to the rating values of the user to other items, which is able to capture more accurate preference context.

However, these works are studied independently and separately, and have not been studied in a generic and unified framework. For this reason, their relationship and potential of combination also have not been fully exploited. In this paper, we propose a novel perspective, i.e., k-granularity preference context, which provides a unified view and absorbs different preference context in existing works as special cases. Based on this new perspective, we develop a novel algorithm that models k-granularity preference context in collaborative filtering (k-CoFi). Furthermore, we study the complementarity of different k-granularity preference context in factorization-based algorithms.

In order to study the effectiveness and complementarity of different k-granularity preference context. We conduct extensive empirical studies on three real-world datasets with several state-of-the-art methods. The results clearly show the effectiveness of fine granularity and smooth granularity, and their complementarity in rating prediction.

We summarize our main contributions as follows: (i) we propose a novel perspective for modeling users' preference context, i.e., k-granularity preference context; (ii) we develop a novel and generic recommendation algorithm that models k-granularity preference context in collaborative filtering, i.e., k-CoFi; and (iii) we study the effectiveness and complementarity of different preference context and have some interesting observations.

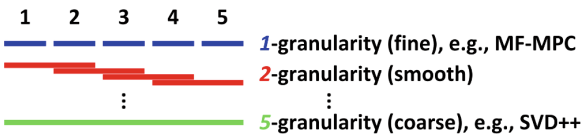


Fig. 1. Illustration of different granularity of 5-star numerical ratings in collaborative filtering.

2 k-Granularity Preference Context

In this section, we describe a novel perspective, i.e., k-granularity preference context, for modeling users' preference context in collaborative filtering. Firstly, we formally define k-granularity preference context in collaborative filtering with categorical rating values. Secondly, we review two state-of-the-art collaborative filtering methods, i.e., SVD++ [2] and MF-MPC [3], from the perspective of virtual user profiles based on 5-granularity preference context and 1-granularity preference context, respectively.

We list some commonly used notations in Table 1.

Table 1. Some notations and explanations.

n	user number
m	item number
$u \in \{1, 2, ..., n\}$	user ID
$i \in \{1, 2, ..., m\}$	item ID
$\mathbb{V} = \{v\}$	a set of rating values
$r_{ui} \in \mathbb{V}$	rating assigned by user u to item i
$\mathcal{R} = \{(u, i, r_{ui})\}$	training data of rating records
$\mathcal{I}_u$	a set of items w.r.t. u
$\mathcal{I}_u^v$	a set of items w.r.t. u and v
$\mathcal{K} = \{k\}$	a set of k-granularity
$\mathcal{S}_k = \{g\}$	a set of rating groups w.r.t. k
$\mathcal{I}_u^{kg}$	a set of items w.r.t. u, k and g
d	size of latent vector
$\mu \in \mathbb{R}$	global average rating value
$b_u \in \mathbb{R}$	user bias of user u
$b_i \in \mathbb{R}$	item bias of item i
$U_{u\cdot} \in \mathbb{R}^{1\times d}$	latent vector of user u
$V_{i\cdot} \in \mathbb{R}^{1\times d}$	latent vector of item i
$W_{i'\cdot}^{kg} \in \mathbb{R}^{1\times d}$	latent vector of item i' w.r.t. k and g
$\hat{r}_{ui}$	predicted rating of user u to item i
T	iteration number in the algorithm
α	weight on regularization terms
λ	weight for combining two k-CoFi

2.1 Definition of Preference Context

In collaborative filtering, we usually have a set of multiclass rating values such as $\{1, 2, 3, 4, 5\}$ in MovieLens 100K and MovieLens 1M. In order to model a rating of a user u to an item i, we can represent it as follows,

$$P(r_{ui}|(u, i, \mathcal{C})), \tag{1}$$

where $\mathcal{C}$ denotes the preference context from the rating data itself rather than some auxiliary temporal or spatial context. In particular, the context can be (i) $\mathcal{C} = \emptyset$ in PMF [6]; (ii) $\mathcal{C} = \mathcal{I}_u \backslash \{i\}$ denoting the rated items (excluding item i itself) by user u in SVD++ [2]; and (iii) $\mathcal{C} = \mathcal{I}_u^v \backslash \{i\}, v \in \mathbb{V}$ denoting the rated items with different rating values $v \in \mathbb{V}$ (excluding item i itself) by user u in MF-MPC [3].

We may view the preference context from a new perspective, i.e., from coarse granularity to fine granularity. For example, (i) $\mathcal{C} = \emptyset$ represents none context, i.e., 0-granularity; (ii) $\mathcal{C} = \mathcal{I}_u \backslash \{i\}$ represents 5-granularity meaning without distinguishing the rating values; and (iii) $\mathcal{C} = \mathcal{I}_u^v \backslash \{i\}, v \in \mathbb{V}$ represents 1-granularity meaning treating each rating value separately. We illustrate this new perspective in Fig. 1.

Following this line of discussion, we propose a novel and generic preference context, i.e., k-granularity preference context, in which k consecutive rating values are treated as one group of ratings without differences. For example, the set of rating groups in 5-granularity and 1-granularity are $\mathcal{S}_5 = \{\{1, 2, 3, 4, 5\}\}$ and $\mathcal{S}_1 = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$, respectively.

We can see that $\mathcal{S}_5$ and $\mathcal{S}_1$ are two extreme cases, one for coarse granularity and the other for fine granularity. Naturally, we may consider the preference context between these two cases, e.g., 2-granularity preference context with $\mathcal{S}_2 = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}\}$. We can take the 2-granularity preference context as a smooth preference context, where we do not distinguish two consecutive rating values. This makes sense because sometimes the difference between two consecutive rating values may not be so significant for a user, especially when a user u is affected by his or her own mood in different situations.

In the following two subsections, we will show how the k-granularity preference context is modeled for rating prediction in two representative works, i.e., SVD++ [2] and MF-MPC [3]. In particular, we will represent the modeling technique by a virtual user profile in each case.

2.2 Virtual User Profile in SVD++

In order to capture the preference context of rated items by the current end user u, a virtual user profile is introduced in SVD++ [2],

$$\begin{aligned} \tilde{U}_{u\cdot}^{\text{SVD++}} &= \frac{1}{\sqrt{|\mathcal{I}_u \backslash \{i\}|}} \sum_{i' \in \mathcal{I}_u \backslash \{i\}} W_{i'\cdot} \\ &= \sum_{g \in \mathcal{S}_5} \frac{1}{\sqrt{|\mathcal{I}_u^{5g} \backslash \{i\}|}} \sum_{i' \in \mathcal{I}_u^{5g} \backslash \{i\}} W_{i'\cdot}^{5g} \\ &= \sum_{k \in \{5\}} \sum_{g \in \mathcal{S}_k} \frac{1}{\sqrt{|\mathcal{I}_u^{kg} \backslash \{i\}|}} \sum_{i' \in \mathcal{I}_u^{kg} \backslash \{i\}} W_{i'\cdot}^{kg} \end{aligned} \tag{2}$$

It is clear that the virtual user profile $\tilde{U}_{u\cdot}^{\text{SVD++}}$ in Eq. (2) is based on the 5-granularity preference context.

2.3 Virtual User Profile in MF-MPC

Similarly, in order to model the fine-granularity preference context of rated items with different values $v \in \mathbb{V}$, a sophisticated virtual user profile of user u is used in MF-MPC [3],

$$\begin{aligned}\tilde{U}_{u\cdot}^{\text{MF-MPC}} &= \sum_{v\in\mathbb{V}} \frac{1}{\sqrt{|\mathcal{I}_u^v \backslash \{i\}|}} \sum_{i'\in\mathcal{I}_u^v\backslash\{i\}} W_{i'\cdot}^{v} \\ &= \sum_{g\in\mathcal{S}_1} \frac{1}{\sqrt{|\mathcal{I}_u^{1g} \backslash \{i\}|}} \sum_{i'\in\mathcal{I}_u^{1g}\backslash\{i\}} W_{i'\cdot}^{1g} \\ &= \sum_{k\in\{1\}} \sum_{g\in\mathcal{S}_k} \frac{1}{\sqrt{|\mathcal{I}_u^{kg} \backslash \{i\}|}} \sum_{i'\in\mathcal{I}_u^{kg}\backslash\{i\}} W_{i'\cdot}^{kg} \end{aligned} \tag{3}$$

We can see that the virtual user profile $\tilde{U}_{u\cdot}^{\text{MF-MPC}}$ precisely captures the information encoded in the 1-granularity preference context.

3 *k*-CoFi

3.1 Problem Definition

In collaborative filtering, the main task is to learn users' preferences from (user, item, rating) triples in the training data, and then predict the rating of each (user, item) pair in the test data.

3.2 Preference Context in *k*-CoFi

With the k-granularity preference context, we have the generic latent representation of a virtual user profile as follows,

$$\tilde{U}_{u\cdot}^{k\text{-gran}} = \sum_{k\in\mathcal{K}} \sum_{g\in\mathcal{S}_k} \frac{1}{\sqrt{|\mathcal{I}_u^{kg} \backslash \{i\}|}} \sum_{i'\in\mathcal{I}_u^{kg}\backslash\{i\}} W_{i'\cdot}^{kg}, \tag{4}$$

where $\mathcal{K}$ denotes a set of k-granularity and $\mathcal{S}_k$ is a set of rating groups w.r.t. k.

We can see that $\tilde{U}_{u\cdot}^{k\text{-gran}}$ in Eq. (4) reduces to $\tilde{U}_{u\cdot}^{\text{SVD++}}$ in Eq. (2) when $\mathcal{K} = \{5\}$, and reduces to $\tilde{U}_{u\cdot}^{\text{MF-MPC}}$ in Eq. (3) when $\mathcal{K} = \{1\}$, which shows that our proposed virtual user profile is a very generic one.

3.3 Prediction Rule

With the virtual user profile, we may define the predicted rating of user u to item i in a similar way to that of SVD++ [2] and MF-MPC [3],

$$\hat{r}_{ui} = (U_{u\cdot} + \tilde{U}_{u\cdot}^{k\text{-gran}})V_{i\cdot}^T + b_i + b_u + \mu, \tag{5}$$

where $U_{u\cdot} + \tilde{U}_{u\cdot}^{k\text{-gran}} \in \mathbb{R}^{1\times d}$ denotes the overall profile of user u, $V_{i\cdot} \in \mathbb{R}^{1\times d}$ is the item-specific latent feature vector, and $b_i \in \mathbb{R}$, $b_u \in \mathbb{R}$, $\mu \in \mathbb{R}$ are item bias, user bias and global average rating, respectively.

3.4 Objective Function

We embed the prediction rule in a commonly used objective function with square loss for rating prediction,

$$\min_{\Theta} \sum_{u=1}^{n} \sum_{i=1}^{m} y_{ui} [\frac{1}{2}(r_{ui} - \hat{r}_{ui})^2 + \text{Reg}(u,i)], \tag{6}$$

where $\text{Reg}(u,i) = \frac{\alpha}{2}||U_{u\cdot}||^2 + \frac{\alpha}{2}||V_{i\cdot}||^2 + \frac{\alpha}{2}||b_u||^2 + \frac{\alpha}{2}||b_i||^2 + \sum_{k\in\mathcal{K}}\sum_{g\in\mathcal{S}_k} \sum_{i'\in\mathcal{I}_u^{kg}\backslash\{i\}} \frac{\alpha}{2}||W_{i'\cdot}^{kg}||^2$ is the regularization term used to avoid overfitting. And $\Theta = \{U_{u\cdot}, V_{i\cdot}, b_u, b_i, W_{i'\cdot}^{kg}\}, u = 1, 2 \ldots, n, i = 1, 2, \ldots, m,\ i' \in \mathcal{I}_u^{kg}\backslash\{i\},\ k \in \mathcal{K},\ g \in \mathcal{S}_k$ are model parameters to be learned.

3.5 Gradients

Denoting $f_{ui} = \frac{1}{2}(r_{ui} - \hat{r}_{ui})^2 + \text{Reg}(u,i)$ as the tentative objective function for the rating triple (u, i, r_{ui}), we have the gradients of the model parameters,

$$\nabla U_{u\cdot} = -e_{ui} V_{i\cdot} + \alpha U_{u\cdot}, \tag{7}$$

$$\nabla V_{i\cdot} = -e_{ui}(U_{u\cdot} + \tilde{U}_{u\cdot}^{k\text{-gran}}) + \alpha V_{i\cdot}, \tag{8}$$

$$\nabla b_i = -e_{ui} + \alpha b_i, \tag{9}$$

$$\nabla b_u = -e_{ui} + \alpha b_u, \tag{10}$$

$$\nabla \mu = -e_{ui}, \tag{11}$$

$$\nabla W_{i'\cdot}^{kg} = -\frac{e_{ui} V_{i\cdot}}{\sqrt{|\mathcal{I}_u^{kg}\backslash\{i\}|}} + \alpha W_{i'\cdot}^{kg}, \tag{12}$$

where $i' \in \mathcal{I}_u^{kg}\backslash\{i\}, k \in \mathcal{K}, g \in \mathcal{S}_k$, and $e_{ui} = r_{ui} - \hat{r}_{ui}$. Notice that the gradients are the same with that of MF-MPC [3] except the virtual user profile $\tilde{U}_{u\cdot}^{k\text{-gran}}$ in Eq. (8), and $W_{i'\cdot}^{kg}$ in Eq. (12) with different granularity k and rating group g.

3.6 Update Rule

Finally, we have the update rules,

$$\theta = \theta - \gamma \nabla \theta \tag{13}$$

where γ is the learning rate, and θ can be $U_{u\cdot}, V_{i\cdot}, b_u, b_i, \mu$ and $W_{i'}^{kg}, i' \in \mathcal{I}_u^{kg}\backslash\{i\}, k \in \mathcal{K}, g \in \mathcal{S}_k$.

3.7 Algorithm

We formally depict the learning procedure in an algorithm shown in Fig. 2, which mainly contains two loops. In the outer loop, we go through the whole set of training rating records T times, and gradually decrease the learning rate via

$\gamma = \gamma \times 0.9$ when we have learned more about the model parameters. In each of the $|\mathcal{R}|$ iterations in the inner loop, we randomly take a rating record (u, i, r_{ui}) from $\mathcal{R}$ for calculating the corresponding gradients and updating the model parameters.

The learning procedure in Fig. 2 is adopted in typical SGD-based implementations of matrix factorization. The increased time complexity during the learning procedure, in comparison with that of PMF [6], is mainly from the k-granularity preference context, because we have to take the rated items as preference context into account. However, during the test period of rating prediction, the time complexity is similar to that of PMF [6] since we can calculate the virtual user profile of each user in advance.

```
1: Initialize the model parameters
2: for t = 1, ..., T do
3:     for t_2 = 1, ..., |R| do
4:         Randomly pick up a (u, i, r_ui) triple from R
5:         Calculate the gradients via Eq.(7-12)
6:         Update the parameters via Eq.(13)
7:     end for
8:     Decrease the learning rate γ ← γ × 0.9
9: end for
```

Fig. 2. The algorithm of k-CoFi.

