Lecture Notes in Electrical Engineering 264

Erol Gelenbe
Ricardo Lent *Editors*

Information Sciences and Systems 2013

Proceedings of the 28th International Symposium on Computer and Information Sciences

Springer

Lecture Notes in Electrical Engineering

Volume 264

For further volumes:
http://www.springer.com/series/7818

Erol Gelenbe · Ricardo Lent
Editors

Information Sciences and Systems 2013

Proceedings of the 28th International Symposium on Computer and Information Sciences

Springer

Editors
Erol Gelenbe
Department of Electrical and Electronics Engineering
Imperial College
London
UK

Ricardo Lent
Imperial College
London
UK

ISSN 1876-1100 ISSN 1876-1119 (electronic)
ISBN 978-3-319-01603-0 ISBN 978-3-319-01604-7 (eBook)
DOI 10.1007/978-3-319-01604-7
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013947374

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The 28th International Symposium on Computer and Information Sciences (ISCIS 2013) was held on 28th and 29th October 2013, and it was hosted at the Institut Henri Poincaré, which is France's institute for the mathematical sciences, in the Latin Quarter of Paris.

In addition to the papers that are contained in these proceedings, the symposium included three keynote presentations:

"**Resilience Techniques for High Performance Computing**" by Prof. Yves Robert (ENS Lyon)

"**Algorithms and Paradoxes in Wireless Networks**" by Dr. Philippe Jacquet (Alcatel-Lucent)

"**Cyber Security Challenges with Analytics and Visualisation**" by Dr. Robert Ghanea-Hercock (BT Exact)

The core of the proceedings is based on 42 papers, including 35 contributed papers, and seven additional papers that describe the EU FP7 NEMESYS project. All of the accepted papers were revised based on numerous referee reports.

The papers in these proceedings are grouped into seven sections:

- Smart Algorithms,
- Analysis Modelling and Optimisation,
- Computational Linguistics,
- Computer Vision,
- Data and Web Engineering,
- Wireless Sensor Networks,
- Network Security, Data Integrity and Privacy.

Each of the sections are the representative of very active research areas in information science and technology, and the papers in these Proceedings emanate from research groups or individuals in 13 different countries.

We thank all those who submitted papers to ISCIS 2013, especially the authors of papers that we were unable to accept, and all the authors whose papers are included in the proceedings. We also thank the programme committee members and numerous referees who contributed their time and effort and whose names appear in the acknowledgement page.

London, October 28, 2013

Erol Gelenbe
Ricardo Lent

Acknowledgment to Programme Committee Members and Referees

Yusuf Sinan Akgül
Timur Aksoy
Georgios Alexandridis
Ethem Alpaydin
Volkan Atalay
Cevdet Aykanat
Selim Balcisoy
Madalina Baltatu
Javier Barria
Mustafa Battal
Olivier Beaumont
Jeremy Bradley
Manfred Broy
Fazli Can
Bin Cao
Berkay Celik
Suheyla Cetin
Nazife Cevik
Sophie Chabridon
Zhenyu Chen
Ilyas Cicekli
Nihan Cicekli
Johanne Cohen
Tadeusz Czachorski
Gokhan Dalkilic
Laurent Delosieres
Mariangiola Dezani
Ozlem Durmaz Incel
Nadia Erdogan
Betul Erdogdu
Güneş Erçal
Gülşen Eryiğit
Engin Erzin
Taner Eskil
Yorgo Istefanopulos
Alain Jean-Marie
Kamer Kaya
Florian Kohlmayer
Stefanos Kollias
Ibrahim Korpeoglu
Dilek Küçük
Marco Kuhrmann
Olcay Kursun
Ricardo Lent
Albert Levi
Aristidis Likas
George Limperopoulos
Peixiang Liu
Rosa Meo
Chris Mitchell
Pabitra Mitra
Christos Nikolaou
Cemal Okan Sakar
Sema Oktug
Ender Ozcan
Oznur Ozkasap
Nihal Pekergin
Yves Robert
Alexandre Romariz
Georgia Sakellari
Huseyin Seker
R. Oguz Selvitopi
Ercan Solak
Sovanna Tan
Andreas Stafylopatis
Eleni Theodoropoulou
Dao Thi
Nigel Thomas

Dimitris Fotakis
Jean-Michel Fourneau
Florence Gajić
Erol Gelenbe
Stephen Gilmore
Jovan Golic
Lin Guan
Ugur Gudukbay
Ergun Gumus
Attila Gursoy
Ugur Halici
Peter Harrison
Richard Hayden
Hakki Toroslu
Salvatore Tucci
Dimitrios Tzovaras
Ozgur Ulusoy
Gülnur Selda Uyanik
Ozlem Uzuner
Zhiguang Xu
Adnan Yazici
Cemal Yilmaz
Arda Yurdakul
Thomas Zeugmann
Qi Zhu

Contents

Part VII Network Security, Data Integrity and Privacy

Part I
Smart Algorithms

Adaptive Curve Tailoring

Trond Steihaug and Wenli Wang

Abstract This paper deals with the problem of finding an average of several curves subject to qualitative constraints and restrictions on the curves. The unknown average curve is the solution of a weighted least squares problem involving the deviation between the given curves and the unknown curve. The qualitative constraints are that some curves are preferred compared to other curves. The qualitative information is converted into constraints on the weights in the least squares problem defining the average curve. The model defining the curves is parameterized and restrictions on the curves are defined in terms of restrictions on the parameters. We give an example where the curves determined from three data sets are required to be monotone and convex. We also show that one curve being preferred restricts the set of possible curves that can be an average curve.

1 Introduction

In this paper we discuss a least squares problem where the purpose is to find an average of several curves where each curve is determined by a set of datapoints. A decision maker compares pairs of curves and rates one curve better than the other. The problem of finding the average curve may be regarded as a multiobjective optimization problem combining quantitative (curves and datapoints) and qualitative (comparisons) information. Each objective is a measure of the deviation between the unknown curve and one of the given curves. The multiobjective optimization problem is converted into a single objective problem where the objective function is a weighted sum of deviations from the unknown curve and the given curves. The weights are determined from the choices made by decision maker.

Dennis and Woods [1, 2] consider an interactive technique for solving multiobjective optimization presented in the framework of the curve fitting problem. It is

T. Steihaug (✉) · W. Wang
Department of Informatics, University of Bergen, Bergen, Norway
e-mail: Trond.Steihaug@ii.uib.no

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_1,

assumed that the user would be content with parameters that minimize the weighted sum of deviation between the data and the curve. The user supplies interactively qualitative information that one set of parameters is preferred above another set of parameters. From this information Dennis and Woods [1, 2] develop a technique to deduce the weights in the minimization problem. Nordeide and Steihaug [3] introduced the concept of a value function to represent the qualitative information given as pairwise comparisons.

An example of such a procedure is found in personalization of hearing instruments. Modern hearing aids contain advanced signal processing algorithms with many parameters that need to be tuned. Because of the large range of the parameter space the tuning procedure becomes a complex task. Some of the tuning parameters are set by the hearing aid dispenser based on the nature of the hearing loss. Other parameters may be tuned on the basis of the models for loudness perception. However, to set the parameters to values that ideally match the needs and preferences of the user an adaptive procedure is needed. In order to cope with the various problems for tuning parameters prior to device usage, [4] describes a method to personalize the hearing aid to actual user preferences.

The example we will describe in the paper is based on laboratory measurements of relative permeability data of core samples based on steady state fluid flow. A core sample is a cylindrical section of a sediment and assumed to be saturated by one of the phases. The fluid (a mixture of the phases oil and gas, oil and water, or gas and water) is introduced at the inlet of the core with a predetermined ratio of the two fluids and injected until the ratio of output is equal the input ratio. The saturation conditions of the core sample is then measured. The whole process is repeated with a new ratio of the fluids. Nearby core samples will give different relative permeability curves and the petroleum engineer will choose an average curve based on a qualitative knowledge of the samples and type of laboratory measurements.

The least squares problem and average curve are formulated in Sect. 2. In Sect. 3 we assume that the model is composed of a finite set of basis functions. The least squares problem will be a linear least squares problem in the parameters. Each curve will be parameterized with the same set of parameters, but with different values. In Sect. 4 we will show that the qualitative information provided by one set of parameters being preferred above another set can be used to limit the possible set of curves that can be defined as an average of the given curves. Section 5 is a short summary of properties of shape preserving spline functions. In Sect. 6 we give a numerical example using the theory developed in the previous sections based on relative permeability curves of core samples from the North Sea.

2 The Least Squares Problem

Given the data points (ξ_i, υ_i), $i = 1, \ldots, N_0$ where each $(\xi_i, \upsilon_i) \in [a, b] \times \mathbb{R} \subset \mathbb{R}^2$. Further, we are given a model $g(\xi, x)$, $g : [a, b] \times \mathbb{R}^m \to \mathbb{R}$ which depends on a set of parameters $x \in \mathbb{R}^m$ to be determined. In many cases additional conditions are

imposed on the model. An example can be that the model should be non-decreasing in the variable ξ. Other examples are discussed in Sect. 5. In the context of tuning hearing aids, the additional conditions could be it should match the hearing loss.

Let the set of possible values of the parameter vector x be $\mathscr{C} \subset \mathbb{R}^m$. In this paper the set $\mathscr{C}$ will be convex and in the example used in the numerical section $\mathscr{C}$ is a polyhedron. Assuming the weights $W_1, \ldots, W_{N_0}$ are all positive, the weighted least squares problem is to solve

$$\min_{x \in \mathscr{C}} \sum_{i=1}^{N_0} W_i [\upsilon_i - g(\xi_i, x)]^2. \tag{1}$$

Given several data sets $(\xi_{i,l}, \upsilon_{i,l})$, $i = 1, \ldots, N_l$, $l = 1, 2, \ldots, N$ and that the same model with different values of parameters adequately represents the data, let x^l denote the solution of (1) using data set l, i.e. x^l is a solution of

$$\min_{x \in \mathscr{C}} \sum_{i=1}^{N_l} W_{i,l} [\upsilon_{i,l} - g(\xi_{i,l}, x)]^2.$$

For two values x and y of the parameters let $e(x, y)$ be a measure of deviation between the models $g(\cdot, x)$ and $g(\cdot, y)$. A suitable function $e(x, y) = \|g(\cdot, x) - g(\cdot, y)\|^2$ where $\|\cdot\|$ is an inner product norm is

$$e(x, y) = \int_a^b W(\xi)[g(\xi, x) - g(\xi, y)]^2 \, d\xi \tag{2}$$

where $W(\xi)$ is a positive weight function or let $e(x, y)$ be a numerical integration rule like the composite Trapezoidal rule of (2).

The formulation of the problem of finding the average curve involving N data sets is in this paper the problem:

$$\min_{x \in \mathscr{C}} \sum_{l=1}^{N} w_l \, e(x^l, x). \tag{3}$$

If x is a solution of (3), then $g(\cdot, x)$ is said to be an average of the curves $g(\cdot, x^1), \ldots, g(\cdot, x^N)$ with respect to the weights $w_1, \ldots, w_N$.

Throughout this paper it is assumed that the weight function $W(\cdot)$ and the model $g(\cdot, x)$ are such that if the error function $e(x, y) = 0$ then the two realizations with parameters x and y are equal pointwise, $g(\xi, x) = g(\xi, y)$ for all $\xi \in [a, b]$.

3 The General Model

Let $\{B_i\}_{i=1}^m$ be basis functions where each $B_i : [a, b] \subset \mathbb{R} \to \mathbb{R}$ and consider the model

$$g(\xi, x) = \sum_{i=1}^{m} x_i B_i(\xi) = b(\xi)^\top x \tag{4}$$

where $b(\xi)^\top = (B_1(\xi), B_2(\xi), \ldots, B_m(\xi))$. For this choice of basis functions, the objective function in (1) is a quadratic function in the parameter x. Define the $N_0 \times m$ matrix C where $C_{i,j} = B_j(\xi_i)$, i.e. row i in C is $b(\xi_i)^\top$, and let $W = diag(W_1, \ldots, W_{N_0})$. The least squares problem (1) is equivalent to the problem

$$\min_{x \in \mathscr{C}} \|W^{1/2}(Cx - v)\|_2^2,$$

or written as a quadratic problem

$$\min_{x \in \mathscr{C}} \frac{1}{2} x^\top A x - q^\top x, \tag{5}$$

where $A = C^\top W C$ and $q = C^\top W v$. Methods for solving (5) will depend on the constraint set given by $\mathcal{C}$.

Let $x^1, \ldots, x^N$ be any points in $\mathbb{R}^m$ and for $\bar{\omega} \neq 0$ define

$$\bar{\omega} = \sum_{l=1}^{N} w_l \quad \text{and} \quad \bar{x} = \frac{1}{\bar{\omega}} \sum_{l=1}^{N} w_l x^l. \tag{6}$$

Consider the unconstrained version of problem (3)

$$\min_{x \in \mathbb{R}^m} \sum_{l=1}^{N} w_l \, e(x^l, x). \tag{7}$$

Theorem 1 *If $\bar{\omega} > 0$ then $x = \bar{x}$ is a solution of (7). Further, if the basis functions $B_1, B_2, \ldots, B_m$ are linearly independent, then x is the unique solution.*

Proof The error function $e(x, y) = \|g(\cdot, x) - g(\cdot, y)\|^2$ where $\|\cdot\|$ is an innerproduct norm. Further, since $g(\xi, w)$ is linear in w and $g(\xi, 0) = 0$ we have for $x \in \mathbb{R}^m$ and $\bar{\omega} \neq 0$

$$\sum_{l=1}^{N} w_l e(x^l, x) = \bar{\omega} e(x, \bar{x}) - \bar{\omega} e(\bar{x}, 0) + \sum_{l=1}^{N} w_l e(x^l, 0).$$

Hence, for $\bar{\omega} > 0$ then $x = \bar{x}$ is a solution of (7).
From the assumption on the error function, $e(x, \bar{x}) = 0$ implies that

$$b(\xi)^{\top}(x - \bar{x}) = 0 \ \ \forall\, \xi \in [a, b].$$

If $B_1(\xi), B_2(\xi), \ldots, B_m(\xi)$ are linearly independent, then $x = \bar{x}$ is the unique solution. □

If $w_l \geq 0, \ \ l = 1, \ldots, N$ and $\bar{\omega} > 0$ then $\bar{x}$ is a convex combination of the points $x^1, \ldots, x^N$. If for $x^l \in \mathscr{C}$ for each l and $\mathscr{C}$ is convex then $\bar{x} \in conv\{x^l, l = 1, \ldots, N\} \subseteq \mathscr{C}$. The next result follows directly from the theorem and shows that if x^l is the solution of the constrained least squares problem (1) and the set $\mathscr{C}$ is convex then $\bar{x}$ is a solution.

Corollary 1 *Let $w \geq 0$, ($w_l \geq 0, l = 1, \ldots N$ and $w \not\equiv 0$) and assume that $\mathscr{C}$ is convex. If $x^l \in \mathscr{C}, \ \ l = 1, \ldots N$ then $\bar{x} \in \mathscr{C}$ and $\bar{x}$ is a solution of (3).*

Let $w \in \mathbb{R}^N, \ \sum_{l=1}^{N} w_l > 0$, and define the function $x : \mathbb{R}^N \to \mathbb{R}^m$

$$x(w) = \frac{\sum_{l=1}^{N} w_l x^l}{\sum_{l=1}^{N} w_l} \ . \tag{8}$$

It follows from the above results that $x(w)$ is a solution of (7) and if $x^l \in \mathscr{C}$, $l = 1, \ldots N$ then $x(w)$ is a solution of (3) provided $\mathscr{C}$ is convex. In the next section we show how to qualitative information can be used to restrict the vector of weights w.

4 Preferences

For simplicity, for a given set of parameters x let the function $g(x)$ denote the function $g(\cdot, x) : \quad [a, b] \to \mathbb{R}$. Consider a pair of realizations of the model $\{g(z^1), g(z^2)\}$ and assume that a decision-maker has a *preference* $g(z^1)$ is better than $g(z^2)$. One realization may be better than the other if it is more physical realistic or in the context of tuning hearing aids, the user feel that one tuning is better than another. We say $z^1 \preceq z^2$ when $g(z^1)$ is better than $g(z^2)$ provided $z^1, \ z^2 \in \mathscr{C}$. Let $v : \mathbb{R}^m \to \mathbb{R}$

$$v(x) = \sum_{l=1}^{N} w_l e(x^l, x) \tag{9}$$

for some $w_l, l = 1, \ldots, N$ to be determined and $e(x, y)$ is a suitable error function. The function (9) is called a value function and v is said to be *consistent* with the ranking if

$$z^1 \preceq z^2 \Rightarrow v(z^1) \leq v(z^2).$$

In the general case we consider a set of M outcomes or realizations $Y = \{z^1, \ldots, z^M\} \subseteq \mathscr{C}$. The set Y has a *partial order* $\mathscr{L} \subseteq Y \times Y$ consisting of paired comparisons, so that for $(x, z) \in \mathscr{L}$ then $x \preceq z$. The function $v : Y \to \mathbb{R}$ is consistent with the partial order $\mathscr{L}$ if for all $(x, z) \in \mathscr{L}$ then $v(x) \leq v(z)$. Then for any $x, z \in Y$ the values $v(x)$ and $v(z)$ can be compared and we deduce that x is better than z if $v(x) \leq v(z)$ even for those pairs not in $\mathscr{L}$.

Dennis and Woods [1, 2] observed for $(x, z) \in \mathscr{L}$ then

$$x \preceq z \Rightarrow v(x) \leq v(z) \quad \Leftrightarrow \quad a(x, z)^\top w \leq 0 \tag{10}$$

where component l in the vector $a(x, z)$ is

$$a(x, z)_l = e(x^l, x) - e(x^l, z).$$

Hence, for every pair $(x, z) \in \mathscr{L}$ we get a linear inequality (10) in the unknown weight $w^\top = (w_1, \ldots, w_N)$.

When $\mathscr{L}$ consist of several comparisons, we get several inequalities which defines the matrix inequality

$$Aw \leq 0,$$

where each row in A is $a(x, z)^\top$ for $(x, y) \in \mathscr{L}$. The matrix A depends on the model, the data and the elements in $\mathscr{L}$.

Note that if the weight w is scaled by a positive constant, then the value function will be scaled with the same constant and the scaled value function is consistent with $\mathscr{L}$ if the unscaled value function is consistent with $\mathscr{L}$. Further, the minimizer is independent of this scale. Hence we will introduce the normalization $e^\top w = 1$ where $e^\top = (1, 1, \ldots, 1)$. The set of possible w so that the value function is consistent with $\mathscr{L}$ is

$$\Omega = \{w \in \mathbb{R}^N \mid Aw \leq 0,\ e^\top w = 1\}. \tag{11}$$

We follow [1, 2] and say that any $w \in \Omega$ is feasible.

For each $w \in \Omega$ the solution $x(w)$ of (7) is defined and the model is $g(\xi, x(w))$. Hence, the set of possible models given the set Ω can be displayed and preferences that reduce the feasible set will give a more narrow 'band' of possible models. In Fig. 3 the feasible set Ω is shown shaded and the corresponding set of curves $g(\cdot, x(w))$ for $w \in \Omega$ are shown shaded in Fig. 4 using the data in the example with the choice that the solid curve is better than the stippled curve in Fig. 2.

5 Shape Preserving Spline Functions

This section summarizes some properties of shape preservation of univariate spline functions [5] that are needed for the numerical experiment in the next section.

Let $\{t_j\}_{j\geq 1}$ be a sequence of numbers in $[a, b] \subset \mathbb{R}$ such that $t_j \leq t_{j+1}$ for all j. The numbers t_j are called knots. A sequence of B-splines $\{B_{j,k}\}_{j,k\geq 1}$ is defined by

$$B_{j,1}(\xi) = \begin{cases} 1, & t_j \le \xi < t_{j+1}, \\ 0, & \text{otherwise}, \end{cases}$$

and the higher degree B-splines by the following recurrence relation

$$B_{j,k}(\xi) = \omega_{j,k}(\xi) B_{j,k-1}(\xi) + (1 - \omega_{j+1,k}(\xi)) B_{j+1,k-1}(\xi),$$

where

$$\omega_{j,k}(\xi) = \begin{cases} (\xi - t_j)/(t_{j+k-1} - t_j), & t_j \le \xi < t_{j+k-1}, \\ 0, & \text{otherwise}. \end{cases}$$

It follows from the definition that each B-spline is a polynomial of degree $k-1$, $B_{j,k}(\xi)$ is determined by the $k+1$ knots $t_j, t_{j+1}, \ldots, t_{j+k-1}, t_{j+k}$, and a B-spline is local, i.e. $B_{j,k}(\xi) = 0$ for $\xi < t_j$ or $\xi > t_{j+k}$.

For a given k and $m \ge k$, consider the set of knots $\{t_j \mid j = 1, \ldots, m+k\}$ called the knot-vector and the m B-splines $B_{j,k}$, $j = 1 \ldots, m$. A *spline function* of order k with the knot vector t is any linear combination of B-splines of order k. The smoothness of the spline function depends on the knot-vector. At every point ξ the sum of the number of knots at the point and the number of continuity conditions at the point is equal to k.

A knot-vector is called k-regular if $m \ge k$, $t_j < t_{j+k}$, $j = 1, 2, \ldots, m$, $t_1 = \cdots = t_k < t_{k+1}$ and $t_m < t_{m+1} = \cdots = t_{m+k}$. Given a k-regular knot-vector, let $a = t_k$, $b = t_{m+1}$. The model (4) will be the spline function $g(\xi, x) = \sum_{i=1}^{m} x_i B_{i,k}(\xi)$ on $[a, b]$.

Lemma 1 *A spline function with non-negative B-spline coefficients x_i, $i = 1, \ldots, m$ is non-negative and if $x_i \ge x_{i-1}$, $i = 2, \ldots, m$ then g is nondecreasing on $[a, b]$. Further, if the knots t_j, $j = k+1, \ldots, m$ have multiplicity at most $k-1$ then g is convex on $[a, b]$ if*

$$\frac{x_i - x_{i-1}}{t_{i+k-1} - t_i} \ge \frac{x_{i-1} - x_{i-2}}{t_{i+k-2} - t_{i-1}}, \qquad i = 3, \ldots, m.$$

It should be pointed out that a spline function can be non-negative even with some coefficients being negative. Similarly, the conditions for non-decreasing and convexity are not necessary conditions.

6 Numerical Example

Normalized relative permeability curves are twice continuously differentiable and non-decreasing functions $g : [0, 1] \to [0, 1]$ so that $g(0) = 0$ and $g(1) = 1$. In addition a second order condition is imposed. This can be convexity or that the function should have an s-form (the second derivative should change sign). In this example we require that the function is convex on the interval.

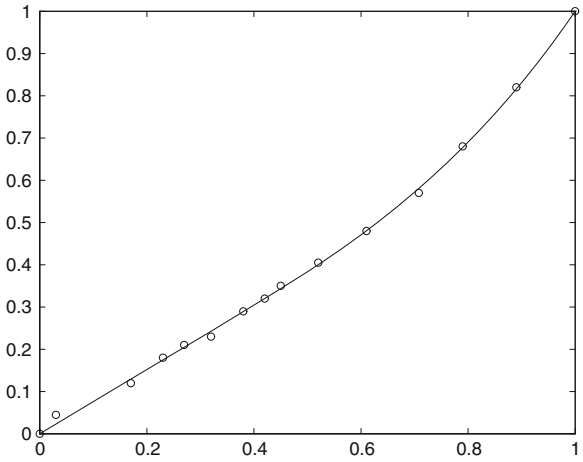

Fig. 1 Data Set 1 marked ∘ with corresponding curve (1)

Table 1 Normalized relative permeability data (ξ_i, v_i), $i = 1, \ldots, N_l$

Set 1	ξ	0.030	0.170	0.230	0.270	0.320	0.380	0.420	0.450	0.520	0.610	0.708	0.790	0.890
	v	0.045	0.120	0.180	0.210	0.230	0.290	0.320	0.350	0.405	0.480	0.570	0.680	0.820
Set 2	ξ	0.400	0.636	0.703	0.782	0.850	0.876	0.950						
	v	0.133	0.370	0.470	0.601	0.720	0.820	0.930						
Set 3	ξ	0.480	0.790	0.870	0.900	0.928								
	v	0.090	0.410	0.550	0.660	0.740								

Let the basis functions be B-splines so that any spline function is two times continuously differentiable. In the following let the knot-vector be (0, 0, 0, 0, 1/3, 2/3, 1, 1, 1, 1) which is 4-regular with interior knots with multiplicity 1. Consider the B-splines $B_{i,4}$, $i = 1, 2, \ldots, 6$. Then the spline function will be normalized if $x_1 = 0$ and $x_6 = 1$, non-negative if $x_i \geq 0$ for $i = 1, 2, \ldots, 6$, non-decreasing if $x_{i+1} - x_i \geq 0$ for $i = 1, 2, \ldots, 5$, and convex if

$$2x_1 - 3x_2 + x_3 \geq 0, \quad 3x_2 - 5x_3 + 2x_4 \geq 0,$$
$$2x_3 - 5x_4 + 3x_5 \geq 0, \quad x_4 - 3x_5 + 2x_6 \geq 0.$$

Let the set $\mathscr{C}$ be the set of $x \in \mathbb{R}^6$ that satisfies the above equalities and inequalities. The set $\mathcal{C}$ is a polyhedron and the problem (5) can be solved with any standard quadratic programming solver. The corresponding function $g(\cdot, x)$ will be non-negative, non-decreasing and convex and solve the least squares problem (1).

Figure 1 shows the curve generated from the first data set in the Table 1 using $W_i \equiv 1$ and the constraints above.

In the data set there are three different sets of data from a laboratory experiment (see page 184 in [6]) and the purpose is to find an average normalized relative

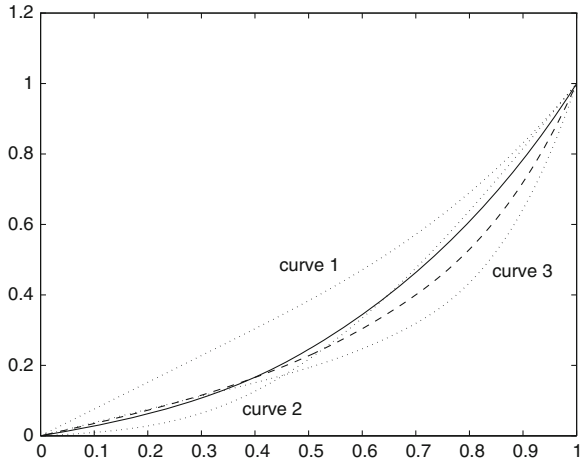

Fig. 2 Spline functions from the data sets, and two realizations

permeability to be used in a reservoir simulator (see [7]). The model to be used is a spline function using the knot vector in the example. The spline function is required to be normalized, non-decreasing and convex. Figure 2 shows the least squares solution for each data set given in Table 1 where curve i is computed using Set i. In addition two curves are computed using the parameters $x(w)$ defined by (8) with $w^\top = (1/5, 3/5, 1/5)$ and $w^\top = (1/7, 2/7, 4/7)$ and these are shown solid and stippled respectively.

Let $W \equiv 1$ in the error function (2). A user has the preference that the solid curve is better than the stippled curve in Fig. 2. Equation (10) gives the inequality $-8.95w_1 - 11.1w_2 + 10.5w_3 \leq 0$. In addition, the weights are supposed to be non-negative and normalized. The possible weights that can define an average of the three given curves and satisfies the preference are given by

$$-8.95w_1 - 11.1w_2 + 10.5w_3 \leq 0,\ \ w_1 + w_2 + w_3 = 1, \text{and} w_i \geq 0\ i = 1, 2, 3.$$

This region is shown shaded in Fig. 3. For each w the set of parameters $x(w)$ is computed from (8).

Each feasible extreme point w_e in the polyhedron is marked with a letter and the corresponding curve $g(\cdot, x(w_e))$ is shown as a solid line in Fig. 4. Any average curve must be in the shaded region in Fig. 4. Since $w = (1, 0, 0)$ and $w = (0, 1, 0)$ are feasible the curves $g(\cdot, x((1, 0, 0)))$ and $g(\cdot, x((0, 1, 0)))$ will be in the shaded region (marked `curve 1` and `curve 2`) while $w = (0, 0, 1)$ is not feasible so $g(\cdot, x((0, 0, 1)))$ (marked `curve 3`) is not completely in the shaded region.

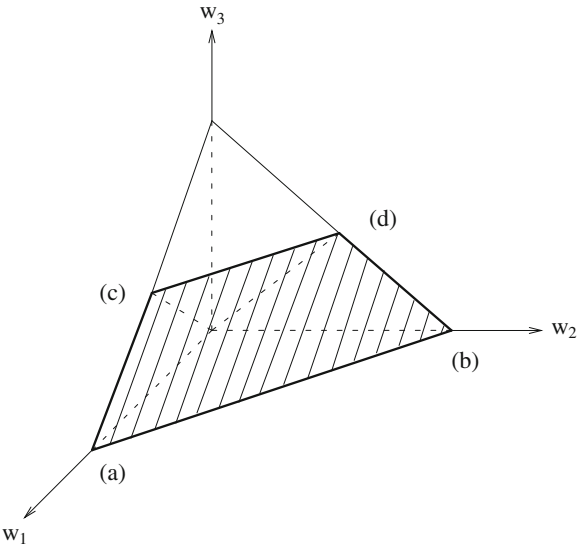

Fig. 3 Set of feasible w

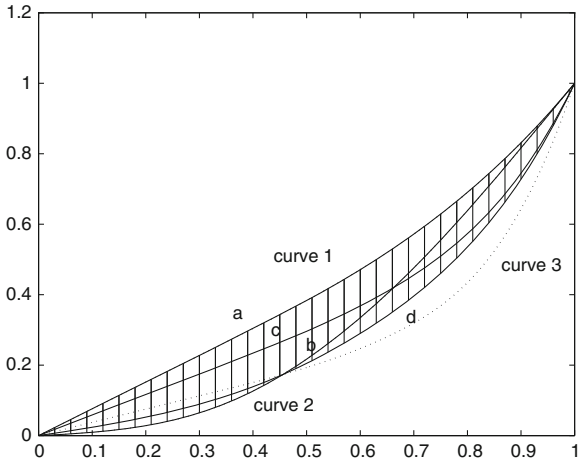

Fig. 4 Possible curves that can be an average are in the shaded region

References

1. Dennis JE, Woods DJ (1987) Interactive graphics for curve-tailoring. In: New computing environments: microcomputers in large-scale computing, SIAM. Philadelphia, pp 123–129
2. Dennis John E, Woods Daniel J (1988) Curve tailoring with interactive computer graphics. Appl Math Lett 1:41–44

3. Nordeide LM, Steihaug T (1994) To rank a Miss without to miss a rank. In: Norsk informatikkkonferanse NIK'94, Norway, 1994. Tapir ISBN 82-519-1428-0, pp 207-218
4. Alexander Y, Job G, Serkan Ö, Erik VDW (2008) Online personalization of hearing instruments. EURASIP Journal on Audio, Speech, and Music Processing, 2008
5. Kuijt Frans, van Damme Ruudt (2001) A linear approach to shape preserving spline approximation. Adv Comput Math 14:25–48
6. Amyx JW Jr, Bass DM (1960) Petroleum reservoir engineering. McGraw-Hill, New York 1960
7. Aziz K, Settari A (1979) Petroleum reservoir simulation. Applied Science Publishers, England

Regularizing Soft Decision Trees

Olcay Taner Yıldız and Ethem Alpaydın

Abstract Recently, we have proposed a new decision tree family called soft decision trees where a node chooses both its left and right children with different probabilities as given by a gating function, different from a hard decision node which chooses one of the two. In this paper, we extend the original algorithm by introducing local dimension reduction via L_1 and L_2 regularization for feature selection and smoother fitting. We compare our novel approach with the standard decision tree algorithms over 27 classification data sets. We see that both regularized versions have similar generalization ability with less complexity in terms of number of nodes, where L_2 seems to work slightly better than L_1.

1 Introduction

A decision tree is an hierarchical structure made up of internal decision nodes and terminal leaves. For classification, the leaves carry the label of one of K classes. The input vector is composed of d attributes, $x = [x_1, \ldots, x_d]^T$. Each decision node m implements function $v_m(x)$ and chooses one of the children accordingly. Let $F_m(x)$ be the output generated by the subtree whose root is m and in a binary tree, let $F_m^L(x)$ and $F_m^R(x)$ denote respectively its left and right children and $v_m(x)$ hence has two outcomes:

$$F_m(x) = \begin{cases} F_m^L(x) & \text{if} \quad v_m(x) > 0 \ /^*\text{true}^*/ \\ F_m^R(x) & \text{otherwise} \quad /^*\text{false}^*/ \end{cases} \tag{1}$$

O. T. Yıldız (✉)
Department of Computer Engineering, Işık University, 34980 Istanbul, Turkey
e-mail: olcaytaner@isikun.edu.tr

E. Alpaydın
Department of Computer Engineering,Boğaziçi University,34342 Istanbul, Turkey
e-mail: alpaydin@boun.edu.tr

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_2,

Given an input to classify, starting from the root node, one applies the function at each internal node and the input is forwarded to one of the two branches depending on the outcome. This process is repeated recursively until a leaf node is hit at which point the class label of the leaf constitutes the output. Depending on the model they assume for $F_m(x)$, decision trees are subcategorized into univariate decision trees [1], multivariate linear decision trees [2], multivariate nonlinear decision trees [3], and omnivariate decision trees [4].

In our recent work [5], we generalized decision trees and proposed soft decision trees. A node at a hard decision tree forwards the input to either its left or the right subtree, whereas a soft decision tree node utilizes a gating function to assign probabilities to its children and merges the decision of its children by these probabilities. That is, we follow all the paths to all the leaves and all the leaves contribute to the final decision but with different probabilities.

In this paper, we extend soft decision trees by adding a regularization term (linear for L_1 regularization, quadratic for L_2 regularization) to handle a localized dimensionality reduction in the nodes, since as we go deeper into the tree, the scope of a node becomes more localized. This paper is organized as follows: In Sect. 2, we briefly review the original soft tree algorithm. We give the details of our regularized version of the original approach in Sect. 3. We give our experimental results in Sect. 4 and conclude in Sect. 5.

2 Soft Decision Trees

As opposed to the (hard) decision mode which redirects instances to its left or right subtree depending on the node function $F_m(x)$, soft decision node redirects instances both to the left and right subtree with probabilities calculated by the gating function $v_m(x)$:

$$F_m(x) = F_m^L(x)v_m(x) + F_m^R(x)(1 - v_m(x)) \tag{2}$$

and to choose among two children, we take $v_m(x) \in [0, 1]$ as the *sigmoid function*:

$$v_m(x) = \frac{1}{1 + \exp[-(w_m^T x + w_{m0})]} \tag{3}$$

Learning the tree is incremental and recursive, as with the hard decision tree. The algorithm starts with one node and fits a constant model. Then, as long as there is improvement, it replaces the leaf by a subtree. This involves optimizing the gating parameters and the values of its children leaf nodes by gradient-descent over an error function. The error function is cross-entropy for classification, and the final output of the tree is filtered through a sigmoid at the root to convert it to a probability:

$$E = r \log y + (1 - r) \log(1 - y)$$

3 Regularized Soft Decision Trees

In general, model selection problem in decision trees have two faces. On the one hand, keeping the node model fixed, one can delve into the optimization of the tree structure and use either pre or post-pruning techniques. On the other hand, one can try to solve the model selection problem at the node level and select one among L candidate models based on both complexity and performance [4].

Our approach in this paper falls into the second category and we use L_1 and L_2-norm regularization techniques to combine the model complexity and error term into one single value. The function we want to minimize is

$$E_{L_1} = \frac{1-\lambda}{N} \sum_t (r^{(t)} \log y^{(t)} + (1 - r^{(t)}) \log(1 - y^{(t)})) + \frac{\lambda}{d} \sum_{i=0}^{d} |w_{mi}|$$

for L_1 regularization, and

$$E_{L_2} = \frac{1-\lambda}{N} \sum_t (r^{(t)} \log y^{(t)} + (1 - r^{(t)}) \log(1 - y^{(t)})) + \frac{\lambda}{d} \sum_{i=0}^{d} w_{mi}^2$$

for L_2 regularization, where $1 - \lambda$ and λ correspond to the weight factor of cross-entropy and model complexity respectively. From a Bayesian perspective, L_1 and L_2 regularization corresponds to Laplacian and Gaussian priors on w_{mi} and the equations above correspond to maximum a posteriori estimates.

In gradient-descent, we use the following update equations:

$$\alpha_m = \prod_{p=m.parent}^{p!=root} \delta_{p,p.parent.left} v_p(x) + \delta_{p,p.parent.right}(1 - v_p(x))$$

$$\frac{\partial E_{L_1}}{\partial w_{mi}} = \frac{1-\lambda}{N}(r - y)(F_m^L(x) - F_m^R(x))\alpha_m v_m(x)(1 - v_m(x))x_i + \frac{sgn(w_{mi})\lambda}{d} \quad (L_1)$$

$$\frac{\partial E_{L_2}}{\partial w_{mi}} = \frac{1-\lambda}{N}(r - y)(F_m^L(x) - F_m^R(x))\alpha_m v_m(x)(1 - v_m(x))x_i + \frac{\lambda w_{mi}}{d} \quad (L_2)$$

where $\delta_{x,y}$ is the Kronecker delta and sgn(x) is the sign function.

4 Experiments

To compare the generalization error and model complexity of our regularized soft trees with both soft trees (without regularization) and hard C4.5 trees, we use 27 two-class data sets from UCI repository [6]. We also compare with a multivariate

Table 1 On the classification data sets, the average error of soft, hard, linear discriminant trees (Ldt). Pairwise comparisons of error rate of soft and hard decision tree algorithms are shown in the second table

Dataset	Hard	Ldt	Soft	Soft(L_1)	Soft(L_2)
Acceptors	16.1 ± 2.0	9.6 ± 0.8	8.7 ± 0.7	7.5 ± 0.3	7.3 ± 0.3
Artificial	1.1 ± 1.8	1.5 ± 1.9	1.1 ± 1.8	0.7 ± 1.6	0.4 ± 1.2
Breast	6.7 ± 1.1	4.9 ± 0.6	3.5 ± 0.7	4.1 ± 0.8	3.5 ± 0.7
Bupa	38.6 ± 4.1	39.1 ± 3.4	39.7 ± 4.2	41.4 ± 3.0	39.0 ± 2.4
Donors	7.7 ± 0.4	5.4 ± 0.3	5.7 ± 0.4	5.6 ± 0.3	5.5 ± 0.3
German	29.9 ± 0.0	25.8 ± 2.0	24.0 ± 3.0	25.9 ± 2.4	24.3 ± 1.4
Haberman	26.6 ± 0.3	27.2 ± 1.5	25.9 ± 1.8	25.0 ± 2.5	24.7 ± 2.4
Heart	28.3 ± 4.7	18.4 ± 2.3	19.7 ± 3.4	18.1 ± 2.5	18.3 ± 2.5
Hepatitis	22.1 ± 4.4	20.4 ± 2.9	20.2 ± 2.4	20.4 ± 4.2	19.0 ± 3.6
Ionosphere	13.1 ± 1.9	12.3 ± 2.2	11.5 ± 2.0	11.5 ± 1.9	12.6 ± 1.1
Krvskp	1.2 ± 0.4	4.5 ± 0.7	1.8 ± 0.6	3.6 ± 0.6	3.0 ± 0.7
Magic	17.5 ± 0.6	16.9 ± 0.1	14.7 ± 0.5	21.6 ± 0.2	20.8 ± 0.2
Monks	12.8 ± 7.8	23.8 ± 8.2	0.0 ± 0.0	3.4 ± 5.3	3.5 ± 7.4
Mushroom	0.0 ± 0.1	1.8 ± 0.5	0.1 ± 0.0	0.1 ± 0.0	0.0 ± 0.0
Musk2	5.5 ± 0.6	6.4 ± 0.3	4.3 ± 0.7	6.3 ± 0.7	5.7 ± 0.7
Parkinsons	13.8 ± 2.3	13.5 ± 2.5	14.3 ± 2.7	13.7 ± 2.7	12.3 ± 2.4
Pima	27.9 ± 3.4	23.1 ± 1.4	24.9 ± 2.0	23.1 ± 0.8	23.1 ± 1.0
Polyadenylation	30.5 ± 1.3	22.6 ± 0.6	22.9 ± 0.5	22.2 ± 0.5	22.3 ± 0.5
Promoters	26.1 ± 9.9	34.4 ± 9.4	15.3 ± 6.7	13.1 ± 7.6	16.7 ± 9.3
Ringnorm	12.2 ± 1.1	22.8 ± 0.3	9.9 ± 1.7	22.4 ± 0.3	22.4 ± 0.4
Satellite47	15.4 ± 1.5	16.7 ± 1.4	12.4 ± 1.4	16.4 ± 1.4	15.4 ± 0.9
Spambase	9.9 ± 0.7	10.1 ± 0.7	7.5 ± 0.5	8.2 ± 0.3	7.6 ± 0.3
Spect	19.1 ± 2.8	20.1 ± 2.4	19.6 ± 2.4	16.9 ± 2.6	16.6 ± 1.8
Tictactoe	23.8 ± 2.2	31.9 ± 2.4	1.8 ± 0.3	1.8 ± 0.4	1.7 ± 0.3
Titanic	21.8 ± 0.5	22.4 ± 0.4	21.5 ± 0.2	21.6 ± 0.3	22.1 ± 0.2
Twonorm	17.0 ± 0.7	2.0 ± 0.1	2.1 ± 0.2	2.1 ± 0.2	2.1 ± 0.2
Vote	5.2 ± 0.7	6.7 ± 2.6	5.1 ± 0.9	6.7 ± 0.9	5.6 ± 1.1
	Hard	Ldt	Soft	Soft(L_1)	Soft(L_2)
Hard		6	0	5	4
Ldt	5		0	1	1
Soft	9	11		5	4
Soft(L_1)	7	4	1		0
Soft(L_2)	8	4	1	2	

linear tree algorithm (Ldt) [7]. We first separate one third of the data set as the test set over which we evaluate the final performance. On the remaining two thirds, we apply 5×2-fold cross validation, which gives a total of 10 folds for each data set. We use parametric 5×2 paired F-test [8] and nonparametric Nemenyi's test to compare algorithms.

Table 1 shows the average and standard deviation of errors of soft and hard decision trees. The table below it shows the pairwise comparison results of error rates—entry

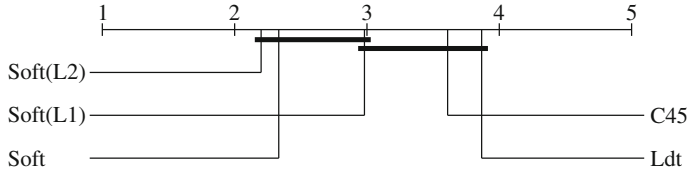

Fig. 1 The result of post-hoc Nemenyi's test applied on the error rates of soft, hard, linear discriminant trees (*Ldt*)

(i, j) in this second table gives the number of datasets (out of 27) on which method i is statistically significantly better than method j with at least 95 % confidence using 5×2 paired F-test. Figure 1 shows the result of post-hoc Nemenyi's test applied on the error rates of the algorithms.

We see that the soft tree and its extensions are significantly more accurate than both hard and linear discriminant trees. Soft tree variants are better than hard trees and Ldt on eight and six datasets respectively; hard trees are only better on three datasets and Ldt are better only on one dataset. L_2 regularization is significantly better than L_1 on two datasets. If we compare nonparametrically, on 18 datasets out of 27, L_2 normalization has smaller error rate, whereas L_1 has smaller error rate only on 9 datasets. Overall, L_2 regularization is slightly better than L_1 in terms of error rate (p-value = 0.06).

Table 2 shows the average and standard deviation of number of nodes of soft and hard decision trees. Again the table below shows the pairwise comparison results in terms of number of nodes. Figure 2 shows the result of post-hoc Nemenyi's test applied on the number of nodes of tree generated by the algorithms.

As expected, soft tree variants are simpler than hard trees. On the average, soft tree variants are significantly smaller than hard trees on 12 datasets, whereas hard trees are smaller only on one dataset. The results also show that regularization pays off. Regularized soft trees are simpler than the original soft trees on three datasets. L_1 regularization is significantly better than L_2 on one dataset according to 5×2 paired F-test. If we compare nonparametrically, on 24 datasets out of 27, L_2 regularization generates smaller trees than L_1 whereas the opposite is true on only one dataset. We can conclude that L_2 regularization leads to significantly smaller trees than L_1 (p-value $< 10^{-6}$).

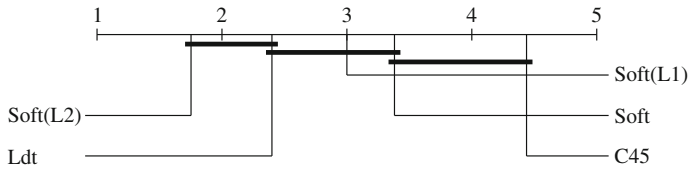

Fig. 2 The result of post-hoc Nemenyi's test applied on the number of nodes of trees generated by soft, hard, linear discriminant trees (*Ldt*)

Table 2 On the classification data sets, the average number of decision nodes of soft, hard, linear discriminant trees (Ldt). Pairwise comparisons are shown in the second table

Dataset	Hard	Ldt	Soft	Soft(L_1)	Soft(L_2)
Acceptors	7.1 ± 6.9	1.1 ± 0.3	7.0 ± 3.9	4.0 ± 1.4	2.2 ± 0.8
Artificial	4.4 ± 1.0	2.0 ± 0.5	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Breast	4.1 ± 2.2	1.6 ± 0.7	1.3 ± 0.5	1.7 ± 0.7	1.3 ± 0.5
Bupa	5.4 ± 3.7	1.7 ± 0.7	4.0 ± 2.3	3.5 ± 1.2	2.4 ± 0.8
Donors	21.0 ± 3.7	3.7 ± 1.9	6.4 ± 3.2	3.2 ± 1.0	1.8 ± 0.9
German	0.0 ± 0.0	3.1 ± 3.5	4.1 ± 2.0	4.1 ± 1.8	1.7 ± 0.5
Haberman	1.0 ± 3.2	1.2 ± 1.8	2.0 ± 1.6	2.5 ± 1.1	1.2 ± 1.0
Heart	3.5 ± 2.7	1.0 ± 0.0	1.6 ± 0.8	1.7 ± 0.8	1.2 ± 0.4
Hepatitis	0.7 ± 0.9	0.9 ± 0.6	2.0 ± 0.8	1.8 ± 0.9	1.2 ± 0.4
Ionosphere	3.8 ± 1.9	2.2 ± 0.8	2.5 ± 1.4	2.1 ± 1.4	1.5 ± 0.8
Krvskp	23.7 ± 4.2	6.2 ± 2.9	6.8 ± 2.5	3.5 ± 1.2	2.7 ± 1.5
Magic	30.1 ± 16.5	19.4 ± 8.1	26.4 ± 6.1	5.3 ± 1.4	2.5 ± 0.7
Monks	11.2 ± 2.4	3.5 ± 3.3	3.0 ± 0.0	4.8 ± 1.8	3.5 ± 1.4
Mushroom	4.9 ± 0.3	10.9 ± 2.5	1.0 ± 0.0	1.3 ± 0.7	1.0 ± 0.0
Musk2	28.5 ± 7.0	6.8 ± 3.8	16.5 ± 4.0	7.4 ± 4.0	4.7 ± 3.0
Parkinsons	3.3 ± 1.7	1.2 ± 0.4	3.6 ± 1.4	1.8 ± 0.8	1.4 ± 0.5
Polyadenylation	22.5 ± 18.4	2.2 ± 2.4	8.9 ± 3.6	5.5 ± 2.7	3.1 ± 2.5
Pima	3.8 ± 2.6	2.2 ± 1.2	3.4 ± 2.4	2.4 ± 1.0	1.7 ± 0.8
Promoters	2.0 ± 1.3	0.8 ± 0.4	1.4 ± 0.5	1.9 ± 0.7	1.5 ± 0.5
Ringnorm	45.7 ± 5.2	1.3 ± 0.5	39.9 ± 7.2	2.8 ± 0.9	2.5 ± 1.2
Satellite47	11.9 ± 4.5	4.0 ± 1.9	12.7 ± 4.3	5.6 ± 3.1	3.6 ± 2.3
Spambase	20.3 ± 8.4	6.0 ± 2.7	5.5 ± 2.1	4.9 ± 2.0	3.1 ± 1.4
Spect	5.2 ± 5.8	1.1 ± 1.6	1.4 ± 1.3	1.8 ± 0.6	1.3 ± 0.5
Tictactoe	22.2 ± 6.8	6.2 ± 3.4	1.3 ± 0.5	1.2 ± 0.4	1.1 ± 0.3
Titanic	4.2 ± 0.6	1.7 ± 0.5	1.1 ± 0.3	1.3 ± 0.5	1.3 ± 0.5
Twonorm	79.9 ± 8.0	1.1 ± 0.3	3.1 ± 1.2	1.9 ± 0.6	2.2 ± 1.1
Vote	2.9 ± 1.7	1.8 ± 0.9	1.7 ± 0.7	1.6 ± 0.5	1.4 ± 0.5
	Hard	Ldt	Soft	Soft(L_1)	Soft(L_2)
Hard		0	1	1	1
Ldt	11		4	2	0
Soft	10	2		1	0
Soft(L_1)	12	3	2		0
Soft(L_2)	13	4	4	1	

5 Conclusions

We extend the soft decision tree model by adding L_1 and L_2 regularization to penalize unnecessary complexity. The extended model is evaluated on 27 classification data sets. We see that both versions improve accuracy slightly and decrease complexity significantly; overall L_2 regularization seems to work slightly better than L_1.

References

1. Quinlan JR (1993) C4.5: programs for machine learning. Morgan Kaufmann, San Meteo, CA
2. Murthy SK, Kasif S, Salzberg S (1994) A system for induction of oblique decision trees. J Artif Intell Res 2:1–32
3. Guo H, Gelfand SB (1992) Classification trees with neural network feature extraction. IEEE Trans Neural Netw 3:923–933
4. Yıldız OT, Alpaydın E (2001) Omnivariate decision trees. IEEE Trans Neural Netw 12(6):1539–1546
5. Irsoy O, Yildiz OT, Alpaydin E (2012) Soft decision trees. In: Proceedings of the international conference on pattern recognition, Tsukuba, Japan, pp 1819–1822
6. Blake C, Merz C (2000) UCI repository of machine learning databases
7. Yıldız OT, Alpaydın E (2005) Linear discriminant trees. Int J Pattern Recogn Artif Intell 19(3):323–353
8. Alpaydın E (1999) Combined 5×2 cv F test for comparing supervised classification learning classifiers. Neural Comput 11:1975–1982

A Simple Yet Fast Algorithm for the Closest-Pair Problem Using Sorted Projections on Multi-Dimensions

Mehmet E. Dalkılıç and Serkan Ergun

Abstract We present a simple greedy algorithm, *QuickCP*, for finding the closest-pair of points on a plane. It is based on the observation that if two points are close to each other, then it is very likely that their sorted projections to x-axis and/or to y-axis will reflect that closeness. Therefore we order our search starting from the pairs with closest x-projections (and closest y-projections) to find the closest pair quickly and make a fast exit. Empirical data up to 2^{26} (over 60 million points) indicates that this approach quickly detects the closest pair and it is, on average 70 % faster than the classical divide-and-conquer algorithm registering, to the best of our knowledge, the fastest recorded solution times in the literature for the planar closest-pair problem. A second contribution of our work is that *QuickCP* can be used as part of the divide-and-conquer algorithms to handle small sub-problems. Measurements on data up to 2^{26} points show that this co-operation speeds up the basic divide-and-conquer algorithm on average 50 %.

1 Introduction

We address one of the most basic problems in Computational Geometry. Given n points on a plane, find the two points closest to each other. Closest-point problems are fundamental to the Computational Geometry and has applications in graphics, computer vision, geographical information systems, and molecular modeling [6, 8, 9].

It is commonly agreed that the best (most practical) way of solving the above stated two-dimensional closest-pair of points problem is to employ the $O(n \log n)$ divide-and-conquer algorithm introduced by Bentley and Shamos [1]. Although there

M. E. Dalkılıç (✉) · S. Ergun
International Computer Institute, Ege University, Izmir, Turkey
e-mail: mehmet.emin.dalkilic@ege.edu.tr

S. Ergun
e-mail: serkan.ergun@ege.edu.tr

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_3,

exist $O(n)$ randomized algorithms [2, 5] for the closest points problem, due to their high implementation complexity, these algorithms are difficult to implement and therefore, the divide and conquer algorithm is often the practical choice for many applications.

Using the sorted projections of the input points in x- and y-dimension, a simpler and faster (than the divide and conquer) algorithm can be developed. Even if the algorithm has a worst-case time complexity of $O(n^2)$, with a much better average time performance it can be practical and useful. Moreover, any algorithm considerably faster than the divide-and-conquer closest pair (DivConCP) algorithm for the small problem sizes, can be used to speed up the DivConCP algorithm by quickly solving the small sub-problems for it.

2 Proposed Algorithm: QuickCP

A typical textbook implementation of the DivConCP algorithm (e.g., [6]) begins with two arrays A and B: in the first array all points are sorted by increasing x-coordinate and in the second array by increasing y-coordinate. Arrays A and B, in fact, contain, respectively, the sorted x-projections and y-projections of the input points. Our observation that these sorted projection arrays can be used in an entirely different fashion than that of the DivConCP algorithm, has led us to the *QuickCP* algorithm given below. Our main assertion is that if two points are close to each other, then it is very likely that their sorted projections to the x-axis and/or to the y-axis will reflect that closeness. Therefore we can walk through the x-sorted and y-sorted lists in rounds $(1, 2, \ldots, n-1)$ pairing each point with its r-closest neighbor in both dimensions in the rth round. Empirical data obtained over many runs of the *QuickCP* algorithm on uniformly distributed random data from 2^4 up to 2^{26} (over 60 million) points suggest that *QuickCP* finds the closest pair and terminates (after guaranteeing that no closer pairs exist) within $O(\log n)$ rounds.

2.1 Algorithm Description

QuickCP algorithm is given in Algorithm 1. It takes a set of two-dimensional points, P, and creates two auxiliary arrays A and B holding the x-sorted and y-sorted points. The algorithm runs in a loop indexed by r. In the first round ($r = 1$), distances for 1-closest-x pairs and 1-closest-y pairs are computed recording the minimum distance seen so far. And in the same round, we compute $\Delta X_{\min}$ which represents the minimum x distance between any 1-closest-x pairs. Similarly we have $\Delta Y_{\min}$ representing the minimum y distance between any 1-closest-y pairs. It is not difficult to convince oneself that at this or any later round no two points can be closer to each other than that of $\sqrt{(\Delta X_{\min})^2 + (\Delta Y_{\min})^2}$. These minimum distances $(\Delta X_{\min}, \Delta Y_{\min})$ together

Algorithm 1 QuickCP(P)

```
// P = {P_1, P_2, ..., P_n} where P_i = (x_i, y_i) i.e., n input points
A ← sortByX(P)   // Sort P by x-values, store in A i.e., {A(1), A(2), ..., A(n)}
B ← sortByY(P)   // Sort P by y-values, store in B i.e., {B(1), B(2), ..., B(n)}
d_min ← ∞
for r = 1 to n − 1 do   // at most n − 1 rounds
    ΔX_min, ΔY_min ← ∞
    for i = 1 to n − r do
        ΔX ← A(i + r).x − A(i).x   // A(j).x is the x-value of A(j)
        ΔY ← A(i + r).y − A(i).y   // A(j).y is the y-value of A(j)
        ΔX_min ← min(ΔX_min, ΔX)   // update ΔX_min
        d_A ← sqrt(ΔX^2 + ΔY^2)   // distance between points A(i) and A(i+r)

        ΔX ← B(i + r).x − B(i).x   // B(j).x is the x-value of B(j)
        ΔY ← B(i + r).y − B(i).y   // B(j).y is the y-value of B(j)
        ΔY_min ← min(ΔY_min, ΔY)   // update ΔY_min
        d_B ← sqrt(ΔX^2 + ΔY^2)   // distance between points B(i) and B(i+r)

        d_min ← min(d_min, d_A, d_B)   // update d_min
    end for
    if d_min < sqrt((ΔX_min)^2 + (ΔY_min)^2) then
        break   // Termination Condition
    end if
end for
```

with the closest pair distance so far ($d_{\min}$) gives us the opportunity to specify the termination condition.

At round r each point in the input is first paired with its r-closest-x neighbor and then with its r-closest-y neighbor. If we take the minimum of the ΔX values over all pairs in round r, any pair in any later round will produce ΔX values greater than or equal to that of round r. The same is valid for the ΔY values. In other words if we define $\Delta X^{\mathrm{i}}_{\min}$ as the minimum ΔX value at round i and $\Delta Y^{\mathrm{i}}_{\min}$ as the minimum ΔY value at round i then

$$\begin{aligned} \Delta X^{1}_{\min} \leq \Delta X^{2}_{\min} \leq \cdots \leq \Delta X^{\mathrm{n\text{-}1}}_{\min} \\ \Delta Y^{1}_{\min} \leq \Delta Y^{2}_{\min} \leq \cdots \leq \Delta Y^{\mathrm{n\text{-}1}}_{\min} \end{aligned} \tag{1}$$

Furthermore, if we have $\Delta X^{\mathrm{r}}_{\min}$ and $\Delta Y^{\mathrm{r}}_{\min}$ at round r, in later rounds it is guaranteed that all pairs will produce distances at least as big as $\sqrt{(\Delta X^{r}_{\min})^2 + (\Delta Y^{r}_{\min})^2}$. In other words, $\sqrt{(\Delta X^{r}_{\min})^2 + (\Delta Y^{r}_{\min})^2}$ is a lower bound for the distance of any two points. Therefore, we compare the $d_{\min}$ at the end of the rth round with this lower bound and if $d_{\min} \leq \sqrt{(\Delta X^{r}_{\min})^2 + (\Delta Y^{r}_{\min})^2}$ then there is no need to look further; the distance of our current closest pair is the final answer.

2.2 Correctness of the Algorithm

Case I Algorithm runs full $n-1$ rounds. At the rth round for each point A_i $(1 \leq i \leq n-r)$ in the x-sorted list, the algorithm computes distance to its r-closest-x neighbor where r-closest-x neighbor is the point A_{i+r} which is the point r positions left of A_i in the x-sorted array. Similarly, at the rth round, the algorithm computes for each point B_j $(1 \leq j \leq n-r)$ in the y-sorted list, the distance to its r-closest-y neighbor where r-closest-y neighbor is the point B_{i+r} which is the point r positions left of B_j in the y-sorted array. Therefore if algorithm runs all the rounds, each point will be paired with every other point (in fact twice; once as a i-closest-x neighbor and j-closest-y neighbor where $1 \leq i, j \leq n-1$). Since the algorithm goes through all possible pairings it will definitely find the closest pair. Since there are $n(n-1)/2$ pairs in total, this represents the worst case time complexity of the algorithm which is $O(n^2)$.

Case II Algorithm exits at a round r $(r < n-1)$. Clearly $d_{\min}$ is the minimum distance of any pair explored so far. Is it the minimum distance of any pair not explored yet? The answer is yes because $d_{\min}$ computed at the round r satisfies $d_{\min} \leq \sqrt{(\Delta X^r_{\min})^2 + (\Delta Y^r_{\min})^2}$ and therefore $d_{\min}$ is a lower bound on the distance of any pair that would be explored in any round $q > r$ due to (1).

2.3 Empirical Average Case Complexity

We have implemented our algorithm in C and measured the round number it terminated, under uniformly distributed random inputs of varying size. The results are shown in Fig. 1 where the best fitting curve to this data is $c_1 \log n + c_2$. Since each round's time complexity is $O(n)$, these empirical results suggest that the average case time complexity of the algorithm is $O(n \log n)$.

3 Experiments

We compared our algorithm with the closest pair algorithms provided by Jiang and Gillespie [4]. In their paper, they compared different versions of the divide and conquer closest pair algorithm (*Basic-7*, *GWZ-3*, *Basic-2*, *2-pass*, *Hybrid*, *Adaptive*, *Combined*). Furthermore, they provided their source code of the test program and the experimental data online at http://www.cs.usu.edu/~mjiang/closestpair/. We simply added our algorithm coded in C and run the same test on an Intel Core i7-920, with 12 GB memory and operating system Ubuntu 11.10 (64 bit). Test uses randomly generated inputs, sized from $2^4, 2^5, \ldots, 2^{26}$. For each input size 10 executions (with different random number seeds) are performed.

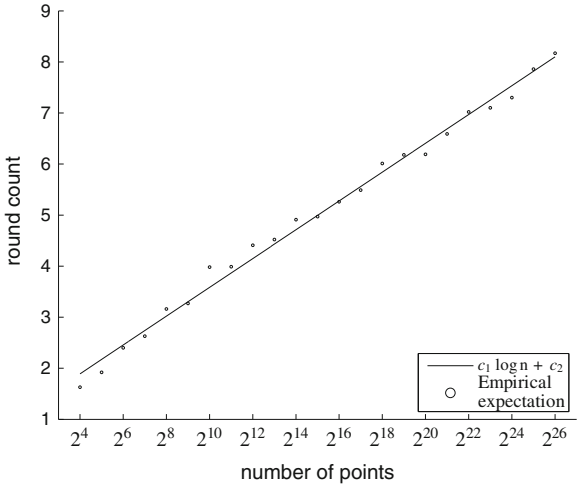

Fig. 1 Expected termination round numbers for different input sizes

Table 1 Total run times (ms) for small input sizes

n	Basic-7	Basic-2	GWZ-3	Comb.	QuickCP
2^4	0.0018	0.0016	0.0017	0.0016	0.0014
2^5	0.0044	0.0038	0.0039	0.0037	0.0033
2^6	0.0109	0.0097	0.0101	0.0080	0.0071
2^7	0.0273	0.0242	0.0250	0.0201	0.0164
2^8	0.0629	0.0575	0.0591	0.0477	0.0371
2^9	0.1428	0.1272	0.1299	0.1030	0.0789
2^{10}	0.3125	0.2725	0.2803	0.2251	0.1709
2^{11}	0.6748	0.5889	0.6016	0.5107	0.3721
2^{12}	1.5000	1.2910	1.3184	1.0527	0.7969
2^{13}	3.1719	2.7461	2.8164	2.2734	1.7031
2^{14}	6.8047	5.7344	5.9609	4.8203	3.5781

We have selected for comparison with our algorithm *QuickCP*, *Basic-7* (the basic divide-and-conquer algorithm computing seven distances for each point in the combine step), *GWZ-3* (a divide and conquer algorithm with an optimized combine step, developed by Ge et al. [3]), *Basic-2* (a hybrid of *Basic-7* and *GWZ-3*, independently developed in [4] and [7]), and *Combined* (an improved version of *Basic-2* by two heuristics, presented in [4]). We have not selected algorithm 2-pass because it was the worst of all.

Tables 1 and 2 display the total run times for *Basic-7*, *GWZ-3*, *Basic-2*, *Combined* and our algorithm *QuickCP* for small (2^4–2^{14}) and large (2^{15}–2^{26}) input sizes.

Figures 2 and 3 show, for various inputs sizes, the total run time ratios of *GWZ-3*, *Basic-2*, *Combined* and *QuickCP* over *Basic-7*. *QuickCP* is the fastest of all. For

Table 2 Total run times (s) for large input sizes

n	Basic-7	Basic-2	GWZ-3	Comb.	QuickCP
2^{15}	0.0150	0.0127	0.0130	0.0105	0.0078
2^{16}	0.0324	0.0265	0.0275	0.0227	0.0170
2^{17}	0.0684	0.0584	0.0586	0.0493	0.0371
2^{18}	0.1518	0.1266	0.1274	0.1079	0.0803
2^{19}	0.3468	0.2893	0.2938	0.2512	0.1878
2^{20}	0.8120	0.6855	0.7045	0.6220	0.4635
2^{21}	1.9800	1.7230	1.7480	1.6050	1.2320
2^{22}	4.4470	3.8940	3.9210	3.6190	2.7480
2^{23}	10.0400	8.6740	8.7770	8.1870	6.2610
2^{24}	22.0260	19.4280	19.4890	18.3900	13.5490
2^{25}	48.4350	42.5040	42.9720	40.8320	29.5090
2^{26}	108.6644	96.3528	97.2837	93.8531	66.6286

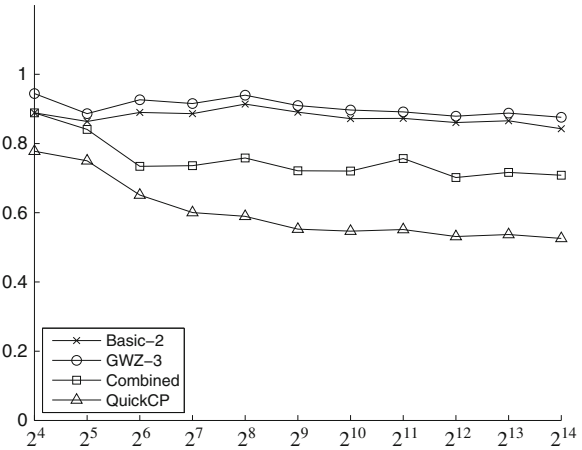

Fig. 2 Total run time ratios of *Basic-2*, *GWZ-3*, *Combined*, and *QuickCP* over *Basic-7* for small input sizes

small (2^4–2^{14}) input sizes the run time ratio of *QuickCP* over *Basic-7* is 0.61 which corresponds to a speed-up of 1.64. For large (2^{15}–2^{26}) input sizes the run time ratio of *QuickCP* over *Basic-7* is 0.57 which corresponds to a speed-up of 1.74. Overall run time ratio of *QuickCP* over *Basic-7* is 0.59 which corresponds to a speed-up of 1.70. In other words, *QuickCP* is on average 70 % faster than *Basic-7*. The second fastest algorithm is *Combined* that is on average 31 % faster than *Basic-7*. *Basic-2* and *GWZ-3*, respectively, overall 15 and 13 % faster than *Basic-7*. Combined is the most engineered and optimized algorithm among the suit in [4] and *QuickCP* is on average 39 % faster with a tendency to increase the difference with larger input sizes.

All Closest Pair Algorithms spend a large portion of their times on the preprocessing i.e., sorting. Thus, unless a sorting algorithm practically faster than QuickSort is

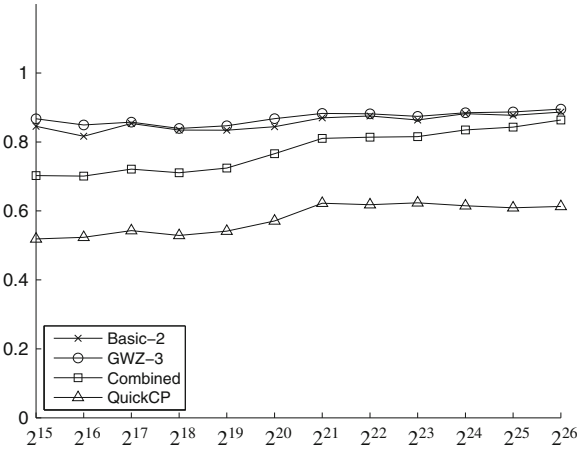

Fig. 3 Total run time ratios of *Basic-2*, *GWZ-3*, *Combined*, and *QuickCP* over *Basic-7* for large input sizes

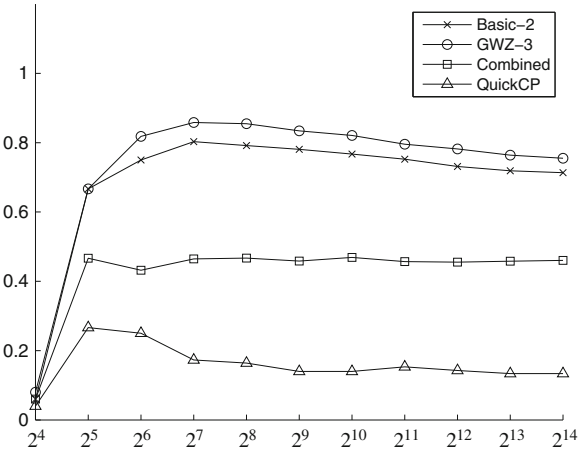

Fig. 4 (Total—preprocessing) time ratios of *Basic-2*, *GWZ-3*, *Combined*, and *QuickCP* over *Basic-7* for small input sizes

found (and takes its place in code libraries), a portion of the run time corresponding to this preprocessing can not be reduced. Therefore, in Figs. 4 and 5 we give the run times where preprocessing times excluded.

When preprocessing times are excluded, *QuickCP* is, on average, 6.95 times faster than *Basic-7* and the speed-up shows a tendency to increase with larger input sizes. Other three algorithms, *Combined*, *Basic-2* and *GWZ-3* line-up with 1.98, 1.40 and 1.32 speed-up over *Basic-7*, respectively.

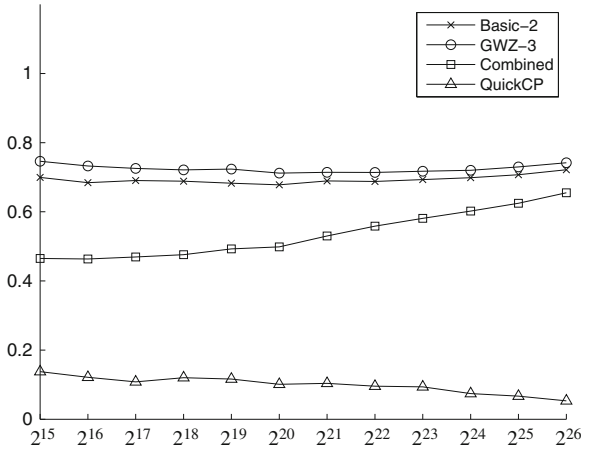

Fig. 5 (Total—preprocessing) time ratios of *Basic-2*, *GWZ-3*, *Combined*, and *QuickCP* over *Basic-7* for large input sizes

4 Improving Divide and Conquer

Noting that *QuickCP* has $O(n^2)$ worst-case time complexity, with the guaranteed $O(n \log n)$ time complexity the divide and conquer closest pair algorithms has an advantage. On the other hand, as the experiments show *QuickCP* is faster. Therefore, we can combine the strong sides of the two, leading to a faster $O(n \log n)$ algorithm. To that extent, we have modified the divide and conquer algorithms (*Basic-7*, *GWZ-3*, *Basic-2*, and *Combined*) so that they switch to *QuickCP* when sub-problems become smaller than a threshold size (determined empirically to be $2\sqrt{n}$). We renamed the new versions of the algorithms as *Basic-7**, *GWZ-3**, *Basic-2**, and *Combined**. We tested the improved versions on the same data and the same setup of the previous section.

Figures 6 and 7 show, for various inputs sizes, the total run time ratios of *Basic-7**, *GWZ-3**, *Basic-2**, and *Combined* over *Basic-7*. Our first observation is that all four algorithms provide a 60 % average speed-up over *Basic-7* for small input sizes. However, for large input sizes this relative speed-up goes down to 51 % for *Basic-2** and *GWZ-3**; and about 42 % for *Basic-7** and *Combined**. Overall average speed improvements relative to *Basic-7* are 55 % for *Basic-2** and *GWZ-3**; and 50 % for *Basic-7** and *Combined**. Compared to the original speed-ups of these algorithms over *Basic-7* we see that *GWZ-3**, *Basic-2** and *Combined* gained an additional 42, 40 and 19 % speed-up, respectively. For instance while the original implementation of *Basic-2* as given in [4] provides a 15 % speed improvement over *Basic-7*, our *QuickCP* boosted version of the same algorithm, *Basic-2**, achieves a 55 % speed improvement over *Basic-7*. *Basic-7**, *QuickCP* embedded version of the basic divide and conquer algorithm is 50 % faster than its original version, *Basic-7*. With this newly found partnership, *Basic-7** becomes about 20 % faster than Combined, the

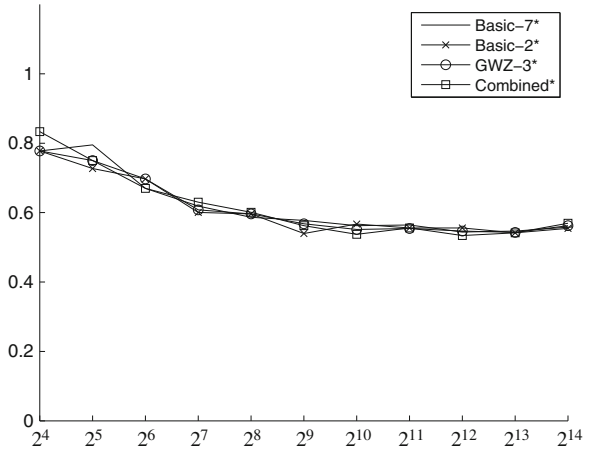

Fig. 6 Total run time ratios of Basic-7*, Basic-2*, GWZ-3* and Combined* over Basic-7 for small input sizes

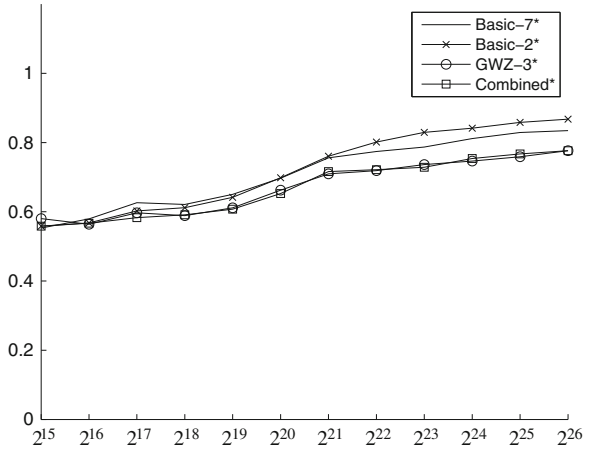

Fig. 7 Total run time ratios of Basic-7*, Basic-2*, GWZ-3* and Combined* over Basic-7 for large input sizes

most optimized divide and conquer algorithm in the set. *Basic-7** is about 35 % faster than other engineered divide and conquer algorithms, *Basic-2* and *GWZ-3*.

An interesting observation with improved divide and conquer algorithms (*Basic-7**, *Basic-2** and *GWZ-3**) is that when the small sub-problems are handled by embedded *QuickCP*, efficiently, their performances tend to become similar. In other words, performance advantages of optimized algorithms (Basic-2 and GWZ-3) over basic algorithm (Basic-7) diminish, possibly because the improvement obtained by embedding *QuickCP* in these algorithms, dominates the other optimizations.

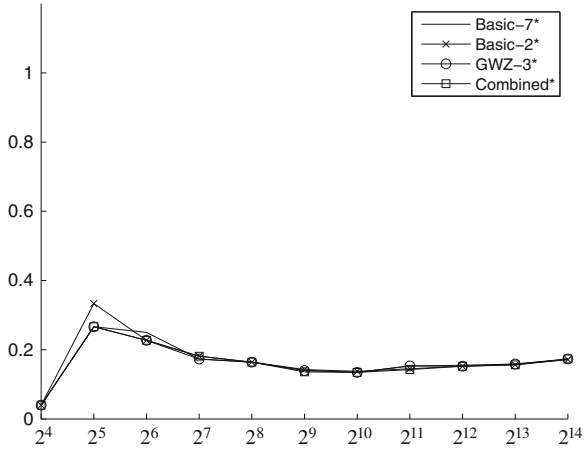

Fig. 8 (Total—preprocessing) time ratios of Basic-7*, Basic-2*, GWZ-3*, and Combined* over Basic-7 for small input sizes

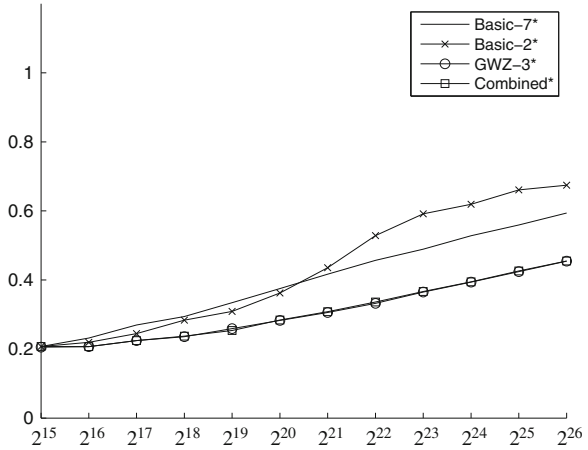

Fig. 9 (Total—preprocessing) time ratios of Basic-7*, Basic-2*, GWZ-3*, and Combined* over Basic-7 for large input sizes

When we exclude the preprocessing times, results as displayed in Figs. 8 and 9, show that compared to the original speed-ups of these algorithms over *Basic-7* we see that *Basic-7**, *GWZ-3**, *Basic-2** and *Combined** gained an additional 221, 260, 251 and 114 % speed-up, respectively.

Even though the impact of this high speed-up improvements on the total run times are relatively less e.g., 40 % for *Basic-2**, this represent a significant improvement on an already engineered and fine-tuned divide and conquer algorithm. As a final note, the run time values given in [4] excludes the preprocessing times and the best

improvement ratio they obtain is 0.558 which corresponds to 79 % speed improvement which is achieved via the *Combined* algorithm. In this paper by embedding the *QuickCP* we have obtained a better version of the *Combined* (labeled *Combined**) with an additional speed improvement of 114 % when preprocessing times are excluded. Furthermore, this 114 % speed-up is a relatively low improvement rate compared to the average improvement rate of about 250 % that we have achieved for the other three divide and conquer algorithms.

5 Conclusion

We have presented a new and practical algorithm, *QuickCP*, for closest-pair problem. *QuickCP* is both simpler and faster than the divide-and-conquer algorithms for computing the closest pair of points. We have implemented and tested *QuickCP* for random inputs up to 60 million points and on average it is 70 % faster than the classical divide and conquer while about 40 % faster than the most optimized divided and conquer algorithm (called *Combined* in the paper). Furthermore, when we exclude the preprocessing times *QuickCP*'s speed advantage gets much bigger.

Another contribution of *QuickCP* is that it can be embedded in the divide and conquer algorithms to quickly handle small cases. We have pursued this option and observed that even the classic divide and conquer closest pair algorithm (called Basic-7 in the paper) becomes faster than the most optimized divide and conquer algorithm.

QuickCP algorithm described in two-dimensions in this paper, can easily be extended to three and upper dimensions. Also we are planning to adapt our approach to other problems in Computational Geometry and Graphics.

Acknowledgments We sincerely thank Minghui Jiang and Joel Gillespie for their generosity of sharing their source code and experiment data. We are planning to do the same after publishing our research.

References

1. Bentley JL, Shamos MI (1976) Divide-and-conquer in multidimensional space. In: Proceedings of the eighth annual ACM symposium on theory of computing, STOC '76. ACM, New York, NY, USA, pp 220–230
2. Dietzfelbinger M, Hagerup T, Katajainen J, Penttonen M (1997) A reliable randomized algorithm for the closest-pair problem. J Algorithms 25(1):19–51
3. Ge Q, Wang HT, Zhu H (2006) An improved algorithm for finding the closest pair of points. J Comp Sci Technol 21:27–31. doi:10.1007/s11390-006-0027-7
4. Jiang M, Gillespie J (2007) Engineering the divide-and-conquer closest pair algorithm. J Comput Sci Technol 22(4):532–540
5. Khuller S, Matias Y (1995) A simple randomized sieve algorithm for the closest-pair problem. Inf Comput 118(1):34–37

6. Kleinberg J, Tardos E (2006) Algorithm design. Addison-Wesley, Boston
7. Pereira JC, Lobo FG (2012) An optimized divide-and-conquer algorithm for the closest-pair problem in the planar case. J Comp Sci Technol 27(4):891–896
8. Preparata FP, Shamos MI (1985) Computational geometry: an introduction. Springer, New York
9. Smid M (1997) Closest-point problems in computational geometry. Handbook of computational geometry, pp 877–935

DARWIN: A Genetic Algorithm Language

Arslan Arslan and Göktürk Üçoluk

Abstract This article describes the DARWIN Project, which is a Genetic Algorithm programming language and its C Cross-Compiler. The primary aim of this project is to facilitate experimentation of Genetic Algorithm solution representations, operators and parameters by requiring just a minimal set of definitions and automatically generating most of the program code. The syntax of the DARWIN language and an implementational overview of the the cross-compiler will be presented. It is assumed that the reader is familiar with Genetic Algorithms, Programming Languages and Compilers.

1 Introduction

The goal of the DARWIN Project is twofold - first, creating a programming language with support for Genetic Algorithm concepts and second, by requiring just a minimal set of specifications, generating all the necessary code automatically. The DARWIN Project thus involves a language design and its compiler implementation.

In general, if creation of a GA solution to a problem is desired, first the data representation is selected, the set of GA operators designed and the parameters specified. For example, lets decide that an array will store our chromosome and that we will implement 1-point cross-over with a binary flip mutation and a simple objective score function. Up to that point it is simple, but there is a lot to do. Next, we have to decide what kind of scaling and how selection will be performed. Thus, we create an array to store the statistics of the current population and define scaling and selection

A. Arslan (✉)
LOGO, Ankara, Turkey
e-mail: arslan.arslan@logo.com.tr

G. Üçoluk
Middle East Technical University, Ankara, Turkey
e-mail: ucoluk@ceng.metu.edu.tr

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_4,

functions to use the information stored there-in. Finally, we code the genetic algorithm, to create a population and start evolving it until a termination condition is satisfied. At that point, we have a completely functioning Genetic Algorithm solution and we can start experimenting with different operators and GA parameters. This can be a time consuming process, since certain GA operators should be rewritten and modified, then the system should be tested against different inputs and parameters. It is custom to have hundreds of runs until a proper set of parameters is obtained. So a tool or a library is needed to facilitate creation of an initial system and afterwards help experimenting in the search of a better system.

Although GALib is easy to use, it is not that easy to extend. For example, creating a chromosome representation matching any of the available chromosome classes is easy, but when a more complex structure is needed a separate class should be derived and all the operators be redefined. This is still acceptable, but if creating a custom population object or even worse, a hand-made genetic algorithm object is required, then the programmer should be pretty much involved in the GALib object implementation and interaction details.

Another drawback of GALib comes from the fact that classes are as "general", i.e. data structure independent, as possible. So inefficient data storage arises, just because the implementation fails to specialize. To cover up, GALib provides many specialized implementations of the same object. For example, GAListGenome, GATreeGenome, GABinaryString and GAArray objects derive from the base class named GAGenome and provide support for specific genome representations. But this is not the end of the story, since the above mentioned four classes are in turn base classes for more and more specific data representations and this results in a fairly big object hierarchy.

In general Object Orientation is a handy concept, because implementation details can be hidden and inheritance is a fast way of complying to the overall library conventions, but expertise in C++ is required to accomplish tasks with even moderate difficulty. Another potential problem is the fact that inheritance hides the implementation details of the base class, so the resultant object inherits both the good and bad sides of his parents. As a result, the cumulative effect of the inefficient implementation during the path of derivation can add up to a point where performance is severely reduced.

The *General-purpose* systems go a step further, compared to a *Algorithm Library*, in the sense that they provide a high-level language to program their libraries. These high-level languages allow extremely fast prototyping, when experimenting with the system parameters. But, there is another reason for having such languages—the algorithm libraries are extremely complex and thus difficult to understand and extend in their low-level implementations. In addition, all such systems require a special *kernel* to execute these high-level GA instructions. The kernel may not be portable and can even rely on some parallel hardware.

As a solution, a programming language naturally recognizing and providing support for genome representations and their operators can be designed and implemented. By having a separate language and a compiler, all the necessary code can be generated for the unspecified genetic operators, thus giving a change for the developer

to focus just on genome representation, operators and proper parameter set design. In addition, since genome data representation is known by the compiler, efficient code for dealing with the genome data structure can be automatically generated, thus eliminating inefficiency issues.

First, previous work related to the DARWIN project will be presented. In Sect. 4, an example of a DARWIN program will be given. In Sect. 5, An overview of the design of DARWIN and its implementation will be covered. We conclude in Sect. 7.