4 Experimental Results

4.1 Datasets and Evaluation Metrics

In our empirical studies, we use three real-world datasets, i.e., MovieLens 100K (ML100K), MovieLens 1M (ML1M) and MovieLens 10M (ML10M), which have been used in a closely related work MF-MPC [3]. In particular, ML100K and ML1M contain about $100,000$ records and $1,000,000$ records respectively with rating values of $\mathbb{V} = \{1, 2, 3, 4, 5\}$, and ML10M contains about $10,000,000$ records with rating values of $\mathbb{V} = \{0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$. Similar to the definition of different k-granularity of ML100K and ML1M in Sect. 2, we have 5-granularity, 1-granularity and 2-granularity of ML10M as follows: $\mathcal{S}_1 = \{ \{0.5\}, \{1\}, \{1.5\}, \{2\}, \{2.5\}, \{3\}, \{3.5\}, \{4\}, \{4.5\}, \{5\} \}$, $\mathcal{S}_2 = \{ \{0.5, 1, 1.5, 2\}, \{1.5, 2, 2.5, 3\}, \{2.5, 3, 3.5, 4\}, \{3.5, 4, 4.5, 5\} \}$, and $\mathcal{S}_5 = \{ \{0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\} \}$. Notice that there are $n = 943$ users and $m = 1,682$ items in ML100K, $n = 6,040$ users and $m = 3,952$ items in ML1M, and $n =$ 71,567 users and $m = 10,681$ items in ML10M.

For each dataset, we randomly take 80% rating records as training data and the remaining 20% rating records as test data. We repeat this procedure for five times and have five copies of training data and test data for each dataset.

As for performance evaluation, we adopt two commonly used evaluation metrics, i.e., mean absolute error (MAE) and root mean square error (RMSE).

4.2 Baselines and Parameter Settings

In our empirical studies, our main purpose is to study the effectiveness of different k-granularity preference context and their complementarity. For this reason, we include the following methods in our experiments:

- $\mathcal{K} = \emptyset$ denoting matrix factorization without preference context, i.e., PMF [6];
- $\mathcal{K} = \{5\}$ denoting matrix factorization with 5-granularity coarse preference context, i.e., SVD++ [2];
- $\mathcal{K} = \{1\}$ denoting matrix factorization with 1-granularity fine preference context, i.e., MF-MPC [3]; and
- $\mathcal{K} = \{2\}$ denoting matrix factorization with 2-granularity smooth preference context.

In our preliminary empirical studies, we find that factorization with $\mathcal{K} = \{1\}$ and $\mathcal{K} = \{2\}$ perform the best and are much better than that with $\mathcal{K} = \{5\}$. Thus, we further study their complementarity via two approaches: (i) $\mathcal{K} = \{1, 2\}$, and (ii) combine the predicted ratings of $\mathcal{K} = \{1\}$ and $\mathcal{K} = \{2\}$ via a weighted combination, i.e., $\lambda \hat{r}_{ui}^{\{1\}} + (1 - \lambda)\hat{r}_{ui}^{\{2\}}$.

For parameter configuration in k-CoFi, we follow MF-MPC [3]. Specifically, we fix the number of latent dimensions $d = 20$, iteration number $T = 50$, and search the best tradeoff parameter $\alpha \in \{0.001, 0.01, 0.1\}$.

4.3 Results

We report the main results in Table 2, from which we can have the following observations:

- Matrix factorization with preference context of $\mathcal{K} = \{5\}$, $\mathcal{K} = \{1\}$, $\mathcal{K} = \{2\}$ performs better than that without preference context, i.e., $\mathcal{K} = \emptyset$, which clearly shows the effectiveness of modeling the preference context in rating prediction;
- Matrix factorization with preference context of $\mathcal{K} = \{1\}$, $\mathcal{K} = \{2\}$ is much better than that with $\mathcal{K} = \{5\}$, which shows the effectiveness of fine and smooth preference context in comparison with the coarse one;
- The performance of matrix factorization with preference context of $\mathcal{K} = \{2\}$ is close to the very strong baseline with $\mathcal{K} = \{1\}$ (i.e., MF-MPC [3]) across three datasets, which shows the effectiveness of the smooth preference context;
- As for the combination of $\mathcal{K} = \{1\}$ and $\mathcal{K} = \{2\}$, we find that the performance can be further improved in most cases, which showcases the complementarity of the two best performing methods associated with the fine granularity and smooth granularity preference context.

Table 2. Recommendation performance of k-CoFi with different values of k. The significantly best results are marked in bold ($p < 0.01$). The values of the tradeoff parameter α or the combination weight λ are also included for reproducibility.

Data	Method	α, λ	MAE	RMSE
ML100K	$k \in \mathcal{K} = \emptyset$, i.e., PMF	$\alpha = 0.1$	$0.7475_{\pm 0.0031}$	$0.9447_{\pm 0.0031}$
	$k \in \mathcal{K} = \{5\}$, i.e., SVD++	$\alpha = 0.001$	$0.7268_{\pm 0.0032}$	$0.9253_{\pm 0.0034}$
	$k \in \mathcal{K} = \{1\}$, i.e., MF-MPC	$\alpha = 0.001$	$0.7087_{\pm 0.0030}$	$0.9086_{\pm 0.0031}$
	$k \in \mathcal{K} = \{2\}$	$\alpha = 0.001$	$0.7090_{\pm 0.0031}$	$0.9073_{\pm 0.0029}$
	$k \in \mathcal{K} = \{1, 2\}$	$\alpha = 0.01$	$0.7084_{\pm 0.0037}$	$0.9072_{\pm 0.0033}$
	$k \in \mathcal{K} = \{1\}$, $k \in \mathcal{K} = \{2\}$	$\lambda = 0.5$	$\mathbf{0.7068}_{\pm 0.0029}$	$\mathbf{0.9051}_{\pm 0.0027}$
ML1M	$k \in \mathcal{K} = \emptyset$, i.e., PMF	$\alpha = 0.001$	$0.6960_{\pm 0.0013}$	$0.8840_{\pm 0.0017}$
	$k \in \mathcal{K} = \{5\}$, i.e., SVD++	$\alpha = 0.001$	$0.6656_{\pm 0.0016}$	$0.8514_{\pm 0.0019}$
	$k \in \mathcal{K} = \{1\}$, i.e., MF-MPC	$\alpha = 0.01$	$0.6599_{\pm 0.0014}$	$0.8441_{\pm 0.0019}$
	$k \in \mathcal{K} = \{2\}$	$\alpha = 0.001$	$0.6586_{\pm 0.0009}$	$0.8466_{\pm 0.0015}$
	$k \in \mathcal{K} = \{1, 2\}$	$\alpha = 0.01$	$0.6578_{\pm 0.0014}$	$0.8426_{\pm 0.0019}$
	$k \in \mathcal{K} = \{1\}$, $k \in \mathcal{K} = \{2\}$	$\lambda = 0.5$	$\mathbf{0.6557}_{\pm 0.0011}$	$\mathbf{0.8404}_{\pm 0.0016}$
ML10M	$k \in \mathcal{K} = \emptyset$, i.e., PMF	$\alpha = 0.01$	$0.6069_{\pm 0.0005}$	$0.7913_{\pm 0.0007}$
	$k \in \mathcal{K} = \{5\}$, i.e., SVD++	$\alpha = 0.01$	$0.6030_{\pm 0.0004}$	$0.7869_{\pm 0.0006}$
	$k \in \mathcal{K} = \{1\}$, i.e., MF-MPC	$\alpha = 0.01$	$0.5945_{\pm 0.0003}$	$0.7779_{\pm 0.0004}$
	$k \in \mathcal{K} = \{2\}$	$\alpha = 0.01$	$0.5964_{\pm 0.0006}$	$0.7800_{\pm 0.0007}$
	$k \in \mathcal{K} = \{1, 2\}$	$\alpha = 0.01$	$0.5948_{\pm 0.0004}$	$0.7783_{\pm 0.0006}$
	$k \in \mathcal{K} = \{1\}$, $k \in \mathcal{K} = \{2\}$	$\lambda = 0.5$	$0.5944_{\pm 0.0004}$	$\mathbf{0.7776}_{\pm 0.0005}$

In order to further study the complementarity of matrix factorization with $\mathcal{K} = \{1\}$ and $\mathcal{K} = \{2\}$, we change the value of the combination weight $\lambda \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. Notice that the hybrid method reduces to MF-MPC [3] when $\lambda = 1$ and to that with $\mathcal{K} = \{2\}$ when $\lambda = 0$. We report the results of RMSE in Fig. 3. Notice that the results of MAE are similar, which are thus not included in the paper. We can see: (i) the hybrid method performs the best with λ around 0.5, which shows the complementarity of the two factorization

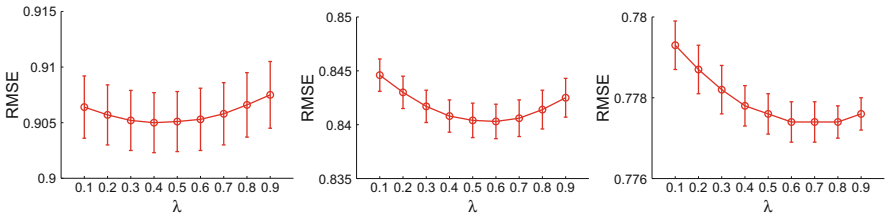


Fig. 3. Recommendation performance of combining 1-granularity and 2-granularity preference context with different values of $\lambda \in \{0.1, 0.2, \ldots, 0.9\}$.

methods with different k-granularity again; and (ii) the trend also suggests an appropriate choice of the value of λ in practice when deploying the proposed method k-CoFi, i.e., practitioners may safely choose to configure it as $\lambda = 0.5$ or so.

5 Related Work

Collaborative filtering or preference prediction in a (user, item) numerical rating matrix has been studied for more than two decades. In this long journey, some seminal algorithms have been developed such as neighborhood-based methods [5] in 1990s, factorization-based methods [6] in 2000s, and deep learning based methods [1] very recently.

In neighborhood-based methods, we usually calculate the (user, user) or (item, item) similarity and construct the neighborhood for each user or item firstly, and then predict the rating for each (user, item) pair by aggregating the preferences of the users or items in the corresponding neighborhood. For either user-oriented methods or item-oriented methods, the similarity measurement and the size of neighborhood are probably two most important factors, which are usually dependent on the data properties such as the number of users, the number of items, and the density of the (user, item) rating matrix.

In factorization-based methods, we turn to predict the missing ratings in a (user, item) rating matrix via reconstructing some decomposed or factorized latent matrices. Our k-CoFi also belongs to this family but with a very general prediction rule based on the k-granularity preference context.

In deep learning-based methods, more than one copies of the factored latent matrices are learned in multiple layers of projection or embedding. In particular, some non-linear activation functions and several layers of projection make it very powerful in recommender systems beyond computer vision, speech recognition and other areas.

In this paper, we study k-granularity preference context, which is vertical to the above three categories of representative methods. In particular, our k-granularity may also be applied to improve neighborhood-based methods and deep learning based methods via estimating the similarity more accurately when constructing a better neighborhood or learning better latent representations in a neural network.

6 Conclusions and Future Work

In this paper, we study a classical rating prediction problem in collaborative filtering from a novel perspective. Specifically, we propose a new k-granularity preference context, which absorbs existing fine and coarse preference context as special cases. We further develop a novel and generic recommendation algorithm, i.e., k-CoFi, with the proposed k-granularity preference context. Finally, we conduct extensive empirical studies, and verify the effectiveness of different

k-granularity and their complementarity, and have some interesting and useful observations.

For future works, we are interested in generalizing the k-granularity preference context with structure information to other recommendation paradigms such as neighborhood- and deep learning based collaborative filtering.

Acknowledgements. We thank the support of National Natural Science Foundation of China No. 61502307, No. 61672358 and U1636202, and Natural Science Foundation of Guangdong Province No. 2014A030310268 and No. 2016A030313038.

References

1. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.-S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, WWW 2017, pp. 173–182 (2017)
2. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008, pp. 426–434 (2008)
3. Pan, W., Ming, Z.: Collaborative recommendation with multiclass preference context. IEEE Intell. Syst. **32**(2), 45–51 (2017)
4. Rendle, S.: Factorization machines with libfm. ACM Trans. Intell. Syst. Technol. **3**(3), 57:1–57:22 (2012)
5. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW 1994, pp. 175–186 (1994)
6. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: Annual Conference on Neural Information Processing Systems, NIPS 2008, pp. 1257–1264 (2008)

Implementation Maximum Overall Coverage Constraint Non-negative Matrix Factorization for Hyperspectral Mixed Pixels Analysis Using MapReduce

Ying Wang[1], Qian Zhou[2(✉)], and Yunfeng Kong[3]

[1] Institute of Intelligence Networks System, Henan University, Kaifeng 475000, Henan, China
wangying@henu.edu.cn
[2] Art Department, Yellow River Conservancy Technical Institute, Kaifeng 475000, Henan, China
zhouqian715@163.com
[3] The College of Environment and Planning, Henan University, Kaifeng 475000, Henan, China
yfkong@henu.edu.cn

Abstract. As an effective blind source separation method, non-negative matrix factorization has been widely adopted to analyze mixed data in hyperspectral image. However some constraints have to be added in the objective function for more accurate estimates due to the existence of local optima. In this paper, a new NMF-based mixed data analysis algorithm is presented, with maximum overall coverage constraint introduced in traditional NMF, referred to as the MOCC-NMF. Furthermore, in order to handle huge computation involved, parallelism implementation of proposed algorithm using MapReduce is described and the new partitioning strategy to obtain matrix multiplication and determinant value is discussed in detail. In the numerical experiments conducted on real hyperspectral and synthetic datasets of different sizes, the efficiency and scalability of the proposed algorithm is confirmed.

Keywords: Hyperspectral image · MapReduce
Non-negative matrix factorization · Convex geometry · Endmember

1 Introduction

Hyperspectral remote sensing image provide large amount of image, space, and spectra information [1], which has a wide range of applications such as terrain classification, mineral detection and exploration, environmental studies, etc. [2, 3]. Since the limitation of spatial resolution of hyperspectral sensors, more than one material can contribute to

This work is supported by National Natural Science Foundation of China (No. 61703141), the Basic and Frontier Technology Research of Henan Province Science and Technology Department (No. 162300410198) and the Henan Postdoctoral Science Found (No. 153040).

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 417–426, 2018.
https://doi.org/10.1007/978-3-319-73830-7_41

the spectrum measured from a single pixel, causing wide existence of mixed data in one scene. In this situation, hyperspectral mixed data analysis is an essential processing step for hyperspectral applications. In recent decades, mixed data analysis algorithms integrated non-negative matrix factorization and convex geometry model have caught increasingly attention.

Due to matrices factorization and determinants calculation involving, the need for developing distributed implementation of NMF-based data analysis algorithms is motivated recently. Many approaches faced up the scalability problem of multiplicative rules for NMF have been made on cloud computing platforms [4, 5]. In this paper, we present a new constrained NMF method to analyze mixed data in hyperspectral image, incorporating a maximum overall coverage constraint, named MOCC-NMF, and scale up it on MapReduce based on our proposed partition schemes.

2 Background and Method

Mixed data analysis is an important and fundamental work in hyperspectral remote sensing image processing, including endmember extraction and spectral unmixing generally. Every pixel in hyperspectral image can be represented a linear combination of pure spectral signatures (called endmembers) weighted by the correspondent abundance fractions, while the abundances and signatures are all non-negative owing to the physical constraints. With the motivation straightforward arisen from the non-negative property of spectral measurement, non-negative matrix factorization (NMF) has been adopted generally to deal with the problem of mixed pixel analysis. In order to render better estimates, some new constraints are investigated in objective function of standard NMF, such as Geometric Optimization Model (GOM) [6], NMF with Barycentric Coordinates (BC-NMF) [7], minimum volume constraint NMF (MVC-NMF) [8], NMF with smoothness constraint (SCNMF) [9] etc. The purpose of all these methods is to find appropriate constraints and objective functions to represent important characteristics of hyperspectral data in the feature space based on the linear mixing model (LMM).

2.1 Linear Mixing Model

In mixed data analysis, the linear mixing model (LMM) is a significant popularity formulation due to its effectiveness and simplicity. Using matrix symbol, the model is given by:

$$\mathbf{X} = \mathbf{EC} + \mathbf{N}, \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{n \times m}$ is the data matrix in which every column vector $\mathbf{x}_i \in \mathbb{R}^{n \times 1}$ is an observation vector at a single pixel in a hyperspectral image with n spectral bands and m is the total number of pixels. $\mathbf{E} \in \mathbb{R}^{n \times p}$ is an endmember matrix, p is the number of endmembers in the image, and $\mathbf{c}_i \in \mathbb{R}^{p \times 1}$ as a column vector of abundance matrix $\mathbf{C} \in \mathbb{R}^{p \times m}$ each of whose entries $\mathbf{c}_{ij}$ is a scalar representing the fractional abundance of endmember vector $\mathbf{e}_i$ in the pixel signature $\mathbf{x}_j$. The LMM used in hyperspectral image

with two physical constrained conditions named abundance non-negative constraint (ANC) and abundance sum-to-one constraint (ASC):

$$\mathbf{x}_{ij}, \mathbf{e}_{ij}, \mathbf{c}_{ij} \geq 0, \tag{2}$$

$$\sum\nolimits_{i=1}^{p} \mathbf{c}_{i,j} = 1, \tag{3}$$

describes that the data cloud is within a simplex in the feature space, whose vertices correspond to the endmembers.

2.2 Non-negative Matrix Factorization

NMF aims to factorize an original high dimensional matrix $\mathbf{V} \in \mathbb{R}^{n \times m}$ into two low-rank matrices $\mathbf{W} \in \mathbb{R}^{n \times r}$ and $\mathbf{H} \in \mathbb{R}^{r \times m}$ with the non-negative assumption that all elements in $\mathbf{V}$, $\mathbf{W}$ and $\mathbf{H}$ are positive such that:

$$\mathbf{V} \approx \mathbf{WH}, \tag{4}$$

the parameter $r \ll \min(n, m)$ is the desired rank of matrix W, and normally it is regarded as the number of features. Especially r means the number of endmembers, p, in a hyperspectral image. To solve the NMF problem, a probabilistic interpretation [10] minimizes the cost function:

$$f(\mathbf{W}, \mathbf{H}) = \frac{1}{2}\|\mathbf{V} - \mathbf{WH}\|_{\mathrm{F}}^2 = \sum\nolimits_{i,j} \left[\mathbf{V}_{i,j} - (\mathbf{WH})_{i,j}\right]^2, \tag{5}$$

which is the Euclidean distance between $\mathbf{V}$ and $\mathbf{WH}$. The significantly creative job to solve the above optimization problem is the multiplicative rule [11]:

$$\mathbf{W}_{i,k} \leftarrow \mathbf{W}_{i,k} \frac{\left(\mathbf{VH}^{\mathbf{T}}\right)_{i,k}}{\left(\mathbf{WHH}^{\mathbf{T}}\right)_{i,k}}, \tag{6}$$

$$\mathbf{H}_{k,j} \leftarrow \mathbf{H}_{k,j} \frac{\left(\mathbf{W}^{\mathbf{T}}\mathbf{V}\right)_{k,j}}{\left(\mathbf{W}^{\mathbf{T}}\mathbf{WH}\right)_{k,j}}. \tag{7}$$

Due to the non-convexity of the objective function in both $\mathbf{W}$ and $\mathbf{H}$ simultaneously, the optimization problem can have local minima. On the other hand, the non-uniqueness of the nonnegative factors is apparent for any non-negative invertible matrix $\mathbf{D}$, $\mathbf{WH} = \mathbf{WDD}^{-1}\mathbf{H}$. So in many applications some additional constraints are added to find the global optimum and mitigate the non-uniqueness of solution [12, 13].

2.3 Maximum Overall Coverage Constraint and MOCC-NMF

In hyperspectral images each pixel can be thought of as a vector in an n-dimensional space, where each band is assigned to one axis of space. Due to the inherent constraints

involved in LMM, ANC and ASC, the data vectors are confined within a $p-1$-dimensional simplex whose vertices are p endmembers of the image. In a mixed image with no pure pixels, in order to find the best approximate endmembers, the simplex has to be extended to hold as many data points as possible. These observations inspire the proposed of the maximum overall coverage constraint by calculating the volume of simplexes $\mathrm{V}_{\mathbb{S}}$ [14]:

$$\mathrm{V}_{\mathbb{S}} = \frac{1}{p!}\sqrt{det\left(\mathbf{E}^{\mathbf{T}}\mathbf{E}\right)}, \tag{8}$$

where $\mathbf{E} = \left[\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_{\boldsymbol{p}}\right]$ is endmember vectors set. Adding another mixed pixel vector to $\mathbf{E}$ composes $\mathbf{E}_+ = \left[\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_p, \boldsymbol{x}_t\right]$, and the volume of this new simplex V_t is calculated by:

$$\mathrm{V}_t = \frac{1}{(p+1)!}\sqrt{det\left(\mathbf{E}_+^{\mathbf{T}}\mathbf{E}_+\right)}. \tag{9}$$

It is clearly that for any $\boldsymbol{x}_t$ within the simplex $\mathrm{V}_{\mathbb{S}}$, V_t is zero. Incorporating the maximum overall coverage constraint into the NMF technique, we propose the MOCC-NMF to solve following optimization problems:

$$\text{minimize} \quad f_{\mathrm{MOCC-NMF}}(\mathbf{E}, \mathbf{C}) = \frac{1}{2}\|\mathbf{X} - \mathbf{E}\mathbf{C}\|_{\mathrm{F}}^2 + \lambda J(\mathrm{V}), \tag{10}$$

where $J(\mathrm{V}) = \sum_{t=1}^{m} (\mathrm{V}_t)^2$ is the constraint item and the regularization coefficient λ is used to control the tradeoff between the accurate reconstruction and the overall coverage.

We here solve the optimization problem mentioned above by the gradient descent idea. The gradient of objective function $f_{\mathrm{MOCC-NMF}}(\mathbf{E}, \mathbf{C})$ about $\mathbf{C}$ is easy to obtain since it is independent of $\mathbf{C}$:

$$\nabla_{\mathbf{C}} f_{\mathrm{MOCC-NMF}}(\mathbf{E}, \mathbf{C}) = \mathbf{E}^{\mathbf{T}}\mathbf{E}\mathbf{C} - \mathbf{E}^{\mathbf{T}}\mathbf{X}, \tag{11}$$

and the gradient $\nabla_{\mathbf{E}} f_{\mathrm{MOCC-NMF}}(\mathbf{E}, \mathbf{C})$ is calculated by:

$$\nabla_{\mathbf{E}} f_{\mathrm{MOCC-NMF}}(\mathbf{E}, \mathbf{C}) = \mathbf{E}\mathbf{C}\mathbf{C}^{\mathbf{T}} - \mathbf{X}\mathbf{C}^{\mathbf{T}} + \lambda \frac{\partial J(\mathrm{V})}{\partial \mathbf{E}}. \tag{12}$$

Unfolding the summations $J(\mathrm{V}) = \sum_{t=1}^{m} (\mathrm{V}_t)^2$, the partial derivative of each item is given by:

$$\frac{\partial (\mathrm{V}_t)^2}{\partial \mathbf{E}} = \frac{1}{[(p+1)!]^2} \frac{\partial det(\mathbf{E}_+^{\mathbf{T}}\mathbf{E}_+)}{\partial \mathbf{E}}$$

$$= \frac{1}{[(p+1)!]^2} \begin{bmatrix} \frac{\partial det(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+)}{\partial \mathbf{E}_{1,1}} & \frac{\partial det(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+)}{\partial \mathbf{E}_{1,2}} & \cdots & \frac{\partial det(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+)}{\partial \mathbf{E}_{1,p}} \\ \frac{\partial det(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+)}{\partial \mathbf{E}_{2,1}} & \frac{\partial det(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+)}{\partial \mathbf{E}_{2,2}} & \cdots & \frac{\partial det(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+)}{\partial \mathbf{E}_{2,p}} \\ \vdots & \vdots & & \vdots \\ \frac{\partial det(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+)}{\partial \mathbf{E}_{n,1}} & \frac{\partial det(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+)}{\partial \mathbf{E}_{n,2}} & \vdots & \frac{\partial det(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+)}{\partial \mathbf{E}_{n,p}} \end{bmatrix}. \tag{13}$$

Each element of $\frac{\partial det(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+)}{\partial \mathbf{E}}$ is expressed as:

$$\frac{\partial det(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+)}{\partial \mathbf{E}_{i,j}} = det(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+) tr\left[(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+)^{-1} \frac{\partial (\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+)}{\partial \mathbf{E}_{i,j}} \right]. \tag{14}$$

Denote $\boldsymbol{x}_t$ in $\mathbf{E}_+$ by $\boldsymbol{e}_{p+1}$, we arrive at:

$$tr\left[(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+)^{-1} \frac{\partial (\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+)}{\partial \mathbf{E}_{i,j}} \right] = \frac{2}{det(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+)} \sum_{l=1}^{p+1} C\langle \boldsymbol{e}_j, \boldsymbol{e}_l \rangle (\mathbf{E}_+)_{i,l}, \tag{15}$$

where $\langle \boldsymbol{e}_j, \boldsymbol{e}_l \rangle$ is an element in $\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+$ and $C\langle \boldsymbol{e}_j, \boldsymbol{e}_l \rangle$ is the cofactor of this entry. Let $\tau = \frac{2}{[(p+1)!]^2}$, we have the multiplicative update rules of MOCC-NMF:

$$\mathbf{E}_{i,k} \leftarrow \frac{\mathbf{E}_{i,k}}{(\mathbf{E}\mathbf{C}\mathbf{C}^{\mathbf{T}})_{i,k}} \left[(\mathbf{X}\mathbf{C}^{\mathbf{T}})_{i,k} - \lambda \tau \sum_{t=1}^{m} \sum_{l=1}^{p+1} C\langle \boldsymbol{e}_j, \boldsymbol{e}_l \rangle (\mathbf{E}_+)_{i,l} \right], \tag{16}$$

$$\mathbf{C}_{k,j} \leftarrow \mathbf{C}_{k,j} \frac{(\mathbf{E}^{\mathbf{T}}\mathbf{X})_{k,j}}{(\mathbf{E}^{\mathbf{T}}\mathbf{E}\mathbf{C})_{k,j}}. \tag{17}$$

3 Distributed MOCC-NMF on MapReduce

Inspired by two functional programming primitives present in Lisp, Google's MapReduce programming model serves for processing large data sets in a massively parallel manner automatically, which expresses once a computation task as a series of Map and Reduce stages [15–20]. In MapReduce model, the datasets are organized as key/value pairs, and in the map stage the input key/values pairs are processed to a set of intermediate pairs by dividing a task into several independent subtasks that can be run in parallel. In the reduce stage intermediate key/value pairs are grouped firstly and merged associated with the same intermediate key then gathering all the sub results to the final output.

To parallelize the computation of MOCC-NMF, some major jobs to be considered including how to implement matrix multiplication and determinant calculation on MapReduce. In the following, we will discuss the strategy of decomposition matrix and determinant, and then demonstrate how to scale up MOCC-NMF on MapReduce.

3.1 Matrix Multiplication and Determinant Calculation Partitioning Schemes

Matrix multiplication is one of the major operations in the updating rule of MOCC-NMF. Thus, in processing matrix factorization, how to implement matrix multiplications on MapReduce will significantly affect the final scalability. Pervious works on distributing matrix multiplication partition **E** and **C** along the long dimension [21] traditionally, along the short dimension [22] to minimize the communication cost and with block-wise scheme [23]. We use symbol $\boldsymbol{e}_{1:n,i}$ to denote the i^{th} column vector of **E**, $\boldsymbol{c}_{i,1:m}$ to denote the i^{th} row vector of **C**. An alternative way is proposed to decompose matrices by computing the outer product that can be parallelized straightforward:

$$\mathbf{EC} = \sum\nolimits_i \boldsymbol{e}_{1:n,i} \otimes \boldsymbol{c}_{i,1:m}. \tag{18}$$

The partition scheme can be designed by the following MapReduce steps:

- Map: Tuple $\langle \mathbf{E}, \mathbf{C}, i, j \rangle$ is shuffled to a worker machine as input split and an intermediate key/value pair $\langle \boldsymbol{e}_{j,i} \cdot \boldsymbol{c}_{i,1:m}, i, j \rangle$ will be emitted.
- Reduce-I: Take $\langle \boldsymbol{e}_{j,i} \cdot \boldsymbol{c}_{i,1:m}, i, j \; \forall \, j \in [1, n] \rangle$ and emit $\langle \boldsymbol{e}_{1:n,i} \otimes \boldsymbol{c}_{i,1:m}, i \rangle$, where $\boldsymbol{e}_{1:n,i} \otimes \boldsymbol{c}_{i,1:m} = \begin{bmatrix} \boldsymbol{e}_{1,i} \cdot \boldsymbol{c}_{i,1:m} \\ \vdots \\ \boldsymbol{e}_{j,i} \cdot \boldsymbol{c}_{i,1:m} \\ \vdots \\ \boldsymbol{e}_{n,i} \cdot \boldsymbol{c}_{i,1:m} \end{bmatrix}$.
- Reduce-II: Take $\langle \boldsymbol{e}_{1:n,i} \otimes \boldsymbol{c}_{i,1:m}, i \; \forall \; i \in [1, p] \rangle$ and output $\mathbf{EC} = \sum_i \boldsymbol{e}_{1:n,i} \otimes \boldsymbol{c}_{i,1:m}$.

In the reasonable decomposition model, the first job in the Map phase is do multiplication, accepting the entry $\boldsymbol{e}_{j,i}$ of column vector $\boldsymbol{e}_{1:n,i}$ and row vector $\boldsymbol{c}_{i,1:m}$, and emitting $\langle \boldsymbol{e}_{j,i} \cdot \boldsymbol{c}_{i,1:m}, i, j \rangle$. The second job in the Reduce phase is do stacking and summation, where vectors $\boldsymbol{e}_{j,i} \cdot \boldsymbol{c}_{i,1:m}$ will be stacked as row vectors of $\boldsymbol{e}_{1:n,i} \otimes \boldsymbol{c}_{i,1:m}$ in Reduce-I stage, and matrices $\boldsymbol{e}_{1:n,i} \otimes \boldsymbol{c}_{i,1:m}$ will be added to generate the final output **EC** in Reduce-II stage.

For paralleling determinant calculation $det\left(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+\right)$, nested MapReduce procedures are arranged as follows:

- Map: Tuple $\left\langle \mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+, 1, j \right\rangle$ is shuffled to a Map node and an intermediate key/value pair $\left\langle \left(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+\right)_{1,j} \cdot C\left(\left(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+\right)_{1,j}\right), j \right\rangle$ will be emitted.
  - Nested MapReduce procedures to calculate $C\left(\left(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+\right)_{1,j}\right)$.

    Continue nesting…
- Reduce: Take $\left\langle \left(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+\right)_{1,j} \cdot C\left(\left(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+\right)_{1,j}\right), j \; \forall \, j \in [1, p+1] \right\rangle$ and output $det\left(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+\right) = \sum_j \left(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+\right)_{1,j} \cdot C\left(\left(\mathbf{E}_+^{\mathbf{T}} \mathbf{E}_+\right)_{1,j}\right)$.