2 Related Previous Work

Genetic Algorithms received wide spread support due to the fact that they are robust and their complexity is not linear with respect to the number of parameters encoded in the solution. In fact, Genetic Algorithms were even used in obtaining near optimal solutions for many of the problems previously considered NP or NP-hard. But the task of producing a successful Genetic Algorithms system is not simple. This is due to the fact that experimentation with different GA operators and parameters is needed. So, some tools for facilitating GA program creation and experimentation with different genome structures and operators are needed.

In its most general case, the GA Programming Environments can be grouped into the following major classes [9]:

Application-oriented systems are programming environments hiding all the implementation details and are targeted at business professionals. These systems focus on a specific domain such as scheduling, telecommunications, finance, etc. A typical application-oriented environment contains a menu-driven interface giving access to parameterized implementations of genetic algorithms targeted at specific domains. The user can configure the application, change parameters and monitor program execution. Examples of application-oriented systems are EVOLVER, OMEGA, PC/BEABLE, XpertRule GenAsys, etc.

Algorithm-oriented systems are programming environments with support for specific genetic algorithm:

Algorithm-specific systems contain just a single powerful genetic algorithm. In general, these systems come in source code and allow the expert user to make changes. The implementations are usually modular in structure and there are nearly no user interfaces. Examples of these systems are ESCPADE, GAGA, GAUSCD, GENESIS and GENITOR.

Algorithm Libraries contain variety of genetic algorithms and operators are grouped in a library. These systems are modular, allowing the programmer to select among parameterized implementations of different algorithms and operators. Usually these libraries are implemented in C/C++ and Lisp. Examples of algorithm libraries are GALib, EM and OOGA.

Algorithm-oriented systems are often GNU Public Licensed and include free source code. Thus they can be easily incorporated into user applications.

Tool Kits are programming systems that contain many programming utilities, algorithms and genetic operators that can be used in a wide range of application domains. These programming systems subdivide into two:

Educational systems help the novice user to get familiar with Genetic Algorithm concepts. Typically these systems are very user friendly and support just a few options for configuring an algorithm. GA Workbench is the only example for an educational system.

General-purpose systems provide a comprehensive set of tools for programming any GA and application. These systems may even allow the expert user to customize any part of the software. Generally, these systems provide:

- Sophisticated user interface
- Parameterized algorithm library
- A high level language to program the system library
- Open architecture for expert user modification.

Examples include EnGENEer, GAME, MicroGA, PeGAsus and Splicer.

From the programmers point of view, *General-purpose* systems are the best tools available for programming GA systems. But, since these are not freely available, the current trend is creating libraries of functions or classes implementing predefined GA operators. Perhaps the most complete and easy to use Genetic Algorithms package is GALib. GALib is a C++ Library of Genetic Algorithm Components and is written and maintained by Matthew Wall, Mechanical Engineering Department of Massachusetts Institute of Technology [10].

GALib is a hierarchy of objects representing different GA constructs. The top level objects are GAGenome, GAPopulation and GAGeneticAlgorithm. The Genetic Algorithms operators are thus implemented as methods of these objects. The library features a full set of ready to use specialized versions of these top level objects, so a working program can be fastly prototyped, but experimentation usually requires modifications in the default behavior implemented by the classes. GALib provides two extension methods—deriving own classes to implement custom behavior, or defining new methods and instructing GALib to use them instead of the defaults.

3 The DARWIN Language

Stated briefly the solution of a problem using a Genetic Algorithm involves these important steps:

- Defining a representation
- Defining the objective score function
- Defining the genetic operators

The DARWIN language is designed to clearly distinguish Genetic Algorithm constructs—the genome and its operator set. It's language compiler has been designed to be a C cross-compiler. The target language is chosen to be C, since C is the most

commonly used language with compilers present on the widest range of platforms. By generating standard C code, portability is ensured on the largest scale possible.

The DARWIN language is designed to be simple, GA oriented language. In order to achieve fast learning and adaptation, the syntax of the language resembles the standard C language syntax. For example, the DARWIN statements look just the same in C, with the minor difference that DARWIN is not as rich with build-in types as C and there are no pointers. The reason for excluding pointers from the language is that they are not currently needed, since all the storage in the generated code is linear. This does not lead to inefficient implementation since the generated code contains C pointers.

The DARWIN language, is capable of distinguishing *genetic algorithm constructs*. In the first place, DARWIN syntax provides means for defining *genes*, *chromosomes*, *populations* and the *genetic algorithm*.

The best possible way to represent them would be to have object classes. But since DARWIN generates plain C code, the DARWIN language has a special syntax construct called *moderators*. Moderator is a function that effectively associates an operation on a data structure, thus creating the effect of encapsulation present in Object Oriented programming. By introducing the concept of moderators, the DARWIN language becomes powerful enough to express a genetic algorithm construct and provide the standard set of operators associated with it. For example, the gene construct has the *print*, *initialize*, *crossover*, *mutate* and *evaluate* operations defined on it.

4 An Example of a DARWIN Program

4.1 Example Problem Definition and its Specifications

Consider an example problem of finding a two dimensional grid distribution of 20 given rectangles such that:

- No rectangles overlap.
- The *bounding box* (a smallest rectangle that includes all the rectangles) has a minimal area.
- Rectangles can be placed are allowed to be placed anywhere provided that one of their sides (any of it) is parallel to the horizontal direction. Furthermore all corners have to remain inside the grid (no clipping).
- The input of the problem consists of the dimensions of the rectangles.
- The grid dimensions are 256×256. The origin of the grid is labeled as $(0, 0)$. This means that the upper-right corner of the grid is $(255, 255)$. The horizontal axis is denoted with the letter x and the vertical axis with y. In a 2-tuple representation our convention will be writing (x, y) (as usual).
- Any coordinate (i, j) have positive integers for both i and j.
- Each rectangle is represented by a tuple $\langle Width, Height, x_{left}, y_{lower} \rangle$

4.2 DARWIN code of the solution

```
% a user defined function ;
external function int random(int start, int end);

gene TRectangle                    % start of genome
                                            definition;
{  int x;
   int y;
   frozen int width;
   frozen int height;
} < evaluator : RectangleEval, % evaluator moderator
                                     specification;
   init : RectangleInit;       % initializer
                              moderator specification;

chromosome TRectangles
{  TRectangle rectangles[20]; };

population Pop
{  TRectangles individuals[100]; };

algorithm GA
{ Pop population; };               % end of genome
                                          definitions;

% gene TRectangle evaluator moderator;
function float RectangleEval(TRectangle gene)
{
   if ((gene.x > 100) || (gene.y > 100))
     return 0;
   return 1;
}

% gene TRectangle initializer moderator;
function RectangleInit(TRectangle gene)
{
   gene.x = random(0,100);
   gene.y = random(0,100);
   gene.width  = random(0, 100-gene.x);
   gene.height = random(0, 100-gene.y);
}
```

5 Implementation

5.1 General Structure

The DARWIN project implementation consists of 4 major parts and an external code template database. It consists of

- Parser
- Post-processor
- Template Database
- Electra Engine
- Unparser

The general structure of the DARWIN cross-compiler is Fig. 1.

The *Parser* is the entity responsible for imposing the DARWIN Language syntax and converting a DARWIN program into a computer readable form—the parse tree. This parse tree is in turn fed into the *Post-Processor*, which in turn performs type-checking and verifies language semantics. The output of the Post-processor is an abstract syntax tree, equivalent to the parse tree but restructured in a form suitable for

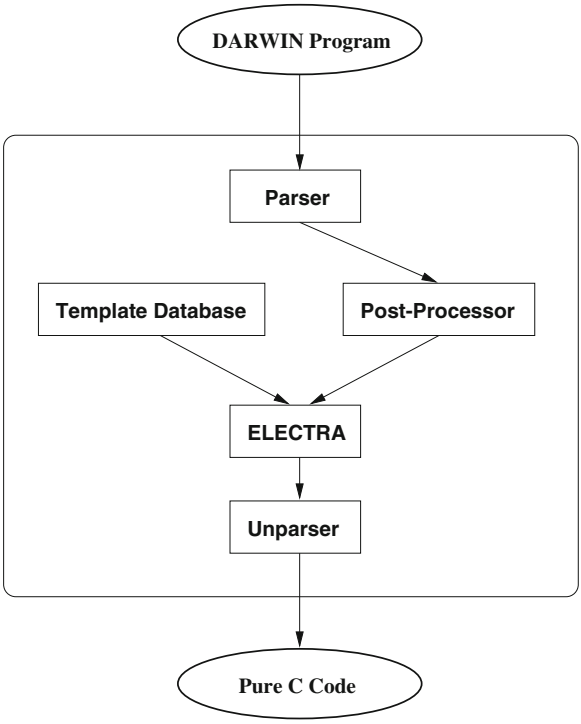

Fig. 1 Structure of DARWIN cross-compiler

traversal by the *Electra Engine*. In addition, the Post-Processor produces the symbol and type tables. Next, the abstract syntax tree is passed to the Electra Engine, which performs abstract tree translation and code generation. The Electra Engine relies on the presence of a *template database*, which is loaded prior to starting the engine. The template database is a text file containing abstract tree templates, which are matched inside the abstract tree of the DARWIN program and translated accordingly. The output of the Electra Engine is an abstract tree representing valid C code. The Unparser, takes this abstract tree and produces a formatted C program code.

5.2 Parser

The Parser converts the DARWIN syntax into its equivalent parse tree. The input is a DARWIN program file and the output of the Parser is a LISP structure, representing its parse tree. The Parser uses the parsing property list driven engine which is easily customizable and more efficient than the table-driven approach employed by LALR(1) parsing.

5.3 Post-Processor

The DARWIN Post-Processor operates in two stages. In the first stage it performs type-checking, manipulates the symbol table and collects all the necessary information that will be needed during the actual code generations. The Post-Processor is also responsible for tagging each branch in the abstract syntax tree. These tags will be used during the code generation phase, since the Electra engine will decide upon them.

After having performed the validity checks of the functions and genetic algorithm constructs, the Post-Processor enters its next stage, where the moderators specified in the genetic constructs are extracted from the *functions* section and are placed at the section they actually belong. During this phase, each of the genetic constructs are inspected and all of their moderators are searched in the functions section. Any inconsistency end up in an exception raise.

5.4 Electra and Template Database

Electra is a rule based unification pattern matcher/substituter and is the core code generating engine of the DARWIN compiler. It effectively traverses the input from the Post-Processor section by section, and decides when some additional code should be generated and how a given subtree should be transformed. So the Electra engine is traversing an abstract syntax tree, matching a given abstract syntax subtree and transforming it. The output is an abstract syntax tree, organized in sections.

6 Unparser

The Unparser is the abstract syntax to C code translator. The implementation adopts the one-pass pretty printer developed by Hearn and Norman [8]. The input of the Unparser is the output of the Electra Engine, where all of the syntactic sugars were removed, code translations done and any additional code generated. The output is a pretty printed C program, that corresponds to the user's DARWIN program.

6.1 Implementation Environment

The DARWIN Cross compiler is implemented in ALISP [4]. ALISP is an Algol like procedural syntactic sugar of Standard LISP [5]. Compared to LISP ALISP is much more readable. Technically they are identical, since an easy 1-to-1 mapping exists from ALISP to LISP counterpart. For portability reasons the project code is distributed as LISP code and ALISP is just used for ease of implementation.

The Standard LISP used belongs to Karabudak et al. [5]. The benefit of this LISP implementation is that it is free and contains an efficient LISP to C translator, thus achieving portability. In addition, when a LISP code is compiled a major speedup of up to 10 times of the interpretive execution speed can be achieved.

7 Conclusion

The DARWIN project aimed at creating a Genetic Algorithm programming to facilitate fast GA system creation and easy experimentation with different parameters and GA operators. We believe, that the goals are met.

Using DARWIN, a working system can be created with just presenting the genome definition and its evaluation function. In this case, the DARWIN compiler generates up to 85–90 % of the total code. Provided with a library of generator functions, a programmer can continue experimenting with different operator sets by just specifying library-provided moderators in the GA construct definitions. This still means, that the systems developer will be writing just 10 % of the total code and the rest will be automatically generated.

If the generator library does not provide the desired functionality, then the programmer will have to implement moderators using the DARWIN language. In an extreme case still we expect the DARWIN cross-compiler to generate around 45–50 % of the code on the average.

DARWIN together with LISP and ALISP is distributed freely under GNU Public License at the following URL: http://www.ceng.metu.edu.tr/ ucoluk/darwin/

References

1. Goldberg DE (1953) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Publishing Company, Inc, Reading
2. Bäck T (1996) Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press, New York, Oxford
3. Michalewicz Z (1992) Genetic algorithms + data structures = evolution programs. Springer, Berlin Heidelberg New York
4. Karabudak E, Üçoluk G (1982) ALISP—The manual
5. Karabudak E, Üçoluk G, Yılmaz T (1990) A C portable LISP interpreter and compiler, METU
6. Marti J, Hearn AC, Griss ML, Griss C (1979) The standard LISP report, SIGPLAN Notices, pp 48–68, no 10, ACM, New York, 14, 1979
7. Griss ML, Hearn AC (1981) A portable LISP compiler. Softw Pract Exp 11:541–605
8. Hearn AC, Norman AC (1979) A one-pass prettyprinter, SIGPLAN Notices, pp 50–58, no 12, ACM, New York, 14, 1979
9. Filho JR, Alippi C, Treleaven P (1994) Genetic algorithm programming environments. IEEE Comput J 27:28–43
10. Wall M (1996) GALib: A C++ library of genetic algorithm components, Ver.2.4

Distributed Selfish Algorithms for the Max-Cut Game

D. Auger, J. Cohen, P. Coucheney and L. Rodier

Abstract The Max-Cut problem consists in splitting in two parts the set of vertices of a given graph so as to maximize the sum of weights of the edges crossing the partition. We here address the problem of computing locally maximum cuts in general undirected graphs in a distributed manner. To achieve this, we interpret these cuts as the pure Nash equilibria of a n-player non-zero sum game, each vertex being an agent trying to maximize her selfish interest. A distributed algorithm can then be viewed as the choice of a policy for every agent, describing how to adapt her strategy to other agents' decisions during a repeated play. In our setting, the only information available to any vertex is the number of its incident edges that cross, or do not cross the cut. In the general, weighted case, computing such an equilibrium can be shown to be PLS-complete, as it is often the case for potential games. We here focus on the (polynomial) unweighted case, but with the additional restriction that algorithms have to be distributed as described above. First, we describe a simple distributed algorithm for general graphs, and prove that it reaches a locally maximum cut in expected time $4\Delta|E|$, where E is the set of edges and Δ its maximal degree. We then turn to the case of the complete graph, where we prove that a slight variation of this algorithm reaches a locally maximum cut in expected time $O(\log \log n)$. We conclude by giving experimental results for general graphs.

D. Auger (✉) · J. Cohen · P. Coucheney · L. Rodier
PRiSM-CNRS, 45 avenue des Etats-Unis, Versailles, France
e-mail: David.Auger@prism.uvsq.fr

J. Cohen
e-mail: Johanne.Cohen@prism.uvsq.fr

P. Coucheney
e-mail: Pierre.Coucheney@prism.uvsq.fr

L. Rodier
e-mail: Lise.Rodier@prism.uvsq.fr

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_5,

1 Introduction

One of the fundamental goals of algorithmic game theory is to design efficient algorithms to compute equilibria of games. In this paper, the focus is about the computation of pure Nash equilibria for the *maximum-cut* (Max-Cut) game, in a distributed selfish manner. The maximum-cut game is close to the *party affiliation* game [6]. For instance, the Max-Cut game [9] can model the competition among n agents communicating via radio signals with only two distinct available frequencies. The purpose of each agent is then to choose her frequency in order to minimize the sum of interferences that she experiences.

The Max-Cut game is also a congestion game, a particular subclass of *potential games* [13] which are known to always possess a pure strategy Nash equilibrium. Computing a pure equilibrium is known to be PLS-complete [14] in many classes of potential games, including the Max-Cut game. Let us recall that the PLS complexity class includes all problems of local search where neighborhoods can be computed in polynomial time. Furthermore, a common belief is that it seems very unlikely that a PLS-complete problem would admit a polynomial time algorithm.

Related Work. The Max-Cut problem has been studied extensively [7], even in the local search setting. It is well known that finding a local optimum for Max-Cut is PLS-complete [14], and there are some configurations from which moving to a local optimum is exponentially long. Every Nash equilibrium of the Max-Cut game corresponds to a local optimum whose computation is sometimes polynomial in the unweighted case [11] but PLS-complete in general [14]. We shall now focus on the unweighted case. The best known approximation algorithm gives an α-approximation [7] where $\alpha = 0.87\ldots$. Khot et al. [10] showed that, if the unique game conjecture is true, then this is the best possible approximation ratio.

A potential game always admits a pure Nash equilibrium: since the potential function, which could take only a finite number of values, is strictly decreasing in any sequence of pure best responses moves, such a sequence must be finite and must lead to a Nash equilibrium [12]. For load-balancing games, which are potential games, the bounds on the convergence time of best-response dynamics have been investigated in [5]. Since agents play in turns, this is often called the *Elementary Stepwise System*. Other results of convergence in this model have been investigated in [8], but all require some global knowledge of the system in order to determine the next move. A stochastic version of the best-response dynamics, where agents only have a local view of the system, has been investigated by Berenbrink et al. [2]. Aldophs and Berenbrink [1] have improved this result by considering the network topology. Christodoulou et al. [4] studied the convergence time to an approximate Nash Equilibrium in the Max-Cut game obtained after a polynomial number of best response steps. Unlike this previous work, we assume that all agents are able to simultaneously change their strategy in a distributed way with a local view of the system.

Definition and basic game theory framework. Let $G = (V, E)$ be an undirected graph. A *cut* in G is a partition of V in two sets A and B (which we call *sides*), and

the *size* of such a cut is the number of edges having their endpoints in different sides. The Max-Cut problem consists in computing a cut which size is maximized.

A cut (A, B) with $A \cup B = V$ will be called *locally maximum* if changing the side of any single vertex only reduces the size of the cut. A simple, non-distributed algorithm which computes such a cut is simply to move vertices one by one from one side to the other as long as it improves the size of the cut. When this is no more possible, a locally maximum cut has been reached. This should make clear that a locally maximum cut can be computed in time $O(|E|)$.

This paper studies a strategic game defined upon Max-Cut. Each vertex i of the graph is an agent and her strategy is to choose one side, i.e $s_i \in \{-1, 1\}$. When strategies are fixed, we define the utility of agent i, denoted by u_i, as the number of its incident edges that cross the cut, i.e.

$$u_i = \left| j \in V : ij \in E \text{ and } s_i \neq s_j \right| .$$

We also denote by $r_i(S)$ the quantity $\delta_i - u_i(S)$, where δ_i is the degree of vertex i. An agent i is then called *improvable* if she may improve her utility by changing her strategy, i.e. if $r_i - u_i > 0$. Given a strategy profile $S = (s_1, s_2, \cdots, s_n)$, denote (S_{-i}, s') the profile where s_i is replaced by s' while strategies of the other agents remain unchanged. A profile S is a *Nash equilibrium* (NE) if there are no agent i and strategy s' such that $u_i(S_{-i}, s') > u_i(S)$, i.e. if no agent is improvable. Hence, Nash equilibria are in one-to-one correspondence with locally maximum cuts as described above.

Distributed algorithms framework. As described above, we focus on distributed selfish algorithms where only local information is available. More precisely, here is the general framework for such an algorithm:

Algorithm 1: Local Selfish Improvement Algorithm (LSIA)

Input: an undirected graph G
Output: a locally maximum cut (S_{-1}, S_1)
Initialization : choose a starting profile $S_0 = (s_1, s_2, \cdots, s_n)$; set $t = 0$
while *some agent is improvable* **do**
  for *each agent i in parallel* **do**
    change strategy s_i with probability $p(t, u_i(t), r_i(t))$
  end
  set $t \leftarrow t + 1$
end

Hence at each round, each agent changes her strategy independently of other agents, with a probability that depends only on the time step, and the number of her incident edges that cross and do not cross the partition. Note that these policies are memoryless, and that players are indistinguishable.

Our contribution. The next section focuses on general graphs. We prove that for a fixed probability of changing one's strategy, Algorithm 1 reaches an equilibrium (i.e., a locally maximal cut) in $4\Delta|E|$ expected rounds. In Sect. 3 we consider complete graphs and prove that for a carefully chosen probabiliy, an equilibrium is reached

within $O(\log \log n)$ steps, where n is the number of vertices in the complete graph. All proofs can be found in the appendix. Finally in Sect. 4 we give some experimental results for different types of graphs.

2 General Graphs

We use as a (maximizing) potential function Φ the size of the cut given a strategy profile $S = (s_1, s_2, \cdots, s_n)$:

$$\Phi(S) = \left| ij \in E : s_i \neq s_j \right|. \tag{1}$$

This potential is simply related to the utility of the agents by

$$\Phi(S) = \frac{1}{2} \sum_{i \in V} u_i(S)$$

In the framework of Algorithm 1, we apply this simple Rule:

Rule 2: (for general undirected graphs)

if *agent i is improvable* **then**
  change her strategy with fixed probability $p = \frac{1}{2\Delta}$
else
  keep the same strategy
end

Using this rule, we prove that:

Theorem 1 *Let Δ be the maximum degree of graph $G = (V, E)$, and let $p = \frac{1}{2\Delta}$ for all agent i and $t > 0$. Then if Rule 2 is applied, a locally maximum cut is reached in $4\Delta|E|$ expected rounds.*

We have not been able to find a bound on the convergence time with a probability depending on the utility in the general case. The algorithm was applied to different topologies of graphs in order to understand their influence on the convergence time.

3 Complete Graphs

In the case of complete unweighted graphs with n vertices, we notice that the search for a maximum cut is equivalent to the load balancing of n unweighted tasks on two identical resources. This special case of load balancing games is analyzed by the

method proposed by Berenbrink et al. [2] and we adapt this one. This rule can be described as follows for a graph G:

Rule 3: (for a complete graph)

if *agent i is improvable* **then**
  change her strategy with probability $p(r_i) = 1 - \frac{n-1}{2r_i}$
else
  keep the same strategy
end

Since the graph is complete, the utility of an agent equals the number of agents having a different strategy. Hence, all agents on the same side have in fact the same probability to switch strategy. Furthermore, only agents in the largest subset have an incentive to switch strategy in order to increase their utility.

In this case, we prove a fast expected convergence:

Theorem 2 *When Rule 3 is applied on a complete graph with n vertices, a locally maximum cut is reached in expected time* $O(\log \log n)$.

Hence Rule 3 for complete graphs is much more efficient than Rule 2.

4 Other Graphs and Simulations

Algorithm 1 is applied to general graphs. We set the probability to change strategy by $p_i(t) = 1 - \frac{u_i(t)+1}{r_i(t)+1}$ for each agent i and round t. Unfortunately, we do not have any theoretical results. In order to approach the convergence time, the algorithm was implemented in different type of graphs: paths, rings, and random graphs. For each type of graphs, considered graphs have a number of vertices n between 10 and 5,000. Each point of the figure is obtained by running a hundred times the algorithm with an initial state where all the agents select a strategy uniformly at random.

Random graphs. For random graphs, a graph is generated with a fixed number n of vertices and a probability α that there is an edge between two vertices. Simulations are done for several values of α and obtained results are similar.

In Fig. 1, where $\alpha = \frac{1}{2}$, we could observe that with a confidence interval of 95 % the upper bound convergence time to reach an equilibrium was less than $n \log |E|$. Since random graphs used are dense (a small diameter and a large degree), the algorithm seems to converge more slowly than for complete graphs. This result underlines that the topology of the graph has an impact on the convergence time (Fig. 2).

Paths and rings. Surprisingly, for paths and rings, the convergence time seems to be constant and we try to give an idea of why. This result's linked to the topology of the graph. Here all agents have a degree 2. Note that for an edge whose configuration is $[-1, 1]$ (or $[1, -1]$), the vertices incident to it do not change strategy. Indeed for rings every agent has the same probability of changing strategy : if $r_i(t) > u_i(t)$, then agent i wants to change strategy with $p_i(t) = p = 1 - \frac{u_i(t)+1}{r_i(t)+1} = \frac{2}{3}$ (because

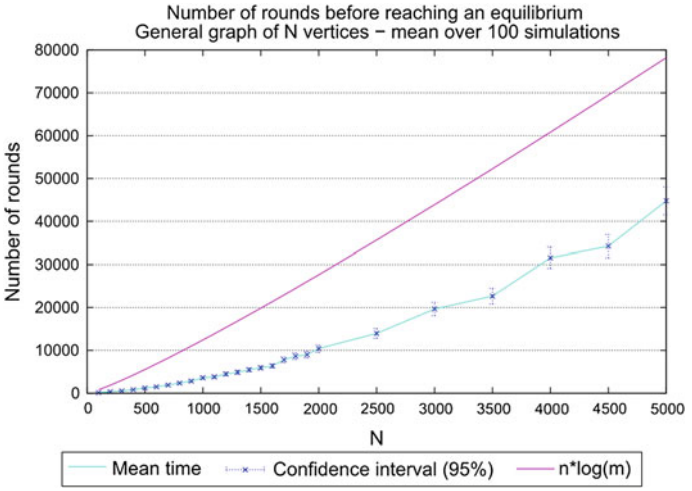

Fig. 1 Mean convergence time to reach an equilibrium for random graphs

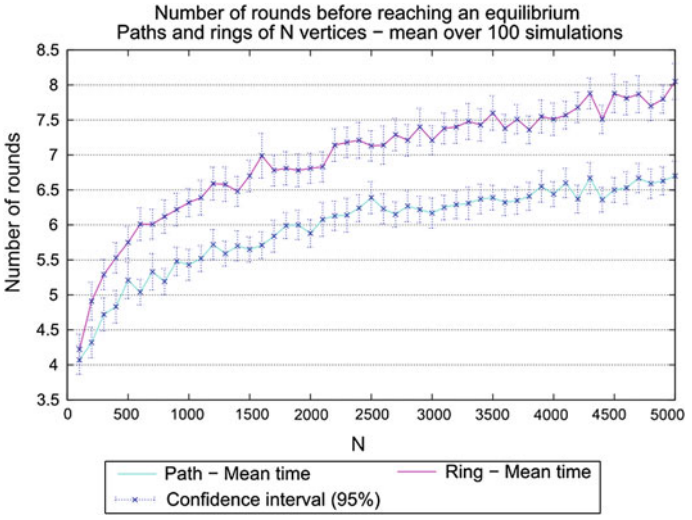

Fig. 2 Mean convergence time to reach an equilibrium for paths and rings

$r_i(t) = 2$ and $u_i(t) = 0$ since agent i has at most 2 neighbors). For path the argument is similar except that the probability of changing strategy for its endpoints is $\frac{1}{2}$.

Each time an edge with the pattern $[-1, 1]$ (or $[1, -1]$) appears in the path or ring then each extremity has no incentive to move. The problem is divided into several sub-problems. Each sub-problem correspond to a path formed, where all agents belong to the same subset. Hence, the remaining expected time to reach an equilibrium seems to be related to the expected time to reach an equilibrium in the longest path.

5 Conclusion

This paper focus on the Max-Cut game which is a potential game. First, for general graphs the probability $\frac{1}{2\Delta}$ of each agent to change strategy is fixed, where Δ is the maximum degree. We proved that with this probability our algorithm reaches an equilibrium in $4\Delta|E|$ expected rounds. Second, we consider the game when G is a complete graph and proved that our algorithm reaches an equilibrium in $O(\log\log n)$ expected time, which is almost a constant time for practical issues.

A slight modification of the algorithm for complete graphs was applied to different topologies of graphs in order to understand their influence on the convergence time. In the general case, if one fixes the probability to change strategies regardless of the agent's current utility, a degradation on the convergence time is observed. In fact, the results of simulations presented underline that the graph topology has a strong influence on the average time of convergence. This could be seen especially in the case of cycles and paths for which this convergence time seems constant. For future works we aim to analyze the link between the convergence time and graphs' topology.

Appendix

Proof (Theorem 1). Let $X(t)$ be the profile of the game at round t. We use as a potential function $\Phi(X(t))$ the sum of the edges of the cut at time t. Let $B(t)$ be the set of vertices having an incentive move at time t, i.e. those vertices i such that $r_i(t) - u_i(t) \geq 1$. Now we decompose $B(t)$ in two parts:

- $B_1(t)$ contains those vertices of $B(t)$ that move at time t;
- $B_0(t) = B(t) \setminus B_1(t)$ contains those vertices of $B(t)$ which do not move at time t.

The edges either entering or leaving the cut at time t must have an end in $B_1(t)$ and the other in $V \setminus B_1(t)$. Hence,

$$\begin{aligned}
\Phi(t+1) &= \Phi(t) + \sum_{i\in B_1(t)} \sum_{j\in V\setminus B_1(t)} \left(\mathbb{1}_{ij\in C(t)} - \mathbb{1}_{ij\in E\setminus C(t)}\right) \\
&= \Phi(t) + \sum_{i\in B_1(t)} \sum_{j\in V} \left(\mathbb{1}_{ij\in C(t)} - \mathbb{1}_{ij\in E\setminus C(t)}\right) - \sum_{i\in B_1(t)} \sum_{j\in B_1(t)} \left(\mathbb{1}_{ij\in C(t)}\right. \\
&\quad \left. -\mathbb{1}_{ij\in E\setminus C(t)}\right) \\
&= \Phi(t) + \sum_{i\in B_1(t)} (r_i(t) - u_i(t)) - \sum_{i\in B_1(t)} \sum_{j\in B_1(t)} \left(\mathbb{1}_{ij\in C(t)} - \mathbb{1}_{ij\in E\setminus C(t)}\right) \\
&\geq \Phi(t) + |B_1(t)| - \sum_{i\in B_1(t)} \sum_{j\in B_1(t)} \mathbb{1}_{ij\in E}
\end{aligned}$$

where for the last inequality we use fact that $r_i(t) - u_i(t) \geq 1$ if $i \in B_1(t)$. Since every vertex of $B(t)$ has probability p of being in $B_1(t)$, independently of other vertices, and every vertex has degree at most Δ, comes:

$$\mathbb{E}[\Phi(t+1)|X(t)] \geq \Phi(t) + p\,|B(t)| - p^2\Delta \cdot |B(t)|$$

Replacing p by $\frac{1}{2\Delta}$ we infer: $\mathbb{E}[\Phi(t+1)|X(t)] \geq \Phi(t) + \frac{|B(t)|}{4\Delta}$.

Let N be the random time when an equilibrium is reached in our process. If $t \geq N$, all vertices remain still. Hence,

$$\begin{aligned}\mathbb{E}[\Phi(t+1) - \Phi(t)] &= \mathbb{E}[(\Phi(t+1) - \Phi(t))\mathbb{1}_{N>t}] + \mathbb{E}[(\Phi(t+1) - \Phi(t))\mathbb{1}_{t\geq N}] \\ &\geq \mathbb{E}[\mathbb{E}[\phi(t+1) - \phi(t)|X(t)]\mathbb{1}_{N>t}] + 0 \\ &\geq \tfrac{1}{4\Delta}\mathbb{P}(N > t)\end{aligned}$$

Summing this from $t = 0$ to n yields $\phi(n+1) - \phi(0) \geq \frac{1}{4\Delta}\sum_{i=1}^{n+1}\mathbb{P}(N > i)$. Hence, taking limits on both sides when n grows

$$\phi(N) - \phi(0) \geq \frac{1}{4\Delta}\sum_{i=1}^{\infty}\mathbb{P}(N > i), \text{ i.e, } \phi(N) - \phi(0) \geq \frac{1}{4\Delta}\mathbb{E}[N]$$

whence we deduce $\mathbb{E}[N] \leq 4\Delta\phi(N) \leq 4\Delta\,|E|$ as announced. □

Proof of Theorem 2

Here a strategy profile may be simply described by the number $x(t)$ of vertices on the largest side at time t, hence $x(t) \geq \frac{n}{2}$. We use a more convenient potential function, namely $\Phi(x) = (x - \frac{n}{2})^2$, which we try to minimize.

Lemma 1 *Let* $t_0 = \lceil \log\log \frac{n^2}{4} \rceil + 4$. *For all* $t \geq 0$ *and* $x \in \{\frac{n}{2}, \cdots, n\}$ *we have*

$$\mathbb{P}\left(x(t_0 + t) \textit{ is a locally maximal cut } |x(t) = x\right) \geq 1/2$$

Proof Let $M(t)$ the random variable corresponding to the number of agents which move at time t. Then

$$\begin{aligned}\mathbb{E}[\Phi(t+1)|x(t) = x] &= \sum_{k=0}^{x}\mathbb{P}(M(t) = k|x(t) = x)\Phi(x-k) \\ &= \sum_{k=0}^{x}\mathbb{P}(M = k|x(t) = x)\left[(x - \frac{n}{2})^2 - 2(x - \frac{n}{2})k + k^2\right] \\ &= \Phi(x) + (n - 2x)\mathbb{E}[M|x(t) = x] + \mathbb{E}[M^2|x(t) = x] \quad (2)\end{aligned}$$

Conditional on $x(t) = x$, $M(t)$ is a binomial random variable with parameter p, hence

$$\mathbb{E}[M|x(t) = x] = px$$

and

$$\mathbb{E}[M^2|x(t)=x] = Var(M|x(t)=x) - \mathbb{E}[M|x(t)=x]^2 = xp(1-p) - (xp)^2$$

which gives in (2) when p is also replaced by $1 - \frac{n-1}{2(x-1)}$:

$$\begin{aligned}
\mathbb{E}[\Phi(x(t+1))|x(t)=x] &= \Phi(x) - \frac{x}{x-1}\left(\frac{n}{2} - x + \frac{1}{2}\right)^2 \\
&= \Phi(x) - \frac{x}{x-1}\left(\frac{1}{2} - \sqrt{\Phi(x)}\right)^2 \\
&\leq \Phi(x) - \left(\frac{1}{2} - \sqrt{\Phi(x)}\right)^2 \\
&\leq \sqrt{\Phi(x)} - \frac{1}{4}
\end{aligned}$$

Since this is true for all x, we deduce that

$$\begin{aligned}
\mathbb{E}[\Phi(x(t+1))] &\leq \mathbb{E}\left[\sqrt{\Phi(x(t))}\right] - \frac{1}{4} \\
&\leq \sqrt{\mathbb{E}[\Phi(x(t))]} - \frac{1}{4}, \qquad (3)
\end{aligned}$$

where the last inequality follows from Jensen's inequality.

From (3) we deduce that

$$\mathbb{E}[\Phi(x(t + \lceil \log\log \frac{n^2}{4} \rceil)|x(t)=x] \leq \Phi(x)^{2^{-\log\log \frac{n^2}{4}}} \leq 2$$

since $\Phi(x) \leq \frac{n^2}{4}$. Four more iterations of (3) give

$$\mathbb{E}[\Phi(x(t+t_0)|x(t)=x] \leq \frac{1}{2}$$

from which the claim follows by Markov's inequality, since a locally maximum cut is reached at a time t if $\Phi(t) < 1$. □

Now according to Lemma 1, for each new run of t_0 steps, the probability to reach a maximal cut is at least $\frac{1}{2}$, hence the expected time to reach such a cut is at most $\frac{t_0}{1-\frac{1}{2}} = 2t_0$.

References

1. Adolphs C, Berenbrink P (2012) Distributed selfish load balancing with weights and speeds. In: Proceedings of the 2012 ACM symposium on principles of, distributed computing, pp 135–144
2. Berenbrink P, Friedetzky T, Hajirasouliha I, Hu Z (2012) Convergence to equilibria in distributed, selfish reallocation processes with weighted tasks. Algorithmica 62:767–786
3. Chien S, Sinclair A (2007) Convergence to approximate nash equilibria in congestion games. In: Proceedings of SODA, pp 169–178
4. Christodoulou G, Mirrokni V, Sidiropoulos A (2012) Convergence and approximation in potential games. Theor Comput Sci 438:13–27
5. Even-Dar E, Kesselman A, Mansour Y (2007) Convergence time to nash equilibrium in load balancing. ACM Trans Algorithms 3(3):32
6. Fabrikant A, Papadimitriou C, Talwar K (2004) The complexity of pure nash equilibria. In: Proceedings of STOCS, ACM Press, pp 604–612
7. Goemans M, Williamson D (1995) Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. J ACM 42(6):1115–1145
8. Goldberg P (2004) Bounds for the convergence rate of randomized local search in a multiplayer load-balancing game. In: Proceedings of the twenty-third annual ACM symposium on principles of distributed computing, pp 131–140
9. Gourvs L, Monnot J (2010) The max k-cut game and its strong equilibria. In: Proceedings of TAMC, Springer, pp 234–246
10. Khot S, Kindler G, Mossel E (2004) Optimal inapproximability results for max-cut and other 2-variable CSPs. In: Proceedings of FOCS, pp 146–154
11. Kleinberg J, Tardos E (2005) Algorithm design. Addison-Wesley Longman Publishing Co., Boston
12. Monderer D, Shapley LS (1996) Potential games. Games Econ Behav 14(1):124–143
13. Rosenthal RW (1973) A class of games possessing pure-strategy Nash equilibria. Int J Game Theory 2:65–67
14. Schaffer A, Yannakakis M (1991) Simple local search problems that are hard to solve. SIAM J Comput 20(1):56–87

Part II
Analysis, Modelling and Optimisation

Distributed Binary Consensus in Dynamic Networks

Arta Babaee and Moez Draief

Abstract Motivated by the distributed binary interval consensus and the results on its convergence time we propose a distributed binary consensus algorithm which targets the shortfall of consensus algorithms when it comes to dynamic networks. We show that using our proposed algorithm nodes can join and leave the network at any time and the consensus result would always stay correct i.e. the consensus would always be based on the majority of the nodes which are currently present in the network. We then analyse our algorithm for the case of complete graphs and prove that the extra time it takes for nodes to implement our algorithm (to cope with the dynamic setting) does not depend on the size of the network and only depends on the voting margin. Our results are especially of interest in wireless sensor networks where nodes leave or join the network.

1 Introduction

In a network of interest there is not always a control centre available that can manage the data gathered from nodes. Moreover, gathering the data along with managing and processing might consume huge amount of time and energy which is not always feasible. In many such networks, the aim of the nodes is to calculate a function dependent on the initial (or current) values of nodes through what is often called distributed computations (i.e. nodes carrying out some internal computations and contacting each other in an iterative manner).

A. Babaee (✉) · M. Draief
Intelligent Systems and Networks Group, Department of Electrical and Electronic Engineering, Imperial College London, London, UK
e-mail: ab3608@imperial.ac.uk

M. Draief
e-mail: mmd@imperial.ac.uk

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_6,

A problem of interest in the context of distributed algorithms is the *binary consensus problem* [1–3] with which the nodes try to come to an agreement on one of two available choices based on the opinion of the majority (say 0 and 1). If the number of choices is more than two this becomes a *multivalued* consensus problem (e.g. [4]).

It was in [2] that for the first time a distributed binary consensus algorithm was introduced that converged to the right result with almost sure probability. This was achieved by adding two intermediate states (in addition to zero and one) and also using a two way communication between nodes. In other words, the consensus would always be correct at the expense of using four communication and memory states instead of two. In [5], this type of binary consensus algorithm was investigated in terms of convergence speed and in [6] it was optimized with respect to the convergence time. Here, we call this type of binary consensus *binary interval consensus* as it is called in [5].

In a real network nodes might have to leave or join the network. In such a network, reaching the *correct* consensus can become problematic. By correct consensus we mean each node has to decide which one of the states was initially held by the majority of the nodes present in the network without being affected by the decision of the nodes which have left the network. Clearly, this decision might have to change depending on the state of the nodes which leave or join the network frequently.

Using the conventional distributed algorithms, all nodes have to restart the consensus algorithm after each time a node joins or leaves the network. This is because normally after running the distributed algorithm, all the nodes change their initial value to the final value. For instance, using the algorithms in [7] and [3], after reaching the consensus all the nodes are either in state 1 or 0.

Needless to say, restarting the algorithm every time a node joins or leaves the network will be time and energy consuming. Moreover, consensus algorithms in [7] and [3] are not error-free.

Motivated by interval consensus algorithm, we suggest a framework for correct majority consensus in dynamic networks. More specifically, we aim to find a consensus algorithm in which nodes can join or leave the network at any time without causing any errors in the final result. Obviously, the final majority will be amongst the nodes which are present in the network.

The structure of this paper is as follows. In Sect. 2 we give an overview of binary interval consensus which is our base algorithm for dynamic consensus. We also mention the result regarding its convergence time in [5]. In Sect. 3 we present our algorithm which utilizes binary interval consensus for dynamic networks. Although our algorithm works for all types of graphs we find its implications on complete graphs using a direct analysis in the same section. We conclude in Sect. 4.

2 Binary Interval Consensus

Here, we give an overview of binary interval consensus algorithm in [2].

2.1 Algorithm

With binary interval consensus each node can have four states. These states are denoted by $0, 0.5^-, 0.5^+$, and 1 where $0 < 0.5^- < 0.5^+ < 1$. Here, being in state 0 or 0.5^- means that a node believes the initial majority was 0 or equivalently the average values of the nodes is between 0.5 and 0.

Consider the case where each pair of nodes (i, j) interact at Poisson rate q_{ij}, the rate matrix Q as defined in [5] is then as follows

$$Q(i,j) = \begin{cases} q_{ii} = -\sum_{l \in V} q_{il} & i = j \\ q_{ij} & i \neq j \end{cases} \tag{1}$$

where V is the set of vertices. Note that this is an asynchronous framework in which at each time step a node can only contact one of its neighbours. Now consider the interaction between any pair of nodes (i, j). At each contact of the two nodes i, j their states get updated using the following:

$$\begin{aligned} (0, 0.5^-) &\to (0.5^-, 0), \\ (0, 0.5^+) &\to (0.5^-, 0), \\ (0, 1) &\to (0.5^+, 0.5^-), \\ (0.5^-, 0.5^+) &\to (0.5^+, 0.5^-), \\ (0.5^-, 1) &\to (1, 0.5^+), \\ (0.5^+, 1) &\to (1, 0.5^+), \\ (s, s) &\to (s, s), \text{ for } s = 0, 0.5^-, 0.5^+, 1 \end{aligned}$$

Using this mapping it is mentioned in [2] that binary interval consensus has the following properties:

Define X_i as the state of node i. Following the interaction of nodes i, j at time t,

- **Mixing**: It can be seen that if $X_i(t) \leq X_j(t)$ then $X_i(t+1) \geq X_j(t+1)$.
- **Contraction**: $X_i(t+1)$ and $X_j(t+1)$ are either equal or one point away from each other (in the sequence $0, 0.5^-, 0.5^+, 1$).
- **Conservation**: Finally,

$$X_i(t+1) + X_j(t+1) = X_i(t) + X_j(t).$$

The last property is of our interest and basically means that the average is preserved throughout the consensus process.

The consensus evolves as follows. The number of nodes in both states 1 and 0 will decrease by 1 only when a node in state 1 interacts with a node in state 0. We denote the set of nodes in state i at time t by $S_i(t)$, also $|S_i| \equiv |S_i(0)|$ $(i = 0, 1)$. If nodes with state 0 are the majority, and α denotes the fraction of nodes in state 0 at the beginning of the process, $\frac{1}{2} < \alpha \leq 1$, and therefore $|S_0| = \alpha n$ and $|S_1| = (1 - \alpha)n$. As the number of nodes in state 1 and 0 decreases at the encounters between 0 and 1, finally there will be no nodes in state 1 left in the network. Also, the number of nodes in state 0 will become $|S_0| - |S_1|$ at the end of this part of the consensus process. There will be only nodes in state 0.5^+, 0.5^-, and 0 left. We denote this phase of the process by *Phase 1*.

Next, the number of nodes in state 0.5^+ will decrease when they interact with nodes in state 0 and consequently after some time the nodes in state 0.5^+ will also disappear and only nodes with state 0 or 0.5^- will remain. We denote this part of the process by *Phase 2*. At the end of Phase 2 the algorithm reaches the consensus. This means that all the nodes agree that the average is in the interval [0,0.5) which indicates that nodes with state 0 initially had the majority.

Note that throughout the consensus process the sum of the values of nodes always stays the same. For example if five nodes start with initial states (0, 1, 0, 0, 1) in the end they will have the states (0.5^-, 0.5^-, 0.5^-, 0.5^-, 0). While the result vector means that all the nodes agree that the average value is between zero and one and the initial majority is zero, the sum of the values always stays 2.

2.2 Convergence Time

In [5], the upper bounds for the expected time for each of these phases have been derived in terms of the eigenvalues of a set of matrices that depend on Q. If S is considered as a non-empty subset of V, Q_S is defined as:

$$Q_S(i,j) = \begin{cases} -\sum_{l \in V} q_{il} & i = j \\ q_{ij} & i \notin S, j \neq i \\ 0 & i \in S, j \neq i \end{cases} \tag{2}$$

The following lemma is then derived:

Lemma 1 *For any finite graph G, there exists $\delta(G, \alpha) > 0$ such that, for any non-empty subset of vertices S ($|S| < n$), if $\lambda_1(Q_S)$ is the largest eigenvalue of Q_S, then it satisfies*

$$\delta(G, \alpha) = \min_{S \subset V, |S| = (2\alpha - 1)n} |\lambda_1(Q_S)|. \tag{3}$$

Note that using this definition, for all non-empty set S, $\delta(G, \alpha) > 0$ because $\lambda_1(Q_S) < 0$.

Theorem 1 *If T is considered as the time of convergence (i.e. the time it takes for nodes in states 1 and 0.5^+ to deplete), it will be bounded as follows,*

$$\mathbb{E}(T) \leq \frac{2}{\delta(G,\alpha)}(\log n + 1). \tag{4}$$

The convergence time directly depends on $\delta(G, \alpha)$.

3 Binary Interval Consensus in Dynamic Networks

Here by dynamic network we mean a network in which nodes can join or migrate from the network. For joining nodes we recommend using binary interval consensus without any changes. For the case of departing nodes we introduce an additional procedure which nodes should run before they can leave the network. This additional procedure would then guarantee consensus to the right result.

3.1 Joining Nodes

We claim the following lemma for a dynamic network where new nodes can join the consensus process,

Lemma 2 *When nodes use the binary interval consensus if new nodes join the network at any given time k with any value (zero or one), the consensus will shift to the right result based on the majority at time k.*

Proof The proof follows by the fact that the binary interval consensus converges to the right result with almost sure probability (by the conservation property). Of all the ways that this consensus can be reached amongst n nodes, one way is that certain nodes will not be contacted by any neighbours or initiate contact with their neighbours until some time step k. We call the number of these specific nodes n_1 and the number of others n_2 (i.e. $n - n_1$). This dynamic is exactly the same as if the consensus starts in a network with size n_2 and then n_1 nodes join the network at any given time k with states 0 or 1.

In other words, using binary interval consensus at each time step the sum of all nodes and consequently the average stays the same. By joining new nodes the sum of all the values of nodes will be as if the new nodes had joined the network before the start of the consensus algorithm and this is what guarantees the right consensus at any time step.

3.2 Departing Nodes

Leaving the network however cannot be dealt with using the same approach. For example, if a node starts with value 1 and wants to leave the network while it has changed its state to 0, the sum of the values of the nodes will not change. This means that the departure of the node with initial value 1 (with current state at 0) not only will not decrease the average but will increase it. Therefore, we need to set up an algorithm for nodes which wish to leave the network to implement before their departure.

If a node is leaving the network it should deduct its value from the sum of the values of all nodes before departure and for this, it needs to remember what its initial value was before the start of the consensus. Consequently, we need one bit of memory for each node that might want to leave the network during or after the consensus process. We then propose the following procedure.

Consider node i with the initial value 1 (respectively 0) which wants to leave the network. Its strategy will be as follows. Note that in the following rules, the update procedure are the same as before for other nodes. Only the node which is leaving should follow these rules,

- If its current state is 1 (respectively 0) it can leave the network right away.
- If its current state is either 0.5^- or 0.5^+ it will set its state to 0 (respectively 1) and then wait to make contact with any node in state 0.5^-, 0.5^+, or 1 (respectively 0). It then leaves the network.
- If its current state is 0 (respectively 1). It will make contact with any node in state 0.5^-, 0.5^+ or 1 (respectively 0) without updating its state, maintaining its state at 0 (respectively 1). It will then wait to contact any node in state 0.5^-, 0.5^+, or 1 (respectively 0). It then leaves.

The following lemma is then true.

Lemma 3 *The mentioned procedure will guarantee the correct consensus at any time after the departure of nodes.*

Proof The proof follows by the fact that using the above procedure the sum of the values of all the nodes present in the network will be the same as the sum of their initial values and hence the consensus will remain correct at all times.

For instance, consider the following two vectors,

$$x_i = (0, 0, 1, 1, 0, 0),$$
$$x = (0.5^-, 0.5^-, 0, 0.5^-, 0, 0.5^-).$$

where x_i denotes the initial values of nodes at the start of the consensus process and x denotes their current value. As it can be seen the consensus process has already ended and all the nodes have chosen 0 as the majority state. Now if node 3 with initial value 1 wants to leave the network it has to reduce the sum of the values by 1. However, as the current state of node 3 is 0, it needs to contact one of the

nodes 1, 2, 4, or 6 which are at state 0.5^-. Let us assume that it waits and finally contacts node 4. The vector of states becomes,

$$x = (0.5^-, 0.5^-, 0, 0, 0, 0.5^-).$$

Note that node 3 has kept its state at 0. It now needs to contact any of the nodes 1, 2 or 6. If it contacts node 6, the following will be the vector of states,

$$x = (0.5^-, 0.5^-, 0, 0, 0, 0),$$

and after node 3 leaves the network the vector of states becomes,

$$x' = (0.5^-, 0.5^-, 0, 0, 0).$$

Therefore the sum of the values will be 1 as there is only one node present in the network with an initial value 1.

Using this framework for dynamic consensus there is no need to restart the process of reaching consensus when nodes join or leave. Furthermore, nodes that do not want to leave or join the network will not need to be aware of the dynamics. They will continue implementing the same algorithm that they had already been running. Only the nodes that want to leave the network have to implement an additional procedure before their departure.

3.3 The Expected Time it Takes before a Node can Leave

As mentioned, nodes which want to leave the network have to make contact with at most two nodes in other states, before they can leave. Therefore it would be useful to know how much this would take and whether it is feasible for nodes to implement this extra procedure. We denote the time to implement the departure procedure by T_{BD}. To find T_{BD} we have to consider several factors such as when the node is leaving (whether it is at the start, during, or after the consensus), what its state is ($1, 0.5^-, 0.5^+$, or 0 which will then determine the number of contacts needed), and whether it is a node in minority or majority state.

Here, we consider the case where a node leaves a complete network after the consensus has been reached. More specifically, consider a network where nodes decide to have a consensus on a specific matter at specific time steps, where each consensus C_k starts at t_{s_k} and finishes at t_{f_k} and there is always a time gap between C_ks (i.e. $t_{s_k+1} > t_{f_k}$). In this scenario, nodes leave at intervals Y_ks, where $Y_k = [t_{f_k}, t_{s_{k+1}})$. As before, we consider 0s as the majority and to consider the worst case we assume that the node which is leaving is in state 0 and initially was 1 (therefore needs at least two contacts with other nodes before it can leave).

Following the mentioned conditions, as all the nodes are either in state 0 or 0.5^-, the departing node has to make contact with two 0.5^-s before it can leave the

network. The expected time for each node before it can leave the network will then be the following:

Lemma 4 *For a departing node in a complete graph using dynamic binary consensus algorithm, the expected time to implement departure procedure,* $\mathbb{E}(T_{BD})$ *is as follows:*

$$\mathbb{E}(T_{BD}) = \frac{n-1}{2(1-\alpha)n} + \frac{n-1}{2(1-\alpha)n-1}.$$

Proof To find the expected time it takes before departure, we use the analysis in [5] for direct computation of the expected time of the depletion of nodes in state 1 in complete graphs (we denoted this by Phase 1 in Sect. 2.1).

Consider the time of convergence for complete graphs. If τ_i is considered as the time of the ith contact of a node in state 0 and 1 ($i = 1, \ldots, |S_1|$), it can be seen that the number of nodes in state 1 for any time $t \geq \tau_{|S_1|} = T_1$ is zero. Furthermore, if $\tau_i \leq t < \tau_{i+1}$, $|S_1(t)| = |S_1| - i$ and $|S_0(t)| = |S_0| - i$. Also, at times τ_i,

$$(|S_0| - i + 1, |S_1| - i + 1) \rightarrow (|S_0| - i, |S_1| - i)$$

It is then derived that if $L_i = \tau_{i+1} - \tau_i$, L_i will be an exponential random variable with the following parameter,

$$\beta_i = (|S_0| - i)(|S_1| - i)/(n-1), \tag{5}$$

where $i = 0, ..., |S_1| - 1$.

Using (5) and the fact that $\mathbb{E}(T_1)$ (i.e. the expected time of Phase 1) can be written as $\sum_{i=0}^{|S_1|-1} \beta_i^{-1}$ would then lead to the following,

$$\mathbb{E}(T_1) = \frac{n-1}{|S_0| - |S_1|}(H_{|S_1|} + H_{|S_0|-|S_1|} - H_{|S_0|}), \tag{6}$$

where $H_k = \sum_{i=1}^{k} \frac{1}{i}$. Accordingly,

$$\mathbb{E}(T_1) = \frac{1}{2\alpha - 1}\log(n) + O(1).$$

Applying (6) to our framework will yield the expected time for the node in state 0 to make the first contact with a node in state 0.5^-, $\mathbb{E}(T_{BD_1})$ as follows:

$$\mathbb{E}(T_{BD_1}) = \frac{n-1}{|S_{0.5^-}| - 1}(H_1 + H_{|S_{0.5^-}|-1} - H_{|S_{0.5^-}|}). \tag{7}$$

Using the fact that after finishing the consensus $S_{|0.5^-|} = 2(1-\alpha)n$ and (7), $\mathbb{E}(T_{BD_1})$ will be given by:

$$\mathbb{E}(T_{BD_1}) = \frac{n-1}{2(1-\alpha)n}. \tag{8}$$

Similarly, $\mathbb{E}(T_{BD_2})$ (the expected time for the departing node to make the second contact) will be given by:

$$\mathbb{E}(T_{BD_2}) = \frac{n-1}{2(1-\alpha)n-1}. \tag{9}$$

Note that after the first contact the number of 0.5^{-}s will be reduced by 1.
Finally $\mathbb{E}(T_{BD})$ will be given by:

$$\begin{aligned}\mathbb{E}(T_{BD}) &= \mathbb{E}(T_{BD_1}) + \mathbb{E}(T_{BD_2}) \\ &= \frac{n-1}{2(1-\alpha)n} + \frac{n-1}{2(1-\alpha)n-1}.\end{aligned} \tag{10}$$

Equation (10) shows that when n is large the time it takes for the node to leave the network (i.e. $\frac{1}{1-\alpha}$) will not grow with the size of the network.

4 Conclusion

Based on the work of the authors in [2] we proposed an algorithm that deals with dynamics of a distributed binary consensus process in which nodes can join and leave the network. We then showed that using this algorithm the consensus process can always converge to the majority of the present nodes even when nodes join or leave the network. Note that the proposed algorithm is not restricted to the specific graph we have considered. However, the bound we have derived is for complete graphs and further work is required to find a general bound.

Acknowledgments Moez Draief is supported by QNRF through NPRP grant number 09-1150-2-148.

References

1. Kashyap A, Basar T, Srikant R (2007) Quantized consensus. Automatica 43.
2. Bénézit F, Thiran P, Vetterli M (2009) Interval consensus: from quantized gossip to voting. In: Proceedings of the (2009) IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '09. IEEE Computer Society Washington, DC, USA, pp 3661–3664
3. Perron E, Vasudevan D, Vojnović M (2009) Using three states for binary consensus on complete graphs. In: INFOCOM 2009, IEEE, pp 2527–2535.
4. Babaee A, Draief M (2013) Distributed multivalued consensus. Comput J.
5. Draief M, Vojnović M (2012) Convergence speed of binary interval consensus. SIAM J Control Optim 50(3):1087–1109
6. Babaee A, Draief M (2013) Optimization of binary interval consensus. Comput Inf Sci II I:281
7. Hassin Y, Peleg D (2002) Distributed probabilistic polling and applications to proportionate agreement. Inf Comput 171:248–268

Computing Bounds of the MTTF for a Set of Markov Chains

F. Aït-Salaht, J. M. Fourneau and N. Pekergin

Abstract We present an algorithm to find some upper and lower bounds of the Mean Time To Failure (MTTF) for a set of absorbing Discrete Time Markov Chains (DTMC). We first present a link between the MTTF of an absorbing chain and the steady-state distribution of an ergodic DTMC derived from the absorbing one. The proposed algorithm is based on the polyhedral theory developed by Courtois and Semal and on a new iterative algorithm which gives bounds of the steady-state distribution of the associated ergodic DTMC at each iteration.

1 Introduction

Finite DTMC models provide a very efficient technique for the study of dynamical systems. However, in many engineering problems, it is still hard to give the precise parameters to describe the chain: we need the exact transition probabilities between the states of the chain. In many cases, we only know an interval for these transition probabilities. This is equivalent to state that the underlying model M is inside of a set of chains described by an entry-wise lower bounding matrix **L** and an entry-wise upper bounding matrix **U**. Many results have been proposed to find bounds on the steady-state distribution when the chain is ergodic (see for instance the polyhedral theory developedby Courtois and Semal [4] and applied by Franceschinis and

F. Ait-Salaht (✉) · J. M. Fourneau
PRiSM, Université de Versailles-Saint-Quentin, CNRS UMR 8144, Boulogne, France
e-mail: safa@prism.uvsq.fr

J. M. Fourneau
e-mail: jmf@prism.uvsq.fr

N. Pekergin
LACL, Université Paris Est, Paris, France
e-mail: nihal.pekergin@u-pec.fr

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_7,

Muntz [7] for reliability problem and more recently by Buchholz [2] or ourselves for a faster algorithm [1]). In [8], the authors give the algorithms for the efficient model checking of such models.

Here we investigate a new problem. We assume that the chain is absorbing and the transition probabilities are imprecise (partially known). More formally, we assume that the chain belongs to a set of absorbing Markov chains having the same set of absorbing states, $\mathscr{A}$. We will denote by $\preceq$ the element-wise comparison of two vectors (or matrices). We also assume that the matrix of the chain $\mathbf{M}$ satisfies $\mathbf{L} \preceq \mathbf{M} \preceq \mathbf{U}$. We study how to find bounds for the MTTF of a such model $\mathbf{M}$. We show first the relation between the MTTF of an absorbing chain and an associated ergodic DTMC built from the absorbing one. We then use the polyhedral theory and the same arguments to construct bounds on MTTF. We use the new numerical technique given in [3] to solve the steady-state distribution of each matrix considered in the polyhedral approach we have already proposed in [1] to study imprecise DTMCs. This algorithm that we apply after some on imprecise absorbing provides at each iteration upper and lower bounds of the MTTF. Therefore, we obtain bounds at the first iteration and at each iteration the bounds are improved. Our algorithm is also numerically stable as it is only based on the product of non negative vectors and matrices. In the following of the paper, we describe how to link the MTTF of an absorbing DTMC with the steady-state distribution of an ergodic associated DTMC. Then in Sect. 3, we briefly introduce the $I\nabla L$ and $I\nabla U$ Algorithms [3]. In Sect. 4, we present how we can combine all these results to derive a bound for the MTTF for a set of Markov chains. We illustrate the approach with some numerical results.

2 Mean Time To Failure

Let us first begin with some notations. All vectors are row vectors, $\mathbf{e}_i$ is a row vector with 1 in position i and 0 elsewhere, the vector with all entries equal to 0 is denoted by $\mathbf{0}$ and $\mathbf{Id}$ is used for the identity matrix. Finally, x^t denotes the transposed vector of x and $||x||$ is the sum of the elements of vector x.

We consider a model defined by an absorbing DTMC noted by $\mathbf{M}$. We assume that we have several absorbing states and no recurrent class. We want to compute the mean time to reach an absorbing state. We show how to transform this initial problem to the construction of an ergodic DTMC and the analysis of its steady-state distribution. More formally, we suppose that we use the following matrix decomposition: $\mathbf{M} = \left[\begin{array}{c|c} \mathbf{Id} & \mathbf{0} \\ \hline \mathbf{R} & \mathbf{Q} \end{array}\right]$ once we have organised the state space to have the absorbing states (i.e. in set $\mathscr{A}$) before the transient states (i.e. in set $\mathscr{T}$).

The absorbing DTMCs are studied through their fundamental matrices [9]. By assuming that there is no recurrent class, we have the following well-known results:

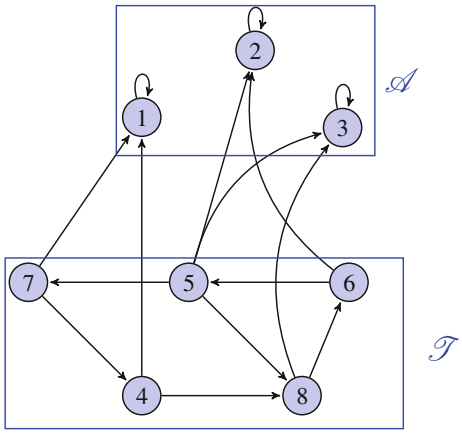

Fig. 1 Transition graph of an absorbing Markov Chain.

- The fundamental matrix $\mathbf{F} = (\mathbf{Id} - \mathbf{Q})^{-1}$ exists.
- We assume that there exist several absorbing points. The probability to be absorbed in state j knowing that the initial state is i is equal to $(\mathbf{F}.\mathbf{R})[i, j]$.
- The mean time before absorption knowing that initial state is i is $(\mathbf{F}.e^t)[i]$ (Fig. 1).

We propose to compute the mean absorbing time, called also MTTF through the steady-state probabilities of an ergodic DTMC built by the underlying absorbing one. We assume that the directed graph of the transient states $\mathscr{T}$ is strongly connected. Note that it does not imply that there is a recurrent class among these states.

Let i be an arbitrary non absorbing state. We consider a new matrix built as follows. First, we aggregate all the absorbing states into one state which is the first one of the state space. Thus, matrix $\mathbf{R}$ is also summed up into a vector r^t. Second, we add a loop on state 1 with probability 0.5. Third, we modify the first row: we add a vector denoted as p_i whose entries are all equal to zero except entry i which is 0.5. Finally, the built stochastic matrix, $\mathbf{M}_i$ is as follows (Fig. 2):

$$\mathbf{M}_i = \left[\begin{array}{c|c} 1/2 & p_i \\ \hline r^t & \mathbf{Q} \end{array}\right]. \tag{1}$$

In the following, set $\mathscr{F}$ will denote the state space of $\mathbf{M}_i$. It contains set $\mathscr{T}$ and one state which represents the aggregation of the absorbing states of $\mathbf{M}$.

Property 1 *Matrix* $\mathbf{M}_i$ *is ergodic. Therefore the steady-state distribution of* $\mathbf{M}_i$ *(denoted as* π_i*) exists.*

Proof The chain is finite. The graph of the states in $\mathscr{T}$ is strongly connected and there is a directed edge between state 1 and state i and between a state in $\mathscr{T}$ and state 1 because vector r is not zero. Therefore, the graph on the whole state space is

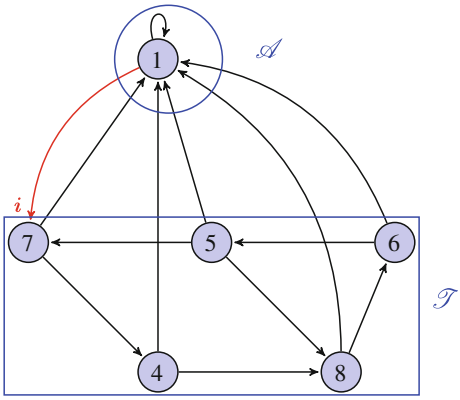

Fig. 2 Transition graph of the new Markov Chain to obtain the time before being absorbed knowing that we begin at state 7.

strongly connected. Finally, as there is a loop at state 1, the chain is aperiodic. Thus, the chain is ergodic and the steady-state distribution exists.

The question is to find a relation between the MTTF of $\mathbf{M}$ and π_i computed from matrix $\mathbf{M}_i$. Let $E[T_i]$ be the mean time before the absorption knowing the initial state is i for the absorbing chain $\mathbf{M}$. On the other hand, for the ergodic DTMC $\mathbf{M}_i$, we know that $1/\pi_i[1]$ is the mean time between two visits to state 1. To compute it, we condition on the first transition out of state 1. We have two possible transitions: a loop in state 1 with probability 1/2, so the time between two visits is 1 and a transition into state i with probability 1/2, so the time between two visits is $(1 + E[T_i])$. Therefore:

$$\frac{1}{\pi_i[1]} = \frac{(1 + E[T_i])}{2} + \frac{1}{2}.$$

Finally, we obtain $E[T_i]$:

$$E[T_i] = \frac{2}{\pi_i[1]} - 2. \tag{2}$$

We have to find bounds on π_i when the matrix is specified by matrices $\mathbf{L}$ and $\mathbf{U}$. To do this, we combine Muntz's approach for imprecise DTMCs and the iterative algorithm presented in [1]. The bounds on $\pi_i[1]$ will then provide bounds on MTTF by using Eq. 2.

Let us first begin with the algorithm for exact calculation of $\pi_i[1]$ thus $E[T_i]$ before proceeding with imprecise Markov chains.

3 Algorithms Based on Monotone Sequences

Let $\mathbf{P}$ be a finite stochastic matrix. We assume that $\mathbf{P}$ is ergodic. We first introduce some quantities easily computed from $\mathbf{P}$.

Definition 1 Set $\nabla_P[j] = \mathbf{min}_i \mathbf{P}[i, j]$ and $\triangle_P[j] = \mathbf{max}_i \mathbf{P}[i, j]$. Remark that ∇_P may equal to vector $\mathbf{0}$ but Δ_P is positive as the chain is irreducible.

Bušić and Fourneau [3] proposed two iterative algorithms based on simple (**max**, +) (resp. (**min**,+)) properties, called $I\nabla L$ (resp. $I\nabla U$) which provide at each iteration a new lower (resp. upper) bound $x^{(k)}$ (resp. $y^{(k)}$) of the steady state distribution of $\mathbf{P}$.

Algorithm 1 Algorithm Iterate ∇ Lower Bound (I∇L)

Require: $a \preceq \pi$, $b \preceq \nabla_{\mathbf{P}}$ and $b \neq \mathbf{0}$.
Ensure: Successive values of $x^{(k)}$.

1: $x^{(0)} = a$.
2: **repeat**
3: $x^{(k+1)} = \mathbf{max}\left\{x^{(k)}, x^{(k)}\mathbf{P} + b(1 - ||x^{(k)}||)\right\}$.
4: **until** $1 - ||x^{(k)}|| < \varepsilon$.

Theorem 1 *Let* $\mathbf{P}$ *be an irreducible and aperiodic stochastic matrix with steady state probability distribution* π. *If* $\nabla_P \neq \mathbf{0}$, *Algorithm* $I\nabla L$ *provides at each iteration lower bounds for all components of* π *and converges to* π *for any value of the parameters* a *and* b *such that* $a \preceq \pi$, $b \preceq \nabla_P$ *and* $b \neq \mathbf{0}$.

One can check that the conditions on the initialisation part of the algorithm require that $||\nabla_{\mathbf{P}}|| > 0$. Similarly, we have proved another algorithm (called $I\nabla U$) to compute a decreasing sequence $y^{(k)}$ of iterative upper bounds. It is based on an initialization with vector $\triangle_P$ and an iteration with operator **min** instead of **max**. Note that combining both theorems we obtain a proved envelope for all the components of vector π. It is also proved in [3] that the norm of the envelope converges to zero faster than a geometric with rate $(1 - ||b||)$. The algorithms have been implemented in a tool called XBorne [5]. These algorithms also have two important properties. First, under some technical conditions, an entry-wise bound on the stochastic matrices provides an entry-wise bound on the steady-state distribution (see [3]). Second, they deal with infinite matrix with some constraints on the associated directed graph [6].

Example 1 Let $\mathbf{P}$ be a stochastic matrix $\mathbf{P} = \begin{pmatrix} 0.6 & 0 & 0.2 & 0.2 & 0 \\ 0.4 & 0.2 & 0.1 & 0.2 & 0.1 \\ 0.2 & 0.1 & 0.2 & 0.3 & 0.2 \\ 0.2 & 0 & 0.2 & 0.3 & 0.3 \\ 0.1 & 0 & 0 & 0.4 & 0.5 \end{pmatrix}$.
We have $\nabla_{\mathbf{P}} = (0.1,\ 0,\ 0,\ 0.2,\ 0)$. For $\varepsilon = 10^{-5}$, algorithm I∇L with $a = b = \nabla_{\mathbf{P}}$ provides the following sequence of lower bounds for the probabilities.

k	1	2	3	4	5	$1 - \|\|x^{(k)}\|\|$
1	0.17	0	0.06	0.22	0.06	0.4900
3	0.2413	0.0102	0.1092	0.2546	0.1446	0.2401
7	0.2869	0.0169	0.1409	0.2826	0.2151	0.0576
11	0.2968	0.0183	0.1481	0.2897	0.2332	0.0139
21	0.2997	0.0188	0.1502	0.2920	0.2389	0.0004
31	0.2997	0.0188	0.1503	0.2921	0.2391	$1.1\ 10^{-5}$

4 Bounds of the MTTF

We now present how to deal with imprecise Markov chains. Franceschinis and Muntz [7] have proposed an approach for bounding steady-state availability. The theoretical background is based on Courtois and Semal polyhedral results on steady-state distribution [4]. We only present here a weak form of the theorem.

Theorem 2 *Given a lower bound* $\mathbf{L}$ *of the transition probability matrix of a given DTMC (let's assume that the chain has n states), we can compute a bounds for its steady-state probability vector* π. *In a first step one compute the steady-state solution of n DTMCs. Transition probability matrix* $\mathbf{L}^s$ *associated with the sth DTMC is obtained from sub-stochastic matrix* $\mathbf{L}$ *by increasing the elements of column s to make* $\mathbf{L}^s$ *stochastic. Let* π^s *be the steady state probability vector solution of the sth DTMC. The lower (resp. upper) bound on the steady state probability of state j is computed as* $\mathbf{min}_s \pi^s[j]$ *(resp.* $\mathbf{max}_s \pi^s[j]$*).*

$$\mathbf{min}_s\, \pi^s[j] \;\leq\; \pi[j] \;\leq\; \mathbf{max}_s\, \pi^s[j]. \tag{3}$$

We have showed in [1] how one can combine theorem 2 and $I\nabla L$ and $I\nabla U$ algorithms to prove new algorithms which provide at each iteration a new component-wise bounds on steady state distribution. To simplify the presentation, we only present the upper bound case (for the lower bound see [1]). The main idea behind the upper bounding algorithm is to compute first, for all $s \in \mathscr{F}$ an upper bound $Y^{(k),s}$ associated with matrix $\mathbf{L}^s$ with $I\nabla U$ algorithm. Then, we apply Muntz's result to deduce an upper bound on steady state distribution of π. This process is iterated until the stopping criterion is reached. The sequences $Y^{(k),s}$ converges faster than a geometric with rate $(1 - ||\nabla_{\mathbf{L}^s}||)$. Once all these sequence have converged, the **max** operator between distributions proved by the polyhedral theory does not change either.

Theorem 3 *Let* $\mathbf{L}$ *be an irreducible sub-stochastic matrix, algorithm 2 provides at each iteration k an element wise upper and lower bounds on the steady-state distribution of any ergodic matrix entry-wise larger than* $\mathbf{L}$.

For a proof and some arguments on complexity, see [1]. We combine all these results to obtain bounds on the MTTF. We consider an absorbing non-negative matrix $\mathbf{M}$ define by $\mathbf{L} \preceq \mathbf{M} \preceq \mathbf{U}$, such that each row sum is less than or equal to 1.

Algorithm 2 Algorithm Iterate ∇ Bounds for imprecise Markov chains

Require: $\forall s \in \mathscr{F}$, $\alpha[s] > 0$.
Ensure: Successive values of $Y^{(k)}$ and $X^{(k)}$.

1: $\forall s \in \mathscr{F}, \mathbf{L}^s = \mathbf{L} + \alpha^t e_s, c^s = \triangle_{\mathbf{L}^s}, b^s = \nabla_{\mathbf{L}^s}, Y^{(0),s} = c^s, X^{(0),s} = b^s$.
2: **repeat**
3: $\forall\, s \in \mathscr{F} Y^{(k+1),\,s} = \mathbf{min}\left\{Y^{(k),\,s}, Y^{(k),\,s}\mathbf{L}^s + b^s(1 - ||Y^{(k),\,s}||)\right\}$.
4: $Y^{(k+1)} = \mathbf{max}_s\left\{Y^{(k+1),s}\right\}$.
5: $\forall\, s \in \mathscr{F} X^{(k+1),\,s} = \mathbf{max}\left\{X^{(k),\,s}, X^{(k),\,s}\mathbf{L}^s + b^s(1 - ||X^{(k),\,s}||)\right\}$.
6: $X^{(k+1)} = \mathbf{min}_s\left\{X^{(k+1),s}\right\}$.
7: **until** $\sum_s(||Y^{(k),s}|| - 1) < \varepsilon$ and $\sum_s(1 - ||X^{(k),s}||) < \varepsilon$.

First, we consider a set of stochastic matrices $\mathscr{P} = \{\mathbf{L}^s \mid \mathbf{L}^s$ is an irreducible stochastic matrix and $\mathbf{L}^s \succeq \mathbf{L}\}$. s is a column index. $\mathbf{L}^s$ is matrix $\mathbf{L}$ where elements in the sth column have been increased as necessary to make the matrix stochastic. We first add the following assumption: **in the following we assume that** $r(k) > 0$ **for all** k. Therefore $\nabla(\mathbf{L}^s) > 0$ and we can apply the algorithms. For each matrix $\mathbf{M} \in \mathscr{P}$, we define transition matrix $\mathbf{M}_{i,\,s}$ as in Equation 1. We use Muntz's approach and the results in [1] about the previous algorithms. Finally, the steady-state distribution using the Algorithm 2 at iteration n is bounded by:

$$\mathbf{min}_s\{X^{(n),s}\} \leq \pi_i \leq \mathbf{max}_s\{Y^{(n),s}\}.$$

Then, the bounds for the average time $E[T_i]$ are:

$$E[\underline{T_i}] = \frac{2}{\mathbf{max}_s\{Y^{(n),s}\}[1]} - 2 \leq E[T_i] \leq \frac{2}{\mathbf{min}_s\{X^{(n),s}\}[1]} - 2 = E[\overline{T_i}].$$

And we conclude with the theorem which states the result and we give a small example.

Theorem 4 *Let* $\mathbf{L}$ *be a sub-stochastic matrix, Algorithm 2 provides at each iteration* k *a lower bound and an upper bound ot the MTTF of any absorbing matrix entry-wise larger than* $\mathbf{L}$ *and defined on the same set of transient states* $\mathscr{T}$.

Example 2 Consider absorbing matrix

$$\mathbf{L} = \left(\begin{array}{cc|cccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0.30 & 0.1 & 0.2 & 0.1 & 0.1 & 0.2 \\ 0.20 & 0.3 & 0.2 & 0.1 & 0 & 0.1 \\ 0.25 & 0.2 & 0.1 & 0.2 & 0.2 & 0 \\ 0.10 & 0 & 0.2 & 0.1 & 0.2 & 0.1 \end{array}\right).$$

Assume that we want to compute the expected time before being absorbed when we begin at state $i = 5$. First, we aggregate state 1 and 2, which are absorbing states in

L and we modify the transition between the first state and state $i = 5$. Note that state 5 is now the fourth state due to aggregation of state 1 and 2.

We obtain an irreducible sub-stochastic matrix

$$\left(\begin{array}{c|cccc} \boxed{0.5} & 0 & 0 & \boxed{0.5} & 0 \\ \hline 0.4 & 0.2 & 0.1 & 0.1 & 0.2 \\ 0.5 & 0.2 & 0.1 & 0 & 0.1 \\ 0.45 & 0.1 & 0.2 & 0.2 & 0 \\ 0.1 & 0.2 & 0.1 & 0.2 & 0.1 \end{array}\right).$$

Note that $r(k) > 0$ for all k. Thus, we can use the Nabla based algorithms as the first column of the matrix has all its entries positive. We then derive the five matrices which are in set $\mathscr{P}$:

$$M_{5,1} = \left(\begin{array}{c|cccc} 0.5 & 0 & 0 & 0.5 & 0 \\ \hline 0.4 & 0.2 & 0.1 & 0.1 & 0.2 \\ 0.6 & 0.2 & 0.1 & 0 & 0.1 \\ 0.5 & 0.1 & 0.2 & 0.2 & 0 \\ 0.4 & 0.2 & 0.1 & 0.2 & 0.1 \end{array}\right), \quad M_{5,2} = \left(\begin{array}{c|cccc} 0.5 & 0 & 0 & 0.5 & 0 \\ \hline 0.4 & 0.2 & 0.1 & 0.1 & 0.2 \\ 0.5 & 0.3 & 0.1 & 0 & 0.1 \\ 0.45 & 0.15 & 0.2 & 0.2 & 0 \\ 0.1 & 0.5 & 0.1 & 0.2 & 0.1 \end{array}\right),$$

$$M_{5,3} = \left(\begin{array}{c|cccc} 0.5 & 0 & 0 & 0.5 & 0 \\ \hline 0.4 & 0.2 & 0.1 & 0.1 & 0.2 \\ 0.5 & 0.2 & 0.2 & 0 & 0.1 \\ 0.45 & 0.1 & 0.25 & 0.2 & 0 \\ 0.1 & 0.2 & 0.4 & 0.2 & 0.1 \end{array}\right), \quad M_{5,4} = \left(\begin{array}{c|cccc} 0.5 & 0 & 0 & 0.5 & 0 \\ \hline 0.4 & 0.2 & 0.1 & 0.1 & 0.2 \\ 0.5 & 0.2 & 0.1 & 0.1 & 0.1 \\ 0.45 & 0.1 & 0.2 & 0.25 & 0 \\ 0.1 & 0.2 & 0.1 & 0.5 & 0.1 \end{array}\right),$$

$$M_{5,5} = \left(\begin{array}{c|cccc} 0.5 & 0 & 0 & 0.5 & 0 \\ \hline 0.4 & 0.2 & 0.1 & 0.1 & 0.2 \\ 0.5 & 0.2 & 0.1 & 0 & 0.2 \\ 0.45 & 0.1 & 0.2 & 0.2 & 0.05 \\ 0.1 & 0.2 & 0.1 & 0.2 & 0.4 \end{array}\right).$$

The hybridation of Muntz and Nabla algorithms provides bounds at each iteration. At iteration $n = Diam^{\mathbf{L}} = 3$, we obtain the first non trivial lower bound on each component of the steady-state distribution. According to Table 2 the average time before being absorbed knowing that the initial state is state 5 is bounded by:

$$2.0072 \leq E[T_5] \leq 2.5043.$$

n	$E[\underline{T_5}]$	$E[\overline{T_5}]$	$\|\|\mathbf{max}_s\{Y^{(n),s}\}\|\| - 1$	$1 - \|\|\mathbf{min}_s\{X^{(n),s}\}\|\|$
3	1.8469	7.8571	0.4733	0.6623
4	1.9104	6.7711	0.3723	0.6009
13	2.0062	3.5435	0.1934	0.2671
34	2.0072	2.5982	0.1642	0.0910
64	2.0072	2.5078	0.1613	0.0745
84	2.0072	2.5043	0.1612	0.0739

5 Concluding Remarks

We now plain to build a model checker for problems modeled with uncertain Markov chain and which will be based on the results in [1] and the algorithms we have presented here. Note that, we can also easily compute bounds on transient probabilities which are only based on the positivity of the matrices.

Proposition 1 *Let* $\mathbf{L} \preceq \mathbf{M}$ *and* $X_0^{\mathbf{L}}$ *a positive vector such that* $\|X_0^{\mathbf{L}}\| \leq 1$ *and* $X_0^{\mathbf{L}} \preceq \pi_0^{\mathbf{M}}$ *then, we have for all k,* $X_k^{\mathbf{L}} \preceq \pi_k^{\mathbf{M}}$ *where* $\pi_k^{\mathbf{M}}$ *is the distribution for the chain* $(\mathbf{M},\ \pi_0^{\mathbf{M}})$ *and* $X_k^{\mathbf{L}} = X_0^{\mathbf{L}}\,\mathbf{L}^k$.

It is also possible to obtain lower bounds on the probability of being absorbed knowing a bound of the initial distribution.

Proposition 2 *Let* $\mathbf{L} \preceq \mathbf{M}$ *such that* $\mathbf{L}(i,\ i) = 1$*, for a state i. Let* $X_0^{\mathbf{L}}$ *be a positive vector such that* $\|X_0^{\mathbf{L}}\| \leq 1$ *and* $X_0^{\mathbf{L}} \preceq \pi_0^{\mathbf{M}}$*. Then for all k, the probability of being absorbed at state i knowing a lower bound of the initial distribution is lower bounded by* $(X_0\,\mathbf{L}^k)(i)$ *and upper bounded by* $(X_0\,\mathbf{L}^k)(i) + 1 - \sum_{j \neq \mathscr{B}}(X_0\,\mathbf{L}^k)(j)$*, where* $\mathscr{B}$ *is the set of absorbing points of* $\mathbf{L}$ *(i.e.* $\mathbf{L}(j,\ j) = 1$*).*

Thus, we have obtained a set of algorithms which looks sufficient to address the numerical resolution of models based on uncertain discrete time Markov chains.

Acknowledgments This work is partially supported by DIGITEO grant MARINA-2010.

References

1. Aït Salaht F, Fourneau J-M, Pekergin N (2012) Computing entry-wise bounds of the steady-state distribution of a set of markov chains. In: 27th international symposium on computer and information sciences, Paris, France. Springer, pp 115–122
2. Buchholz P (2005) An improved method for bounding stationary measures of finite Markov processes. Perform Eval 62:349–365
3. Busic A, Fourneau J-M (2011) Iterative component-wise bounds for the steady-state distribution of a markov chain. Numer Linear Algebra Appl 18(6):1031–1049
4. Courtois PJ, Semal P (1984) Bounds for the positive eigenvectors of nonnegative matrices and for their approximations by decomposition. J ACM 31(4)

5. Fourneau J-M, Le Coz M, Pekergin N, Quessette F (2003) An open tool to compute stochastic bounds on steady-state distributions and rewards. In: 11th international workshop on modeling, analysis, and simulation of computer and telecommunication systems (MASCOTS, 2003) Orlando, FL, IEEE Computer Society 2003
6. Fourneau J-M, Quessette F (2012) Some improvements for the computation of the steady-state distribution of a Markov chain by monotone sequences of vectors. In ASMTA, Lecture notes in computer science. Springer
7. Franceschinis G, Muntz RR (1994) Bounds for quasi-lumpable markov chains. In: Proceedings of the 16th IFIP working group 7.3 international symposium on computer performance modeling measurement and evaluation, Performance '93. Elsevier Science Publishers B. V., pp 223–243
8. Haddad S, Pekergin N (2009) Using stochastic comparison for efficient model checking of uncertain markov chains. In: QEST 2009, 6th international conference on the quantitative evaluation of systems, Budapest, Hungary. IEEE Computer Society, pp 177–186
9. Trivedi KS (2002) Probability and statistics with reliability, queuing and computer science applications, 2nd edn. Wiley, Chichester, UK

Analysing and Predicting Patient Arrival Times

Tiberiu Chis and Peter G. Harrison

Abstract We fit a Hidden Markov Model (HMM) to patient arrivals data, represented as a discrete data trace. The processing of the data trace makes use of a simple binning technique, followed by clustering, before it is input into the Baum-Welch algorithm, which estimates the parameters of the underlying Markov chain's state-transition matrix. Upon convergence, the HMM predicts its own synthetic traces of patient arrivals, behaving as a fluid input model. The Viterbi algorithm then decodes the hidden states of the HMM, further explaining the varying rate of patient arrivals at different times of the hospital schedule. The HMM is validated by comparing means, standard deviations and autocorrelation functions of raw and synthetic traces. Finally, we explore an efficient optimal parameter initialization for the HMM, including choosing the number of hidden states. We summarize our findings, comparing results with other work in the field, and proposals for future work.