In fact, the MapReduce paradigm can be used to calculate any giant determinant with appropriate numbers of nesting level according to the computing capability of each worker node.

3.2 Distributed MOCC-NMF

To analyze hyperspectral mixed data, the pseudo-code for implementation of the distributed MOCC-NMF's updating rule (16) is shown in Algorithm 1 while the program of another updating rule (17) is similar to Algorithm 1 except the maximum overall coverage constraint.

Algorithm 1. Distributed MOCC-NMF

Input: Non-negative hyperspectral image data matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$, maximum number of iterations k_{max}.

Output: Two non-negative matrices $\mathbf{E} \in \mathbb{R}^{n \times p}$ and $\mathbf{C} \in \mathbb{R}^{p \times m}$.

Initialization: $\mathbf{E}^{(0)}$, $\mathbf{C}^{(0)}$, the number of endmembers p, $k = 1$.

FOR $k \leq k_{max}$

  /*Compute $\mathbf{XC}^{\mathbf{T}}$ on MapReduce*/

  FOR $i = 1$ to m

    FOR $j = 1$ to n

      emit intermediate key/value pair $\langle \boldsymbol{x}_{j,i} \cdot \boldsymbol{c}^{\mathbf{T}}{}_{i,1:p}, i, j \rangle$; /*Map*/

      stack $\boldsymbol{x}_{j,i} \cdot \boldsymbol{c}^{\mathbf{T}}{}_{i,1:p}$, to generate $\boldsymbol{x}_{1:n,i} \otimes \boldsymbol{c}^{\mathbf{T}}{}_{i,1:p}$; /*Reduce I*/

    ENDFOR

    add $\boldsymbol{x}_{1:n,i} \otimes \boldsymbol{c}^{\mathbf{T}}{}_{i,1:p}$ to output $\mathbf{XC}^{\mathbf{T}}$; /*Reduce II*/

  ENDFOR

  /*Compute $\mathbf{CC}^{\mathbf{T}}$ on MapReduce*/;

  …

  /*Compute $\mathbf{ECC}^{\mathbf{T}}$ on MapReduce*/;

  …

  FOR $t = 1$ to m

    FOR $l = 1$ to $p + 1$

      /* Calculation $C\langle \boldsymbol{e}_j, \boldsymbol{e}_l \rangle$ on MapReduce*/

    ENDFOR

  ENDFOR

  Product constraint item;

  Update $\mathbf{E}^{(k)}$ with (16) and $\mathbf{C}^{(k)}$ with (17);

ENDFOR

4 Experimental Evaluation

To verify the performance of the proposed distributed MOCC-NMF (DMOCC-NMF), experiments have been conducted in this section. The DMOCC-NMF algorithm is implemented in Matlab2015a using the MATLAB Distributed Computing Server. We build the DMOCC-NMF in the parallel pool of 64–128 workers, run on a local cluster consisted of machines equipped with CPU Intel i5-4590, 3.2 GHz, 8 GB RAM, 500 GB HD disk. Both synthetic and real world datasets have been used in these tests. Real hyperspectral image (named R1) used in our experiments has been captured by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) over Cuprite, Nevada, 1995. The dataset provide by ENVI5.3 consists of 350 × 400 pixels, 50 bands image in SWIR from 1.9908 to 2.4790 μm, and total size of 15 MB. The synthetic images of various sizes 105 × 105 × 50 (10 MB), 525 × 525 × 50 (200 MB), 1050 × 1050 × 50 (1 GB) and 2100 × 2100 × 50 (4 GB) (named S1, S2, S3 and S4 respectively) are generated using a set of average endmembers spectral reflectances as endmembers selected from the Cuprite95 image.

The DMOCC-NMF execution performance on different datasets is listed in Table 1, where the elapse time of each component in updating rule (16) is reported. It can be noticed that the major computation cost is consumed in calculation $\mathbf{XC}^{\mathbf{T}}$ and the constraint item due to giant matrices and nested MapReduce procedures involved. High parallelism degree of the DMOCC-NMF demonstrates better speedup capability with the size increase of synthetic datasets, which is benefit to massive hyperspectral mixed data analysis.

Table 1. Execution performance on different datasets.

Component	Elapse time (s)				
	R1	S1	S2	S3	S4
$\mathbf{XC}^{\mathbf{T}}$	10.17	9.83	20.88	37.44	54.14
$\mathbf{ECC}^{\mathbf{T}}$	1.24	1.05	3.54	6.44	9.44
Constraint item	15.24	19.44	41.04	72.57	102.11

The ratio of the elapse time to the size versus number of endmembers (i.e. factorization dimensionality) is presented in Fig. 1 for all datasets. With prior information, the estimated number for endmembers p of each data is between 5 and 25. It shows that the slope of time complexity is stable with a slight decline w.r.t factorization dimensionality. The results clearly establish that the proposed DMOCC-NMF is linearly scalable and can be used to analyze any complex hyperspectral scene.

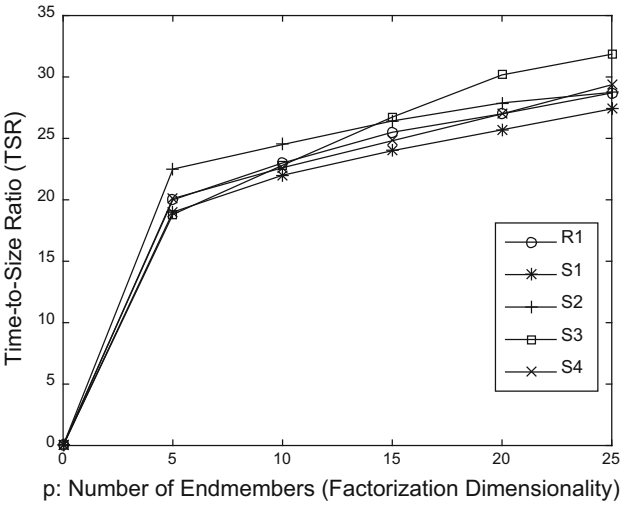


Fig. 1. Time-to-Size Ratio w.r.t. factorization dimensionality.

5 Conclusions

In this paper, we presented a novel mixed data analysis method for hyperspectral remote sensing image named Maximum Overall Coverage Constraint Non-negative Matrix Factorization (MOCC-NMF), and extended the distributed version (DMOCC-NMF) by paralleling the major computational tasks in updating rules to the MapReduce paradigm. The experimental results with synthetic and real data showed that the partition scheme is an effective parallel implementation. Consequently, the DMOCC-NMF is more amenable to be utilized to massive hyperspectral endmembers extraction and abundance estimation.

References

1. Kim, B., Landgrebe, D.A.: Hierarchical classifier design in high-dimensional numerous class cases. IEEE Trans. Geosci. Remote Sens. **29**(4), 518–528 (1991)
2. Landgrebe, D.: Hyperspectral image data analysis. Sig. Process. Mag. IEEE **19**(1), 17–28 (2002)
3. Keshava, N.: A survey of spectral unmixing algorithms. Lincoln Lab. J. 55–78 (2008)
4. Haut, J.M., Paoletti, M., Plaza, J., Plaza, A.: Cloud implementation of the k-means algorithm for hyperspectral image analysis. J. Super Comput. **73**(1), 514–529 (2017)
5. Dong, C., Zhao, H., Wang, W.: Parallel nonnegative matrix factorization algorithm on the distributed memory platform. Int. J. Parallel Prog. **38**(2), 117–137 (2010)
6. Geng, X., Ji, L., Zhao, Y., Wang, F.: A new endmember generation algorithm based on a geometric optimization model for hyperspectral images. IEEE Geosci. Remote Sens. Lett. **10**(4), 811–815 (2013)

7. Ji, L., Geng, X., Yu, K., Zhao, Y.: A new non-negative matrix factorization method based on barycentric coordinates for endmember extraction in hyperspectral remote sensing. Int. J. Remote Sens. **34**(19), 6577–6586 (2013)
8. Miao, L., Qi, H.: Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. IEEE Trans. Geosci. Remote Sens. **45**(3), 765–777 (2007)
9. Pauca, V.P., Piper, J., Plemmons, R.J.: Nonnegative matrix factorization for spectral data analysis. Linear Algebra Appl. **416**(1), 29–47 (2006)
10. Sajda, P., Du, S.: Recovery of constituent spectra using non-negative matrix factorization. In: Proceedings of SPIE – The International Society for Optical Engineering, vol. 5207, pp. 321–331 (2003)
11. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788–791 (1999)
12. Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints. J. Mach. Learn. Res. **5**, 1457–1469 (2004)
13. Pascualmontano, A., Carazo, J.M., Kochi, K., Lehmann, D., Pascualmarqui, R.D.: Nonsmooth Nonnegative Matrix Factorization (nsNMF). IEEE Trans. Pattern Anal. Mach. Intell. **28**(3), 403–415 (2006)
14. Geng, X., Zhao, Y., Wang, F., Gong, P.: A new volume formula for a simplex and its application to endmember extraction for hyperspectral image analysis. Int. J. Remote Sens. **31**(4), 1027–1035 (2010)
15. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. In: Conference on Symposium on Operating Systems Design and Implementation, p. 10 (2008)
16. Gai, K., Qiu, M., Zhao, H., Tao, L., Zong, Z.: Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. J. Netw. Comput. Appl. **59**, 46–54 (2016)
17. Gai, K., Qiu, M., Ming, Z., Zhao, H., Qiu, L.: Spoofing-jamming attack strategy using optimal power distributions in wireless smart grid networks. IEEE Trans. Smart Grid **8**(5), 2431–2439 (2017)
18. Gai, K., Qiu, M., Sun, X.: A survey on FinTech. J. Netw. Comput. Appl. **PP**, 1 (2017)
19. Gai, K., Qiu, M., Zhao, H.: Energy-aware task assignment for mobile cyber-enabled applications in heterogeneous cloud computing. J. Parallel Distribut. Comput. **111**, 126–135 (2018)
20. Gai, K., Qiu, L., Chen, M., Zhao, H., Qiu, M.: SA-EAST: security-aware efficient data transmission for ITS in mobile heterogeneous cloud computing. ACM Trans. Embedded Comput. Syst. **16**(2), 60 (2017)
21. Robila, S.A., Maciak, L.G.: A parallel unmixing algorithm for hyperspectral images. Optics East (2006)
22. Liu, C., Yang, H.C., Fan, J., He, L.W., Wang, Y.M.: Distributed nonnegative matrix factorization for web-scale dyadic data analysis on MapReduce. In: International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, pp. 681–690, April (2010)
23. Yin, J., Gao, L., Zhang, Z.: Scalable nonnegative matrix factorization with block-wise updates. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) ECML PKDD 2014. LNCS, vol. 8726, pp. 337–352. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44845-8_22

Improved Three-Dimensional Model Feature of Non-rigid Based on HKS

Fanzhi Zeng[1], Jiechang Qian[2(✉)], Yan Zhou[1], Changqing Yuan[2], and Chen Wu[1]

[1] Department of Computer, Foshan University, Foshan 528000, Guangdong, China
coolhead@126.com, zhouyan791266@163.com, wu_chk@163.com
[2] School of Automation, Foshan University, Foshan 528000, Guangdong, China
512502487@qq.com, 1252919197@qq.com

Abstract. The recognition and retrieval of 3D models have been a hot spot in the field of computer vision. Since the non-rigid shapes can generate various deformations, the recognition and retrieval of non-rigid 3D models are more complex and challenging than rigid one. Therefore, the key to the recognition and retrieval of non-rigid 3D models is to extract a feature which obtains substantial description ability and stability. An improved HKS feature named NSIHKS (NSIHKS, new scale Invariance heat kernel signature) was used to describe the shape of models in the paper. NSIHKS contains intrinsic invariance, scale transformation invariance, robustness et al. Moreover it has good resistance even under faint noise. Firstly, the NSIHKS features of each model were extracted and processed with clustering algorithm. Secondly, an efficient algorithm of similarity measurement was designed on the basis of Ming distance. Finally, NSIHKS features of each model in the standard data set were compared via the aforementioned distance algorithm. Experimental results of standard data set in this field show that this feature has good effect on the application of non-rigid 3D model retrieval.

Keywords: 3D model retrieval · 3D non-rigid model · Heat kernel signature
Shape features · Clustering

1 Introduction

With the rapid development of computer technology, computer software and hardware have greatly improved. Therefore, the acquisition technology of 3D models are development rapid and the 3D models are widely used too [1–4]. Nowadays, the field of 3D models are mainly used in product design, architectural design, film animation, medical science et al. [5–8]. The 3D models are not only growing rapidly in quantity, but also in application requirement. Due to the accumulation of a large number of 3D models, the application of the 3D model library is also increasing. According to the statistics, in the design and production, more than 80% of new products are modified on the basis of the original product. Rapid retrieval of the required model is an urgent need in various fields, so that efficiency can be improved and costs reduced. So how to retrieve in the massive 3D models have become the focus of research in recent years. The external

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 427–437, 2018.
https://doi.org/10.1007/978-3-319-73830-7_42

shape of rigid 3D models do not change with motion, so it is sufficient for rigid 3D models to have translation, rotation and scale transformation invariance [9]. In our life, the shape of the non-rigid is everywhere, and the application has become more and more extensive. The study of non-rigid model retrieval has been a challenge in the field of multimedia retrieval.

In this paper, a new recognition and retrieval algorithm for the feature of the non-rigid 3D models are proposed. For the recognition and retrieval of non-rigid 3D models, design a robust non-rigid 3D models matched algorithm base on the feature of HKS and the depth analysis the characteristics of 3D models. The algorithm contains three main stages: Firstly, HSIHKS is extracted by analyzing the internal vacuous and shape complex of 3D models structure feature. Secondly, a clustering method is designed based on sample model classified information and supervised learning process. Thirdly, the similarity comparative method is obtained with clustering result and distance calculation. Finally, the model retrieval process is completed when the similarity index between models are calculated based on NSHHKS. The algorithm from this paper has some highlights in complex structural non-rigid model retrieval efficiency and other aspects. And the algorithm is used for the retrieval of non-rigid three-dimensional model, which has better effect and shows the superiority of this algorithm by comparing with other algorithms. Moreover, this algorithm can be applied to the company. Help them better manage the model library, make design and manufacture of 3D models more efficient.

The remainder of this paper is organized as follows: In Sect. 2, we give a brief review of related works. In Sect. 3, algorithms for extraction of the NSIHKS feature. In Sect. 4, algorithms for obtainment clustering coefficient. In Sect. 5, algorithms for similarity measurement. In Sect. 6, the experimental results are discussed and analyzed. Some conclusions are drawn in Sect. 7.

2 Related Work

Since rigid shapes are not changed according to motion, it is enough for rigid 3D models to possess features as translation, rotation and scale transformation invariance. However, non-rigid shapes exist in every aspect of our lives, which makes its application become more and more extensive. There are abundant changes in non-rigid substance, therefore it is a great challenge to study non-rigid model retrieval in multi-media retrieval area. Most non-rigid changes can be similarly described as isometric transformation, hence one of the keys to non-rigid 3D models retrieval is to design an efficient feature. This feature should possess isometric invariance besides translation, rotation and scale transformation invariance. In 2006, Reuter et al. proposed the eigenvalue sequence of Laplace-Beltrami Operator as shape features, which owned good isometric invariance [10]. Moreover, it presented good retrieval effect in SHREC11 non-rigid shape retrieval competition [11]. But this method is using single spectrum signature, it cannot describe the similarity of 3D models in a good way. Furthermore it is easily inflected by faint noise. In 2007, Rustamov et al. proposed to demonstrate model shapes by extracting global point signature on the basis of Laplace-Beltrami Operator eigenvalue and

characteristic function and it achieved good results [12]. But the disturbance caused by characteristic function corresponding Laplace-Beltrami Operato similar eigenvalue could influence the stability of GPS feature points. To solve this problem, in 2009, Sun et al. proposed HKS base on Heat Kernel Spread Theory. This feature generates multi-scale description of 3D models by choosing different time to control heat spread [13]. HKS features have the following four points: (i) Intrinsic invariance. It is also known as isometric invariance. (ii) Integrity. It includes all the essential geometric information of all 3D models. (iii) Multi-scale attribute. When t is small, it only reflects local features of 3D models at point x and when t is large, it reflects the global structure of the 3D models at point x. (iv) Stability. It is stable against flow disturbance. All advantages of HKS feature have attracted interest from researchers. When scale normalization pre-treatment is used to rigid 3D models, the feature of rigid 3D models could possess scale transformation invariance. However, scale normalization pre-treatment is not suitable to non-rigid 3D models [14, 15]. HKS doesn't have scale transformation invariance, furthermore it includes lots of redundant information. Thus, Ovsjanikov et al. extracted every vertex HKS of 3D models, then obtained feature lexical set via K-means clustering, and retrieved 3D models with 3D model global shape features gained by BoW method [16]. To solve scale problem, in 2010, Bronsten et al. proposed SIHKS and resolved SIHKES scale problem by via the time shift invariance of Fourier transform amplitude. This feature is also one of local features and SIHKS can be used in shape retrieval of non-rigid 3D models via BoW method [17, 18].

3 The NSIHKS Feature Extraction Algorithm

The traditional HKS feature is formed with heat value, which is the sequence of heat from point x to point y, in given time t [13]. The HKS feature has intrinsic invariance, integrity, stability et al. But the stability of this feature is not sufficient as different scale of non-rigid three-dimensional model. When the model scales changes, the description effect is influenced greatly. On resolving the influence caused by scale changes, researchers creates new function based on heat kernel theory, then gain the feature with scale transformation invariance by taking the logarithm, the derivative and the Fourier transform [17, 18]. Owing to the complexity and variability of non-rigid three-dimensional model structure, the gained feature is still not enough accuracy. In this paper, we expand the HKS feature with scale transformation invariance and adopt the new function created by heat kernel theory. Meanwhile, combine the heat changes regulation under different scale. Let time parameters is better to adapt different scale. A NSIHKS feature is proposed in order to describe shape feature precisely. The NSIHKS feature extraction algorithm is summarized as follows.

Algorithm 1 (algorithm for NSIHKS feature extraction)

Input: the non-rigid 3D model of the .off format
Output: NSIHKS feature matrix
Initial conditions: number of eigenvalues and coefficient: k, α.

Procedure:
Step 1: For each vertex in non-rigid three-dimensional model, calculate the area of neighborhood triangular patches, structure diag($A_{n\times n}$) matrix. etc. n is the number of vertices of the model.

Step 2: Calculate the weight matrix $W = diag(\sum_{l\neq i} w_{il}) - (w_{ij})$, etc. w_{ij} is calculated by using Eq. (1), shown in Fig. 1:

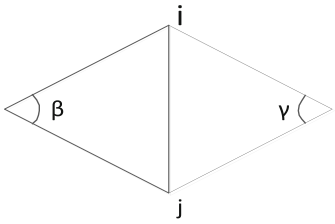


Fig. 1. The weight of vertex i and j

$$w_{ij} = \begin{cases} (\cot\gamma + \cot\beta)/2 & (i,j) \text{ is an edge;} \\ 0 & \text{else.} \end{cases} \tag{1}$$

Step 3: Then the discretization Laplace-Beltrami operator can be approximately represented as $L = A^{-1}W$, characteristic decomposition of L, compute the eigenvalues λ_i and eigenvectors φ_i, where subscripts $i = 1, 2, \ldots, k$.
Step 4: Calculate the $t_{\max}$ and $t_{\min}$ of the model as following:

$$\begin{cases} t_{max} = 4\ln 10/\lambda_2 \\ t_{min} = 4\ln 10/\lambda_{300} \end{cases} \tag{2}$$

Step 5: Calculate the maximum and minimum of τ using Eq. (3):

$$\begin{cases} \tau_{max} = ceil\,(\log \alpha(t_{max})) \\ \tau_{min} = floor\,(\log\ \alpha(t_{min})) \end{cases} \tag{3}$$

Step 6: Obtain sequence τ through uniformity sample in $[\tau_{min}, \tau_{max}]$.
Step 7: Calculate NSIHKS feature, using Eq. (4):

$$NSIHKS\,(x, \alpha^{\tau}) = \frac{-\sum_{i=0}^{\infty} \lambda_i \alpha^{\tau} \log \alpha e^{-\lambda_i \alpha^{\tau}} \varphi_i^2(x)}{\sum_{i=0}^{\infty} e^{-\lambda_i \alpha^{\tau}} \varphi_i^2(x)} \tag{4}$$

Step 8: Take the Fourier transform of the NSIHKS feature from the Step. 7, then calculate the amplitude.
Step 9: Selection from 2 to 10 dimensions of the amplitude as the final NSIHKS feature.

The new algorithm shows advantages:

(1) Through structure the new function, we improve the robustness of the feature under scale transformation.
(2) By analyzing the model's law of variation with time of HKS, the time parameter setting is optimized and make the feature more adaptable and robustness.

4 The Clustering Algorithm

The same kind of model is different in structure, and even the same model can be different in scale. Due to the complexity structure of non-rigid 3D models, it is hard to find a method to directly compare the NSIHKS feature between two models. Therefore, in this paper, adopts a method of supervised learning with sample models. Obtain the cluster center of the models in data set.

In order to obtain the clustering coefficients of NSIHKS feature in the data set, extraction samples in the data set to estimation the coefficients. The algorithm is describe as follows:

Algorithm 2 (algorithms for obtainment clustering coefficient)

Input: NSIHKS feature matrix
Output: Cluster center
Initial conditions: number of sampling m, number of iterations d, number of cluster center q and the expected value of the average information entropy p.
Procedure:
Step 1: Samples were randomly selected in the model library. if there is a class of model is not sampled, then resampling.
Step 2: According to the clustering center of the current setting, perform cluster operations to the NSIHKS features of each model.
Step 3: We denote the normalized distance between model p, and q as dis(p,q), by comparing the feature after clustering.
Step 4: Carry out retrieval test using the sample models. According to dis(p,q), the results of the retrieval are arranged from small to large. Calculate the number of the model which is the same type with the sample in the first 15 of the retrieval results, denote n. According to Eq. (5), calculate the information entropy and the average information entropy.

$$\begin{cases} A_i(q) = -\sum_{i=1}^{mk} p_i \cdot lb(p_i) \\ \tilde{A} = -\sum_{i=1}^{mk} (1/m_i \sum_{j=1}^{m} A_i(p_i)) \cdot lb(m_i/m) \end{cases} \quad \text{where } p_i = n/m_i \tag{5}$$

Step 5: Adjust the clustering coefficients according to the average information entropy. If the reach iteration number or the expected value of the average information entropy, jump to Step 5, otherwise jump to Step 3.

Step 6: The final clustering coefficient is calculated.

5 Similarity Measurement

Algorithm 3 (algorithms for similarity measurement)

Input: NSIHKS feature of the query example 3D model p
Output: the similarity between the query and the models in 3D model database
Initial conditions: A 3D model database which contains at least one 3D model q
Procedure:

Step 1: The similarity between model $p(x_1, x_2, \cdots x_i)$, and $q(x_1, x_2, \cdots, x_j)$ are calculated and normalized as follows Eq. (6):

$$d(x_i, x_j) = (\sum_{k=1}^{d} \left| x_{ik} - x_{jk} \right|^q)^{1/q} \tag{6}$$

Step 2: Start with the first row of one NSIHKS feature matrix. Matching two row in the different NSIHKS feature matrix when the minimum value is obtained by Eq. (6).
Step 3: Eliminate the matching points, jump to Step 2, until matching completed.

6 Experimental Result and Analysis

All algorithms proposed in this paper are implemented and tested using Matlab2016b on a PC with the following specifications-CPU: Intel(R) I5-3230, 2.60 GHz, RAM: 8 GB DDR3L, OS: Windows7 SP1 of 64 bits.

6.1 Datasets

In the experiment, we adopt international standard data set SHREC2010 [19]. This data set contains total 200 models files in off formats for 10 categories. Select two types of models from SHREC2010 library, show in Fig. 2. Their types are 'people' and 'pliers',

a b c d

Fig. 2. Two categories model examples

to explore the NSIHKS feature. Finally, the effect of algorithm is shown in retrieval results. And compared with other algorithms.

6.2 Analysis for NSIHKS Feature

The NSIHKS feature of the models are shown in Figs. 3 and 4. The color represents the NSIHKS feature of each vertex in the model at 1024 s. Analyze Fig. 3(a) and (b) model, person (a) model is in large scale, and person (b) model is in small scale. Because the NSIHKS feature is improved, it has scale transformation invariance. So you can see it in colors, the NSIHKS feature have good robustness in different scales of the same model. Figure 3 model (c) and (d) also reflect such a regularity. Analyze Fig. 3(a) and (c) model, for the same category of models, the NSIHKS feature also has good robustness under the same scale, where the shape have different form. The analysis of Figs. 3 and 4 shows that the feature of different category are obvious.

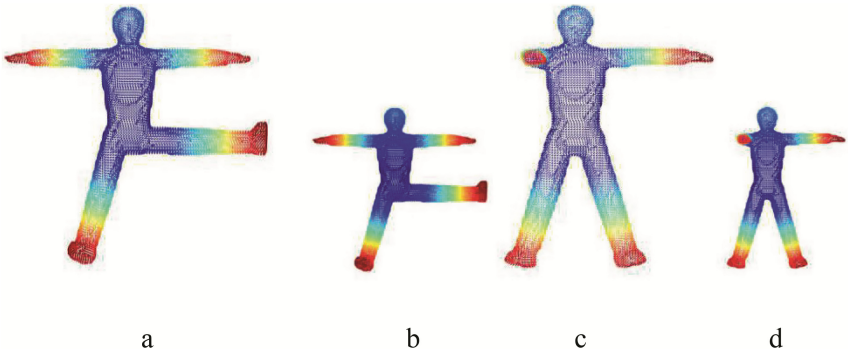


Fig. 3. NSIHKS feature of human models

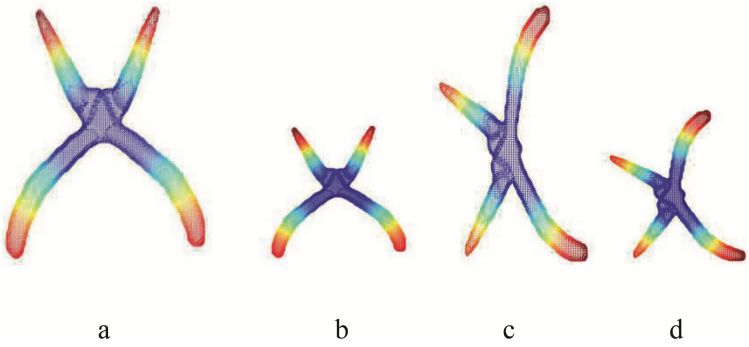


Fig. 4. NSIHKS feature of pliers models

6.3 Analysis for Retrieval Results

In the experiment, Calculate every vertex feature of each model, the first 200 minimum eigenvalues and eigenvectors of the Laplace-Beltrami operator are used. And $\alpha = 2$. The NSIHKS features of each vertex are 200 dimensional vector, obtain the 10 vector by screening. In the algorithm of clustering coefficient, the initialization coefficient is 5000 and the iteration number is 50. The retrieval results are shown in Fig. 5. It can be seen from the retrieval results that the method presented in this paper has a good retrieval effect.

Fig. 5. Retrieval results

6.4 Benchmarking with Other Algorithms

To evaluate the effectiveness of the method in this paper, compared with the algorithm SIHKS [17, 18], HKS [13], according to some single value evaluation indexes [19] and Precision-recall curves. Table 1 and Fig. 6 shows that NSIHKS is superior to SIHKS and HKS in the retrieval of non-rigid body library in SHREC2010.

Table 1. SHREC2010 evaluation index of retrieval effectiveness on non-rigid body library

Retrieval methods	NN	FT	ST	EM	DCG
NSIHKS	0.968	0.69	0.809	0.605	0.909
SIHKS	0.845	0.555	0.713	0.507	0.829
HKS	0.805	0.348	0.556	0.371	0.719

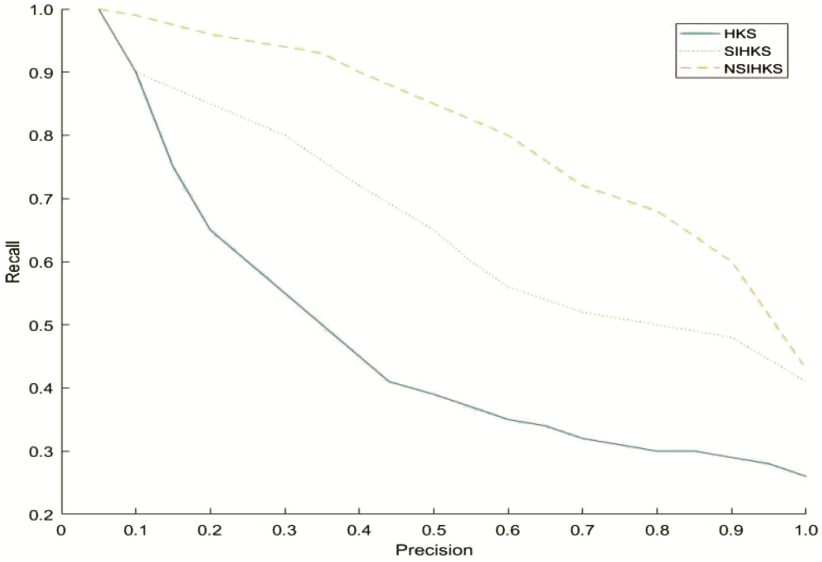


Fig. 6. Precision-recall curves among different algorithms in SHREC2010

7 Conclusions

3D model retrieval is an active research direction in computer graphic, machine vision and model recognition. Moreover it has extensive application. In this paper, a feature called NSIHKS is proposed which is applied to non-rigid 3D model retrieval. This method is improved to achieve high accuracy in retrieval and it shows good retrieval effect in the experiments. However, NSIHKS feature ignores model information in space. It does not have universal applicability because it is not suitable the models with the changes of topological structure. In the future work, we need to study features of model space information. Deep learning, genetic algorithm, data mining and analysis could extract the relation on model's space feature [20–25]. And deep learning has attracted interest in all fields and achieves good effect in every field. Hence, deep learning should be combined with 3D model retrieval research method. In this way we can make the recognition and retrieval intellectualize. Then improve retrieval effect and efficiency.

Acknowledgements. The authors would like to thank the editors and the anonymous reviewers for their constructive comments to further improve the quality of this paper. This work is partially supported by the following projects in China: the National Natural Science Foundation of China (No. 61602116), Natural Science Foundation of Guangdong Province (No. 2015A030313635, No. 2017A030313388), Science and Technology Project of Guangdong Province (No. 2014A010103037), Special Fund for Science and Technology Innovation of Foshan City (No. 2015AG10008, No. 2014AG10001, No. 2016GA10156), Education Department of Guangdong Province (No. 2015KTSCX153) and Outstanding Youth Teacher Training Program of Foshan University (No. FSYQ201411).