1 Introduction

Hidden Markov models (HMMs) have been used in various fields, ranging from Bioinformatics to Storage Workloads [1]. HMMs were first used in the late 1960s in statistical papers by Leonard E. Baum for statistical inference of Markov chains [2] and also for statistical estimation of Markov process probability functions [3]. Speech recognition became a field for training HMMs in the 1970s and 1980s [4], with many such speech models still used today [5]. In the late 1980s, HMMs acted as tools to analyse biological sequences and one result was the prediction of protein coding regions in genome sequences [6]. Following this, another use of HMMs has

T. Chis (✉) · P. G. Harrison Department of Computing, Imperial College London, Huxley Building, 180 Queens Gate, London SW7 2RH, UK
e-mail: tc207@doc.ic.ac.uk

P. G. Harrison
e-mail: pgj@doc.ic.ac.uk

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_8,

been to model common groups of protein sequences, an important topic in computational biology. HMMs have therefore been an asset in locating genes given an uncharacterized DNA sequence. The GENSCAN HMM ([6]) has been used for Eukaryotic gene finding and models the length distribution and sequence composition for each sequence type. The most probable path through the model (for the particular sequence) is found using the Viterbi algorithm. Note the path returned by Viterbi will contain the coordinates of the predicted genes. The accuracy of the GENSCAN HMM has been tested using metrics such as Sensitivity and Specificity of the data examples used. These are just some possible HMM applications in Biology. Extensions include the classification of proteins given an amino-acid sequence, modelling multiple sequences with HMM pairs, etc. In this paper, however, we focus on modelling patient arrivals at a hospital. In training a HMM on patient workloads using the standard Baum-Welch algorithm, to produce synthetic traces of their arrivals, one can decode the hidden states with the Viterbi algorithm and provide meaning to trends in the data. The HMM is simulated over 1, 000 runs to produce means and standard deviations (with 95 % confidence intervals) for raw and synthetic traces. Autocorrelation functions are computed, based on original and lagged versions of the patient arrival traces, which act as another validation metric for our model. However, before considering these HMM-generated simulations, the HMM is defined more precisely and three associated problems, underlying the estimation of its parameters and its interpretation, are solved.

1.1 What is a Hidden Markov Model?

A hidden Markov model (HMM) is a probabilistic model (a bivariate Markov chain) which encodes information about the evolution of a time series. The HMM consists of a hidden Markov chain $\{C_t\}$ (where t is an integer) with states not directly observable and a discrete time stochastic process $\{O_t\}_{t\geq 0}$, which is observable. Combining the two, we get the bivariate Markov chain $\{(C_t, O_t)\}_{t\geq 0}$ where all the statistical inference is done on $\{O_t\}$, as $\{C_t\}$ is not observed. Worth noting is that C_t governs the distribution of the corresponding O_t, and thus we assume that C_t is the only variable of the Markov chain that affects the probability distribution of O_t.

An illustration of the Markov chain and the interaction of hidden states with the possible observations is shown in Fig. 1, which shows a directed acyclic graph (DAG) specifying *conditional independence* relations for a HMM. The Markov chain

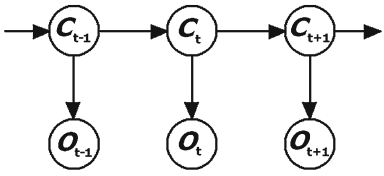

Fig. 1 A directed acyclic graph (*DAG*) showing conditional independence relations for a HMM. [9]

(with its hidden states C_i) has each node conditionally independent from its non-descendants given its parents. For example, given $C_1, C_2, \ldots, C_{t-1}, C_t$, we have that C_{t+1} is independent of $C_1, C_2, \ldots, C_{t-1}$ (i.e. the first Markov property). The observations O_i are linked to the Markov chain $\{C_t\}$ through *probability emissions* (i.e. C_i produces O_i at time i with a specified probability), where only the O_i are revealed in the HMM.

Generally, when constructing a HMM, there are three main problems that need to be solved. First, given the model λ, find the probability of a particular sequence of observations O, which can be solved by the *Forward-Backward algorithm*. Second, given a sequence of observations O, find the most optimal set of model parameters A, B, π[1]. This may be solved by statistical inference through the *Baum-Welch Algorithm* [7], which uses the Forward-Backward algorithm iteratively. Third, find the path of hidden states most likely to generate a sequence of observations O. This is solved using a posteriori statistical inference in the *Viterbi Algorithm* [8]. These three problems are solved using their respective statistical algorithms, as explained in the next section.

1.2 Solution of the Three Main Problems

Solutions to the three main problems, which use the same format as those seen in [9], are not presented in this paper. Nonetheless, we provide some brief notes on these solutions. The calculations of the Forward-Backward variables will assume Rabiner's solution [10]. The Baum-Welch algorithm uses the complete "alpha" and "beta" sets produced by the Forward-Backward algorithm. Using re-estimation formulas, Baum-Welch updates the model $\lambda = (A, B, \pi)$ in an iterative process, which terminates with convergent parameters and an optimal model. Finally, the Viterbi solution is an optimal state sequence, which essentially reveals the hidden part of the HMM λ based on some corresponding observations. For the training and subsequent simulation of the HMM on patient arrivals, these three solutions are implemented and executed on the observation traces. In the next section, we describe how these discrete traces are collected and processed to become inputs (as observation traces) into the Baum-Welch algorithm.

2 Collecting the Hospital Arrivals Trace

The data describing patient arrival times is anonymised data characterising patient arrival times at a London hospital between April 2002 and March 2007, as used in the study [11]. We extracted the arrival times for a period of four weeks, resulting in a "Hospital trace" that was output into a csv file, read into a Java class, and stored

[1] A is the state transition matrix, B is the observation emissions matrix, and π is the initial distribution.

as an array. With the trace collected, the next stage of the transformation process is assigning “bins" to the trace data.

This Hospital trace was binned by assigning the number of patients arriving every hour. After analysing the frequency of patient arrivals over four weeks, almost one third of cases witnessed no patients. On the other hand, two patients arrived in the same hour about 17 % of the time. In fact, on very few occasions were there more than eight patients in 1 h. Thus, choosing the one hour bin sizes resulted in an ideal range of values for forming clusters around our data points. The Hospital trace becomes a vector with arrivals and is input into a K-means clustering algorithm to obtain our observation traces.

The Hospital trace was limited to five or less clusters because after clustering with $K = 6$, there were two empty clusters present (i.e. with centroids 0.0). As we input the value of K manually, we decided to use a value of three clusters, as it gave closer means to the raw data when compared to HMM-generated data. The three clusters are listed here and are essentially our observation values:

$$\begin{pmatrix} 2.4 \\ 5.09 \\ 0.4 \end{pmatrix} \tag{1}$$

In (1), observation values (from top to bottom) represent: moderately frequent arrivals, frequent arrivals and very few arrivals. Having performed the K-means clustering on the Hospital trace, we obtain the observation trace ideal for input into the Baum-Welch algorithm.

3 Training the Baum-Welch Algorithm

For the training, we observe patient arrivals for $5,000$ h and therefore can input an observation trace of $5,000$ data points into the Baum-Welch algorithm. Note the sequence of observation values are defined as in (1) to populate this observation trace. We initialize each parameter (A, B, π) with equiprobable distributions and once all parameters converge, the simulation of the Baum-Welch algorithm produces $1,000$ different synthetic observation traces. Means and standard deviations (with confidence intervals) are obtained from the $1,000$ traces. After the simulation, the Baum-Welch algorithm produces an **initial distribution** (π), a **state transition matrix** (A) and an **observation emissions matrix** (B), which converged to:

$$\pi = (1.0, 0.0)$$

$$A = \begin{pmatrix} 0.8631 & 0.1369 \\ 0.1143 & 0.8857 \end{pmatrix}$$

Table 1 Arrivals/bin statistics on the raw and HMM-generated Hospital traces after 1,000 simulations

Trace	Mean	Std dev
Raw	1.6294	1.6828
HMM	1.6293 ± 0.0029	1.6815 ± 0.0014

$$B = \begin{pmatrix} 0.4896 & 0.3278 & 0.1826 \\ 0.0814 & 0.0 & 0.9186 \end{pmatrix}$$

From these results, we can observe that initially, there is a certainty we will start in state one. The probability we stay in this state is 0.8631 and therefore the probability that we move back to the other state is 0.1369 (as the rows in the transition probability matrix must sum up to one). Once in state two, the probability of going to state one is 0.1143 and the probability that we stay in state two is 0.8857. Overall, matrix A shows us that the model will most likely stay in the current state for some time. The first row in the observation emissions matrix (B) reveals various levels of patient arrivals which are all likely to occur. It seems that state one is more active in general. On average, there is approximately an 82 % chance of seeing at least 2.4 patients per hour in this state. Therefore, we label state one as the *dense* state. On the other hand, in state two, observation three is very likely to occur, and observation two never occurs. Therefore, from this state we expect to observe few patients in the one hour interval. In fact, one may label this the *sparse* state. Once we have obtained the initial distribution, state transition matrix and observation emissions matrix, the HMM will generate its own sequence of 5, 000 observations using these parameters. Then, means and standard deviations of the raw and HMM-generated traces are compared to validate our model. These results are presented in Table 1.

Analysing the arrivals/bin (i.e. the expected arrivals per hour) after 1, 000 simulations produces excellent results. Table 1 reveals that the bin-means match almost perfectly, and more pleasingly, the standard deviations are identical to two decimal places. The 95 % confidence intervals are small, given the 1, 000 population size (i.e. number of simulations). These statistics suggest that the HMM faithfully reproduces meaningful representations of patient arrival times at the individual bin level.

4 Viterbi-Generated Sequence of Hidden States

Another validation technique for the HMM is to train the Viterbi algorithm on observation sequences (i.e. sequences with values one, two or three) to generate its corresponding hidden state sequence. Equivalently, it reveals which state produced which observation and attempts to provide some explanation of the hidden states. Analysing an observation sequence of 4, 800 states (not shown), produced by Viterbi after training on a sequence of 4, 800 observations, the initial state is two (sparse). After several observations in this state, we switch to state one (dense). We continue to oscillate between these two states until the end of the sequence, almost in

symmetric fashion. An explanation to this oscillating pattern can be given as follows: state one represents *day* and state two represents *night*. To test this claim, the Viterbi sequence of 4, 800 states was analysed in greater detail by counting the number of times each state occurred: 2, 533 observations were generated from state one and 2, 267 from state two. Therefore, the Viterbi algorithm has generated 52.77 % day states and 47.23 % night states, approximately fifty-fifty. This is expected because a single 24 h day is divided into 12 h of day (e.g. 7am–7pm) and 12 h of night (e.g. 7pm to 7am). Therefore, the labelling of states as day and night has given meaning to the distribution of hidden states. In the next section, we discuss the last form of HMM validation, the autocorrelation function, which reflects the dynamics of the arrivals sequence rather than just static, per-bin statistics.

5 Autocorrelation

This section shows the results of comparing the autocorrelation functions (ACFs) of the raw, unclustered traces and then on the HMM-generated traces. The following graphs show the autocorrelation of patient arrival times at increasing lags for each trace. Figures 2 and 3 match well as both show little autocorrelation. The HMM-generated ACF shows less variation than the raw ACF, due to the choice of clusters perhaps. In the next section, we provide some insight into the process of choosing the optimal hidden states for a HMM.

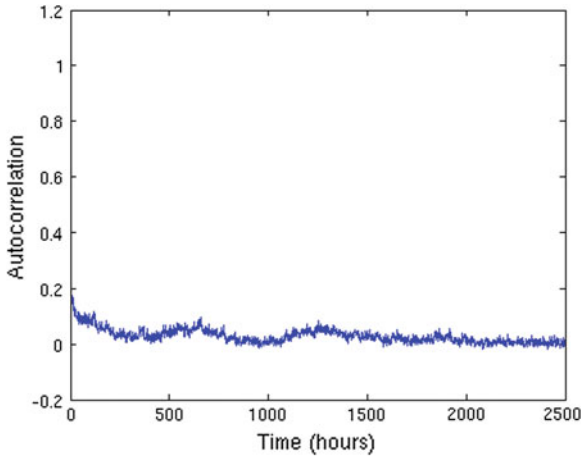

Fig. 2 ACF for raw patient arrivals

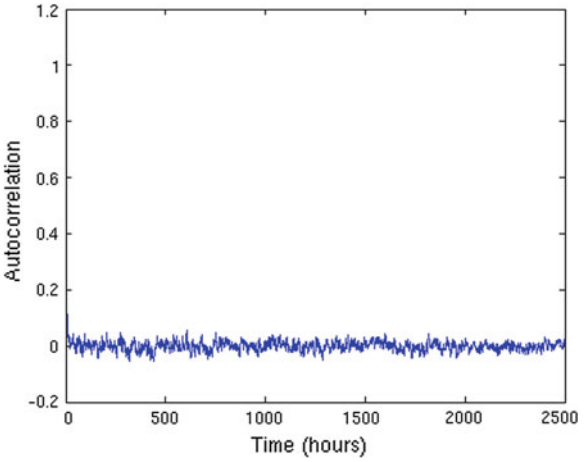

Fig. 3 ACF for HMM-generated patient arrivals

6 Optimal Number of Hidden States

As discussed in [12] there are two methods for finding the optimal number of hidden states. The first is *top-down* and is a binary split scheme. Initially, states are iteratively split into new states, continuing until no further improvement can be made. The second method is *bottom-up* and can be seen in [13]. In this scheme, there are many initial states and, during training, states are merged together until we end up with a small, optimal number of states. Choosing between these two methods is the key to obtaining the ideal number of states for a HMM. Methods of checking for the optimal number of states is observing the emission probabilities that are output from the Baum-Welch algorithm. For example, if two rows in the emissions matrix have very similar entries then we have (at least) one too many states for our HMM. We analyse an example of the emissions matrix produced for the hospital patient arrival times:

$$\begin{pmatrix} 0.003 & 0.422 & 0.575 \\ 0.437 & 0.157 & 0.406 \\ 0.02 & 0.941 & 0.039 \\ 0.0 & 0.996 & 0.004 \end{pmatrix}$$

Notice that the third and fourth rows have similar entries. Either our algorithm has not converged fully or, more likely, we have too many hidden states and thus our HMM had not been optimally set up. An immediate improvement is to initialise our HMM as either a two or three hidden state model. After applying our own method of analysing rows in the emissions matrix, we can compare our two remaining cases to find which number of hidden states helps the HMM perform better (in terms of

means and standard deviations). The comparison simply depends on the analysis of entries in the emissions matrix.

In the aforementioned example, the top-down or bottom-up methods are not necessary for such simple models. In fact, the small number of states used helped achieve the optimal HMM setup without the need to merge states. In [12], a graph of directed accuracies is produced for different hidden states, to identify the point of convergence precisely. Using similar techniques, we explore a simple bottom-up method of merging hidden states. Ideally, an improvement to the bottom-up technique would be to derive a systematic approach which works in any given scenario (e.g. storage workloads, patient arrivals, etc.) and produces the optimal number of hidden states. This includes analysing the emissions matrix to identify the effect of each state on the observation set. Then, a modification of the Baum-Welch algorithm is required to merge the chosen states whilst simultaneously storing any information about transition probabilities. This would provide an approximation to the state transition matrix for optimal states. Finally, we complete the Baum-Welch algorithm by using the new transition matrix to construct the new emissions matrix and calculate the initial probability distribution.

Despite the merging technique seeming computationally expensive, it is advantageous for achieving an HMM with optimal states when contextual information of the time series is not available. The power of this technique lies in the identification of the "most efficient setup", simply by using the original HMM parameters of a less efficient setup. Therefore, this technique is entirely self-contained in the sense that the computations are limited within the bounds of the Baum-Welch algorithm.

7 Conclusion and Extensions

The Hospital arrivals model was found to train successfully on patient arrivals, collected over months of analysis. The means and standard deviations matched well for raw and HMM-generated traces and both traces exhibited little autocorrelation. HMM parameters, fully converged after training, were used to predict the model's own synthetic traces of patient arrivals, therefore behaving as a fluid input model (with its own rates). An enhancement could be to assume instead that the arrival process is Poisson, with corresponding rates, and produce a cumulative distribution function for the patient arrivals workload.

For our model, there was significant focus on the Viterbi algorithm. Given the correct model parameters, as validated by means and standard deviations of raw and synthetic traces as well as autocorrelation functions, we focused on the hidden trends highlighted by the Viterbi algorithm. By counting the number of occurences of each state in the hidden state sequence and assigning each state to a category, the meaning of the hidden states became more apparent. The power, and straightforward implementation, of this process can lead to decoding of time series from which we want answers. Other applications of the Viterbi algorithm, as seen in industry, include speech recognition ([10]), biology([14]), etc. These papers have

tried similar decoding techniques using the Viterbi algorithm to deduce the physical significance of traces or parts of traces. As an extension, we aim to use Viterbi on different periods of the patient time series and understand how the hidden state distribution is updated. We also plan to consider more expressive traces that include type information for each arriving patient, so that a model with more than two hidden states would be required.

References

1. Harrison PG, Harrison SK, Patel NM, Zertal S (2012) Storage workload modelling by hidden markov models: application to flash memory. Perform Eval 69:17–40
2. Baum LE, Petrie T (1966) Stastical inference for probabilistic functions of finite markov chains. Ann Math Stat 37:1554–1563
3. Baum LE, Eagon JA (1967) An inequality with applications to statistical estimation for probabilistic functions of a markov process and to a model for ecology. Bull Am Math Soc 73:360–363
4. Rabiner LR (1989) A tutorial on hidden markov models and selected applications in speech recognition. IEEE 77:257–286
5. Ashraf J, Iqbal N, Khattak NS, Zaidi, AM (2010) Speaker Independent Urdu Speech Recognition Using HMM
6. Burge C, Karlin S (1997) Prediction of complete gene structures in human genomic DNA. J Mol Biol. Stanford University, California, USA, Computer and Information Sciences, pp 78–94
7. Baum LE, Petrie T, Soules G, Weiss N (1970) A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann Math Stat 41:164–171
8. Viterbi AJ (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans Inf Theor 13:260–269
9. Chis T (2011) Hidden markov models: applications to flash memory data and hospital arrival times. Department of Computing, Imperial College London
10. Rabiner LR, Juang BH (1986) An introduction to hidden markov models. IEEE ASSP Mag 3:4–16
11. Au-Yeung SWM, Harder U, McCoy E, Knottenbelt WJ (2009) Predicting patient arrivals to an accident and emergency department. Emerg Med J 26:241–244
12. Zhang L, Chan KP (2010) Bigram HMM with context distribution clustering for unsupervised Chinese part-of-speech tagging. Department of Computer Science, Hong Kong
13. Brand M (1999) An entropic estimator for structure discovery. In; Proceedings of the (1998) conference on advances in neural information processing systems II. MIT Press. MA, USA, Cambridge, pp 723–29
14. Krough A, Brown M, Mian S, Sjolander K, Haussler D (1994) Hidden markov models in computational biology. J Mo Biol 235:1501–1531 (University of California, Santa Cruz, USA, Computer and Information Sciences)

Optimal Behaviour of Smart Wireless Users

Boris Oklander and Erol Gelenbe

Abstract We consider a smart or cognitive user (CU) that operates as a secondary user of a cognitive channel. Before transmission, the CU samples the channel until it estimates that it can be accessed successfully. When the CU transmits a packet, it may nevertheless be unsuccessful because its estimate was wrong. The CU then knows about its failure and stops the ongoing transmission sometime before the transmission ends. Then the CU restarts sensing the channel. In this paper we analyse the total delay experienced by a CU. We derive the first and second moments of the effective transmission time of a packet sent by such a smart user, in view of the fact that the sensing and errors made in transmitting a packet when the channel is actually unavailable, will introduce additional delay. This is similar to a "vacation time" in queues, though it differs from conventional vacation time models because such "sampling vacations" can be beneficial to the future transmission.

1 Introduction

Admission control [1] is a classical topic in communications, but cognition [2] as a means to enhance the performance of users and the network is relatively new. Cognitive Users' (CU) can achieve better performance based on smart access to resources such as power and spectrum. However, adaptation to dynamically changing conditions of communication channels is a challenge for CUs. Acting as a secondary user (SU), a CU cedes priority to ongoing transmissions of PUs and accesses a channel only after sensing and estimating that it is not used by PUs. Thus efficient CUs will trade between competing requirements of sensing, transmitting and inter-

B. Oklander (✉) · E. Gelenbe
Intelligent Systems and Networks,Imperial College London, London SW7 2BT, UK
e-mail: b.oklander@imperial.ac.uk

E. Gelenbe
e-mail: e.gelenbe@imperial.ac.uk

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_9,

ference avoidance, subject to power limitations. Sensing can also be driven by the CUs desire to avoid a network attack [3].This of course requires understanding the interdependence of the various processes involved and their impact on performance, which will be comprehensively examined in a forthcoming paper [4]

Various approaches for CU modeling appear in the literature. A pricing mechanism for control of the CU buffer is proposed in [5] assuming possible transmission failures due to interruptions caused by the PUs. In [6], a network of CUs contending for spectrum access is considered and queueing delays are studied using a fluid queue approximation, where the sensing is assumed to be accurate at all times while the time it takes to sense the channel is neglected. Similar sensing characteristics are assumed in [7]. In [8], the joint design of spectrum sensing and access mechanisms is discussed and a joint design and is shown to improve system throughput. A different model for joint sensing and access is proposed in [9], where the sensing comes at the expanse of shorter transmissions. The throughput of the proposed model is optimized by means of dynamic programming, while the queueing behavior is studied through simulations. In [10, 11], CR systems with finite buffer and imperfect sensing are studied. A QoS-aware sensing algorithm is designed in [12]. In [13] CU is modeled as a M/G/1 queue with vacations assuming perfect sensing, while in [14, 15] the CUs are modeled as a preemptive resume priority (PRP) M/G/1 queue, and PUs are modeled as high priority users.

In this paper we present a CU queueing model with walking type server [16] or vacations [17]. The analysis yields explicit expressions for the first two moments of CU packet service time, and show that as opposed to the conventional server vacations that prolong the effective service time, here they *reduce* the service and response time.

2 System Description

Packets arrive to a CU's input buffer according to a stationary arrival process with rate λ[packets/sec]. The CU operates as a secondary user trying to avoid inference with the PUs. This requires sensing of the channel and estimating its state. We assume that at any given time the CU either senses the channel or transmits over it, but it cannot do both of the operations simultaneously. The CU repeats the sensing period τ[s] until it estimates that the channel is free and the outcomes of successive sensing periods are assumed to be independent; then if there are any packets in the queue, transmission of the first packet in the queue begins, otherwise the CU starts a new sensing phase. We denote by $f(\tau)$ the probability that the CU estimates the channel to be free after sensing it for time τ. When CU decides that the channel is free, it carries out a transmission if there is a packet in its input queue. However the transmission may not succeed because the estimation was actually wrong, and the whole sampling process has to be re-started. If T[s] is the total transmission time of a packet, we denote by $s(\tau, T)$ the probability that when the CU estimates that the channel is free, the transmission is actually successful. We denote by ϵ the length of

the period from the start of the packet transmission until the CU knows whether it is successful or not with the assumption that $\varepsilon \leq T$ so that before the packet transmission is complete the CU already knows whether it has to repeat the whole process.

Denote S the total effective service time of the CU for a packet that is transmitted in direct succession to a packet that was previously successfully transmitted, and denote $E[S]$ its expected value and $E[S^2]$ its second moment, we can define S recursively as:

$$S = \begin{cases} \tau + D & \text{w.p. } f(\tau) \\ \tau + S' & \text{w.p. } 1-f(\tau) \end{cases} \tag{1}$$

where S' and S'' below have the same distribution as S, while D is given by:

$$D = \begin{cases} T & \text{w.p. } s(\tau,T) \\ \varepsilon + S'' & \text{w.p. } 1-s(\tau,T) \end{cases} \tag{2}$$

As a result, we have the following coupled equations:

$$E[S] = \tau + f(\tau)E[D] + (1-f(\tau))E[S] \tag{3}$$

$$E[D] = s(\tau,T)T + (1-s(\tau,T))\epsilon + (1-s(\tau,T))E[S] \tag{4}$$

that yield:

$$E[S] = T + \frac{1-s(\tau,T)}{s(\tau,T)}\epsilon + \frac{1}{s(\tau,T)f(\tau)}\tau \tag{5}$$

$E[D]$ can then be obtained from $E[S]$. Similarly, and the second moments are:

$$E[S^2] = f(\tau)E[(\tau + D)^2] + (1-f(\tau))E[(\tau + S)^2] \tag{6}$$

where in turn:

$$E[D^2] = s(\tau,T)T^2 + \big(1-s(\tau,T)\big)E[(\epsilon + S)^2] \tag{7}$$

Combining Eqs. (6, 7) and substituting the expressions of $E[S]$ and $E[D]$ leads to the following expression of the second moment $E[S^2]$:

$$E[S^2] = T^2 + \frac{f(\tau)(1-s(\tau,T))(2E[S]\epsilon+\epsilon^2)+(2E[S]\tau-\tau^2)}{s(\tau,T)f(\tau)} \tag{8}$$

2.1 Dealing with the Idle Queue

As soon as the CU queue becomes idle just after the end of a successful transmission, the CU will again commence sampling the channel, and will do so indefinitely until a new arrival occurs. When the arrival occurs, it will have to wait until the end of

the current sampling period, so that the CUs server can be viewed to be on vacation for successive periods of duration $V = \tau$ until the first arrival occurs. After the first arrival occurs, the first service of duration S_0 can begin, but the duration will differ from S because it will start with the residual time τ'. After the sensing interval ends, it will be identical in distribution to D if the channel is sensed to be free and it will be identical in distribution to S otherwise:

$$S_0 = \begin{cases} \tau' + D & \text{w.p. } f(\tau) \\ \tau' + S & \text{w.p. } 1-f(\tau) \end{cases} \tag{9}$$

It can be shown that the first two moments of S_0 are given by:

$$E[S_0] = T + (\tau' - \tau) + \frac{1-s(\tau,T)}{s(\tau,T)}\epsilon + \frac{1}{s(\tau,T)f(\tau)}\tau \tag{10}$$

and

$$E[S_0^2] = (\tau')^2 + f(\tau)(2E[D]\tau' + E[D^2]) + (1-f(\tau))(2E[S]\tau' + E[S^2]) \tag{11}$$

Therefore the appropriate model for the system we consider is a server with vacation [16], but with an exceptional (shorter) service for the first packet arriving after an idle period [17]. This system is equivalent to a GI/GI/1 queue with exceptional service for the first packet in each busy period.

2.2 Computing the Probabilities $f(\tau)$ and $s(\tau,T)$

The probabilities $f(\tau)$ and $s(\tau,T)$ are the missing parameters in the CU model. While the operational parameters τ and T could be deliberately chosen by CU, their impact on $f(\tau)$ and $s(\tau,T)$ is not clear. Obviously, these probabilities will depend on specific channel conditions and sensing techniques. Nevertheless, it may be useful to relate these quantities using some robust model in order to better understand their interdependencies as well as their impact on the CUs performance.

We propose a general method for coupling the quantities τ, T, $f(\tau)$ and $s(\tau,T)$ by making use of Cumulative Distribution Functions (CDFs) and an on-off model of channel occupancy. From the CUs perspective, the channel alternates between on and off periods during which a transmission of CU would fail or succeed, respectively. The duration of the (off) period is denoted by T_{on} (T_{off}) and its CDF by $F_{on}(t)$ ($F_{off}(t)$). We characterize CU by the estimation time T_{est} which is the time interval from the moment the sensing starts until CU correctly estimates the channels state. We denote by $F_{est}(t)$ the CDF of T_{est}. Given the CDFs $F_{on}(t)$, $F_{off}(t)$, $F_{est}(t)$, and the CDF of the residual life-time of the off period T_{res_off} $F_{res\_off}(t)$, which can be calculated from $F_{off}(t)$, we obtain:

$$E[T_{on}] = \int_{t=0}^{\infty} (1 - F_{on}(t))dy \tag{12}$$

The T_{on} obtained in the same manner. The probability P_{on} (P_{off}) of the channel to be in on (off) state, respectively, is given by:

$$P_{on} = \frac{E[T_{on}]}{E[T_{on}] + E[T_{off}]} \tag{13}$$

P_{off} complements P_{on} to unity.

We can calculate now the probabilities $f(\tau)$ and $s(\tau,T)$:

$$\begin{aligned} f(\tau) &= P_{off} Pr(T_{res\_off} > \tau) Pr(T_{est} \leq \tau) \\ &\quad + (1 - P_{off} Pr(T_{res\_off} > \tau)) Pr(T_{est} > \tau) \\ &= P_{off}(1 - F_{res\_off}(\tau)) F_{est}(\tau) \\ &\quad + (P_{on} + P_{off} F_{res\_off}(\tau))(1 - F_{est}(\tau)) \end{aligned} \tag{14}$$

and

$$\begin{aligned} s(\tau,T) = & \\ & \frac{P_{off} Pr(T_{res\_off} > \tau) Pr(T_{est} \leq \tau)}{f(\tau)} Pr(T_{res\_off} > \tau + T) \\ & + \frac{(1 - P_{off} Pr(T_{res\_off} > \tau) Pr(T_{est} > \tau)}{f(\tau)} P_{off} Pr(T_{res\_off} > T) \\ = & \frac{P_{off}(1 - F_{res\_off}(\tau) F_{est}(\tau)}{f(\tau)} (1 - F_{res\_off}(\tau + T)) \\ & + \frac{P_{on} + P_{off} F_{res\_off}(\tau))(1 - F_{est}(\tau))}{f(\tau)} P_{off}(1 - F_{res\_off}(T)) \end{aligned} \tag{15}$$

In Fig. 1 we plot the values of $f(\tau)$ and $s(\tau,T)$ for three different systems, noted as (a),(b) and (c). In all the three systems, T_{on}, T_{off} and T_{est} are exponentially distributed with corresponding rates λ_{on}, λ_{off} and λ_{est}. The values of (λ_{on}, λ_{off}, λ_{est}) in systems (a), (b) and (c) are (10, 10^{-2}, 0.5), (10^{-4}, 10^{-3},1) and (3, 0.5, 0.2) respectively. The plots show that in the same range of operational parameters τ and T, the characteristics of $f(\tau)$ and $s(\tau,T)$ are quite different and are affected by the dynamics of the channels state.

3 Performance Optimization for the CU

In this section, we analyze the performance of the CU which is modeled by a server of a walking type. The performance measures of interest that are calculated and optimized for are throughput, delay, energy, interference, QoS and system response time.

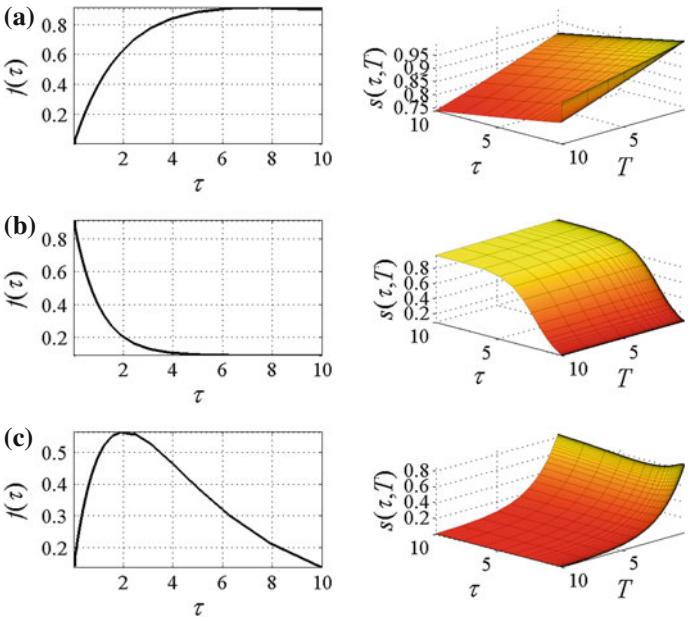

Fig. 1 Characteristics of $f(\tau)$ and $s(\tau,T)$ for systems (**a**), (**b**) and (**c**)

3.1 Throughput

For the throughput analysis we assume that CU has always packets to transmit. In this case, throughput γ is the ratio of the packet transmission time T and the first moment of the average service time $E[S]$ which was calculated in (5).

$$\gamma(\tau,t) = \frac{T}{E[S]} = \left[1+\frac{1-s(\tau,T)}{s(\tau,T)}\left(\frac{\epsilon}{T}\right)+\frac{1}{s(\tau,T)f(\tau)}\left(\frac{\tau}{T}\right)\right]^{-1} \tag{16}$$

In order to maximize its throughput, CU could set the values of the operational parameters (τ, T)=(τ^*, T^*) such that

$$(\tau^*, T^*) \in \underset{(\tau,T)}{\operatorname{argmax}}\gamma(\tau, T) \tag{17}$$

From (16) it is clear that in order to maximize its throughput, CU should set the parameters τ and T in a manner that would increase the values of $f(\tau)$ and $s(\tau,T)$ and decrease the ratios ϵ/T and τ/T. These generally conquering requirements result in an inherent trade-off in (17). The exact method for solving this optimization problem would depend on the properties of the functions $f(\tau)$ and $s(\tau,T)$. The functional behavior of $\gamma(\tau,T)$ could be demonstrated through the following example. Assume a channel and a CU for which T_{on}, T_{off} and T_{est} are exponentially distributed with

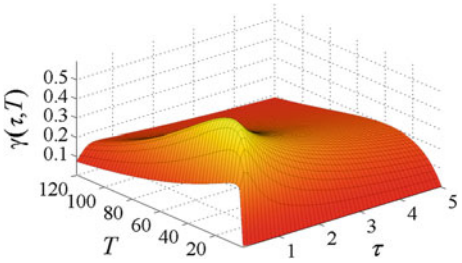

Fig. 2 Throughput versus parameters (τ, T). The system is characterized by $T_{on}\sim\exp(10^{-3})$, $T_{off}\sim\exp(10^{-2})$, $T_{est}\sim\exp(10)$ and $\varepsilon = 0.5T$. The optimal parameters are $(\tau^*, T^*)=(0.42,16.57)$ and the maximal throughput is $\gamma(\tau^*, T^*)=0.59$

parameters $\lambda_{on} = 10^{-3}[\mathrm{s}^{-1}]$, $\lambda_{off} = 10^{-2}[\mathrm{s}^{-1}]$ and $\lambda_{est} = 10[\mathrm{s}^{-1}]$, respectively. In this system, the channel is in the "on" state 91 % of the time on average, and for the complementary 9 % of the time it is in the "off" state (available for CUs transmissions). Additionally, assume that $\varepsilon = 0.5T$. Using the Eqs. (12)–(16), we are able to calculate $\gamma(\tau,T)$ (see Fig. 2). The numerical solution of (17) shows that the optimal parameters for this scenario are $(\tau^*,T^*) = (0.42,16.57)$ and the maximal throughput is $\gamma(\tau^*,T^*) = 0.59$.

In addition to reducing the average service or system times, it is often important to provide Quality of Service (QoS) guarantees. We consider here QoS guarantees of the form that the difference between the service time S and its expectation $E[S]$ would be small with high probability. It is reasonable to set the requirement of $|S-E[S]|$ to be small relatively to some proportion c of the packet size T. Such guarantee involves $Var(S)$ and can be evaluated by applying Chebyshevs inequality:

$$P(|S - E[S]| \geq cT) \leq \frac{Var(S)}{(cT)^2} \tag{18}$$

From (18) it is clear, that for any value of c, providing better QoS involves minimizing the ratio $Q = Var(S)/T^2$. It can be shown that:

$$Q(\tau,T) = \frac{(\tau+\epsilon f(\tau))^2 - f(\tau)s(\tau,T)(f(\tau)\epsilon^2+2\epsilon\tau+\tau^2)}{(Tf(\tau)s(\tau,T))^2} \tag{19}$$

In order to maximize QoS, CU could set the values of the operational parameters $(\tau,T) = (\tau^*,T^*)$ such that:

$$(\tau^*,T^*) \in \underset{(\tau,T)}{\operatorname{argmin}} Q(\tau,T) \tag{20}$$

Again, we demonstrate the dependence of the QoS on the operational parameters (τ,T) for the previously mentioned example (see Fig. 3).

The numerical solution of (20) shows that the optimal parameters for this scenario are $(\tau^*,T^*) = (0.44,14.77)$ and $Q(\tau^*,T^*) = 0.59$. This means that the probability for service time to deviate from its expectation more than c times of the packet length is less than $0.59c^{-2}$.

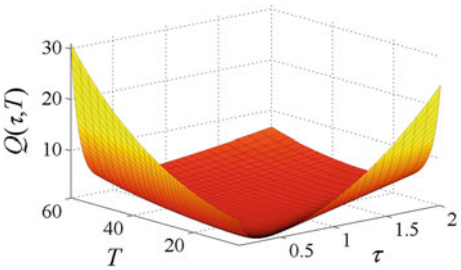

Fig. 3 $Q(\tau,T)$ versus (τ,T). The system is characterized by $T_{on}\sim\exp(10^{-3})$, $T_{off}\sim\exp(10^{-2})$ and $T_{est}\sim\exp(10)$. The optimal parameter values are $(\tau^*,T^*) = (0.44,14.77)$

4 Conclusions and Future Work

A CU's sensing and transmission was modeled as a server of walking type yielding closed form expressions for the first two moments of the equivalent response time, and showed that the system's performance can be optimized with a judicious choice of a few parameters. In future work we will study "sensible" decisions for access to multiple channels [18], and how different users may share and synchronize [19] distributed resources [20] and information. Learning techniques [21, 22] and online databases with imperfect information [23] will also be considered.

References

1. Gelenbe E, Mang X, Onvural R (1997) Bandwidth allocation and call admission control in high-speed networks. IEEE Commun Mag 35(5):122–129
2. Gelenbe E (2009) Steps towards self-aware networks. Commun ACM 52(7):66–75
3. Gelenbe E, Loukas G (2007) A self-aware approach to denial of service defense. Comput J 51(5):1299–1314
4. Oklander B, Gelenbe E (2013) Cognitive users: a ccomprehensive analysis, Submitted for publication
5. Li H (2011) Socially optimal queuing control in cognitive radio networks subject to service interruptions: To queue or not to queue? IEEE Trans Wirel Commun 10(5):1656–1666. doi:10.1109/TWC.2011.030411.101220
6. Wang S, Zhang J, Tong L (2012) A characterization of delay performance of cognitive medium access. IEEE Trans Wirel Commun 11(2):800–809. doi:10.1109/TWC.2012.010312.110765
7. Hwang GU, Roy S (2012) Design and analysis of optimal random access policies in cognitive radio networks. IEEE Trans Commun 60(1):121–131. doi:10.1109/TCOMM.2011.112311.100702
8. El-Sherif A, Liu K (2011) Joint design of spectrum sensing and channel access in cognitive radio networks. IEEE Trans Wirel Commun 10(6):1743–1753. doi:10.1109/TWC.2011.032411.100131
9. Hoang A, Liang YC, Zeng Y (2010) Adaptive joint scheduling of spectrum sensing and data transmission in cognitive radio networks. IEEE Trans Commun 58(1):235–246. doi:10.1109/TCOMM.2010.01.070270
10. Hamza D, Aissa S (2011) Impact of sensing errors on the queueing delay and transmit power in cognitive radio access. In: Communications and Information Technology (ICCIT), International conference on 2011, pp 53–58, Doi: 10.1109/ICCITECHNOL.2011.5762693

11. Wang J, Huang A, Wang W, Yin R (2012) Queueing analysis for cognitive radio networks with lower-layer considerations. In: Wireless communications and networking conference (WCNC), 2012 IEEE, pp 1269–1274, Doi: 10.1109/WCNC.2012.6213973
12. Choi JK, Kwon KH, Yoo SJ (2009) Qos-aware channel sensing scheduling in cognitive radio networks. In: Ninth IEEE international conference on computer and information technology, 2009. CIT '09, vol 2, (2009), pp 63–68. Doi: 10.1109/CIT.2009.115
13. Piazza D, Cosman P, Milstein L, Tartara G (2010) Throughput and delay analysis for real-time applications in ad-hoc cognitive networks. In: Wireless communications and networking conference (WCNC), 2010 IEEE, pp 1–6 (2010). Doi: 10.1109/WCNC.2010.5506370
14. Wang LC, Wang CW, Adachi F (2011) Load-balancing spectrum decision for cognitive radio networks. IEEE J Sel Areas Commun 29(4):757–769. doi:10.1109/JSAC.2011.110408
15. Wang LC, Wang CW, Feng KT (2011) A queueing-theoretical framework for qos-enhanced spectrum management in cognitive radio networks. IEEE Wirel Commun 18(6):18–26. doi:10.1109/MWC.2011.6108330
16. Gelenbe E, Iasnogorodski R (1980) A queue with server of walking type (autonomous service). Ann Inst Henri Poincare 16:63–73
17. Takagi H (1991) Queueing analysis: a foundation of performance evaluation, vol 1 : vacation and priority systems. QUEUEING ANALYSIS, North-Holland
18. Gelenbe E (2003) Sensible decisions based on qos. CMS 1(1):1–14
19. Gelenbe E, Sevcik KC (1979) Analysis of update synchronisation algorithms for multiple copy data bases. IEEE Trans Comput C-28(10):737–747
20. Aguilar J, Gelenbe E (1997) Task assignment and transaction clustering heuristics for distributed systems. Inf Sci 97(1):199–219
21. Fourneau JM, Gelenbe E (1999) Random neural networks with multiple classes of signals. Neural Comput 11(4):953–963
22. Gelenbe E (2000) The first decade of g-networks. Eur J Oper Res 126(2):231–232
23. Gelenbe E, Hébrail G (1986) A probability model of uncertainty in data bases. In: ICDE. IEEE Computer Society, pp. 328–333, 1986

Hyper-Heuristics for Performance Optimization of Simultaneous Multithreaded Processors

I. A. Güney, Gürhan Küçük and Ender Özcan

Abstract In Simultaneous Multi-Threaded (SMT) processor datapaths, there are many datapath resources that are shared by multiple threads. Currently, there are a few heuristics that distribute these resources among threads for better performance. A selection hyper-heuristic is a search method which mixes a fixed set of heuristics to exploit their strengths while solving a given problem. In this study, we propose learning selection hyper-heuristics for predicting, choosing and running the best performing heuristic. Our initial test results show that hyper-heuristics may improve the performance of the studied workloads by around 2%, on the average. The peak performance improvement is observed to be 41% over the best performing heuristic, and more than 15% over all heuristics that are studied. Our best hyper-heuristic performs better than the state-of-the art heuristics on almost 60% of the simulated workloads.

1 Introduction

Today, the Simultaneous Multi-Threaded (SMT) processors aim to increase the system throughput by executing instructions from different threads in a single clock cycle. These processors are widely utilized in both high-end (e.g. Intel core i7) and low-end (e.g. Intel Atom) computers, and they try to satisfy the high system throughput requirements in high-end machines and the efficient and effective utilization of

I. A. Güney (✉) · G. Küçük
Department of Computer Engineering, Yeditepe University, Istanbul, Turkey
e-mail: iguney@cse.yeditepe.edu.tr

G. Küçük
e-mail: gkucuk@cse.yeditepe.edu.tr

E. Özcan (✉)
School of Computer Science, University of Nottingham, Jubilee Campus, Nottingham, UK
e-mail: ender.ozcan@nottingham.ac.uk

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_10,

system resources in low-end machines, together. SMT processors are also utilized in highly popular contemporary chip multi processors (e.g. Intel Xeon servers).

In SMT processors, there are many datapath resources (Issue Queue, Re-Order Buffer, Load/Store Queue, Physical Register Files, Arithmetic Logic Units and cache structures) that are shared by multiple running threads. In an uncontrolled environment, threads assume that all the shared datapath resources are solely dedicated to themselves, and, inadvertently, they go into a race for stealing datapath resources from each other. In such case, a single thread may take over the Issue Queue, pollute the caches and fill the Physical Register Files by the instructions that are introduced into the processor from a mispredicted path even though the branch misprediction rate is known to be high. As a result, today, the throughput obtained from SMT processors are much lower than the potential throughput that can be actually obtained.

There are various strategies to improve the efficiency of these processors. First, there are fetch policies that try to regulate the stream of instructions that are introduced to the pipeline. These techniques change the distribution of shared resources, indirectly. Most famous examples of these are ICOUNT, which gives fetch priority to threads with less resource occupancy, BCOUNT, which favors threads with the fewest unresolved branches, MISSCOUNT, which gives priority to threads with fewest outstanding D-cache misses, STALL, which triggers fetch-lock when a load operation stays to be outstanding beyond some threshold number of cycles and FLUSH, which measures resource clog and when it happens recovers by flushing the stalled instructions [1, 2].

Beside these fetch throttling techniques, there are resource partitioning techniques that directly distribute shared resource partitions among running threads. Basically, these techniques dynamically decide how a shared resource is to be partitioned and distributed. The most famous example of these techniques is known as Dynamically Controlled Resource Allocation (DCRA) [3]. In DCRA, each thread and the datapath resource are dynamically tracked by a number of hardware counters. For example, when a thread has a pending cache miss, it is immediately labeled as a slow thread, or when a resource is not used by a thread for a threshold number of cycles, then the thread for that resource becomes inactive. Then, the DCRA mechanism tries to give more resources to the slow threads by stealing from fast or inactive threads.

SMT resource distribution via hill climbing (HILL) is another resource partitioning mechanism that runs in epochs (periodic intervals) [4]. HILL assumes that there is a certain optimum in the performance graph and it tries to reach to that peak by dynamically changing resource distributions in a greedy fashion. In the initial trial epochs, each thread gets its chance to show its performance with extra resources. At the end of these trial epochs, the performance of each thread is compared and the best performing (and the most deserving) thread is selected for receiving additional resources. Then, these trial epochs and the consequent resource distribution are done inside an infinite loop as long as the processor is running.

The Adaptive Resource Partitioning Algorithm (ARPA) introduces efficiency metric into the picture [5]. Similar to HILL, ARPA tries to give more resources to the most deserving thread by stealing resources from the others. The efficiency metric, committed instructions per resource entry (CIPRE), is a thread specific metric which

is evaluated at the end of each epoch. When a thread does a great job and commits many instructions with limited number of resources, its CIPRE value becomes high, and ARPA gives more resources to that thread. In HILL, a thread can show the best performance and be chosen to receive more resources every epoch regardless of its efficiency. As a result, a thread may starve to its death, since it cannot perform better than some other thread. ARPA solves this issue implicitly by its efficiency metric. When a thread receives more resources its CIPRE value gets lower and lower if it commits similar amount of instructions every epoch. In such cases, a thread with worse performance may get its share, since its efficiency may go up after a while.

Vandierendonck and Seznec [6] propose a new fetch throttling mechanism called Speculative Instruction Window Weighting. This mechanism fetches instructions from the thread with least amount of work left in the pipeline. The amount of work left for each thread is predicted by assigning weights to instructions. These weights are determined by the instruction type, confidence level of branch instructions and confidence level, prediction result and memory-level parallelism for memory instructions. By limiting the maximum amount of work of a thread, distribution of datapath resources is also achieved.

Another fetch policy by Eyerman and Eeckhout [7] takes memory-level parallelism into consideration. Their design predicts long-latency loads and the number of instructions a thread must go down in the instruction stream in order to exploit memory-level parallelism. The algorithm stalls fetching if the thread has reached the number of instructions predicted by the MLP-predictor, or flushes instructions beyond the predicted number of instructions in case a long-latency load is identified.

Heuristics (meta-heuristics) are problem specific inexact, rule of thumb computational methods. DCRA, HILL and ARPA are examples of such heuristics. In literature, different heuristics with different performances for almost all "hard" problems could be found. It has been observed that each heuristic may be successful in solving different problem instances. Hyper-heuristics are general methods which search the space generated by a set of heuristics rather than solutions to directly solve a given problem. A goal is designing intelligent and automated approaches enabled to combine the strengths of heuristics while avoiding their weaknesses for solving not only the instances in hand, but also the unseen ones. Hyper-heuristics have been applied to many static problems, whereas there are a few studies on their applications to dynamic environment problems. In these studies, either a theoretical problem or a benchmark function is used, where the changes in the environment can be controlled.

As a result, the heuristics, described up to this point, try to achieve a better performance compared to the one that we can observe in a baseline configuration, in which resource sharing is not regulated at all. The motivation behind this study is very simple. We show that there is no single algorithm that performs best in all the SMT workloads. In some workloads, DCRA works best, in some other workloads HILL works better than others, and in the rest of the workloads ARPA performs best. Our literature survey, preliminary studies and the variety of problem instances and the observed dynamic changes show us that hyper-heuristics area very suitable choice for the performance optimization on SMT processors. Although, in the literature, there are only a few heuristics proposed for solving this problem, there is

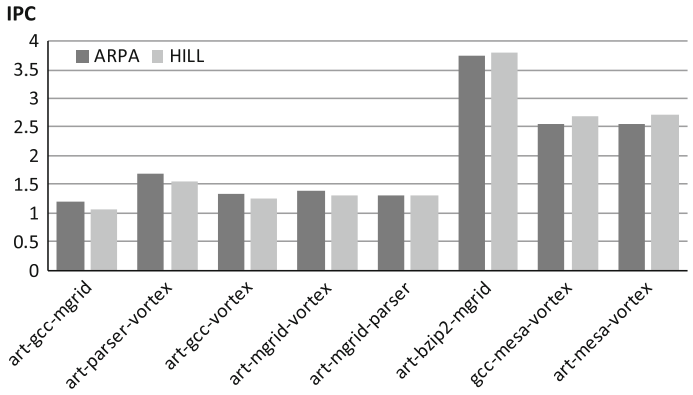

Fig. 1 ARPA and HILL performance comparison on a few SMT workloads

no study showing how general these heuristics are or providing a thorough performance analyses for the proposed heuristics. More importantly, there is no real world application of hyper-heuristics to such a dynamic environment problem in hand.

In this study, we aim to optimize the performance of SMT processors by partitioning datapath resources among running threads by using hyper-heuristics. Since, HILL and ARPA heuristics have similar periodic nature; we studied combining both under several hyper-heuristics throughout this study.

Figure 1 shows the performance (Instruction Per Cycle) graphs for some of the workloads that we studied. In art-gcc-mgrid and art-parser-vortex workloads ARPA performs better than HILL (12% and 8%, respectively). However, there are some other workloads in which ARPA performs worse than HILL. For instance, in gcc-mesa-vortex and art-mesa-vortex workloads HILL performs more than 5% better. Note that while ARPA is successfully running art-parser-vortex workload, when parser benchmark is replaced with mesa benchmark everything changes upside down and HILL starts to become more successful. This graph shows us that some heuristics can be successful in some of the workloads and some others can be successful in some other workloads. To the best of our knowledge, there is no such study that work on hyper-heuristics to dynamically select the proper heuristic at run time, and, in this study, we are aiming to fill this gap.

2 Proposed Design

The initial design starts by creating a habitat that may run both ARPA and HILL, interchangeably. Both ARPA and HILL track down runtime statistics collected by a number of hardware counters. For instance, both of them require the number of committed instructions for each thread in time periodic intervals called *epochs*. They also need a comparator circuitry to decide if the performance of a trial epoch is greater

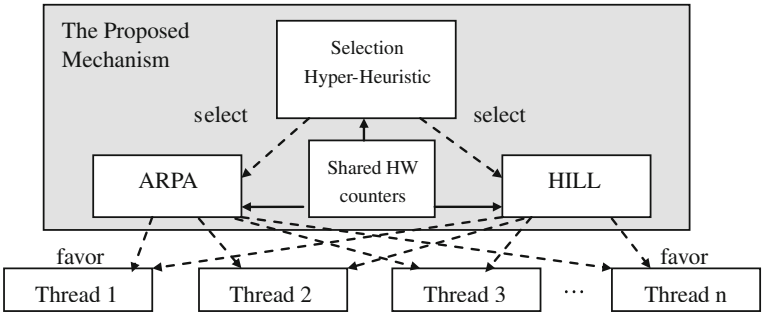

Fig. 2 The proposed design

than the performance value experienced by the other trial epochs or the CIPRE value of a thread is greater than the others. The resulting circuitry that runs both heuristics is less complex than what one may expect.

As shown in Fig. 2, our proposed design brings ARPA and HILL SMT partitioning heuristics, together. The job of these heuristics is to favor one of the running threads and to award it with more resources. The shared hardware counters keep runtime statistics that are required by the heuristics' (and the hyper-heuristic's) evaluation functions. A few example counters are *committed instructions per cycle per epoch* (IPCepoch_i, for the i*th* epoch), CIPRE and *fetched instructions per cycle per epoch* (FIPCepoch_i). The main responsibility of our proposed hyper-heuristic is the careful selection of the heuristic that is to be utilized for the next epoch.

There are two types of high level hyper-heuristic methodologies managing a set of low level heuristics: *selection* and *generation* [8]. Selection hyper-heuristics frequently consist of two successive stages of heuristic selection and move acceptance [9]. Most of the simple selection hyper-heuristic components are introduced in [10]. For example, *random permutation gradient* heuristic selection creates a permutation list of low level heuristics and chooses a low level heuristic in the given order one by one at each step to apply on the current solution. If a chosen heuristic makes an improvement, the same heuristic is utilized.

There are more elaborate hyper-heuristics making use of machine learning techniques. For example, a reinforcement learning based hyper-heuristic assigns a utility score for each heuristic which is increased using a rewarding mechanism after improvement or decreased as a punishment mechanism after a worsening move [11]. A heuristic is chosen based on this score which then gets updated at each step. Different strategies can be utilized for heuristic selection, one of them being selection of the low level heuristic with maximum score. There is theoretical [12] as well as empirical evidence [13] that hyper-heuristics are effective solution methodologies.

We investigated different heuristic selection methods and hyper-heuristics. A simplified variant of a reinforcement learning based hyper-heuristic is employed, in this study. This variant uses different success criteria based on two successive stages and

```
IF IPCepoch(i) >= IPCepoch(i-1) THEN
    Keep the current heuristic running for the next epoch
ELSE
    Change the heuristic
ENDIF
```

Fig. 3 The pseudocode for HH1

```
commitOverFetch(i) ←IPCepoch(i) / FIPCepoch(i)
IF commitOverFetch(i) >= commitOverFetch(i-1) THEN
    Keep the current heuristic running for the next epoch
ELSE
    Change the heuristic
ENDIF
```

Fig. 4 The pseudo code for HH2

also different heuristic selection mechanisms to choose a low level heuristic at each step. The success measure is used as the utility score of a heuristic.

Hyper-Heuristic 1 (HH1): Our first hyper-heuristic is based on *committed instructions per cycle per epoch* (IPCepoch$_i$) metric. This success measure is a good indicator for the processor performance during an epoch. When this value is decreasing in the current epoch, we can directly say that something is going wrong. In such cases, HH1 punishes the heuristic that is used in the previous epoch and changes it with an alternative heuristic for the incoming epoch. In our study, we only utilized two heuristics (ARPA and HILL), and, hence, we choose the alternative heuristic in such cases. Figure 3 shows the pseudocode for this hyper-heuristic.

Hyper-Heuristic 2 (HH2): The second hyper-heuristic is based on a different metric which we call *commit over fetch*, as shown in Fig. 4. Generally, the number of instructions that enters the processor may not match the number of instructions that exits the processor by a successful completion. IPCepoch$_i$ value can be equal to but generally much less than the *fetched instructions per cycle per epoch* (FIPCepoch$_i$). The main reason for this phenomenon is due to the speculative nature of today's processors. To improve the processor throughput, the processors run instructions in an out of program order and have hardware branch predictors that may fill the processor pipeline from speculative paths. When the branch outcome is incorrectly predicted, instructions that are fetched from the wrong path are all flushed. Here, in this metric, we measure if the number of flushed instructions are increasing. When this happens, HH2 punishes the previously utilized heuristic by selecting the alternative heuristic.

Hyper-Heuristic 3 (HH3): Our final proposed hyper-heuristic design for performance optimization of simultaneous multithreaded processors is based on the random permutation gradient hyper-heuristic. Here, we propose a slightly complex evaluation

```
IF IPCepoch(i) / IPCepoch(i-1) >thresholdValue THEN
    Keep the current heuristic running for the next epoch
oneMoreChance← 0
ELSE
oneMoreChance++
    IF oneMoreChanceis 1 AND the current heuristic is ARPATHEN
        Giving one more chance to the current heuristic
    ELSE
commitOverFetch(i) ←IPCepoch(i) / FIPCepoch(i)
        IF commitOverFetch(i) >= commitOverFetch(i-1) THEN
           Keep the current heuristic running for the next epoch
        ELSE
           Change the heuristic
oneMoreChance← 0
        ENDIF
    ENDIF
ENDIF
```

Fig. 5 The pseudo code for HH3

function. First, we check if the IPCepoch_i improves as we do in HH1 with a minor twist. By adding a *threshold value* to the algorithm, we want to tolerate the small fluctuations in the performance due to external factors (phase changes in threads, increased cache steals among threads, etc.), which are not related to the performance of the running heuristic. Secondly, in our study, we observed that the overall performance may radically drop in a number of epochs. To stabilize our algorithm further, we give one more chance to the running heuristic, if it is ARPA, even when the drop in IPCepoch_i is below our threshold value. In our experiments, we found that ARPA is a more successful heuristic compared to HILL, and this is for an insurance not to punish a well-performing heuristic, mistakenly. Finally, as in HH2, we check if the efficiency of the last epoch is not worse than the efficiency of its predecessor. If this is the case, then we continue using the same heuristic; otherwise, we change the heuristic. The pseudo code of the algorithm is given in Fig. 5.

3 Computational Experiments

3.1 Processor Specifications

M-Sim [13] is used in our study to simulate the SMT processor. M-Sim is modified to support ARPA and HILL in any epoch in order to run these heuristics in mixed order. We arbitrarily chose 7 benchmarks from SPEC2000 benchmark suite in order to evaluate our work. We ran our tests for all 35 possible 3-thread mixtures consisting

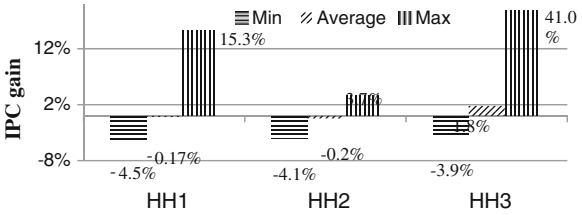

Fig. 6 The worst, the average and the best results of hyper-heuristics over ARPA

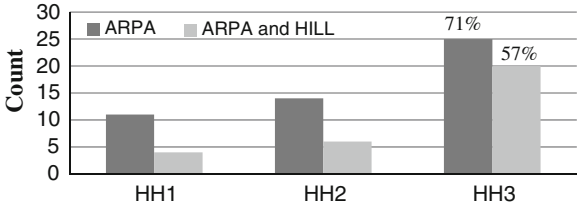

Fig. 7 Number of workloads in which hyper-heuristics perform better than heuristics

of these benchmarks. 10 M instructions from each thread are skipped before per-cycle simulation begins and it stops after 5 Mcycles. The epoch size is set to 32 Kcycles.

The simulated processor can decode/issue/commit 8 instructions per cycle. Reorder buffer, issue queue and load/store queue sizes are 64, 40 and 32 entries, respectively. There are 128 integer and 128 floating point registers. L1 instruction cache is 2-way with 32 KB capacity and L1 data cache is 4-way with 32 KB capacity. The L2 cache is unified; 512 KB in size and it is 4-way. All caches use least recently used replacement policy. Access to L2 cache is 20 cycles. The main memory has 2 ports and access time is 300 cycles for the first chunk and inter-chunk access delay is 6 cycles.

3.2 Tests and Results

Figure 6 shows the average and peak results of our proposed hyper-heuristics compared to ARPA. We are not comparing our results to HILL, since its results are generally worse than the results of ARPA. As the result clearly indicates, none of the hyper-heuristic is no worse than ARPA, and, suprisingly, HH3 performs 1.8% better than ARPA and 2.4% better than HILL (not shown on graph), on the average. The peak performance values on a few workloads seem to be very promising, as well (41% IPC gain on *art-bzip2-mgrid* mixture).

In Fig. 7, we show the number of workloads that perform better than ARPA (the leftmost bar) and number of workloads better than both ARPA and HILL (the rightmost bar) out of 35 workloads. Again, the best performing hyper-heuristic is

HH3 which performs better on 25 workloads over ARPA, and 20 workloads over both ARPA and HILL.

4 Conclusion and Future Work

In this study, we investigate the performance of some selection hyper-heuristics that mix multiple heuristics with different abilities. The results show that best performing hyper-heuristic is better than the well-known approaches in almost 60 % of the studied workloads. Moreover, it generates as much as 41 % of performance improvement in some workloads with 1.8–2.4% over ARPA and HILL on average. The results are very promising, and we believe that there is still room for research to design better hyper-heuristics in this area.

References

1. Tullsen DM, Eggers SJ, Emer JS, Levy HM, Lo JL, Stamm RL (1996) Exploiting choice: instruction fetch and issue on an implementable simultaneous multithreading processor. In: 23rd annual International symposium on computer architecture, New York, USA, pp 191–202
2. Tullsen DM, Brown JA (2001) Handling long-latency loads in a simultaneous multithreading processor. In: 34th annual ACM/IEEE international symposium on microarchitecture. IEEE Press, Washington, DC, pp 318–327
3. Cazorla FJ, Ramirez A, Valero M, Fernandez E (2004) Dynamically controlled resource allocation in SMT processors. In: 37th annual IEEE/ACM international symposium on microarchitecrure. IEEE Press, Washington, DC, pp 171–182
4. Choi S, Yeung D (2006) Learning-based SMT processor resource distribution via hill-climbing. In: 33rd annual international symposium on computer architecture. IEEE Press, , Washington, DC, p- 239–251
5. Wang H, Koren I, Krishna CM (2008) An adaptive resource partitioning algorithm for SMT processors. In: 17th international conference on parallel architectures and compilation techniques, NY, USA, pp 230–239
6. Vandierendonck H, Seznec A (2011) Managing SMT resource usage through speculative instruction window weighting. ACM Trans Arch Code Opt 8(3):12
7. Eyerman S, Eeckhout L (2009) Memory-level parallelism aware fetch policies for simultaneous multithreading processors. ACM Trans Arch Code Opt 6(1):3
8. Burke EK, Hyde M, Kendall G, Ochoa G, Özcan E, Woodward JR (2010) A classification of hyper-heuristic approaches. In: Kochenberger GA (ed) Handbook of metaheuristics. Springer, London (Vol. 146 of Intl Series Opt Res Man Sci)
9. Özcan E, Bilgin B, Korkmaz EE (2008) A comprehensive analysis of hyper-heuristics. Intell Data Anal 12(1):3–23
10. Cowling P, Kendall G, Soubeiga E (2001) A hyperheuristic approach to scheduling a sales summit. In: Selected papers from the third international conference on practice and theory of automated timetabling. Springer, London, pp 176–190
11. Nareyek A (2003) Choosing search heuristics by non-stationary reinforcement learning. In: Resende MG et al (eds) Metaheuristics: computer decision-making. Kluwer, Dordecht, pp 523–544

12. Lehre PK, Özcan E (2013) A runtime analysis of simple hyper-heuristics: to mix or not to mix operators. In: Pre-conference on proceedings of foundations of general algorithms XII, pp 91–98
13. Burke EK, Gendreau M, Hyde M, Kendall G, Ochoa G, Özcan E, Qu R (2013) Hyper-heuristics: a survey of the state of the art. J Oper Res Soc (to appear)
14. Sharkey JJ, Ponomarev D, Ghose K (2005) M-SIM: a flexible, multithreaded architectural simulation environment. Department of Computer Science, Binghamton University, Technical Report No.CS-TR-05-DP01

A Model of Speculative Parallel Scheduling in Networks of Unreliable Sensors

Zhan Qiu and Peter G. Harrison

Abstract As systems scale up, their mean-time-to-failure reduces drastically. We consider parallel servers subject to permanent failures but such that only one needs to survive in order to execute a given task. This kind of failure-model is appropriate in at least two types of systems: systems in which repair cannot take place (e.g. spacecraft) and systems that have strict deadlines (e.g. navigation systems). We use multiple replicas to perform the same task in order to improve the reliability of systems. The server in the system is subject to failure while it is on and the time to failure is memoryless, i.e. exponentially distributed. We derive expressions for the Laplace transform of the sojourn time distribution of a tagged task, jointly with the probability that the tagged task completes service, for a network of one or more parallel servers with exponential service times and times to failure.

1 Introduction

Analysis of fault-tolerant systems has attracted renewed interest in view of recent trends such as harsh environments, real-time systems and downtime costs, where servers that are subject to breakdowns may have a serious impact on system performance and reliabiity. For example, industrial applications usually have strict reliability requirements, since faults may lead to economic losses, environmental damage or even personal injury. Fault-tolerant systems have for long been constructed to reduce downtime and to ensure high availability, high-performance, and that critical computations are not compromised [8]. In this paper, we consider systems with unreliable

Z. Qiu (✉) · P. G. Harrison
Department of Computing, Imperial College London, Huxley Building,
180 Queen's Gate, London SW7 2AZ, UK
e-mail: zq11@doc.ic.ac.uk

P. G. Harrison
e-mail: pgh@doc.ic.ac.uk

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_11,

servers with permanent faults, where, once a server has failed, it will never recover. This kind of scenario can be found in at least two types of systems:

- In systems in which repair cannot take place. For example, in Wireless Sensor Networks (WSNs), sensors are prone to failure for several reasons: a node may die due to battery depletion, or may be destroyed accidentally, or may be incapacitated by a malicious adversary. What makes things worse is that in most cases, faulty sensors cannot be easily repaired or replaced promptly since those networks may be employed in hostile environments that are too dangerous for a human to access, such as disaster sites, a polluted area or a hazardous chemical leak in a building [7].
- A system is often required to respond to user actions within strict time constraints—e.g. subject to a hard deadline [1]. In real-time systems, even a small amount of downtime due to repairs cannot be tolerated, such as airline reservation systems, navigation systems and flight-control computers. Failures in real-time systems can result in data corruption and lower performance, leading to catastrophic failures. Thus we can treat failures in these systems as “permanent failures" and need to develop strategies to make such systems fault-tolerant.

In order to reach the goal of reliable computing with respect to permanent failures discussed in this paper, one approach is to use protective redundancy. Nowadays, a broad class of systems deploy multiple servers as replicas to perform the same task and, therefore, become an effective approach to both higher performance and improving the reliability of service-oriented applications [2]. As noted above, one example is using replicas in systems with servers that work in space, e.g. spacecraft. The Mars rover Curiosity experienced its first significant malfunction in march 2013, and has been forced to use a backup computer to take over the primary operations [9].

Replicas that provide basic (not composite) services in themselves are assumed to be independent, i.e., the success or failure of one replica is independent of the others [11]. Here, we use independent replicas to form a reliable whole out of less-reliable parts, whilst also reducing the sojourn time of the system. A task is sent to multiple replicas and its result is that returned by the server that responds first. Multi-server queueing systems with server breakdowns are more flexible and applicable in practice than single server counterparts. When some of the servers fail, the system doesn’t fail and we can use the result from other replicas. In this paper, we use analytical methods to obtain the probability distributions of sojourn times with servers having exponential service times: for systems with m servers for $m \geq 1$, the case $m = 1$ being a standard calculation, of course.

The rest of the paper is organized as follows: Section 2 gives a description of the system we consider and the method we use to model it. In Sect. 3, the Laplace transform of sojourn time distribution, together with its mean and variance, is obtained for an M/M/1 queue with unreliable server. In Sect. 4, analogous performance measures for systems with two parallel servers are derived. Finally we derive the corresponding result for systems with multiple parallel servers in Sect. 5 and conclude in Sect. 6.

2 Model Discription

A server may encounter a failure when it is not idle and we assume the time to occurrence of a failure in the server, X, is exponentially distributed with parameter $\alpha(\alpha > 0)$; hence, crucially, the time to failure is memoryless. The corresponding probability density function (pdf) of X is

$$f(t) = \begin{cases} \alpha e^{\alpha t} & (t > 0) \\ 0 & (t \leq 0) \end{cases} .$$

The *reliability function* $R(t)$ at any time t, i.e. the probability that the system will survive until time t, is defined as

$$R(t) = \text{Prob}\{X > t\} = 1 - \text{Prob}\{X \leq t\} = 1 - F(t) = e^{-\alpha t} .$$

Because of the assumed memoryless property, a device that has survived to time t is as good as new in terms of its remaining life [10].

A *tagged customer* is a special task (synonymously, customer), the pdf of sojourn time of which we seek, where the sojourn time T of a customer is the time interval that elapses from the arrival of the customer to his departure from the server. Let $T_R(t) = P(T \leq t)$ denote the joint probability that a tagged customer arriving at equilibrium completes service and has sojourn time that does not exceed a positive real value t. The probability that the sojourn time of a customer is finite is not 1, since there is a non-zero probability that the server fails before the customer completes service. The probability that a customer completes service is the marginal probability $T_R(\infty)$.

In our analysis, we consider the probability that the sojourn time of the tagged customer exceeds a positive value t at equilibrium, $P(T > t)$, by deriving the Laplace Transform $L(s)$ of $T_R(t)$ and hence the mean sojourn time $E[T \mid T < \infty]$. Note that the unconditional mean sojourn time is infinite, but we derive the expected sojourn time of customers that do complete service.

3 M/M/1 Single Server Queue with Unreliable Servers

For an M/M/1 queue, customers arrive to the system according to a Poisson process with rate $\lambda > 0$, and service time is exponentially distributed with parameter $\mu > 0$. The queueing discipline is FCFS, with no more than one customer being served at any given time. When calculating the probability that the sojourn time of the tagged customer exceeds t after arriving at a working system, we need to consider two possibilities:

- the server is still functioning after time t and the customer hasn't completed service by that time, i.e., the customer is waiting in the queue or being served;

- the server encounters a failure before time t and the tagged customer hasn't completed service before the failure.

We denote the probabilities for the above two cases by $P_{work}(t)$ and $P_{fail}(t)$, respectively. First we denote the cumulative distribution function (CDF) of the sojourn time of a tagged customer in a queue without server failure as $W_r(t) = P(T \leq t)$. For an M/M/1 queue [5], $W_r(t) = 1 - e^{-(\mu-\lambda)t}$, thus

$$P_{work}(t) = R(t)(1 - W_r(t)) = e^{-(\alpha+\mu-\lambda)t}. \tag{3.1}$$

In order to obtain $P_{fail}(t)$, we assume the server fails at time point $t_1 (0 \leq t_1 \leq t)$, and assume that the sojourn time of the customer is longer than t_1, thus

$$P_{fail} = \int_0^t f(t_1) \int_{t_1}^{\infty} w_r(t_2) \mathrm{d}t_2 \mathrm{d}t_1 = \frac{\alpha}{\alpha + \mu - \lambda}(1 - e^{-(\alpha+\mu-\lambda)t}). \tag{3.2}$$

Therefore, the joint probability that the tagged customer can complete service and his sojourn time does not exceed a positive real value t is

$$P(T \leq t) = 1 - \left(P_{work} + P_{fail}\right) = \frac{\mu - \lambda}{\alpha + \mu - \lambda}(1 - e^{-(\alpha+\mu-\lambda)t}). \tag{3.3}$$

The probability that the tagged customer completes service is the marginal probability $(\mu - \lambda)/(\alpha + \mu - \lambda)$. The Laplace Transform of the sojourn time, conditioned on the tagged customer completing service, is

$$L(s) = \frac{\alpha + \mu - \lambda}{s + \alpha + \mu - \lambda} \tag{3.4}$$

and the expected sojourn time for customers that complete service is

$$E[T] = \frac{1}{\alpha + \mu - \lambda}. \tag{3.5}$$

This result may appear a little surprising at first sight since we know that the tagged customer's service time must have mean μ^{-1}. However, we need to remember that the result we get is not the mean sojourn time for every tagged customer, only those who complete service before the failure of the server; these must be quicker! If we set $\alpha = 0$, then we obtain that $L(s) = (\mu - \lambda)/(s + \mu - \lambda)$ and $E[T] = 1/(\mu - \lambda)$, which is just the mean sojourn time for an M/M/1 queue.

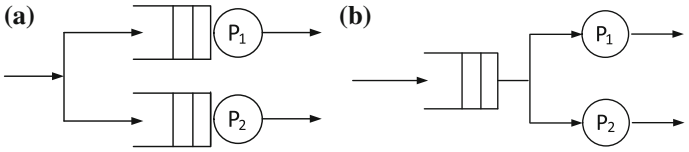

Fig. 1 System with two parallel servers

4 Sojourn Time of a System with Two Parallel Servers

Following the result of Sect. 3, we now consider the scenario where the same request is issued to multiple replicas and the result is taken from whichever replica responds first; as illustrated in Fig. 1a. Let the failure density distributions and reliability functions of the multiple parallel servers be:

$$\mathbf{f} = \{f_1(t), f_2(t), \ldots, f_n(t)\} \ , \ \mathbf{R} = \{R_1(t), R_2(t), ..., R_n(t)\}.$$

Two servers are scheduled simultaneously, and a customer will complete if at least one server does not encounter a failure before his completion instant. This model allows the servers to (instantaneously) communicate updates on the status of their copies to each other. The system then cancels remaining outstanding requests once the first result is received. Thus, whilst the two servers are both functioning, they can be treated as if they are sharing a single queue. If one of the servers fails, then the model simplifies to an M/M/1 queue. Thus, our model is equivalent to a system with only one single queue in front of two independent, failure-prone servers, as shown in Fig. 1b.

When a customer arrives at the system and finds the two servers functioning, there are three ways in which he cannot complete in a time period t:

- both servers remain functioning and the customer doesn't complete service in both servers;
- one of the servers remains functioning, but the customer hasn't completed; the other server has already failed;
- both the servers fail before the customer has completed.

Let w_{r_i} and W_{r_i} be the sojourn time density function and CDF in the case that i out of n servers are still functioning, respectively; in this section $n = 2$. If both the two servers are functioning, then the sojourn time of a tagged customer in this model is the queueing time plus the minimum of the two servers' service times, as follows:

$$W_{r_2}(t) = W_{r_{min}}(t) = 1 - e^{-(\mu_1+\mu_2-\lambda)t} \ .$$

when one of the servers fails, i.e., $n = 2, i = 1$, we have

$$W_{r_1}(t) = 1 - e^{-(\mu_i-\lambda)t} \ .$$

where μ_i is the service rate of the server that is still functioning. Assume the two servers are not identical, then first we consider the probability that server P_1 fails before the task completes and the task is still waiting or being served in server P_2. Before P_1 fails, the response time density function is $w_{r_2}(t) = w_{r_{min}}(t) = (\mu_1+\mu_2-\lambda)e^{-(\mu_1+\mu_2-\lambda)t}$; after P_1 fails, the response time density function changes to $w_{r_1}(t) = (\mu_2 - \lambda)e^{-(\mu_2-\lambda)t}$, thus

$$\begin{aligned} P_{f_1\&w_2}(t) &= R_2(t) * \int_0^t f_1(t_1)\Big(\int_{t_1}^{\infty} w_{r_{min}}(x_2)\mathrm{d}x_2\Big)\Big(\int_{t-t_1}^{\infty} w_{r_1}(x_1)\,\mathrm{d}x_1\Big)\mathrm{d}t_1 \\ &= \frac{\alpha_1}{\alpha_1+\mu_1}(e^{-(\alpha_2+\mu_2-\lambda)t} - e^{-(\alpha_1+\alpha_2+\mu_1+\mu_2-\lambda)t}). \end{aligned}$$

Similarly, we can obtain the probability that server P_2 fails before the task completes and the task is still waiting or being served in server P_1. Next we consider the probability that the task doesn't complete before both the servers fail. If P_1 fails before P_2, we have

$$\begin{aligned} P_{f_1<f_2}(t) &= \int_0^t f_1(t_1)\Big(\int_{t_1}^{\infty} w_{r_{min}}(x_2)\mathrm{d}x_2\Big)\int_{t_1}^{t} f_2(t_2)\Big(\int_{t_2-t_1}^{\infty} w_{r_1}(x_1)\mathrm{d}x_1\Big)\mathrm{d}t_2\mathrm{d}t_1 \\ &= \frac{\alpha_1\alpha_2}{(\alpha_2+\mu_2-\lambda)(\alpha_1+\alpha_2+\mu_1+\mu_2-\lambda)}(1 - e^{-(\alpha_1+\alpha_2+\mu_1+\mu_2-\lambda)t}) \\ &\quad + \frac{\alpha_1\alpha_2}{(\alpha_1+\mu_1)(\alpha_2+\mu_2-\lambda)}(e^{-(\alpha_1+\alpha_2+\mu_1+\mu_2-\lambda)t} - e^{-(\alpha_2+\mu_2-\lambda)t}) \ . \end{aligned}$$

Similarly, we can obtain the probability for the case that P_2 fails before P_1. Therefore, the probability that the sojourn time of the tagged customer exceeds a postive value t in equilibrium is

$$P(T > t) = \underbrace{R_1(t) * R_2(t) * (1 - W_{r_{min}}(t))}_{\text{both servers are functioning}} + \underbrace{P_{f_1\&w_2} + P_{w_1\&f_2}}_{\text{one of the servers fails}} + \underbrace{P_{f_1<f_2} + P_{f_1>f_2}}_{\text{both servers fail}} .$$

and the joint probability that the tagged customer completes service and that his sojourn time does not exceed the positive real value t is

$$\begin{aligned} P(T \le t) = 1 &- \frac{\alpha_1\alpha_2(\alpha_1+\alpha_2+\mu_1+\mu_2-2\lambda)}{(\alpha_1+\mu_1-\lambda)(\alpha_2+\mu_2-\lambda)(\alpha_1+\alpha_2+\mu_1+\mu_2-\lambda)} \\ &- \frac{\alpha_2(\mu_1-\lambda)}{(\alpha_2+\mu_2)(\alpha_1+\mu_1-\lambda)}e^{-(\alpha_1+\mu_1-\lambda)t} \\ &- \frac{\alpha_1(\mu_2-\lambda)}{(\alpha_1+\mu_1)(\alpha_2+\mu_2-\lambda)}e^{-(\alpha_2+\mu_2-\lambda)t} \\ &- \frac{\mu_1\mu_2(\mu_1+\mu_2+\alpha_2-\lambda)+\alpha_1(\mu_1\mu_2+\alpha_2\lambda)}{(\alpha_1+\mu_1)(\alpha_2+\mu_2)(\alpha_1+\alpha_2+\mu_1+\mu_2-\lambda)}e^{-(\alpha_1+\alpha_2+\mu_1+\mu_2-\lambda)t} \ . \end{aligned} \tag{4.6}$$

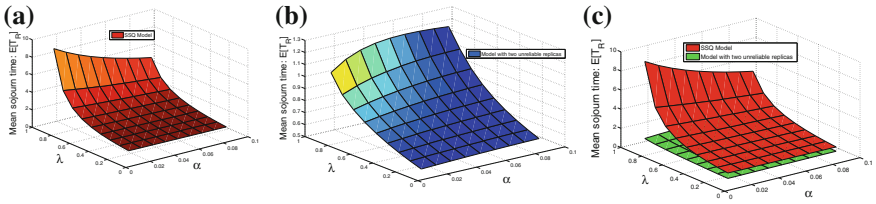

Fig. 2 Illustration that the mean sojourn time in the system with two replicas is less than in the M/M/1 model. **a** Mean sojourn time of the M/M/1 model. **b** Mean sojourn time of system with two replicas. **c** Comparison of mean sojourn time

when the two servers are identical, we could simplify the above equation as

$$P(T \leq t) = 1 - e^{-(2\alpha+2\mu-\lambda)t} - \frac{2\alpha}{\alpha+\mu}(e^{-(\alpha+\mu-\lambda)t} - e^{-(2\alpha+2\mu-\lambda)t}) - \frac{2\alpha^2}{(\alpha+\mu-\lambda)(2\alpha+2\mu-\lambda)}(1 - e^{-(2\alpha+2\mu-\lambda)t}) - \frac{2\alpha^2}{(\alpha+\mu)(\alpha+\mu-\lambda)}(e^{-(2\alpha+2\mu-\lambda)t} - e^{-(\alpha+\mu-\lambda)t}) \ . \tag{4.7}$$

As $t \to \infty$, the term $1 - 2\alpha^2/((\alpha+\mu-\lambda)(2\alpha+2\mu-\lambda))$ is the probability that the tagged customer completes service in this model, as expected. Since the probability that a customer completes service in a system with both servers functioning is $(2\mu - \lambda)/(2\alpha + 2\mu - \lambda)$, and the probability in a system with only one of the servers still functioning is $(\mu - \lambda)/(\alpha + \mu - \lambda)$, the unconditional probability that a customer completes service is $1 - (1 - (2\mu - \lambda)/(2\alpha + 2\mu - \lambda))(1 - (\mu - \lambda)/(\alpha + \mu - \lambda)) = 1 - 2\alpha^2/((\alpha + \mu - \lambda)(2\alpha + 2\mu - \lambda))$, also as expected.

It follows that the Laplace Transform of the sojourn time, conditioned on a customer completing service is

$$L(s) = \frac{(\alpha(3\lambda - 4\mu) - (\lambda - 2\mu)(\lambda - s - \mu))(\alpha + \mu - \lambda)(2\alpha + 2\mu - \lambda)}{(s + \alpha + \mu - \lambda)(s + 2\alpha + 2\mu - \lambda)(3\alpha\lambda - \lambda^2 - 4\alpha\mu + 3\lambda\mu - 2\mu^2)} \tag{4.8}$$

and the mean value of the sojourn time of a customer that completes service, is

$$E[T] = \frac{\alpha^2(7\lambda - 8\mu) - 2\alpha(5\mu^2 + 3\lambda^2 - 8\mu\lambda) + (\mu - \lambda)^2(\lambda - 2\mu)}{(2\alpha + 2\mu - \lambda)(\alpha + \mu - \lambda)(3\alpha\lambda + 3\lambda\mu - 4\alpha\mu - \lambda^2 - 2\mu^2)} \ . \tag{4.9}$$

As a check, setting the failure rate $\alpha = 0$ yields $L(s) = (2\mu - \lambda)/(s + 2\mu - \lambda)$ and $E[T] = 1/(2\mu - \lambda)$, which are just the measures for a system without failures.

It can be seen from Fig. 2a and b that for both the M/M/1 model and the two replicas model, with the increase of the load on the system, the mean sojourn time increases. This is as expected since, with the increase in the load, customers spend

more time waiting for service. Actually, issuing the same customer to multiple servers not only increases the reliability of the system, but also reduces the response time. Figure 2c shows the mean sojourn time in the model with two replicas less than in the M/M/1 model.

5 Sojourn Time in Systems with Multiple Servers

We now extend the model with two servers to a system with $N \geq 2$ multiple, identical replicas. Let w_{r_i} be the sojourn time density function and P_i be the probability that the response time exceeds a positive value t when i out of n servers are still functioning. Thus

$$w_{r_i}(t) = (i\mu - \lambda)e^{-(i\mu-\lambda)t} \ .$$

and

$$P_0 = \int_0^t f(t_1)\Big(\int_{t_1}^{\infty} w_{r_n}(x_n)\mathrm{d}x_n\Big)\ldots\Big(\int_{t_{n-1}}^{t} f(t_n)\int_{t_n-t_{n-1}}^{\infty} w_{r_1}(x_1)\mathrm{d}x_1\Big)\mathrm{d}t_n\ldots\mathrm{d}t_2\mathrm{d}t_1,$$

for $i = 0$.

$$P_{n-1} = R(t)^{n-1}\int_0^t f(t_1)\Big(\int_{t_1}^{\infty} w_{r_n}(x_n)\mathrm{d}x_n\Big)\Big(\int_{t-t_1}^{\infty} w_{r_{n-1}}(x_{n-1})\mathrm{d}x_{n-1}\Big)\mathrm{d}t_1,$$

for $i = n-1$.

$$P_i = R(t)^i\int_0^t f(t_1)\Big(\int_{t_1}^{\infty} w_{r_n}(x_n)\mathrm{d}x_n\Big)\Big(\int_{t_1}^{t} f(t_2)\int_{t_2-t_1}^{\infty} w_{r_{n-1}}(x_{n-1})\,\mathrm{d}x_{n-1}\Big)\ldots$$

$$\Big(\int_{t_{n-i-1}}^{t} f(t_{n-i})\int_{t_{n-i}-t_{n-i-1}}^{\infty} w_{r_{i+1}}(x_{i+1})\mathrm{d}x_{i+1}\Big)\Big(\int_{t-t_{n-i}}^{\infty} w_{r_i}(x_i)\mathrm{d}x_i\Big)\mathrm{d}t_i\ldots\mathrm{d}t_1,$$

for $i < n-1$.

$$P_n = e^{-(n\alpha+n\mu-\lambda)t}, \text{ for } i = n.$$

Thus, the joint probability that at least one of the servers reamains working and the response time doesn't exceed the positive value t is

$$P(T \leq t) = 1 - P_n - n!P_0 - \sum_{i=1}^{n-1}\binom{n}{i}(n-i)!P_i \ . \tag{5.10}$$

5.1 Example

Let $n = 2$, and $\alpha_1 = \alpha_2 = 0.5$, $\lambda = 0.9$, $\mu = 1$. Then the mean sojourn time $E[T] = 0.6955$ for customers who complete service, equal to the result we obtained for the two-server case above. For $n = 3$, and $\alpha_1 = \alpha_2 = \alpha_3 = 0.05$, $\lambda = 0.5$, $\mu = 1$, we obtain the mean sojourn time $E[T] = 0.4134$ using the above method.

6 Conclusion

Speculative replication of tasks is commonly used to enhance system performance and reliability. We have developed analytical methods to find the sojourn time probability distribution of successful tasks in an abstract model where replicas are submitted to independent parallel servers, the first to finish providing the result. We considered the case where failed servers could not be repaired and so were lost after a failure. In an alternate scenario, rather than a server failing a task fails, as in a parallel database search or in a system with mirroring, for example. In this case, on a failure, only the task at the front of a particular queue is lost—the server continues processing the next task in its queue. Such systems can be modelled using the notion of negative customers: not only do these remove the task at the front of a queue on a failure, a completing task sends killing signals to remove any replicas that are still incomplete [3, 4, 6]. A model based on these ideas is currently under development by the authors, with a view to optimising not only performance but also energy saving; there is a trade-off between shorter response times achieved by replication and the increased energy required in utilising more resources.

References

1. Ben-Ari M (2006) Principles of concurrent and distributed programming. Addison-Wesley Longman, Boston
2. Dean J, Barroso LA (2013) The tail at scale. Commun ACM 56(2):74–80
3. Gelenbe E (1989) Random neural networks with positive and negative signals and product form solution. Neural Comput 1(4):502–510
4. Gelenbe E (1993) G-networks with triggered customer movement. J Appl Prob 30:742–748
5. Harrison PG, Patel NM (1992) Performance modelling of communication networks and computer architectures (International Computer S. Addison-Wesley Longman, Boston
6. Harrison P, Pitel E (1993) Sojourn times in single server queues with negative customers. J Appl Prob 30:943–963
7. Macedo DF, Correia LH, dos Santos AL, Loureiro AA, Nogueira JMS, Pujolle G (2006) Evaluating fault tolerance aspects in routing protocols for wireless sensor networks. Challenges in Ad Hoc Networking, Springer, Berlin, In, pp 285–294
8. Maxion RA, Siewiorek DP, Elkind SA (1987) Techniques and architectures for fault-tolerant computing. Ann Rev Comput Sci 2(1):469–520

9. Nathan (2013) Nasas mars rover curiosity forced to backup computer as result of computer glitch. http://planetsave.com/2013/03/03/nasas-mars-rover-curiosity-forced-to-b%ackup-computer-as-result-of-computer-glitch/
10. Stewart WJ (2011) Probability, Markov chains, queues, and simulation: the mathematical basis of performance modeling. Princeton University Press, New Jersey
11. Tang C, Li Q, Hua B, Liu A (2009) Developing reliable web services using independent replicas. In: Fifth International Conference on Semantics, Knowledge and Grid (SKG 2009) IEEE. pp 330–333

Energy-Aware Admission Control for Wired Networks

Christina Morfopoulou, Georgia Sakellari and Erol Gelenbe

Abstract Overprovisioning and redundancy has contributed towards better survivability and performance in networks, but has led to inefficient use of energy. Proposals for energy aware networks of the near future aim to reduce the energy consumption by switching off or putting to sleep individual network devices. Here we propose a mechanism that is taking this concept once step further through the use of admission control. Admission control has been traditionally used in wired networks to control traffic congestion and guarantee quality of service. We propose a two-fold approach. First, an admission control mechanism delays the users that are projected to be the most energy demanding, and whose acceptance would require the turning on of devices. At the same time, an auto-hibernation mechanism regulates the rate at which machines are turned off due to inactivity. Collectively, the two mechanisms contribute towards energy saving by monitoring both at the level of entry in the network and at the level of active operation.

1 Introduction

The carbon imprint of ICT technologies is estimated to be over 2 % of the world total, similar to that of air travel [5]. Yet, research on the energy consumption of ICT systems and its backbone, the wired network infrastructure, is still at an early stage. Until recently, energy saving in networks focused more on longer battery life and

C. Morfopoulou · E. Gelenbe (✉)
Electrical and Electronic Engineering Department, Imperial College London, ISN Group, LondonSW7 2BT, UK
e-mail: c.morfopoulou@imperial.ac.uk

G. Sakellari
Computer Science Field, School of ACE, University of East London, London E16 2RD, UK
e-mail: g.sakellari@uel.ac.uk

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_12,

smart wireless network design [6] and more attention has been devoted to energy consumption of Cloud Computing [1].

The mechanism proposed in this paper acknowledges the need for energy awareness in networks and the design of network components. By leveraging the admission control idea, traditionally used in networks to control traffic congestion, we propose an energy aware admission control mechanism that takes advantage of the future network devices that will have the capability to be temporarily switched off or transit to a sleep state. More specifically, in our experiments we examine the potential savings in energy by turning off nodes that have been inactive for a long enough period of time and using admission control to determine whether a user should be accepted into the network. The admission control mechanism is responsible to make sure that all nodes of the required path are ON before admission of the new user and also try to delay users before entering the network in order to increase efficiency. Performance investigations show savings up to 20 % in the total network power consumption revealing that this idea of admission control can be of large importance on top of energy saving mechanisms of future network devices.

The remainder of this paper is organized as follows: the next section refers to the related existing work, both in respect to admission control and the behaviour of power consumption of wired networks; Sect. 3 presents our proposed energy-aware admission control algorithm; Sect. 4 reports the configuration of the experiments and experimental results of the algorithm achieved on a real testbed; finally we comment on the results in Sect. 5 and we conclude the paper in Sect. 6.

2 Previous Work

2.1 Admission Control

Admission control in wired networks has been traditionally used as a way to control traffic congestion and guarantee QoS [13]. The metrics considered in the decision of whether to accept a new flow into a network are mainly bitrate, delay, packet loss and jitter [9]. To the best of our knowledge, there is no work proposing the use of energy as a criterion to an admission control algorithm in wired networks. In our recent work [14], we examined the possibility of using energy criteria to admit users in the network using an ideal power profile of the nodes. This paper examines such an admission control algorithm in experiments using real online power measurements and by turning on/off nodes.

2.2 Power Consumption in Energy Aware Networks

As described in the survey of [2] several techniques have been recently proposed in order to enable energy efficiency in networks. Early work suggesting the idea of energy savings in the Internet [10] proposed routing modifications so as to aggregate

traffic along a few routes, leading to a modification of the network topology based on route adaptation and putting nodes and devices to sleep. The problem of energy aware routing is examined via an analytical approach in [8] where an optimization algorithm is built that minimizes a composite energy and QoS cost function. In [7] an autonomic algorithm that utilizes adaptive reinforcement learning is proposed that attempts to minimize the power consumption while meeting the requested end-to-end delay bounds. In [3] two widely used routers are measured in terms of system power demand based on different configurations of line cards and operating conditions. Traffic was shown to have only around 2 % impact on power consumption. A similar study is presented in [11], suggests that the impact of port utilization on power consumption is under 5 %. So, these studies indicate that the power consumption of current networking equipment is largely independent of its utilization and thus there is significant energy waste when the nodes are inactive or lightly loaded. In [12] a rate-adaptation for individual links is examined based on the utilization and the link queuing delay. Moreover, a sleeping approach is proposed, where traffic is sent out in bursts at the edge routers enabling other line cards to sleep between successive bursts of traffic. In [4] the authors select the active links and routers to minimize the power consumption via simple heuristics that approximately solve a corresponding NP-hard problem.

All this work, identifies the potential savings by turning on/off nodes but ignores implementation obstacles like the need for a mechanism that manages the network, the extra power needed to turn on/off a node and the induced delay thereof. In this work, we propose an Energy Aware Admission Control (EAAC) mechanism, evaluate its performance with experiments in a PC-based network topology and explore the implementation challenges and problems that arise.

3 Energy Aware Admission Control Mechanism

3.1 Problem Description

The existing work identifies the gains of turning devices off in order to increase energy efficiency in networks. This though, still entails huge challenges as the current networking equipment is not capable of entering and exiting a low-energy sleep mode. Moreover, turning a machine off and on comes with energy and delay costs. Most of the research conducted so far is based on simulations of the future behaviour of the networking equipment. In this work we use a real laboratory PC-based testbed with on/off capabilities. Thus, we are able to examine a real case scenario where the nodes are turned off when idle and they are informed to wake up where they are needed to route traffic. The power consumption is monitored in real time using a power meter and the delays induced to the new traffic flows are measured in order to examine the trade off associated with the Quality of Service (QoS) provided.

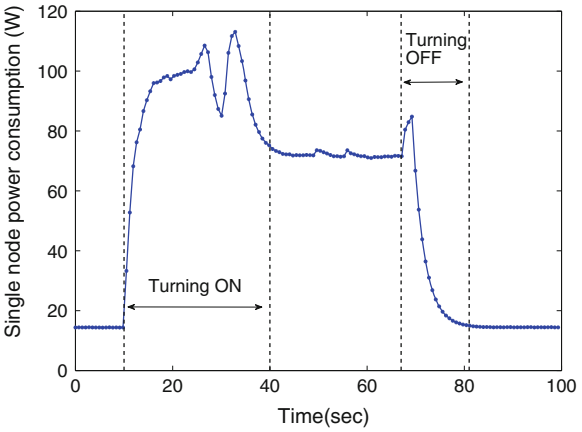

Fig. 1 Power consumption measurements when turning on (10–40 s) and off (67–80 s) a node

We first examined the behavior of our PC-based routers during their shutdown and wake up times. Figure 1 shows the power consumption of one of our testbed's nodes. It is initially turned off and then it receives a "wake on lan" packet (10th unit S). The process of "turning on" lasts from 10 to around 40 s and then the node is asked to turn off again at the 67th second and its finally off around 80 s. We can observe significant spikes in power consumption when turning on and off which result in a large energy waste, since the nodes are not processing any traffic during this time. So, when deciding to turn off a machine one has to take into consideration the additional energy spent for turning off and on and whether this is smaller than the savings during the time for which the node is off. Also, it is evident that the time needed to wake up a node in our PC-based routers and turn it off is quite long (we have measured an average of approximately 35 s for booting and 10 s for shutting down). Current networking research targets to introduce sleep modes in routers or individual router components and the relevant times should be significantly smaller in the future.

3.2 Algorithm Description

The steps of the proposed Energy Aware Admission Control (EAAC) algorithm are

1. A new user i informs the EAAC about its source s_i and destination d_i. It also sets a maximum time limit W_i that the user is willing to wait until it is admitted into the network.
2. The EAAC calculates the minimum hop path π_i from s_i to d_i and collects the information about the current state of these nodes (whether they are currently ON or OFF).

3. If all of the nodes on the shortest path are ON, the flow is admitted into the network, else the flow is sent to a waiting queue.
4. If the waiting time of a flow in the waiting queue W_i expires, the EAAC turns on the nodes on the path that are OFF. Once they are ON, the flow is admitted.
5. The requests in the waiting queue are occasionally re-evaluated for admittance.

Note that in parallel, an auto-hibernation mechanism is running on all nodes which turns off the machines when they are inactive for a specific amount of time.

3.3 Implementation Details

The auto-hibernation mechanism decides when to turn a node off based on whether the node has been inactive for a specific amount of time. The inactivity of a node is based on the amount of traffic that has been processed, which is being monitored by a built-in function of the node.

As for the 'waking up' mechanism; when the waiting time of a user request expires, the admission control mechanism sends 'wake on lan' packets to the nodes that are on the user's path and that are currently off. Note that a 'stay alive' flag is set on the nodes that are already ON, to prevent the auto-hibernation mechanism from turning them off, while trying to turn on the rest of nodes on the path.

4 Experiments

4.1 Configuration of the Experiments

In order to evaluate our mechanism we conducted our experiments on the real testbed located at Imperial College London. Our testbed consists of 12 PC-based routers as shown in Fig. 2 and the intermediate nodes are connected to a Watts up?. Net power meter. [1]

In the experiments we had 5 users corresponding to 5 Source–Destination (S–D) pairs independently making requests to send traffic into the network at random intervals. More specifically, we have the pairs (106,109), (103, 211), (109, 106) and (106, 103) which generate traffic every 150–200 s, with randomly distributed bandwidth request of 200–500 Kbps. Note that these source-destination pairs were selected to cover the whole network and not leave any nodes unused. We assume that all users are willing to wait at least for $W = T_{on}$. T_{on} is the time it takes to turn on a node, in case that one or more nodes of the required path are OFF at the time of the user's arrival. Moreover, we add a voluntary waiting time randomly distributed between 0 and 20 s, which is additional to the T_{on}, thus $W = T_{on} + [0 - 20]$. We

[1] http://www.wattsupmeters.com.

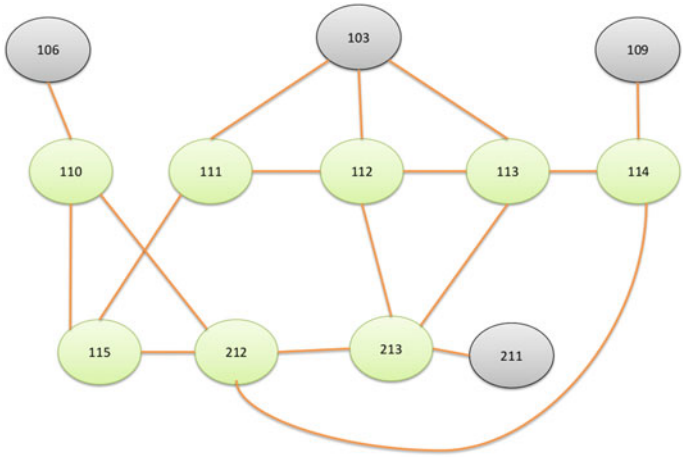

Fig. 2 Experimental topology. *Green nodes* can be turned off/on and their power consumption is being measured. *Grey nodes* are sources/destinations and are always on

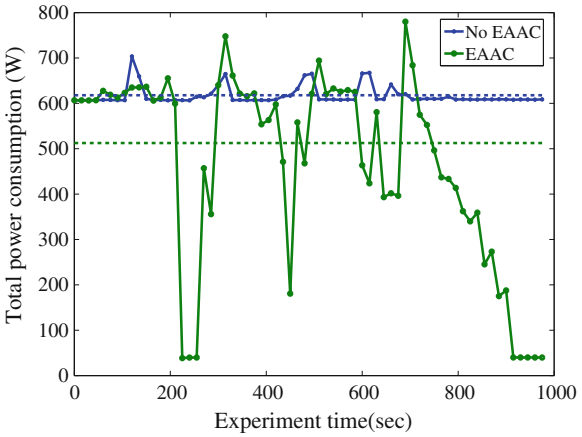

Fig. 3 Power consumption results. *Dotted lines* represent the average values. 17 % average power savings are observed when using EAAC

run the experiment for 1,000 s comparing the EAAC with the no admission control case, where the users are not willing to wait and thus all nodes are constantly ON, ready to carry traffic.

The total power consumption over time for both cases is shown in Fig. 3. As expected, the power consumption of the no admission control case where all nodes are ON, is almost constant. On the other hand, when using the EAAC we observe sharp falls from turning off the nodes. However, there are also significant spikes from the extra power needed to turn on nodes, as was investigated in Fig. 1.

The average value (measured until the 700 s for fairness) indicates a saving of 105 W which corresponds to 17 %. This energy saving comes at the cost of delaying the users before entering the network. The resulted average waiting time for the EAAC in this experiment is 22 s. This number is the average over the users who were put in the waiting queue as well as the users who found all nodes of their required path ON and were immediately admitted in the network. The users that are put in the waiting queue will have to wait for at least T_{on} until the required nodes are turned on. The voluntary waiting times of the flows were respected in all cases.

4.2 Impact of the Voluntary Waiting Time

In the experiments of Sect. 4 the time that each user is willing to wait until admitted to the network was chosen to be $W = T_{on} + [0 - 20]$. Here we investigate the impact that this time has to the energy saving. We therefore compare the case where the users are only willing to wait for $W = T_{on}$, to larger waiting times. In the first case the additional voluntary waiting times are distributed between 0–20 ($W = T_1$) seconds and in the second case distributed between 20–40 ($W = T_2$). In Fig. 4 it can be observed that the willingness of users to wait more can further increase the energy savings in the network, since for $W = T_2$ we observe higher savings (20 %) compared to $W = T_1$ (17 %) and $W = T + on$ (13 %).

In Fig. 5 we show the total number of times that nodes were turned off during the experiments. It can be observed that the total measured power consumption (Fig. 4) is not directly related with the total number of turned off nodes and it should not be considered as a measure of efficiency. Thus, a static solution that would try to turn off as many nodes as possible would not necessary be the most energy efficient.

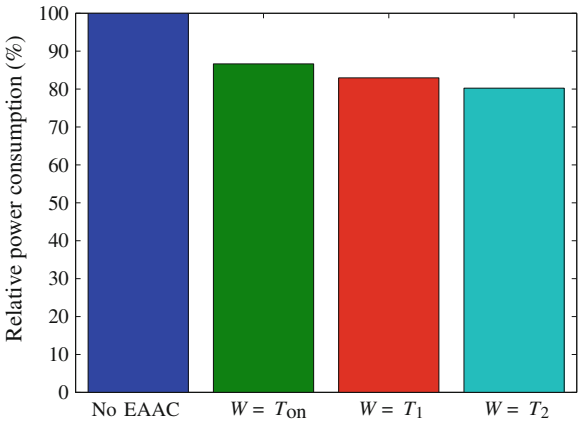

Fig. 4 Power consumption results for different values of the voluntary waiting time W. Larger values of W result in greater energy savings

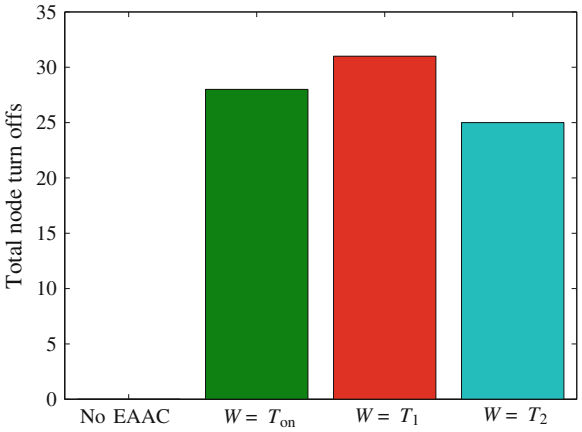

Fig. 5 Total number of times that nodes have been turned off during the experiment

5 Discussion of the Results

In this paper we present and evaluate a novel Energy Aware Admission Control mechanism which can manage an 'energy aware' network in which nodes can turn off if idle. The EAAC monitors the network nodes and requests in order to reduce energy consumption and ensure path availability for any new request.

We evaluate the performance of the EAAC mechanism in a realistic scenario where the PC-based routers are controlled by a mechanism that detects inactivity and turns them off and then turns them on again when needed. The contribution of this paper is twofold. Firstly, we point out the implementation obstacles of turning off and on devices in a network and examine the resulting energy expenditure and delay cost. Secondly, we propose and evaluate the Energy Efficient Admission Control mechanism as a solution to energy efficiency and network management.

We show that significant savings can be achieved and identify the importance of sleep states that can be entered and exited quickly. In addition, we use the notion of 'voluntary waiting time' considering that some users may be willing to wait before using the network. We show that larger voluntary waiting times can further improve the energy efficiency.

6 Future Work

The experiments and results described here show the effectiveness of the Energy Aware Admission Control mechanism and reveal room for potential energy savings. These energy savings largely depend on the capability of the nodes to enter and exit a low-power sleep state quickly as turning on the nodes on demand will induce a

waiting delay for users before being admitted into the network. Though, it is expected that future hardware and operational systems will feature these capabilities.

Our future work will concentrate on examining the trade off between turning off machines and network QoS metrics, such as latency, and seek to identify optimal operating points in terms of energy efficiency and QoS.

References

1. Berl A, Gelenbe E, Giuliani G, De Meer H, Dang M, Pentikousis K et al (2010) Energy-efficient cloud computing. Comput J 53(7):1045–1051. doi:10.1093/comjnl/bxp080
2. Bianzino A, Chaudet C, Rossi D, Rougier J et al (2010) A survey of green networking research. Commun Surv Tutorials IEEE 99:1–18. doi:10.1109/SURV.2011.113010.00106
3. Chabarek J, Sommers J, Barford P, Estan C, Tsiang D, Wright S (2008) Power awareness in network design and routing. In: INFOCOM 2008, IEEE. pp 457–465. Doi:10.1109/INFOCOM.2008.93.
4. Chiaraviglio L, Mellia M, Neri F (2009) Reducing power consumption in backbone networks. In: IEEE International Conference on, Communications, ICC'09, pp 1–6. Doi:10.1109/ICC.2009.5199404.
5. Gartner I (2007) Gartner estimates ICT industry accounts for 2 percent of global CO2 emissions. www.gartner.com/it/page.jsp?id=503867.
6. Gelenbe E, Lent R (2004) Power-aware ad hoc cognitive packet networks. Ad Hoc Netw 2(3):205–216. doi:10.1016/j.adhoc.2004.03.009
7. Gelenbe E, Mahmoodi T (2011) Energy-aware routing in the cognitive packet network. In: International Conference on Smart Grids, Green Communications, and IT Energy-aware Technologies (Energy 2011)
8. Gelenbe E, Morfopoulou C (2011) A framewok for energy aware routing in packet networks. Comput J 54(6)
9. Gelenbe E, Sakellari G, D' Arienzo M et al. (2008) Admission of QoS aware users in a smart network. ACM Trans Autono Adaptive Syst 3(1):4:1–4:28.
10. Gupta M, Singh S (2003) Greening of the Internet. Comput Commun Rev 33(4):19–26
11. Mahadevan P, Sharma P, Banerjee S (2009) A power benchmarking framework for network devices. In Proceedings of IFIP Networking.
12. Nedevschi S, Popa L, Iannaccone G, Ratnasamy S, Wetherall D (2008) Reducing network energy consumption via sleeping and rate-adaptation. In: NSDI'08: Proceedings of 5th Symposium on Networked Systems Design and Implementation, USENIX Association, Berkeley, pp 323–336.
13. Perros HG, Elsayed KM (1996) Call admission control schemes: a review. IEEE Commun Mag 34(11):82–91
14. Sakellari G, Morfopoulou C, Mahmoodi T, Gelenbe E (2012) Using energy criteria to admit ows in a wired network. In: 27th International Symposium on Computer and, Information Sciences (ISCIS).

Part III
Computational Linguistics

Named Entity Recognition in Turkish with Bayesian Learning and Hybrid Approaches

Sermet RehaYavuz, Dilek Küçük and Adnan Yazıcı

Abstract Named entity recognition is one of the significant textual information extraction tasks. In this paper, we present two approaches for named entity recognition on Turkish texts. The first is a Bayesian learning approach which is trained on a considerably limited training set. The second approach comprises two hybrid systems based on joint utilization of this Bayesian learning approach and a previously proposed rule-based named entity recognizer. All of the proposed three approaches achieve promising performance rates. This paper is significant as it reports the first use of the Bayesian approach for the task of named entity recognition on Turkish texts for which especially practical approaches are still insufficient.

1 Introduction

Named entity recognition (NER) is a well-established information extraction (IE) task and is defined as the extraction of the names of people, organizations, and locations, possibly with some temporal and monetary expressions [1]. Approaches to the NER task range from rule-based systems to learning-based and statistical systems which make use of annotated corpora and are therefore freed from human intervention as reviewed in [2]. Bayesian learning [3], hidden Markov models (HMMs) [4, 5], support vector machines (SVM) [6], and conditional random fields (CRF) [7] are

S. Reha.Yavuz · A. Yazıcı (✉)
Department of Computer Engineering, Middle East Technical University,
06800 Ankara, Turkey
e-mail: yazici@ceng.metu.edu.tr

D. Küçük
Electrical Power Technologies Group , TÜBİTAK Energy Institute,
06800 Ankara, Turkey
e-mail: dilek.kucuk@tubitak.gov.tr

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_13,

among the widely employed machine learning/statistical techniques for several IE tasks including the NER task.

Considerable work on NER for well-studied languages such as English has been reported in the literature, yet, NER research on other languages including Turkish is quite rare. Among the related literature on Turkish texts, in [5], an HMM based statistical name extractor is described, in [8] the first rule-based NER system for Turkish is presented, the latter system is turned into a hybrid recognizer which is shown to outperform its rule-based predecessor [9]. A system utilizing CRF and a set of morphological features is presented in [10] and finally, a rule learning system for the NER task in Turkish texts is presented in [11].

In this paper, we target at the NER problem on Turkish texts and propose two approaches to address the problem: the former approach is based on Bayesian learning and the latter one is a hybrid approach combining the capabilities of the former approach with that of the rule-based named entity recognizer [8]. The evaluation results demonstrate that both of the presented approaches achieve promising performance rates on the evaluation corpora and the approaches are also compared with related literature.

The rest of the paper is organized as follows: in Sect. 2, the NER system employing the Bayesian learning approach is described and Sect. 3 presents the hybrid approach comprising two distinct hybrid systems with different characteristics. Evaluation results of the proposed approaches and their comparison with related work are provided in Sect. 4 and finally Sect. 5 concludes the paper.

2 The Bayesian Learning Approach for Named Entity Recognition in Turkish

The Bayesian learning approach proposed in this paper is a modified version of the Bayesian approach presented in [3]. It also utilizes the probabilities of tokens conforming to a set of features to be named entities, along with the probabilities of tokens used in the original Bayesian approach. In the following subsections, we first describe the original *BayesIDF* method proposed in [3] for information extraction and then present our approach based on this method.

2.1 BayesIDF Method

The Bayesian method, called *BayesIDF*, as described in [3] is based on the well-known Bayes' rule provided below. In classification problems, the denominator is usually ignored since it will be the same for all hypotheses and therefore, the rule simply states that the posterior probability of a hypothesis H is proportional to the product of the probability of observing the data conditioned on H, $Pr(D|H)$, and

the prior probability of H, $Pr(H)$.

$$Pr(H|D) = \frac{Pr(D|H)Pr(H)}{Pr(D)}$$

Within the context of information extraction, a hypothesis of the form $H_{p,k}$ corresponds to "k tokens beginning at position p to constitute a field instance" and out of all possible hypotheses the most probable one (with the highest $Pr(D|H_{p,k})Pr(H_{p,k})$) will be chosen [3]. In this Bayesian approach, $Pr(H_{p,k})$ is calculated as follows [3]:

$$Pr(H_{p,k}) = Pr(\text{position} = \text{p})\text{Pr}(\text{length} = \text{k})$$

In order to estimate the position, the instances in the training date are sorted based on their position, then grouped into bins of a certain size and frequencies for these bins are calculated, the position estimate for a test instance is found after interpolation between the midpoints of the closest bins. As the length estimate, the ratio of the number of instances of length k over all instances is used [3]. The $Pr(D|H_{p,k})$, the second probability necessary to calculate $Pr(D|H_{p,k})$ is found as follows where w is the number of tokens to be considered before and after an instance:

$$\left[\prod_{j=1}^{w} Pr(\text{before}_j = t_{p-j})\right]\left[\prod_{j=1}^{k} Pr(in_j = t_{p+j-1})\right]\left[\prod_{j=1}^{w} Pr(\text{after}_j = t_{p+k+j-1})\right]$$

In the training data set, for each token before/inside/after a field instance, the above probabilities are calculated as the ratio of the number the occurrences before/inside/after a field instance over all occurrences of a token [3].

2.2 Proposed Bayesian Approach

Our Bayesian approach for NER on Turkish texts uses the following modified formula to calculate $Pr(D|H_{p,k})$:

$$\frac{\sum_{j=1}^{BSD} Pr(before_j = t_{p-j}) + \frac{\sum_{j=1}^{k} Pr(in_j = t_{p+j-1})}{k_{avg}} + \sum_{j=1}^{ASD} Pr(after_j = t_{p+k+j-1}) + \frac{Pr(FC)}{k_{avg}}}{BSD + ASD + FC + 1}$$

In the above formula, $Pr(FC)$ is calculated as follows:

$$Pr(FC) = \sum_{f=1}^{FC}\sum_{j=1}^{k} \sigma(t_{p+j-1}, f)\phi(f)$$

where $\sigma(t_{p+j-1}, f)$ is 1 if t_{p+j-1} conforms to feature$_f$ and 0 otherwise and $\phi(f)$ is the probability that a token conforming to feature$_f$ is a named entity.

In the formula for $Pr(D|H_{p,k})$, *BSD* stands for *Before Surroundings Distance* and *ASD* stands for *After Surroundings Distance* which correspond to the number of tokens to be considered before and after named entity instances in the training set, respectively. As a matter of fact, we have used these parameters instead of the context parameter, w, in the original *BayesIDF* method [3] summarized in the previous subsection so that these parameters can independently be set to different values. We have also included in the formula the probabilities calculated for each of the enabled features as $Pr(FC)$, where the number of features is denoted as FC (for *feature count*). Features can also be assigned weights so that the corresponding probabilities are multiplied with certain coefficients corresponding to these weights. Instead of multiplying the probabilities (as the contribution of the occurrence of each of the tokens to the overall probability is assumed to be independent [3]), we have added them together and normalized the resulting summations by dividing them to $BSD + ASD + FC + 1$. The reason for using summations instead of products is that due to the scarcity of the available annotated corpora used for training, using products (as in [3]) has resulted in very low probabilities which in turn has lead to low success rates. We should also note that the probability summation regarding the inside tokens and $Pr(FC)$ are divided by the average number of tokens in a named entity, k_{avg}, as calculated from the training data set.

Below, we first describe the details of the parameters used during the training phase including *BSD*, *ASD*, and *case sensitivity*. Next, we describe the features (or feature sets when applicable) employed with pointers to related studies. It should again be noted that assigning different coefficients to distinct features we can alleviate or boost the effects of these features.

The parameters utilized by the modified Bayesian method proposed:

- *Before Surroundings Distance* (BSD): In any Bayesian technique, a number of tokens before and after a field instance are used for training and estimation which are called *surroundings*. *BSD* is the number of tokens before a named entity where probabilities for these tokens will be calculated and utilized during the calculation of the probability for a candidate named entity. To illustrate, when *BSD* is three, corresponding probabilities of three tokens before a named entity will be calculated during training phase and will be utilized during the estimation phase.
- *After Surroundings Distance* (ASD): As its name implies, *ASD* parameter is similar to *BSD* and it specifies the number of following tokens that will contribute to the probability of a candidate named entity to be classified so.
- *Case Sensitivity*: This parameter specifies whether each token should be considered in a case-sensitive or in a case-insensitive manner. For instance, if case-sensitivity is turned off, the tokens *bugün* and *Bugün* (meaning '*today*') will be considered as the same token during the calculation of the probabilities.

The features utilized by the proposed method:

- *Case Feature*: This feature is used to map each token to one of the four classes: *all-lower-case*, *all-upper-case*, *first-letter-upper-case*, and *inapplicable*. The last class is for representing those tokens comprising punctuation marks and/or numbers. This feature is especially important for candidate named entities as case

information is known to be a plausible clue for person, location, and organization names in several languages, including Turkish.

- *Length Feature*: Similar to the previous feature, this feature maps each token to different length-related classes: *zero-length*, *one-char*, *two-chars*, *three-chars*, *four-chars*, and *longer*. Again, the probability of being in a named entity for each of these classes is calculated and utilized during the training and estimation phases.
- *Alphanumeric Feature*: This feature maps each token to one of four classes according to the nature of characters included in the tokens and again the probability of being in a named is calculated for each of these classes, to be used during the estimation phase. These classes are: *all-alpha*, *all-numeric*, *alphanumeric*, and *inapplicable*.
- *NF (Nymble Features)*: This set of features comprises a subset of the features utilized by the Nymble NER system for English [4]. Nymble is a statistical system which uses a variant of the HMM and achieves successful results over the related literature [4]. This feature set, obtained from the feature set of the Nymble system, encompasses some of the features defined above as it includes the following: *two-digit-number*, *four-digit-number*, *alphanumeric*, *other-number*, *all-capital*, *first-capital*, *lower-case*, *other*.
- *Lexical Resource Feature*: The feature of appearance in a lexical resource (name lists, gazetteers, etc.) is also considered as a distinct feature, i.e., *lexical resource feature*. As the required resources; person, location, and organization name lists of the rule-based recognizer for Turkish [8] are utilized.

3 The Hybrid Approaches

We have also proposed two different hybrid named entity recognition systems which utilize the Bayesian learning based recognizer described in the previous section and the rule-based named entity recognizer [8]. These two hybrid systems are briefly described below:

1. *Training-Phase Hybrid System*: This hybrid system is first trained on all the available training data, as the Bayesian learning based recognizer. But before carrying out the estimations on the test data, the rule-based recognizer [8] is run on the test data and the resulting annotated test data is also utilized as additional training data to update the probabilities to be employed by the hybrid system.
2. *Estimation-Phase Hybrid System*: This hybrid system version utilizes the output of the rule-based recognizer [8] during the estimation phase as a distinct feature. The output of the rule-based recognizer is parsed and the tokens corresponding to the annotated named entities are assigned additional scores to be used along with the probabilities calculated during the training phase. When this hybrid system makes estimations on the test data, if all of the elements of a considered token group are annotated as named entities by the rule-based system, this token group

gets an additional score of 1.0 while a token group partially annotated by the rule-based system gets an additional score of 0.5.

4 Evaluation and Discussion

During the testing of the proposed system in Freitag's work [3], a threshold value is used so that tokens with posterior probabilities above this threshold value are annotated with the corresponding named entity types and the remaining tokens are not annotated. In our study, we have tried several different threshold values during testing and the best results obtained are reported in this section.

The performance evaluation of the proposed approaches has been carried out with the widely used metrics of precision, recall, and F-Measure. These metrics, as utilized in studies such as [9, 12, 13], also give credit to partial named entity extractions where the type of the named entity is correct but its span is not correct. The exact formulae for the metrics are presented below:

$$\text{Precision} = \frac{\text{Correct} + 0.5 * \text{Partial}}{\text{Correct} + \text{Spurious} + 0.5 * \text{Partial}}$$

$$\text{Recall} = \frac{\text{Correct} + 0.5 * \text{Partial}}{\text{Correct} + \text{Missing} + 0.5 * \text{Partial}}$$

$$\text{F}-\text{Measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

In these formulae, *Correct* is the number of correctly-estimated named entities in terms of their type and span (i.e., their location and the number of tokens included). *Spurious* is the number of incorrectly-estimated named entities, that is, those estimated ones which are not in the answer key. *Missing* is the number of named entities which are missed by the recognizer though they exist in the answer key. Lastly, *Partial* denotes the number of partially-estimated named entities which are of the correct type but their spans are not correct [9, 12].

In order to train the Bayesian learning based recognizer and test its performance, we have used one of the data sets which was previously compiled and annotated with named entities [9]. This news text data set comprises 50 news articles from METU Turkish corpus [14] with a total word count of 101,700 where each article contains about 2,000 words. The annotated entities in this set encompass 3,280 person, 2,470 location, 3,124 organization names along with 1,413 date/time and 919 money/percent expressions, hence amounting to 11,206 named entities.

We have evaluated the performance of the proposed Bayesian learning based recognizer on this data set with ten-fold cross validation. During this evaluation, *BSD* and *ASD* are taken as 1 and among the feature set, only NF features are enabled. The ten-fold cross validation results of the recognizer are presented in Table 1.

Table 1 Ten-fold cross validation results of the Bayesian learning based named entity recognizer on the data set

Named Entity Type	Precision (%)	Recall (%)	F-Measure (%)
Person	94.41	86.54	90.30
Location	92.10	82.54	87.06
Organization	88.78	87.29	88.03
Date	90.07	77.19	83.13
Time	86.13	75.17	80.28
Money	76.65	94.71	84.73
Percent	88.27	96.09	92.01
Overall	*90.74*	*85.40*	*87.99*

Table 2 Ten-fold cross validation results of the Bayesian learning based named entity recognizer on the data set (for different recognizer configurations)

Configuration	Precision (%)	Recall (%)	F-Measure (%)	Effect
Baseline	74.67	74.32	74.49	
Baseline+Alphanumeric feature	74.42	74.20	74.31	Negative
Baseline+Case feature	75.08	74.53	74.80	Positive
Baseline+Case sensitivity	75.31	75.85	75.58	Positive
Baseline+Length feature	74.31	73.89	74.10	Negative
Baseline+Lexical resource feature	70.75	75.77	73.17	Negative
Baseline+NF	74.95	74.32	74.63	Positive

In order to test the individual contribution of each the parameters/features, we have carried out evaluations, first on a baseline configuration of the recognizer, and then turning each of the parameters/features on, while turning the remaining ones off. During the evaluations, the coefficients are all set to 1.0 and in the baseline configuration, *BSD* and *ASD* are 0 and all of the features are turned off. The corresponding evaluation results are provided in Table 2 where the first row corresponds to the performance results of the recognizer in the baseline configuration. The last column denotes the effect of turning the corresponding parameter/feature on, to the overall F-Measure.

Experiments with different *BSD* and *ASD* as integers within the [1–5] scale have shown that the best F-Measure rates are obtained when *BSD* and *ASD* are both 1. Increasing these values have positive effects for some named entity types while having negative effects on others, and the overall F-Measure rates corresponding to higher *BSD* and *ASD* values are less than those rates when *BSD* and *ASD* are both set to 1.

We have used another news article from METU Turkish corpus [14] as the test set to evaluate and compare our Bayesian learning based recognizer as well as the two hybrid recognizers described in Sect. 3 built on the top of the former system, with the previously proposed rule-based system [8] and its hybrid counterpart [9]. The systems that require training have been trained on the aforementioned news text

Table 3 Evaluation results of the proposed named entity recognizers and related work.

Named entity recognizer	Precision (%)	Recall (%)	F-Measure (%)
Rule-based recognizer [8]	94.71	81.36	87.53
Hybrid (Rule-based+Rote learning recognizer [9])	94.27	81.9	87.65
Bayesian learning based recognizer	96.16	81.82	88.41
Training phase hybrid recognizer	92.68	87.56	90.05
Estimation phase hybrid recognizer	93.38	89.57	91.44

data set (with a word count of 101,700). The news article used as the new test data set has 2057 words and after its annotation to create the answer key, it comprises 228 annotated named entities where there are 102 person, 42 location, and 71 organization names along with 11 date expressions, 1 monetary and 1 percent expressions with no instance of time expressions. The evaluation results of the newly proposed two recognizers and those recognizers previously proposed for Turkish texts are presented in Table 3. The performance evaluations of the named entity recognizers proposed in the current paper are given in the last three rows.

The results in Table 3 show that the proposed Bayesian learning based recognizer achieves better results than the rule-based and hybrid (rule-based+rote learning) systems previously proposed [9]. The hybrid systems proposed, in turn, achieve better performance rates than their predecessor, the Bayesian learning based recognizer. Among the proposed hybrid systems, the latter estimation-phase hybrid system achieves higher success rates than the training-phase hybrid system. However, as pointed out at the beginning of this section, it should be noted that the proposed three approaches do not used predefined threshold values, instead, they try several alternative values during testing and the highest results achieved by the systems are given in Table 3. Therefore, in order to make a more appropriate comparison with the related work, a predetermined threshold value should be obtained (either heuristically or through a learning procedure) and utilized during the testing of the proposed systems, as future work.

To summarize, all of the proposed systems achieve promising results on the test data set which is a significant contribution to NER research on Turkish texts, as related research is quite insufficient compared to studies on languages such as English and, to the best of our knowledge, the proposed systems are the first to apply a Bayesian approach to this task on Turkish texts with a limited training data set. Yet, we expect that the results should be verified on larger test corpora and can be improved by increasing the annotated training data set, both of which are plausible future research directions. Other important future research topics include deeper elaboration of the employed parameters and features on larger corpora to better evaluate their effects.

5 Conclusion

Named entity recognition, as the other information extraction tasks, gains more significance every day, mostly due to the increase in the size of natural language texts, such as those on the Web, that need to be processed. In this paper, we target at named entity recognition in Turkish texts and propose two approaches for this problem: the first one is a Bayesian learning based approach and the second approach comprises two hybrid recognizers with different characteristics where the Bayesian learning system is basically utilized together with a previously proposed rule-based recognizer to achieve better performance rates. The evaluation results have shown that the proposed three approaches achieve promising results and the two hybrid approaches perform better than the Bayesian learning based recognizer. Yet, in order to further verify and increase the success rates of the proposed approaches, larger annotated corpora are necessary and the lack of such corpora is known to be one of the main problems against information extraction research on Turkish texts.

References

1. Grishman R (2003) Information extraction. In: Mitkov R (ed) The Oxford Handbook of Computational Linguistics. Oxford University Press, Oxford
2. Turmo J, Ageno A, Catala N (2006) Adaptive information extraction. ACM Comput Surv 38(2):1–47
3. Freitag D (2000) Machine learning for information extraction in informal domains. Mach Learn 39(2–3):169–202
4. Bikel DM, Miller S, Schwartz R, Weischedel R (1997) Nymble: a high-performance learning name-finder. Proceedings of the Fifth Conference on Applied Natural Language Processing, In, pp 194–201
5. Tür G, Hakkani-Tür D, Oflazer K (2003) A statistical information extraction system for Turkish. Nat Lang Eng 9(2):181–210
6. Li Y, Bontcheva K, Cunningham H (2009) Adapting SVM for Data Sparseness and Imbalance: A Case Study on Information Extraction. Nat Lang Eng 15(2):241–271
7. McCallum A, Li W (2003) Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. Proceedings of the Seventh Conference on Natural, Language Learning, In, pp 188–191
8. Küçük D, Yazıcı A (2009) Named entity recognition experiments on Turkish texts. Proceedings of the International Conference on Flexible Query Answering Systems, In, pp 524–535
9. Küçük D, Yazıcı A (2012) A hybrid named entity recognizer for Turkish. Expert Syst Appl 39(3):2733–2742
10. Yeniterzi R (2011) Exploiting morphology in Turkish named entity recognition system. Proceedings of the ACL Student Session, In, pp 105–110
11. Tatar S, Çicekli İ (2011) Automatic rule learning exploiting morphological features for named entity recognition in Turkish. J Inf Sci 37(2):137–151

12. Küçük D, Yazıcı A (2011) Exploiting information extraction techniques for automatic semantic video indexing with an application to Turkish news videos. Knowl Based Syst 24(6):844–857
13. Maynard D, Tablan V, Ursu C, Cunningham H, Wilks Y (2001) Named entity recognition from diverse text types. In: Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP)
14. Say B, Zeyrek D, Oflazer K, Özge U (2002) Development of a corpus and a treebank for present-day written Turkish. In: Proceedings of the 11th International Conference of Turkish, Linguistics (ICTL)

Transfer Learning Using Twitter Data for Improving Sentiment Classification of Turkish Political News

Mesut Kaya, Guven Fidan and I Hakkı Toroslu

Abstract In this paper, we aim to determine the overall sentiment classification of Turkish political columns. That is, our goal is to determine whether the whole document has positive or negative opinion regardless of its subject. In order to enhance the performance of the classification, transfer learning is applied from unlabeled Twitter data to labeled political columns. A variation of self-taught learning has been proposed, and implemented for the classification. Different machine learning techniques, including support vector machine, maximum entropy classification, and Naive-Bayes has been used for the supervised learning phase. In our experiments we have obtained up to 26 % increase in the accuracy of the classification with the inclusion of the Twitter data into the sentiment classification of Turkish political columns using transfer learning.

1 Introduction

Social Media has become a global forum for people to express their subjective thoughts, opinions, and feelings. People express their opinions about almost anything like products, social events, news etc. People are curious about other peoples opinions. In the news domain, in general, people are still more interested in the opinions of a special group of experts, namely newspaper columnists rather than ordinary peoples comments on social media. With the rapid growth of Twitter among people,

M. Kaya · I. H. Toroslu (✉)
Department of Computer Engineering, Middle East Technical University, Ankara, Turkey
e-mail: e1502509@ceng.metu.edu.trlmesut.kaya@agmlab.com

Guven Fidan
R&D Department AGMLab, Ankara, Turkey
e-mail: guven.fidan@agmlab.com

I. H. Toroslu
e-mail: toroslu@ceng.metu.edu.tr

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_14,

almost all of the columnists and journalists have Twitter accounts and share their personal opinions informally on Twitter. Therefore, it is possible to analyze columnists' opinions and feelings both from Twitter data and from their newspaper columns.

In our previous work [16], through several experiments we have shown that it is possible to obtain better sentiment classification results for Turkish political columns by providing a list of effective words and increasing the weight of these features within the model created by using the training data. Besides, one of the difficulties of the sentiment classification of news data is the lack of tagged data. Since the columns are not short texts, the annotation task is difficult and expensive. In order to have a good performance from sentiment classification, large amount of annotated data is needed.

In order to provide a wide effective words list and overcome the lack of tagged data problem, in this work, we adapt the *shape transfer learning* approach, aiming to extract the knowledge from source tasks to be applied to a target task [1]. In this paper, features are transferred from Twitter domain to news domain in an unsupervised way. The idea is to extract important features (such as, unigrams) from columnists' Twitter accounts and use them in the training phase of the sentiment classification of political columns. By using unlabeled data from Twitter domain, the need and the effort to collect more training data can be reduced and the performance of classifiers can be increased.

The content of this paper is as follows: In Sect. 2 related works and the literature are reviwed. In Sect. 3, transfer learning methodology is explained with the details of the algorithms used and background information is given. In Sect. 4, experimental setup and the evaluation metrics used are covered, and detailed analyses of the evaluations are given. Finally, In Sect. 5, the work is concluded.

2 Related Work

In this section, we briefly summarize the related work on sentiment classification and its applications on the news domain.

In their book, Opinion Mining and Sentiment Analysis Pang and Lee provide a detailed survey of sentiment analysis from natural language processing (NLP) and Machine Learning (ML) perspectives [2], and they also describe several application domains. News is one of them.

Viondhini and Chandrasekaran [17] states that in the text categorization Machine Learning techniques like naive bayes (NB), support vector machine (SVM) and maximum entropy(ME) have achieved great success. They also state that other used ML techniques in the NLP area are: K-nearest neighborhood (KNN), N-gram model.

There are some works on the application of the sentiment analysis to the news domain with different approaches [10–15]. One of the recent works in this domain is our recent work on the sentiment analysis of Turkish political columns [16]. As an initial work on Turkish political domain, sentiment classification techniques are incorporated into the domain of political news from columns in different Turkish

news sites. The performances of different machine learning methods are measured and the problem of sentiment classification in news domain is discussed in detail.

There are lots of sentiment analysis techniques and different areas that these techniques are applied. Transfer learning and domain adaptation techniques are used widely in ML [3]. Transfer learning has been applied in many different research areas containing NLP problems, learning data across domains, image classification problems etc. [1].

One of the problems that transfer learning and domain adaptation are applied is sentiment classification. Ave and Gommon conducted an initial study to customize sentiment classifiers to new domains [4]. Bliter et al. extend structural correspondence learning (SCL) to sentiment classification to investigate domain adaptation for sentiment classifiers by focusing on online reviews for different types of products [5]. Li et al. outline a novel sentiment transfer mechanism based on constructed non-negative matrix tri-factorizations of term document matrices in the source and target domains [3].

In our work, different than the other domain adaptation and transfer learning methods applied in sentiment classification tasks, we use unlabeled data with unsupervised feature construction, and transferring knowledge from short text (Tweets) to long text (columns), which is not a common technique applied in transfer learning. Besides, our work is an initial work for applying transfer learning for sentiment classification of Turkish texts.

3 Transfer Learning Methodology

3.1 Background

3.1.1 Transfer Learning

Transfer Learning's main goal is to extract useful knowledge from one or more source tasks and to transfer the extracted information into a target task where the roles of source and target tasks are not necessarily the same [1].

In our work, we aim to solve sentiment classification of Turkish political columns (target task) by extracting and transferring features from unlabeled Twitter data in an unsupervised way (source task). Source domain is Twitter and contains unlabeled data; target domain is news and contains labeled data. Notice that source and target data does not share the class labels. Besides, the generative distribution of the labeled data is not the same as unlabeled data's distribution.

Our main motivation is the assumption that even unlabeled Twitter data collected from columnist's verified accounts may help us to learn important features in the politic news domain. By using this assumption, we use transfer learning. This kind of transfer learning is categorized as self-taught learning which is similar to inductive transfer learning. Self-taught learning was first proposed by Raina et al [6].

3.1.2 F-Score for Feature Ranking

In order to measure the importance of a feature for a classifier, we use F-Score (Fisher score) [8, 9]. F-Score has been chosen, since it is independent of the classifiers, so that we can use it for 3 different classifiers we use in experiments.

Given the training instances $x_i, i = 1, 2, 3, \ldots, l$ the F-score of the jth feature is defined as:

$$F(j) = \frac{(\bar{x}_j^{(+)} - \bar{x}_j)^2 + (\bar{x}_j^{(-)} - \bar{x}_j)^2}{\frac{1}{n_+ - 1}\sum_{i=1}^{n_+}(x_{i,j}^{(+)} - \bar{x}_j^{(+)})^2 + \frac{1}{n_- - 1}\sum_{i=1}^{n_-}(x_{i,j}^{(-)} - \bar{x}_j^{(-)})^2} \quad (1)$$

where n_+ and n_- are the number of positive and negative instances in the data set respectively; $\bar{x}_j$, $\bar{x}_j^+$ and $\bar{x}_j^-$ represents the averages of the jth feature of the whole positive-labeled and negative-labeled instances; $\bar{x}_{i,j}^+$ and $\bar{x}_{i,j}^-$ represent the jth feature of ith positive and negative instance. Larger F-Score means that the feature has more importance for the classifier.

3.1.3 TF-IDF Weighting

Term frequency-inverse document frequency measures how important a feature (word) to a document and it is used as weighting factor in text mining applications. Variations of tf-idf calculations are available, and in this work we use the following formulations:

Given a corpus D, a document d and a term t in that document term frequency, inverse term frequency and tf-idf are calculated by multiplying tf and idf, where tf is the number of times t occurs in d over total number of terms in d and idf is the logarithm of number of docs in D divided by number of documents in D that t occurs in.

3.2 Data Sets

In our experiments we use three different data sets, one from the news domain and the other two from Twitter domain. Articles from the news domain are collected via specific crawlers and annotated, we have 400 annotated columns. Tweets from columnists' Twitter accounts are collected by using Twitter4J API [1]. Search API used to collect all accessible tweets of the columnists. 123,074 tweets of columnists are collected. In order to collect random tweets, more than 100,000, Streaming API is used. The formulation of labeled news data that will be used in the rest of the paper is as follows:

[1] http://twitter4j.org/en/index.html

$$T = \left\{ (x_l^{(1)}, y^{(1)}), (x_l^{(2)}, y^{(2)}) \ldots, (x_l^{(m)}, y^{(m)}) \right\}$$

A news data is represented as $(x_l^{(j)}, y^{(j)})$ where $x_l^{(j)} = (f_1, f_2 \ldots, f_k)$ is a term vector of the text and each f_k is tf-idf value for features of the sample data and $y^{(j)} \in \{pos, neg\}$.

The formulation of unlabeled Twitter data collected from columnist's Twitter account U_1 and from random Twitter accounts U_2 that will be used in the rest of the paper are: $U_1 = \left\{ z_{u1}^1, z_{u1}^2 \ldots, z_{u1}^a \right\}$ and $U_2 = \left\{ z_{u2}^1, z_{u2}^2 \ldots, z_{u2}^b \right\}$. In U_1 and U_2, each z_{u1}^a corresponds to an unlabeled tweet of columnists' verified Twitter accounts and each z_{u2}^b corresponds to an unlabeled tweet of random Twitter accounts and they contain number of occurrences of each feature within tweet.

In both of the algorithms explained in detail below, by using less frequent and most frequent features in U_2 noisy features are eliminated from U_1. Then, by using filtered U_2, a list L_u of sorted features according to their occurrences in the all documents is generated. Then, the number of occurrences of the features are normalized using the log_x function (best x is chosen after several experiments). Actually the normalized list L_u contains feature and value pairs. Notice that, after normalization some features in L_u are eliminated. For the second algorithm described below, by using the labeled training set T, we calculate F-score of each feature and a list L_v of sorted features according to their F-scores is generated.

3.3 Algoritms

We propose two different approaches in the unsupervised construction of the transferred features:

3.3.1 Algorithm-1

Unsupervised feature construction without using the knowledge of the feature rankings within the classifier used. The algorithm used is given below:

Algorithm 1: Unsupervised feature construction without feature rankings

Input: T, U_1 and U_2
Output: Learned Classifier C for Classification Task
1 Construct L_u by using U_1 and U_2
2 Construct new labeled set $\bar{T} = \left\{ (\bar{x}_l^{(i)}, y^{(i)}) \right\}_{i=1}^{m}$ by L_u and T
3 Learn a classifier C by applying supervised learning algorithm (SVM, Naive Bayes or Maximum Entropy).
4 **return** C

Without transfer learning tf-idf values for T are as follows:

$$\forall i \forall j \left(x_l^{(j)}(f_j) \right) = \frac{count\left((x_l^{(j)}(f_j)) \right)}{j} \times idf \tag{2}$$

After applying steps 1 and 2, with transfer learning (we simply increase the term frequency of transferred features) the tf-idf in $\bar{T}$ are as follows:

$$\forall i \forall j \left(x_l^{(j)}(f_j) \right) = \frac{count\left((x_l^{(j)}(f_j)) \right) + log_x \left(valueoff_j inL_u \right)}{j} \times idf \tag{3}$$

while constructing $\bar{T}$ only the transferred features of L_u, are included into $\bar{T}$. Namely, features in the target domain that do not appear in L_u are eliminated.

3.3.2 Algorithm-2

Unsupervised feature construction with using the knowledge of feature rankings within the classifier used. The algorithm is given below:

Algorithm 2: Unsupervised feature construction with feature rankings

Input: T, U_1 and U_2
Output: Learned Classifier C for Classification Task
Construct L_u by using U_1 and U_2
Use T to calculate f-score of each feature and L_v.
Combine L_u and L_v and have a list L_{u+v}.
Transfer knowledge in L_{u+v} to obtain $\bar{T} = \left\{ (\bar{x}_l^{(i)}, y^{(i)}) \right\}_{i=1}^{m}$
Learn a classifier C by applying supervised learning algorithm (SVM, Naive Bayes or Maximum Entropy).
return C

By combining L_u and L_v a third list L_{u+v} is constructed as follows: $L_{u+v} = c_1 L_u + c_2 L_v$. In order to decide on optimal c_1 and c_2 values several experiments are conducted. Different than Algorithm-1, after transferring information the tf-idf in $\bar{T}$ become as follows:

$$\forall i \forall j \left(x_l^{(j)}(f_j) \right) = \frac{count\left((x_l^{(j)}(f_j)) \right) + log_x \left(valueoff_j inL_{u+v} \right)}{j} \times idf \tag{4}$$

In Algorithm-2 $\bar{T}$ is constructed by using only the transferred features from L_{u+v}.

Table 1 Baseline Results

	NB	ME	SVM
Unigram	71.81	75.85	71.12
Unigram+adjective	71.81	75.59	72.95
Unigram+effective words	71.81	76.31	73.70

4 Evaluation

4.1 Experimental Setup

In the experiments, K-fold-cross-validation [7] is conducted by adopting K to be 3. 200 positive and 200 negative news items are used to make a 3-fold-cross-validation in the data experiments. Two experiments are adopted by using 3 different machine learning methods: namely NB, ME and SVM.

In order to have a baseline, sentiment classification of news columns are generated by using unigrams as features without transferring any knowledge from Twitter domain(for this purpose, we use the values from our previous work [16]). Table 1 shows the baseline results.

Transfer learning, adopted with and without the feature ranking information are applied to sentiment classification, and the results are compared with the results of the baseline. To evaluate the performance of the different experiments the following typical accuracy metric, which is commonly used in text classification, is used:

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

4.2 Results

In the first set of experiments, unsupervised feature construction for transfer learning is applied without using the feature ranking knowledge. The amount of transferred features is from 1 to 100 %. The classifier C is learned by using only transferred features.

Figure 1 shows the performances of 3 different machine learning methods for varying amount of transferred features. Notice that these results are for the cases in which only the transferred features are included. In other words, while transferring the knowledge, for creating the classifier features, only those which are in the L_u list are used. Features in the labeled data that are not in L_u are eliminated. In this case SVM performed better than NB and ME. When compared with the baseline results, for SVM there is a 5.67 % improvement. For NB there is small change and for ME there is a 4 % decrease. Therefore, by using only the transferred features without feature ranking knowledge provides significant information only for SVM.

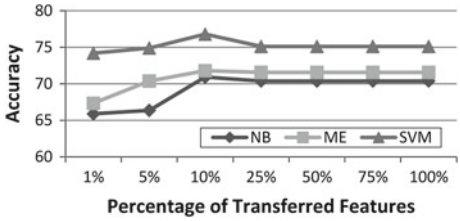

Fig. 1 Accuracy values of classifiers with only transferred features

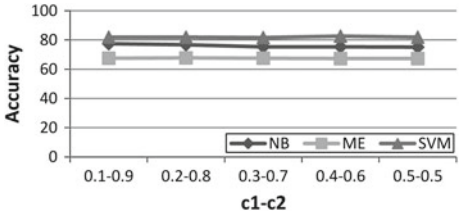

Fig. 2 Accuracy values for f-score of features included for different values of c1–c2

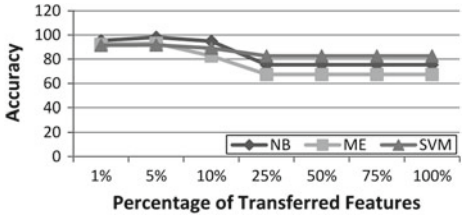

Fig. 3 Accuracy values for f-score of features included for c1 = 0.4 and c2 = 0.6

In the second set of experiments, feature rankings obtained by using F-scores are used for unsupervised feature construction. In the previous section, the details of the method used are explained. In Algorithm-2 features are combined as $L_{u+v} = c_1 L_u + c_2 L_v$ list. In Fig. 2, accuracy values for experiments are shown by using different c_1 and c_2 values. We observe that varying c_1 and c_2 values do not make significant changes. Therefore, we took $c_1 = 0.4$ and $c_2 = 0.6$ in the rest of the experiments.

Figure 3 shows the accuracy values when the amount of features transferred from constructed list $L_{(u+v)}$ varies. Combining two lists (L_u obtained from unlabeled data and L_v obtained from F-scores of the features) produces a very good performance gain. We can see from the Figure that especially transferring 5 % of constructed $L_{(u+v)}$ list provides very useful information for the classification task for all techniques that we have tried. For NB up to 98.116 % accuracy values are obtained. Comparing with the baseline results for NB, this corresponds to a 26.306 % performance gain. We observe a 19.435 % performance gain with a 93.135 % accuracy value for ME. Finally, for SVM a 15.43 % performance gain with a 91.74 % accuracy value is reached. However, if the amount of the transferred features are in 10–25 % range of list $L_{(u+v)}$, then the accuracy performance decreases for all techniques, and after that the results does not change. Roughly, we can say that features that carry important information for these classifiers are in the first 10 % of transferred features.

5 Discussion and Future Work

Although we transfer knowledge from short text to larger text and transfer features from unlabeled data in an unsupervised way, transfer learning method produced a very good improvement in the accuracy of the sentiment classification of Turkish political columns, over 90 %. In terms of relative performances, we see that in SVM, NB and ME, transferred information improves the performance. It is also observed that, transferring features that are not in the first 10 % of the transferred features decreases the performance.

We observe that the amount of transferred features do not make huge differences after a significant amount (25 %). Besides, in the second set of experiments conducted by using F-score information of the features the best results are obtained by transferring 1–10 % amount of features. This means that features carrying the most important information are the ones with higher frequency in the bag-of-words framework of transferred data.

An important outcome of this study is using feature ranking information (F score) combined with the unlabeled data turns out to be an effective method for transfer learning used in sentiment classification.

As future work, transferring from longer texts (different columns) can be analyzed, and using Transfer Learning in the Sentiment Classification of News Data with labeled data with Domain Adaptation techniques can be analyzed. Besides, using feature rankings together with unlabeled data can be adapted to different domains.

References

1. Pan SJ, Yang Q (2010) A Survey on Transfer Learning. IEEE Trans knowl data Eng 22(10):1345–1359
2. Pang B, Lee L (2008) Opinion mining and sentiment analysis. Found Trends Inf Retrieval 2(1):1–135
3. Li T, Sindhwani V, Ding C, Zhang Y (2010) Bridging domains with words: opinion analysis with matrix tri-factorizations. In: Proceedings of the Tenth SIAM Conference on Data Mining (SDM). pp 293–302
4. Aue A, Gamon M (2005) Customizing sentiment classifiers to new domains: a case study. http://research.micosoft.com/anthaue
5. Blitzer J, Dredze M, Pereira F (2007) Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification. In: Proceedings of 45th Annual Meeting of the Association, Computational Linguistics. pp. 432–439.
6. Raina R, Battle A, Lee H, Pracker B, Ng AY (2007) Self-thaught learning: transfer learning from unlabeled data. In: Proceedings of 24th International Conference on, Machine Learning. pp 759–766.
7. Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. Proc 14th Int Joint Conf. Artif Intell 2(12):1137–1143
8. Chen YW, Lin CJ (2006) Combining SVMs with various feature selection strategies. In: Guyon I, Gunn S, Nikravesh M, Zadeh L (eds) Feature extraction, foundations and applications. Springer, Berlin

9. Chang YW, Lin CJ (2008) Feature ranking using linear SVM. In: JLMR, vol 3, WCCI2008 workshop on casuality, Hong Kong.
10. Fortuna B, Galleguillos C, Cristianini N (2009) Detecting the bias in the media with statistical learning methods. Theory and applications Yatlor and Francis Publisher, Text Mining
11. Evgenia B, van der Goot E (2009) News bias of online headlines across languages. Conference Proceedings, Lodz University Publishing House, The study of conflict between Russia and Georgia. Rhetorics of the media
12. Strapparava C, Mihalcea R (2007) (2007) Semeval 2007 task 14: affective text. In: Proceedings of ACL
13. Godbole N, Srinivasaiah M, Skiena S (2007) Large-scale sentiment analysis for news and blogs. In: Proceedings of the International Conference on Weblogs and Social media (ICWSM)
14. Balahur A, Steinberger R (2009) Rethinking sentiment analysis in the news: from theory to practice and back. WOMDA'09, pp 1–12.
15. Mullen T, Malouf R (2006) A prelimianry investigation into sentiment analysis of informal political discourse. Proceedings of the AAAI symposium on computational approaches to analyzing weblogs, In, pp 159–162
16. Kaya M, Toroslu IH, Fidan G (2012) Sentiment analysis of turkish political news. The 2012 IEEE/WIC/ACM International Conference on Web Intelligence.
17. Viondhini G, Chandrasekaran RM (2012) Sentiment analysis and opinion mining: a survey. Int J Advanced Res Comput Sci Softw Eng 2(6)

A Fully Semantic Approach to Large Scale Text Categorization

Nicoletta Dessì, Stefania Dessì and Barbara Pes

Abstract Text categorization is usually performed by supervised algorithms on the large amount of hand-labelled documents which are labor-intensive and often not available. To avoid this drawback, this paper proposes a text categorization approach that is designed to fully exploiting semantic resources. It employs the ontological knowledge not only as lexical support for disambiguating terms and deriving their sense inventory, but also to classify documents in topic categories. Specifically, our work relates to apply two corpus-based thesauri (i.e. WordNet and WordNet Domains) for selecting the correct sense of words in a document while utilizing domain names for classification purposes. Experiments presented show how our approach performs well in classifying a large corpus of documents. A key part of the paper is the discussion of important aspects related to the use of surrounding words and different methods for word sense disambiguation.

1 Introduction

Text categorization is the task of automatically assigning documents to predefined categories based on their content. This task is becoming increasingly important in modern information technologies and web-based services that are faced with the problem of managing a growing body of textual information [1].

Text categorization is an active area in Information Retrieval and Machine Learning, and supervised learning algorithms are usually applied to design the target classification function [2]. Such algorithms treat categories as symbolic labels and use a set of documents, that are previously assigned to the target categories by human experts, as training data.

The drawback of these approaches is that the classifier performance depends heavily on the large amount of hand-labelled documents as they are the only source

N. Dessì (✉) · S. Dessì · B. Pes
University of Cagliari, Via Ospedale 72, 09124 Cagliari, Italy
e-mail: dessi@unica.it

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_15,

of knowledge for learning the classifier. Being a labor-intensive and time consuming activity, the manual attribution of documents to categories is extremely costly. To overcome these difficulties, semi-supervised learning techniques have been proposed that require only a small set of labelled data for each category [3].

The problem is that all of these methods require a training set of pre-classified documents and it is often the case that a suitable set of well categorized (typically by humans) training documents is not available. This creates a serious limitation for the usefulness of the above learning techniques in operational scenarios ranging from the management of web-documents to the classification of incoming news into categories, such as business, sport, politics, etc.

Most importantly, text categorization should be based on the knowledge that can be extracted from the text content rather than on a set of documents where a text could be attributed to one or another category, depending on the subjective judgment of a human classifier.

Going beyond the above mentioned approaches, recent research introduced text categorization methods based on leveraging the existing knowledge represented in a domain ontology [4]. The basic idea is to use an ontology for providing a functionality that is similar to the knowledge provided by human experts with a manual document classification.

Ontologies are used as data-models or taxonomies to add a semantic structure to the text by annotating it with unambiguous topics about the thematic content of the document. The novelty of these ontology-based approaches is that they are no dependent on the existence of a training set and rely solely on a set of concepts within a given domain and the relationships between concepts.

One of the best known sources of external knowledge is WordNet [5, 6], a network of related words, that organizes English nouns, verbs, adjectives and adverbs into synonym sets, called synsets, and defines relations between these synsets.

In this paper, we propose a text categorization approach that is designed to fully exploiting semantic resources as it employs the ontological knowledge not only as lexical support, but also for deriving the final categorization of documents in topic categories.

Specifically, our work relates to apply WordNet for selecting the correct sense of words in a document, and utilizes domain names in WordNet Domains [7, 8] for classification purposes. Experiments are presented that show how our approach performs well in classifying a large corpus of documents. A key part of the paper is the discussion of important aspects we considered in implementing our approach.

The paper is organized as follows. Section 2 presents the proposed approach and Sect. 3 shows the experimental evaluations. Conclusions are outlined in Sect. 4.

2 The Proposed Categorization Process

We propose a method that performs three main steps: (1) discovering the semantics of words in the document, (2) disambiguating the words, (3) categorizing the document according to the semantic meaning of its nouns.

Step1: Discovering the semantics of words in the document The goal of this step is to find out the possible meanings (or senses) of a word in a document. Starting from a document represented by a vector d of its terms i.e. $d = (t_1, t_2, \ldots, t_n)$, we adopt a popular approach to the analysis of unstructured text. It is based on the bag-of-words (BOW) paradigm that uses words as basic units of information. Disregarding grammar, a text is represented as a collection of words (i.e. the parts of speech (POS) as nouns, adjective, verbs etc). The terms inside the BOW are suitably tagged for their POS. After the elimination of stop-words (conjunctions, propositions, pronouns, etc), the remaining words are used as concepts that represent the document.

Then we assume WordNet as the semantic resource that represents a set of concepts within the document and the relationships between these concepts. WordNet provides the possible senses for a large number of words and additional knowledge (such as synonyms, hypernyms, hyponyms, etc) for each possible meaning of a word. The unique characteristic of WordNet is the presence of a wide network of relationships between words and meanings, including some compound nouns and proper nouns such as "credit card" and "Margareth Thatcher".

Other works have confirmed [9, 10] that knowledge extracted from other semantic resources, such ODP [11] and Wikipedia [12], can facilitate text categorization. However, it has been observed [13] that, being not structured thesauri as WordNet, these resources cannot resolve synonymy and polysemy directly, i.e. they have limits in disambiguating words.

The ontology entities (i.e. the concepts) occurring in the analyzed document are identified by matching document terms with entity literals (used as entity names) stored in WordNet. This process shifts the analysis focus from the terms occurring in a document to the entities and semantic relationships among them and produces a set of appropriate synsets from WordNet within each term. However, these synsets do not represent the unambiguous matching between the document terms and their sense, because multiple synsets can be identified by the same concept. This drawback is motivated by the fact that documents often use synonyms, terms might be related to each other, or the term in one document is not well understood by WorldNet. In these circumstances, it is necessary to eliminate homonyms and polysemic words that negatively affect the categorization task.

Step2: Disambiguating the word sense Usually denoted as Word Sense Disambiguation (WSD), this task aims to give the correct meaning to the ambiguous words, i.e. to words with multiple meanings, according to the context in which the word is used.

For each word w, assuming it has m senses or synsets $(s_1, s_2, \ldots, s_m)$, usually known as sense inventory, the WSD method selects only one correct sense in order to build the so-called Bag of Synsets (BOS) that univocally represents the ontological knowledge about the document. As we rely on WordNet, the semantic similarity between two terms is in general defined as a function of distance between the terms in this hierarchical structure. As the concrete form of the function may be expressed

according to different criteria, there is a wide variety of approaches for calculating semantic similarities of terms [14].

For WSD purposes, our method leverages on [15] that proposes to disambiguate separately nouns, verbs, adjectives and adverbs using surrounding words in a sentence. The idea is selecting the most appropriate sense of w according to the semantic similarity between w with its context. For example, the sense of word "star" in the sentence "the sun is a star that irradiates energy" is about an astronomic fact, while in the sentence "Marylin was a movie star" it is about an actress. In this case, it is possible for a human to select the correct sense. Therefore, we try to emulate this behaviour by taking into account the context in which the word appears.

Specifically, the context of each word w in a sentence is a 2N sized window that is defined by considering the N words that surround w to the left and the N words that surround w to the right. As the complexity of the disambiguation process can vary according to the size of N, we experimentally evaluated the optimal size of the context window.

So far, we have tested the disambiguation process with four different methods proposed by Jiang and Conrath [16], Lin [17], Resnik [18], and Leacock and Chodorow [19]. The first three methods fall in the category of methods based on information contents of terms that aim to give a measure of how specific and informative a term is. The semantic similarity between two terms is based on the information content of their lowest common ancestor node. As the occurrence probability of a node decreases when the layer of the node goes deeper, the lower a node in the hierarchy, the greater its information content.

The Leacock and Chodorow method falls in the category of methods based on the hierarchical structure of an ontology that typically use a distance measure to quantify the similarity between two nodes in the directed acyclic graph of the ontology and then use this measure to assess the relatedness between the corresponding terms in the ontology. Specifically, Leacock and Chodorow calculated the number of nodes in the shortest path between two terms and then scaled the number by the maximum depth of the ontology to quantify the relatedness of the terms.

Since each category of methods has its own traits, we conducted experiments to know which method is suitable for disambiguating words within their context windows. As presented in the following section, the Jiang and Conrath method resulted in a higher precision.

A distinctive aspect of this step is that it does not depend on a specific WSD method and allows the customization of the disambiguation process. This is a very positive aspect as not all the WSD methods are valid or perform well in all the possible contexts.

Step3: Document Categorization In this step, the documents (already annotated as a result of the previous process) are attributed to categories by considering their lexical annotations. The key part of this step is the definition of the categories that are going to be considered. As in the previous step, we try to emulate the behaviour of human experts that manually label the documents as they have intensional knowledge about the document domain.

Instead of using labels or manually constructed catalogues, we rely on WordNet Domains [7, 8], a lexical resource created in a semi-automatic way by augmenting WordNet with domain labels: we consider these labels as topic categories. In more detail, WordNet synsets have been annotated with at least one semantic domain label, selected from a set of about two hundred labels structured according to the WordNet Domains Hierarchy.

Information brought by domains is complementary to what is already in WordNet. A domain may include synsets of different syntactic categories and from different WordNet sub-hierarchies. Domains may group senses of the same word into homogeneous clusters, with the side effect of reducing word polysemy in WordNet.

Each synset of WordNet was labelled with one or more labels, using a methodology which combines manual and automatic assignments. Semantic domains are areas of human knowledge (such as POLITICS, ECONOMY, SPORT) exhibiting specific terminology and lexical coherence. The label FACTOTUM was assigned in case all other labels could not be assigned.

Within WordNet Domains, each document is classified in one or more domains according to its relevance for these domains, and domains in top positions are considered more relevant for that document. Consequently, the categorization is independent from the existence of a training set and it relies solely on semantic resources as the ontology effectively becomes the classifier.

3 Experiments

In order to evaluate the performance of our approach, we used the dataset SemCor [20], created by the Princeton University. It is a subset of the English brown corpus containing about 700,000 running words. In SemCor, all the words are tagged by part of speech (POS), and more than 200,000 content words are also lemmatized and sense-tagged manually according to Princeton WordNet.

SemCor is composed by 352 tagged-documents: 186 documents are sense-tagged for every POS, and only the verbs are sense-tagged in the remaining 166 documents. For our experiments we used SemCor 2.1, and specifically we have considered the 186 documents that are sense-tagged for every POS. This complete dataset was assumed as test set to assert the precision of the proposed method. Our experiments are based on WordNet 2.1.

First, the Bag Of Words was built from the 186 documents of SemCor 2.1. Specifically, the POS, the lemma and the sense number of each word have been extracted, while ignoring no-tagged words, punctuations and stop words. Finally, we obtained a total of 186 files, one for each document, that were used as input BOWs.

Then, we disambiguated these 186 files using the approach proposed in the previous section. Specifically, we disambiguated separately nouns, verbs, adjectives and adverbs using four different methods to measure the semantic similarity between terms. We calculated the precision in disambiguating terms as the rate between the number of synsets that were correctly disambiguated by the algorithm and the total number of synsets in the BOS.

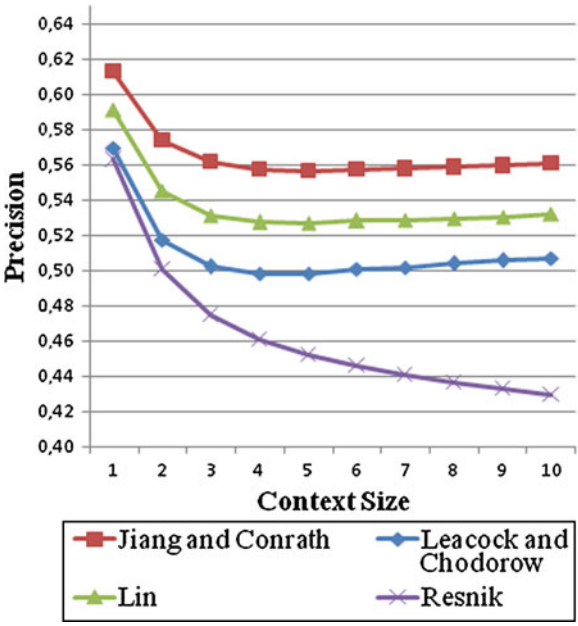

Fig. 1 Overall precision of disambiguation methods within the context size

Figure 1 shows the overall precision reached by each method in disambiguating documents within contexts of increasing size. It is evident that the size of the context is an important parameter which greatly affects the disambiguation performance and it is also sensitive to the disambiguation method. Enlarging the context window introduces noise in disambiguation process and the minimum sized context is also the optimal context. As well, the Jiang and Conrath method outperforms the other methods. Figure 2 details the precision of the four methods only for nouns and confirms previous results about the context window size and the Jiang and Conrath method. Finally, Fig. 3 details the precision reached by the Jiang and Conrath method in disambiguating each POS. As we can see, disambiguating verbs results in a very low accuracy.

Finally, the classification step was performed by considering a BOW composed only by the disambiguated nouns. Over 186 documents, the proposed approach exactly classifies 144 documents into 3 different WordNet Domains. About the remaining 42 documents, the method correctly classifies 39 documents into 2 different domains, and only 1 domain was properly attributed to 3 documents. Table 1 shows results about the classification of the first eight documents within the WordNet Domains. In brackets, the domain frequency, i.e. the number of nouns of the document that refer to that domain.

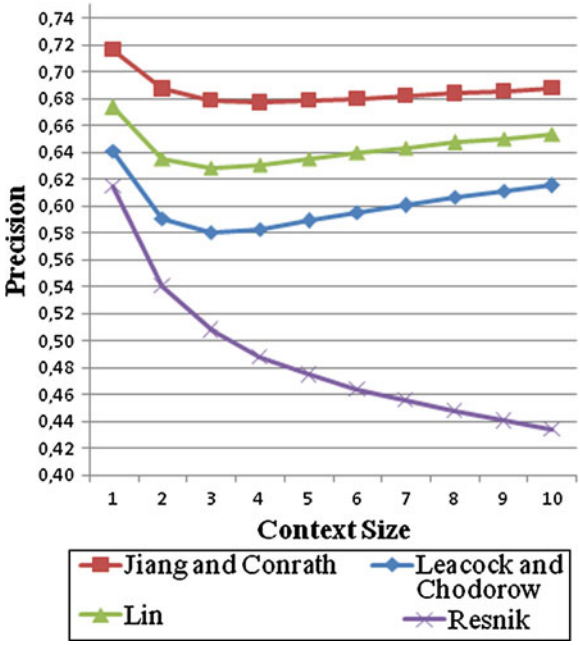

Fig. 2 Overall precision of nouns reached by each disambiguation method

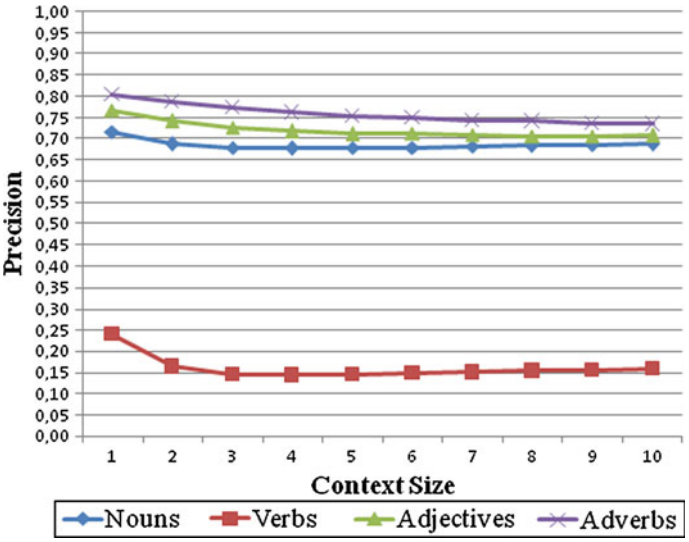

Fig. 3 Overall precision reached by Jiang and Conrath for each POS

Table 1 Classification of the first 8 documents into 3 domains

Document	1° Domain	2° Domain	3° Domain
1	Politics (49)	Law (45)	Administration (39)
2	Music (24)	Person (22)	Metrology (21)
3	Politics (39)	Geography (21)	Anthropology (20)
4	Religion (69)	Psychological_features (25)	Mathematics (18)
5	Person (56)	Psychological_features (22)	Mathematics (21)
6	Person (31)	Administration (30)	Commerce (20)
7	Medicine (32)	Psychological_features (14)	Buildings (11)
8	Publishing (61)	Literature (50)	Linguistics (39)

4 Conclusions

This paper has shown a good performance of text categorization in which semantic relations of terms drawing upon two kinds of thesauri, WordNet and WordNet Domains, were used. These external semantic resources cover a large number of domains. It enabled the easy implementation of our approach (without document labelling efforts) to handle the text categorization tasks with multiple different domains and large documents sets.

Our experimental data consisted of a total of 186 documents from SemCor Corpus. A series of experiments demonstrate the different levels of performance for four well known disambiguation methods when they are applied in different context windows.

The presented approach provides very promising results for text categorization in real world contexts where the growing body of documents makes them complex to be catalogued as it is the case of enterprise content management. A fully semantic approach to text categorization can reduce the difficulty to retrieve and manage information in an automated manner, as the volume of data becomes unmanageable giving rise to inefficiencies and costs that are not easily measurable, but have a strong impact on productivity.

Acknowledgments This research was supported by RAS, Regione Autonoma della Sardegna (Legge regionale 7 agosto 2007, n. 7), in the project *DENIS: Dataspaces Enhancing the Next Internet in Sardinia*. Stefania Dessì gratefully acknowledges Sardinia Regional Government for the financial support of her PhD scholarship (P.O.R. Sardegna F.S.E. Operational Programme of RAS).

References

1. Liu T, Yang Y, Wan H et al (2005) An experimental study on large-scale web categorization. In: Posters proceedings of the 14th international World Wide Web conference, pp 1106–1107
2. Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv 34(1):1–47
3. Zhu XJ (2008) Semi-supervised learning literature survey. http://pages.cs.wisc.edu/~jerryzhu/research/ssl/semireview.html

4. Bai R, Wang X, Liao J (2010) Extract semantic information from WordNet to improve text classification performance. In: Proceedings of the international conference on Advances in computer science and information technology. LNCS 6059:409–420
5. Miller GA (1995) WordNet: a Lexical database for English. Commun ACM 38(11):39–41
6. Fellbaum C (ed) (1998) WordNet: an electronic Lexical database. MIT Press, Cambridge
7. Magnini B, Cavaglià G (2000) Integrating subject field codes into WordNet. In: Proceedings of LREC-2000, 2nd international conference on language resources and evaluation, Athens, Greece, pp 1413–1418
8. Bentivogli L, Forner P, Magnini B et al (2004) Revising WordNet domains hierarchy: semantics, coverage, and balancing. In: Proceedings of COLING workshop on multilingual Linguistic resources. Switzerland, Geneva, pp 101–108
9. Kulkarni S, Singh A, Ramakrishnan G et al (2009) Collective annotation of Wikipedia entities in web text. In: Proceedings of ACM KDD, pp 457–466
10. Mihalcea R, Csomai A (2007) Wikify!: linking documents to encyclopedic knowledge. In: Proceedings of ACM CIKM, pp 233–242
11. Ontology Design Patterns, http://ontologydesignpatterns.org/
12. Bizer C, Lehmann J, Kobilarov G et al (2009) DBpedia: a crystallization Point for the Web of Data. J Web Semant: Sci, Serv Agents WWW 7:154–165
13. de Buenaga Rodriguez M, Gomez-Hidalgo J, Diaz-Agudo B (1997) Using WordNet to complement training information in text categorization. In: Proceedings of the 2nd international conference on recent advances in natural language processing (RANLP'97), pp 150–157
14. Gan M, Dou X, Jiang R (2013) From ontology to semantic similarity: calculation of ontology-based semantic similarity. Sci World J , Article ID 793091, p 11
15. Basile P, De Gemmis M, Gentile AL et al (2007) UNIBA: JIGSAW algorithm for Word Sense Disambiguation, In: Proceedings of the 4th international workshop on semantic evaluations (SemEval-2007), pp 398–401
16. Jiang J, Conrath D (1997) Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings on international conference on research in computational linguistics, pp 19–33
17. Lin D (1998) An information-theoretic definition of similarity. In: Proceedings of the international conference on machine learning, Madison, pp 296–304
18. Resnik P (1995) Using information content to evaluate semantic similarity in a taxonomy. In: Proceedings of the 14th international joint conference on, artificial intelligence, pp 448–453
19. Leacock C, Chodorow M (1998) Combining local context and WordNet similarity for word sense identification. In: Fellbaum C (ed) WordNet: an electronic lexical database, pp 265–283 (MIT Press)
20. Miller GA, Leacock C, Tengi R et al (1993) A semantic concordance. In: Proceedings of ARPA workshop on human language technology, pp 303–308

Emotion Analysis on Turkish Texts

Z. Boynukalin and P. Karagoz

Abstract Automatically analyzing the user's emotion from his/her texts has been gaining interest as a research field. Emotion classification of English texts is studied by several researchers and promising results have been achieved. In this work, an emotion classification study on Turkish texts is presented. To the best of our knowledge, this is the first study conducted on emotion classification for Turkish texts. Due to the nature of Turkish language, several pruning tasks are applied and new features are constructed in order to improve the emotion classification accuracy. We compared the performance of several classification algorithms for emotion analysis and reported the results.