References

1. Liu, M., Vemuri, B.C., Amari, S.I., Nielsen, F.: Shape retrieval using hierarchical total Bregman soft clustering. IEEE Trans. Pattern Anal. Mach. Intell. **34**(12), 2407–2419 (2012)
2. Tangelder, J.W., Veltkamp, R.C.: A survey of content based 3D shape retrieval methods. Multimed. Tools Appl. **39**(3), 441–471 (2008)
3. Li, B., Schreck, T., Godil, A., Alexa, M.: SHREC'12 track: sketch-based 3D shape retrieval. In: Euro Graphics Workshop on 3D Object Retrieval, pp. 109–118 (2012)
4. Li, B., Lu, Y., Godil, A., Schreck, T., Aono, M., Johan, H., Saavedra, J.M., Tashiro, S.: SHREC'13 track: large scale sketch-based 3D shape retrieval. In: Euro Graphics Workshop on 3D Object Retrieval, pp. 89–96 (2013)
5. Chew, B.-S., Chau, L.-P., He, Y., Wang, D., Hoi, S.C.H.: Spectral geometry image: image based 3D models for digital broadcasting applications. IEEE Trans. Broadcast. **57**(3), 636–645 (2011)
6. Yu, M., Atmosukarto, L., Leow, W.K., Huang, Z., Xu, R.: 3D model retrieval with morphing-based geometric and topological feature maps. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 656–658 (2003)
7. Vyshali, S., Subramanyam, M.V., Soundara Rajan, K.: Topology coding in spectral domain for effective medical image retrieval. Int. J. Eng. Sci. Technol. **3**(11), 7884–7890 (2011)
8. Gao, Y., Tang, J., Li, H., Dai, Q.: View-based 3D model retrieval with probabilistic graph model. Neurocomputing **73**(10-12), 1900–1905 (2010)
9. Yang, Y., Lin, H., Zhu, Q.: A review of 3D model retrieval based on content. Comput. J. **27**(10), 1297–1310 (2004)
10. Reuter, M., Wolter, F.E., Peinecke, N.: Laplace-Beltrami spectra as 'Shape-DNA' of surfaces and solids. Comput. Aided Des. **38**(4), 342–366 (2006)
11. Lian, Z.H., Godil, A., Bustos, B., et al.: A comparison of methods for non-rigid 3D shape retrieval. Pattern Recogn. **46**(1), 449–461 (2013)
12. Rustamov, R.M.: Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In: Proceeding of the 5th Eurographics Symposium on Geometry Processing, pp. 225–233. Eurographics Association Press, Aire-la-Ville (2007)
13. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. Comput. Graph. Forum **28**(5), 1383–1392 (2009)
14. Zhou, Y., Zeng, F., Lu, Y., Zhou, Y.: Research on the rapid retrieval method of three-view model components in the manufacturing field. J. Sun Yat-sen Univ. (Natural Science Edition) (04) (2014)
15. Zhou, Y., Zeng, F.: 2D compressive sensing and multi-feature fusion for effective 3D shape retrieval. Inf. Sci. **409–410**, 101–120 (2017)
16. Ovsjanikov, M., Bronstein, A.M., Bronstein, M.M., et al.: Shape Google: a computer vision approach to isometry invariant shape retrieval. In: Proceedings of the International Conference on Computer Vision Workshops, Kyoto, Japan, pp. 320–327 (2009)
17. Bronstein, M.M., Kokkinos, I.: Scale-invariant heat kernel signatures for non-rigid shape recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, USA, pp. 1704–1711 (2010)
18. Kokkinos, I., Bronstein, M.M., Yuille, A.: Dense scale-invariant descriptors for images and surfaces. Center for Visual Computing, Ecole Centrale Paris (2012)
19. Shilane, P., Min, P., Kazhdan, M., et al.: The princeton shape benchmark. In: Proceedings of the 2004 International Conference on Shape Modeling and Applications, Genova, Italy, pp. 167–178 (2004)

20. Boscaini, D., Masci, J., Melzi, S., Bronstein, M., Castellani, M., Vandergheynst, P.: Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. Comput. Graph Forum **34**, 13–23 (2015)
21. Fang, Y., Xie, J., Dai, G., Wang, M., et al.: 3D deep shape descriptor. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2319–2328 (2015)
22. Sinha, A., Bai, J., Ramani, K.: Deep learning 3D shape surfaces using geometry images. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 223–240. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_14
23. Monti, F., Boscaini, D., Masci, J., et al.: Geometric deep learning on graphs and manifolds using mixture model CNNs (2016)
24. Qiu, M., Ming, Z., Li, J., Gai, K., Zong, Z.: Phase-change memory optimization for green cloud with genetic algorithm. IEEE Trans. Comput. **64**(12), 3528–3540 (2015)
25. Gai, K., Qiu, M., Sun, X.: A Survey on FinTech. J. Netw. Comput. Appl. **PP**, 1 (2017)

An Object Detection Algorithm for Deep Learning Based on Batch Normalization

Yan Zhou[1], Changqing Yuan[2(✉)], Fanzhi Zeng[1], Jiechang Qian[2], and Chen Wu[1]

[1] Department of Computer, Foshan University, Foshan 528000, Guangdong, China
zhouyan791266@163.com, coolhead@126.com, wu_chk@163.com
[2] School of Automation, Foshan University, Foshan 528000, Guangdong, China
1252919197@qq.com, 512502487@qq.com

Abstract. Based on the advantage of deep learning in object extraction, in this paper we design a deep network that adds Batch-Normalization to the convolution layer. Batch-Normalization has three main advantages. Firstly, it normalizes the input data, which can speed up the fitting of parameters. Secondly, Batch-Normalization can reconstruct the distribution of the input data, so that the feature of input data will not be lost. Thirdly, Batch-Normalization is able to prevent over-fitting, so it can replace Dropout, Local Response Normalization to simplify the network. The network in this paper adopted region proposal to get region of interests. Training classification and position adjustment at the same time to improve accuracy. Comprehensive experimental results have demonstrated the efficacy of the proposed network for objects detection.

Keywords: Deep learning · Batch Normalization · Object detection
Distribution reconstruction

1 Introduction

With the development of intelligence, object extraction on images has become an important technology and has been widely used in all fields of life, for example, in fields of machine vision, intelligent transportation, security monitoring, aerospace and military. In general, features of objects detection can be roughly divided into four categories, i.e. Color-based feature extraction, Texture-based feature extraction, Shape-based feature extraction, and method based on Fusion features. Color-based feature express the color range, color space distribution etc. Zeng et al. designed an algorithm [1] according to the color symmetry of the human body to form a lower feature dimension.

Texture-based object detection method describe the distribution of the value of the Pixels. Mu et al. proposed an improved LBP method [2], firstly, a binary image was generated by LBP, secondly a two-dimensional histogram was used to generate eigenvectors, this method abstracted out features with fewer dimensions from the local binary vector. Wu et al. proposed a faster object detection method CENTRIST [3],

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 438–448, 2018.
https://doi.org/10.1007/978-3-319-73830-7_43

where the method of block scanning is used to obtain the eigenvector of each block without the need of the eigenvector of the whole image. The block eigenvector can be perfectly connected with the linear classifier to achieve pedestrian detection, which can greatly improve the detection speed.

Shape-based feature extraction approach uses the calculated vectors to describe the contour features. Zhou et al. proposed a face recognition method using an improved SIFT feature [4]. Based on the SIFT approach of extracting the stable feature points from the whole image, this method proposes a feature extraction method in 6 key regions, which greatly reduced the area to be scanned. Dalal and Triggs proposed HOG feature [5], in which the original image is converted into a gradient image and weight the gradient value to the gradient discrete direction to obtain a gradient direction histogram. Since HOG features can well describe the outline of the human body, it has good effect on pedestrian detection. Lienhart and Maydt proposed Hear-like feature [6], where the algorithm uses a variety of templates to perform block-like integral operations similar to Hear features, which can represent the more features of many different objects. Compared with Hear features, Hear-like features have wider applicability in object extraction.

Method based on fusion features takes advantage of different features. Yang and Yang proposed a method of mixed features [7] for vehicle detection, where combines a global detector for scanning the entire window with HOG algorithm and a partial detector for partial scanning with LBP algorithm, which has good effect in vehicle detection. Ke and Sukthankar combined SIFT features with PCA [8], to improve the detection speed; Walk et al. proposed a multi-feature fusion method on pedestrian detection [9], where combines HOG features, CSS features and Optical Flow features to form multi-features. The integration of multi-features makes pedestrian detection very accurate, however, due to the high dimensionality of the eigenvectors of this method, the detection speed is relatively slow.

All the local features are classified by machine learning. The most commonly used classifiers are SVM, HIKSVM [10, 11], and Adaboost [12]. To sum up, these methods of feature extraction and classification are complex, and the accuracy is high only when these features are applied to a specific object.

In this paper, we propose an algorithm for deep learning based on Batch Normalization (BN), using deep learning classifier instead of machine learning classifier. Moreover, the BN layer can not only replace the previous Dropout layer, L2 Regular term, Local Response Normalization, but also can greatly improve the learning rate so that training speed greatly improved without affecting the classification effect. We do not need to pursue the accuracy of the initial parameters when initializing the parameters, only to initialize randomly, making the training converge rapidly.

The remainder of this paper is organized as follows: In Sect. 2, we give a brief review of related works. In Sect. 3, the general framework of the algorithm is proposed in this paper. In Sect. 4, the experimental results are discussed and analyzed. In Sect. 5, we analyzed the conclusion of this paper and discussed future work.

2 Related Works

In the past few years, with the computer speed improved, the application of neural network in object detection began to attract people's attention, especially in deep learning. The original method of neural network uses window-sliding to extract convolution features, which is a very slow because the entire image has to be scanned. Therefore, Girshick et al. proposed a convolutional neural network based on region proposals [13]. Although this method does not need to scan the whole image, it is necessary to extract the convolution feature for each region proposals, so the speed is still very slow. He et al. proposed a pyramid-based network structure [14], where only one convolution operation is required, the calculation speed is much higher than the algorithm in [13], but this method is not accurate enough to locate the location of the object. So Girshick proposed an improved algorithm [15], it has two main improvements over the pyramid-based network structure. The first one is the ROI-pooling layer, for each ROI region, we can find the corresponding position in convolution feature map. The role of ROI-pooling layer is pooling the convolution feature corresponding to the ROI to a fixed size. The second one is the classifier that contains Bbox regression and class scores. Ren et al. Proposed an algorithm [16] that uses RPN structure instead of region proposal to speed up objects detection. Ioffe and Szegedy proposed an algorithm [17], in which the distribution of the input data can be reconstructed, so as to accelerate the training speed. He et al. proposed a method of judging the category of objects at the pixel level [18], which can quickly get the outline of objects. The literature [19, 20] achieved good image restoration effect by Generative Adversarial Networks. Zhou and Zeng proposed an algorithms [21], where uses deep learning on 3D model retrieval. The literature [22, 23] proposed algorithms to recognize the attacks of wireless communications. when combined neural network, the algorithms will be more prominent. Compared with the local features, deep learning has the characteristics of simple algorithm, strong applicability and high accuracy.

3 The Detection Algorithm

3.1 The BN Transform

The role of a BN layer is to normalize the data to a random distribution space with a mean of 0 and a variance of 1, and make the distribution of input data and output data roughly the same, reducing the times of parameter changes during training. The normalized formula is Eq. (1):

$$\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}] + \in}} \tag{1}$$

Where $x^{(k)}$ represents a batch of data, $E[x^{(k)}]$ represents the mean of the batch, and $Var[x^{(k)}]$ represents the variance of the data.

However, only normalization will destroy the distribution of the characteristics of the original sample. In order to reconstruct the distribution of the original space input

data, we need a set of adaptive reconstruction parameters, we use Eq. (2) to find these parameters on the basis of Eq. (1):

$$y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)} \tag{2}$$

Where: $\gamma^{(k)}$, $\beta^{(k)}$ is the reconstruction parameters, these two parameters is BN to train the parameters.

Algorithm 1 (Training a Batch-Normalized Network)

Inputs: Trainable parameters $\gamma^{(k)}$, $\beta^{(k)}$; Batch size input data $\{x^{(1)}, x^{(2)} \ldots x^{(k)}\}$.
Outputs: The normalized data for each batch: $y^{(k)}$.
Step.1: if loss $= y^{(k)} - x^{(k)} >$ T do
Step.2: Initialization parameters: get $\gamma^{(k)}$, $\beta^{(k)}$ randomly.
Step.3: for k = 1 … K do
Step.4: Compute $y^{(k)}$in Eq. (2)
Step.5: end for
Step.6: end if
Step.7: for k=1 … K do
Step.8: Save the mean of the batches:

$$E[x] = E_{\beta}(\mathrm{E}[\mathrm{x}^{(\mathrm{k})}]) \tag{3}$$

Save the variance of the batches:

$$Var[x] = \frac{m}{m-1} E_{\beta}[\mathrm{var}(x^{(k)})] \tag{4}$$

Step.9: Compute $y^{(k)}$, the function can be expressed as:
// For clarity, $\gamma = \gamma^{(k)}$, $\beta = \beta^{(k)}$

$$y = \frac{\gamma}{\sqrt{Var[x] + \in}} \cdot x + (\beta - \frac{\gamma E[x]}{\sqrt{Var[x] + \in}}) \tag{5}$$

Step.10: end for.

3.2 The Training Algorithm

For deep learning, the core of training is to adjust parameters through iterative back-propagation operations according to the detected loss. In summary, the training algorithm of this article, namely the main algorithm, can be expressed by the following process:

Algorithm 2 (Training Algorithm)

Inputs: Image sequence $\{\mathrm{Im}\}^{\mathrm{n}}$
Outputs: detection result, $\{\mathrm{Bbox}\}_{\mathrm{n}}^{\mathrm{k}}$, $\{\mathrm{Scores}\}_{\mathrm{n}}^{\mathrm{k}}$
Step.1: The region proposals (ROI) are formed by inter-pixel correlation aggregation [11], and each training photo can form up to 2500 region proposals. the region proposals is extracted as Eq. (6):

$$s(r_i, r_j) = a_1 s_{color}(r_i, r_j) + a_2 s_{texture}(r_i, r_j) + a_3 s_{size}(r_i, r_j) + a_4 s_{fill}(r_i, r_j) \quad (6)$$

where $a_i \in [0, 1]$, represents whether the corresponding feature is used or not.
Step.2: for **abs**($\mathrm{l}^{(\mathrm{k})} - \mathrm{l}^{(\mathrm{k}-1)}$) $\leq$ 0.01; k++
Step.3: Normalize the input data using Algorithm 1.
Step.4: After multi-layer convolution, batch normalization and pooling to form a feature map of m $\times$ n $\times$ 256 dimensions.
Step.5: For each ROI region, the ROI convolution feature is mapped to a convolution feature map by a certain ratio. The ROI-Pool structure downsamples the feature dimensions of the ROI region in a fixed dimension of 7 $\times$ 7 to form a 7 $\times$ 7 $\times$ 256 dimensional feature map.
Step.6: Forming a highly abstract feature of 4096 dimensions through two full convolution networks.
Step.7: Classifying and locating at the same time in the classification process [13], the classification of 1000 objects can be classified, and the method of bounding box regression can be used to locate the object.
Step.8: The loss function is the sum of the classification loss and position loss, shown in Eq. (7):

$$\mathrm{l}^{(k)} = \begin{cases} \mathrm{l}_{\mathrm{cls}} + \lambda \mathrm{l}_{\mathrm{loc}} & \mu \text{ is foreground} \\ \mathrm{l}_{\mathrm{cls}} & \mu \text{ is background} \end{cases} \quad (7)$$

Where $\mathrm{l}^{(\mathrm{k})}$ is the loss of k epoch, $\mathrm{l}_{\mathrm{cls}}$ is the classification classification loss, determined by the probability of true classification u, as shown in Eq. (8):

$$\mathrm{l}_{cls} = -\log P_u \quad (8)$$

$\mathrm{l}_{\mathrm{loc}}$ is the position loss, which is the loss between the predicted parameter t^{u} and the real need to translate the scaling parameter t^{*}:

$$\mathrm{l}_{loc} = \sum_{i=1}^{4} g(t_i^u - t_i^*) \quad (9)$$

In Eq. (9) g(x) is the Smooth L_1 loss function, and shown in Eq. (10):

$$g(x) = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| - 0.5 & \text{others} \end{cases} \tag{10}$$

Step.9: end for.

4 General Framework

The entire deep network consists of five convolution pool layers, one layer of ROI pool, one classifier, the overall structure can be shown in Fig. 1.

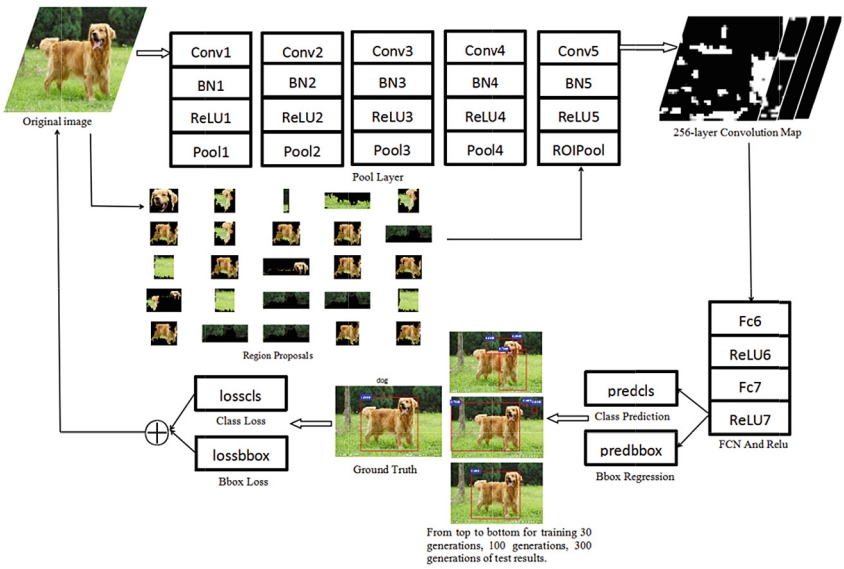


Fig. 1. General framework

In Fig. 1, Region Proposals (ROI) are obtained from Original image and used to combine Convolution Map, so that the network can get the convolution feature of each ROI. The end of the network is a classifier for Bbox regression and class score.

5 Experience Result and Analysis

The experiment is based on the i5 CPU running on the MATLAB platform. The training data is based on the VOC2007 data set. The parameters are initialized to a random number between 0 and 1. In order to speed up the training speed, we made the following improvements to the experiment:

(1) Do not use Dropout layer, LRN layer, try to use less convolution pool layer, so that the depth of the structure is limited to 28 layers.
(2) Reduce the size and frame size of the VOC 2007 data set to one half at the same time.
(3) Using larger learning rate 0.01, so that the increase the amount of change in parameters.

5.1 Experimental Result

(1) Feature Extraction:

Figure 2 shows the partial feature map on the fifth convolution layer. The difference object makes the convolution feature different. Every object has it's own feature map.

Fig. 2. Feature map

(2) Experimental Results:

The detection results are shown in Fig. 3, it can be seen that the algorithm has a high accuracy when detecting single object and two overlapping objects, and the detection effect is still good when multiple objects overlap, and only in very complicated situation, detection occurs deviation.

Fig. 3. Detection results

5.2 Experimental Analysis

(1) Average Precision: the average accuracy can be expressed as the Eq. (11):

$$AP_i = \int_0^1 P_i(R)dR_i \tag{11}$$

Where P_i represents the accuracy rate, R_i represents the recall rate, i = 1, 2, 3,…, n.

Figure 4 is the experimental accuracy of some of the object records, from the figure can be seen that most of the categories of accuracy are more than 0.55.

(2) mean Average Precision: mean Average Precision can be is expressed as Eq. (12):

$$mAP = \sum_{i=1}^{10} AP_i \Big/ 10 \tag{12}$$

The accuracy of the deep learning networks is higher than that of SPP-NET, which is comparable to that of R-CNN (AlexNet), but is lower than the accuracy of R-CNN (VGG16) and Fast-RCNN (VGG16) (Table 1). As for the training epoch, the convergence epoch of this paper is only 800, much lower than the training epoch of other methods.

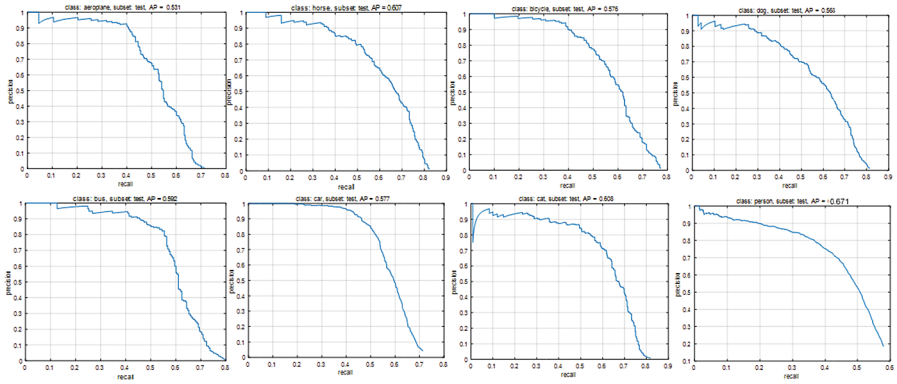


Fig. 4. Average Precision

Table 1. Compares the experimental results of several deep learning object detection networks

Networks	mean Average Precision	Convergent epoch
R-CNN (AlexNet)	58.5%	>2000
R-CNN (VGG16)	66.0%	——
SPP_net(ZF-5)	54.2%	——
Fast-RCNN (VGG16)	70.0%	>2000
Network in this paper	58.3%	800

6 Conclusion

Due to its wide applicability, high accuracy, concise algorithm and deep learning has been widely used in object detection. In the past two years, deep learning not only has made breakthroughs in object detection, but also has brought new applications in fields of machine vision, intelligent transportation and military informatization, such as human-computer interaction, automatic driving, target tracking, etc. Under the premise of ensuring the accuracy, deep learning network of this paper significantly reduced training epoch, and the method in this paper still has a good effect on object detection when the configuration requirements are not high. Moreover, the detection accuracy and speed are high when the objects are not complex. In the future research, we will further use deep learning to improve the accuracy and training speed of object detection, achieve object protection based on object detection, and add deep learning features to retrieve 3D models.

Acknowledgements. The authors would like to thank the editors and the anonymous reviewers for their constructive comments to further improve the quality of this paper. This work is partially supported by the following projects in china: National Natural Science Foundation of China (No. 61602116), Natural Science Foundation of Guangdong Province (No. 2015A030313635, No. 2017A030313388), Science and Technology Project of Guangdong Province (No. 2014A0 10103037), Special Found for Science and Technology Innovation of Foshan City

(No. 2015AG10008, No. 2016GA10156, No. 2014AG10001), Education Department of Guangdong Province (No. 2015KTSCX153) and Outstanding Youth Teacher Training Program of Foshan University (No. FSYQ201411).

References

1. Zeng, B., Wang, G., Lin, X.: Real-time pedestrian detection based on color self-similarity. J. Tsinghua Univ. (Sci. Technol.) **52**(04), 571–574 (2012)
2. Mu, Y., Yan, S., Liu, Y., et al.: Discriminative local binary patterns for human detection in personal album. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–8. DBLP (2008)
3. Wu, J., Geyer, C., Rehg, J.M.: Real-time human detection using contour cues. In: IEEE International Conference on Robotics and Automation, pp. 860–867. IEEE (2011)
4. Zhou, Z., Yu, S., Zhang, R., Yang, X.: A method of face recognition based on SIFT operator. J. Image Graph. **13**(10), 1882–1885 (2008)
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 886–893. IEEE Xplore (2005)
6. Lienhart, R., Maydt, J.: An extended set of Haar-like features for rapid object detection. In: 2002 Proceedings of International Conference on Image Processing, vol.1, pp. I-900-I-903. IEEE (2002)
7. Yang, X., Yang, Y.: A high efficiency vehicle detection method based on HOG-LBP. Comput. Eng. **09**, 210–214 (2014)
8. Ke, Y., Sukthankar, R.: PCA-SIFT: a more distinctive representation for local image descriptors. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, vol. 2, pp. II-506-II-513. IEEE (2004)
9. Walk, S., Majer, N., Schindler, K., et al.: New features and insights for pedestrian detection. In: Computer Vision and Pattern Recognition, pp. 1030–1037. IEEE (2010)
10. http://www.dataguru.cn/thread-371987-1-1.html
11. Maji, S., Berg, A.C., Malik, J.: Classification using intersection kernel support vector machines is efficient. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–8. DBLP (2008)
12. Freund, Y., Schapire, R.E.: A desicion-theoretic generalization of on-line learning and an application to boosting. In: Vitányi, P. (ed.) EuroCOLT 1995. LNCS, vol. 904, pp. 23–37. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-59119-2_166
13. Girshick, R., Donahue, J., Darrell, T., et al.: Region-based convolutional networks for accurate object detection and segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **38**(1), 142–158 (2016)
14. He, K., Zhang, X., Ren, S., et al.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1904–1916 (2015)
15. Girshick R.: Fast R-CNN. In: IEEE International Conference on Computer Vision, pp. 1440–1448. IEEE Computer Society (2015)
16. Ren, S., Girshick, R., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. In: nternational Conference on Neural Information Processing Systems, pp. 91–99. MIT Press (2015)
17. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456 (2015). JMLR.org

18. He, K., Gkioxari, G., Dollár, P., et al.: Mask R-CNN (2017)
19. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. In: ICLR (2017)
20. Zhao, J., et al.: Energy-based generative adversarial networks. In: ICLR (2017)
21. Zhou, Y., Zeng, F.: 2D compressive sensing and multi-feature fusion for effective 3D shape retrieval. Inf. Sci. 101–120 (2017)
22. Gai, K., Qiu, M., Ming, Z., Zhao, H., Qiu, L.: Spoofing-jamming attack strategy using optimal power distributions in wireless smart grid networks. IEEE Trans. Smart Grid **8**(5), 2431–2439 (2017)
23. Gai, K., Qiu, M., Tao, L., Zhu, Y.: Intrusion detection techniques for mobile cloud computing in heterogeneous 5G. Secur. Commun. Netw. **9**(16), 3049–3058 (2016)

Towards a Novel Protocol Analysis Framework for Industrial Control Systems

Jiye Wang[1], Liang Zhou[2], Xindai Lu[3(✉)], Huan Ying[2], and Haixiang Wang[4]

[1] China Electric Power Research Institute, Beijing, China
wangjiye@epri.sgcc.com.cn
[2] Information and Communication, China Electric Power Research Institute, Beijing, China
{zhouliang,yinghuan}@epri.sgcc.com.cn
[3] Electric Power Research Institute, State Grid Zhejiang Electric Power Company, Hangzhou, China
lu_xindai@zj.sgcc.com.cn
[4] Graduate School, China Electric Power Research Institute, Beijing, China
a6215916@qq.com

Abstract. Nowadays industrial controls systems (ICS) are becoming more and more robust and intelligent, owing to the development of industrial networking technology. While, on the other hand, security issues arise and pose great challenges. Among these issues, the security of ICS protocols receives the attention from both academy and industry in recent years. Due to the close and proprietary nature of industrial protocols, it is difficult to conduct protocol analysis and protection on these protocols. To address this issue, we propose a novel protocol analysis framework, named ICS-PAS, for ICS protocols. ICS-PAF could differentiate unknown protocols and their command types, extract protocol format and recognize the data types of protocol payloads. In addition, ICS-PAF could also infer and model the state transition of ICS protocols. ICS-PAS requires no prior knowledge and could deal with binary protocols. We also conduct comprehensive experiments to verify the performance of ICS-PAS. The results show that ICS-PAS outperforms traditional approaches in terms of recognition accuracy and efficiency.

Keywords: Industrial Control Systems · Protocol analysis · Security

1 Introduction

Network protocol analysis plays an important role in network security. For example, most viruses and Trojans use private protocols to communicate. In industrial control systems, most protocols are designed and implemented by device manufactures, and are usually proprietary protocols. To detect intrusions and defends against virus and worms, nowadays security tools heavily rely on protocol analysis such as deep packet inspection. Therefore, the analysis for unknown or proprietary protocols is an urgent need for network security.

M. Qiu (Ed.): SmartCom 2017, LNCS 10699, pp. 449–456, 2018.
https://doi.org/10.1007/978-3-319-73830-7_44

ICS Protocol analysis relies heavily on artificial analysis, and requires lots of human labors. Researchers resort to automatic approaches [1–4] for parsing unknown protocols. However, there are still challenges remains to be addressed.

First, it is non-trivial to extract the format of ICS protocols since there is usually no delimiter which could be used to split packets into different segments. Second, most ICS protocols are proprietary ones, and there is little prior knowledge that could be used to help understanding ICS protocols. Third and last, the-state-of-the-art approaches show effectiveness in text protocols analysis, while are not suitable for parsing ICS protocols which are usually binary ones.

In this paper, we propose a novel framework, called ICS-PAF, for analyzing ICS protocols. We are not aiming at providing a complete solution, but take a solid step towards ICS protocol analysis. Our main contributions are summarized as follows:

- We propose an ensemble clustering approach to differentiate protocol commands without requiring prior knowledge.
- We adopt Needleman-Wunsch algorithm to split packets into different segments, and build state machine to represent protocol state transitions.
- We conduct comprehensive experiments to evaluate ICS-PAF, and the results show the effectiveness of ICS-PAF.

The reminder paper is structured as follows. We describe the background and related work first, and then present the design and implementation details of ICS-PAF. Then we conduct some experiments to demonstrate the effectiveness and efficiency of our approach. Finally, we close with the conclusion.

2 Background and Related Work

2.1 Background

We take Modbus/TCP as an example to show how to analyze ICS protocols. Modbus/TCP adds a header of the Modbus protocol data unit to the application layer on the TCP/IP protocol stack. The security flaws of this protocol lead to the attacker's attack on the industrial control system by constructing or tampering with the packets. The application data unit (ADU) of the Modbus application layer contains the header and protocol data unit (PDU). Modbus is a typical Request-and-Response ICS protocol, which packets always appear in pairs. Figure 1 shows the format of Modbus/TCP protocol.