1 Introduction

Sensing the emotions from text is gaining more interest, since textual information is not only used for describing events or facts but is also a good source for expressing opinion or emotional state, which makes texts a good source for sentiment and emotion analysis. Classification of the texts as generally positive or negative is the focus of sentiment analysis, and one step further of it, recognizing the particular emotion that is expressed in text is the task of emotion analysis.

Emotion analysis of texts can help developing computers that are able to recognize and analyze human emotion, thus computers that are emotionally intelligent. Market analysis, affective computing, natural language interfaces, and e-learning environments are the example applications of this field. Previous studies on emotion analysis are mainly concentrated on English texts and we aim to close the gap of analysis of Turkish texts.

Turkish is one of the morphologically rich languages (MRL) with its agglutinative structure. That means most of the words are constructed by adding suffixes to

Z. Boynukalin (✉) · P. Karagoz
Middle East Technical University, Ankara, Turkey

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_16,

the roots of the words. Morphology of the language decides the rules of language on creation of the word. This structure provides to form larger number of words from a single root, and make natural language processing (NLP) tasks harder than other languages. Morphological parser for Turkish generally does not return a single answer, because there exists more than one possible constructions of the words, which is morphological ambiguity of words. Therefore we applied several preprocessing tasks due to the nature of Turkish language and evaluated the effect of various feature combinations. Our contribution in this work can be summarized as follows:

- We developed a framework for the analysis of Turkish texts for emotion classification. To achieve our goal, a benchmark dataset is translated to Turkish and used for training and evaluation.
- Existing preprocessing and feature selection approaches that were applied to English texts are modified and extended to get better results with a morphologically rich language, Turkish.
- Different feature selection and feature weighting approaches are combined to increase the success of the system.

This paper is organized as follows. In Sect. 2, previous studies on emotion analysis are summarized. In Sect. 3, the proposed framework for emotion analysis on Turkish texts is presented. In Sect. 4, experimental results are given. Finally, the paper is concluded in Sect. 5.

2 Previous Work

The studies on emotion analysis in the literature are mostly on English texts. In this section, we summarize these studies.

Neviarouskaya et al. [1] applied the rule-based approach for affect classification in on-line communication environments. The work in [2] also proposed a rule based method, providing a deeper analysis. In [2], for each sentence, subject-verb-object triplets are extracted. A lexicon consisting of word-valence pairs is used in the study and by applying a set of rules to the triplets a valence for the sentence is calculated. Another study [3] also uses a similar approach for emotion extraction.

Liu et al. [4], demonstrated an approach that uses a large scale real world knowledge about the inherent affective nature of everyday situations to understand the underlying semantics of knowledge. In the study, sentences are classified into six basic emotions in the context of an e-mail agent. Alm [5] used children fairy tales data and classified the emotional affinity of sentences using supervised machine learning with SNoW (Sparse Network of Winnows) learning architecture.

In another study, [6, 7], the aim is to investigate the expression of emotion in language. Categorization of sentences into Ekman's six basic emotions [8] is performed on the data collected from blogs. Corpus-based unigram features and features derived from emotion lexicons are used in machine learning. Strapparava and Mihalcea [9] applied several knowledge based and corpus based methods for the

automatic identification of six basic emotions. Emotion analysis of news headlines is the basic focus. To recognize emotions in news headlines, Katz et al. [10] applied a supervised approach.

Calvo and Kim [11] proposed a dimensional approach that can be used for visualization and detection of emotions. They compared their approach with the statistically driven techniques. It is stated that emotions are better represented in a three dimensional space of valence, arousal, and dominance.

Danisman and Alpkocak [12] used ISEAR dataset and applied Vector Space Model (VSM) for classification. Five emotion classes, which are anger, disgust, fear, sadness and joy, are used. Classification by using Naive Bayes, Support Vector Machines and Vector Space Model classifiers are compared. Also to improve the success, emotional words from Word Net Affect and WPARD are added to the training dataset. With the use of all data sources, this study reached 70.2 % accuracy with 5 classes.

3 Emotion Analysis Framework

Emotion analysis method employed is basically classification of a given text into the classes of anger, fear, sadness and joy. In order to define the text as a term vector, firstly a set of preprocessing tasks are applied to extract terms, and then feature selection is done. The last step is applying classification technique to construct the model.

3.1 Preprocessing

In the preprocessing phase, on a given text, punctuations and proper names are removed, morphological analysis and spell checking of the words is applied and stop words removal is done. Some exceptions are needed to be implemented for some words. Zemberek library [13] is used for morphological analysis.

Handling Punctions and Proper Names. Punctuations and proper names are not needed and removed from data since they do not provide useful information in our analysis. The emotion in the sentence is not related to the proper names and removing them increases the efficiency of the analysis. The proper names are detected with a simple approach. If the word is in the middle of the sentence and starting with a capital letter, or if the word contains the character apostrophe('), then the word is thought to be a proper name and removed from the sentence.

Handling Typing Problems. Since a word may be mistyped, spell checking of the words is needed. By using Zemberek's functions each word is controlled and if the word does not exist in the lexicon directly, then suggestion function of Zemberek is used and the first suggestion, which has the highest probability, is considered as the correct version of the word. If there does not exist a suggestion, the word is removed from the sentence. An important reason for needing a spell check is typing by using

English keyboard. Authors do not type the letters ı, ö, ü, ç, ş, ğ, instead i, o, u, s, g are typed in English keyboard. These words must be corrected not to cause an information loss. For example, *üzgün* (*sad*) is an important emotional word, if we do not correct the word *uzgun* as *üzgün* in a sentence, then the feature *üzgün* will not appear in the sentence and this emotional word will be disregarded erroneously.

Handling Stop Words. Stop words are the words that are filtered out of our data, that is they do not provide any information for computation. Furthermore, removing these words increases the success of the system in certain cases. A public stop word list that is found on the Web, [1] is rearranged for emotion classification. Some words that may be important in our case, such as *kimse* (*nobody*), *çok* (*many*), *niye* (*why*), *rağmen* (*in spite of*), are removed from the list.

Morphological Analysis and Stemming. Morphological analysis has high importance due to the agglutinative structure of Turkish. The inflectional suffixes of the words should be cleaned, whereas the formation suffixes should not, because formation suffixes are the suffixes that change the words meaning completely. As stated earlier, Zemberek gives the results in decreasing order of probability, and we consider the first suggestion as the correct result. Since our system is based on the word relations in data, the words that have the same root and meaning are important. The aim of morphological analysis is extracting the words with the same meaning and root, even if they appear to be different due to the suffixes they have. Stemming, which is considering the roots of each word, is the general convention in this process. We also used the roots of the words, however we added some keywords related to the changed meaning of the word.

Handling Negation. In Turkish language, negation can be given in several ways. The first one is the general way of negating a verb with a suffix. Two types of suffixes exist for negating a word. The first one is negating a noun with a suffix, which reverts the meaning of the noun. This is like the less suffix in English, such as homeless. In Turkish it is *siz* suffix as in the word *evsiz (homeless)*. Another negation type is with the word *değil*, which negates the meaning of the verb or the noun that it comes after. For example, the sentence I am not sick can be expressed in Turkish as *Hasta değilim*. Here *hasta* means sick, and should be negated. The last negation type is done with the word *olmamak*. The English example not being sick can be translated to Turkish as *hasta olmamak*. Here, as in theprevious case, the word *hasta* should be negated.

Handling Emotion-sensitive Suffixes. Some suffixes in Turkish, assign important meaning to the word. As an example, *-ebilmek* suffix in word *gidebilmek*, means being able to go in English. The phrase able to is important in emotion analysis, since it gives a sad feeling when used with negation, like I was not able to go. In Turkish if we do not handle the suffix properly, the sentence would mean I did not go, and this brings information loss. Therefore, examining these suffixes increases the success of the system. Another suffix, *-ceğiz* also expresses sad feeling strongly, and finally suffix *-sin* expresses giving order to someone and is especially used in

[1] http://www.devdaily.com/java/jwarehouse/lucene/contrib/analyzers/common/

anger emotion. When one of these suffixes occurs in the word, a special keyword is added to the sentence, regarding the suffix.

Handling Exceptions. Some exceptions are implemented, due to the confusion of negation suffix in some cases. For example, the suggested result of Zemberek for the word *gülmeye* in the sentence *Gülmeye başladım. I started laughing*, says that there is a negation in the word. This is due to the structure of Turkish, and Zemberek library analyzes the word with a different approach, as in the sentence *Gülmeyesin*. (*You should not laugh*). However,the use of the second approach is not very common in the language, therefore, we defined an exception so that for the words ending with *-meye* and *-maya*, negation is not handled even if the best result of Zemberek says so.

3.2 Feature Selection

After preprocessing, the next task is the feature selection. Different features are combined and tested in order to obtain the best representation of the text for emotion analysis. Initially, all distinct words are considered as features, as in bag of words (BOW) approach. Then n-grams are extracted. For instance, in our ISEAR dataset there exists 3,757 distinct words (unigrams), 26,314 distinct bigrams and 34,234 distinct trigrams. Weighted Log Likelihood Ratio (WLLR) [14] scoring is used for generating scores to the features.

By using WLLR, the most distinctive features are extracted and this affects the success of the system considerably. Each unigram, bigram, trigrams score is calculated by using the WLLR formula. Then, n-grams are sorted in descending order of scores and the top n n-grams are selected in this study. This n value is determined on the basis of the experimental results. Since this selection is done for each class, the most valuable n features are extracted for each class.

For the selected features, we have implemented three different feature value assignment methods: binary score assignments on the basis of the presence of the term in the text, *tf* weighting *tf-if* weighting. In our evaluation on ISEAR data set, the best result is obtained under *tf*. Therefore, the results presented in the experiments are obtained under *tf* weighting.

3.3 Classification

The main focus in this study is emotion classification and several methods are used and compared for this purpose. Our dataset, ISEAR dataset, has four basic emotions, joy, sad, fear and anger. Classification with these classes are performed with different methods and several tests are applied. The effects of the following items are examined: using different classification methods (Naive Bayes Classifier, Complement

Naive Bayes and SVM, from Weka [15]), using bigrams and trigrams in addition to unigrams, using WLLR scoring in feature selection.

4 Experiments

In this paper, the details of the experimental settings and experiment results are presented. Experiments are performed on three sets of data which are ISEAR dataset. The effects of using different methods are analyzed and compared to each other.

4.1 Dataset

Within the scope of our work, ISEAR data set [16] dataset is translated into Turkish, in order to be used in emotion classification on Turkish texts. ISEAR is mainly a project, in which 3,000 people from 37 countries are involved. These people are asked to write the situations that they experienced 7 major emotions and reactions to those emotions. We collected a team of 33 people for the translation task. After the translations are collected, all the results are controlled for their reliability. Meanwhile, it should be noted that the typing mistakes in the data are not corrected manually. Zemberek library is used to correct these words. There exists some sentences in the dataset that are included in more than one classes, since some people give the same answers for different emotions. We did not eliminate those sentences to have a dataset that is similar to the original version. Samples of the original dataset and translated versions can be seen in Table 1 . Original dataset contains 7 emotions in total, and 4 of them are translated to Turkish for this study. In the translated set there are 1,073 instances for joy class, 1,036 instances for sad class, 1,083 instances for anger class and 1,073 classes for fear class.

Table 1 Samples from ISEAR Data

Sentence and translation	Emotion
Having passed an exam	joy
Bir sınavdan geçtiğimde	
Saw poverty in the countryside	sad
Kırsal bölgelerde gördüğüm yoksulluk	
When my sister took something that belonged to me without my permission	anger
Kız kardeşim eşyalarımı izinsiz kullandığında	
While paddling in the river during a storm. I feared drowning	fear
Fırtınalı bir havada nehirden geçerken. Boğulmaktan korkmuştum	

Table 2 Descriptions of feature sets

Feature set	No of Feature	Details
FS 1	3,757	All distinct words
FS 2	800	100 unigram + 100 bigram from each class by WLLR
FS 3	1,600	200 unigram + 100 bigram + 100 trigram from each class by WLLR
FS 4	2,400	200 unigram + 200 bigram + 200 trigram from each class by WLLR
FS 5	2,400	300 unigram + 200 bigram + 100 trigram from each class by WLLR

4.2 Results

At the first step, classification is performed by using all distinct words as features under *tf* weighting for feature value assignment. After that, rather than using all distinct words as features, we used WLLR ranking for feature selection, and performed experiments with different combinations of n-grams. It should be noted that the features with high scores are selected equally for each class.

As presented in Table 3, we can say that the most successful result of NB classifier is the one with 2,400 features (200 unigram, 200 bigram and 200 trigram from each class) selected by using WLLR scoring and *tf* weighting method and as the best score 78.12 % of accuracy is obtained. The increasing number of features with WLLR scoring resulted in a higher accuracy up to a limit, and then the accuracy started to decrease.

The most successful result of Complement Naive Bayes classifier is the one with 2,400 features (200 unigram, 200 bigram and 200 trigram from each class by WLLR) as presented in Table 4. The highest accuracy obtained is 81.34 %, which is a highly encouraging result. The effect of using different weighting methods did not make a big difference as did in NB classifier.

As seen in Table 5, the highest result of SVM classifier is obtained with the same feature set of NB and CNB Classifiers, 2400 features in total, 200 unigrams, 200

Table 3 Classification results of ISEAR Dataset with NB classifier

Feature set	Accuracy (%)	Kappa	F-measure
FS 1	71.50	0.62	0.72
FS 2	63.91	0.52	0.65
FS 3	76.74	0.69	0.77
FS 4	78.12	0.71	0.78
FS 5	78.00	0.71	0.78

Table 4 Classification results of ISEAR Dataset with complement NB classifier

Feature set	Accuracy (%)	Kappa	F-measure
FS 1	74.70	0.66	0.75
FS 2	61.57	0.49	0.62
FS 3	78.97	0.72	0.79
FS 4	81.34	0.75	0.81
FS 5	81.27	0.75	0.81

Table 5 Classification results of ISEAR Dataset with SVM classifier

Feature set	Accuracy (%)	Kappa	F-measure
FS 1	72.54	0.63	0.72
FS 2	61.01	0.48	0.76
FS 3	74.47	0.66	0.75
FS 4	75.87	0.68	0.76
FS 5	75.19	0.67	0.75

bigrams and 200 trigrams selected with WLLR scoring as. The highest accuracy is 75.87 %.

To sum up the results, the highest accuracies reached by each classifier are; NB-78.12 %, CNB-81.34 % and SVM-75.87 %. CNB is the most successful classifier among the three techniques. The effect of using WLLR in the feature selection process is especially important since it increases success. The experiments have shown that feature selection is effective for the result, and WLLR scoring provides a good performance for selecting the distinctive features.

In Table 6, the detailed result of the experiment that gives highest accuracy is shown. The table presents the confusion matrix and the accuracy of each class can be seen. The highest accuracy is obtained for class *joy*, and the lowest accuracy is obtained for class *sad*. Maximum confusion occurs between *anger* and *sad* classes, such that 111 of the items that are in *sad* class, classified to be in *anger* class erroneously. This is not unexpected when we look at the dataset closer. This result can be explained by the observation that the expressions of people for sadness and anger are similar. In some situations, a fact makes someone angry, and the same fact makes the other person sad.

Table 6 Confusion matrix of ISEAR Dataset for complement NB classifier (under FS4)

Classt	Joy	Anger	Sad	Fear	Accuracy
Joy	915	68	51	39	0.85
Anger	70	881	48	84	0.81
Sad	83	111	780	62	0.75
Fear	53	67	60	893	0.83

5 Conclusion

In this work, we focused on emotion analysis of Turkish texts and it is shown that using machine learning methods for Turkish texts on analysis of emotions gives promising results.

Within the scope of this study, ISEAR dataset, which is composed of questionnaire answers of many people from different countries and cultures, is translated to Turkish with the help of 33 people. In the preprocessing phase, in addition to conventional tasks including morphological analysis and stemming, morphological structure of Turkish is taken into consideration and several exceptions for Turkish language are added. Handling negations and some special suffixes is important, since it expands the knowledge and increases the accuracy. Three different classification techniques, SVM, NB and CNB are applied on Turkish ISEAR dataset with 4 classes. The best result is obtained by CBN with 81.34 % classification accuracy.

This study can be further extended by analysis for more types of emotions as a future work. In addition, performance of the proposed framework for other morphologically rich languages can be investigated.

References

1. Neviarouskaya A, Prendinger H, Ishizuka M (2011) Affect analysis model: novel rule-based approach to affect sensing from text. Nat Lang Eng 17:95–135
2. Shaikh MAM (2011) An analytical approach for affect sensing from text. PhD Thesis.
3. Yashar M (2012) Role of emotion in information retrieval. PhD Thesis, University of Glasgow.
4. Liu H, Lieberman H, Selker T (2003) A model of textual affect sensing using real-world knowledge. In: Proceedings of the 2003 international conference on intelligent user interfaces, ACM, pp 125–132.
5. Alm CO, Roth D, Sproat R (2005) Emotions from text: machine learning for text-based emotion prediction. In: Proceedings of the conference on human language technology and empirical methods in natural language processing, pp 579–586
6. Aman S, Szpakowicz S (2007) Identifying expressions of emotion in text. In: Text, speech and dialogue, pp 196–205
7. Aman S, Szpakowicz S (2008) Using roget's thesaurus for fine-grained emotion recognition. In: Proceedings of the 3rd international joint conference on natural language processing, pp 296–302.
8. Dalgleish T, Power MJ (1999) Handbook of cognition and emotion. Wiley, Wiley Online Library
9. Strapparava C, Mihalcea R (2008) Learning to identify emotions in text. In: Proceedings of the 2008 ACM symposium on Applied, computing, pp 1556–1560.
10. Katz P, Singleton M, Wicentowski R (2007) Swat-mp: the semeval-2007 systems for task 5 and task 14. In: Proceedings of the 4th international workshop on semantic evaluations, pp 308–313.
11. Calvo RA, Mac Kim S (2012) Emotions in text: dimensional and categorical models. Comput Intell.
12. Danisman T, Alpkocak A (2008) Feeler: emotion classification of text using vector space model. In AISB 2008 convention communication, interaction and social. Intelligence 2:53–59
13. Akın AA, Akın MD (2007) Zemberek, an open source NLP framework for turkic languages. Structure.

14. Nigam K, McCallum AK, Thrun S, Mitchell T (2000) Text classification from labeled and unlabeled documents using em. Mach Learn 39(2–3):103–134
15. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The weka data mining software: an update. ACM SIGKDD Explor Newsletter 11(1):10–18
16. Scherer KR, Wallbott H (2004) Evidence for universality and cultural variation of differential emotion response patterning. J Personality Soc Psychol 66:310–328

A Comparative Study to Determine the Effective Window Size of Turkish Word Sense Disambiguation Systems

Bahar İlgen, Eşref Adalı and A. Cüneyd Tantuğ

Abstract In this paper, the effect of different windowing schemes on word sense disambiguation accuracy is presented. Turkish Lexical Sample Dataset has been used in the experiments. We took the samples of ambiguous verbs and nouns of the dataset and used bag-of-word properties as context information. The experi-ments have been repeated for different window sizes based on several machine learning algorithms. We follow 2/3 splitting strategy (2/3 for training, 1/3 for test-ing) and determine the most frequently used words in the training part. After re-moving stop words, we repeated the experiments by using most frequent 100, 75, 50 and 25 content words of the training data. Our findings show that the usage of most frequent 75 words as features improves the accuracy in results for Turkish verbs. Similar results have been obtained for Turkish nouns when we use the most frequent 100 words of the training set. Considering this information, selected al-gorithms have been tested on varying window sizes {30, 15, 10 and 5}. Our find-ings show that Naïve Bayes and Functional Tree methods yielded better accuracy results. And the window size ± 5 gives the best average results both for noun and the verb groups. It is observed that the best results of the two groups are 65.8 and 56 % points above the most frequent sense baseline of the verb and noun groups respectively.

1 Introduction

Word Sense Disambiguation (WSD) is one of the critically important tasks of *Natural Language Processing* (NLP) area which aims to determine the correct usage (*or sense*) of the ambiguous words. It has received more attention with the increasing

B. İlgen (✉)
Istanbul Kültür University, Istanbul, Turkey
e-mail: b.ilgen@iku.edu.tr

E. Adalı · A. C. Tantuğ
Istanbul Technical University, Istanbul, Turkey

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_17,

needs of the research area. And it is still a widely studied task since most of the NLP applications require sense information to achieve acceptable performance. These NLP applications include *Machine Translation* (MT), automatic summarization, language understanding, and many others.

WSD systems basically assign the most confident sense to the ambiguous word by considering all possible set of candidates in the context. Context of the ambiguous word supplies important clues to find and assign correct meaning (or usage) to the ambiguous word. Although neighbor words usually include this information, it will be better to know the effective range in advance.

Context is usually regarded as set of word and characters falling in a specified distance. The context can be used in two different ways [1]:

- *Bag-of-words approach*
- *Relational information*

The bag-of-words model is a simplifying representation used in NLP and *Information Retrieval* (IR). In this model, a text (such as a sentence or a document) is represented as an unordered collection of words, disregarding word order and grammar. In relational information approach, context is used in terms of some relations to the target word. These relations cover collocations, selectional preferences, syntactic relations, semantic categories etc.

In WSD tasks, we use the term "target word" (or head word) to describe the candidate ambiguous word. This is the word to be disambiguated. In the scope of this work, we work on varying window sizes that surround our ambiguous *target word* and check their contribution to the results. The windowing scheme considers the target word to be center of the window. For example, when we use window size ± 10, whole window includes the 21 words together with our target word.

Windowing scheme have been used in the scope of different NLP tasks. If this information is known in advance, it can be used to increase the performance of the whole system. It is useful to eliminate redundant information that coexists with the ambiguous target word. Determination of the effective range makes easier and faster to reach the suitable sense of the ambiguous word. On the other hand, the nature of the task may already require using windowing scheme.

In the scope of this work, we follow bag-of-words approach and use most frequent content words of the training data. The most frequent word set has been used as features and whole data is encoded using binary information of existence of each feature in the context. We have also determined the effective size of content words (or features) before evaluating the windowing scheme. The effective number of bag-of-words features has been determined by evaluating the accuracy results after using most frequent 100, 75, 50 and 25 content words. After finding the proper feature size separately for noun and verb sets, we've kept these settings for all the experiments to determine the effective window size.

The paper is organized as follows. Section 2 describes the previous related work to determine effective window size of target word in WSD systems. Section 3 introduces the dataset that we use and some of the necessary preparation steps.

Section 4 describes the selected algorithms and our experimental results. Finally, Sect. 5 presents our conclusions.

2 Related Work

Since windowing schemes are needed in many NLP tasks, it is important to determine the effective distance around the target word. Some researchers have studied on this topic [2].

In one of the study [3], the performance of different windowing schemes using two conceptual hierarchies based semantic similarity metrics was analyzed. They have used Maximum Relatedness Disambiguation algorithm for the experiments [4]. The algorithm uses a quantitative measure of similarity between word senses in context to disambiguate the candidate ambiguous word. Similar experiments have been carried out on different windowing schemes. Experiments have been done by using WordNet and subset of nouns in SenSeval-2 English Lexical Sample dataset. The subset they used contains 914 noun instances of the data source which includes 5 target words (*art*, *authority*, *bar*, *bum*, and *chair*). Different similarity metrics have been tested for varying windowing schemes. Their results suggest that the best performing window size on average is 7.

Some other experiments have been carried out for Hindi WSD. The effect of the context window size, stop word removal and stemming has been investigated to determine their impact level on WSD accuracy [5]. They state that the increase in window size improves the results after stop word removal. It is also said that, in some cases this also increases number of irrelevant words and results in performance degradation.

Yarowsky [6] has also conducted several experiments using different window sizes. The findings suggest that local ambiguities need smaller windows (window of k = 3 or 4). On the other hand, semantic or topic-based ambiguities need larger windows of 20–50 words.

3 Dataset and Preparation

3.1 Dataset

WSD is a complex task since it can be separated into subgroups of the different approaches. Considering the selection strategy of the candidate ambiguous word, we can follow: (1) *Lexical Samples* (LS) approach or, (2) *All words* (AW) approach. LS approach aims to disambiguate the occurrences of small sample of previously selected target words. It is based on usage of hand-labeled training texts of previously selected words. LS datasets have been widely used with the supervised WSD techniques.

All-words approach is the second option which aims to disambiguate all the words in a running text.

We've chosen LS approach and conducted our experiments on *Turkish Lexical Sample Dataset* (TLSD) which has been prepared to carry out Turkish NLP tasks on it [7]. Before collection of TLSD samples, several data samples [8, 9] in different languages were examined to create our lexical dataset. Well-known Turkish websites on news, health, education and sports were taken as main resources for this work.

This dataset covers the most ambiguous Turkish nouns and verbs. Both noun and verb groups consist of 15 candidate ambiguous words each of which has certain number of samples. These ambiguous word candidates have been selected by considering the most ambiguous words in Dictionary of *Turkish Language Association* (TLA) [10]. The average length for a lexical sample varies between 5 and 10 sentences. Each sample has one sense (i.e., usage) of ambiguous word which is marked as "*head*" word. This is the word to be disambiguated. These words have been labeled with the proper sense number of the dictionary of TLA.

The total number of samples in the dataset is 3616. We have excluded the 3rd group named others[1] of the dataset and use verbs and nouns groups for our experiments. There are approximately 100 samples under each ambiguous word category. The samples are stored in XML format which includes the sense labeled head words in it. Head words of these samples had been labeled by 5 voters using a questionnaire. Table 1 shows the number of senses of both noun and verb groups. "Number of senses" column shows the total number of senses that we found in TLA. "Number of

Table 1 Word groups of Turkish Lexical sample dataset

Nouns			Verbs		
Target word	Number of senses	Number of actual senses	Target word	Number of senses	Number of actual senses
açık	16	12	aç	27	15
baskı	8	7	al	33	15
baş	13	9	at	33	18
derece	7	6	bak	17	14
dünya	7	7	çevir	15	11
el	10	5	çık	56	19
göz	13	8	geç	38	16
hat	9	9	gel	36	18
hava	14	10	gir	19	7
kaynak	8	7	gör	20	14
kök	12	7	kal	21	12
kör	7	6	ol	25	17
ocak	11	6	sür	16	11
yaş	9	7	ver	22	14
yüz	15	12	yap	20	11

[1] Adjectives, adverbs and prepositions.

actual senses" column shows the numbers which have been covered in lexical samples of TLSD. There is a small difference between these values since we considered the more common and frequent senses along the data collection phase. We have omitted some of the rare uses of the words. This dataset has been used in our previous studies [7, 11] on Turkish WSD using collocational features of the candidate head word. In the scope of this work, we follow bag-of-words approach to determine effective window size. In other words, we consider the context co-occurrence properties.

3.2 Preparation and Encoding

A pre-processing step is applied to obtain stemmed forms of the words and remove redundant morphological information since Turkish exhibits inflectional and derivational suffixation. As a first preparation step, we removed all stop words defined for Turkish Language from TLSD samples. After obtaining content words, all lexical samples of the ambiguous word groups were morphologically analyzed. We used a finite-state two-level Turkish morphological analyzer [12] and a morphological disambiguation tool [13]. for morphological decomposition and morphological disambiguation tasks respectively. The latter tool is needed since the output of the analyzer is ambiguous.

Before testing different windowing schemes, we did additional experiments to determine the effective number of features. To achieve this, we first determined the most frequently used content words in the training part. We have scanned the entire training portion (i.e., we took 67 % of the dataset as training portion which approximately equals to 67∼70 samples in the dataset) and ranked the most frequent content words. We repeated this step 4 times and determined the most frequent 100, 75, 50 and 25 content words of the lexical samples. For each feature set, both training and test portions of the dataset are encoded using this information. For example, if we take most frequent 25 words as features, we encoded both training and test parts (samples) by using these 25 words. We have used a binary vector structure which assigns "1" to corresponding cell of the vector if the feature exists in the lexical sample and assigns "0" otherwise. After repeating similar test for 50, 75 and 100 most frequent words, we have obtained the accuracy results by giving *.arff*[2] files consisting of these vectors to Weka [14]. Our findings show that the most frequent 75 and 100 content words yielded better accuracy results for verb and noun groups respectively. In the rest of the experiments, we kept and used these settings to determine effective window size.

The samples have been adjusted to take ±n words which surround the head word. We decided to carry out our experiments for different values of the "n" which consist of 30, 15, 10 and 5 (preceding and following) words.

[2] An ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes.

4 Experimental Results

In the final phase, we investigated several basic machine learning (ML) algorithms on our annotated dataset. We took the window size ±30, ±15, ±10 and ±5 for selected bag-of-word features. Considering the effective feature size, we took most frequent 75 and 100 content words as features for verbs and nouns respectively.

4.1 Algorithms and Approach

In the scope of this work, we have tested six different ML algorithms on our data set consisting of **Naïve Bayes**, **IBk, KStar** (Exemplar-based), **J48** (C4.5 algorithm-decision tree), **SVM** (Support Vector Machines) and **FT** (Functional Trees). We choose these methods from different ML algorithm classes since the experimental results provide an insight about the effective groups of algorithms. Weka 3.6.5 has been used for the experiments. We have chosen 2:3 splitting ratio and divided our data into two sets: one with ∼67 percent of the source data, for training the model, and one with ∼33 % of the source data, for testing the model.

4.2 Results

The experiments have been repeated on two sets consisting of; Turkish nouns and Turkish verbs of TLSD. Although different algorithms yielded better results for certain cases, we obtained better results by using Naïve Bayes and FT methods considering the window size average both for verb and the noun groups. Considering this and the ease of comparison among accuracy results of different window sizes, we present the results belong these two algorithms for Turkish verb and noun classes. Figure 1 presents the Naïve Bayes and the FT algorithm results of noun group. Similarly, Fig. 2 shows the accuracy results of the verb group. The average results of different window size (WS) values have been presented together with the value of most-frequent-sense baseline (MFB). Because Turkish verbs are much more ambiguous than the noun group (i.e., sense labels up to 60 or more), the baseline values are lower. There are some cases which we have obtained better results with different windowing schemes. On the other hand, average success ratio of Naïve Bayes and FT algorithm is better for both noun and verb groups. Accuracy results are higher for the noun group since the MFB baselines are also higher for this group. Although we selected FT and Naïve Bayes methods to present our results of the windowing scheme, other algorithms give similar and good results for varying window sizes. It is observed that the best results of the two groups are 65,8 and 56 % points above the most frequents sense baseline of verbs and nouns respectively. The window size averages of the noun and verb groups have been also presented in Table 2.

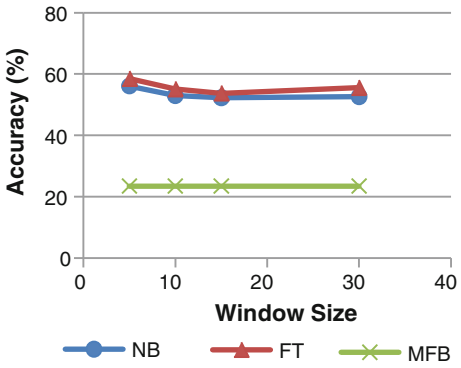

Fig. 1 Naive Bayes and FT averages of Turkish nouns for given window size

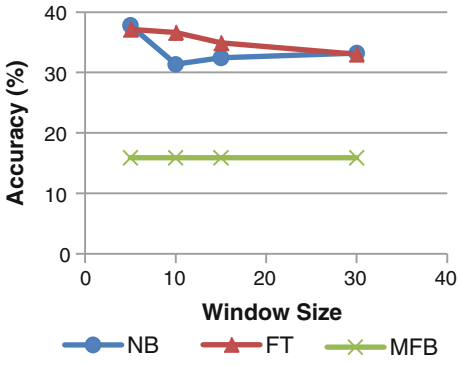

Fig. 2 Naive Bayes and FT averages of Turkish verbs for given window size

Table 2 Average accuracy (%) values for given window size of noun and verb groups

Accuracy(%)		MFB	30	15	10	5
Noun	Naïve Bayes	23,47	52,64	52,24	53,01	56,08
	FT		55,51	53,73	55,07	58,47
Verb	Naïve Bayes	15,87	33,19	32,44	31,35	37,81
	FT		33,03	34,90	36,61	37,14

5 Conclusions

In this work, we presented the accuracy results of different windowing schemes on Turkish WSD data. Although there are few studies on windowing scheme, most of them have been investigated on English and other languages. We have carried out these experiments on Turkish lexical sample data by using different methods. We have both determined the effective number of bag-of-word size (i.e., as an answer to the question of, how many content words will be included as features to encoding scheme?) and effective window size. Our findings are compatible with the previous studies. The results show that the smaller window scope is more informative on

disambiguation process. And the best performing window size on the average is 5. We hope that this information will be useful in different approaches of the WSD area.

References

1. Ide N, Veronis J (1998) Introduction to the special issue on WSD: the state of the art; special issue on word disambiguation. Comput Linguist 24(1):1–40
2. Navigli R (2009) Word sense disambiguation: a survey. ACM Comput Surv 41(2):1–69. doi:10.1145/1459352.1459355
3. Altintas E, Karsligil E, Coskun V (2005) The effect of windowing in word sense disambiguation computer and information sciences-ISCIS 2005. Springer, pp 626–635.
4. Banerjee S, Pedersen T (2002) An adapted Lesk algorithm for word sense disambiguation using WordNet computational linguistics and intelligent text processing. Springer,pp 136–145.
5. Singh S, Siddiqui TJ (2012) Evaluating effect of context window size, stemming and stop word removal on Hindi word sense disambiguation. In , Paper presented at the international conference on information retrieval and knowledge management (CAMP)
6. Yarowsky D (1993) One sense per collocation. Paper presented at the proceedings of the workshop on human language technology, In
7. Ilgen B, Adali E, Tantug A (2012a) Building up lexical sample dataset for Turkish word sense disambiguation. In , Paper presented at international symposium on the innovations in intelligent systems and applications (INISTA)
8. Seo H-C, Rim H-C, Kim S-H (2001) KUNLP system in Senseval-3. In: Paper presented at the proceedings of SENSEVAL-2 workshop.
9. Escudero G, Màrquez L, Rigau G (2004) TALP system for the english lexical sample task. In: Paper presented at the Proceedings of SENSEVAL-3, Barcelona, Spain.
10. Güncel Türkçe Sözlük (2005) Turkish Language Association.
11. Ilgen B, Adali E, Tantug A (2012b) The impact of collocational features in Turkish word sense disambiguation. In: Paper presented at the IEEE 16th international conference on intelligent, engineering systems (INES).
12. Oflazer K (1994) Two-level description of Turkish morphology. Literary Linguist Comput 9(2):137–148
13. Yuret D, Türe F (2006) Learning morphological disambiguation rules for Turkish. Paper presented at the proceedings of the main conference on human language technology conference of the North American chapter of the association of computational linguistics, In
14. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. ACM SIGKDD Explor Newsl 11(1):10–18

Part IV
Computer Vision

Eyes Detection Combined Feature Extraction and Mouth Information

Hui-Yu Huang and Yan-Ching Lin

Abstract In this paper, we propose an eye localization approach combining facial feature extraction and mouth information in color image. This approach includes feature extraction, face mask label, mouth location and eye region detection. In order to quickly obtain facial information, the coarse facial region is first localized by V-J filter, and then the compact facial region by our proposed filter can be decide and non-face region will be filtered out. Next, we label a face mask to modify skin region, extract the facial features and locate mouth information. Finally, the position of two eyes is detected based on the prior information. We present the experimental results on the various cases, including profile face, frontal face, and whether glasses or not, to demonstrate the effectiveness of our method.

1 Introduction

In recent years, there are many researches which focus on human biological attributes. Owing to human biological attributes, many applications have been developed on the different fields. As for applications, one of face detections can apply to recognition, monitor, and database. Eye detection can avoid or alarm a fatigue situation for driving a car. In addition, for mouth detection, it can provide a lip-read recognition to help the blind persons, and real-time smile detection applied in camera. Based on this interesting scheme, in this paper, we will focus on locate the correct eye position based on our proposed method.

Xu and Goto [1] presented a novel approach for face detection in still image based on the AdaBoost algorithm. Rahman et al. [2] developed face detection method in color images. Lin et al. [3] presented a driver fatigue detection based on mouth

H. -Y. Huang (✉) · Y. -C. Lin
Department of Computer Science and Information Engineering, National Formosa University, 632 Yunlin, Taiwan
e-mail: hyhuang@nfu.edu.tw

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_18,

geometrical features. Montazeri and Nezamabadi-pour [4] proposed a method which can automatically detect eye in extensive range of images with different conditions. Lu and Li [5] presented a method which is based on variance projections function. The eyes' feature can be extracted through variance projection at any illumination. Although these techniques provided some feasible solutions, however, the key challenge for eye detection is to correct the eye position in various face postures, specifically frontal posture or bold-framed glasses. To achieve this goal, we propose in this paper an eye detection approach based on mouth information to advance the correct. According to the message of mouth localization, the face area can be partitioned into left-side and right-side regions, and then the eye feature can be detected by means of our proposed technology.

The remainder of this paper is organized as follows: Section 2 presents the proposed method. Experimental results and discussions are presented in Sect. 3. We conclude our work in Sect. 4.

2 Proposed Method

This approach consists of feature extraction, face mask detection, mouth localization and eye detection. Details of procedures are mentioned in the following.

2.1 Extracting the Facial Area

Face extraction is very important step, which can provide the facial features to exactly detect the mouth and eyes locations. And the main procedures have two steps: face detection and filtering out non-face. For face detection, here, we take Viola and Jones (V–J) method [6] to work this field that possesses a high efficiency and accuracy to locate the face region in an image. In addition, although the face detection by V–J detector has a higher performance, the threshold may seriously affect the located result. Hence, we define a ratio of skin region to decide which the correct face region is, so that it can improve the accuracy of face location. In other words, the non-face region after V–J detector can be further filtered out [7].

2.2 Labeling Face Mask

Generally, the appearance of face posture has frontal face, profile face, or others. For localization field for facial features, such as mouth and eyes, it existed higher wrong on profile face image because its technique mainly depends on the percentage of skin color and non-skin color. In order to solve this case, we propose a procedure to reduce this mistake. This procedure employs skin segmentation and morphologic processing

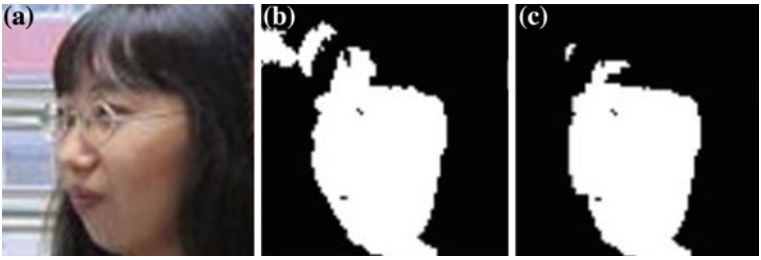

Fig. 1 Results of skin-color detection. **a** Original image. **b** Tseng's [8] method. **c** Our proposed skin condition

to find the compact skin region (called face mask) and to decrease non-skin effect only considering the white and yellow skinned people.

1. **Skin-Color Detection**

In order to segment face region and non-face region, we use the skin-color characteristic to achieve this process. We adopt the *YCbCr* color space to obtain the skin-color region by pixel-based processing. The *YCbCr* color space is defined as Eq. (1). According to our experiments and Ref. [7], the better skin constraint for face images is expressed as Eq. (2).

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & -0.587 & 0.114 \\ -0.168 & -0.331 & 0.5 \\ 0.5 & -0.418 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}, \tag{1}$$

$$\text{Skin} = \begin{cases} 1, \text{ if } \begin{cases} 60 \le Y \le 250, \\ 80 \le Cb \le 125, \\ 135 \le Cr \le 170, \end{cases} \\ 0, \text{ otherwise}, \end{cases} \tag{2}$$

where Y, Cb, and Cr denote Luma, blue, and red chrominance components, respectively. R, G, and B present the red, green, and blue color pixel values in *RGB* color space. Figure 1 present the results of the different skin-color segmentation methods compared with Tseng's [8] color condition and our proposed condition.

2. **Design of Ratio Mask**

In order to clearly present the skin-color region in a face image, we use a ratio filter [7] to achieve this purpose. The filter aims to emphasize skin pixels and to decrease the non-skin pixel. The ratio filter is to compute number of skin pixels and non-skin pixels on 5×5 mask, and then to compare those of skin pixels. Figure 2 shows the result of ratio mask.

Fig. 2 **a** Original image. **b** Result without using ratio filter. **c** Result using the ratio filter

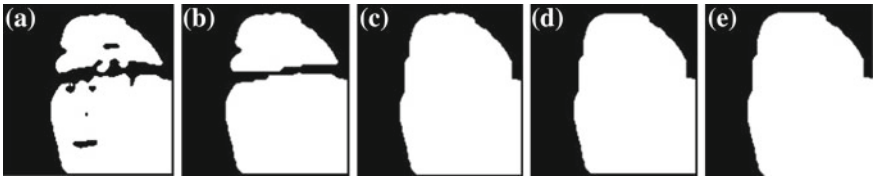

Fig. 3 The refined face mask. **a** The binary image by ratio filter. **b** The result obtained by computing "And" operation with vertical result. **c** Vertical result of (**b**). **d** (**c**) by step (2). **e** Refined face mask.

3. Face Mask

Based on Ref. [7], in this paper, we further modify their method to obtain a refined face mask. The modified procedures which worked by the horizontal result and the "AND" operation with vertical result are described as follows. Figure 3 shows a sequence of results about the refined face mask.

1. The pixel value of "AND" image is set 255 between the first point from top to down.
2. It is to find the first point which pixel is 0 for horizontal direction, and then to set this region as white pixels. Here, the first point is a black pixel searched from median to left and from median to right in face mask. Then, this pixel is set 255 between two these first points.
3. It is to take a dilation using 3×3 mask for white pixels to obtain a refined face mask.

2.3 Mouth Localization

The mouth localization is obtained after performing Canny edge detector and projection processing. The procedures have four parts. First step is to recover and extract the original image information in face mask. Second step is to detect the edge for this facial image by Canny detector. Third step is to estimate and define the mouth range on horizontal and vertical directions. Finally, the mouth position is located in

the maximum projection computing the projection of horizontal edge and vertical edge of this range [7].

2.4 Eye Region Detection Using Facial Features

Based on the mouth information, we can locate the eye position more precisely by our proposed method. Details of procedures are described in the following subsections.

1. **Recovering the Face Region and Features**

In order to increase the computational speed, we use the red color component in *RGB* color space to efficiently segment the face image into the non-skin area and skin area, and then to represent a binary image. The segmentation condition is defined as

$$\text{Color} = \begin{cases} 1, \text{ if } R > 95. \\ 0, \text{ otherwise,} \end{cases} \tag{3}$$

where R is red component in the *RGB* color space.

Based on this process, the facial features (black color) within this face mask can be recovered, and these features can take advantage of the next procedure to find eye features quickly. The recovering result is shown in Fig. 4.

2. **Labeling and Segmenting Eye Area**

Based on the previous results, here, the eye area will be specified by means of the horizontal detection. First, we will segment the horizontal region from the original face image. According to the spatial geometrics relationship of facial features and our observation, the eye, in general, is positioned the upper at one half of face region. Hence, we define a rational eye horizontal range expressed as

$$\begin{cases} H_1 = H_o \times 0.3 \\ H_2 = H_o \times 0.47 \\ W = W_o \times 0.15 \end{cases}, \tag{4}$$

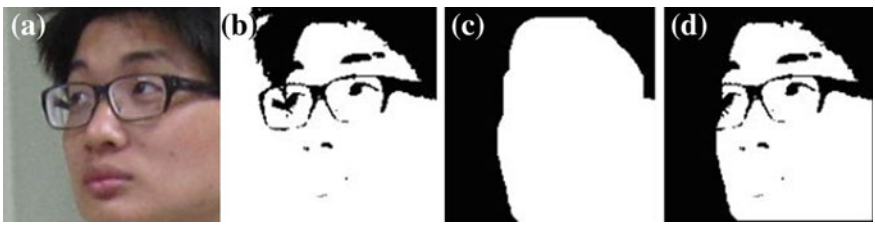

Fig. 4 **a** Original image. **b** Binary image. **c** Face mask (skin region). **d** Recovering result

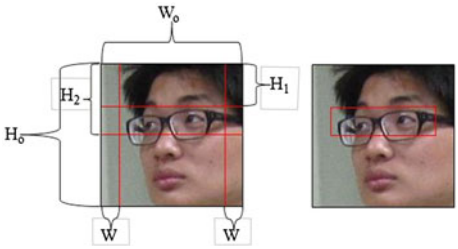

Fig. 5 The diagram of eye region notation

where H_1, H_2, and W denote the first position and the last position on image height (H_o) and the width range for eye region on image width (W_o), respectively. Figures 5 and 6 note the specified region. Next, based on Eq. (4) and face mask, we can easily extract the key information of this specified eye block. According to the spatial geometric relationships that the mouth is located at the medium position between eyes, we can easily divide this eye region to left eye and right eye. Afterward, we will individually take the relative parts to do the projection processing. Figure 7 shows the regions of left eye and right eye.

3. Localizing Eye Position Using Projection

After segmenting the eye region, we will compute the projections of the horizontal and vertical to locate the eye position. In order to increase the correct accuracy of eye location, we design a strategy to filter the projection data.

For horizontal projection, owing to eyebrows and glasses frame which seriously affect the accuracy, hence, we assign the appropriate condition expressed as Eq. (5) to enhance the accuracy. Afterward we will truncate two horizontal positions from the top in this specified eye region to reduce the eyebrows factor and to obtain the correct position of eyes.

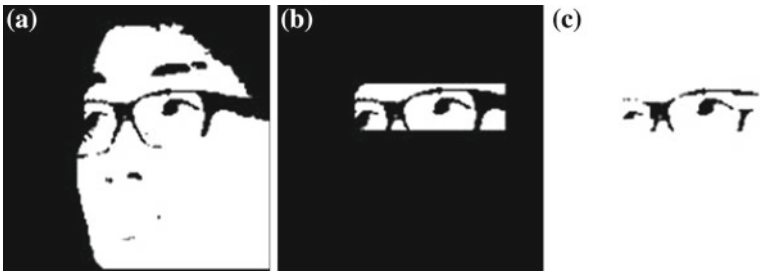

Fig. 6 Eye region for horizontal direction. **a** Recovered image, **b** and **c** specified eye region

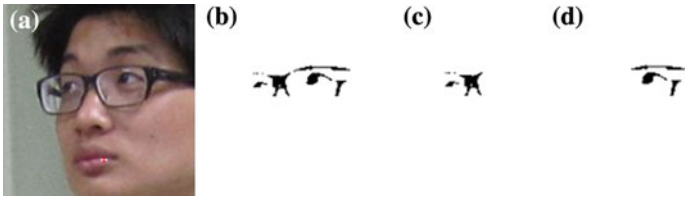

Fig. 7 Eye region of right side and left side. **a** Mouth location noted by red color. **b** Eye region. **c** Left-side eye region. **d** Right-side eye region

$$\begin{cases} HD_1[i] = |H[i-1] - H[i]| \\ HD_2[i] = |H[i] - H[i+1]| \end{cases}, \\ \begin{cases} HH[i] = H[i], \text{ if } HD_1[i] < 5 \text{or} HD_2[i] < 5, \\ HH[i] = 0, \text{ otherwise}, \end{cases} \tag{5}$$

where i is the position, $H[i]$ is the horizontal projection of the current position, $HD_1[i]$ and $HD_2[i]$ denote the difference between the forward position, next one, and the current one, respectively. $HH[i]$ is the filtered projection for horizontal projection.

For vertical projection, owing to our data contained glasses and no glasses, the features of some of glasses styles may affect the accuracy for eye location. In order to overcome this problem, we firstly specify the reasonable condition to solve this problem and reduce the mistake location, this condition is expressed as

$$\begin{cases} V_2[i] = V[i], \text{ if } Rad[i] < 0.45, \\ V_2[i] = 0, \quad \text{otherwise}. \end{cases} \tag{6}$$

$$\begin{cases} VD_1[i] = |V[i-1] - V[i]| \\ VD_2[i] = |V[i] - V[i+1]| \end{cases}, \\ \begin{cases} VV[i] = V_2[i], \text{ if } VD_1[i] < 5 \text{or} VD_2[i] < 5, \\ VV[i] = 0, \quad \text{otherwise}, \end{cases} \tag{7}$$

where i is the position, $V[i]$ is the projection of the current position, $Rad[i]$ is the height of the eye region, $VD_1[i]$ and $VD_2[i]$ depict the difference between the forward position next one, and the current one, respectively. $VV[i]$ is the filtered projection for vertical projection. Based on the projection processing, we can compute the maximum value of horizontal and vertical projections to detect the eye position more precisely.

Table 1 Evaluation of eye detection

Data set	Samples	Accuracy ratio		Fail ratio
		At least one eye (%)	Two eyes (%)	Two eyes (%)
Glasses	121	89	72	10
No-glasses	139	99	94	0.7
Total	260	95	84	5

Fig. 8 Facial feature localization. **a** and **b** With glasses, **c** no-glasses

3 Experiments and Discussions

Here, we use 260 face images with various cases, such as frontal, profile, glasses and no-glasses, to verify our proposed method. The empirical environment is operated in a 2.80 GHz Intel® Core(TM) i5-2300 CPU with 4 GB RAM PC and MS Virtual studio C# language. On the whole, the executed time for face images after V–J detector is spent about 0.75 s.

Table 1 shows the result of eye detection. In spite of the wrong cases existed in this approach when persons take the glasses, it still has 89 % accuracy at least one eye detection. On average, it can achieve 95 % accuracy at least one eye and 84 % accuracy at two eyes. Figure 8 shows some of location results for front and profile poses whether grasses or not.

However, there are existed some wrong case in this current method. It is because the width of grasses frame may cause the wrong ratio, or face mask labeling error, etc. Figure 9 shows some of wrong cases.

Fig. 9 Wrong cases

4 Conclusion Remarks

We have presented an efficient eye detection method based on feature extraction and the mouth information. Experimental results verify that this approach can obtain a good accuracy under various face postures, which exhibits the robustness of our proposed eye localization method. In the future, progress in current algorithm about face posture estimation will be studied.

Acknowledgments This work was supported in part by the National Science Council of Republic of China under Grant No. NSC100-2628-E-150-003-MY2.

References

1. Xu Y, Goto S (2011) Proposed optimization for AdaBoost-based face detection. In: 3rd International Conference on Digital Image Processing, pp 800907–1-5
2. Rahman H, Jhumur F, Yusuf SU, Das T, Ahmad M (2012) An efficient face detection in color images using eye mouth triangular approach. In: IEEE Conference of Informatics, Electronics and Vision. pp 530–535
3. Li L, Chen Y, Xin L (2010) Driver fatigue detection based on mouth information. In: 8th International World Congress on Intelligence Control and Automation, pp 6058–6062
4. Montazeri M, Nezamabadi-pour H (2011) Automatic extraction of eye field from a gray intensity image using intensity filtering and hybrid projection function. In: IEEE Conference of Communications, Computing and Control Application. pp 1–5
5. Lu Y, Li C (2010) Recognition of driver eyes' states based on variance projections function. In: IEEE Conference of 3rd International Congress on Image and, Signal Processing. pp 1919–1922
6. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: IEEE Computer Vision and Pattern Recognition. pp 511–518
7. Huang HY, Lin YC (2012) An approach for mouth localization using face feature extraction and projection technique. In: IEEE Conference of International Computer Symposium. pp 247–257
8. Tseng YC (2005) DSP based real-time human face recognition system. National Sum Yat-Sen University, Taiwan, Master Thesis

Depth from Moving Apertures

Mahmut Salih Sayar and Yusuf Sinan Akgül

Abstract Two new focus measure operators for Shape From Focus to estimate the three-dimensional shape of a surface are proposed in this paper. The images are formed by a camera using moving apertures. The well-focused image pixels are identified by frequency analysis or matching with the all-focused image of the scene. Frequency analysis involves the usage of classical focus measure operators and summing up for each aperture to find the focus quality to be maximized. The lesser method uses the match-points and error ratios of any matching algorithm used within an all-focused image region and all same-focus-different-aperture images to find the total displacement. The inverse of total displacement can be used as a focus quality measure. The experiments on real images show that the introduced ideas work effectively and efficiently.

1 Introduction

In recent years, the research area of retrieving the 3D structure of a scene has gained much attention in Computer Vision. Shape From Focus [3] (SFF) and Shape From Defocus [7] (SFD) are two of the popular approaches in this area. SFF is a method to find the best-focused image in the image-set for each pixel. SFD is a method that can estimate a blurriness value for each image pixel, defining a focus measure operator. SFF uses any focus measure operator for each image and selects the image that gives the maximum response. Depth of the object can be estimated by using

M. S. Sayar (✉) · Y. S. Akgül
Computer Vision Lab, Gebze Institute of Technology, 69121 Gebze, Kocaeli, Turkey
e-mail: salelltd@yahoo.com

Y. S. Akgül
e-mail: akgul@bilmuh.gyte.edu.tr

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_19,

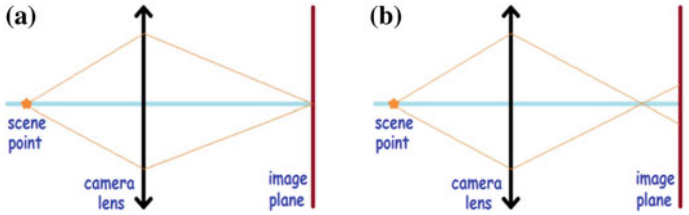

Fig. 1 Camera model for classic image formation. **a** Well-focused on the object. **b** Bad-focused on the object

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v}, \tag{1}$$

where f is the focal length of the lens, u is the distance between the camera lens and the image plane, v is the depth of the object.

Despite the number of algorithms in the literature that use SFF, the problem of finding a focus measure that can find a ground-truth depth map for all scene conditions is still a major challenge in computer vision. A classical method called sum of modified laplacians (SML) that finds depth map by using pixel frequencies was proposed by Nayar and Nakagawa [3]. Aydin and Akgul [1] proposed a new method called Adaptive Focus Measure (AdaFocus) that uses all-focused image information around the pixel as a cue to improve the results for traditional SFF methods.

Most of the methods in literature use full aperture [1–7], and/or rely on special scene conditions [5]. In this paper, we propose a new method that finds the depth map of the scene by the use of moving apertures. Unlike [1], our method uses all-focused image information to measure the displacements of each point in all images received using different apertures. However, our method can also make use of frequency analysis, since focus blurring effects occur on every bad-focused image part regardless of apertures. Thus, it is also possible to use [1] or [3] in conjunction with our method.

The rest of the paper is organized as follows. Section 2 gives background of the focus measure operators in literature. Section 3 describes the methodology that we have defined, and the focus measure operators that we have developed. We describe the experiments performed in Sect. 4. Finally, we provide concluding remarks in Sect. 5.

2 Previous Work

In classical image formation process that uses full aperture, the light rays pass through the lens of camera before they reach the sensors and form the image. A sharp image region is formed around well-focused objects, and blurry image region is formed around bad-focused objects; but the size or the center of the object's image does not change.

The objective of a focus measure operator is to evaluate the blurriness of all pixels on images retrieved under different focus settings. SML[3] is a classical focus measure operator that can measure the blurriness of an image region by means of frequency analysis. The focus quality value of a pixel at (x_0, y_0) is found using

$$\begin{aligned} ML\left(I\left(x,y\right)\right) =& \mid -I(x+s,y)+2I(x,y)-I(x-s,y)\mid \\ &+\mid -I(x,y+s)+2I(x,y)-I(x,y-s)\mid, \end{aligned} \tag{2}$$

$$SML\left(x_0,y_0\right)=\sum_{(x,y)\in\Omega_{x_0,y_0}} ML\left(I\left(x,y\right)\right) \text{ for } ML\left(I\left(x,y\right)\right)\geq T, \tag{3}$$

where s is the step-size, T is a threshold value, and $\Omega(x_0, y_0)$ is a window around the pixel at (x_0, y_0).

Since well-focused regions usually have higher frequencies, and bad-focused regions usually have lower frequencies; it is possible to estimate the depth map of a scene using SML [3]. The best focus setting for each pixel is selected as the focus setting that gives the maximum(or minimum) focus measure response, and the depth of the pixel can be found using Eq. 1.

However, SML [3] is proven to be weak around depth discontinuities (due to edge-bleeding [6]), and textureless regions. Nair and Stewart [2] tried to eliminate the edge-bleeding problem using an all-focused image of the scene. Aydin and Akgul [1] developed a method called Adaptive Focus Measure (AFM) that uses all-focused image information to address the edge-bleeding problem. The all-focused image of a scene is obtained by using a lens with an aperture as small as possible. According to [1], the all-focused image information can be mixed with any focus measure operator using

$$AFM\left(x_{0,}y_0\right)=\sum_{(x,y)\in\Omega_{x_0,y_0}} \omega_{x_0,y_0}\left(x,y\right)FM\left(x,y\right), \tag{4}$$

$$\omega_{x_0,y_0}\left(x,y\right)=e^{-\left(\frac{\Delta d}{\gamma_1}+\frac{\Delta I_f}{\gamma_2}\right)}, \tag{5}$$

where FM(x, y) is the focus measure value for the pixel at (x, y), $\omega_{x0,y0}$ (x, y) is the weight of the pixel (x, y) inside the region $\Omega_{x0,y0}$, Δd is the distance between (x, y) and (x_0, y_0), and ΔIf is the effective color difference between the all-focus pixels at (x, y) and (x_0, y_0).

The key difference of the proposed methods in this paper and most of the state-of-the-art approaches in literature is the effective usage of moving-apertures. While the classic approaches rely on only one image of the scene for each focus setting, our robust methods rely on many different-apertured images of the same scene. Moreover; like [1], our flexible methods can make use of all-focus image information and other focus measure operators. Unlike [1, 3] and [5]; our method is also capable of making use of matching methods.

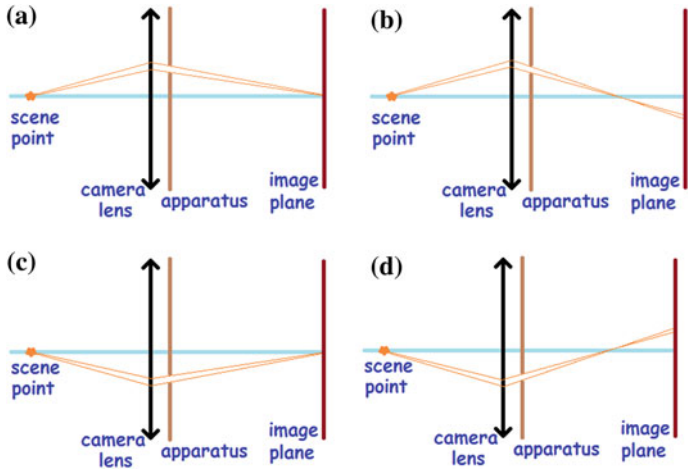

Fig. 2 Camera model for moving-aperture image formation. **a**, **c** Well-focused on the object. All light-rays meet at a certain point regardless of aperture position. **b**, **d** Bad-focused on the object. Aperture position determines the image position

3 Depth from Moving Apertures

Moving apertures can be implemented by using an apparatus in front of camera having differently-positioned holes. As it can be observed in Fig. 2, the concept of moving apertures causes the light to form different-apertured images in the same focus setting which may have different center points. It is possible to model a focus measure operator based on the displacements of the center points. Moreover, the frequency of image pixels can still be used as another clue since focal blurring is not avoided. In this way, we avoid all kinds of restrictions on scene conditions; the flexibility of our approach is also proven by the ability of our method to operate with SFF, SML [3], AdaFocus [1] and matching algorithms.

3.1 Aperture Integrals Focus Measure

The position or blurriness of a scene point in an image changes with the aperture position or size. Any classical method such as SML [3] or AdaFocus [1] can be used to measure the blurriness of each point in each image that were taken using different apertures on the same focus setting. The Aperture Integrals Focus Measure (AIFM) operator is:

$$AIFM\,(x,\,y) = \sum_{i=1}^{N_d} FM\,(x,\,y)\,[i], \tag{6}$$

where FM(x, y)[i] is the result of a classical focus measure operator for image pixel at (x,y) for aperture i and N_d is the number of apertures used.

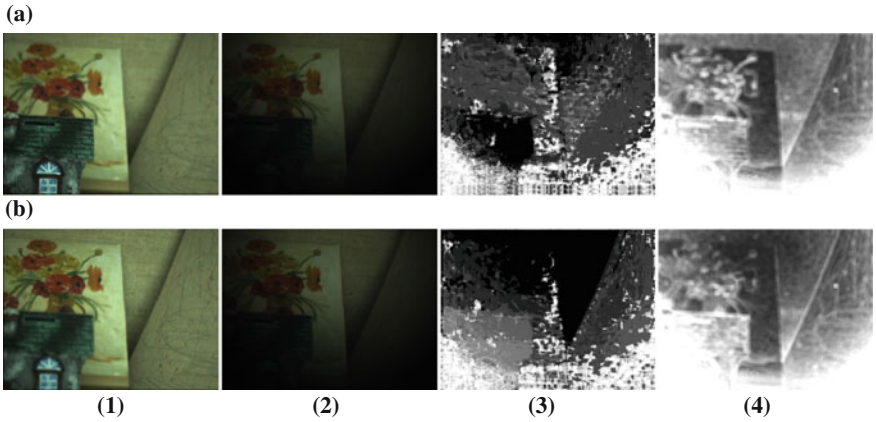

Fig. 3 Template matching results for: **a** Focus setting 6 using aperture 6. **b** Focus setting 29 using aperture 6. (*1*) w/full aperture. (*2*) w/aperture 6. The region around the center of aperture responds better in terms of frequency i.e. it has less blur effect. (*3*) Displacements of pixels. (Normalized) (*4*) Error results of NCC. (Normalized) All-focus image can be seen in Fig. 5**a1**

However, frequency of the pixels is also inversely proportional to the distance of the pixel to the center of each aperture (Fig. 3). Fortunately, mean value of the frequencies for all apertures can be used to address this problem. Since the number of apertures is equal for all images, summing up is enough to determine a correct focus measure value.

3.2 Matching Focus Measure

Template Matching can be used to find how much a pixel has moved from where it was in the all-focused image. For each focus setting; a small all-focused image window around the pixel is matched within a larger window in the images for all different aperture settings. The sum of displacement values for valid matches is used as a focus measure. Note that the template matching method is required to match correctly under different lighting conditions and/or different aperture sizes and/or focus blur effect. In this paper, we used Normalized Cross Correlation (NCC) [4] as our template matching algorithm.

Some image regions that have very low intensity values cannot be matched properly using the NCC algorithm, thus we define zero-threshold to reject these regions. Zero-threshold is used to avoid errors on very dark regions, so it is usually a very small intensity value. If the scene does not contain very dark regions, zero-threshold is not necessary.

Template matching algorithms may find bad matches due to noise, blur, bad lighting, scaling, rotation, etc. The failure of matching can be determined by using a

matching error-threshold. Matching error threshold avoids wrong matches and thus, it avoids wrong additions to the focus measure value. However, a very strict error threshold also causes wrong results due to the lack of displacement information. Therefore, it is hard to estimate a good matching error threshold value for all experiments. On the other hand, it is also possible to try different error-thresholds for each image.

Since the matching method may not be able to match the all-focused image region with many different-apertured images, it becomes necessary to define aperture-threshold that rejects the current focus setting if the number of accepted matches is lower than the threshold value. Otherwise, the small number of apertures also makes a small focus measure value. Taking average solves the issue a bit, but very small number of apertures may not make a good average either. Moreover, the real reason for the inability to match is that the regions are not similar. (In our case, probably too much focal-blurring, scaling, or movement.) In fact, it is this behavior which should remind us that this focus should not be the "best" focus for the current pixel. Aperture-threshold is usually selected as a small integer such as 3, 4 or 5.

The Matching Focus Measure (MFM) is

$$MFM\,(x,\,y) = \begin{cases} \infty, & \text{if} \quad c\,(x,\,y) < T_a, \\ TFM\,(x,\,y)\,, & \text{otherwise}, \end{cases} \tag{7}$$

where

$$N_c(n,\,x,\,y) = \begin{cases} 1, \text{ if} & E_n(x,\,y) < T_e \wedge I_f(x',\,y') > T_Z, \\ 0, \text{ otherwise}, \end{cases} \tag{8}$$

$$c\,(x,\,y) = \sum_{i=1}^{N_d} N_c(i,\,x,\,y), \tag{9}$$

$$TFM\,(x,\,y) = \sum_{i=1}^{N_d} N_c\,(i,\,x,\,y)\,d\,(I_i\,(x,\,y))\,, \tag{10}$$

where $d(I_i(x, y))$ is the distance from the image pixel $I_i(x, y)$ to its matched pixel in all-focused image $I_f(x', y')$, $E_i(x, y)$ is the resulting absolute error value for matching between all-focused image (I_f) and the image retrieved using aperture $i(I_i)$, N_d is the number of apertures used, T_e is the matching error-threshold, T_z is the zero-threshold, and T_a is the aperture-threshold.

4 Experiments

A mechanical setup involving two step motors and a control board was used, as well as a Basler camera, and a projector to get the ground-truth information using Scharstein-Szeliski [5] method. The apparatus had 12 holes, and we had a total of 78 different positions by moving it. Some positions were ignored due to the lack of scene's visibility. We used an intensity threshold of 10 to determine the visibility

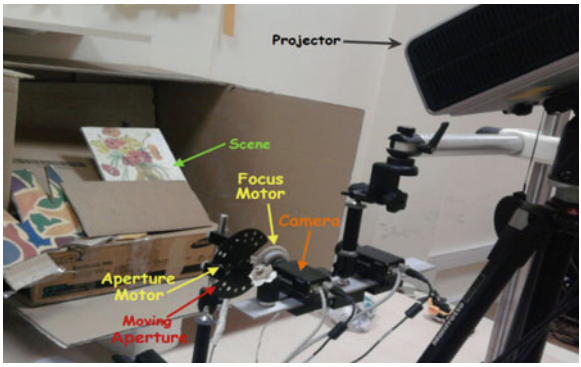

Fig. 4 Experimental setup

Table 1 Average errors for experiments

Method	Avg. error (Abs) ratio (Exp1) (%)	Avg. error (Exp1)	Avg. error ratio (Exp2)(%)	Avg. error (Abs) (Exp2)
SML with apertures(step = 1)	3.51	1.05	2.84	0.85
SML with apertures(step = 2)	3.93	1.18	3.24	0.97
SML with apertures(step = 3)	4.28	1.28	3.42	1.03
AdaFocus with apertures(step = 1)	3.64	1.09	4.24	1.27
AdaFocus with apertures(step = 2)	4.08	1.22	5.00	1.5
AdaFocus with apertures(step = 3)	4.48	1.34	5.34	1.6
SML without apertures(step = 1)	3.78	1.13	2.60	0.78
SML without apertures(step = 2)	3.59	1.08	2.59	0.78
SML without apertures(step = 3)	3.49	1.05	2.75	0.82
AdaFocus without apertures(step = 1)	3.68	1.1	3.90	1.17
AdaFocus without apertures(step = 2)	3.54	1.06	3.49	1.05
AdaFocus without apertures(step = 3)	3.47	1.04	3.59	1.08
Template Matching	6.73	2.02	8.48	2.54

of a certain pixel. 54 aperture positions for 1st experiment, 55 for 2nd experiment were selected by using a threshold of 10 % to visibility score. In order to reduce noise and get more reliable images, temporal average of 3 images was used. The image resolution of our camera was 640×480, the number of focus steps used in our experiments was 30. 8 bits binary coding, exposure times of 50, 100 and 150 ms were used for [5] with bitwise threshold of 5 and codewise threshold of 2. For SML [3] and AdaFocus [1], a step size of 1,2 and 3 were used seperately, with a constant window size of 25 and focus measure threshold of 5. For AdaFocus [1]; 5 was used as γ_1, and 12 was used as γ_2. For template matching, an all-focus window size of 15, and aperture window size of 29 were used, with a zero-threshold of 3, aperture-threshold of 3 and an error threshold of %0.5. The mean value of all results for all 3 step-sizes were used to map the ground-truth result of [5] into our system and modified by

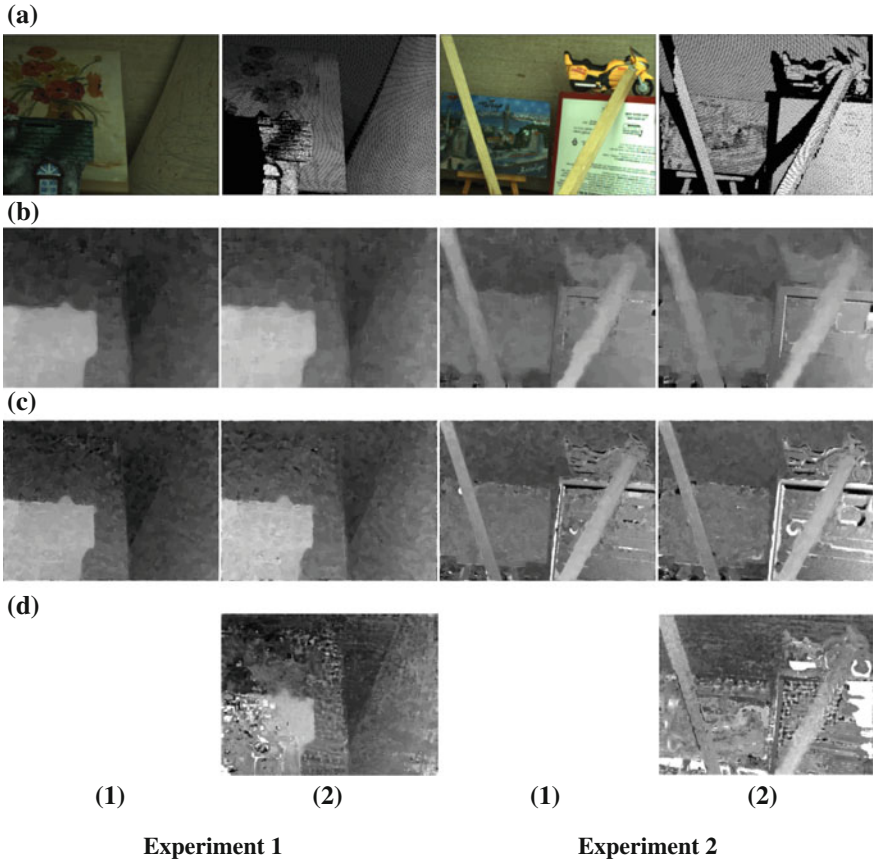

Fig. 5 **a1** All-focus image. **a2** Ground truth depth map. **b** SML. **c** AdaFocus. **d** Template Matching. (*1*) w/full aperture. (*2*) w/moving apertures

hand. Resulting depth maps for two experiments are shown in Fig. 4 and their error ratios in Table 1.

5 Conclusions

In this paper, we have defined a new concept called moving apertures to acquire precise depth maps. The resulting depth maps and error rates prove that the concept of moving apertures can improve the results of any focus measure operator depending on the number of apertures used. Our method can work with methods that use frequency analysis or matching without any restrictions for the scene conditions.

However; improvements and changes, such as the selection and parameters of matching algorithm, are necessary for MFM. It is possible to improve the results of both AIFM and MFM by applying different frequency analysis methods, focus measure operators, or matching methods.

For future work, we plan to use our system for producing ground truth depth maps for SFF maps.

Acknowledgments This work was supported by TUBITAK Career Project 105E097.

References

1. Aydin T, Akgül YS (2008) A new adaptive focus measure for shape from focus. British machine vision conference, Leeds (September)
2. Nair HN, Stewart CV (1992) Robust focus ranging. Computer Vision and Pattern Recognition, pp 309–314
3. Nayar SK, Nakagawa Y (August 1994) Shape from focus. Pattern Anal Mach Intell 16(8):824–831
4. Pratt WK (1978) Digital image processing. Wiley, New York
5. Scharstein D, Szeliski R (2003) High-accuracy stereo depth maps using structured light. Computer Vision and Pattern Recognition, p. 195–202, Wisconsin
6. Schechner YY, Kiryati N (2000) Depth from defocus versus stereo: how different really are they? Int J Comput Vis 39(2):141–162
7. Subbaro M, Surya G (1994) Depth from defocus: a spatial domain approach. Int J Comput Vis 13(3):271–294

Score Level Fusion for Face-Iris Multimodal Biometric System

Maryam Eskandari and Önsen Toygar

Abstract A high performance face-iris multimodal biometric system based on score level fusion techniques is presented in this paper. The aim of multimodal biometric systems is to improve the recognition performance by fusing information from more than one physical and/or behavioral characteristics of a person. This paper focuses on combining the strengths of face and iris modalities by employing the optimization method particle swarm optimization (PSO) to select facial features and the well known matching score level fusion technique, Weighted-Sum Rule, in order to obtain better recognition accuracy. Prior to performing fusion of face and iris modalities, standard feature extraction methods on face and iris images are employed separately. The unimodal and multimodal systems are experimented on different subsets of FERET, BANCA, and UBIRIS databases. Evaluation of the overall results based on recognition performance and ROC analysis demonstrates that the proposed multimodal biometric system achieves improved results compared to unimodal and other multimodal systems.

1 Introduction

Multimodal biometric systems aim to recognize a person based on his/her physical and/or behavioral characteristics. Recently, biometric recognition systems that are unique and cannot be lost or forgotten are widely used instead of traditional methods for person identification in many applications. In this respect, face, iris, fingerprint,

M. Eskandari (✉) · Ö. Toygar
Department of Computer Engineering, Eastern Mediterranean University, Mersin 10 Northern Cyprus Gazimağusa, Turkey
e-mail: maryam.eskandari@emu.edu.tr

Ö. Toygar
e-mail: onsen.toygar@emu.edu.tr

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_20,

speech and other characteristics in a unimodal or multimodal style may be involved to identify human beings in a reliable and secure manner [1].

Performance of unimodal systems may be affected by factors such as lack of uniqueness, non-universality and noisy data [2] and to overcome the limitations of single biometric traits multimodality is a solution. The recognition performance of biometric systems is improved using multimodal biometric fusion technology by extracting information from multiple biometric traits [3]. In this study, face and iris biometric systems are used for fusion due to many similar characteristics and unprecedented interest compared to other biometric technologies [4]. In order to fuse information, four different fusion levels are considered generally: sensor level, feature level, matching score level and decision level [3, 5, 6]. Among all fusion levels, the most popular one is matching score fusion level because of the ease in accessing and combining the scores produced from different matchers [3].

Recently, many researchers study on multimodality in order to overcome the limitations of unimodal biometrics. In [7], Vatsa et al. proposed an intelligent 2υ-support vector machine based match score fusion algorithm to improve the recognition performance of face and iris by integrating the quality of images. Liau and Isa in [3] proposed a face-iris multimodal biometric system based on matching score level fusion using support vector machine (SVM) to improve the performance of face and iris recognition by selecting an optimal subset of features. The authors used discrete cosine transformation (DCT) for facial feature extraction and log-Gabor filter for iris pattern extraction. The article emphasizes the selection of optimal features using particle swarm optimization (PSO) algorithm and the use of SVM for classification. A support vector machine (SVM) based fusion rule is also proposed in [4] to combine two matching scores of face and iris. Lumini and Nanni in [8], applied a strategy to obtain an appropriate pattern representation by extracting the information using an over-complete global feature combination and finally the selection of the most useful features has been done by sequential forward floating selection (SFFS). A multimodal identification scheme based on RBF (radial basis function) neural network fusion rules has been proposed by Wang et al. in [9]. The proposed method uses transformation-based score fusion and classifier-based score fusion. They concatenate normalized face and iris matching scores in order to classify a person from his/her face and iris images. A more recent approach has been proposed in [10] to use local and global feature extractors on face-iris multimodal recognition. Local Binary Patterns method is used as facial feature extractor and subspace LDA as iris feature extractor. The authors used Tanh score normalization and Weighted Sum Rule fusion techniques. In [11], the authors proposed a face-iris verification fusion problem from error rate minimization point of view at score level fusion. Their work optimizes the target performance with respect to fusion classifier design.

In this study, we apply a local feature extraction method, namely local binary patterns (LBP), to extract facial features. For iris recognition, a publicly available library by Masek and Kovesi [12] is used to extract iris features. In order to remove redundant and irrelevant data of facial features, an optimization method namely PSO is applied to select the optimized feature sets of the original face features. The fusion of these two modalities, namely face and iris, is then performed using Weighted

Sum Rule on face and iris scores. The proposed face-iris multimodal system in this study is also compared with unimodal and other multimodal systems using receiver operator characteristics (ROC) analysis.

The main contribution of the paper is to remove redundant and irrelevant data before combining facial features with iris counterparts. The optimal facial features are obtained by applying PSO and LBP-based facial features and the fusion of face and iris features is performed.

The organization of the paper is as follows. Section 2 presents face and iris recognition briefly. Normalization technique and multibiometric fusion are detailed in Sect. 3. Section 4 discusses databases, experimental results and ROC analysis. Finally Sect. 5 concludes the paper.

2 Feature Extraction on Face and Iris Biometrics

Human face is one of the most attractive and active areas for biometric recognition. In this study, facial features are extracted with a local feature extraction method namely LBP [13]. The number of partitions used for implementing LBP is N = 81 and (8, 2) circular neighborhood is used in the algorithm. In this study, each face image is divided into 81 non-overlapping subregions and the extracted texture descriptors generated using the LBP histogram are then concatenated to a global description. This global description causes a high dimensional face feature vector. In order to solve the problem arisen by high dimensionality, Particle Swarm Optimization (PSO) is applied on LBP features to select an optimized subset of facial features to increase the recognition performance.

particle swarm optimization (PSO) technique was introduced by Kennedy and Eberhart in 1995 [3]. Generally, initialization of PSO is done using a population of random solutions called particles and a fitness function is used for evaluation. In this study, recognition rate is used as fitness function. Each particle is treated as a point in an n-dimensional feature space. PSO is able to memorize at each iteration the best previous positions and this ability leads to update each particle by two best values *pbest* and *gbest*. *Pbest* represents the best fitness value position. *Gbest* is the index of the best particle among all the particles in the population. The velocity for ith particle is represented as $V_i = (v_{i1}, v_{i2}, \ldots, v_{in})$. The particles are updated as:

$$v_{id} = w v_{id} + c1 * rand_1()(p_{pbest} - x_{id}) + c2 * rand_2()(p_{gbest} - x_{id}),$$
$$x_{id} = \begin{cases} 1 \text{ if} \frac{1}{1+e^{-v_{id}}} > rand_3() \\ 0 \text{ otherwise} \end{cases} \tag{2.1}$$

where w is the inertia weight, c1 and c2 are acceleration constants, rand_1, rand_2, rand_3 are random numbers .

In PSO implementation, the inertia weight is set to 1, and the acceleration constants c1 and c2 are both set as 2. The feature selection in this study is based on a bit string

of length M, where M is the number of face features extracted from LBP feature extraction method ('1' means that feature is selected and '0' means that it is not selected) [14].

In our multimodal biometric system, FERET [15] and BANCA [16] face databases are used to measure the performance of face recognition. Face image preprocessing, training, testing and matching are common processing steps used on face databases. In order to reduce illumination effects on the face images, histogram equalization (HE) and mean-and-variance normalization (MVN) [17] are applied in preprocessing stage, in order to detect face images in BANCA database, Torch3 vision [18], a powerful machine vision library, is employed. Finally the last step is using Manhattan distance measurement between train and test face feature vectors to compute the matching scores.

On the other hand, one of the most reliable and secure biometric recognition systems that remains stable over the human lifetime [1, 19] is iris recognition. Iris recognition leads to a higher accuracy rate compared to other biometric recognition systems [19] and it is due to the valuable iris pattern information and its invariability through a lifetime [9, 20, 21].

In this study, Masek's iris recognition system [12] is applied to extract iris features. A noisy iris image database, UBIRIS [22] in two different subsets, is used to measure the performance of our unimodal and multimodal biometric systems. In this respect, since UBIRIS iris images are noisy and Masek's iris recognition system is written for nonnoisy databases such as CASIA [23], a modification has been done in the algorithm in order to normalize the brightness level of UBIRIS images to obtain a better detection on iris images. Iris image preprocessing, training, testing and matching are several stages for iris recognition process.

In Masek's iris recognition system, an automatic segmentation system based on the Hough transform is considered in the localization step and then the extracted iris region is normalized into a fixed rectangular form (20×240), 1D Log-Gabor filters are employed to extract the phase information of iris and the Hamming distance is employed for matching [12].

3 Normalization and Multibiometric Fusion

Produced matching scores from face and iris images need to be normalized due to non-homogeneity. Normalization of matching scores transforms the different matchers into a common domain and range in order to avoid degradation in fusion accuracy [24]. In this study, the produced matching scores of face are based on Manhattan distance measurement and the iris matching scores are obtained using Hamming distance measurement, therefore it is needed to normalize the face and iris scores to be in the common domain and range.

This study considers the tanh technique to normalize the face and iris matching scores. This robust and efficient method was introduced by Hampel et al. [25] and works very well for noisy training scores.

Development of the multimodal biometric system of face and iris verifiers is the most significant step in this study after producing the matching scores and normalization. In order to develop our multimodal face-iris system, we applied Weighted-Sum Rule on normalized face and iris scores. Combination of the face and iris scores based on the Weighted-Sum Rule is used to fuse the normalized scores.

Combined matching scores of the individual matchers can be computed by Weighted Sum Rule. In Jain and Ross [26], weighted sum of scores from different modalities using user-specific weights have been proposed. Generally, weights are computed using equal error rate (EER), distribution of scores, quality of the individual biometrics or empirical schemes [14]. In this study, empirical weighting scheme is employed to calculate the weights due to its efficiency compared to others [26]. In order to decide on the weights, a small subset of FERET and CASIA database with 50 individuals and 4 samples from each database are used. This subset is only employed for weight selection.

The proposed algorithm of our face-iris multimodal biometric system is as:

- Face/Iris image preprocessing (Detection + Histogram Equalization (HE) + Mean-Variance Normalization (MVN)).
- Extraction of face/iris features using LBP and Masek's code.
- Face feature selection using PSO.
- Obtaining iris matching scores using Hamming distance measurement.
- Obtaining face matching scores using Manhattan distance measurement.
- Face/Iris scores normalization using *Tanh* normalization method.
- Face/Iris scores fusion using *Weighted Sum Rule.*

4 Databases, Experimental Results and ROC Analysis

Our multimodal biometric system is tested on a multimodal biometric database using FERET and BANCA face databases and noisy UBIRIS iris database in two different subsets. In the first subset of experiment, for this study, from FERET database, 4 different frontal face images for 170 different subjects are selected. Accordingly, we selected 170 noisy subjects from UBIRIS iris databases with 4 samples for each subject. Two samples are selected randomly to be used for training and the remaining two samples for testing the system. In the second subset, 40 different subjects of BANCA face database with 10 samples are used with 5 samples randomly selected for training and the rest for testing. From UBIRIS noisy iris database, 40 different subjects are selected with 8 samples in which 3 samples used for training and the remaining 5 samples to test the system.

We applied LBP local feature extractor with and without PSO on unimodal face images and Masek's iris recognition system on iris images. Table 1 illustrates the performance of the algorithms implemented using face and iris images in both subsets. For Local Binary Patterns method, the number of partitions used is $N = 81$ and (8, 2) is the parameter for circular neighborhood. As represented in Table 1, the

Table 1 Performance comparison of face and iris unimodal systems

	FERET-UBIRIS		BANCA-UBIRIS	
Face recognition	LBP Without PSO	LBP With PSO	LBP Without PSO	LBP With PSO
	88.82	90.89	89.50	91.50
Masek's iris recognition	89.41		95.84	

Table 2 Recognition performance of multimodal systems

	FERET-UBIRIS		BANCA-UBIRIS	
	Masek's iris recognition + Face-LBP			
	Without PSO	With PSO	Without PSO	With PSO
Weighted Sum Rule	97.35	98.23	98.00	99.00
Median Rule	87.05	87.94	90.50	89.50
Min Rule	74.70	75.88	75.50	78.00
Max Rule	82.94	84.70	83.50	88.00

best accuracy is obtained using the local feature extractor LBP with PSO for face recognition system.

The best accuracy as demonstrated in the Table 1 is 90.89 % in FERET-UBIRIS subset for face recognition using LBP local feature extractor along with PSO feature selection. In BANCA-UBIRIS subset, the best accuracy is also obtained using LBP with PSO feature selection method as 91.50 %. However for iris recognition the performances of 89.41 and 91.50 % are achieved using Masek's iris recognition system in the first and second subsets respectively.

As shown in Table 2, fusion of face and iris scores leads to a higher recognition accuracy compared to the unimodal biometric systems. The fusion of face and iris matching scores is achieved using Weighted Sum Rule, Median Rule, Min Rule and Max Rule. As shown in this table, the best result for fusing the face and iris scores is obtained using Weighted Sum Rule, LBP with PSO and Masek's iris recognition system as 98.23 and 99 % in both subsets respectively.

Face-iris multimodal system is also compared with unimodal systems using ROC analysis. False Acceptance Rate (FAR) and False Rejection Rate (FRR) are used as a function of decision threshold which controls the tradeoff between these two error rates. The probability of FAR versus the probability of FRR is plotted for different values of decision threshold. The Equal Error Rate (EER) of each system, given on top of the curves in Figs. 1 and 2 is obtained from the point on ROC curve where the value of FAR is equal to the value of FRR. The unimodal systems for the first subset achieve the performance of 9 % EER for face and 10.5 % EER for iris on FERET and UBIRIS dataset. The proposed multimodal face and iris method achieves a performance of 2.5 % EER. The improvement of the multimodal system over the unimodal methods in FERET-UBIRIS subset is clearly demonstrated on

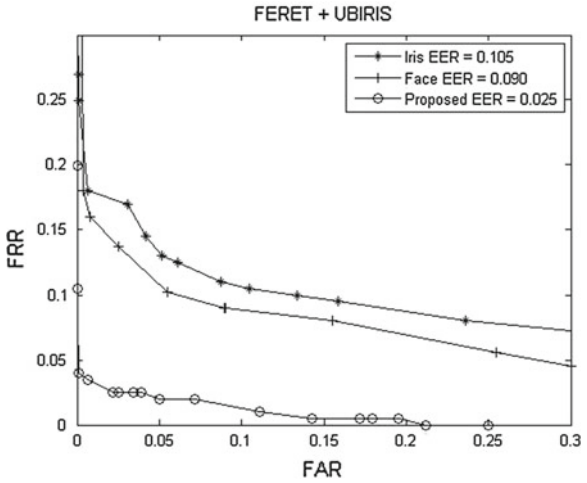

Fig. 1 ROC curves of unimodal and multimodal methods on FERET and UBIRIS Datasets

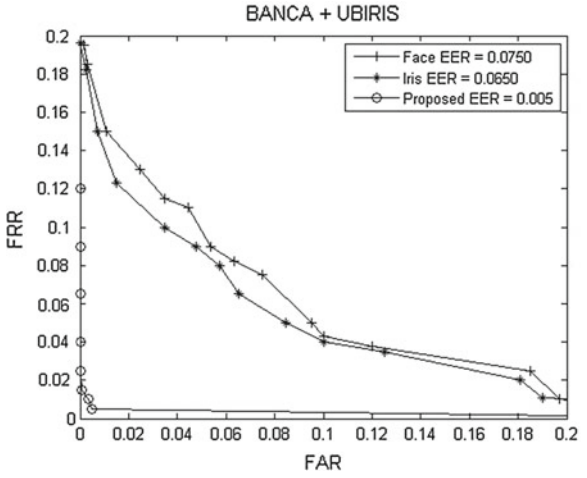

Fig. 2 ROC curves of unimodal and multimodal methods on BANCA and UBIRIS Datasets

ROC curves in Fig. 1. As demonstrated in Fig. 2, the unimodal systems achieve the performance of 7.5 % EER for face and 6.5 % EER for iris on BANCA and UBIRIS dataset, where the multimodal face and iris method achieves a performance of 0.5 % EER.

ROC curves of multimodal systems using other fusion methods such as Min Rule, Max Rule and Median Rule are demonstrated in Figs. 3 and 4 on FERET + UBIRIS and BANCA+ UBIRIS datasets, respectively. The results also show that the

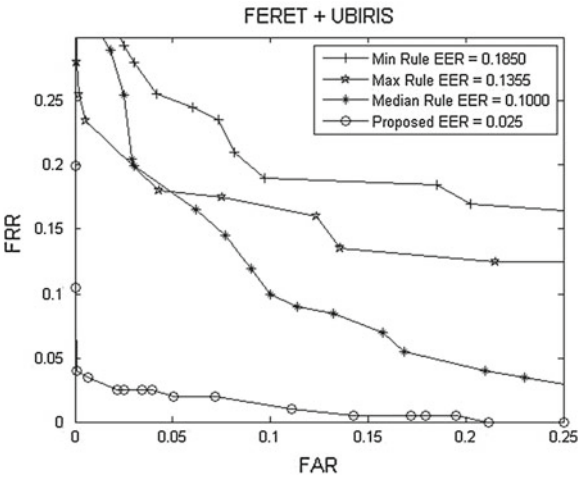

Fig. 3 ROC curves of multimodal methods on FERET and UBIRIS Datasets

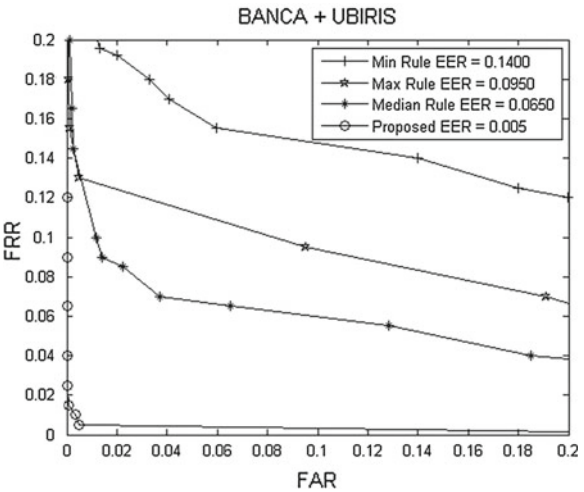

Fig. 4 ROC curves of multimodal methods on BANCA and UBIRIS Datasets

proposed method with Weighted Sum Rule combination method is better than the other multimodal methods on both datasets.

In general, the proposed multimodal system for face-iris recognition using Weighted Sum Rule and LBP with PSO achieves the best results compared to the other multimodal and unimodal systems used in this study.

5 Conclusion

This study presents the fusion of face and iris biometrics using Tanh normalization technique, Particle Swarm Optimization (PSO) feature selection method and Weighted-Sum Rule at matching score fusion level. A local feature extractor, namely LBP is applied to extract facial features and then PSO is used to select an optimized subset of features. The fusion of the scores is conducted by using tanh normalization of face and iris scores and Weighted-Sum Rule fusion method. The experiments performed on FERET+UBIRIS and BANCA+UBIRIS datasets demonstrated that the fusion of face and iris with feature selection and Weighted-Sum Rule achieves improved recognition accuracy compared to unimodal and other multimodal biometrics systems. Equal Error Rates on ROC curves demonstrate that the best recognition performance of the multimodal face-iris system is obtained using tanh score normalization and Weighted-Sum Rule fusion method whenever PSO is applied to select the optimized features.

References

1. Poursaberi A, Araabi BN (2007) Iris recognition for partially occluded images: methodology and sensitivity analysis. EURASIP J Adv Signal Process doi:10.1155/2007/36751
2. Jain AK, Ross A (2004) Multibiometric systems. Commun ACM 47(1):34–40
3. Liau HF, Isa D (2011) Feature selection for support vector machine-based face-iris multimodal biometric system. Exp Syst Appl 38(9):11105–11111
4. Wang F, Han J (2009) Multimodal biometric authentication based on score level fusion using support vector machine. Opto-Electron Rev. 17(1):59–64
5. Nandakumar K (2005) Integration of multiple cues in biometric systems. Michigan State University
6. Ross A, Nandakumar K, Jain AK (2006) Handbook of multibiometrics. Springer, Berlin
7. Vatsa M, Singh R, Noore A (2007) Integrating image quality in 2v-SVM biometric match score fusion. Int J Neural Syst 17(5):343–351
8. Lumini A, Nanni L (2008) Over-complete feature generation and feature selection for biometry. Exp Syst Appl 35(4):2049–2055
9. Wang Y, Tan T, Wang Y, Zhang D (2003) Combining face and iris biometric for identity verification. Proceedings of 4th international conference on audio- and video-based biometric person authentication (AVBPA), vol 1. pp 805–813
10. Eskandari M, Toygar Ö (2012) Fusion of face and iris biometrics using local and global feature extraction methods. Signal Image Video Process doi:10.1007/s11760-012-0411-4
11. Toh KA, Kim J, Lee S (2008) Biometric scores fusion based on total error rate minimization. Pattern Recogn 41(3):1066–1082
12. Masek L, Kovesi P (2003) MATLAB source code for a biometric identification system based on Iris patterns. University of Western Australia, The School of Computer Science and Software Engineering
13. Ahonen T, Hahid A, Pietikainen M (2006) Face description with local binary patterns: application to face recognition. IEEE Trans Pattern Anal Mach Intell 28(12):2037–2041
14. Raghavendra R, Dorizzi B, Rao A, Kumar GH (2011) Designing efficient fusion schemes for multimodal biometric system using face and palmprint. Pattern Recogn 44(2011):1076–1088
15. Philips PJ, Wechsler H, Huang J, Rauss P (1998) The FERET database and evaluation procedure for face recognition algorithms. Image and Vision Computing 16(5):295–306

16. Bailly-BE, Bengio S, Bimbot F, Hamouz M, Kittler J, Mariethoz J, Matas J, Messer K, Popovici V, Poree F, Ruiz B, Thiran JP (2003) The BANCA database and evaluation protocol. Proceedings of 4th international conference audio and video-based biometric person authentication AVBPA2003, (LNCS), vol 2688. Germany, pp 625–638.
17. Pujol P, Macho D, Nadeu C (2006) On real-time mean-and- variance normalization of speech recognition features. IEEE International conference, acoustics, speech and signal processing
18. Torch3vision, http://torch3vision.idiap.ch
19. Adam M, Rossant F, Amiel F, Mikovicova B, Ea T (2008) Reliable eyelid localization for iris recognition, ACIVS2008. LNCS 5259:1062–1070
20. Daugman JG (1993) High confidence visual recognition of persons by a test of statistical independence. IEEE Trans Circuits Syst Video Technol 14(1):21–30
21. Ma L, Tan T, Wang Y, Zhang D (1997) Personal identification based on Iris texture analysis. IEEE Trans Pattern Anal Machine Intell 25(12):1519–1533
22. UBIRIS iris database, http://iris.di.ubi.pt
23. CASIA-IrisV3, http://www.cbsr.ia.ac.cn/IrisDatabase.htm
24. Jain A, Nandakumar K, Ross A (2005) Score normalization in multimodal biometric systems. Pattern Recogn 38(12):2270–2285
25. Hampel FR, Rousseeuw PJ, Ronchetti EM, Stahel WA (1986) Robust statistics: the approach based on influence functions. Wiley, New York
26. Jain K, Ross A, (2002) Learning user-specific parameters in a multibiometric system. In: Proceedings of international conference on image process, New York, pp 57–60

Feature Selection for Enhanced 3D Facial Expression Recognition Based on Varying Feature Point Distances

Kamil Yurtkan, Hamit Soyel and Hasan Demirel

Abstract Face is the most dynamic part of the human body which comprises information about the feelings of people with facial expressions. In this paper, we propose a novel feature selection procedure applied to 3-Dimensional (3D) geometrical facial feature points selected from MPEG-4 Facial Definition Parameters (FDPs) in order to achieve robust classification performance. Distances between 3D feature point pairs are used to describe a facial expression. Support Vector Machine (SVM) is employed as the classifier. The system is tested on 3D facial expression database BU-3DFE and shows significant improvements with the proposed feature selection algorithm.

1 Introduction

Face and facial expressions have attracted reasonable interest of the researchers with the current developments in computer graphics and image processing. With the accurate face and facial expression analysis, it is possible to implement systems based on Human–Computer Interaction (HCI) as facial expressions involve most of the information about the feelings of a human. Therefore, many researchers are now

K. Yurtkan (✉)
Computer Engineering Department, Cyprus International University, Mersin 10, Lefkosa, Turkey
e-mail: kyurtkan@ciu.edu.tr

H. Soyel
School of Electronic Engineering and Computer Science, Queen Mary University of London, LondonE1 4NS, UK
e-mail: hsoyel@eecs.qmul.ac.uk

H. Demirel
Electrical and Electronic Engineering Department, Eastern editerranean University, Mersin 10, Gazimağusa, Turkey
e-mail: hasan.demirel@emu.edu.tr

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_21,

studying face and facial expression analysis for the next generation robust recognition systems.

Earlier studies of facial expressions based on Charles Darwin's general principles of facial expressions in 19th century [2]. Darwin's studies categorized human facial expressions into seven classes which are anxiety, anger, joy, surprise, disgust, sulkiness and shyness with many sub classes in each. Later on, Ekman's studies in 1970s classified human facial expressions in seven basic categories which are anger, disgust, fear, happiness, sadness, surprise and neutral citeEkman78, Ekman76. Ekman also studied facial movements and mimics that make an expression and developed Facial Action Coding System (FACS) with his colleagues.

In order to standardize facial information and movements, the Moving Pictures Experts Group (MPEG) defined Facial Animation (FA) specifications in the MPEG-4 standard. MPEG-4 became an international standard in 1999 [1].

Currently, robust facial expression recognition is still a challenge and there is still room for further improvements in recognition performances. Some of the important studies are the studies of Wang et al. [10] in which LDA based classifier is employed achieving 83.6 % overall recognition rate and studies of Soyel et al. [8] who proposed NSGA-II based feature selection algorithm resulting in 88.3 % overall recognition rate using 3D feature distances. 2D appearance feature based Gabor-wavelet (GW) approach is proposed by Lyons et al. and performed at 80 % average recognition rate [7].

In this paper, we propose a novel feature selection algorithm based on selected facial feature point distances extracted from 3D feature pairs. The algorithm employs variance as a metric of informativeness to select the most discriminative 3D distances. Support Vector Machine (SVM) is employed as the classifier. The performance of the feature selection algorithm is tested on 3D facial expression database BU-3DFE [11]. In our study, we follow the P. Ekman's pioneering studies about the classification of facial expressions in seven basic classes. The proposed feature selection procedure is applied to improve the classification of six basic expression classes listed above, excluding the neutral face.

The organization of the paper is as follows. In Sect. 2, the SVM classifier system is described. Sect. 3 gives details of the proposed novel feature selection procedure together with facial representation employed. We report our performance results on BU-3DFE database in Sect. 4.

2 Support Vector Machine Classifier System

Considering geometrical facial feature points representing a face, facial expression recognition can be defined as a vector classification problem. In this way, we define each face as a row vector of 3D geometrical feature point distances. Our approach is based on selection of the most informative feature point distances for expression classification. Classifier implementation is based on Support Vector Machine

(SVM) classifiers. We improve our previous study which is based on 3D facial feature positions [12].

There are 15 2-class SVM classifiers in the overall system. Each SVM classifier employs a linear kernel function (dot product) that maps the training data into kernel space. Penalty coefficient C used for each classifier is 1. 2-class classifiers are organized in the same way as [12]. Classifiers include all the combinations of six expression classes which are anger-disgust, anger-fear, anger-happiness, anger-sadness, anger-surprise, disgust-fear, disgust-happiness, disgust-sadness, disgust-surprise, fear-happiness, fear-sadness, fear-surprise, happiness-sadness, happiness-surprise and sadness-surprise.

Each 2-class classifier is trained independently. The details about the training and testing phases of the classifiers are given in Sect. 4 by reporting the performances on BU-3DFE database [11]. The 15 two-class classifiers classify an input face vector as one of the six basic expressions by applying majority voting. The expression with the maximum number of classifications is selected as the recognized expression. The SVM classifier system is illustrated in Fig. 1 in which feature point distances are shown for visualization purpose only.

The input vector consists of 3D facial feature point distances describing a face in one of the six basic expressions. All facial feature point distances are arranged as a row vector to represent a face. Consider a row vector definition of a facial feature point as shown in Eq. 1, V_i being a 3D vector definition for a facial feature point.

$$V_i = \left[V_{i_x} \; V_{i_y} \; V_{i_z} \right] \tag{1}$$

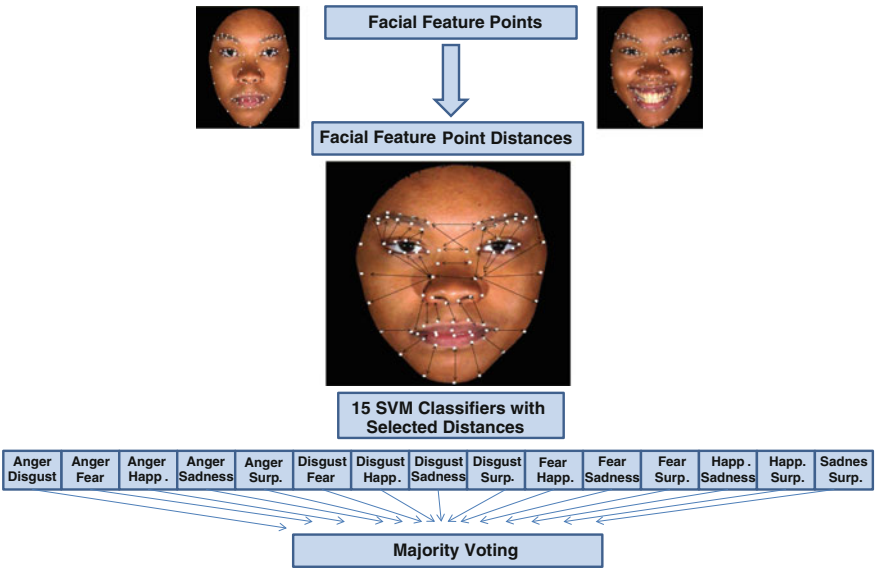

Fig. 1 SVM Classifier system used for facial expression recognition

Each facial feature position is then taken as a reference, and all the combinations of two of 83 points, C(83,2). 3D distances between each facial feature point and the others are calculated according to Eq. 2. In total, each face vector contains data of 83 feature points available in BU-3DFE [11] database. Thus, a face vector definition is derived from all combinations of two of 3D feature point distances, C(83,2) which brings 3403 distance values. As a result of these calculations, a face is described with a vector *FV* of selected distances from overall 3403 3D distance values, DV_{ij}'s calculated as in Eq. 2. Face vectors are defined as *FV*s shown in Eq. 3 where t, k stands for last feature point distance value indices selected to represent a face after applying feature selection procedure. After combining all row vectors for all faces, a matrix *FM* is formed as shown in Eq. 4. The number of face vectors is represented by n. The matrix *FM* is divided into two subsets, in order to obtain training and test sets, described in Sect. 4.

$$DV_{ij} = \sqrt{(V_{i_x} - V_{j_x})^2 + (V_{i_y} - V_{j_y})^2 + (V_{iz} - V_{jz})^2} \tag{2}$$

where i = 1, 2, . . ., 82 and j = i + 1, i + 2, . . ., 83 and $DV_{ij} \in$ C(83,2)

$$FV_m = \left[DV_{0,0}\ DV_{0,1}\ DV_{0,2} \ldots\ldots DV_{t,k} \right] \tag{3}$$

$$FM = \begin{bmatrix} FV_1 \\ FV_2 \\ \ldots\ldots \\ FV_n \end{bmatrix} \tag{4}$$

3 Feature Selection for Expression Classification

Facial representation is important to define the facial characteristics of the expressions to solve the expression classification problem. MPEG-4 defines Facial Definition Parameters (FDPs) and Facial Animation Parameters (FAPs) on a generic face model in its neutral state are used to represent facial expressions and animations [1]. Facial expressions can be modeled with deformations on the neutral face by using MPEG-4 FAPs [1]. In our study, we employ 3D geometrical facial feature point data to drive 3D distances representing a face.

Our initial experiments [12] show that using 3D feature point data defines facial expression deformations on the face well and achieves acceptable recognition rates as presented in Table 1. All 83 feature point positions with the classifier depicted in Fig. 1 are used to obtain results in Table 1. Therefore, we consider 3D feature point data as a base for feature selection procedure. It is seen from MPEG-4 FAP implementation that although most of the FDPs are affected among 6 basic expressions, some FDPs are weakly influenced from facial expression deformations; even some of them are not affected. Therefore, the selection of feature points which are

informative for facial expression deformations are very critical. The proposed feature selection procedure results in subset of facial feature point distances which are highly affected by expression deformations and include most of the information.

The proposed method is a variance based procedure that measures informativeness of 3D facial feature point distances in order to select the most discriminative features improving expression classification performance. Informativeness and uncertainty are measured with strong metrics of variability which are variance and entropy [3]. More variance on the distribution of a feature point distance means the distance is more varying, therefore, carries more information about the expression and hence it is more discriminative for that expression. On the other hand, low variance distance means a stable distance value which does not change enough to characterize an expression. In this paper, we propose a novel variance based feature selection procedure, which selects a set of the most discriminative 3D facial feature distances for six basic facial expressions. Face definitions are characterized with the selected high variance distance values specific for the six basic facial expressions.

Face vectors contain distance values for all combinations of 3D distances between feature points before feature selection. We measure the variances of each feature point distance combination in 3D for the neutral and the six basic expressions, and analyzed the overall variances of the feature points as shown in Eq. 5, where X is a 3D distance (consider the columns of *FM* matrix, each column represents one distance value) and μ is the mean value for the related distance value. Variance measurements completed on training samples from BU-3DFE database by measuring the variance of each distance among neutral and six basic facial expressions. These variance values are then sorted in descending order resulting in a decreasing function of distance values. Figure 2 shows the graph of sorted variance values of 3D feature point distances among neutral and 6 basic expressions. Training set includes 90 % of face vectors and test set includes 10 %. BU-3DFE database includes 100 samples with neutral and 6 expressions; therefore 630 vectors are reserved for training. We select training and test set distribution similar to the other methods we compare.

$$Var(x) = E[(X - \mu)^2] \tag{5}$$

The decreasing variance function implies that after sorting the variances in descending order, feature point distances show significant differences in variance for some breaking points. Thus, high variance distance values mean that expression can be captured when the face is deformed from neutral to any one of the six basic expressions by looking at these distances. Feature selection algorithm eliminates low variance feature point distances in face vectors in order to use informative distance values for the recognition. Using decreasing function of variance, breaking points are detected which are showing significant difference in variance above standard deviation. Then, brute force search is employed among these breaking points to find the combination maximizing the overall recognition rate. Eliminating low variance distances result in 3,172 high variance distance values selected from overall 3,403 distance values.

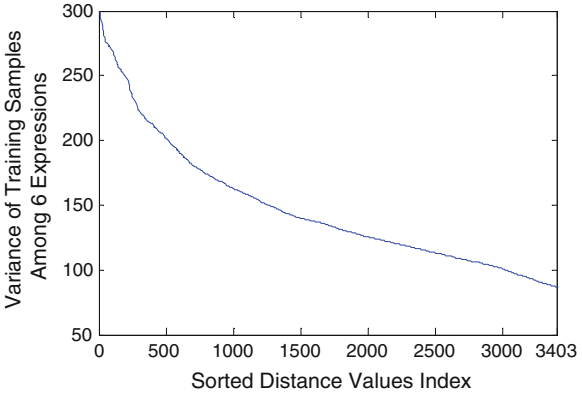

Fig. 2 Variance analysis of 3,403 feature point distances for training samples among neutral face and 6 basic expressions

4 Performance Analysis

The proposed system is tested on 3D facial expression database BU-3DFE [11]. Recognition rates are reported for six basic facial expressions. Database includes 100 samples with neutral and six basic expressions. Each expression has 4 different intensities, 1 being the lowest, and 4 being the highest intensity. In our recognition tests, we use level 4, the highest intensity expressions. *FM* is constructed as described in Sect. 2 including 100 samples with neutral and 6 basic expressions, in total, 700 row vectors describing 700 faces. 15 two-class SVM classifiers are trained separately with 90 % of the row vectors of FM matrix and 10 % of row vectors are used for testing. As described in Sect. 2, total face vectors are divided into training and test sets. Training vectors are used in the variance analysis for feature selection procedure. We classify expressions using 15 two-class SVM classifiers and applying majority voting among six expression classes.

The system is tested several times on BU-3DFE database. It is person independent which means that if a subject appears in the training set it never appears in the test set. In order to validate the test results, recognition tests are repeated 8 for random combinations of rows of *FM* matrix. Then, the average rates are reported.

Table 1 presents recognition rates of basic expressions using our SVM classifier system with 83 3D feature point positions [12]. It is seen from Table 1 that 3D feature point data are efficient in expression classification as our earlier studies justifies this fact [8, 12]. Yin et al. also reports the average recognition performance of feature point data in BU-3DFE database for six basic expressions as around 83 % [11]. In Table 2, improvements in the recognition rates after applying our proposed feature selection procedure can be observed. Overall recognition rate is improved significantly from 83 to 87.5 % while maintaining high rates for each expression.

Table 1 Recognition rates for proposed SVM classifier using 3,403 distances

Expression	Recognition rate (%)
Anger	93.3
Disgust	80.0
Fear	60.0
Happiness	93.3
Sadness	80.0
Surprise	86.7
Overall	83.33

Table 2 Recognition rates for proposed SVM classifier using 3,172 selected high variance feature point distances

Expression	Recognition rate (%)
Anger	85.00
Disgust	83.75
Fear	80.00
Happiness	95.00
Sadness	83.75
Surprise	97.50
Overall	87.50

The comparison of the proposed system is done with the current systems in the literature which are also tested on BU-3DFE database with similar experimental setup. The comparison is given in Table 3. From the aspect of overall recognition rate, the proposed system has one of the highest rates among other person independent methods we compare. We also see from Table 3 that other methods we compare [9, 10] reports high recognition rates for some of the expressions, while some other expressions are still in relatively low rates, in acceptable rates. For example, Tang et al. reports 74.2 % recognition rate for fear expression whereas surprise expression is recognized with 99.2 %. We also address this problem in our expression recognition studies that some expressions are recognized with very high rates according to others resulting in high overall recognition rate. The proposed method achieves high rates for each expression in this sense.

Comparing proposed method with [12], which is our previous study, we observe significant improvements on the recognition rates of fear and sadness expressions. In [12], a dedicated study also presented for the recognition of fear expression but improvements are reported using only fear test faces separately. Using 3D distances overcomes poor recognition rate of fear expression of our previous study as well. Also the average recognition rate is improved while maintaining high rates for all expressions. Experimental results justify the effectiveness of feature selection procedure applied to 3D feature point distances. Although variance is a simple metric to compute, it is very efficient in measuring informativeness of data for facial expression

Table 3 Performance comparison of proposed recognition system

	Recognition rate (%)				
Expression	Soyel et al. [8]	Wang et al. [10]	Tang and Huang [9]	Yurtkan and Demirel [12]	Proposed
Anger	85.9	80.0	86.7	**100.0**	85.0
Disgust	**87.4**	80.4	84.2	86.7	83.8
Fear	**85.3**	75.0	74.2	60.0	80.0
Happiness	93.5	95.0	**95.8**	93.3	95.0
Sadness	82.9	80.4	82.5	80.0	**83.8**
Surprise	94.7	90.8	**99.2**	93.3	97.5
Overall	88.3	83.6	87.1	85.6	87.5

recognition problem when using 3D facial geometry. Variance analysis detects the most dynamic feature distances during facial expression deformations of the face which carry information about expressions. Selection of these informative feature distances improves recognition performance significantly.

5 Conclusion

In this study, we propose a novel feature selection procedure applied to 3D feature point distances in order to improve facial expression recognition performance. We employ 3D facial feature point distances for facial expression representation. Performance results show that the proposed system achieves competitive and comparable recognition rates on BU-3DFE database. The system is open for further improvements considering expression specific feature selection procedures. The other strong metric of informativeness, which is entropy, is also under research for the next level improvements on 3D feature distance based expression classifier.

References

1. Abrantes G, Pereira F (1999) MPEG-4 facial animation technology: survey, implementation and Results. IEEE Trans Circuits Syst Video Technol 9(2):290–305
2. Darwin C, edited by Darwin F (1904) The expression of the emotions in man and animals, 2nd edn. J. Murray, London

3. Ebrahimi N, Soofi E. (1999), in Slottje DJ (ed), Measuring Informativeness by entropy and variance. N. Advances in Econometrics, Income Distribution, and Methodology of Science (Essays in Honor of Camilo Dagum), Springer, Berlin
4. Ekman P, Friesen W (1978) The facial action coding system: a techinique for the measurement of facial movement. Consulting Psychologists Press, San Francisco
5. Ekman P, Friesen W (1976) Pictures of facial affect. Consulting, Psychologist, Palo Alto, CA
6. Kamil Yurtkan, Hasan Demirel (2010), Facial expression synthesis from single frontal face image. 6th International Symposium on Electrical and Computer Systems, 25–26, European University of Lefke. Gemikonağı, TRNC
7. Lyons M, Budynek J, Akamatsu S (1999) Automatic classification of single facial images. IEEE Trans Patt Anal Mach Intell 21:1357–1362
8. Soyel H, Tekguc U, Demirel H (2011) Application of NSGA-II to feature selection for facial expression recognition. Comput Elect Eng, Elsevier 37(6):1232–1240
9. Tang H, Huang TS. (2008) 3D Facial expression recognition based on properties of line segments connecting facial feature points. IEEE International Conference on Automatic Face and Gesture Recognition 2008
10. Wang J, Yin L, Wei X, Sun Y (2006) 3D facial expression recognition based on primitive surface feature distribution. Comput Vision Patt Recog 2:1399–1406
11. Yin L, Wei X, Sun Y, Wang J, Rosato M (2006) A 3d facial expression database for facial behavior research. In Proceedings of International Conferance on FGR, pp 211–216, UK
12. Yurtkan K, Demirel H (2012) Person independent facial expression recognition using 3D facial feature positions. Computer and Information Sciences 3, Springer, vol 3, 27th International Symposium on Computer and, Information Sciences, pp 321–329

Part V
Data and Web Engineering

DAPNA: An Architectural Framework for Data Processing Networks

Hasan Sözer, Sander Nouta, Andreas Wombacher and Paolo Perona

Abstract A data processing network is as a set of (software) components connected through communication channels to apply a series of operations on data. Realization and maintenance of large-scale data processing networks necessitate an architectural approach that supports analysis, verification, implementation and reuse. However, existing tools and architectural styles fall short to support all these features. In this paper, we introduce an architectural style and framework for documenting and realizing data processing networks. Our framework employs reusable and composable data filters. These filters are annotated with their deployment information. The overall architecture is specified with an XML-based architecture description language. The specification is processed by a toolset for analysis and code generation. The framework has been utilized for defining and realizing an environmental monitoring application.

1 Introduction

Data collection, processing and interpretation are essential tasks for almost all scientists. Automating these tasks is important (and sometimes necessary) to save time and prevent errors. Unfortunately, not all scientists have the necessary computer science background to realize this automation. They have to use standard tools, create scripts for different data processing steps and manually integrate them. This approach is not always adequate. In some research domains, there are too many data processing

H. Sözer (✉)
Özyeğin University, İstanbul, Turkey
e-mail: hasan.sozer@ozyegin.edu.tr

S. Nouta · A. Wombacher
University of Twente, Enschede, The Netherlands

P. Perona
Institute of Environmental Engineering, EPFL, Lausanne, Switzerland

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_22,

tasks. The integration of these tasks is subject to complex inter-task dependencies and timing constraints. Moreover, the processing tasks can be distributed among multiple computers at remote locations. The CCES project RECORD [18] is an example for this case, where scientists need to monitor and analyze data regarding a river and its surroundings to calibrate their environmental models. Such cases demand the development of a data processing network (DPN), which we define as a set of (software) components connected through communication channels to apply a series of operations on data.

The realization of a DPN should be documented with an architecture description to facilitate communication, analysis and verification, and to support implementation and reuse. Otherwise, realizing and maintaining a large-scale DPN design can be cumbersome. DPNs can be documented with a set of existing architectural styles [11] such as the Pipe-and-Filter style [6] or its variants [16]. These styles can be sufficient for documentation and communication purposes. Analysis and verification can also be supported if the architecture is formally specified [2]. However, the lack of an architectural framework hinders the possibility for automated analysis, reuse and code generation.

In this paper, we introduce an architectural framework called DAPNA for documenting, analyzing and realizing DPNs. The framework employs composable data filters. These filters can be reused both within the same project and across different projects. The DPN architecture is described using a style that we have defined as a specialization of the Pipe-and-Filter style. In addition, this style also incorporates deployment information in terms of annotations on data filters that can be distributed on the Internet or on intranets. The overall DPN architecture is specified with an XML-based architecture description language (ADL). The specification is processed by a toolset for analysis and code generation. The framework has been utilized for defining and realizing an environmental monitoring application in the context of the CCES Project RECORD [18] case study.

The remainder of this paper is organized as follows. In Sect. 2, we describe DAPNA and its utilization for analysis and realization of a DPN. In Sect. 3, we discuss the actually applied case study for environmental monitoring. In Sect. 4, related previous studies are summarized. Finally, in Sect. 5 we discuss some future work issues and provide the conclusions.

2 The DAPNA Framework

The overall process for realizing a DPN with DAPNA is depicted in Fig. 1. The designer specifies the DPN using an XML-based ADL, for which we have devised an XML schema [13]. The DAPNA framework [13] processes such a description for analysis and code generation. First it analyzes the provided architecture description for checking its conformance to the architecture style and violation of any constraints. It also checks if the specified DPN is connected without missing any links. If there is an error detected, the analysis results are provided to the designer. If not,

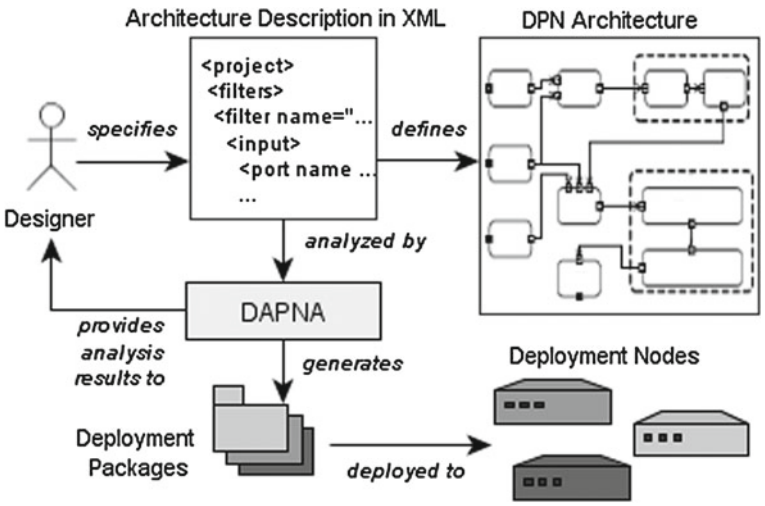

Fig. 1 The overall process for DPN realization

DAPNA generates a set of deployment packages. These packages should be copied and executed on the corresponding deployment nodes as specified in the architecture description.

We have performed a domain analysis [3] to define the basic concepts and their relations pertaining to a DPN. By investigating the literature [12] and example applications [18], we have defined a domain model [13]. Based on this domain model and the existing Pipe-and-Filter style [6], we have defined our ADL that defines the set of elements and relations taking part in a DPN architecture description. In addition to the basic concepts such as *pipes* and *filters*, we have introduced the concept of *binding* to support the (hierarchical) composition of filters. Furthermore, we have defined particular types of *data sinks* and *data sources* that are common in a DPN. DPN elements, relations and their properties can be found in [13].

DAPNA comprises a reusable library of communication primitives, data source and sink definitions, and data filters. This library can be further extended with user-defined, composable data filters. In addition to the filter elements, DAPNA provides several reusable abstractions for data sources, serilalization options, random data generators (for training purposes) and conditional data flows. An overview of the framework elements can be found in [13]. The framework hides many implementation details from the user. For instance, the utilization of communication mechanisms (i.e., TCP/IP sockets) is taken care of by the framework according to the topology specified in the documented DPN. Furthermore, the input ports, transformations and output ports are all equipped with standard recovery mechanisms [13]. For example, if a node within the DPN is temporarily unavailable, the other nodes retransmit messages within a predefined interval. When the number of (unsuccessful) attempts exceed a threashold, the node failure is reported to the user.

In the following, we first explain the specification of a DPN with DAPNA. Then, in the next section, we discuss the application of DAPNA on a case study. The following listing shows the overall structure of a DPN description specified with our XML-based ADL [13].

```
<project>
<filters>
<filter name="Sample Filter"
runat="127.0.0.1" buffer="10">
<input> ... </input>
<transformation> ... </transformation>
<output> ... </output>
</filter>
...
</filters>
</project>
```

Listing 1 The basic structure of a DPN description

The description consists of a root element called `<project>`, which contains a `<filters>` element. The `<filters>` element consists of a set of `<filter>` elements. The `<filter>` element has three attributes: `name`, `runat` and `buffer` (optional). The first attribute depicts the filter name and must be unique. The second attribute shows the IP address of the machine where the filter will be deployed on. The last attribute indicates the maximum number of data packages inside the internal buffer. A `<filter>` element has three sub elements: an `<input>` element, a `<transformation>` element and an `<output>` element.

The `<transformation>` element defines how the incoming data packets will be processed. Hereby, one of the predefined transformations (e.g., merge) can be used. Optionally, an external application can be utilized to process data packets. The following listing shows an example transformation definition that makes use of MATLAB [4].

```
<transformation type="external">
<command>matlab.exe</command>
<parameters>
<parameter>-nodesktop</parameter>
<parameter>-wait</parameter>
<parameter>-r</parameter>
<parameter>load('%f'); correlatePixels(); save -v6
    '%f'; exit;</parameter>
</parameters>
...
</transformation>
```

Listing 2 A transformation definition that makes use of an external application

Note that the `type` attribute is defined as `external`. The use of the external application is specified including a set of parameters within the `parameters` element.

The other two sub elements of a <filter> element are the <input> and <output> elements. These elements comprise (possibly) multiple <port> elements. An example <port> element is listed in the following.

```
<port name="output" type="gateway">
<conditions>
<condition name="weekdays">
<time type="lessequal">
<format>M</format><value>5</value>
</time>
</condition>
<condition name="weekend">
<not><reference name="weekdays"/></not>
</condition>
</conditions>
<pipes>
<pipe destination="Node1.input" condition="weekend"
    type="tcp">
<portnumber>1444</portnumber>
</pipe>
</pipes>
</port>
```

Listing 3 An OutputPort of type gateway

Each <port> element has two attributes: name and type. The name must be unique. The output can be stored in a file, presented on the screen, or it can be sent to another filter for further processing. In the example listing above, the filter type is defined as gateway, which means that the output should be passed to connected and active pipes that are defined in the <pipes> element as part of the <port> element. The <pipes> element consists of a set of <pipe> elements, each of which has three attributes: destination, condition (optional) and type. Pipes can connect filters residing at both local and remote hosts. The condition attribute contains the name of a condition that determines whether the pipe accepts data packages.

3 Case Study

The CCES project RECORD [18] is investigating the effects of restoring a river on flora, aquatic fauna, river morphology and ground water. One aspect of the project is the development of a model for river morphology changes. To evaluate the proposed models, the researchers installed two towers with cameras taking pictures on a regular basis at least once per day over three years. During floods, the rate of taking pictures has to be increased significantly. The pictures document the gravel bars in a restored river segment. From these pictures, the shape of the gravel bar must be inferred using image processing techniques. These shapes can then be compared with the shapes derived from the morphology models. The huge amount of pictures requires

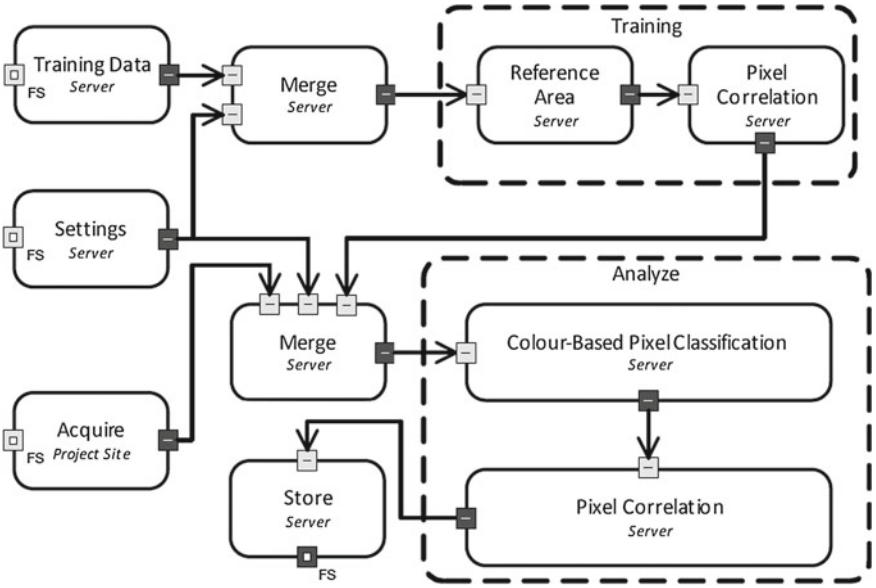

Fig. 2 A data processing view for the CCES project RECORD [18] case study

automation of the image processing. Since image processing is based on classification of image properties, the classification has to be trained especially after bigger changes in the morphology, thus, this again is a repetitive process. A detailed description of the hydrological aspects concerning image processing can be found in [15].

Figure 2 shows an example DPN architecture with a graphical notation [13]. This is a simplified example used within the CCES project RECORD [18] case study. Here, the DPN is based on a training part (upper part of Fig. 2) and an analysis part (lower part of Fig. 2). The *Training* step consists of two filters: the *Reference Area* filter determines the areas within the picture which are most discriminative for the classification of water pixels for differentiating water and non water pixels. In the *Pixel Correlation* filter, relationships between pixels are inferred, i.e., if a pixel is classified as water, then all its correlated pixels are water. The *Training* step uses as input an information consisting out of a collection of *Training Data* provided by a filter and a set of configuration *Settings* provided again by a filter. The two inputs are assembled in a *Merge* filter, where each training image is associated with the configuration settings, comparable to a database join operation.

The *Analyze* step uses as input the merged configuration *Setting* information, the output of the *Training* and the pictures *Acquired* from the towers. First, the *Colour-based Pixel Classification* filter uses the RGB value distributions derived in the training step to classify pixels as water or non-water. This result is then further processed by applying the derived *pixel correlations* from the training step. The outcome of the *Analyze* step is sent to the *Store* filter, which stores the incoming information in the file system without further processing. In this example, the data

processing is not distributed. The filters related to training and analysis are deployed on the same node, labelled as *Server*; however, the acquisition of data is performed at a remote site (i.e., *Project Site*).

Documenting a DPN in conformance to an architecture style makes the data processing steps, their relationships and the overall topology explicit. As such, it facilitates the communication of architectural design decisions. Furthermore, it supports domain-specific analysis, code generation and reuse.

We have evaluated the usability of the DAPNA framework in the context of the CCES project RECORD [18]. At the moment of writing this paper, a functional DPN has not been deployed yet in the field. However, we have designed a DPN and utilized the DAPNA framework to realize the corresponding system on a local testbed with real data flow. The system has worked on over a thousand images of the use case within a day. We have observed that the high level definition of the pipes makes it easy to chain the various transformations. As a drawback we noticed, the users were not completely comfortable with editing XML documents. Hence, we plan to realize a graphical editor in the Eclipse GMF framework. Another challenge is the definition and integration of locally installed tools and their integration with the DPN. It turned out that executing external scripts on different platforms and operating systems required different parameters in the DPN specification.

4 Related Work

A comparison of architectural styles for network-based software architectures can be found in [9]. In this work, after a classification and comparison of the existing styles, the *Representational State Transfer (REST) style* for distributed hypermedia systems is introduced. In comparison with ours, the REST style is more focused on the network properties instead of the data processing aspects. Depending on the interest of the stakeholders, both styles can be utilized as part of an architecture description.

Many algorithms have been developed for stream processing [12] and in particular for image processing [14, 17]. Our approach provides the means to document the utilization, composition and configuration of these solutions as part of the software architecture design.

Workflows can be created with a Workflow Management Systems like Kepler [10] or Taverna [19] using an intuitive drag-and-drop interface. A workflow is defined as the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules [7]. A DPN is in fact a workflow, but a workflow is not always a DPN. DAPNA is specialized for defining and realizing DPNs. In addition, DAPNA supports distributed deployment and processing by definition. Workflows may implicitly support some decentralization aspects, but there is usually a central system needed to execute the workflow.

5 Conclusions and Future Work

We have introduced DAPNA, an architectural framework for documenting, analyzing and realizing data processing networks (DPNs). DAPNA abstracts away many implementation details, while enabling designers to develop, configure and deploy DPNs. As such, our approach makes the development of DPNs less effort-consuming and less error-prone. In the future, we will investigate possibilities for performing dynamic analysis and runtime integration of DAPNA with analysis tools such as MATLAB [4].

Acknowledgments This work has been carried out as part of the CCES project RECORD [18].

References

1. van der Aalst WMP, Weijters AJMM (2004) Process mining: a research agenda. Comput Ind 53(3):231–244
2. Abowd G, Allen R, Garlan D (1995) Formalizing style to understand descriptions of software architecture. ACM Trans Softw Eng Methodol 4(4):319–364
3. Arrango G (1994) Domain analysis methods. In: Schafer, Prieto-Diaz R, Matsumoto M (eds) Software reusability. Ellis Horwood, Chichester, pp 17–49
4. Bishop RH (1996) Modern control systems analysis and design using MATLAB and SIMULINK. Addison Wesley, Boston
5. Bodensta L, Wombacher A, Wieringa R, Jaeger MC, Reichert M (2009) Monitoring service compositions in mode4sla - design of validation. In: Cordeiro J, Filipe J (eds) Proceedings of ICEIS (4), pp 114–121
6. Clements P, Bachmann F, Bass L, Garlan D, Ivers J, Little R, Nord R, Stafford J (2002) Documenting software architectures: views and beyond. Addison-Wesley, Boston
7. Coalition WM (1995) The workflow reference model. Document number TC00-1003, issue 1.1 (Jan 1995)
8. Dashofy E, van der Hoek A, Taylor R (2001) A highly-extensible, xml-based architecture description language. In: Proceedings of the working IEEE/IFIP conference on software architectures, Amsterdam, The Netherlands
9. Fielding R (2000) Architectural styles and the design of network-based software architecture. Ph.D. dissertation, Department of Information and Computer Science, University of California, Irvine
10. Kepler (2011) The kepler project. http://kepler-project.org. (Accessed on Aug 2011)
11. Monroe R, Kompanek A, Melton R, Garlan D (1997) Architectural styles, design patterns, and objects. IEEE Softw 14(1):43–52
12. Muthukrishnan S (2003) Data streams: algorithms and applications. In: Proceedings of the 14th annual ACM-SIAM symposium on discrete algorithms (Jan 2003)
13. Nouta C (2011) Data processing networks made easy. M. Sc. thesis, Department of Computer Science, University of Twente, Enschede, The Netherlands
14. Pal N, Pal S (1993) A review on image segmentation techniques. Pattern Recogn 26:1277–1294
15. Pasquale P, Perona P, Schneider P, Shrestha J, Wombacher A, Burlando P (2010) Modern comprehensive approach to monitor the morphodynamic evolution of restored river corridors. Hydrology Earth Syst Sci Discuss 7:8873–8912
16. Shaw M, Clements P (1997) A field guide to boxology: preliminary classification of architectural styles for software systems. In: Proceedings of the 21st international computer software and applications conference. Washington, DC, pp 6–13

17. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8):888–905
18. Swiss Experiment (2011) Record:home - swissexperiment. http://www.swiss-experiment.ch/index.php/Record:Home
19. Taverna (2011) Taverna - open source and domain independent workflow management system. http://www.taverna.org.uk. (Accessed on Aug 2011)

Crescent: A Byzantine Fault Tolerant Delivery Framework for Durable Composite Web Services

Islam Elgedawy

Abstract Composite web service delivery is a very complex process that involves many complex tasks such as capacity management, components discovery, provisioning, monitoring, composition and coordination, customers' SLAs management, moreover it requires cancellation and billing management. Indeed, managing all these tasks manually is a very cumbersome operation, nevertheless it is time consuming and prone to errors. To overcome such problems, this paper proposes Crescent; a Byzantine fault tolerant service delivery framework for durable composite web services. Crescent ensures the full automation of the composite web service delivery process. Furthermore, Crescent combines between quorum-based and practical Byzantine fault tolerance protocols as well as components adaptive parallel provisioning approaches to ensure reliable Byzantine fault tolerant service delivery. Crescent enables customers to have differentiated levels of service by allowing the composite web service to support different types of workflows. Experimental results showed that Crescent increases the reliability and throughput of composite web service delivery when compared with existing approaches.

1 Introduction

Durable composite web service is a web service realizing a given business process; by invoking other existing web services (known as components). Such durable composite web services are expected to have a long life-time, and expected to have a high volume of customers' requests. Service delivery system must ensure the fulfilment of the customers' Service Level Agreement (SLAs). However, this is not an easy task,

I. Elgedawy (✉)
Computer Engineering Department, Middle East Technical University, Northern Cyprus Campus, Mersin 10 , Guzelyurt, Turkey
e-mail: Elgedawy@metu.edu.tr

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_23,

as the delivery process for durable composite web services involves many complex tasks such as:

- **Byzantine Fault Tolerance (BFT)** Delivery The delivery system should ensure reliable delivery in spite of the Byzantine failures of the composite service components, as well as the delivery system modules themselves. This simply will maximize the service delivery determinism.
- **Automated Capacity Management** The delivery system should adopt both reactive and predictive strategies to be able to automatically manage the composite service capacity for different time scales, which is mandatory to ensure having enough capacity for handling incoming demand; including its unexpected spikes without degrading the service performance.
- **Automated Coordination and Execution** The delivery system should coordinate between different components executions to ensure results correctness for customers' requests. Also it should automatically create alternative composition plans according to customers' SLAs if original ones have failed.
- **Automated Task Recovery** The delivery system should be able to automatically recover from different tasks failures to ensure system liveness. This applies for the invoked components as well as for the delivery system modules.
- **Automated SLA Management** The delivery system should allow users to create, update and customize their SLAs and their corresponding workflows. Also the delivery system should monitor customers' SLAs during execution; to detect any signs of violations, and automatically handle such problems without customers involvement.
- **Automated Adaptive Composition** The delivery system should be able to dynamically create different composition plans for different types of workflows defined in the users' SLAs. Such composition plan should ensure BFT delivery for all required workflow tasks without violating customers' SLAs.
- **Automated Components Discovery** The delivery system should be able to automatically find and test new components so that they can be used during composition plans generation required for realizing workflows defined in customers' SLAs.
- **Automated Cancellation Management** The delivery system should be able to automatically manage users and components cancellation if cancellation option is supported, and of course it has to be reflected on composite web service billing.
- **Automated Billing Management** The delivery system should be able to automatically create customers bills based on components consumption as well as any SLAs policies and agreements.

Unfortunately, we could not find any exiting work that discusses a composite web service delivery system that fulfills all the above requirements. Most of existing work is focusing on Byzantine Fault tolerance for atomic web services such as [1–4] and discussing general management aspects for atomic web services such as performance monitoring as in [5]. However, there exist some work addressed the issue of composite web service fault tolerance such as [6–8]. Work in [6] is mainly focusing on coordination Byzantine fault tolerance but ignored components fault tolerance at all. On the other hand, work [7, 8] focused on components fault tolerance without

supporting Byzantine faults tolerance, and totally ignored any forms of coordination fault tolerance. As we can see there is no existing work that supports Byzantine (or even normal) fault tolerance for both components and coordinators, which we believe is a mandatory requirement for reliable composite web services delivery. Therefore, this paper identifies this gap and proposes Crescent; a Byzantine fault tolerance delivery framework for composite web services. Crescent ensures BFT for composite web service delivery, by using a light-weight replication-based BFT protocol (such as PBFT protocol [9]) to ensure the BFT for service delivery modules, and uses a speculative quorum-based BFT protocol to ensure components BFT (such as Zyzzyva [10]), adopting components parallel provisioning (i.e. invoking different multiple components at the same time to realize a given workflow task). Crescent manages the service capacity by combining between predictive and reactive component provisioning approaches to dynamically handle incoming demand spikes. Experimental results showed that Crescent increases the reliability and throughput of composite web service delivery when compared with existing approaches.

2 Separating Composite Web Service Logic from Its Execution

Different customers have different needs, so that the composite web service must embrace such customers diversity and heterogeneity and be able to support differentiated levels of service. Hence, a composite web service should not follow a single workflow to fulfill all customers' requests, instead it should adopt different workflows to suit different customers' SLAs. Hence, we require a composite web service to be defined as a collection of workflows such that chosen workflows are stored in a workflow pool that customers could choose from to select the suitable workflow. We formally define a composite web service as $\{\omega_1, \omega_2, ..., \omega_k\}$, where ω_i is a realizable workflow, k is the number of supported realizable workflows, a workflow ω_i is defined as by any workflow langauge, however, we require a task projection set $\triangle$ to be given as well, where $\triangle(\omega_i) = \{\tau_1, \tau_2, ..., \tau_m\}$, where τ_i is any realizable task appeared in ω_i. τ_i is realized by invoking a single component, or by invoking a set of parallel components $\{s_1, s_2, ..., s_p\}$, forming what is known as a *components cluster*. Based on the proposed composite web service model, we describe component provisioning for a given workflow using a *composition plan*. A composition plan could be serial or parallel. A serial composition plan uses only one component to realize a given workflow task. If a component fails during execution, a new composition plan has to be formed with a new component replacing the failing one. On the other hand, a parallel composition plan uses more than one component running in parallel to realize a workflow task (i.e. components cluster), which of course dramatically minimizes task failure error. For example, a workflow serially constituted from three tasks τ_1, τ_2, and τ_3, could be realized via serial composition plan as $\{\langle\tau_1, s_1\rangle, \langle\tau_2, s_4\rangle, \langle\tau_3, s_6\rangle\}$ or via a parallel composition plan as $\{\langle\tau_1, \{s_1, s_2, s_3\}\rangle, \langle\tau_2, \{s_4, s_5\}\rangle, \langle\tau_3, \{s_6, s_7, s_8\}\rangle\}$. In this paper, we will adopt only parallel composition plans as we need to support Byzantine fault tolerance for workflow tasks, such that each components cluster will

have a minimum number of $3f + 1$ components needed to guarantee BFT for f Byzantine faults. Also we require each component service to have a functional interface for normal invocation, and a management interface for monitoring and control operations. Adopting the proposed model, enables us to decouple composite web service logic (i.e. abstract workflows) from its realization (i.e. components). Hence, when a customer request is submitted to the composite web service, the suitable workflow and its realizing components are identified and executed on the fly in a context-based manner.

3 Crescent: A BFT Composite Web Service Delivery Framework

This section proposes Crescent a BFT service delivery framework for durable composite web services. It specifies the needed modules and their interactions. We followed the separation of concerns design principle to allocate a clear cohesive function for each module. The proposed framework is depicted in Fig. 1. We identify Crescent modules as follows:

- **SLA Manager**: SLA manager is responsible for communicating with customers to create the required SLAs. Such SLAs could be tailored from scratch, created as variants from existing SLA, created from customizable templates or simply reusing existing SLAs. The SLA manager must ensure that each SLA contains references to the required workflows and workflows priorities. We extended the

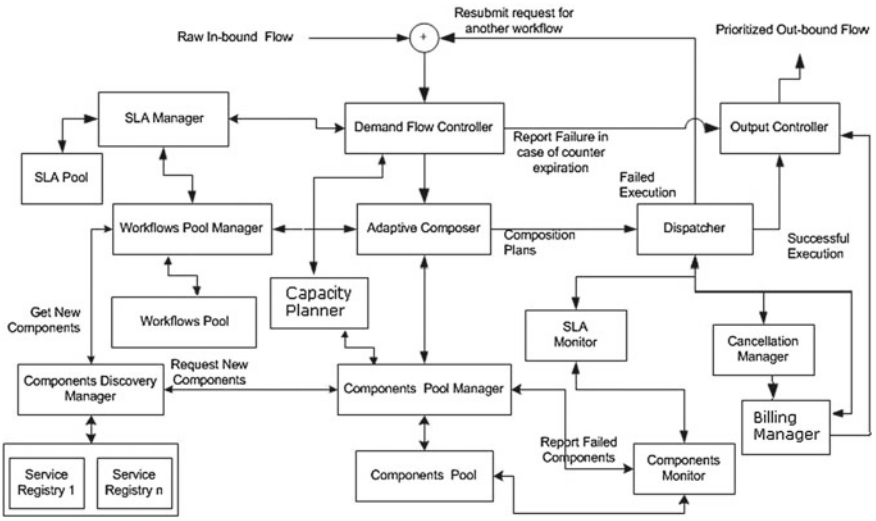

Fig. 1 Crescent: a composite web service delivery framework

Web Service Level Agreement (WSLA) language proposed by IBM to be able to describe the composite web services SLA in a machine-understandable format.

- **SLA Pool**: This pool contains all created SLA as well as their variants. This pool not only will be used for providing the different SLA options for customers, but also facilitates and simplifies request submission process, as customers could just refer to their SLA reference (i.e. unique reference using URI) in their requests without the need to submit their SLA file every time they need issue a request.
- **Demand Flow Controller (DFC)**: DFC collects customers' requests in addition to any resubmitted requests for failed workflows. It prioritizes all of these requests according to the customers' SLAs (adopting predefined business rules), then associates each request with its highest priority workflow, then forwards these prioritized requests to the adaptive composer to find suitable realizing components. DFC keeps a failure counter for every resubmitted request, when this counter reaches a predefined threshold, DFC associate a new workflow to the request, which is the next in priority order, and tries again. If all assigned workflows are tried and failed, DFC reports failure for such request.
- **Workflows Pool**: It is the pool containing all the supported workflows created by the service provider. This pool is managed by a workflow pool manager, that is contacted when workflows need to be added/deleted (automatically and/or manually).
- **Adaptive Composer (AC)**: AC processes requests in a prioritized FIFO manner. It generates the suitable composition plans for each request and submits it to the dispatcher as a BPEL script. AC adopts any suitable service composition approach to identify the required components (such as approach discussed in [11]). To find suitable components, it contacts the Component Pool Manager (CPM) to search the component pool. If no components are found, or the existing number of components is less than the required quorum, the CPM reports failure to the AC, and then AC switches to a less priority workflow, and tries again. Components selection could be defined according to a predefined assignment policy (e.g. round robin, least cost, most recently used, least recently used, random, etc). In case cancellation requests are issued, AC should retrieve the adopted composition plan from its history log, then forward it to the dispatcher to execute orders cancellation.
- **Components Pool**: It is the pool containing all information about the discovered components that will be used to generate composition plans. Components in the pool are indexed according to workflow tasks, and could be discovered and approved on the fly or off-line, as sometimes negotiations and contracting between service providers are still done manually. This pool is managed by a components pool manager, that is contacted when components need to be added/deleted (automatically and/or manually). Component pool manager can delete any components violating their SLA, and can generate discovery requests for new components by communicating with the components discovery manager.
- **Capacity Planner**: It is the module responsible for ensuring having enough components in the components pool to handle incoming demand. It gets actual demand statistics from the demand flow controller, and runs different forecast predication algorithms to predict required capacity, then contacts the components pool

manager to make sure enough components exists. If more components are required, the components pool manager contacts the components discovery manager to get more components.

- **Components Discovery Manager**: It is responsible for finding new components for realizing the workflows' tasks. It could have access to different service registries (internal and external). It could suggest different variations to the workflow if components could not be found for the original workflow. Such variations should be added to the workflows pool as new workflows. Techniques for finding such components could be found in [11].
- **Components Monitor**: It is responsible for keeping up-to-date information about components capacities, by invoking the suitable operations in the components management interfaces. Such information are provided to the component pool manager to update statistics of the components. Components monitor can do regular or on-demand checks, according to the defined business rules.
- **Dispatcher**: It is the orchestrator, coordinator, and initiator for the composition plan execution. It executes the specified parallel provisioning plan, and decides on the response based on the adopted BFT protocol. If execution failed, it resubmit the request to the demand flow controller asking for alternative composition plan. In case cancellation requests are issued, it communicates with cancellation and billing managers to make sure defined cancellation and compensation actions are performed.
- **Cancellation Manager**: It is responsible for contacting components for cancelling accomplished requests, and makes sure all cancellation business rules are met, and payment balance is recollected. Cancellation transactions could be a long lived transaction, as for example it could take days to refund credit card payments, hence we created a separate module for managing cancellations.
- **Billing Manager**: It is responsible for generating users bills and make sure payments have been made according to their SLAs. Also it is responsible for pay component providers according to their SLAs, it communicates with the Dispatcher and Cancellation manager to get up-to-date information about consumed components. It communicates with users via the output controller.
- **Output Controller**: It is responsible for reporting the composite service responses to customers. It could prioritize responses (e.g. late responses could go out first), and could reformat responses according to customers' SLAs.

Crescent manages the service capacity by combining between predictive and reactive component provisioning approaches to dynamically handle incoming demand spikes. As predictive approaches define the capacity planning big picture (i.e. big time scales), while reactive approaches could help in correcting errors endured during capacity planning such as responding to unplanned demand peaks. The whole idea of provisioning is to allocate just enough components only when they are needed to minimize delivery costs. In other words, we need to avoid both over-provisioning and under-provisioning.

4 Crescent Byzantine Fault Tolerance Approach

Failure of any of the Crescent modules (such as composers and dispatchers) will jeopardize the whole service delivery process, hence we have to guarantee the BFT for Crescent modules. Also we have to guarantee the BFT for the components realizing workflow tasks. If we followed a pure BFT replication approach (such as PBFT protocol [9]), all Crescent modules as well as workflow realizing components had to be replicated, and synchronized, which is not a practical approach. As this will lead to very high communication overhead between all replicas; degrading the overall performance. Hence, we argue that we should combine between practical and quorum-based BFT approaches to minimize such communication overhead, and avoid components replication. To ensure BFT for Crescent modules, we simply adopt any practical BFT protocol. However, these protocols requires replicas synchronization, where each module replica has to communicate with other modules replicas to process the request, which creates massive communication overhead that degrades the service performance. Therefore, we did not apply PBFT protocols strictly, however we did some optimizations to minimize such communication overhead, and to isolate the client from the adopted BFT protocol. We achieved such objectives by adopting the concept of a view and a primary (i.e. a leader) discussed in Paxos [12]. A view in Paxos is a collection of replicas, such replicas elect one replica to be the leader or the primary. All communication should be done via the leader, and the leader should keep other replicas up to date. Each Crescent module view will have a leader that communicates with other views leaders.

In order to ensure BFT for workflow tasks realization, we have to ensure components redundancy (i.e. required by BFT protocols). Crescent achieves such redundancy via component parallel provisioning rather than via component replication in order to avoid the high costs of replication. Hence, requests will be submitted to a components cluster(s), and the majority response of such cluster will be accepted as the correct answer. Redundancy via provisioning does not require replicas synchronization or management, hence communication overhead is heavily minimized, as we will need only one phase for read and write operations. We adopt the speculation principle discussed in Zyzzyva [10] to ensure task delivery BFT, hence we need at least $3f + 1$ components in the components cluster. Such number is guaranteed to exist, as Crescent adaptive composer submits requests to the dispatcher only when parallel composition plan is successfully constructed with component clusters of $3f + 1$ components. The dispatcher in Crescent will be the initiator for the quorum-based protocol, it has to wait for a matching $3f + 1$ responses to accept the response. If it received a number between $2f + 1$ and $3f + 1$, it requires commit certificates from components as in Zyzzyva, if less than $2f + 1$ responses are received, this means the response is compromised. Therefore, Crescent accepts this case as failure and starts a failure recovery protocol to ensure composite web service functional correctness. Error recovery is started by the dispatcher, as it resubmit the request along with a list of compromised components back to the composer in order to find an alternative composition plan with no compromised components.

5 Experiments

In this section, we provide simulation experiments conducted to verify basic Crescent concepts and to compare Crescent against existing approaches for composite web services delivery. We basically compare Crescent against the following three approaches. The first approach does not ensure BFT neither for the composite web service components not for delivery framework modules, as in the approaches discussed in [7] and FACTS [8], also as in reactive approaches currently adopted by industry (such as BPEL). The second approach ensures BFT for the composite web service components but not for the delivery framework modules, as if the composite web service coordinator adopted the BASE approach [3] for ensuring components BFT. The third approach ensures BFT for the delivery framework modules but not for the composite web service components, as in approach discussed in [6]. Crescent ensures the BFT for both the components and delivery framework modules. We compare these approaches using delivery success probability and throughput. As network latency is the main bottleneck affecting response times on the Internet, we will simulate our experiments with high network latency values compared to components processing times. Choice of latency values and processing times is arbitrary as long as such constraint applies. We assumed that there exists 25 ms latency between users and the composite web service delivery system, and 10 ms latency between the delivery system and the composite web service realizing components, also we assumed 5 ms processing time for all composite web service components. We used such values with all approaches in order to have comparable results. The composite service design is arbitrary for our experiments, as ensuring BFT for delivery framework modules and component parallel provisioning are independent from the service design. Hence, we generated a simple composite web service with a workflow of three sequential tasks. Also we generate demand for the composite service following a poisson distribution with an arrival rate 60 requests per minute for 24 h. Then generated demand spikes with arrival rates of 100, 200, 300, 400, and 500 request per

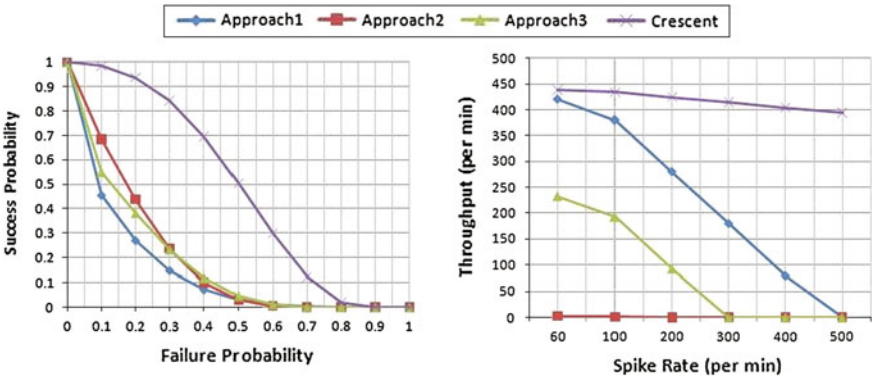

Fig. 2 Crescent comparison experiments

minute for a period of one hour, then we computed the success rate and throughput for all approaches in every case. To minimize costs, we choose redundancy degree of 4, which is enough to guarantee one Byzantine fault tolerance. Figure 2 depicts the obtained results. As we can see Crescent outperformed existing approaches.

6 Conclusion

In this paper, we proposed Crescent a Byzantine fault tolerant delivery framework for durable composite web services. We ensured the BFT for the delivery process by combining between quorum-based and practical BFT approaches, where redundancy in quorum-based approach is achieved via provisioning rather than replication, while the practical BFT approach is optimized by using single leader approach. Experimental results showed that Crescent increases the reliability and throughput of composite web service delivery when compared with existing approaches.

References

1. Zhao W (2007) Bft-ws: A byzantine fault tolerance framework for web services. In: Proceedings of the middleware for web services, workshop
2. Pallemulle SL, Thorvaldsson HD, Goldman KJ (2008) Byzantine fault-tolerant web services for n-tier and service oriented architectures. In: Proceedings of the 28th IEEE international conference on, distributed computing systems (ICDCS)
3. Castro M, Rodrigues R, Liskov B (2003) Base: using abstraction to improve fault tolerance. ACM Trans Comput Syst 21(3):236–269
4. Merideth MG, Iyengar A, Mikalsen T, Rouvellou I, Narasimhan P (2005) Thema: Byzantine-fault-tolerant middleware for web services applications. In: Proceedings of the 24th IEEE symposium on reliable distributed systems (SRDS)
5. Papazoglou MP, van den Heuvel W-J (2005) Web services management: a survey. IEEE Internet Comput 9(6):58–64
6. Zhao W, Zhang H (2008) Byzantine fault tolerant coordination for web services business activities. In: Proceedings of the 2008 IEEE international conference on services, computing, vol 1, pp 407–414
7. Onditi VO, Dobson G, Hutchinson J, Walkerdine J, Sawyer P (2008) Specifying and constructing a fault-tolerant composite service. IEEE sixth European conference on web services
8. Liu A, Li Q, Huang L, Xiao M (2010) Facts: a framework for fault-tolerant composition of transactional web services. IEEE Trans Serv Comput 3(1):46–59
9. Castro M, Liskov B (1999) Practical byzantine fault tolerance. In: Proceedings of the third symposium on Operating systems design and implementation, vol 99, pp 173–186
10. Kotla R, Alvisi L, Dahlin M, Clement A, Wong E (2010) Zyzzyva: speculative byzantine fault tolerance. ACM Trans Comput Syst 27(4):7:1–7:39
11. Elgedawy I, Tari Z, Thom JA (2008) Correctness-aware high-level functional matching approaches for semantic web services. ACM Trans Web, Special Issue on SOC 2(2):12
12. Lamport L (2006) Fast paxos. Distrib Comput 19(2):79–103

Morphological Document Recovery in *HSI* Space

Ederson Marcos Sgarbi, Wellington Aparecido Della Mura, Nikolas Moya and Jacques Facon

Abstract Old documents frequently appear with digitalization errors, uneven background, bleed-through effect etc... Motivated by the challenge to improve printed and handwritten text, we developed a new approach based on morphological color operators using *HSI* color space. Our approach is composed of a morphological background estimation for foreground/background separation and text segmentation, a background smoothing and a color text recovery. Experimental results carried onto ancient documents have proven that *ISH* lexicographic order is the most effective to estimate the background and recover ancient texts in uneven and foxed background images.

1 Introduction

An obstacle in the restoration and interpretation of old document images is the lack of quality. Two distinct problems appear when one works with old scanned documents: the deterioration process of old documents coming from manipulation errors, unsuitable storage environments, stains, foxing marks, ink and paper alterations caused by

E. M. Sgarbi (✉) · W. A. D. Mura
UENP - Campus Luiz Meneghel, Bandeirantes-PR, Brazil
e-mail: sgarbi@uenp.edu.br

W. A. D. Mura
e-mail: wellington@uenp.edu.br

N. Moya
UNICAMP - Universidade de Campinas, Campinas-SP, Brazil
e-mail: nikolasmoya@gmail.com

J. Facon
PPGIa-PUCPR-Pontifícia Universidade Católica do Paraná, Curitiba-PR, Brazil
e-mail: facon@ppgia.pucpr.br

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_24,

the action of moisture, etc... And the scanning process when parts of the verso are mixed with the recto image due to paper transparency.

Some works have been conducted on the document recovery. In [2] the authors have proposed a color document strategy mixing the standard Mean Shift algorithm with a modified one. After converting the RGB image to $L^*u^*v^*$, while a modified Mean Shift algorithm based on R-Nearest Neighbors Colors concept extracts different local maxima, the standard Mean Shift algorithm extracts the global maxima. The pixels are then classified to the closest previously extracted mode. Finally the $L^*u^*v^*$ image is converted back to RGB.

In [8] the authors have presented a restoration strategy based on rational filter. After the RGB image conversion to $YCbCr$, the rational filter is applied to Y luminance to smooth regions and highlight the edges, generating the image Y_r. A tonality adjustment curve is used in Y_r to improve the contrast between the text and the background. This result is multiplied by Y/Y_r to adjust the chrominance. The Otsu 's thresholding method permits to estimate the background image Y_{bw}. The chrominance channels of background pixels are then modified, while the others remain unchanged. Finally the image in the $YCbCr$ color space is converted back to RGB.

In this paper we propose a morphological methodology to make the appearance of the background paper more homogeneous, reducing the changes generated by moisture, stains and foxing marks, preserving the original appearance of the characters. The rest of the paper is organized as follows. Section 2 describes the proposed approach based on morphological background estimation. Experimental results are discussed in Sect. 3.

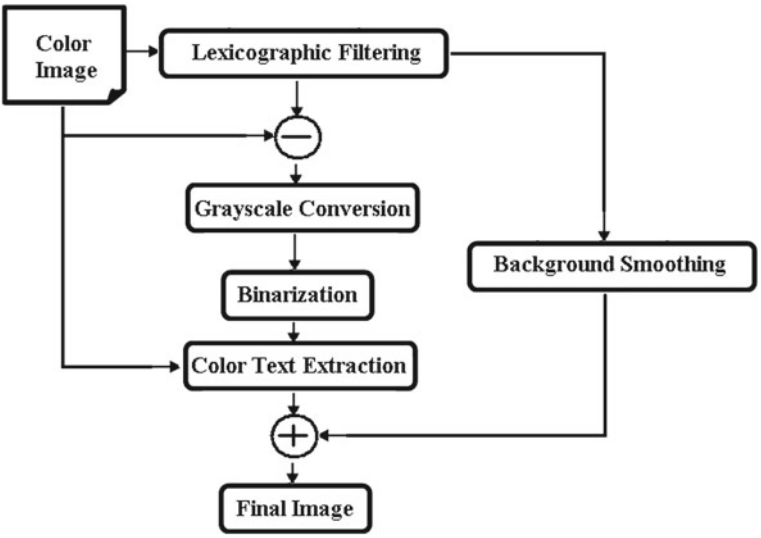

Fig. 1 Flowchart of old document image recovery

2 Methodology

The document recovery strategy is depicted in Fig. 1. The image background approximation is carried out by morphological color operators based on a lexicographic order applied to the HSI color space varying the H, S and I channel order. The foreground/background separation is then performed. After converting the result to grayscale, it is segmented by a binarization technique. And then the document image is recovered by text preservation and background smoothing.

2.1 Mathematical Color Morphology

The ancient document image processing in a wide variety of colors, texts, shapes, damaged by foxing defects, show-through, scanning errors requires flexible tools.

The theory of Mathematical Morphology is to quantitatively describe geometric structures present in the image, and offers a wide range of tools to binary and grayscale images [9]. The advantage of mathematical morphology is its efficiency and robustness for the extraction of inclusions and extrusions in complex images. It is the reason why we have decided to employ mathematical morphology tools to achieve old document recovery in a flexible and fast manner.

The color morphological tools will be used in our approach to estimate the colored background and to recover the colored text. Based on complete lattices, the theory of mathematical morphology, requires an algebraic structure T (complete lattice) and an order relation "$\leq$".

Since there is no natural means for total ordering of multivariate pixels in the case of color images, this notion of ordination is not easy to define. Among color morphology approaches using marginal, reduced, partial and conditional ordering [1], the lexicographic ordering onto HSI color space was chosen in our approach. This choice was motivated by the fact that, by representing a complete order like ordination published in [6], the lexicographic ordering avoids the generation of false colors.

Let's define the three channels as $C1, C2$ and $C3$. The minimum value between two three-component vectors $P_1(C1_1, C2_1, C3_1)$ and $P_2(C1_2, C2_2, C3_2)$, also named as infimum $\wedge$, is defined as follows:

$$\begin{aligned} &\wedge \{P_1(C1_1, C2_1, C3_1), P_2(C1_2, C2_2, C3_2)\} \\ &= \begin{cases} C1_1 < C1_2 \\ or \\ C1_1 = C1_2 \text{ and } C2_1 < C2_2 \\ or \\ C1_1 = C1_2 \text{ and } C2_1 = C2_2 \text{ and } C3_1 < C3_2 \end{cases} \end{aligned} \tag{1}$$

In case of Ci, $i = \{1, 2, 3\}$ channel being the H one, a distance function $d(H_i, H_{ref})$ between H_i and a hue reference H_{ref} is required ([4, 7]) as follows:

$$d(H_i, H_{ref}) = \begin{cases} |H_i - H_{ref}| \text{ if } |H_i - H_{ref}| < \pi \\ or \\ 2\pi - |H_i - H_{ref}| \text{ if } |H_i - H_{ref}| > \pi \end{cases} \quad (2)$$

The maximum value, also named as supremum $\vee$, can be defined in a similar way. Then, from this lexicographic order, morphological erosion ε and dilation δ of image f by using the structuring element B at the pixel x with respect to structuring element support set $E \subset \Re$ are defined as follows:

$$\varepsilon^B(f(x)) = \wedge\{f(y) - B(x - y) : y \in E\} \quad (3)$$

$$\delta^B(f(x)) = \vee\{f(y) + B(x - y) : y \in E\} \quad (4)$$

2.2 Background Estimation

The color background estimation consists in a specific morphological reconstruction ρ,

$$\rho(f) = \lim_{n \to +\infty} \underbrace{\delta_g(\delta_g(...\delta_g(f)))}_{n} \text{ with } \delta_g(f) = \delta(f) \wedge g$$

where the color geodesic dilation $\delta_g(f)$ is based on restricting the color dilation of the marker image f to the mask g. In our proposed approach, the marker image f is composed as follows: the first row and column and the last row and column of the marker image are composed of the original image pixels and the rest of the pixels are defined as $H = S = I = 0$.

2.3 Text Segmentation

Featuring an estimation of the background image with its peculiarities (color mixing, foxing defects, acquisition errors), it is possible to segment the text by subtracting the original image g from its estimated background $\delta_g(f)$. The colored text images are converted to grayscale and binarized using the Johannsen 's process [5].

2.4 Old Document Image Recovery

To recover and reconstruct an old document image, three steps are performed: First the binarized image is used to mask the original colored text in the original image. This step permits to feature an image with original color text. Then an averaged estimation

of background is carried out from the morphological reconstruction image. This step permits to feature an image with averaged background. Finally the reconstruction of the old document image is carried out by adding the two images.

3 Experiments

Experiments were carried out modifying the order of components H, S and I in lexicographic ordination where the H_{ref} value was the majoritary H found in the image.

Figure 2 depicts some results of color background estimation with some channel combinations. By testing all the combinations of the three channels $[(HSI), (SIH), ..., (ISH)]$ to many images, we concluded that the ISH lexicographic order (Fig. 2d) was the only successful one to estimate the background. This order permits a background estimation adding few foreground information while other channel combinations reconstruct too much foreground with background (Fig. 2b, c).

Figures 2, 3 and 4 depict some interesting results of color morphological recovery onto ancient document images. The document images present heterogeneous, discoloration, foxed background and interesting parts as colored text and drawings (Figs. 2a, 3a and 4a).

The depicted results show that without using prior knowledge about the old document images:

- The reconstruction following the ISH lexicographic order has successfully estimated the background (Figs. 2d, 3b and 4b);
- The Johannsen 's process [5] has sucessfully binarized the text (Figs. 3c and 4c);
- The segmentation of colored text is well performed (Figs. 3d and 4d);
- The color background estimation is similar to original image background (Figs. 3e and 4e)
- The recovery of the old document images with original colored text and drawings and color background estimation is realistic (Figs. 2e, 3e and 4e).

Without old document databases including available correspondent groundtruth images, to assess the segmentation efficiency of the proposed approach, a Bible color image with a genuine background and a gothical text was synthesized (Fig. 5). And the text segmentation was evaluated by computing the metrics FM (F-Measure [3]), NRM (Negative Rate Metric [3]). The rates of $FM = 0.99365$ and $NRM = 0.00635$ depicted in Table 1 show the efficiency of proposed approach.