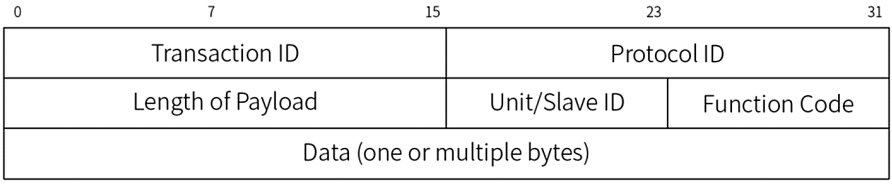


Fig. 1. Modbus packet format

Transaction ID is used for message synchronization. *Protocol ID* is a unique identifier for Modbus/TCP. *Unit/Slave ID* specifies the number of slave devices, ranging from 1–247 to 0 for broadcast. *Length of Payload* gives the total length of the message after this field. *Function Code* is 2 bytes long, which represents the type of command that the message contains. It represents the type and granularity of the operation data. *Function code* is in the range of 0–255. 0 is an illegal function code, and the most high position bit 0/1 is used to distinguish between normal and abnormal reply. Therefore, the request function code appears only in the range of 1–127. *Data* segment is more complex. The common-seen data segment is for reading and writing operations. Read request is composed of request address and quantity. Write request also includes the value to be written. Read response usually includes the number of bytes and the reply data; write response usually includes the address and quantity.

2.2 Related Work

In protocol data clustering, Cui et al. [3] proposed *Discover* to model and classify data frames according the strings which are transformed from the data frame. This approach can identify some string-based protocols (such as HTTP), but cannot handle binary protocols. Netzob [11] is a tool that can be used to reverse engineer, model and fuzz communication protocols. Netzob can analyze both text-based protocols and binary protocols.

Another research direction is leveraging machine-learning approaches. Gopalratnam et al. [4] proposes a supervised machine learning method for dividing protocol contents into different types. It requires large amounts of training data to refine the model and ensures a high-level recognition precision. However, it is non-trivial for collecting enough training samples for proprietary protocol data. Liang [5] resorted to supervised machine learning techniques to do data frame clustering. Another unsupervised learning method [6] uses boosting algorithm to implement the accuracy of clustering. To summarize, supervised machine learning methods can only identify very limited protocol formats. Unsupervised machine learning methods could only be used for protocol classification, but cannot identify specific learning information.

Some research work used Markov model to do protocol classification. Through the establishment of feature vectors, the unknown protocol was classified by forward learning method. This method has a certain ability to analyze private protocols. The third method is based on statistics. Pan et al. [8] proposed a protocol analysis method for mining binary sequences frequently. In terms of protocol format extraction, one classical work is the PI project which uses biological DNA sequence alignment method. By adding the blank character, it achieves string alignment and format parsing. The core of this project is the biological sequence alignment algorithm proposed by NeedleMan [7]. But it has the disadvantage of low fault-tolerance rate. It depends on correct classification so that the correct format could be extracted. Luo and Shunzheng [9] proposed a method using statistical frequent sequence and association rules to extract protocol formats from network packets. Lin et al. [10] proposes a system that abstracts protocol formats by executing instructions from protocol software. In the process of network protocol reverse parsing, a lot of data need to be processed.

3 Design and Implementation

3.1 Overview

As shown in Fig. 2, ICS-PAF divides unknown protocol analysis process into three phases: protocol command classification, protocol format extraction and protocol state inference. In protocol command classification phase, we design an ensemble-clustering approach for differentiate protocol commands phase; in protocol format extraction, we use classic Needleman-Wunsch algorithm to split protocol payloads into different parts; in protocol state inference phase, we infer protocol states and build state machines to represent state transitions.

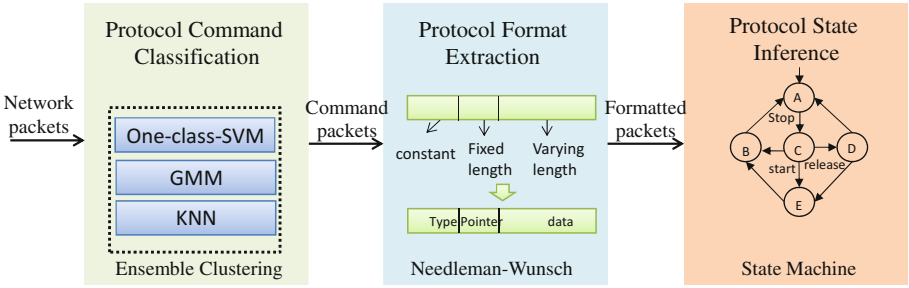


Fig. 2. ICS-PAF architecture

3.2 Protocol Command Classification

The objective of Protocol Command Classification is to classify ICS packets into different command types. Sine we are facing with unknown ICS protocols, it is impractical to obtain labeled training data. Therefore, we resort to unsupervised learning approaches to cluster network packets into different types.

Firstly we extract features from ICS network packets such as packet length, byte entropy, etc. Then, GMM, one-class-SVM and KNN Cluster algorithms are applied to network traffic and obtain preliminary clustering results. Lastly, the preliminary results are passed to ensemble clustering model and the final clustering results are given based on voting algorithm.

As for KNN algorithm, the key is to appropriately define the distance between two ICS packets. Therefore, we design a byte-sequence similarity metric to measure the distance between two network packets. We observed that, for ICS protocols, the bytes which are closer to the protocol header usually contain important information and play a more important role in differentiating ICS packets. For example, the function code field of ICS protocols is usually located in the first several bytes. Inspired by this observation, we design a heuristic to calculate the distance of ICS packets. Each byte in an ICS packet is given different weight according to its location in the packet.

3.3 Protocol Format Extraction

After classifying ICS packets into different types, we then extract protocol format. Generally speaking, an ICS packet consists of an identifier, a function code, data length area, pointer area, and payload area (usually variable). Currently, we could not recover the exact format of ICS protocols. We adopt Needleman-Wunsch algorithm to split packets into different data segments by identifying constant-value data, fixed-length data and varying-length data. To simplify the problem, we assume that ICS protocol packets are byte alignment and the varying-length data segment is always at the end of packets. We slightly modify Needleman-Wunsch algorithm to fit ICS protocols.

3.4 Protocol State Inference

Since it is impractical to obtain high-level information of ICS protocols, we build state machine based on low-level protocol states and transitions. However, there are no explicit declaration of states and transitions in ICS protocols. We need to infer the latent states of ICS protocols. As shown in Fig. 3, we build a state machine for each communication channel between two machines, such as HMI to PLC, or PLC to Server. A State represents the communication of two endpoints at one time point, and the state transition is driven by the command sent or received. However, in the actual protocol operation, there may be a state transition which is not consistent with the normal transition in the definition. At this time, it is necessary to add other possible state transition.

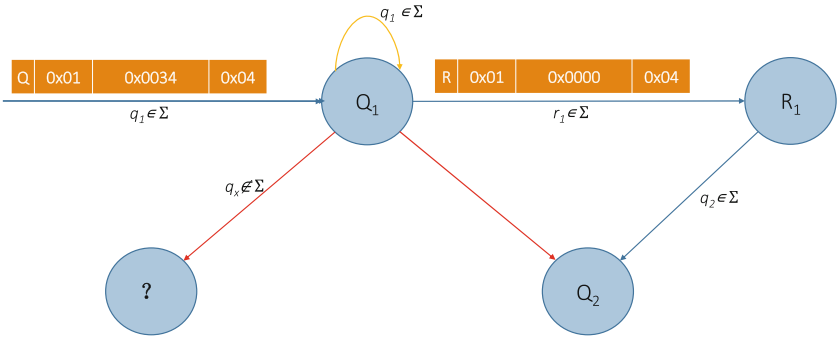


Fig. 3. An example of state machine for ICS protocol

Loss packet. When the state is at Q1, the command q2 is transferred to the Q2 state defined by Q2. The possible reason for the transition is packet loss or security incidents in the network.

Retransmission. When the state is in Q1, the command q1 appears again, and it is looped back to Q1. The possible reason for the transfer is retransmission or security incidents in the network.

Unknown transition. The last class is the commands that are not in the alphabet, then transfer back to the start of the state machine. Due to the existence of the rule of packet drop and retransmission, the state machine will soon turn back to the right state.

The occurrence of unknown transfer indicates that there is no operation in the training data, which is more likely to anomaly events and should be regarded as abnormal traffic.

3.5 Implementation

ICS-PAF could be divided into three modules. The first is data capturing module which captures network traffic from *pcap* file or directly from the network interfaces. The second is data parsing module which is responsible for parsing the content of network data. The parsing module is divided into the known protocol parsing module and the unknown protocol parsing module. The protocol parsing module works similar with Wireshark. The unknown protocol parsing module parses unknown ICS protocols. The unknown parsing module is divided into three parts, one is the unknown protocol classification module, the second is the unknown protocol format parsing module, and the third is the protocol state machine generation module. The unknown protocol classification module classifies the network data frame into different types, and gives the classification results and detailed information. In the unknown protocol format parsing module, the data frame set is abstracted and the general information of the data format is obtained. The third and last module is protocol state machine abstraction module, which establishes protocol conversion state machine through protocol format and protocol data frame information.

ICS-PAF is written in *Python* language. We use PyShark to parse network packets and obtain application-level payloads of ICS protocols. WxPython is used as GUI library to provide users a friendly user interface.

4 Experiment

To evaluate the effectiveness and efficiency of ICS-PAF, we conduct some experiments. Before demonstrate the experimental results, we first present the setup of our experiments.

Experimental Setup. The experiments were conducted on a desktop computer of Intel Core i5 CPU with 16G memory. The operating system is Ubuntu 14.04 with Linux kernel 3.4. To evaluate ICS-PAF, we prepare some network packets. We collect Modbus and IEC104 packets using Wireshark in real production environments. Besides, we also capture some HTTP and FTP packets and comparing the parsing results with ICS protocols. To evaluate the performance of ICS-PAF on unknown protocols, all these protocols are treated as unknown protocols in testing phase.

Exp 1. In this experiment we evaluate the accuracy of Protocol Command Classification of ICS-PAF. We compare our approach with Discover and Netzob using HTTP, FTP, Modbus and IEC104. As shown in Fig. 3, in HTTP command classification, ICS-PAF is 1% better than Discover and Netzob, however, the two approaches are slightly better than ICS-PAF. Three approaches show very good performance when parsing HTTP and FTP protocols. This is because HTTP and FTP are text-based protocols and easy to be classified. When dealing with ICS protocols, Modbus and IEC104, ICS-PAF

is much better than Discover and Netzob. For all the three approaches, the classification results of Modbus are better than the results of IEC104. The reason is Modbus is simpler than IEC104. In general, ICS-PAF has obvious advantage compared with the two competitors, Discover and Netzob when classifying ICS protocols.

Exp 2. In this experiment, we evaluate the effectiveness of protocol format extraction. Note that we adopt Needleman-Wunsch algorithm to handle ICS protocols. The performance of Needleman-Wunsch algorithm is dependent on the number of network frame data. Therefore, we test how the number of tested data will impact the performance of ICS-PAF. We take modbus as an example and the result is given in Fig. 4. We can see that the accuracy grows with the increasing of data frame numbers. At the beginning stage, the growth is fast and gradually become gentle when the data frame number reaches a certain extent. The results show that if we want to achieve reasonable accuracy, we should feed more data to the algorithm.

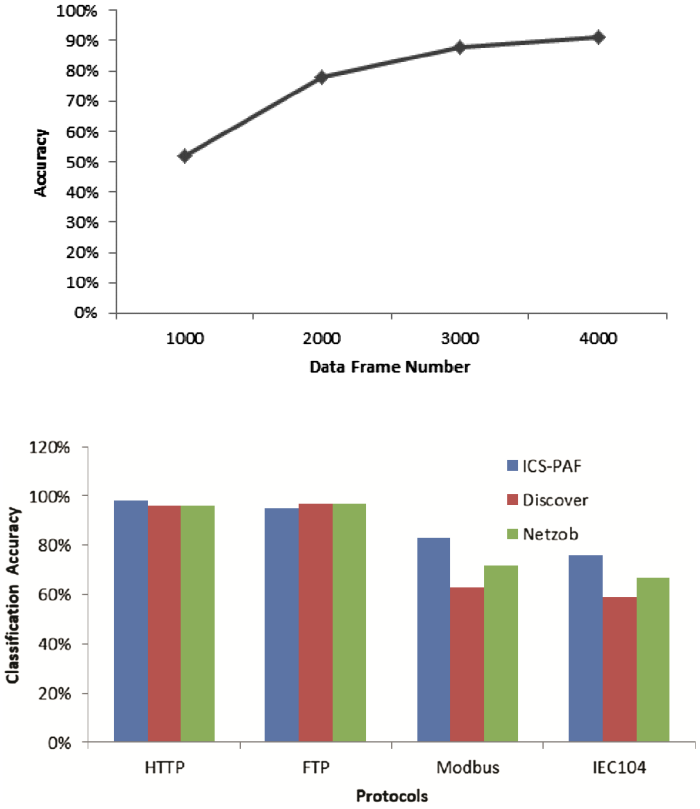


Fig. 4. Protocol command classification results

5 Conclusion

This paper proposed a novel approach named ICS-PAF to analyze ICS protocols. It leverages ensemble clustering approaches to classify network packets into different command types. A Classical algorithm is used to extract ICS format of ICS protocols. We also build state machine to characterize the state transition of ICS protocols. We implement ICS-PAF and uses ICS and non-ICS network packets collected from real-world environments to evaluate the performance of ICS-PAF. The experiment results show the effectiveness of ICS-PAF and demonstrate that ICS-PAF has taken a step towards unknown ICS protocol analysis.

Our future work is two-folds. The first is further improving the performance of our approach. The second is using more ICS protocols to evaluate the practicability of ICS-PAF.

Acknowledgement. The authors gratefully acknowledge the anonymous reviewers for their helpful suggestions.

References

1. Kang, H.-J., Kim, M.-S., Hong, J.W.-K.: A method on multimedia service traffic monitoring and analysis. In: Brunner, M., Keller, A. (eds.) DSOM 2003. LNCS, vol. 2867, pp. 93–105. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39671-0_9
2. Sen, S., Spatscheck, O., Wang, D.: Accurate scalable in-network identification of P2P traffic using application signature. In: Proceedings of the 13th International Conference on World WideWeb, Madrid, pp. 512–521 (2004)
3. Cui, W., Kannan, J., Wang, H.J.: Discover: automatic protocol reverse engineering from network traces. In: Proceeding of 16th USENIX Security Symposium on USENIX Security Symposium, Austin, pp. 1–14 (2007)
4. Gopalratnam, K., Basu, S., Dunagan, J., Wang, H.: Automatically extracting fields from unknown network protocols. In: First Workshop on Tackling Computer Systems Problems with Machine Learning Techniques
5. Liang: Clustering-based network application recogniztion system. Master thesis, Shandong University, China
6. Freund, Y., Schapire, R.: A short introduction to boosting. J. Jpn. Soc. Artif. Intell. **14**(5), 771–780 (1999)
7. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins (1970)
8. Pan, F., Hong, Z., Du, Y.: Efficient protocol reverse method based on network trace analysis. Int. J. Dig. Content Technol. Appl. **20**(6), 201–210 (2012)
9. Jianzhen, L., Shunzheng, Y.: Position-based automatic reverse engineering of network protocols. J. Netw. Comput. Appl. **36**, 1070–1077 (2013)
10. Lin, Z., Jian, X., Xu, D., Zhang, X.: Automatic protocol format reverse engineering through context-aware monitored execution. In: 15th Symposium on Network and Distributed System Security (NDSS). Internet Society (2008)
11. Netzob. https://github.com/netzob/netzob. Accessed 3 Nov 2017

Author Index