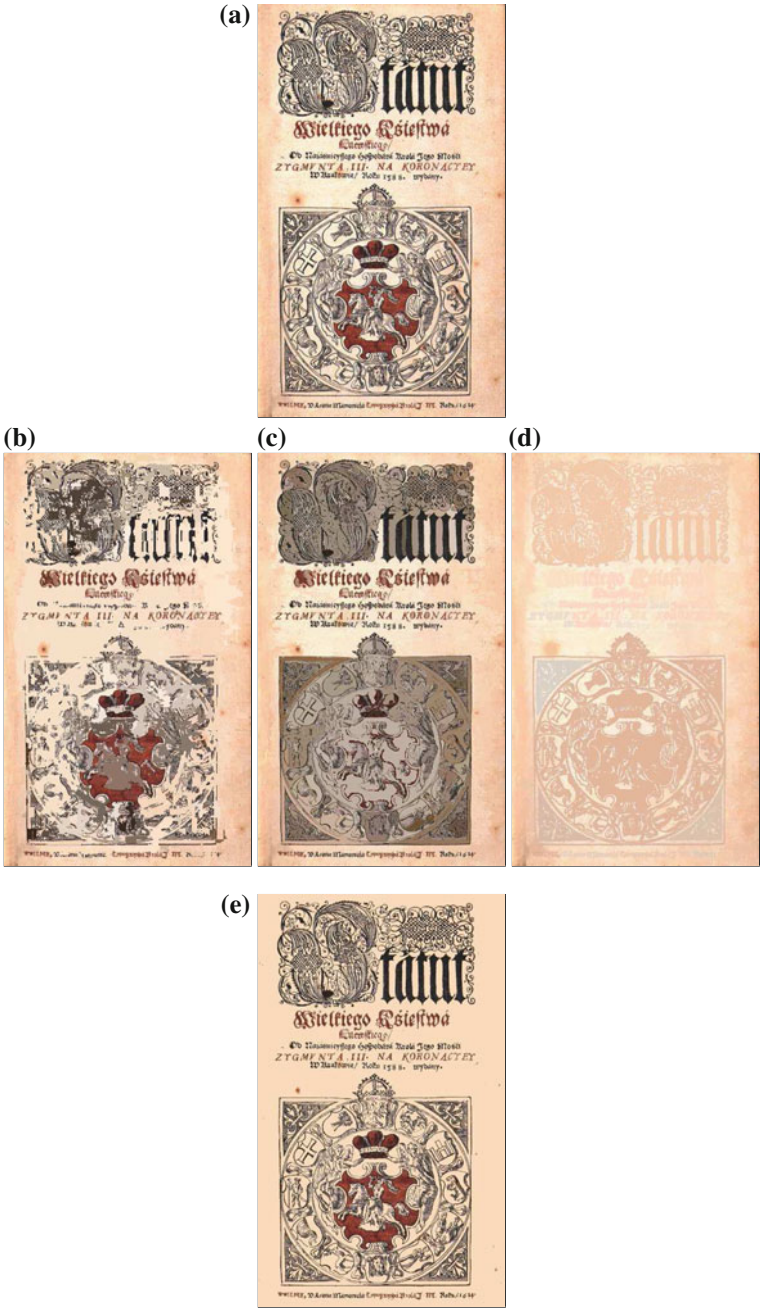

Fig. 2 Experiments: **a** Original image, **b** (HSI), **c** (SHI), **d** (ISH), and **e** Recovery

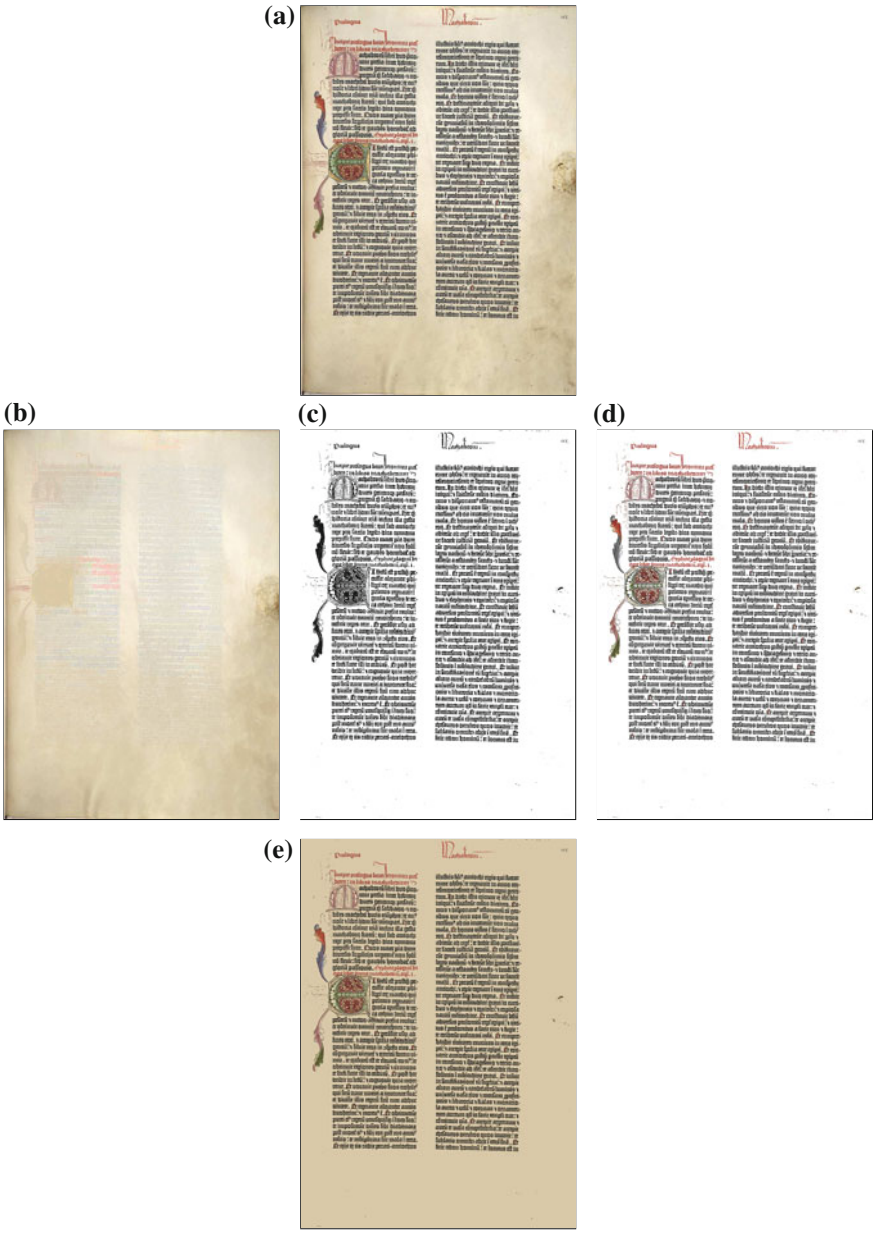

Fig. 3 Recovery results by proposed approach

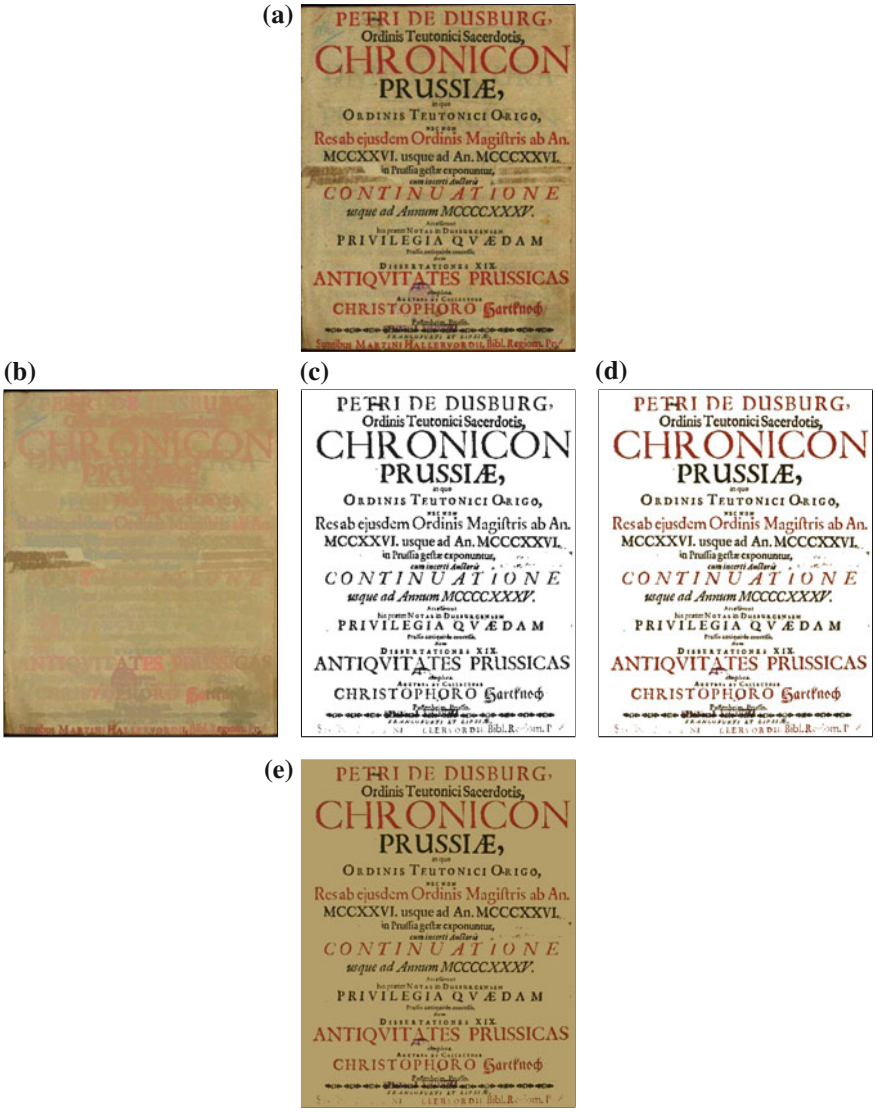

Fig. 4 Recovery results by proposed approach

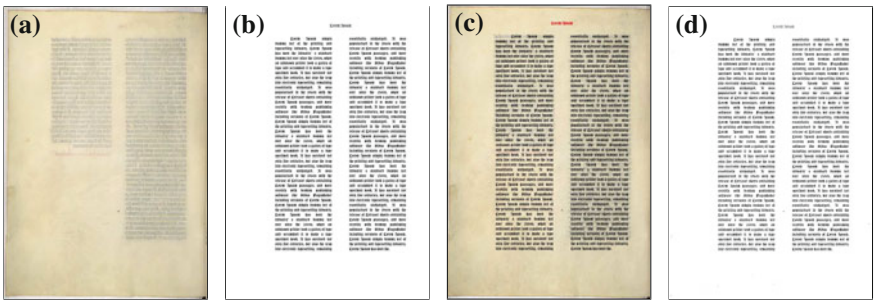

Fig. 5 A synthesized Bible image with a genuine Bible background and a gothical text

Table 1 Text segmentation evaluation using the synthesized image

	FM	NRM
Average	0.99365	0.00635

4 Conclusion

An innovative approach to recover ancient document images using morphological color operators has been proposed. Experimental results carried onto ancient documents have proven that ISH lexicographic order is the most effective to estimate the background and recover ancient texts in uneven and foxed background images. Future works will study the influence of other color spaces.

References

1. Aptoula E, Lefèvre S (2009) Multivariate mathematical morphology applied to color image analysis. In: Collet C, Chanussot J, Chehdi K (eds) Multivariate image processing, Chapter 10
2. Drira F, Lebourgeois F, Emptoz H (2007) A coupled mean shift-anisotropic diffusion approach for document image segmentation and restoration. In: The 9th international conference on document analysis and recognition (ICDAR), pp 814–818
3. Gatos B, Ntirogiannis K, Pratikakis I (2009) ICDAR 2009 document image binarization contest (DIBCO 2009). In: International conference on document analysis and recognition, pp 1375–1382
4. Hanbury A (2001) Lexicographical order in the HLS colour space. Technical report N-04/01/MM, Centre de morphologie mathmatique, cole des Mines de Paris
5. Johannsen G, Bille J (1982) A threshold selection method using information measures. In: Proceedings, 6th internatiional conference on pattern recognition, Munich, Germany, pp 140–143
6. Knuth DE (1998) The art of computer programming. Sorting and searching, vol 3, 2nd edn. Addison-Wesley
7. Peters A II (1970) Mathematical morphology for anglevalued images. In: Proceedings of SPIE, non-linear image processing VIII, vol 3026, pp 84–94

8. Ramponi G, Stanco F, Dello Russo W, Pelusi S, Mauro P (2005) Digital automated restoration of manuscripts and antique printed books. In: Proceedings of EVA 2005 electronic imaging and the visual arts, Firenze, Italy, pp 764–767
9. Soille P (2004) Morphological image analysis: principles and applications, 2nd edn. Springer

Ontological Approach to Data Warehouse Source Integration

Francesco Di Tria, Ezio Lefons and Filippo Tangorra

Abstract In the early stages of data warehouse design, the integration of several source databases must be addressed. Data-oriented and hybrid methodologies need to consider a global schema coming from the integration of source databases, in order to start the conceptual design. Since each database relies on its own conceptual schema, in the integration process a reconciliation phase is necessary, in order to solve syntactical and/or semantic inconsistencies among concepts. In this paper, we present an ontology-based approach to perform the integration of different conceptual schemas automatically.

1 Introduction

The data warehouse design based on hybrid or data-driven methodologies [1, 2] always performs an analysis of the source databases, in order to understand the underlying semantic concepts inherent to the domain of interest [3]. Then, a global schema is produced that represents the source databases in an integrated way. To accomplish the integration process, a reconciliation phase is useful to solve syntactical and/or semantic inconsistencies among the concepts represented in the different databases.

While syntactical problems are traditionally solved using data dictionaries, the current trend to solve semantic problems is based on using an ontological approach [4] instead of the Entity/Relationship (ER) model. The reason is that ER schemas are used to represent locally-true concepts, or concepts that are true in the domain to be modeled. On the contrary, ontologies are used to represent necessarily-true concepts, or concepts that are true in any domain and, therefore, widely accepted and shared.

F. D. Tria (✉) · E. Lefons · F. Tangorra
Dipartimento di Informatica, Università degli Studi di Bari "Aldo Moro",
via Orabona 4, 70125 Bari, Italy
e-mail: francescoditria@di.uniba.it

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_25,

The process of constructing ontologies is called knowledge representation and it requires a lot of effort, because of the difficulties in formulating a comprehensive and rigorous conceptualization in the scope of a given domain. For these reasons, an ontology must be treated as a "reusable" artifact. When used in data warehousing, the designer must rely on a well-formed ontology, avoiding *ad hoc* modifications and extracting the parts of interest.

In this paper, we present an integration strategy based on an ontological approach to produce a global conceptual schema. This is then transformed into a relational schema to be given in input to a methodology for data warehouse design (in particular, the hybrid one we described in [5]).

The paper is organized as follows Sect. 2 presents the related work to the exploitation of ontologies in data warehouse design. Section 3 introduces our hybrid design methodology. Section 4 explains the integration strategy we propose. Section 5 shows a step-by-step example. Finally, Sect. 6 contains a few concluding remarks.

2 Related Work

The main issue in source integration deals with semantic inconsistencies among conceptual schemas. This can be addressed using techniques derived from artificial intelligence [6] and adopting an ontological approach, which is widely used in the semantic web [7].

An important work is described in [8]. The authors' approach is based on local ontologies for designing and implementing a single data source, inherent to a specific domain. Next, the data warehouse design process aims to create a global ontology coming from the integration of the local ontologies. Finally, the global ontology is used along with the logical schemas of the data sources to produce an integrated and reconciled schema, by mapping each local concept to a global ontological concept automatically.

The work of Romero and Abelló [9] is also based on an ontological approach but it skips the integration process and directly considers the generation of a multidimensional schema starting from a common ontology, namely *Cyc*. The final schema must be validated by the user in order to solve inconsistencies.

In [10], the authors propose a methodology to integrate data sources using a common ontology, enriched with a set of functional dependencies. These constraints support the designer in the choice of primary keys for dimension tables and allow the integration of similar concepts using common candidate keys.

3 Methodology Overview

The data warehouse design methodology we propose here is composed of the following phases, in that order:

- *Requirement analysis*. Decision makers' business goals are represented using the i^* framework for data warehousing [11]. The designer has to detect the information requirements and to translate them into a workload, containing the typical queries that allow the extraction of the required information. Then, the goals of the data warehouse must be transformed into a set of constraints, defining facts and dimensions to be included in the multidimensional schema. To this aim, both the workload and the constraints must be given in input to the conceptual design, in order to start the modeling phase in an automatic way.
- *Source analysis and integration*. The schemas of the different data sources must be analyzed and then reconciled, in order to obtain a global conceptual schema. The integration strategy is based on an ontological approach and, therefore, we need to work at the conceptual level. To this end, a reverse engineering from data sources to a conceptual schema is necessary, in order to deal with the concepts. The conceptual schema that results from the integration process must then be transformed into a relational schema, which constitutes the input to the data warehouse conceptual design. Since the transformation primitives from the conceptual to the logical levels are a well-known topic in literature, they are not addressed in this paper.
- *Data warehouse conceptual design*. This phase is based on the Graph-oriented Hybrid Multidimensional Model ($GrHyMM$, [5]) that identifies the facts in the source relational schema on the basis of constraints derived from the *Requirement analysis*. For each correctly-identified fact, it builds an attribute tree [12] to be remodeled using the constraints. Finally, the resulting attribute trees are checked in order to verify whether all the trees agree with the workload [13].
- *Data warehouse logical design*. The attribute trees are transformed into a relational schema—for instance, a snow-flake schema—considering each tree as a cube, having the root as the fact and the branches as the dimensions, possibly structured in hierarchies.
- *Data warehouse physical design*. The design process ends with the definition of the physical properties of the database on the basis of the specific features provided by the database system, such as indexing, partitioning, and so on.

We focus on *Source analysis and integration* phase of the methodology here.

4 Source Analysis and Integration

The preliminary step is the source analysis devoted to the study of the source databases. If necessary, the designer has to produce, for each data source, a conceptual schema along with a data dictionary, storing the description in the natural language

of the concepts modeled by the database. Then, the integration process proceeds incrementally using a binary operator that, given two conceptual schemas, produces a new conceptual schema.

Assumption Given the conceptual schemas $S_1, S_2, \ldots, S_n, n \geq 2$, we assume that
$G_1 = integration\,(S_1, S_2)$, and
$G_i = integration\,(G_{i-1}, S_{i+1})$, for $i = 2, \ldots, n-1$. □

In detail, the integration process of two databases S_i and S_j is composed of the following steps:

1. *Ontological representation*. In this step, we consider an ontology describing the main concepts of the domain of interest. If such an ontology does not exist, it must be built by domain experts. The aim is to build a shared and reusable *ontology*.
2. *Predicate generation*. For each concept in the ontology, we introduce a unary predicate. The output of this step is a set of *predicates*, which represents a vocabulary to build definitions of concepts using the first-order logic.
3. *Ontological definition generation*. For each concept in the ontology, we also introduce a definition on the basis of its semantic relationships. This definition is the description of the concept at the ontological level (that is, the common and shared definition). The output of this step is a set of *ontological definitions*.
4. *Entity definition generation*. For each entity present in the data sources and described in the data dictionary, we introduce a definition using the predicates. Therefore, an entity definition is a logic-based description of a concept in the database. The output of this step is a set of *entity definitions*.
5. *Similarity comparison*. Assuming that similar entities have a very close description, we can detect whether (a) entities that have different names refer to the same concept, and (b) entities that have the same name refer to different concepts. To do so, we utilize a set of inferring rules, the so-called *similarity comparison rules*, to analyze the logic-based descriptions and a metric to calculate the pairwise similarity of entity definitions.

In detail, given two schemas $S_i(A_i^1, A_i^2, \ldots, A_i^o)$ and $S_j(A_j^1, A_j^2, \ldots, A_j^m)$, where A_t^h is the hth entity of schema S_t, we compare the logic definition of A_t^h (for $h = 1, \ldots, o$) with that of A_j^q (for $q = 1, \ldots, m$). For each comparison, we calculate a similarity degree d and an output list K. The output list contains the possible ontological concepts shared by both the logic definitions.

Assuming we can compare the logical definitions of entities A_i^h and A_j^q and calculate both the similarity degree d and the output list K, we can observe one of the following cases:

(i) A_i^h is *equivalent* to A_j^q, if $d \geq x$;
(ii) A_i^h is a *generalization* of A_j^q, if the definition of A_j^q is part of the definition of A_i^h (or *vice versa*);
(iii) A_i^h and A_j^q are both *specializations* of a concept present in the ontology, if $0 < d \leq x$ and $K \neq \emptyset$;

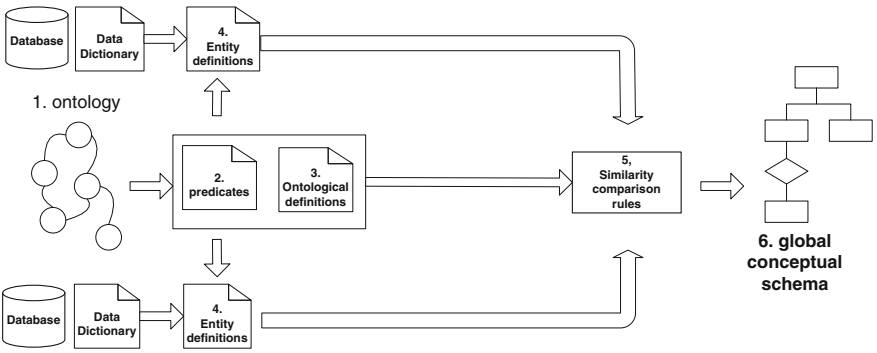

Fig. 1 Integration process diagram

(iv) A_i^h and A_j^q are *linked* via the relationship γ present in the ontology;

where x is a fixed threshold value. For convenience, we fixed x at 0.70.

6. *Global conceptual schema generation*. The final *global conceptual schema* G_u is built using the results obtained by the similarity comparison process and applying some generation rules.

   In detail, we have $G_u(A_u^1, A_u^2, \ldots, A_u^p)$, where for $s = 1, \ldots, p$,

   (i) $A_u^s = A_i^h \approx A_j^q$, if we observe case 5(i);
   (ii) $A_u^s = \{A_i^h, A_j^q\}$, if we observe case 5(ii);
   (iii) $A_u^s = \{K, A_i^h, A_j^q\}$, if we observe case 5(iii);
   (iv) $A_u^s = \{\gamma, A_i^h, A_j^q\}$, if we observe case 5(iv).

   Figure 1 shows the graphical representation of the integration process.

When a further schema S_w has to be integrated, the integration process starts from step 4, using the result of step 6 and the schema S_w.

5 Working Example

In this Section, we provide a complete example of source analysis and integration in order to highlight how the ontology supports the designer in the data warehouse conceptual design.

The case study aims to integrate two databases: (1) *Musical Instruments* and (2) *Fruit & Vegetables*. *Musical Instruments* is the database used by an on-line shop, in order to manage the sales of musical instruments and accessories. *Fruit & Vegetables* is the database used by a farm, in order to manage the wholesale of fruit and vegetables. Their essential conceptual schemas are provided in Fig. 2.

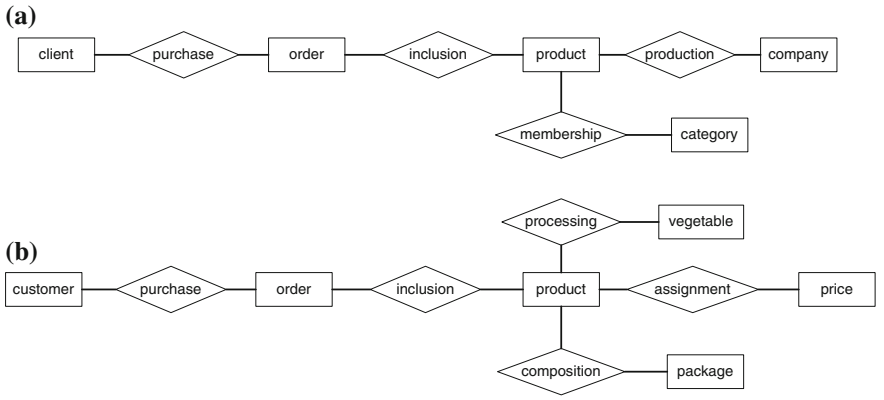

Fig. 2 Source databases: **a** *Musical Instruments*, and **b** *Fruit & Vegetables*

The first phase of the source integration is the ontological representation. To this end, we built our ontology starting from *OpenCyc*, the open source version of *Cyc* [14].

Therefore, we extracted from *OpenCyc* the concepts of interest [15] related to the business companies and sales activity, that is the most frequent domain in data warehousing. The relationships considered are $isA(X, Y)$ to indicate that X is a specialization of Y, and $has(X, Y)$ to indicate that X has an instance of Y.

Using the ontology previously introduced, we defined the predicates to be used as a vocabulary for the logical definitions of database entities. Each predicate corresponds to a concept present in the ontology. For each ontological concept, we also provide an extended definition, using the predicates previously introduced. So, we obtained a logical definition for each ontological concept.

The second phase is the generation of the entity definition.

For each database entity, we created a logical definition using the predicates we had previously generated. Indeed, such predicates represent the vocabulary for the construction of the concepts using the *first-order logic*.

Notice that these definitions often disagree with the ontological ones. In fact, entities are always defined without considering common and shared concepts, since entities represent local concepts. This means we assume that the database designer ignores the ontology. So, given $S_1(client, order, product, company, category)$ and $S_2(customer, order, product, vegetable, package, price)$, we have to create $G_1 = integration\,(S_1, S_2)$, by comparing each entity of S_1 with each entity of S_2.

The third and last phase is the comparison of the entity definitions in order to check whether two entities refer to the same concept or not. The comparison is done automatically using inferring rules defined in *first-order-logic*. These rules check the similarity degree between two lists L_1 and L_2 containing a logical definition of a database entity [16].

The similarity degree d is given by

$$d(n, l, m) = 0.5 \times \frac{l+1}{l+n+2} + (1 - 0.5) \times \frac{l+1}{l+m+2},$$

where

- n is the number of predicates p such that $p \in L_1$ and $p \in L_2$,
- l is the number of common predicates, and
- m is the number of predicates p such that $p \in L_1$ and $p \in L_2$.

The complete result of the case study is reported in Table 1. For each comparison between entities, both the similarity degree d (in the top cell) and the generalization list K (in the bottom cell) are reported. (The symbol "=" means that the entities are *equivalent.*)

5.1 Global Conceptual Schema Generation

Now we examine the results of the similarity comparison. We note that client and customer are always used as synonyms. However, the comparison results indicate that the client and customer have not been defined in the same manner and, therefore, they refer to different database entities. We observe that their similarity degree d is not zero and they present one common ontological concept (*viz*, *social Being*). This suggests introducing into the global schema G_1 the *Social Being* entity and two specializations corresponding to a client who is a social being with an account (that is, a registered user) and a client who is a social being with a legal title (that is, a company having a shop). This has been obtained by applying rule 6(iii) in Sect. 4.

Another generalization that has been detected is that between *product* in *Musical Instruments* and *product* in *Fruit & Vegetables*. Even if there is a syntactical concordance, the terms refer to very different items: the former refers to an instrument,

Table 1 Results of the comparison

		Fruit & Vegetables					
		customer	order	product	vegetable	package	price
Musical Instruments	client	0.367	0.128	0.166	0.183	0.183	0.25
		{*social Being*}	Ø	Ø	Ø	Ø	Ø
	order	0.155	0.888	0.252	0.284	0.311	0.222
		Ø	=	Ø	Ø	Ø	Ø
	product	0.162	0.118	0.583	0.325	0.287	0.229
		Ø	Ø	{*goods*}	Ø	Ø	Ø
	company	0.2	0.155	0.183	0.2	0.2	0.266
		Ø	Ø	Ø	Ø	Ø	Ø
	category	0.225	0.18	0.385	0.417	0.225	0.291
		Ø	Ø	Ø	Ø	Ø	Ø

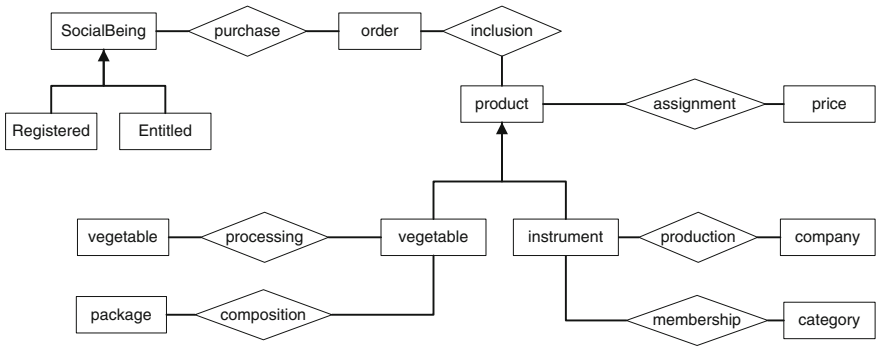

Fig. 3 Global conceptual schema

the latter to a fruit or a vegetable. However, these are both goods having a monetary value and are produced to be sold. Then, we created a generalization, namely *product*, which is an item having an assigned price. The specific products have been introduced as specializations, each with its own relationships. For example, an instrument is produced by a company. On the other hand, the producer of vegetables is missing information in the *Fruit & Vegetables* database. This has also been obtained by applying rule 6(iii) in Sect. 4.

Finally, it is worth noting that the *order* entities have been defined in the same way in both databases. So, they do not present a generalization because they refer to the same concept. This is the only overlapping concept. This has been obtained by applying rule 6(i) in Sect. 4.

The global conceptual schema G_1 is shown in Fig. 3. If we had to add the schema of another source database S_3, we should perform $G_2 = integration\,(G_1, S_3)$. After we have obtained a final global conceptual schema representing an integrated data source, we have to transform this schema into a relational one in order to use it in our hybrid data warehouse design methodology [5].

6 Conclusions

In this paper, we have presented an approach to construct a global conceptual schema coming from the integration of (two) relational databases. This approach is mainly based on an ontology containing common and shared concepts. The language we used is the predicate calculus, in order to define a set of inferring rules to automatically compare the similarity of two entities.

To this aim, we provide a logical definition for each database entity. For the sake of simplicity, we measure the similarity of two logical definitions and, using the comparison results, we are able to state whether the entities refer to the same concept or not. The final conceptual schema is built analyzing the comparison results. Thus,

the definition of an expert system able to reason on the comparison results is our next step to obtain an integrated schema automatically.

Since we claim that this approach can be applied also to attributes, future work will mainly focus on the problems arising when also the similarity between relationships has to be measured. Moreover, we intend to investigate the use of ontology in order to detect any type of ontological relationship existing between entities. In our opinion, this will allow the designer to discover inter-schema relationships.

References

1. Ballard C, Herreman D, Schau D, Bell R, Kim E, Valencic A (1998) Data modeling technique for data warehousing. IBM Corporation
2. Romero O, Abelló A (2009) A survey of multidimensional modeling methodologies. Int J Data Warehouse Min 5:1–23
3. Di Tria F, Lefons E, Tangorra F (2012) Hybrid methodology for data warehouse conceptual design by uml schemas. Inf Software Technol 54(4):360–379
4. Euzenat J, Shvaiko P (2007) Ontology matching. Springer
5. Di Tria F, Lefons E, Tangorra F (2011) GrHyMM: a graph-oriented hybrid multidimensional model. In: Proceedings of the 30th international conference on ER 2011, Brussels, Belgium, LNCS 6999. Springer, pp 86–97
6. Chen Z (2001) Intelligent data warehousing: from data preparation to data mining. CRC Press
7. Sure Y, Erdmann M, Angele J, Staab S, Studer R, Wenke D (2002) OntoEdit: collaborative ontology development for the semantic web. In: Proceedings of the 1st international semantic web conference, Sardinia, Italy, LNCS 2342. Springer Verlag, pp 221–235
8. Hakimpour F, Geppert A (2002) Global schema generation using formal ontologies. In: Proceedings of the 21st international conference on conceptual modeling, Tampere, Finland, LNCS 2503. Springer, pp 307–321
9. Romero O, Abelló A (2010) A framework for multidimensional design of data warehouses from ontologies. Data Knowl Eng 69:1138–1157
10. Bakhtouchi A, Bellatreche L, Ait-Ameur Y (2011) Ontologies and functional dependencies for data integration and reconciliation. In: Proceedings of the 30th international conference on ER 2011, Brussels, Belgium, LNCS 6999. Springer, pp 98–107
11. Mazón JN, Trujillo J, Serrano M, Piattini M et al (2005) Designing data warehouses: from business requirement analysis to multidimensional modeling. In: Cox K (ed) Requirements engineering for business need and it alignment. University of New South, Wales Press, pp 44–53
12. dell'Aquila C, Di Tria F, Lefons E, Tangorra F (2009) Dimensional fact model extension via predicate calculus. In: Proceedings of the 24th international symposium on computer and information sciences. IEEE Press, North Cyprus, pp 211–217
13. dell'Aquila C, Di Tria F, Lefons E, Tangorra F (2010) Logic programming for data warehouse conceptual schema validation. In: Proceedings of the 12th international conference on data warehousing and knowledge discovery, Bilbao, Spain, LNCS 6263. Springer, pp 1–12
14. Lenat DB (1995) Cyc: a large-scale investment in knowledge infrastructure. Commun ACM 38(11):32–38
15. Reed S, Lenat DB (2002) Mapping ontologies in Cyc. AAAI 2002 Conference workshop on ontologies for the semantic web. Edmonton, Canada
16. Ferilli S, Basile TMA, Biba M, Di Mauro N, Esposito F (2009) A general similarity framework for Horn clause logic. Fundam Inf 90(1–2):43–66

Adaptive Oversampling for Imbalanced Data Classification

Şeyda Ertekin

Abstract Data imbalance is known to significantly hinder the generalization performance of supervised learning algorithms. A common strategy to overcome this challenge is synthetic oversampling, where synthetic minority class examples are generated to balance the distribution between the examples of the majority and minority classes. We present a novel adaptive oversampling algorithm, VIRTUAL, that combines the benefits of oversampling and active learning. Unlike traditional resampling methods which require preprocessing of the data, VIRTUAL generates synthetic examples for the minority class during the training process, therefore it removes the need for an extra preprocessing stage. In the context of learning with Support Vector Machines, we demonstrate that VIRTUAL outperforms competitive oversampling techniques both in terms of generalization performance and computational complexity.

1 Introduction

In supervised learning, the accuracy of the predicted labels depends highly on the model's ability to capture an unbiased and sufficient understanding of the characteristics of different classes. In problems where the prevalence of classes is imbalanced, it is necessary to prevent the resultant model from being skewed towards the majority (negative) class and to ensure that the model is capable of reflecting the true nature of the minority (positive) class. Real world applications often result in such skewed models that lead to "predictive misunderstanding" of the minority class, since normal examples which constitute the majority class in classification problems are generally abundant, whereas the examples of interest are generally rare and form the minority class. Examples of classification problems with data imbalance include predicting

Ş. Ertekin (✉)
Massachusetts Institute of Technology, Cambridge, MA 02142, USA
e-mail: seyda@mit.edu

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_26,

pre-term births [13], identifying fraudulent credit card transactions [6], classification of protein databases [19], detecting oil spills from satellite images [18] and fraud detection in mobile telephone communications [15].

The approaches that deal with the problem of imbalanced datasets fall into two major categories, namely data sampling and algorithmic modification. An example of algorithmic modification is to assign distinct costs to classification errors [10, 21], where the misclassification penalty for the positive class is assigned a higher value than of the negative class. Another technique is to use a recognition-based, instead of discrimination-based inductive learning [20]. These methods find the classification boundary by imposing a threshold on the similarity measure between the target class and the query object. The major drawback of those methods is the need for tuning the similarity threshold of which the success of the method mostly relies on. On the other hand, discrimination-based learning algorithms have been shown to yield better prediction performance in most domains. Another major direction is to modify the kernel function or matrix according to the training data distribution to adjust the classification boundary [23]. However, these methods do not perform very well with high imbalance ratios and are hard to implement. In [11, 12], active learning was proposed as a method to deal with the class imbalance problem and was demonstrated to yield better generalization performance than some class-specific weighting methods and resampling techniques.

From a data-centric perspective, the challenges that arise from data imbalance is addressed by two resampling strategies, namely oversampling and undersampling [7, 17]. Both resampling methods introduce additional computational costs of data preprocessing, which can be overwhelming in case of very large scale training data. Undersampling has been proposed as a good means of increasing the sensitivity of a classifier. However, this method may discard potentially useful data that could be important for the learning process. Oversampling has been proposed to create synthetic positive instances from the existing positive samples to increase the representation of the class and has been shown to be highly useful than undersampling and it dramatically improves classifiers' performance even for complex data [9, 14, 16]. No information is lost in oversampling since all samples of the minority and the majority class are preserved. Nevertheless, oversampling causes longer training time during the learning process due to the increased number of training instances. Furthermore, if a complex oversampling method is used, it suffers from high computational costs during preprocessing of the data. It is also inefficient in terms of memory since more instances in addition to the original training data have to be stored. Other costs associated with the learning process (i.e., extended kernel matrix in kernel classification algorithms) are other drawbacks of oversampling.

SMOTE [7] is one of the most widely known oversampling approaches where the minority class is oversampled by creating synthetic examples rather than with replacement. In SMOTE, the k nearest positive neighbors of all positive instances are identified and synthetic positive examples are created and placed randomly along the line segments joining the k minority class nearest neighbors. Various extensions of SMOTE have been proposed that specify which minority examples to oversample [1] and where on the connecting line segment to create a synthetic example [5]. SMOTE

can also be integrated with various learning algorithms [2, 8] therefore it has become a commonly used benchmark for the evaluation of oversampling algorithms. We use SMOTE as one of the baselines in our experiments as well.

In this paper, we propose VIRTUAL (Virtual Instances Resampling Technique Using Active Learning), which is an integrative algorithm of oversampling and Active Learning [22] to form an adaptive technique for the resampling of minority class instances. We constrain our discussion to class imbalance problem in standard two-class classification tasks with Support Vector Machines (SVMs).

2 VIRTUAL Algorithm

VIRTUAL is a hybrid method of oversampling and active learning that forms an adaptive technique for the resampling of minority class instances. In contrast to traditional oversampling techniques that act as an offline step that generate virtual instances of the minority class prior to the training process, VIRTUAL leverages active learning to intelligently and adaptively oversample the data during training, removing the need for an offline and separate preprocessing stage. It uses an incremental online SVM algorithm LASVM [3] where its model is continually updated as it processes training instances one by one. In SVMs, the informativeness of an instance is measured by its distance to the hyperplane which separates two classes from each other. Most informative instances lie in the margin and they are called *support vectors*. Active learning aims to select potentially the most informative instance at each iteration and that corresponds to the closest instance to the hyperplane. VIRTUAL targets the set of support vectors during training with active learning, and resamples new instances based on this set. Since most support vectors are found during early stages of training in active learning, corresponding virtual examples are also created in the early stages. This prevents the algorithm from creating excessive and redundant virtual instances and integrating the resampling process into the training stage improves the efficiency and generalization performance of the learner compared to other competitive oversampling techniques.

2.1 Active Selection of Instances

Let S denote the pool of real and virtual training examples unseen by the learner at each active learning step. Active learning is an iterative process that, at each iteration, queries the instances in S and selects the most informative instance to the learner. Instead of searching for the most informative instance among all the samples in S, VIRTUAL uses an efficient active learning strategy and queries a randomly picked smaller pool from S, as in [11]. From the small pool, VIRTUAL selects an instance that is closest to the existing hyperplane according to the current model. If the selected

instance is a real positive instance (from the original training data) and if it becomes a support vector, VIRTUAL advances to the oversampling step, which is explained next. Otherwise, the algorithm proceeds to the next iteration to select another instance.

2.2 Virtual Instance Generation

VIRTUAL oversamples the real minority class instances (instances selected from the minority class of the original training data) which become support vectors in the current iteration. It selects the k nearest minority class neighbors ($x_{i\to 1} \cdots x_{i\to k}$) of x_i based on their similarities in the kernel transformed higher dimensional feature space. We limit the neighboring instances of x_i to the minority class so that the new virtual instances lie within the minority class distribution. Depending on the amount of oversampling required, the algorithm creates v virtual instances. Each virtual instance lies on any of the line segments joining x_i and its neighbor $x_{i\to j}$ ($j = 1, \ldots, k$). In other words, a neighbor $x_{i\to j}$ is randomly picked and the virtual instance is created as $\bar{x}_v = \lambda \cdot x_i + (1 - \lambda)x_{i\to j}$, where $\lambda \in (0, 1)$ determines the placement of $\bar{x}_v$ between x_i and $x_{i\to j}$. All v virtual instances are added to S and are eligible to be picked by the active learner in the subsequent iterations.

In the beginning, the pool S contains all real instances in the training set. At the end of each iteration, the instance selected is removed from S, and any virtual instances generated are included in the pool S. VIRTUAL terminates when there are no remaining instances in S.

2.3 Complexity Analysis of VIRTUAL

We analyze the computation complexity of SMOTE and VIRTUAL. The computation complexity of VIRTUAL is $O(|SV(+)| \cdot v \cdot \mathscr{C})$, where v is the number of virtual instances created for a real positive support vector in each iteration, $|SV(+)|$ is the number of positive support vectors and $\mathscr{C}$ is the cost of finding k nearest neighbors. The computation complexity of SMOTE is $O(\left|X_R^+\right| \cdot v \cdot \mathscr{C})$, where $\left|X_R^+\right|$ is the number of positive training instances. $\mathscr{C}$ depends on the approach for finding k nearest neighbors. The naive implementation searches all N training instances for the nearest neighbors and thus $\mathscr{C} = kN$. Using advanced data structure such as kd-tree, $\mathscr{C} = k \log N$. Since $|SV(+)|$ is typically much less than $\left|X_R^+\right|$, VIRTUAL incurs lower computation overhead than SMOTE.

3 Experiments

VIRTUAL is compared against two methods, Active Learning (AL) and SMOTE. AL solely adopts the active learning with small pools strategy [11] without preprocessing or creating any virtual instances during learning. On the other hand, SMOTE

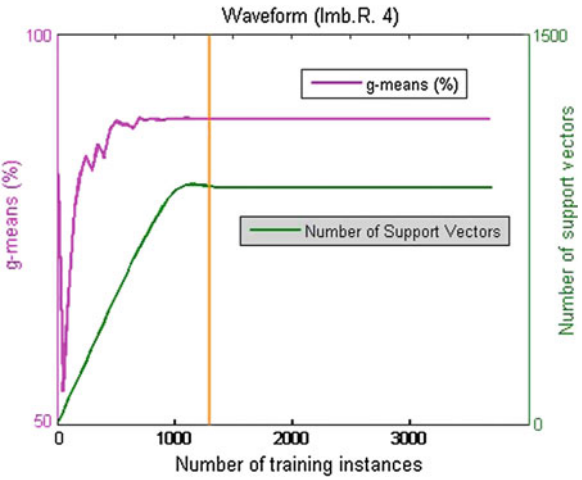

Fig. 1 Saturation of number of support vectors and g-means for *Waveform* (Imb.R.=4). The vertical line indicates where support vectors saturate and training stops

preprocesses the data by creating virtual instances before training and uses random sampling in learning. Experiments elicit the advantages of adaptive virtual sample creation in VIRTUAL. In our experiments, we use the g-means metric which is the geometric mean of sensitivity and specificity, $g = \sqrt{sensitivity \cdot specificity}$. It is a commonly used metric for measuring the performance of algorithms for imbalanced data classification. To achieve high g-means, a classification algorithm should strive to correctly classify both positive and negative examples, and thus a naive algorithm would be penalized for misclassifying positive examples.

On a simulated *Waveform* dataset [4] with an imbalance ration of 4:1, we show in Fig. 1 that when the number of support vectors saturates, g-means also stabilizes. Introducing additional instances to the learner after a point does not change the model, as the training instances in the margin are already exhausted. Accordingly, it makes sense to employ an early stopping criteria to eliminate the remainder of learning iterations which has little, if any, impact on the prediction performance. A theoretically sound method to stop training is to check if there are still unseen training instances in the margin, the distance of the newly selected instance is compared to the support vectors of the current model. If the new selected instance by active learning (closest to the hyperplane) is not closer than any of the support vectors, we conclude that the margin is exhausted. A practical implementation of this idea is to count the number of support vectors during the active learning process. If the number of the support vectors stabilizes, it implies that all possible support vectors have been selected into the model. We adopt this early stopping strategy in our experiments.

We study the performance of VIRTUAL on several benchmark datasets. *Reuters-21578* is a popular text mining dataset and we test the algorithms with the top 10 most populated categories of *Reuters-21578*. We also used 4 datasets from the popular UCI

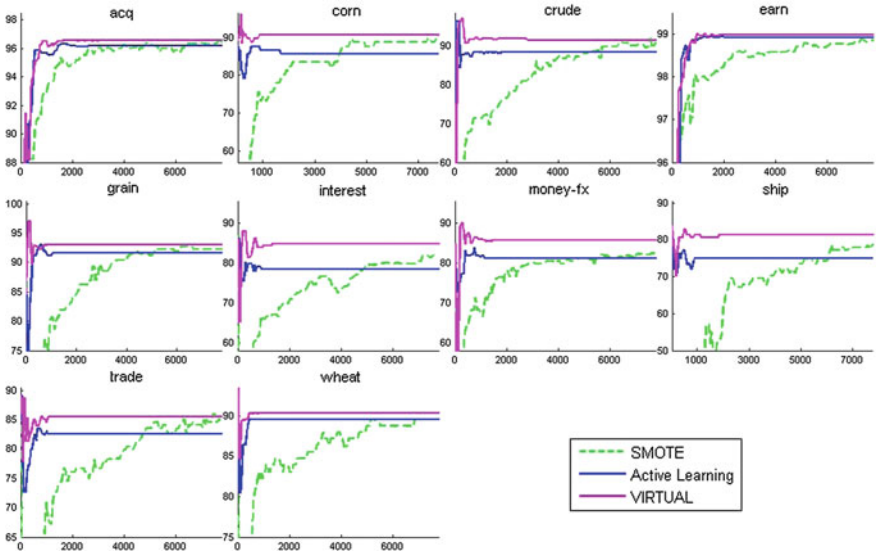

Fig. 2 Comparison of SMOTE, AL and VIRTUAL on 10 largest categories of *Reuters-21578*. We show the g-means (%) (y-axis) of the current model for the test set versus the number of training samples (x-axis) seen

Machine Learning Repository. *Letter* and *satimage* are image datasets. The 'letter A' is used as the positive class in the *letter* dataset and 'class 4' (damp grey soil) is used as positive class in the *satimage* dataset. *Abalone* and *Wisconsin breast cancer* (*breast*) are biology and medical diagnosis datasets respectively. In *abalone*, instances labeled as 'class 7' form the positive class and in *breast*, 'malignant' instances constitute the positive class. These datasets cover a wide range of imbalance ratios.

In Fig. 2, we provide the details on the behavior of SMOTE, AL and VIRTUAL for the Reuters datasets. We note that in all the 10 categories VIRTUAL outperforms AL in g-means metric after saturation. The difference in performance is most pronounced in the more imbalanced categories, e.g. *corn*, *interest* and *ship*. In the less imbalanced datasets such as *acq* and *earn*, the difference in g-means of both methods is less noticeable. The g-means of SMOTE converges much slower than both AL and VIRTUAL. However, SMOTE converges to higher g-means than AL in some of the categories, indicating that the virtual positive examples provide additional information that can be used to improve the model. VIRTUAL converges to the same or even higher g-means than SMOTE while generating fewer virtual instances.

In Table 1, the support vector imbalance ratio of all the three methods are lower than the data imbalance ratio, and VIRTUAL achieves the most balanced ratios of positive and negative support vectors in the Reuters datasets. Despite the fact that the datasets originally have different data distributions, the portion of virtual instances which become support vectors in VIRTUAL is consistently and significantly higher

Table 1 Support vectors with SMOTE (SMT), AL and VIRTUAL

Dataset		Imb.Rt.	$\#SV(-)/\#SV(+)$			$\#SV_V(+)\#$V.I.	
			SMT	AL	VIRTUAL	SMT (%)	VIRTUAL (%)
Reuters	acq	3.7	1.24	1.28	1.18	2.4	**20.3**
	corn	41.9	2.29	3.08	1.95	17.1	**36.6**
	crude	19.0	2.30	2.68	2.00	10.8	**50.4**
	earn	1.7	1.68	1.89	1.67	6.0	**24.2**
	grain	16.9	2.62	3.06	2.32	7.2	**42.3**
	interest	21.4	1.84	2.16	1.66	13.3	**72.2**
	money-fx	13.4	1.86	2.17	1.34	8.2	**31.1**
	ship	38.4	3.45	4.48	2.80	20.0	**66.5**
	trade	20.1	1.89	2.26	1.72	15.4	**26.6**
	wheat	35.7	2.55	3.43	2.22	12.3	**63.9**
UCI	abalone	9.7	0.99	1.24	0.99	30.4	**69.2**
	breast	1.9	1.23	0.60	0.64	2.9	**39.5**
	letter	24.4	1.21	1.48	0.97	0.98	**74.4**
	satimage	9.7	1.31	1.93	0.92	37.3	**53.8**

Imb.Rt. is the data imbalance ratio and $\#SV(-)/\#SV(+)$ represents the support vector imbalance ratio. The rightmost two columns compare the portion of the virtual instances selected as support vectors in SMOTE and VIRTUAL

than that in SMOTE. These results confirm our previous discussion that VIRTUAL is more effective in generating informative instances.

Table 2 presents g-means and the total learning time for SMOTE, AL and VIRTUAL. Classical batch SVM's g-means values are also provided as a reference

Table 2 g-means and total learning time using SMOTE, AL and VIRTUAL

Dataset		g-means(%)				Total learning time (sec)		
		Batch	SMOTE	AL	VIRTUAL	SMOTE	AL	VIRTUAL
Reuters	acq	96.19 (3)	96.21 (2)	96.19 (3)	**96.54 (1)**	2271	146	203
	corn	85.55 (4)	89.62 (2)	86.59 (3)	**90.60 (1)**	74	43	66
	crude	88.34 (4)	91.21 (2)	88.35 (3)	**91.74 (1)**	238	113	129
	earn	98.92 (3)	**98.97 (1)**	98.92 (3)	**98.97 (1)**	4082	121	163
	grain	91.56 (4)	92.29 (2)	91.56 (4)	**93.00 (1)**	296	134	143
	interest	78.45 (4)	83.96 (2)	78.45 (4)	**84.75 (1)**	192	153	178
	money-fx	81.43 (3)	83.70 (2)	81.08 (4)	**85.61 (1)**	363	93	116
	ship	75.66 (3)	78.55 (2)	74.92 (4)	**81.34 (1)**	88	75	76
	trade	82.52 (3)	84.52 (2)	82.52 (3)	**85.48 (1)**	292	72	131
	wheat	89.54 (3)	89.50 (4)	89.55 (2)	**90.27 (1)**	64	29	48
UCI	abalone	**100 (1)**	**100 (1)**	**100 (1)**	**100 (1)**	18	4	6
	breast	98.33 (2)	97.52 (4)	98.33 (2)	**98.84 (1)**	4	1	1
	letter	99.28 (3)	99.42 (2)	99.28 (3)	**99.54 (1)**	83	5	6
	satimage	**83.57 (1)**	82.61 (4)	82.76 (3)	82.92 (2)	219	18	17

"Batch" corresponds to the classical SVM learning in batch setting without resampling. The numbers in brackets denote the rank of the corresponding method in the dataset

point. In Reuters datasets, VIRTUAL yields the highest g-means in all categories. Table 2 shows the effectiveness of adaptive virtual instance generation. In categories *corn*, *interest* and *ship* with high class imbalance ratio, VIRTUAL gains substantial improvement in g-means. Compared to AL, VIRTUAL requires additional time for the creation of virtual instances and selection of those which may become support vectors. Despite this overhead, VIRTUAL's training times are comparable to AL. In cases where minority examples are abundant, SMOTE requires substantially longer time to create virtual instances than VIRTUAL. But as the rightmost columns in Table 1 show, only a small fraction of the virtual instances created by SMOTE become support vectors. Therefore SMOTE spends a large amount of time to create virtual instances that will not be used in the model. On the other hand, VIRTUAL has already a short training time and uses this time to create more informative virtual instances. In Table 2, the numbers in parentheses give the ranks of the g-means prediction performance of the four approaches. The values in bold correspond to a win and VIRTUAL wins in nearly all datasets. The Wilcoxon signed-rank test (2-tailed) between VIRTUAL and its nearest competitor SMOTE reveals that the zero median hypothesis can be rejected at the significance level 1 % ($p = 4.82 \times 10^{-4}$), implying that VIRTUAL performs statistically better than SMOTE in these 14 datasets. These results highlight the importance of creating synthetic samples from the informative instances rather than all the instances.

4 Conclusions

We presented a novel oversampling technique, VIRTUAL, to address the imbalanced data classification problem in SVMs. Rather than creating virtual instances for each positive instance as in conventional oversampling techniques, VIRTUAL adaptively creates instances from the real positive support vectors selected in each active learning step. These instances are informative since they are created at close proximity of the hyperplane. Thus, VIRTUAL needs to create fewer virtual instances and incurs lower overhead in data generation. Empirical analysis shows that VIRTUAL is capable of achieving higher g-means than active learning (AL) and SMOTE. Experimental results also show that VIRTUAL is more resilient to high class imbalance ratios due to its capability of creating more balanced models using the virtual instances created. Furthermore, in most cases, the training time of VIRTUAL is substantially shorter than SMOTE.

References

1. Barua S (2012) Monirul Islam, Xin Yao, and Kazuyuki Murase. Mwmote-majority weighted minority oversampling technique for imbalanced data set learning, IEEE Trans Knowl Data Eng

2. Blagus R, Lusa L (2012) Evaluation of smote for high-dimensional class-imbalanced microarray data. In machine learning and applications (ICMLA), 2012 11th international conference on, IEEE, 2012, vol 2, pp 89–94
3. Bordes A, Ertekin S, Weston J, Bottou L (2005) Fast kernel classifiers with online and active learning. J Mach Learn Res (JMLR) 6:1579–1619
4. Breiman L, Friedman J, Olshen R, Stone C (1984) Classification and regression trees. Wadsworth
5. Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C (2009) Safe-level-smote: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In advances in knowledge discovery and data mining. Springer, pp 475–482
6. Chan PK, Stolfo SJ (1998) Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection. In: Proceedings of the 4th ACM SIGKDD international conference on knowledge discovery and data mining, pp 164–168
7. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority oversampling technique. J Artif Intell Res 16:321–357
8. Chawla NV, Lazarevic A, Hall LO, Bowyer KW (2003) Smoteboost: improving prediction of the minority class in boosting. In knowledge discovery in databases: PKDD 2003. Springer, pp 107–119
9. Chen Sheng, He Haibo, Garcia Edwardo A (2010) Ramoboost: ranked minority oversampling in boosting. IEEE Trans Neural Networks 21(10):1624–1642
10. Domingos P (1999) Metacost: a general method for making classifiers cost-sensitive. In: Proceedings of the 5th international conference on knowledge discovery and data mining, pp 155–164
11. Ertekin S, Huang J, Bottou L, Giles L (2007) Learning on the border: active learning in imbalanced data classification. In: Proceedings of the 16th ACM conference on information and knowledge management (CIKM), ACM, 2007, pp 127–136
12. Ertekin S, Huang J, Giles CL (2007) Active learning for class imbalance problem. In: Proceedings of the 30th annual international ACM SIGIR conference, 2007
13. Grzymala-Busse JW, Zheng Z, Goodwin LK, Grzymala-Busse WJ (2000) An approach to imbalanced datasets based on changing rule strength. In: Proceedings of learning from imbalanced datasets, AAAI workshop
14. He H, Bai Y, Garcia EA, Li S (2008) Adasyn: adaptive synthetic sampling approach for imbalanced learning. In neural networks, 2008. IJCNN 2008. (IEEE world congress on computational intelligence). IEEE international joint conference on, IEEE, 2008, pp 1322–1328
15. Hilas Constantinos S, Mastorocostas Paris As (2008) An application of supervised and unsupervised learning approaches to telecommunications fraud detection. Knowl Based Syst 21(7):721–726
16. Japkowicz N, Stephen S (2002) The class imbalance problem: a systematic study. Intell Data Anal 6(5):429–449
17. Japkowicz N (2000) The class imbalance problem: Significance and strategies. In: Proceedings of 2000 international conference on, artificial intelligence (IC-AI'2000), 1, pp 111–117
18. Kubat M, Holte RC, Matwin S (1998) Machine learning for the detection of oil spills in satellite radar images. Mach Learn 30(2–3):195–215
19. Radivoja P, Chawla NV, Dunker AK, Obradovic Z (2004) Classification and knowledge discovery in protein databases. J Biomed Inf 37(4):224–239
20. Bhavani R, Adam K (2004) Extreme re-balancing for svms: a case study. SIGKDD Explor Newslett 6(1):60–69
21. Thai-Nghe N, Gantner Z, Schmidt-Thieme L (2010) Cost-sensitive learning methods for imbalanced data. In The 2010 international joint Conference on neural networks (IJCNN), IEEE, 2010, pp 1–8
22. Tong S, Koller D (2002) Support vector machine active learning with applications to text classification. J Mach Learn Res (JMLR) 2:45–66
23. Wu G, Chang EY (2004) Aligning boundary in kernel space for learning imbalanced dataset. In: Proceedings of the 4th IEEE international conference on data mining (ICDM 2004),pp 265–272

Part VI
Wireless Sensor Networks

Energy-Aware Distributed Hash Table-Based Bootstrapping Protocol for Randomly Deployed Heterogeneous Wireless Sensor Networks

Ghofrane Fersi, Wassef Louati and Maher Ben Jemaa

Abstract Distributed Hash Table (DHT)-based protocols having the structure of a ring, are promotive solutions to randomly deployed location unaware Wireless Sensor Networks (WSN). These protocols are able to ensure routing and data management efficiently and independently from any location information. However, their bootstrapping phase is challenging since each node should be well positioned into a global virtual ring at the same time. Simultaneous nodes joining leads to multiple inconsistencies. Moreover, existing DHT-based bootstrapping protocols do not take into account nodes heterogeneity whereas most of WSN are nowadays heterogeneous. In this paper, we propose an energy-aware bootstrapping protocol that not only organizes directly the nodes into a global consistent virtual ring but also allows it to build an energy efficient backbone for routing and data management. Simulation results show that our bootstrapping protocol improves drastically the DHT-based routing protocol performance and extends the network lifetime.

1 Introduction

Wireless Sensor Networks (WSN) [11] are emergent solutions to monitor given area or to supervise some phenomena such as temperature, pressure, vibration etc. These networks are made up of wireless sensors having limited processing and memory

G. Fersi (✉) · W. Louati · M. B. Jemaa
Research Unit of Development and Control of Distributed Applications (ReDCAD), Department of Computer Science and Applied Mathematics, National School of Engineers of Sfax, University of Sfax, 1173-3038 Sfax, BP, Tunisia
e-mail: ghofrane.fersi@redcad.org

W. Louati
e-mail: wassef.louati@redcad.org

M. B. Jemaa
e-mail: maher.benjemaa@enis.rnu.tn

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_27,

capabilities with critical power supply. WSN are gaining a growing interest in harsh cases where the supervised area cannot be accessed directly by human beings, which is the case of natural disasters monitoring and hostile detection in wars. This interest is due to the ability of these networks to be randomly deployed and to form autonomously and in a totally distributed manner, a functionnal network.

Distributed Hash Table (DHT)-based protocols having the structure of a ring [3, 7] are well suited to these networks due to their ability to ensure data management and routing tasks in large networks without mattering about the nodes physical locations in a totally distributed manner. The main problem facing these protocols is: how do DHT-based protocols bootstrap? Effectively, bootstrapping in these networks is a challenging issue since it should prepare a consistent infrastructure to be used in data management and routing. This infrastructure consists on organizing all the nodes in the network in a global consistent ring with the increasing order of their identifiers. At the case of randomly deployed Wireless Sensor Networks, all the nodes are scattered simultaneously and search their places in the virtual ring at the same time. Simultaneous search leads to conflicting information about the nodes virtual neighbors. That's why at the end of this step, nodes in the DHT-based protocols are organized into multiple virtual rings. An additional step is required to heal all rings into a global consistent one. This needs a lot of message exchanging which is too energy consuming.

In a previous work, we have proposed a bootstrapping protocol [4] that orchestrates the nodes joining in a way that avoids concurrent nodes joining. This approach assumes that WSN are homogeneous and made up of sensors having the same characteristics. However, nowadays, most of WSN are made up of heterogeneous sensors having different capabilities and various modalities. Since energy optimization is one of the most crucial issues in WSN, it is beneficial to take into account the nodes characteristics when setting up the network in a way that preserves the energy of weak nodes.

In this paper, we propose an energy-aware bootstrapping approach for use in heterogeneous WSN using DHT-based protocols. This approach organizes directly all the nodes into one global consistent ring with constructing an energy-aware backbone where most of the paths between virtual neighbors are mainly made up of energy-powerful nodes. This in turn ensures energy-aware routing.This paper is structured as follows: The related work is given in Sect. 2. In Sect. 3, we specify the problem that we aim solving. We describe our energy-aware bootstrapping protocol in Sect. 4. Simulation results showing the performance of our approach are reported in Sect. 5. Section 6 concludes the paper.

2 Related Work

There are some DHT-based protocols that have proposed bootstrapping approaches. Iterative Successor Pointer Rewiring Protocol (ISPRP) [2] is a simple bootstrapping protocol for use in DHT-based protocols. As a first step, each node maintains exactly

one and only one successor and predecessor. This leads to the formation of multiple consistent local rings. To ensure global ring consistency, a specified node in each ring floods the network which consumes a lot of energy. Virtual Ring Routing (VRR) uses also a similar procedure to ISPRP. Our proposed approach ensures bootstrapping without flooding the network.

The linear method [8] shares with ISPRP the same steps to reach local ring consistency. In order to ensure the global ring consistency, it does not use the flooding step. The linear method assumes that the address space is linear and not a ring. The edges in the virtual graph are undirected. Total ordering of nodes addresses is used in order to distinguish right and left neighbors. To form a global ring, the leftmost node establishes an edge to the rightmost node. Approaches [2, 8] require two steps in order to reach the steady state: multiple consistent ring formation, then, rings fusion. Our protocol does not need any ring merging since it ensures directly the formation of a global consistent ring.

In [4, 5], we have proposed a bootstrapping protocol for use in DHT-based protocols. This bootstrapping scheme organizes the nodes into a tree in order to orchestrate their joining process and avoid at most concurrent nodes joining. Such bootstrapping strategy allows the direct formation of a global consistent ring without the need of a lot of message exchanging. However, this bootstrapping protocol does not take into account the nodes heterogeneity.

For the best of our knowledge, our proposed protocol is the first DHT-based bootstrapping protocol that takes into account the nodes heterogeneity and that takes advantage from this diversity to prepare an energy-efficient backbone that improves the performance of DHT-based routing protocols.

3 Problem Description

Given a set of sensors scattered randomly in a field without having any location information. These sensors are heterogeneous and are classified into two types: Weak sensors, having a very critical amount of energy $E < E_0$ and strong sensors having more important amount of energy $E >= E_0$, where E_0 is the strong nodes energy threshold. Since strong nodes have relatively high amounts of energy, their transmitting power can be increased in order to increase their transmitting range. We assume that the graph presenting the network is connected. In other words, we assume that each node in the network is able to communicate with at least another node. Physical neighbors are detected by HELLO beacons exchanging. Since we have here two different transmitting powers, physical neighbors detection is ensured by two handshake messages. In order to discover its physical neighbors, a node broadcasts HELLO messages. The nodes from which it receives HELLO response messages are considered as its physical neighbors. Nodes identifiers are independent from any location information. They can be the Hash of their MAC addresses. Initially, all the nodes have no information about their virtual neighbors.The problem that we aim

solving in this paper is how to organize all the nodes into one global consistent virtual ring and how to take advantage from nodes having more energy supply in order to improve the performance of the DHT based routing protocol?

4 The Proposed Bootstrapping Approach

In this section, we describe our proposed approach in details. All the nodes in our approach are organized into a tree. There is, from the beginning, one initially active node. This node should be strong and is chosen by the network administrator before the network deployment. In a given network there is one and only one initially active node in order to avoid concurrency problems. When scattered, it allows its child strong node having the smallest identifier to join the network. If such a strong node does not exist, the child weak node having the smallest identifier joins the network and so on. When the current joining node has no more children, it sends a message to its parent that in turn chooses in the same way another child node to join the network. Recursively, all the nodes join the network using at most strong nodes in the joining process. We apply in this section our bootstrapping approach to VRR. There is from the beginning a maximal transmitting power Pt_1 for weak nodes and a maximal transmitting power Pt_2 for strong nodes. In the first round, the initially active strong node starts by broadcasting HELLO beacons with its maximal transmitting power Pt_1 in order to cover the maximum nodes number, and starts a *hearing* step. During this step, each strong node having received this beacon, responds by a HELLO response message. When a weak node receives this HELLO beacon, it computes the distance between it and the source node according to the Received Signal Strength Indicator (RSSI) as above. The HELLO beacon contains a field indicating the nature of the source node: weak or strong. At the reception of this beacon, the destination node consults this field in order to deduce the used transmitting power (Pt_1 if a strong node and Pt_2 otherwise). According to the Friis free space model [6] we have:

$$Pr = Pt((\lambda^2 \cdot Gt \cdot Gr)/((4 \cdot \pi \cdot d)^2 \cdot L) \tag{1}$$

where Pr is the received power and Pt is the transmitting power. We assume in our case that transmit gain Gt, reception gain Gr are equal to 1, the system loss L is equal to 1.2. (The system loss includes all the factors that can attenuate the received power strength such as interference, reflection, refraction, etc), λ is the wavelength. The distance d between the transmitting and receiving node can be estimated by the receiving node as follows:

$$d = \sqrt{(Pt \cdot \lambda^2 \cdot Gt \cdot Gr)/(Pr \cdot 16 \cdot \pi^2 \cdot L)} \tag{2}$$

The receiving node can now calculate the required transmitting power to deliver successfully a message:

Algorithm 1 Bootstrapping pseudo code: *FirstNode* checks whether the current node is the first node or not. *src* is the identifier of the source node, *SentHELLOResponse* checks if the current node has already sent HELLO Reponse message or not. *RequiredPt* is the necessary transmitting power to deliver successfully a message to a given node. *PtMax* is the maximal transmitting power that a node can use in transmitting messages. *waiting_set* is the set containing the children identifiers that are waiting to join the network, *nbWaitingSet* is the number of nodes in the waiting set, *me* is the identifier of the current node. *MyParent* is the identifier of the parent of the current node. *Ring_based_protocol_joining* applies the employed ring based protocol joining process to the current node.

```
if FirstNode then
    status=active
    sendHello()
else
    status=inactive
end if
procedure RECEIVE(HELLO, src)
    if SentHELLOResponse = false then
        if nature = strong then
            send HELLOResponse(me) to src
        else
            if RequiredPt ≤ PtMax then
                send HELLOResponse(me) to src
            end if
        end if
    else
        if nature(myParent) = weak & nature(src) = strong then
            send_delete(me) to myParent
        end if
    end if
end procedure
procedure RECEIVE(HELLOResponse, src)
    add waiting_set (src)
    nbWaitingSet=nbWaitingSet+1
    if timerHearing isExpired then
        if nbWaitingSet ≥ 1 then
            send_Permission(me) to waiting_set(1)
        else
            send_Reallocation(me) to myParent
        end if
    end if
end procedure
procedure RECEIVE(Permission, src)
    Ring_based_protocol_joining(me)
    status=active
    sendHello()
end procedure
procedure RECEIVE(Reallocation, src)
    remove waiting_set (src)
    nbWaitingSet=nbWaitingSet-1
    if nbWaitingSet ≥ 1 then
        send_Permission(me) to waiting_set(0)
    else
        send_Reallocation(me) to myParent
    end if
end procedure
procedure RECEIVE(Delete, src)
    delete waiting_set (src)
    nbWaitingSet=nbWaitingSet-1
end procedure
```

$$Pt_{required} = (d^2 \cdot RXThresh \cdot 16 \cdot \pi^2 \cdot L)/(\lambda^2 \cdot Gt \cdot Gr) \quad (3)$$

RXThresh is the receiving threshold in the network interface. It is the minimum received signal strength that is necessary for correctly decoding the messages.

Each wireless interface has multiple transmission power levels in order to support fine-grained power control. For example, a 433 MHz CC1000 radio, has a total of 31 transmission power levels [10]. The sensor power level that should be chosen is the level verifying that:

$$Elected\ power\ transmission\ level = min\left\{Li >= Pt_{required}\right\} \quad (4)$$

where L_i is the sensor transmission power levels.

If $Elected\ power\ transmission\ level <= Pt_{Max}$ (Pt_{Max} is the maximal allowed transmitting power for a weak sensor), the weak node is able to send a message to the strong node. Otherwise, it ignores these HELLO beacons.

At the end of the hearing step, the active node sorts the identifiers of the strong nodes from which it has received HELLO response messages. Then it sorts the identifiers of the weak nodes that have responded it and stores all the nodes in its *waiting set*. After that, it sends to the first node in its waiting set a Permission message allowing it to start its joining process. This node will be the virtual neighbor of the first active node. The newly active node starts the round 2 and broadcasts HELLO beacons. Each strong node in round i that have received HELLO beacon and that did not have sent any HELLO response message to any node in all the i-1 rounds sends directly HELLO response message to the source node of the HELLO beacon. If a weak node that did not have sent any HELLO response message during the i-1 rounds receives this HELLO message, it should first verify its ability to deliver successfully the message to the source node by computing the required transmitting power as stated before, and then decides whether to respond or not to this HELLO message. Since the nodes are scattered randomly and there are only few strong nodes in the network, there will be some cases were the current strong node do not receive any HELLO response messages from any strong nodes. It receives only HELLO responses from weak nodes since the other strong nodes are placed far from it. In such cases, the current strong node sorts only the identifiers of the weak nodes and sends a Permission message to the first weak node in its waiting set.

When a node receives a Permission message it starts its joining process. It sends a setup request message with its own identifer to its corresponding parent (considered as its *proxy*). At the reception of this message, the parent node picks from its routing table the node having the closest identifer to the joining node's identifier and forwards the message to its corresponding physical hop. This procedure is repeated until the node having the closest identifier to the joining node's identifier is reached. This node is considered as the virtual neighbor of the joining node. It adds this new joining node as a new virtual neighbor and responds it by a setup message. This message is routed recursively to the proxy node using the same procedure of the setup request message routing. Once the proxy is reached, this latter sends the message to the new joining

node. This joining node is then informed about the identifier of its virtual neighbor. The source, the destination as well as all the intermediate nodes of the setup message save in their routing tables the next physical hops towards the new virtual path. The setup message contains also the identifiers of the virtual neighbors of the source node. The new joining node chooses its alternative virtual predecessor or successor and sends it a setup request message. The node that receives a setup request from a node whose identifier is closer to its identifier than the identifier of its current neighbor, adds the new node as its virtual neighbor and deletes the old virtual neighbor from its routing table and from its virtual neighbors set. It also sends a teardown message to the old virtual neighbor in order to delete all the path between them. We can easily deduce that the path between the weak node and its virtual neighbors is mainly made up of strong intermediate nodes. When this weak node is well placed into the virtual ring, it starts the next round by broadcasting HELLO messages. All the steps stated before are repeated until a new strong node becomes active and starts broadcasting HELLO beacons. HELLO beacons can be received by:

- Strong nodes: Respond directly the source node by a HELLO response message.
- Weak nodes that are already associated and whose parents are weak nodes: if they have not yet joined the network, they verify if their transmitting powers allow them to send HELLO response to this strong node. In such a case, the weak node is associated to this new strong node and sends *Delete* message to its old parent to favour the use of the strong nodes in the setup process.
- Weak nodes that are not yet associated: If their transmitting power allows them to respond this strong node, they will respond it by HELLO response message.

When the recently active node does not receive during its hearing step any HELLO response message, it realizes that the tree branch is finished. It sends to its parent a *Reallocation* message. When the parent node receives this message, it sends a *Permission* message to the next node in its waiting set. When the parent receives a *Reallocation* message from its child and its waiting set is empty, it sends in turn a Reallocation message to its parent. When the waiting set of the first active node in the network becomes empty, the steady state is reached. In order to face the problem of message loss, each received message should be acknowledged after at most 1 second. Figure 1 illustrates an example of our proposed bootstrapping scheme. Node 1 is the root node. It broadcasts HELLO messages and starts its hearing process. The order of all nodes joining is given in black numbers. Our proposed approach pseudcode is given in algorithm 1.

5 Simulation Results

All the simulations were carried out using NS2.33 [9]. We compared the perfomance of VRR after the application of the bootstrapping protocol considering that the network is homogeneous [4] and after the application of the energy-aware bootstrapping protocol taking into account the nodes energy levels.

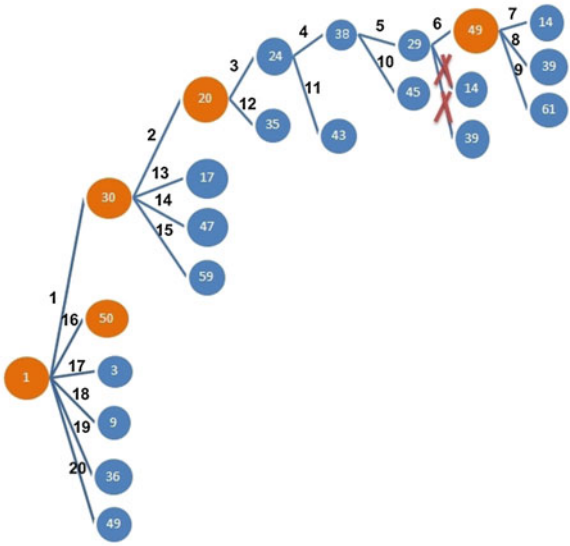

Fig. 1 Bootstrapping tree

We varied the number of nodes from 50 to 80. The percentage of strong nodes is 10 % in all simulation scenario. The transmitting ranges of weak and strong nodes are respectively 40 and 85 meters. The initial amounts of energy are respectively 0.5 Joules for weak nodes and 2 Joules for strong nodes. All the nodes in the network send each 200 seconds a message to the sink.

We measured the time of the first node's death, the average number of hops needed to route a message from a given source node to the sink and the percentage of strong nodes that have participated in the routing process. We have also measured the consumed energy by all the nodes in a 70-Nodes network after 44 transmissions from all the nodes to the sink.

Figure 2 shows that applying the energy-aware approach increases the number of the strong nodes that are used in routing. The energy-aware bootstrapping protocol builds at the network startup a strong backbone mainly made by strong nodes. Most of the weak nodes use multiple intermediate strong nodes to build their routing tables and to find their neighbors. Since the transmitting range of these strong nodes is important, the average number of hops in the VRR routing process is decreased as shown in Fig. 3.

Figure 4 shows that the lifetime of the network is drastically increased when VRR is used after the application of an energy-aware bootstrapping phase. Effectively, the bootstrapping strategy favours the use of the strong nodes in the network setup and consquently most of the routing tables that VRR uses in its routing process have strong nodes as next hops. Hence, the number of weak nodes intermediate nodes in the routing process will be decreased. Their amount of energy will be preserved which extends their lifetime.

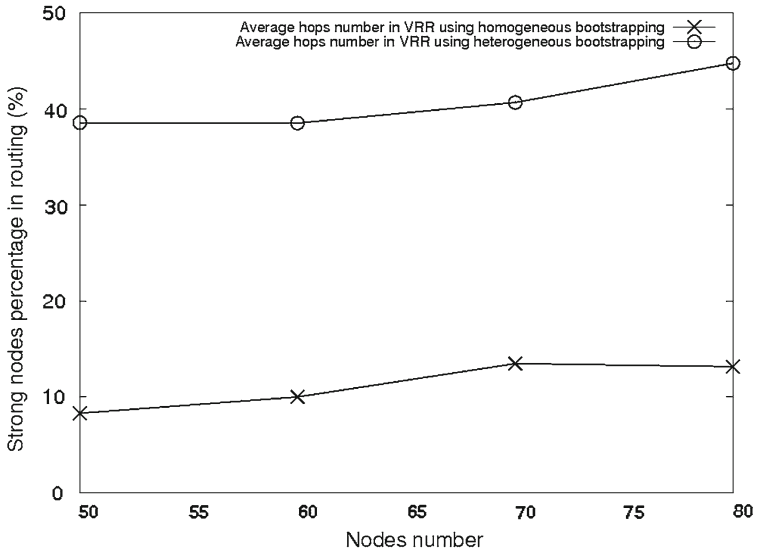

Fig. 2 Percentage of strong nodes in routing

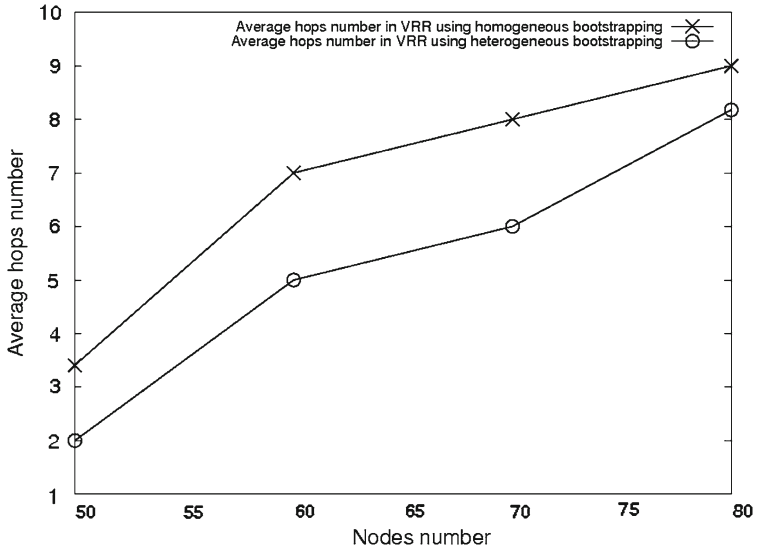

Fig. 3 Routing performance

Figure 5 depicts the consumed energy by each sensor during 44 VRR transmissions between these nodes and the sink. Most of the nodes consume more important amounts of energy when VRR is applied after the application of the energy-unaware bootstrapping protocol. This is because this bootstrapping strategy does not take into account nodes energy levels in the network setup.

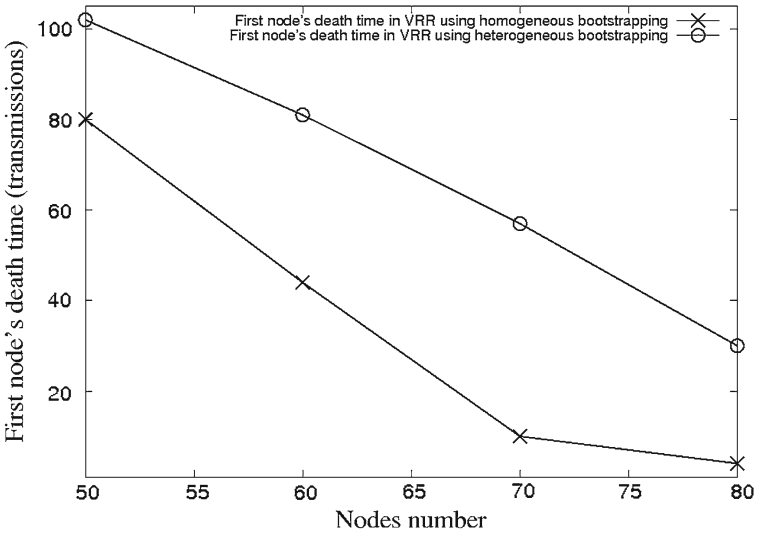

Fig. 4 Network lifetime

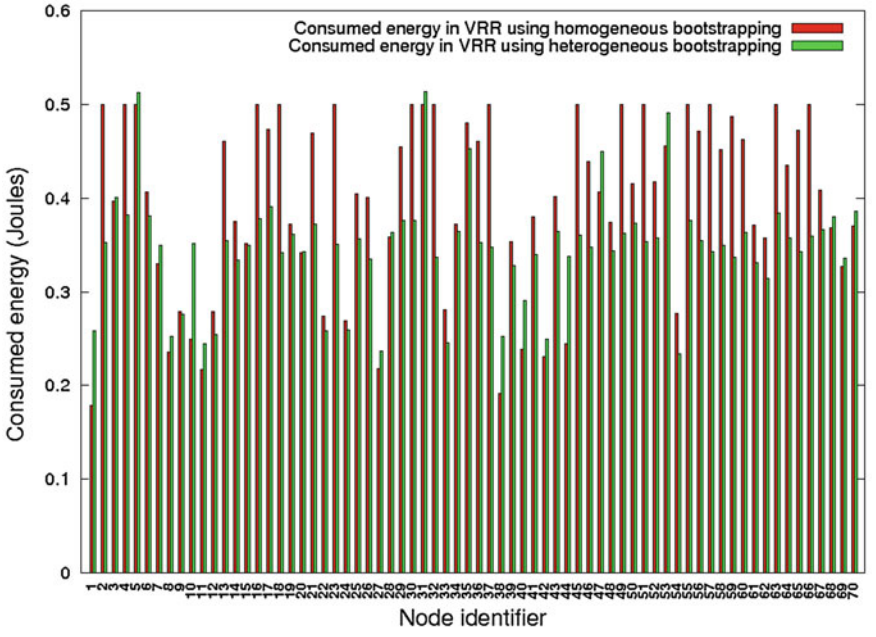

Fig. 5 Consumed energy in a 70-Nodes network

6 Conclusion

In this paper, we have presented an energy-aware DHT-based bootstrapping protocol for heterogeneous randomly deployed WSN. The main goal of our protocol is to build simultaneously and efficiently a global consistent ring encompassing all the nodes in the network and an energy-aware network backbone.

The main results reported in this paper showed that our bootstrapping approach achieves efficiently the nodes organization into a virtual ring. It also initializes nodes routing tables in a way that allows efficient DHT-based routing, minimizing the total number of hops to reach the destination in the routing process. These hops are mainly achieved by strong nodes. This preserves the weak nodes energy and extends their lifetime as depicted in the simulation results.

References

1. Caesar M, Castro M, Nightingale EB, O'Shea G, Rowstron A (2006) Virtual ring routing: Network routing inspired by DHTs. In: Proceedings of SIGCOMM, Pisa, Italy, 2006
2. Cramer C, Fuhrmann T (2005) Self-stabilizing ring networks on connected graphs. University of Karlsruhe (TH), Fakultaet fuer Informatik, Technical report 2005-5
3. Fersi G, Louati W (2013) Distributed hash table-based routing and data management in wireless sensor networks: a survey. ACM/Springer. Wirel Netw 19(2):219–236
4. Fersi G, Louati W, Ben Jemaa M (2013) Consistent and efficient bootstrapping ring-based protocol in randomly deployed wireless sensor networks. In: Proceedings of intenational conference on telecommunications (ICT), Maroc, 2013
5. Fersi G, Louati W, Ben Jemaa M (2013) Time estimation of a ring-based bootstrapping protocol in wireless sensor networks. In: Proceedings of livecity workshop on smart and pervasive communications for enhanced communities in conjunction with SACONET conference, France, 2013
6. Friis HT (1946) A note on a simple transmission formula. In: Proceedings of IRE, pp 254–256, 1946
7. Fuhrmann T (2005) The use of scalable source routing for networked sensors. In: Proceedings of the 2nd IEEE workshop on embedded networked sensors, Australia, 2005
8. Kutzner K, Fuhrmann T (2007) Using linearization for global consistency in SSR. In: Proceedings of 4th international IEEE workshop on hot topics in P2P systems, Long Beach, CA, 2007
9. NS2 website Available at http://www.isi.edu/nsnam/ns/
10. Xing G, Lu C, Zhang Y, Huang Q, Pless R (2007) Minimum power configuration for wireless communication in sensor networks. ACM Trans Sens Netw 3:11
11. Yick J, Mukherjee B, Ghosal D (2008) Wireless sensor network survey. Comput Netw 52(12):2292–2330

Sensor-Activity Relevance in Human Activity Recognition with Wearable Motion Sensors and Mutual Information Criterion

Oğuzcan Dobrucalı and Billur Barshan

Abstract Selecting a suitable sensor configuration is an important aspect of recognizing human activities with wearable motion sensors. This problem encompasses selecting the number and type of the sensors, configuring them on the human body, and identifying the most informative sensor axes. In earlier work, researchers have used customized sensor configurations and compared their activity recognition rates with those of others. However, the results of these comparisons are dependent on the feature sets and the classifiers employed. In this study, we propose a novel approach that utilizes the time-domain distributions of the raw sensor measurements. We determine the most informative sensor types (among accelerometers, gyroscopes, and magnetometers), sensor locations (among torso, arms, and legs), and measurement axes (among three perpendicular coordinate axes at each sensor) based on the mutual information criterion.

1 Introduction

Automatic recognition of human activities has received considerable attention in the recent years. The ambiguous nature of daily human activities makes their correct recognition a challenging problem: Any specific activity can be performed in different styles by different people, a person can perform multiple activities simultaneously, and there is no one-to-one cause-effect relationship between consecutive activities [1]. Using a set of sensors and intelligent detection algorithms, activities can be automatically recognized and classified. In supervised approaches, activities are

O. Dobrucalı (✉) · B. Barshan
Department of Electrical and Electronics Engineering, Bilkent University,
06800Ankara, Bilkent, Turkey
e-mail: dobrucali@ee.bilkent.edu.tr

B. Barshan
e-mail: billur@ee.bilkent.edu.tr

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_28,

recognized based on priorly trained recognition models, whereas in unsupervised approaches, no prior models are assumed. The application areas of human activity recognition include identification systems that use the individual's physical and behavioural attributes (e.g., gait patterns), surveillance systems detecting unusual human activities, interactive systems responding according to the users' behaviour, synthesis of human activities in entertainment and robotics industries, medical treatment, and autonomous nursing of the elderly and disabled people [2, 3]. In addition to these, some recent studies are focused on localization of pedestrians and recognition of their activities at the same time [4].

Visual data recorded by video cameras [3], motion data acquired from wearable inertial and magnetic sensors [2], and acoustic data captured from microphones and vibration sensors [5] are the measurement types widely used in human activity recognition. In some studies, sensors are embedded in the environment, limiting the mobility area of the user, whereas in others, they are worn on the body to directly acquire motion data. In the latter, each sensor unit comprises a set of sensors that may include accelerometers, gyroscopes, magnetometers, inclinometers, goniometers, and tilt switches [2]. Motion data are analysed while being captured at either a single or multiple sensor locations. Commonly used sensor locations in recent studies are the head, ears, shoulders, wrists, torso, waist, thighs, calves, and ankles. Wearable sensors are preferable because they provide unbounded monitoring area and do not interfere with privacy. On the other hand, the user may forget to put them on or may feel distressed and uncomfortable while wearing them [6].

Sensor configuration is a fundamental issue in using wearable motion sensors that involves specifying (1) the number and type of sensors employed in each sensor unit, (2) the positions where the sensor units are placed, and (3) the axes along which each sensor provides informative measurements. To boost the overall recognition performance, feature sets in the time- and frequency-domains are commonly determined with respect to fixed sensor configurations [7]. The missing aspect here is an objective criterion for selecting a suitable sensor configuration. Various sensor configurations have been proposed in the literature. A couple of comparative studies [8–11] investigate sensor configurations and feature sets in terms of recognition accuracy. Optimal feature sets are selected while the sensor configuration is employed at full capacity[1] and vice versa. Consequently, any choice of sensor configuration seems to be highly dependent on the feature set. This fact, for instance, is explicitly shown in [7], where the authors compare the recognition performances of different feature sets computed from the measurements captured at several locations. Employing a basic classifier, they show that the contributions of the subsets of the sensor locations to the recognition performance vary, as does the feature set.

In this study, we identify the most informative sensor configuration by using the time-domain distributions of raw sensor data and the mutual information criterion. The approach presented here is independent of any feature set and classifier constraint, making it more objective and reliable than those mentioned above.

[1] *The sensor configuration at full capacity* implies the configuration employing all available sensor types, sensor locations, and measurement axes.

The rest of this article is organized as follows: In the next section, we briefly describe the human activity dataset used in this study. In Sect. 3, we introduce the mutual information criterion for determining the informative sensor configurations. We provide the methodology and the results of the analysis in Sect. 4. Finally, we summarize the throughputs of this study and provide directions for future research in the last section.

2 Human Activity Dataset

The dataset analysed in this study is acquired by our research group at Bilkent University and used in some of our previous studies [8, 11, 12]. Descriptions of the 19 types of daily and sports activities included in this dataset and the physical characteristics of the eight subjects can be found in [8] and [12], respectively.

In this study, sensor units are placed at five locations on the body: torso (T), right arm (RA), left arm (LA), right leg (RL), and left leg (LL). Each sensor unit includes three tri-axial devices: an accelerometer (ACC), a gyroscope (GYRO), and a magnetometer (MAGN), whose measurement axes are aligned with a reference Cartesian coordinate frame, as depicted in Fig. 1a. The operating ranges of these devices are $\pm 18g$, $\pm 1200°/s$, and $\pm 75\ \mu$T, respectively [13]. Here, g is the gravitational acceleration constant which is 9.80665 m/s^2. The positioning of the sensor units on the body is illustrated in Fig. 1b.

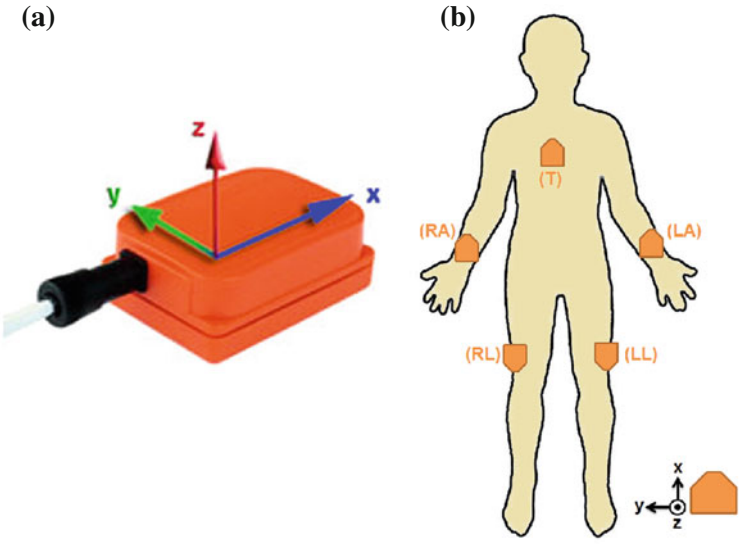

Fig. 1 **a** The sensor unit used in this study (reprinted from www.xsens.com/en/general/mtx) and **b** the unit positions on the subject's body. (The outline of the human body figure is taken from www.anatomyacts.co.uk/learning/primary/Montage.htm)

The set of activities in the dataset is denoted by $\mathcal{A}$. When a subject performs an activity, the signals captured from the sensor axes on the subject's body are simultaneously recorded at 25 Hz sampling rate. Let each recorded discrete-time sequence be denoted by $U_{ijkc}[n]$ where $i \in \{1, 2, \ldots, 5\}$ is the sensor location index representing {T, RA, LA, RL, LL} respectively, $j \in \{1, 2, 3\}$ is the sensor type index representing {ACC, GYRO, MAGN} respectively, $k \in \{x, y, z\}$ is the measurement axis symbol, $c \in \mathcal{A}$ is the activity symbol, and $n \in \{1, 2, \ldots, 7500\}$ is the discrete-time index.

3 Mutual Information

Entropy is a measure of the information contained in a random variable in terms of its uncertainty. The entropy of a continuous random variable $\mathcal{X}$ is $H(\mathcal{X}) = -\int f_{\mathcal{X}}(x) \log_2 f_{\mathcal{X}}(x)\, dx$ bits, where $f_{\mathcal{X}}$ is the marginal probability density function (PDF) of $\mathcal{X}$. *Conditional entropy* is a measure of the uncertainty of a random variable with the knowledge of another. The conditional entropy of $\mathcal{X}$, given another continuous random variable $\mathcal{Y}$ is $H(\mathcal{X}|\mathcal{Y}) = -\int\int f_{\mathcal{X},\mathcal{Y}}(x, y) \log_2 f_{\mathcal{X}|\mathcal{Y}}(x|y)\, dx\, dy$ bits, where $f_{\mathcal{X},\mathcal{Y}}$ and $f_{\mathcal{X}|\mathcal{Y}}$ are the joint and the conditional PDFs, respectively.

Based on entropy and conditional entropy, the *mutual information* between two random variables is the reduction of the uncertainty of one random variable using the knowledge about the other. Thus, the mutual information between $\mathcal{X}$ and $\mathcal{Y}$ is given by $\mathcal{I}(\mathcal{X}, \mathcal{Y}) = \mathcal{H}(\mathcal{X}) - \mathcal{H}(\mathcal{X}|\mathcal{Y}) = \mathcal{H}(\mathcal{Y}) - \mathcal{H}(\mathcal{Y}|\mathcal{X})$ bits. $\mathcal{I}(\mathcal{X}, \mathcal{Y})$ is bounded by zero and $\min\{H(\mathcal{X}), H(\mathcal{Y})\}$, therefore, it is a scalable measure. Mutual information is also the *Kullback-Leibler distance* between the joint PDF of the random variables and the product of their marginal PDFs. For continuous random variables $\mathcal{X}$ and $\mathcal{Y}$,

$$\mathcal{I}(\mathcal{X}, \mathcal{Y}) = \int\int f_{\mathcal{X},\mathcal{Y}}(x, y) \log_2 \frac{f_{\mathcal{X},\mathcal{Y}}(x, y)}{f_{\mathcal{X}}(x)\, f_{\mathcal{Y}}(y)} \,\mathrm{dx}\,\mathrm{dy}. \tag{1}$$

Consequently, mutual information becomes a measure of the dependence between two random variables, in terms of the information that one random variable provides about another [14]. Mutual information can be also used to measure the correlation between two random variables, instead of computing the linear correlation coefficient between them. The benefits of using mutual information are as follows [15]:

- Mutual information allows us to compute the correlation between multi-variate random vectors with either the same or different dimensions.
- Since mutual information depends on PDFs, it is not affected by the range of observed values of random variables.
- Mutual information can capture linear and non-linear dependencies between random variables.

4 Methodology

To determine the most informative measurement axes, sensor types, and sensor locations for human activity recognition, we employ the mutual information criterion, which is suitable for computing the dependence between human activities and any combination of measurement types, including acceleration, rotational speed, and magnetic field strength.

We define the human activity classes as a discrete random variable, $\mathcal{C} \in \mathcal{A}$ where each activity in $\mathcal{A}$ is represented by a distinct symbol. $\mathcal{C}$ is uniformly distributed such that $\Pr\{\mathcal{C} = c\} = \frac{1}{|\mathcal{A}|}$ where $|\mathcal{A}| = 19$. In the following subsections, we compute the mutual information between $\mathcal{C}$ and the measurements, grouped in different ways.

4.1 Determining Informative Measurement Axes

In this section, we compute the amount of activity-related information provided by each measurement axis of the sensors on the subject's body. For this purpose, we represent the output of each measurement axis with a distinct continuous random variable $\mathcal{M}_{ijk} \in \Re$. For each subject in the dataset, we define the mutual information between $\mathcal{C}$ and $\mathcal{M}_{ijk}$ according to Eq. (1) as

$$\mathcal{I}_l\left(\mathcal{M}_{ijk}, \mathcal{C}\right) = \sum_{c \in \mathcal{A}} \int_{m_{ijk} \in \Re} f_{\mathcal{M}_{ijk},\mathcal{C}}\left(m_{ijk}, c\right) \log_2 \frac{f_{\mathcal{M}_{ijk},\mathcal{C}}\left(m_{ijk}, c\right)}{f_{\mathcal{M}_{ijk}}\left(m_{ijk}\right) \Pr\{\mathcal{C} = c\}} \, dm_{ijk} \tag{2}$$

where $l \in \{1, 2, \ldots, 8\}$ is the subject index.

Since the different activity types are independent of each other, there is no dependence between the values of $\mathcal{C}$, and $f_{\mathcal{M}_{ijk},\mathcal{C}}\left(m_{ijk}, c\right) = \left\{\bigcup f_{\mathcal{M}_{ijk}|\mathcal{C}}\left(m_{ijk}|c\right) : c \in \mathcal{A}\right\}$. Here, $\hat{f}_{\mathcal{M}_{ijk}|\mathcal{C}}$, a histogram-based non-parametric estimate of $f_{\mathcal{M}_{ijk}|\mathcal{C}}$, is computed based on the corresponding observation sequence $U_{ijkc}[n]$. Then, $f_{\mathcal{M}_{ijk}}$ is estimated as $\hat{f}_{\mathcal{M}_{ijk}}\left(m_{ijk}\right) = \sum_{c \in \mathcal{A}} \hat{f}_{\mathcal{M}_{ijk}|\mathcal{C}}\left(m_{ijk}|c\right)$. Consequently, $\mathcal{I}_l\left(\mathcal{M}_{ijk}, \mathcal{C}\right)$ is approximated by a Riemann sum as follows:

$$\mathcal{I}_l\left(\mathcal{M}_{ijk}, \mathcal{C}\right) \approx \sum_{c \in \mathcal{A}} \sum_{m_{ijk} \in M_{ijk}} \hat{f}_{\mathcal{M}_{ijk},\mathcal{C}}\left(m_{ijk}, c\right) \log_2 \frac{\hat{f}_{\mathcal{M}_{ijk},\mathcal{C}}\left(m_{ijk}, c\right)}{\hat{f}_{\mathcal{M}_{ijk}}\left(m_{ijk}\right) \Pr\{\mathcal{C} = c\}} \, \delta m_{ijk} \tag{3}$$

where M_{ijk} is the set of histogram bin centre points and δm_{ijk} is the sufficiently small histogram bin width over the range of $U_{ijkc}[n]$ for $c \in \mathcal{A}$.

For each measurement axis, the mean, the highest, and the lowest mutual information computed over all subjects are plotted in Fig. 2a. According to the figure, it

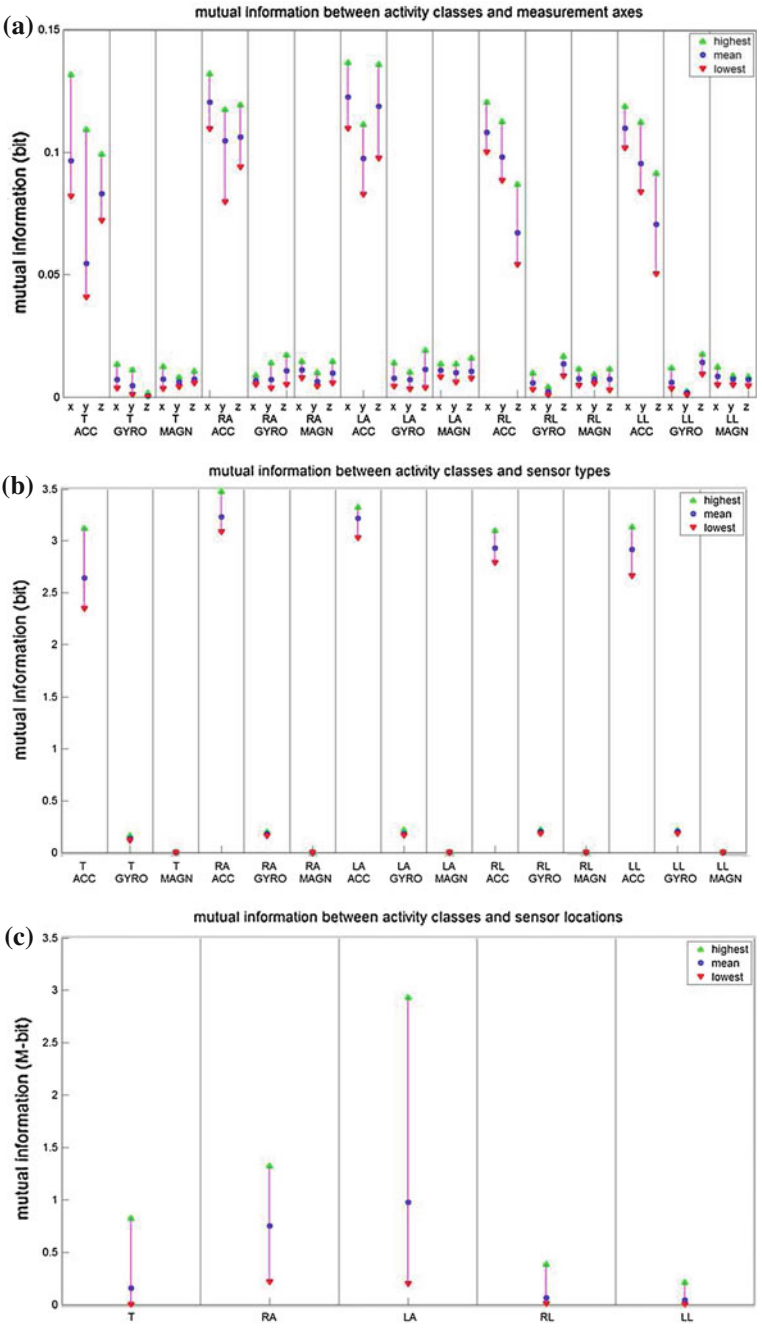

Fig. 2 The mean, the highest, and the lowest mutual information between the human activities and each **a** measurement axis, **b** sensor type, and **c** sensor location (*T* torso, *RA* right arm, *LA* left arm, *RL* right leg, *LL* left leg, *ACC* accelerometer, *GYRO* gyroscope, *MAGN* magnetometer)

is observed that at all sensor locations, the measurement axes of the accelerometers provide more activity-related information than those of gyroscopes and magnetometers. Furthermore, the x-axis of the accelerometers, which is perpendicular to the ground, is observed to contain higher activity-related information than the other two axes of the accelerometers, at all sensor locations.

4.2 Determining Informative Sensor Types

In this section, we compute the amount of activity-related information provided by each sensor type on the subject's body. Here, *sensor type* indicates the combination of the three measurement axes that belong to the same device (i.e., accelerometer, gyroscope, magnetometer) at the same sensor location. We represent each sensor type with a distinct continuous random vector $\mathcal{T}_{ij} = \left[\mathcal{M}_{ijx}\ \mathcal{M}_{ijy}\ \mathcal{M}_{ijz}\right]^{\mathrm{T}} \in \Re^3$, where $\{x, y, z\}$ are the symbols of the measurement axes that belong to the sensor type j at sensor location i. Based on Equation (1), the mutual information between $\mathcal{C}$ and $\mathcal{T}_{ij}$ for the lth subject in the dataset is given by:

$$\mathcal{I}_l\left(\mathcal{T}_{ij}, \mathcal{C}\right) = \sum_{c \in \mathcal{A}} \int\limits_{t_{ij} \in \Re^3} f_{\mathcal{T}_{ij},\mathcal{C}}\left(t_{ij}, c\right) \log_2 \frac{f_{\mathcal{T}_{ij},\mathcal{C}}\left(t_{ij}, c\right)}{f_{\mathcal{T}_{ij}}\left(t_{ij}\right) \Pr\{\mathcal{C} = c\}} \mathrm{dt_{ij}} \tag{4}$$

Since the different activities are independent of each other, $f_{\mathcal{T}_{ij},\mathcal{C}}\left(t_{ij}, c\right) = \left\{\bigcup f_{\mathcal{T}_{ij}|\mathcal{C}}\left(t_{ij}|c\right) : c \in \mathcal{A}\right\}$. Here, $\hat{f}_{\mathcal{T}_{ij}|\mathcal{C}}$, a histogram-based non-parametric estimate of $f_{\mathcal{T}_{ij}|\mathcal{C}}$, is computed based on the corresponding observation sequences $U_{ijxc}[n]$, $U_{ijyc}[n]$, and $U_{ijzc}[n]$. Then, $f_{\mathcal{T}_{ij}}$ is estimated as $\hat{f}_{\mathcal{T}_{ij}}\left(t_{ij}\right) = \sum_{c \in \mathcal{A}} \hat{f}_{\mathcal{T}_{ij}|\mathcal{C}}\left(t_{ij}|c\right)$. Using these estimates, $\mathcal{I}_l\left(\mathcal{T}_{ij}, \mathcal{C}\right)$ is approximated by a Riemann sum as follows:

$$\mathcal{I}_l\left(\mathcal{T}_{ij}, \mathcal{C}\right) \approx \sum_{c \in \mathcal{A}} \sum_{t_{ij} \in T_{ij}} \hat{f}_{\mathcal{T}_{ij},\mathcal{C}}\left(t_{ij}, c\right) \log_2 \frac{\hat{f}_{\mathcal{T}_{ij},\mathcal{C}}\left(t_{ij}, c\right)}{\hat{f}_{\mathcal{T}_{ij}}\left(t_{ij}\right) \Pr\{\mathcal{C} = c\}} \Delta t_{ij} \tag{5}$$

where $T_{ij} = \left(M_{ijx} \times M_{ijy} \times M_{ijz}\right)$ is the vector representing the set of histogram bin centres and $\Delta t_{ij} = \left(\delta m_{ijx} \cdot \delta m_{ijy} \cdot \delta m_{ijz}\right)$ is the histogram bin volume with sufficiently small histogram bin widths δm_{ijx}, δm_{ijy}, and δm_{ijz} over the range of $U_{ijxc}[n]$, $U_{ijyc}[n]$, and $U_{ijzc}[n]$ for $c \in \mathcal{A}$, respectively.

For each sensor type, the mean, the highest, and the lowest mutual information computed over all subjects are plotted in Fig. 2b. According to the figure, it is again observed that at all sensor locations, accelerometers provide considerably more activity-related information than gyroscopes and magnetometers. The information content of gyroscope measurements are slightly higher than those of magnetometers at all sensor locations.

4.3 Determining Informative Sensor Locations

In this section, we compute the amount of activity-related information provided by each of the five sensor locations on the subject's body. Here, *sensor location* indicates the combination of the three sensor types at the same location. We represent each sensor location by a distinct continuous random vector $\mathcal{S}_i = [\mathcal{T}_{i1}\ \mathcal{T}_{i2}\ \mathcal{T}_{i3}]^{\mathrm{T}} \in \Re^9$, where $\{1, 2, 3\}$ are the indices of the sensor types at location i. For the lth subject in the dataset, the mutual information between $\mathcal{C}$ and $\mathcal{S}_i$ is given by:

$$\mathcal{I}_l\left(\mathcal{S}_i, \mathcal{C}\right) = \sum_{c \in \mathcal{A}} \int\limits_{s_i \in \Re^9} f_{\mathcal{S}_i,\mathcal{C}}\left(s_i, c\right) \log_2 \frac{f_{\mathcal{S}_i,\mathcal{C}}\left(s_i, c\right)}{f_{\mathcal{S}_i}\left(s_i\right) \Pr\left\{\mathcal{C} = c\right\}} \mathrm{d}s_i \tag{6}$$

Because of the independence between the activities, $f_{\mathcal{S}_i,\mathcal{C}}\left(s_i, c\right) = \{\bigcup f_{\mathcal{S}_i|\mathcal{C}}\left(s_i|c\right) : c \in \mathcal{A}\}$. The $\hat{f}_{\mathcal{S}_i|\mathcal{C}}$ is a histogram-based non-parametric estimate of $f_{\mathcal{S}_i|\mathcal{C}}$ and computed based on the corresponding observation sequences $U_{i1xc}\left[n\right], U_{i1yc}\left[n\right], \ldots, U_{i3zc}\left[n\right]$. Then, the estimate of $f_{\mathcal{S}_i}$ is given by $\hat{f}_{\mathcal{S}_i}\left(s_i\right) = \sum\limits_{c \in \mathcal{A}} \hat{f}_{\mathcal{S}_i|\mathcal{C}}\left(s_i|c\right)$ and $\mathcal{I}_l\left(\mathcal{S}_i, \mathcal{C}\right)$ is approximated by a Riemann sum as follows:

$$\mathcal{I}_l\left(\mathcal{S}_i, \mathcal{C}\right) \approx \sum_{c \in \mathcal{A}} \sum_{s_i \in S_i} \hat{f}_{\mathcal{S}_i,\mathcal{C}}\left(s_i, c\right) \log_2 \frac{\hat{f}_{\mathcal{S}_i,\mathcal{C}}\left(s_i, c\right)}{\hat{f}_{\mathcal{S}_i}\left(s_i\right) \Pr\left\{\mathcal{C} = c\right\}} \Delta s_i \tag{7}$$

where $S_i = (T_{i1} \times T_{i2} \times T_{i3})$ is the vector representing the set of histogram bin centres and $\Delta s_i = (\Delta t_{i1} \cdot \Delta t_{i2} \cdot \Delta t_{i3})$ is the histogram bin volume. Because of memory allocation problems that arise when an enormous number of very small histogram bins are handled in $\Re^9$, the histogram bins in the estimation of $f_{\mathcal{S}_i|\mathcal{C}}$ cannot be set as small as in the estimations of $f_{\mathcal{M}_{ijk}|\mathcal{C}}$ and $f_{\mathcal{T}_{ij}|\mathcal{C}}$. Therefore, the approximation in Eq. (7) is not as accurate as in Eqs. (3) and (5). Despite this, the results are admissible for the comparison between the sensor locations.

For each sensor location, the mean, the highest, and the lowest mutual information computed over all subjects are plotted in Fig. 2c. According to the figure, it is observed that the arms provide the highest activity-related information, whereas the legs provide the lowest.

5 Conclusion

In this study, we employ a human activity dataset comprised of inertial and magnetic sensor measurements of 19 types of daily and sports activities and investigate the useful sensor types, sensor locations, and measurement axes of body-worn devices that provide high activity-related motion information.

In many human activity recognition studies, the measurements acquired from all sensors in the configuration are considered with equal significance. However,

the results of this study indicate that the sensors do not contribute equal amount of information during the activity recognition process. The results of the proposed analyses can be used to determine the level of significance of each sensor.

Based on the mutual information criterion, we identify the linear acceleration measurements at all sensor locations as the most informative measurement type. Among the linear acceleration measurements, the measurement axes along the direction perpendicular to the ground are more informative than the others. In terms of sensor location, we identify the arms and the torso as the first and the second most informative locations, respectively. The legs are less informative compared to the extremities. The mutual information based approach proposed in this study can be used in selecting the most suitable sensor configuration among a set of possibilities.

Future cognitive systems are expected to be able to adapt to recent states, make estimates, and optimize their operating conditions autonomously. In parallel with this forecast, we envision a system which employs body-worn sensors that will be able to activate the informative sensors and suspend the others while simultaneously making activity decisions. In this way, the inputs to the recognition unit will be simplified to improve its decision accuracy. This study constitutes the first step of our research on such adaptive human activity recognition systems. In future work, our aim is to complete the relevance and redundancy relationships between sensor-feature-activity class trio.

References

1. Kim E, Helal S, Cook DJ (2010) Human activity recognition and pattern discovery. IEEE Pervas Comput (1):48–53
2. Preece SJ, Goulermas JY, Kenney LPJ, Howard D, Meijer K, Crompton R (2009) Activity identification using body-mounted sensors-a review of classification techniques. Physiol Meas 30(4):R1–R33
3. Turaga PK, Chellappa R, Subrahmanian VS, Udrea O (2008) Machine recognition of human activites: a survey. IEEE T Circ Syst Vid 18(11):1473–1488
4. Altun K, Barshan B (2012) Pedestrian dead reckoning employing simultaneous activity recognition cues. Meas Sci Technol 23(2):025103
5. Töreyin BU, Dedeoğlu Y, Çetin AE (2005) HMM based falling person detection using both audio and video. In: Sebe N, Lew M, Huang T (eds) Computer Vision in Human-Computer Interaction. Lecturer Notes Computational Science, Springer, Berlin, pp 211–220
6. Ganyo M, Dunn M, Hope T (2011) Ethical issues in the use of fall detectors. Ageing Soc 31(8):1350–1367
7. Preece SJ, Goulermas JY, Kenney LPJ, Howard D (2009) A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. IEEE T Bio-med Eng 56(3):871–879
8. Altun K, Barshan B (2010) Human activity recognition using inertial/magnetic sensor units. In: Salah A, Gevers T, Sebe N, Vinciarelli A (eds) Human Behavior Understanding. Lecture Notes Computational Science, Springer, Berlin, pp 38–51
9. Atallah L, Lo B, King RC, Gitang G-Z (2011) Sensor positioning for activity recognition using wearable accelerometers. IEEE T Bio-med Circ Syst 5(4):320–329

10. Fish B, Khan A, Chehade NH, Chien C, Pottie G (2012) Feature selection based on mutual information for human activity recognition. 2012 IEEE International Conference Acoustics Speech. pp 1729–1732
11. Yüksek MC (2011) A comparative study on human activity classification with miniature inertial and magnetic sensors, MSc. thesis, Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey
12. Altun K, Barshan B, Tunçel O (2010) Comparative study on classifying human activities with miniature inertial and magnetic sensors. Pattern Recogn 43(10):3605–3620
13. Xsens Technologies BV (2010) MTi and MTx user manual and technical documentation, Document MT0100P, Revision O
14. Cover TM, Thomas JA (1991) Elements of Information Theory. Wiley, New York
15. Li W (1990) Mutual information functions versus correlation functions. J Stat Phys 60(5–6):823–837

Routing Emergency Evacuees with Cognitive Packet Networks

Huibo Bi, Antoine Desmet and Erol Gelenbe

Abstract Providing optimal and safe routes to evacuees in emergency situations requires fast and adaptive algorithms. The common approaches are often too slow to converge, too complex, or only focus on one aspect of the problem, e.g. finding the shortest path. This paper presents an adaptation of the Cognitive Packet Network (CPN) concept to emergency evacuation problems. Using Neural Networks, CPN is able to rapidly explore a network and allocate overhead in proportion to the perceived likelihood of finding an optimal path there. CPN is also flexible, as it can operate with any user-defined cost function, such as congestion, path length, safety, or even compound metrics. We compare CPN with optimal algorithms such as Dijkstra's Shortest Path using a discrete-event emergency evacuation simulator. Our experiments show that CPN reaches the performance of optimal path-finding algorithms. The resulting side-effect of such smart or optimal algorithms is in the greater congestion that is encountered along the safer paths; therefore we indicate how the quality of service objective used by CPN can also be used to avoid congestion for further improvements in evacuee exit times.

1 Introduction

Emergency evacuation of large areas or buildings [1–3] is effectively a complex and transient transshipment problem, from multiple sources to multiple destinations, where edge capacity varies based on the presence and intensity of hazards. Finding

H. Bi (✉) · A. Desmet · E. Gelenbe
Intelligent Systems and Networks Group Department of Electrical and Electronic Engineering, Imperial College London, London, England
e-mail: huibo.bi12@imperial.ac.uk

A. Desmet
e-mail: a.desmet10@imperial.ac.uk

F. Gelenbe
e-mail: e.gelenbe@imperial.ac.uk

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_29,

dynamic and real-time solutions to this type of problem requires a highly reactive and adaptive system due to the fast-changing nature of the environment [4–11].

Indeed, not only are the threats which trigger the evacuation themselves dynamic, but in addition to this, the flows of evacuees in the building generally tend to be unstable. For instance, the decision to send evacuees down what appears to be the safest path may lead to a sudden increase in congestion, which could ultimately create a deadly stampede situation. On the other hand, a coarse-grained graph representation of a 3-storey building can contain up to 300 nodes and twice as many edges, making it virtually impossible to resolve and monitor *every* possible egress path in the building.

Owing to the size of the building graphs, using an algorithm such as Dijkstra's each time the conditions change can be computationally expensive. Furthermore, these algorithms focus all efforts on finding the *optimal* solution, instead of considering a collection of safe paths for optimal transshipment. Spanning-Tree algorithms [12] support distributed resolution of routes; however they only focus on one single optimal path and the convergence speed may also hinder performance in scenarios where the environment constantly changes. Opportunistic communications can help users self-determine their evacuation path [13, 14], but also suffers from a "hoarding" effect since most users only have access to the same limited amount of information, which leads them to collectively follow what they perceive as being the shortest path. Unlike techniques mentioned previously which search for one optimal path, algorithms inspired from transshipment theory [15, 16] handle capacity-constrained links and maximize the use of *all* available paths. In particular, time-expanded graphs provide exact solutions to transhipment problems, but become prohibitively complex for large graphs, since the complexity not only depends on the scale of the network, but also the "time horizon" which in turn determines the time-expansion of the graph. Finally, the traffic forecasting solution proposed by Lu et al. [17, 18] also finds routes which maximize the flow of evacuees, but the algorithm cannot handle dynamic hazards: this would modify the availability and capacity of each edge and disrupt the algorithm's forecasted congestion time-series.

We believe that an ideal routing algorithm for evacuation purposes would not only be decentralized, but also able to self-monitor and constantly make swift adjustments instead of requiring a new convergence process or a full graph search each time the environment changes. This algorithm should be able to optimize the evacuation process not only in terms of shortest evacuation path, but perhaps combine other metrics such as safety, congestion, simplicity of the route or more. Such "self-aware" or nature inspired [19] routing algorithms have been developed for computer network packet routing [20] using the Random Neural Network [21] to construct the routing algorithm. The Cognitive Packet Network (CPN) [22] algorithm is an example of an autonomic communication system [23] which is suggested in this paper as a means to route *evacuees* rather than as a means to forward network packets.

CPN is designed to optimize any measurable Quality of Service (QoS) metric such as delay minimisation [24] and energy optimisation [25, 26] using reinforcement learning [27, 28] with a recurrent Random Neural Network [21]. It has been demonstrated against network disruptions such as "Worm-like" network attacks [29], where parts of the network suddenly become unavailable. Similarities exist between

computer network packet routing and evacuee routing, in particular, the objectives: to find the best path(s) across a graph with respect to some measurable metric. Evacuees can be seen as data packets; paths and places within the area can be likened to routers and links which also create delays or become unavailable. Thus, in this paper we present an adaptation of the CPN concept to the emergency evacuation problem, and evaluate its performance by simulating a building evacuation.

In the following section, we first present the core concept of CPN and then discuss how we apply it to the emergency evacuation problem. The next section presents the simulation model used to evaluate CPN as an evacuee routing algorithm. Finally, simulation results are analyzed, and conclusions are drawn regarding the effectiveness of an evacuee assistance system based on the CPN routing algorithm.

2 The Cognitive Packet Network

The Cognitive Packet Network concept aims at solving the problems experienced by large and fast-changing networks, where the convergence time of "overall" routing schemes eventually becomes slower than the rate at which the network conditions change, thus resulting in a constant lag which hinders performance. CPN alleviates this issue by letting each network node send a small flow of "Smart Packets" (SP) which are dedicated to network condition monitoring and new route discovery. Every node in the network hosts a Random Neural Network (RNN) which is used to direct incoming SPs towards their next hop. SPs are also allowed to "drift" away from the RNN's recommendation and explore new paths randomly—in a manner comparable to "ant colony" algorithms. Each time an SP reaches its intended destination, an acknowledgement (ACK) message containing all measurements made by the SP travels back to the source along the original path—with loops removed—and updates all nodes visited with fresh measurements. Every node along the path uses the measurements carried by SP ACKs to adjust their RNN by performing Reinforcement Learning (RL), they also store this information in a table used to source-route the payload-carrying Dumb Packets (DP). This routing table stores a fixed amount of optimal paths discovered by SPs and allows a node to instantly switch to an alternate path if ACKs indicate that the best path is no longer optimal. Unlike other algorithms which query each node in the network or perform exhaustive graph searches, CPN uses its RNN to intelligently allocate its routing overhead based on how worthwhile a path appears to be: the best path(s) are regularly monitored, while proportionally less bandwidth is allocated to paths which show less potential. A detailed presentation and overview of CPN performance can be found in [22].

Applying the CPN concept to emergency evacuation problems can only be achieved if two main requirements are fulfilled:

- A graph-based representation of the area is available, where edges represent paths, and nodes represent physical areas in the building. This graph contains information

on distances, edge and node capacities, and the presence of any relevant *static* hazards, such as stairs, narrow passages, uneven terrain, etc.
- The area to evacuate is covered by a sensor network which monitors any *dynamic* hazard relevant to the scenario. Typical hazards include: fire, smoke, water, congestion, restricted visibility, etc.

In the context of emergency evacuations, the CPN's DPs are effectively the evacuees; while SPs reside in the application's server and "virtually" explore the area, collecting information from the graph and sensors found along the way. This information is then processed into a cost value which is used to compare paths, and CPN ultimately aims at finding the route which exhibits the lowest cost. An advantage of CPN is that its process is independent of the cost function chosen. It can be *any* measurable function, from the shortest and safest path, to functions factoring in the "simplicity" of the path, congestion, or weighed more towards safety, at the cost of increased distance.

3 Simulation Model and Experiment

We use the Distributed Building Evacuation Simulator (DBES), a Discrete-Event Simulator (DES) to evaluate the effectiveness of CPN for evacuee routing in fire-related emergency evacuations. The area to evacuate is a building, based on the three lower floors of Imperial College London's EEE building. Each floor in this building has a surface area of approximately 1,000 m^2. In order to run the CPN algorithm, a coarse graph representation of this building is made (Fig. 1).

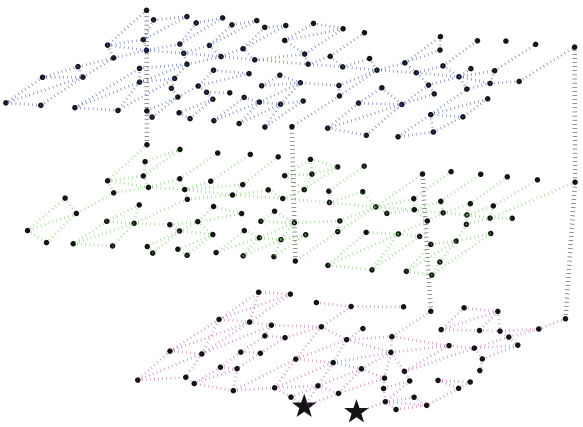

Fig. 1 Graph representation of the building model. The two black stars on the ground floor mark the position of the building's exits

Effective length is a cost function which compounds path length with hazard intensity along each edge. For a given path p composed of a collection of edges V, the cost G is:

$$G_{t,p} = \sum_{V} l(v) \cdot f(v,t) \tag{1}$$

where *l(v)* is the length of the edge, and *f(v, t)* is the fire intensity on the edge v at time t, so that:

$$f(v,t) = \begin{cases} \gg \text{ average edge length} & \text{if the edge is affected by fire,} \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

This ensures that the effective length of a path exposed to fire will always be greater than any other *safe* path in the building.

Our experiments feature three different scenarios for comparison purposes. The first scenario, referred to as "autonomous", simulates cases where evacuees do not receive any assistance: they simply evacuate through the shortest path, and keep trying a different one if they are blocked by the fire—until they either evacuate or perish. The two other scenarios provide evacuees with individual assistance: each time the fire spreads, Dijkstra's Shortest Path algorithm is executed from each node, and determines the *optimal* path, with respect to *effective length*. The third scenario is akin to the previous one, however the decision engine uses CPN. The results of ten randomized simulations are presented in the following section. The fire outbreak is located on the first floor in a position which is not immediately threatening to any user. Yet as the fire expands, it quickly reaches one of the main evacuation pathways and prompts a major re-routing of all evacuees.

4 Results

Before running full-scale randomized simulations, we experimented with the CPN parameters, and in particular with those related to Smart Packets. At the beginning of each simulation, we allow each node to send SPs to provide their RNN with basic training. We noticed that 5 SPs per node is enough for them to resolve at least one egress path. More than 50 % of the nodes are able to find the shortest path after 10 SP/node. These are mostly nodes located along the main paths, which take advantage of the numerous SP ACKs transiting through them; while the path length from "leaf" nodes are generally below 120 % of the corresponding shortest path. We also found that the Drift Parameter—an SP's probability to choose the next hop at random over the RNN's advice—has an impact on the quality of the routes found by CPN. The first path found by an SP provides the very first reinforcement learning to the RNN's neurons. This path is generally not optimal: the first SPs effectively travel at random since the RNN is not trained yet. With very low drift rates, subsequent SPs will be

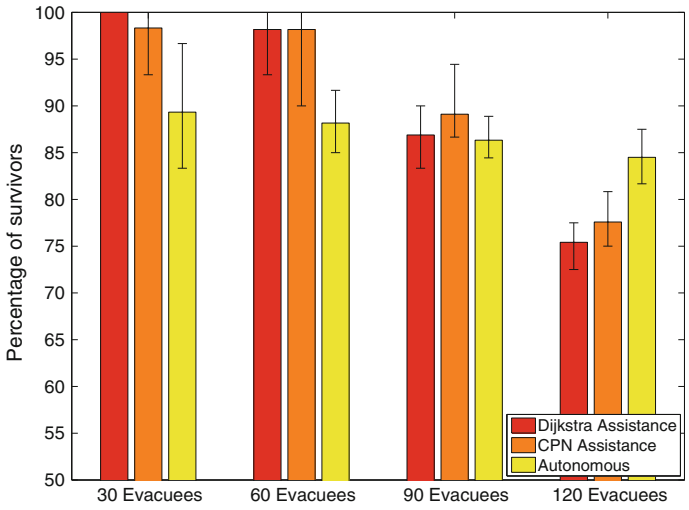

Fig. 2 Percentage of survivors for each scenario. The results are the average of 10 randomized simulation runs, and error bars shows the min/max result in any of the 10 simulation runs.

virtually unable to explore anything beyond this initial path that has set up the RNN's neurons. As a result, the same sub-optimal path is visited over and over and the same neurons are always reinforced, which results in overtraining. While low drift rates let CPN quickly find an exit path for each node, the routing performance soon ceases to increase and may remain far below optimal. On the other hand, allowing the SPs to constantly drift leads to very slow set-up times, but all shortest path will eventually be discovered since the SPs effectively perform a random walk.

Figure 2 presents the results of ten randomized experiments. The bars show the percentage of successful evacuees, and the error bars show the single highest and lowest values found in any of the ten iterations. The experiments with lower densities (30 and 60 evacuees) clearly show that CPN reaches the performance level of the Dijkstra's SP algorithm. The results show that approximately 10 % of the "autonomous" users die. These evacuees discover that their initial escape path is blocked by the fire, and while they backtrack the fire has time to spread to all exits, effectively trapping them inside the building. On the other hand, evacuees which benefit from the assistance systems receive an early warning that their path is dangerous and re-route early enough to escape the building alive. As the density of evacuees increases (90–120 evacuees), the assistance systems progressively lose their advantage, and become clearly detrimental to the evacuees. This is largely due to congestion forming in the building, as can be seen on Table 1. The Dijkstra's algorithm tends to simultaneously send all evacuees through one single optimal path, which creates a synchronized rush and results in the highest levels of congestion. Because CPN is decentralized, each node reacts to the fire and updates its path recommendation at a different time, and this has a tendency to disperse users which in turn slightly reduces the congestion.

Table 1 Average number of nodes traversed by the evacuees that were seen to have a *non-zero queueing time that indicates some level of congestion*, for different path finding algorithms. We see (from left to right) that as the path finding algorithm gets closer to the optimal, the level of congestion increases since more of the evacuees follow the same best evacuation paths

Number of evacuees	Autonomous paths	CPN based paths	Dijkstra based paths
30	3.0	3.5	3.5
60	10.3	12.4	13.0
90	18.6	21	22.3
120	27.8	30.2	31.7

Interestingly, the CPN assistance system is able to switch *before* the shortest path becomes affected by fire. This is due to the fact that SPs not only visit the shortest path, but also drift alongside it. Such drifting SPs are exposed to the fire surrounding the shortest path; and since these SPs return with higher cost values, the RL punishes the corresponding neuron. Finally, the scenario where users are left to evacuate on their own exhibits a large "entropy" amongst users, which means that the hallways within the building are less subject to congestion and evacuees can travel faster. In summary, shortest path algorithms guide evacuees to exits effectively but with the side effect of causing congestion. In low density, congestion is negligible and Dijkstra's algorithm, which calculates the optimal solution, gives the best result. As the density increases, congestion will offset the advantage of travelling over the shortest paths. Instead, non-optimal algorithms, which scatter users across multiple paths, create less congestion and improve the outcome. Table 1 illustrates this.

5 Conclusions

In this paper we have proposed an adaptation of CPN, a Self-Aware computer network routing protocol, to the problem of emergency building evacuation. We have proven that CPN is able to reach performance levels which are on par with those of *optimal* algorithms. Simulations also reveal that using the combined safety and length of an egress path as metric is ineffective in high-density scenarios. In these cases, it appears that reducing congestion and distributing the flow along every available path (even those which constitute a detour) is a more effective strategy. Future research will be directed in this area: we will define a new routing metric that represents a path's travel time, taking congestion into account. While we have proved the effectiveness of CPN in this paper, our next step in this research project will be to assess the complexity of CPN against other algorithms. Of particular interest is the *time-to-live* and quantity of SPs required to reach the performance levels of algorithms which perform full graph searches at each step, since these two parameters widely contribute to CPN's complexity.

References

1. Fischer C, Gellersen H (2009) Location and navigation support for emergency responders: a survey. IEEE Pervas Comput 9(1):38–47
2. Gelenbe E, Wu FJ (2012) Large scale simulation for human evacuation and rescue. Comput Math Appl 64(12):3869–3880. doi:10.1016/j.camwa.2012.03.056
3. Malan DJ, Fulford-Jones TR, Nawoj A, Clavel A, Shnayder V, Mainland G, Welsh M, Moulton S (2004) Sensor networks for emergency response: challenges and opportunities. IEEE Pervas Comput 3(4):16–23
4. Chen D, Mohan CK, Mehrotra KG, Varshney PK (2010) Distributed in-network path planning for sensor network navigation in dynamic hazardous environments. Wireless Comm Mob Comput 12:739
5. Chen PY, Chen WT, Shen YT (2008) A distributed area-based guiding navigation protocol for wireless sensor networks. In: IEEE international conference on parallel and distributed systems, pp 647–654
6. Chen PY, Kao ZF, Chen WT, Lin CH (2011) A distributed flow-based guiding navigation protocol in wireless sensor networks. In: International conference on parallel processing, pp 105–114
7. Chen WT, Chen PY, Wu CH, Huang CF (2008) A load-balanced guiding navigation protocol in wireless sensor networks. In: IEEE Global telecommunication conference, pp 1–6
8. Li M, Liu Y, Wang J, Yang Z (2009) Sensor network navigation without locations. In: IEEE INFOCOM, pp 2419–2427
9. Li Q, Rosa MD, Rus D (2003) Distributed algorithms for guiding navigation across a sensor network. In: ACM international conference mobile computing and networking, pp 313–325
10. Tseng YC, Pan MS, Tsai YY (2006) Wireless sensor networks for emergency navigation. IEEE Comput 39(7):55–62
11. Pan MS, Tsai CH, Tseng YC (2006) Emergency guiding and monitoring applications in indoor 3D environments by wireless sensor networks. Int J Sensor Networks 1(1/2):2–10
12. Dimakis N, Filippoupolitis A, Gelenbe E (2010) Distributed building evacuation simulator for smart emergency management. Comput Jl 53(9):1384–1400
13. Gelenbe E, Görbil G (2011) Opportunistic communications for emergency support systems. Procedia Comput Sci 5:39–47
14. Gorbil G, Filippoupolitis A, Gelenbe E (2011) Intelligent navigation systems for building evacuation. Comput Inf Sci Lecture Notes Electr Eng
15. Hoppe B, Tardos É (1995) The quickest transshipment problem. In: Proceedings of the 6th annual ACM-SIAM symposium on discrete algorithms, Society for Industrial and, Applied Mathematics, pp 512–521
16. Hamacher HW, Tjandra SA (2002) Mathematical modelling of evacuation problems – a state of the art. In: Schreckenberg M, Sharma SD (eds) Pedestrian and evacuation dynamics. Springer, Berlin, pp 227–266
17. Lu Q, George B, Shekhar S (2005) Capacity constrained routing algorithms for evacuation planning: a summary of results. In: Bauzer Medeiros C, Egenhofer M, Bertino E (eds) Advances in spatial and temporal databases, lecture notes in computer science, vol 3633. Springer, Berlin, pp 291–307
18. Lu Q, Huang, Y, Shekhar S: Evacuation planning: A capacity constrained routing approach. In: WeiThooYue MGA, Chen H (eds) Intelligence and security informatics. Springer, Berlin, pp 111–125
19. Gelenbe E (2012) Natural computation. Comput J 55(7):848–851
20. Gelenbe E (2004) Cognitive packet network. U.S. Patent 6,804,201
21. Gelenbe E (1993) Learning in the recurrent random neural network. Neural Comput 5(1): 154–164. doi:10.1162/neco.1993.5.1.154
22. Gelenbe E (2009) Steps towards self-aware networks. Commun ACM 52:66–75

23. Dobson S, Denazis S, Fernández A, Gaiti D, Gelenbe E, Massacci F, Nixon P, Saffre F, Schmidt N, Zambonelli F (2006) A survey of autonomic communications. ACM Trans Auton Adapt Syst 1(2):223–259 (http://doi.acm.org/10.1145/1186778.1186782)
24. Gelenbe E, Lent R, Nunez A (2004) Self-aware networks and qos. Proc IEEE 92(9):1478–1489
25. Gelenbe E, Lent R (2004) Power-aware ad hoc cognitive packet networks. Ad Hoc Netw 2(3):205–216
26. Gelenbe E, Morfopoulou C (2010) A framework for energy aware routing in packet networks. The Computer Journal 54(6):850–859. doi:0.1093/comjnl/bxq092 (first published online: December 15, 2010)
27. Halici U (2000) Reinforcement learning with internal expectation for the random neural network. Eur J Oper Res 126(2):288–307. doi:10.1016/S0377-2217(99)00479-8
28. Gelenbe E, Şeref E, Xu Z (2001) Simulation with learning agents. Proc IEEE 89(2):148–157. doi:10.1109/5.910851
29. Sakellari G, Gelenbe E (2010) Demonstrating cognitive packet network resilience to worm attacks. In: Proceedings of the 17th ACM conference on computer and communications security, ACM, pp 636–638

Detection and Evaluation of Physical Therapy Exercises by Dynamic Time Warping Using Wearable Motion Sensor Units

Aras Yurtman and Billur Barshan

Abstract We develop an autonomous system that detects and evaluates physical therapy exercises using wearable motion sensors. We propose an algorithm that detects all the occurrences of one or more template signals (representing exercise movements) in a long signal acquired during a physical therapy session. In matching the signals, the algorithm allows some distortion in time, based on dynamic time warping (DTW). The algorithm classifies the executions in one of the exercises and evaluates them as correct/incorrect, giving the error type if there is any. It also provides a quantitative measure of similarity between each matched execution and its template. To evaluate the performance of the algorithm in physical therapy, a dataset consisting of one template execution and ten test executions of each of the three execution types of eight exercises performed by five subjects is recorded, having a total of 120 and 1,200 exercise executions in the training and test sets, respectively, as well as many idle time intervals in the test signals. The proposed algorithm detects 1,125 executions in the whole test set. 8.58 % of the 1,200 executions are missed and 4.91 % of the idle time intervals are incorrectly detected as executions. The accuracy is 93.46 % only for exercise classification and 88.65 % for simultaneous exercise and execution type classification. The proposed system may be used for both estimating the intensity of the physical therapy session and evaluating the executions to provide feedback to the patient and the specialist.

A. Yurtman (✉) · B. Barshan
Department of Electrical and Electronics Engineering, Bilkent University, TR-06800 Ankara, Bilkent, Turkey
e-mail: yurtman@ee.bilkent.edu.tr

B. Barshan
e-mail: billur@ee.bilkent.edu.tr

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_30,

1 Introduction

Physical therapy is a widely used type of rehabilitation in the treatment of patients with various disorders. The patients mostly need to repeat one or more exercise movements advised by the specialist in physical therapy sessions. In hospitals or rehabilitation centers, specialists monitor the patients and provide feedback about their performance. However, they often alternate between the patients and cannot monitor each patient continuously [2]. They cannot count the number of correct executions for each patient, and hence cannot estimate the intensity of the therapy session. They provide subjective and qualitative feedback. Moreover, once they learn how to perform them, most patients continue their physiotherapy exercises at home with no feedback at all; hence, they are likely to execute the exercises erroneously [13]. For this purpose, an autonomous system is developed to detect and evaluate physical therapy exercises using wearable motion sensor units.

2 Related Work

Several different sensor modalities are used in physical rehabilitation, including inertial, visual, strain, and medical sensors. However, many of the earlier studies are based on estimating the activity/therapy intensity [1, 6, 10] or the energy expenditure [8] using the sensors rather than determining the accuracy of the physical therapy exercises. In numerous studies, a 3-D real-time human body model is constructed to observe the movements [1, 2, 4, 11, 12, 18, 19]. However, in the previous studies, either the exercise executions are cropped manually, the subject marks each execution by pressing a button, or the subject performs each execution when s/he is informed by the system by a sound or on-screen notification. Furthermore, no idle time periods are involved in the studies evaluating the executions.

The following studies are most similar to ours:

- In myHeart neurological rehabilitation concept [5], the accuracy of arm movements are determined by applying a threshold to the result of an open-end variant of DTW, based on the signals acquired from a garment containing strain sensors. A different threshold is used for each subject, and a classification accuracy of 85 % is achieved. The system is limited to the arm movements only and wearing the garment may be difficult for some patients.
- In [13], strain sensors worn on the arm are used to provide real-time feedback to neurological patients undergoing motor rehabilitation. Seven exercises are executed by a healthy subject wearing a left-handed sensorized long-sleeve shirt both correctly and incorrectly at various speeds. The system checks whether the measured signals "match *at most once* a prefix of one of several stored references, used as templates" [13] based on the open-end DTW in order to classify and evaluate the executions. The disadvantage of the proposed system is again the difficulty of wearing the sensorized shirt.

- m-Physio platform [9] classifies the physical rehabilitation activities as correct/incorrect using accelerometers. In m-Physio, the smartphone iPhone containing a tri-axial accelerometer is mounted on the patient's leg or arm depending on the exercise he performs. Some parameters about the exercise durations and filtering of the signals are predetermined. The exercises are classified by some rules as well as by the standard DTW algorithm applied to the accelerometer signals to compare the exercise performed with the pre-recorded template. The limitation of this methodology is the need to determine the parameters for each exercise, the need to touch the iPhone screen to mark the start and end of each exercise execution, and the utilization of the sensors at one location.

Unlike these studies, our system automatically detects the executions in a therapy session and evaluates them while handling idle time periods.

3 Multi-Template Multi-Match Dynamic Time Warping

To detect the occurrences of multiple exercise templates in a recorded signal, we propose to use a novel algorithm, namely, *multi-template multi-match dynamic time warping* (*MTMM-DTW*), based on dynamic time warping (DTW), both of which are explained below. This method makes it possible to identify correct and incorrect executions of an exercise (identifying the type of error if there is any), the counting of the exercises, and their classification over different exercise types.

Dynamic time warping DTW is an algorithm that nonlinearly transforms the time axes of two signals individually such that the transformed signals are most similar to each other, where the (dis)similarity is often measured by the Euclidean distance (see Fig. 1). The minimized distance is called *DTW distance* [7]. DTW can be applied to multi-dimensional signals. The computational complexity of the DTW algorithm is $\mathcal{O}(NM)$ where N and M are the lengths of the two signals.

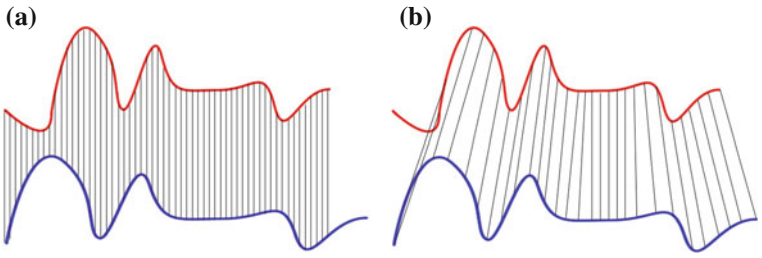

Fig. 1 Comparison of the Euclidean and DTW distance measures. **a** The Euclidean distance compares the samples at the same time instants, whereas **b** the DTW distance compares the samples that belong to similar shapes with each other to minimize the distance. Retrieved from http://upload.wikimedia.org/wikipedia/commons/6/69/Euclidean_vs_DTW.jpg

DTW algorithm can be modified to search for one signal (*template*) inside the other signal (*test signal*) by matching the template with the subsequence of the test signal that minimizes the DTW distance. This algorithm is called *subsequence DTW* [7]. Although the subsequence and the transformations of the time axes of the two signals are jointly optimized, the subsequence DTW algorithm has the same computational complexity as the standard DTW algorithm.

Multi-template multi-match DTW We develop the MTMM-DTW algorithm based on subsequence DTW to detect all the occurrences of one or more template signals in a long signal with transformation of the time axes as in DTW. The MTMM-DTW algorithm works iteratively as follows: In each iteration, the subsequence DTW is executed separately for all of the template signals, and the subsequence with the smallest distance is selected as the matching subsequence of that iteration. The matched subsequence of the test signal is made invisible to the future subsequence DTW executions to prevent it from matching again in the future iterations. (10 %-prefix and 10 %-suffix of the subsequence are left visible to allow some overlapping between the matched subsequences.) If no subsequence is matched in the iteration due to certain constraints (see [16] for details), the MTMM-DTW algorithm stops. In this way, possibly multiple subsequences are obtained, each of which matches to one of the templates with a DTW distance.

We use the MTMM-DTW algorithm with the templates being different execution types of the exercises to classify the executions in one of the execution types of one of the exercises. By including correct and incorrect execution types of the exercises, the algorithm can be used to evaluate the executions as correct/incorrect, giving the error type if there is any. It can also provide a quantitative measure of similarity between each matched execution and its template.

4 Experiments and Results

Five MTx units manufactured by Xsens Technologies [15] are fixed to different positions on the subject's body. Each unit contains three tri-axial devices: an accelerometer, a gyroscope, and a magnetometer. Two different sensor configurations are used to capture leg and arm movements (see Fig. 2).

Eight physical therapy exercises are considered, which are the most commonly assigned exercises to patients, mostly for orthopedic rehabilitation. They were suggested and approved by a physical therapy specialist [14]. The exercises are shown in Fig. 3. Exercises 1–5 and 6–8 use leg and arm sensor configurations, respectively.

In the experiments, there are 5 subjects, 8 exercises, and 3 execution types of each exercise: one correct and two incorrect. For the incorrect execution types, we select the two most common errors for each exercise: fast execution (*type-1 error*) and execution in low amplitude (*type-2 error*). For the templates, each subject performs each execution type of each exercise three times and one of them is selected as the template for that particular execution type, exercise, and subject so that there are $3 \times 8 \times 5 = 120$ templates in the dataset. For the test dataset, for each subject and exercise, the subject simulates a physical therapy session by executing the exercise

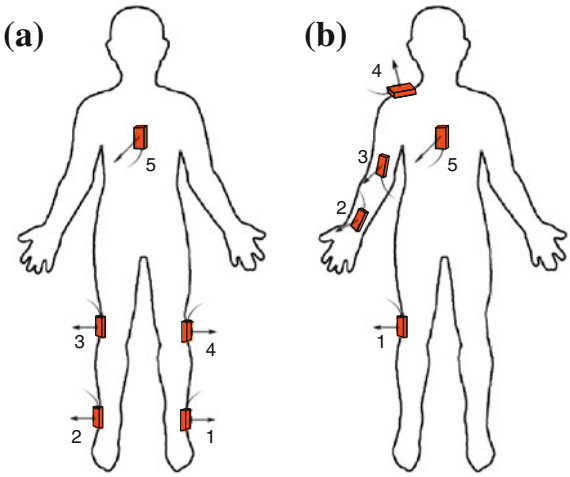

Fig. 2 The two sensor configurations for **a** the *right leg* and **b** the *right arm* movements. The sensor units are shown as *boxes* with the *arrows* and the cables indicating the z and $-x$ directions, respectively

10 times correctly, waiting idly for some time, then executing the exercise 10 times with type-1 error, waiting idly, and then executing it 10 times with type-2 error. There are a total of 1,200 executions in the test dataset.

Before applying the MTMM-DTW algorithm, we normalize the signals such that all of the axes of accelerometer, gyroscope, and magnetometer signals have unit variance on average in the whole dataset. We apply MTMM-DTW to the experimental data such that each subsequence must have a duration of at least half of the matching template and the subsequences are allowed to overlap with each other up to 5 % of their durations in the beginning and at the end.

The results are summarized as follows (see Table 1): 1, 125 executions are detected in the whole dataset containing 1,200 executions. This shows that the system makes −6.25 % error in counting the exercises. The average misdetection (MD) and false alarm (FA) rates[1] are 8.58 and 4.91 %, respectively, for the whole dataset. The overall accuracy of the system in exercise classification only[2] is 93.46 %, whereas the overall accuracy in simultaneous exercise and execution type classification[3] is 88.65 %. The overall sensitivity and specificity rates are 91.42 and 95.09 %, respectively, in the whole dataset. The recognized executions and the correctness of their evaluation by the system are shown in Fig. 4 for exercises 1–2 of subject 5. In the whole dataset, there are no exercise misclassifications at all.

[1] The MD and FA rates are calculated as MD rate $= \frac{\#\text{false negatives}}{\#\text{positives}}$ and FA rate $= \frac{\#\text{false positives}}{\#\text{negatives}}$.

[2] The accuracy of exercise classification is calculated as $\frac{\#\text{correct exercise classifications}+\#\text{true negatives}}{\#\text{positives}+\#\text{negatives}}$.

[3] The accuracy of exercise and execution type classification is calculated as $\frac{\#\text{correct exercise and execution type classifications}+\#\text{true negatives}}{\#\text{positives}+\#\text{negatives}}$.

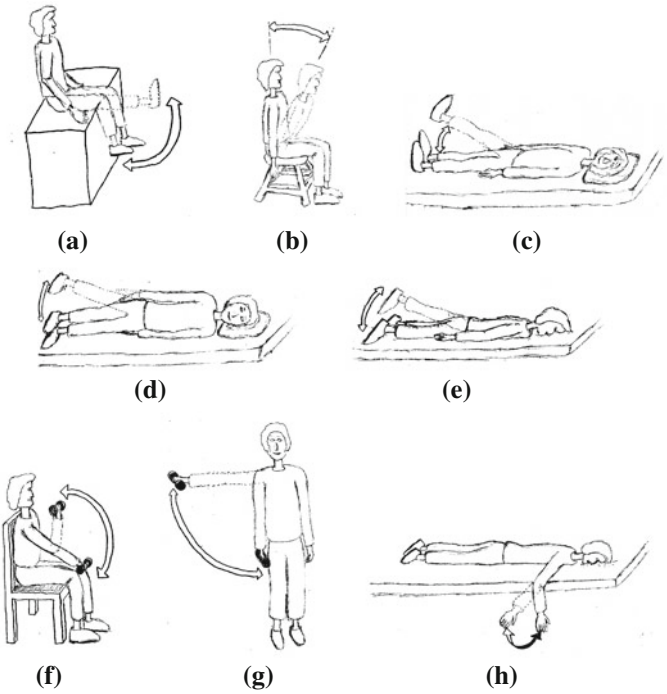

Fig. 3 Physical therapy exercises. In each exercise, the subject moves his/her leg or arm from the solid position to the dotted position, waits for 5 s, and returns to the solid position. **a** Exercise 1; **b** Exercise 2; **c** Exercise 3; **d** Exercise 4; **e** Exercise 5; **f** Exercise 6; **g** Exercise 7; **h** Exercise 8

Table 1 Summary of experimental results

Number of detected executions	1,125
Number of actual executions	1,200
Accuracy of exercise classification	93.46 %
Accuracy of exercise and execution type classification	88.65 %
Misdetection rate	8.58 %
False alarm rate	4.91 %
Sensitivity	91.42 %
Specificity	95.09 %

5 Discussion

Detection and classification of the executions are based on prerecorded templates; thus, the patient needs to execute the exercise(s) once for each execution type under the supervision of a physical therapy specialist to record the templates. The execution types contain both correct and incorrect ways of executing a particular exercise. Then, in a physical therapy session, the system automatically detects and provides

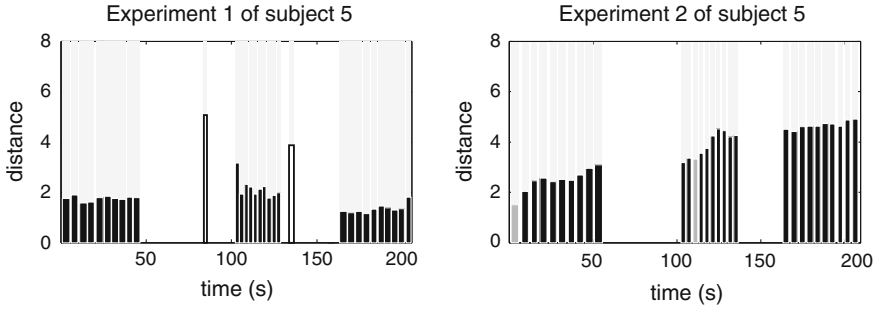

Fig. 4 Detection and classification of exercise executions in exercises 1–2 of subject 5. Each detected execution is shown as a bar whose width is the execution's duration and height is the DTW distance between the detected subsequence and the matching template. Black and gray filled bars indicate the correct and incorrect execution type classifications, respectively, when the exercise is classified correctly. Unfilled bars indicate the executions classified incorrectly

feedback about the exercise executions. Unlike body model- or rule-based systems, the proposed system requires no adjustments or configuration specific to patients, exercises, sensor types and placements; the only need is recording of the template executions. This eliminates the need for an engineer to reconfigure the system for every new exercise or execution type. Moreover, any sensor configuration representing the movements properly may be directly used without making any adjustments to the algorithm, provided that the configuration is the same in recording template signals and using the system. Furthermore, other sensor types can also be used with the same algorithm. For instance, the wearable inertial sensors and magnetometers can be replaced by the tags of a tag-based localization system that estimates the 3-D positions of the wearable tags and exactly the same methodology can be used for evaluation of the exercise executions [17].

For the detection of multiple occurrences of multiple templates, we select an approach based on DTW because the DTW algorithm is much more flexible than the absolute and Euclidean distance measures in comparing two signals since it tries to match similar parts of the signals. This may be beneficial when the variation in different executions of the same physical therapy exercise is considered. The speeds or durations of some parts of the exercise movement may change, which should be tolerable. For instance, if the exercise contains a phase at which the patient waits for 5 s, the distance should not increase significantly when the patient waits for 4 or 6 s. On the other hand, at the same time, the distance measure should not tolerate significant differences in amplitude that may occur, for example, when the patient waits for 5 s in a different position or orientation. If the absolute or Euclidean distance measures are used, both variation types (amplitude and time) affect the distance value and it is not possible to allow one of them while penalizing the other. In contrast, the DTW algorithm naturally compensates linear or nonlinear changes in the time (or sample) axis but not changes in amplitude, which is desireable in this scheme.

Unlike most of the previous studies, once the exercise movements are recorded, the system proposed in this paper automatically detects the movements as well as the idle time periods, if there are any, during an exercise session, independent of the number of exercise types. The system also classifies each movement as one of the exercise types and evaluates it, indicating the error type if there is any. The patient neither needs to press a button in the beginning and the end of each execution nor select the exercise type that he is going to perform. A physiotherapist is needed only while recording the movements in order to make sure that the patient performs the exercise correctly or with a predetermined error. Then, the patient can perform the exercises anywhere provided that he properly wears the sensors, and can observe how well he performs. Since the system also counts the executions, it can be used to notify the patient when he completes the advised number of repetitions in a given time interval. The results may also be checked by an expert to observe the patient's progress.

6 Conclusion and Future Work

In this paper, a novel algorithm, MTMM-DTW, is proposed to detect all of the occurrences of multiple templates in a long signal and is applied to the wearable inertial sensor and magnetometer signals in the area of physical therapy to detect and evaluate individual exercise executions in a physical therapy session. The system is autonomous in that it automatically detects the executions, classifies them as one of the exercises and further classifies them as correctly or incorrectly executed, indicating the error type if there is any. In the newly recorded dataset consisting of 120 templates and 1,200 exercise executions in addition to the idle time intervals, the system makes $-6.25\,\%$ error in counting the exercises. The accuracy is $93.46\,\%$ for exercise classification and $88.65\,\%$ for both exercise and execution type classification. Thus, the system can be used for autonomous feedback in physical therapy.

In a future study, the proposed system may be implemented to run in real-time to provide immediate feedback after each execution. The system can be optimized differently for the individual exercise types and subjects. The experiments may be performed in a more robust way to minimize intra-class variations. Other sensor technologies may be used, such as RF localization [17], to directly get the position information instead of rate information provided by accelerometers and gyroscopes (the second derivative of the linear position and the first derivative of the angular position (angle), respectively). In this case, the relative positions of the RF tags worn on the body can be estimated in 3-D space without any drift errors that exist in inertial sensors. The main drawback of such a system compared to inertial sensing would be that it would radiate radio waves to the environment and would limit the user to a restricted area. Moreover, the system may be used with only one template of the correct execution of each exercise so that the executions are classified in one of the exercises according to the most similar template and then evaluated by applying a threshold to the corresponding DTW distance—if the distance is below the threshold,

meaning that the execution is sufficiently similar to the template, the execution is classified as correct. However, in this case, the threshold would probably have to be determined individually for each exercise or each subject, and it may be difficult to set an appropriate threshold value based on the template signals.

References

1. Brutovsky J, Novak D (2006) Low-cost motivated rehabilitation system for post-operation exercises. In: Proceedings of 28th annual international IEEE-EMBS. New York, USA, pp 6663–6666 (30 Aug–3 Sept)
2. Fergus P, Kafiyat K, Merabti M, Taleb-bendiab A, El Rhalibi A (2009) Remote physiotherapy treatments using wireless body sensor networks. In: Proceedings of integrated circuit wireless Communication Network. New York, USA, pp 1191–1197 (21–24 June)
3. Keogh EJ, Pazzani MJ (1999) Scaling up dynamic time warping to massive datasets. Lect Notes Comput Sci 1704:1–11 (Springer: Berlin, Heidelberg, Germany)
4. Kifayat K, Fergus P, Cooper S, Merabti M (2010) Body area networks for movement analysis in physiotherapy treatments. In: Proceedings of 24th IEEE integrated circuit advanced information network. Perth, Australia, pp 866–872 (20–23 April)
5. Giorgino T, Tormene P, Maggioni G, Pistarini C, Quaglini S (2009) Wireless support to post-stroke rehabilitation: myheart's neurological rehabilitation concept. IEEE T Inf Technol B, 13(6):1012–1018 (Nov 2009)
6. Milenkovic M, Jovanov E, Chapman J, Raskovic D, Price J (2002) An accelerometer-based physical rehabilitation system. In: Proceedings of Southeast symposium System. Huntsville, AL, USA, pp 57–60 (18–19 March)
7. Müller M (2007) Information retrieval for music and motion, vol 6. Springer, Berlin, Heidelberg, Germany
8. Pitta F, Troosters T, Probst VS, Spruit MA, Decramer M, Gosselink R (2006) Quantifying physical activity in daily life with questionnaires and motion sensors in COPD. Eur Respir J 27(5): 1040–1055
9. Raso I, Hervás R, Bravo J (2010) m-Physio: personalized accelerometer-based physical rehabilitation platform. In: Proceedings of 4th integrated circute mobile ubiquitous computing, systems, services and technologies. Florence, Italy, pp. 416–421 (25–30 Oct)
10. Stéphane C, Mathieu H, Patrick B (2008) Accelerometer-based wireless body area network to estimate intensity of therapy in post-acute rehabilitation. J Neuroeng Rehabil 5(1):20–31
11. Tao Y, Hu H (2008) A novel sensing and data fusion system for 3-D arm motion tracking in telerehabilitation. IEEE IMTC 57(5):1029–1040
12. Tao Y, Hu H, Zhou H (2007) Integration of vision and inertial sensors for 3D arm motion tracking in home-based rehabilitation. Int J Robot Res 26(6):607–624
13. Tormene P, Giorgino T, Quaglini S, Stefanelli M (2009) Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation. Artif Intell Med 45(1):11–34
14. Tuğcu İ MD (2012) Physical therapy specialist. Department of physical therapy and rehabilitation, Gülhane military medical academy, Turkish Armed Forces Rehabilitation Centre. Personal, communication (June 2012)
15. Xsens Technologies BV (2009) Enschede, The Netherlands. MTi and MTx user manual and technical documentation. http://www.xsens.com
16. Yurtman A (2012) Recognition and classification of human activities using wearable sensors. M.S. Thesis, Department of Electrical and Electronics Engineering, Bilkent University, Turkey.

17. Yurtman A, Barshan B (2012) Human activity recognition using tag-based localization (Etiket-tabanlı konumlama ile insan aktivitelerinin tanınması). In: Proceedings of IEEE 20th conference of signal processing communicational Application Fethiye, Muğla, Turkey (18–20 April)
18. Zheng H, Davies RJ, Black ND (2005) Web-based monitoring system for home-based rehabilitation with stroke patients. In: Proceedings of 18th IEEE computional medical system. Dublin, Ireland, pp 419–424 (23–24 June)
19. Zhou H, Hu H (2005) Inertial motion tracking of human arm movements in stroke rehabilitation. In: Proceedings of 2005 IEEE conference mechatronics and automation, vol 3. Ontario, Canada, pp 1306–1311

Part VII
Network Security, Data Integrity and Privacy

Commutative Matrix-based Diffie-Hellman-Like Key-Exchange Protocol

Alexander G. Chefranov and Ahmed Y. Mahmoud

Abstract A matrix-based Diffie-Hellman-like key exchange protocol and utilizing it secure key-exchange protocol similar to HMQV are proposed. The proposed key exchange protocol uses matrix multiplication operation only; it does not rely on the complexity of the discrete logarithm problem contrary to the prototype and its known variants. Two-way arrival at the common key, similar to that employed in the Diffie-Hellman protocol, is provided by specially constructed commutative matrices. The trap-door property ensuring the proposed protocol security is based on exploiting of a non-invertible public matrix in the key generating process.

1 Introduction

The key-exchange Diffie-Hellman (DH) protocol without transfer of the secret key over an insecure channel was suggested in [1]. It is based on the computational complexity of discrete logarithm problem (DLP) solving over a finite field $GF(q)$, where q is prime. It uses a publicly available primitive element α of $GF(q)$. Each user generates an independent secret random number X_i from a set of integers $\{1, 2, \ldots, q-1\}$ but makes publicly available $Y_i = \alpha^{X_i} \bmod q$. If users i and j want to communicate privately, they use the value of $K_{ij} = \alpha^{X_i X_j} \bmod q = {Y_i}^{X_j} \bmod q = {Y_j}^{X_i} \bmod q$ as a secret key. This technique requires rather long numbers (200-bit big numbers are considered for estimation of its security as 2^{100} operations complexity in [1]). Conventional computers usually deal with 32- or 64-bit numbers. DH protocol was extended to matrix rings in [2], but it is still based on the complexity of the DLP.

A. G. Chefranov (✉)
Department of Computer Engineering, Eastern Mediterranean University, Famagusta, North Cyprus
e-mail: Alexander.chefranov@emu.edu.tr

A. Y. Mahmoud
Information Technology Department, Al-Azhar University, Gaza Strip, Palestine
e-mail: ahmed@alazhar.edu.ps

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_31,

Its security was discussed in [3]. Further DH protocol matrix oriented modifications based on DLP are proposed and discussed in [4–8]. A group-theoretic public-key exchange protocol is proposed in [9]. This protocol [9] requires the transference of the full sets of group elements by both parties willing to get a common key and is based on the complexity of solving conjugacy equations. A non-commutative group-theoretical DH protocol extension using exponentiation is described in [10].

Herein, we propose a DH-like protocol using commutative matrices represented as conjugates to diagonal matrices. The exploited trap-door function is a matrix multiplication with the zero-determinant matrix. In DH, a publicly known primitive element is used to obtain a public key from a private key. Similarly, in the proposed protocol, some publicly available zero-determinant matrix is used to construct a public key from the private ones. The commutativity of the matrices applied as private keys allows both parties to arrive at the same key by different ways of calculation. This is similar to the procedure in the DH protocol except that we multiply matrices instead of exponentiation of big numbers. The proposed protocol has high performance as computationally very simple one because it applies few operations of multiplication to matrices whose entries are conventional numbers. Contrary to DH, even for 16×16 matrices with short 7-bit integer entries it provides substantial security of 2^{112} search space size. The DH protocol was found to be susceptible to the intruder-in-the-middle attack [11] that led to the invention of numerous DH extensions providing resistance to the attack. Currently, MQV and HMQV are considered as the most secure key-exchange protocols based on the DH [12, 13]. Similarly, our DH-like protocol is also susceptible to the attack, and we show how to extend our protocol to a secure key-exchange protocol resembling MQV and HMQV. The rest of the paper is organized as follows: Sect. 2 presents our DH-like key exchange protocol, and Sect. 3 analyses its security. Section 4 describes briefly MQV and HMQV and introduces a secure extension of our DH-like protocol that is similar to HMQV. A conclusion is given in Sect. 5.

2 The Protocol

Assume two communicating parties, $A_i, i = \overline{1,2}$, share two publicly available matrices

$$M \in GL(m, F), G \in GN(m, F), \tag{1}$$

where $GL(m, F)$ is the set of all invertible $m \times m$ matrices with entries from the field F with $|F|$ elements, and $GN(m, F)$ is the set of $m \times m$ matrices with entries from the field F and having rank $m - 1$ and zero determinant value,

$$rank(G) = m - 1, \det(G) = 0. \tag{2}$$

Assume that a party A_i has two secret matrices (its private key)

$$X_{ij} = M^{-1} D_{ij} M \in GL(m, F), \tag{3}$$

where $D_{ij} \in GL(m, F)$ is a diagonal matrix, $i = \overline{1,2}$, $j = \overline{1,2}$. It is easy to see that these matrices commute:

$$\begin{aligned} X_{1i}X_{2j} &= M^{-1}D_{1i}MM^{-1}D_{2j}M = M^{-1}D_{1i}D_{2j}M \\ &= M^{-1}D_{2j}D_{1i}M = M^{-1}D_{2j}MM^{-1}D_{1i}M = X_{2j}X_{1i}, \end{aligned} \tag{4}$$

since diagonal matrices commute.

With the following protocol, the parties can obtain a key, K, shared by both parties without the transfer of K.

2.1 The Protocol

1. User A_i calculates his public key

$$Y_i = X_{i1}^G X_{i2}, i = \overline{1,2}. \tag{5}$$

2. User A_i sends his public key Y_i to the user A_{3-i},$i = \overline{1,2}$.
3. User A_i calculates the common key $K = K_1 = K_2$ using his private key and the received public key of his partner

$$K_i = X_{i1}^{Y_{3-i}} X_{i2}, i = \overline{1,2}. \tag{6}$$

The protocol results in the same value $K = K_1 = K_2$ for both parties since, due to (4)–(6), the following holds:

$$K_1 = X_{11}Y_2X_{12} = X_{11}X_{21}GX_{22}X_{12} = X_{21}X_{11}GX_{12}X_{22} = X_{21}Y_1X_{22} = K_2.$$

Note that due to (2) and (3)

$$rank(K) = m - 1, \det(K) = 0. \tag{7}$$

Example 1 Let $m = 2$, $F = Z_5 = \{0, 1, .., 4\}$, and according to (1), (2),

$$\begin{aligned} M = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, M^{-1} = \begin{bmatrix} 3 & 1 \\ 4 & 2 \end{bmatrix}, G = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}, D_{11} = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}, \\ D_{12} = \begin{bmatrix} d_3 & 0 \\ 0 & d_4 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}, D_{21} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, D_{22} = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}. \end{aligned} \tag{8}$$

Then from (3) and (8) one calculates

$$
\begin{aligned}
X_{11} &= \begin{bmatrix} 3 & 1 \\ 4 & 2 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 4 & 8 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ 3 & 0 \end{bmatrix}, \\
X_{12} &= \begin{bmatrix} 3 & 1 \\ 4 & 2 \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 6 & 4 \\ 8 & 8 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ 2 & 3 \end{bmatrix}, \\
X_{21} &= \begin{bmatrix} 3 & 1 \\ 4 & 2 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ 4 & 4 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 4 & 4 \\ 1 & 4 \end{bmatrix}, \\
X_{22} &= \begin{bmatrix} 3 & 1 \\ 4 & 2 \end{bmatrix}\begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 4 & 0 \end{bmatrix}.
\end{aligned} \tag{9}
$$

And according to (5), from (9)

$$
\begin{aligned}
Y_1 &= \begin{bmatrix} 0 & 2 \\ 3 & 0 \end{bmatrix}\begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}\begin{bmatrix} 3 & 3 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} 3 & 3 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 2 & 1 \end{bmatrix} \\
Y_2 &= \begin{bmatrix} 4 & 4 \\ 1 & 4 \end{bmatrix}\begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 4 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 4 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 4 & 2 \end{bmatrix}.
\end{aligned} \tag{10}
$$

Finally, by (6) and (10),

$$
\begin{aligned}
K_1 &= \begin{bmatrix} 0 & 2 \\ 3 & 0 \end{bmatrix}\begin{bmatrix} 0 & 0 \\ 4 & 2 \end{bmatrix}\begin{bmatrix} 3 & 3 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 3 & 3 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 0 & 0 \end{bmatrix}, \\
K_2 &= \begin{bmatrix} 4 & 4 \\ 1 & 4 \end{bmatrix}\begin{bmatrix} 2 & 1 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 4 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 4 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 0 & 0 \end{bmatrix}.
\end{aligned} \tag{11}
$$

Thus, from (11) one gets $K = K_1 = K_2$, that satisfies (7).

3 Security Analysis of the Protocol

An opponent knowing M, G and viewing $Y_i, i = \overline{1,2}$, is not able to obtain K because he needs the secret matrices $X_{ij}, i = \overline{1,2}, j = \overline{1,2}$ for that purpose. He can try to get them by substituting (3) into (5) and solving the resulting system of nonlinear algebraic equations with respect to the $2m$ unknown diagonal elements $D_{i1}(l,l), D_{i2}(l,l), l = \overline{1,m}$. For the example above, (3) and (8) yield the following private key matrices

$$
\begin{aligned}
X_{11} &= \begin{bmatrix} 3 & 1 \\ 4 & 2 \end{bmatrix}\begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 3d_1 & d_2 \\ 4d_1 & 2d_2 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 3(d_1+d_2) & d_1+4d_2 \\ 4d_1+d_2 & 3(d_1+d_2) \end{bmatrix}, \\
X_{12} &= \begin{bmatrix} 3 & 1 \\ 4 & 2 \end{bmatrix}\begin{bmatrix} d_3 & 0 \\ 0 & d_4 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 3d_3 & d_4 \\ 4d_3 & 2d_4 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 3(d_3+d_4) & d_3+4d_4 \\ 4d_3+d_4 & 3(d_3+d_4) \end{bmatrix},
\end{aligned}
$$

and the following matrix equation

$$Y_1 = \begin{bmatrix} 3(d_1+d_2) & d_1+4d_2 \\ 4d_1+d_2 & 3(d_1+d_2) \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} 3(d_3+d_4) & d_3+4d_4 \\ 4d_3+d_4 & 3(d_3+d_4) \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 2 & 1 \end{bmatrix},$$

or

$$\begin{aligned} Y_1 &= \begin{bmatrix} 2d_1+4d_2 & d_1+2d_2 \\ d_1+3d_2 & 3d_1+4d_2 \end{bmatrix} \begin{bmatrix} 3(d_3+d_4) & d_3+4d_4 \\ 4d_3+d_4 & 3(d_3+d_4) \end{bmatrix} \\ &= \begin{bmatrix} 2d_1d_4+4d_2d_4 & d_1d_4+2d_2d_4 \\ d_1d_4+3d_2d_4 & 3d_1d_4+4d_2d_4 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 2 & 1 \end{bmatrix} \end{aligned} \tag{12}$$

is obtained.
Denoting

$$z_1 = d_1d_3, z_2 = d_1d_4, z_3 = d_2d_3, z_4 = d_2d_4, \tag{13}$$

the following system of equations can be derived from (12)

$$\begin{aligned} 2z_2 + 4z_4 &= 2, \\ z_2 + 2z_4 &= 1, \\ z_2 + 3z_4 &= 2, \\ 3z_2 + 4z_4 &= 1. \end{aligned} \tag{14}$$

From the last two equations of (14) one can write

$$z_2 = 2 + 2z_4. \tag{15}$$

The first two equations of (14) give

$$z_2 = 1 + 3z_4. \tag{16}$$

Hence, (15) and (16) imply

$$z_4 = 1, z_2 = 4. \tag{17}$$

The other unknowns, z_1, z_3, are not defined. From (13) and (17), it follows that

$$z_4 = 1 = d_2d_4, z_2 = 4 = d_1d_4. \tag{18}$$

From (18), one gets

$$4d_2 = d_1. \tag{19}$$

It is easy to check that D_{11} defined in (8) satisfies (19).

As in the example above after taking (3) into account, the system (5) may be solved as a system of m^2 linear algebraic equations by introducing m^2 unknowns $z_{ilk} = D_{i1}(l,l) \cdot D_{i2}(k,k), l = \overline{1,m}, k = \overline{1,m}$. However, due to (2), one of its rows is a linear combination of its other rows. Hence, the right-hand side of (5) has one row that is a linear combination of its other rows. The system of m^2 equations therefore has $m^2 - m$ linear independent equations. It is thus undetermined and its solution has m free variables, enumeration of whose possible values is required in order to find a solution of (3), (5). In the worst case this enumeration requires checking of $|F|^m$ combinations. If e.g., $|F| = 128, m = 16$, then the number of combinations to check is 2^{112} which is unfeasible for the current level of computer development.

4 Matrix-Based DH-Like Secure Protocol Extension

The proposed matrix-based DH-like protocol bares the same deficiencies as the original DH key-exchange protocol does, e.g., it is susceptible to the intruder-in-the-middle attack [11]. Secure protocols extending DH original protocol are considered, e.g., in [11–13]. These algorithms use some information known in advance to the both communicating parties (passwords, static keys). These secure protocols are based on the original DH key-exchange protocol and may be used in multiplicative (for the Discrete Logarithm Problem) or in additive (Elliptic Curves) fields [13, p. 3].

Secure MQV and HMQV protocols are described in [13, Figs. 1, 2]. Assuming that certified static public keys, $A = g^a$, $B = g^b$, (calculated using respective private keys $a, b \in Z_q$) of the communicating parties, $\hat{A}$, $\hat{B}$, respectively, are known to the parties in advance, MQV and HMQV may be represented by Fig. 1.

Contrary to the original DH key-exchange protocol using some secret numbers (ephemeral, dynamic keys) x, y, generated by users, and respective public values

Fig. 1 Computation of the session key K by MQV and HMQV (H is a hash function)

$$\hat{A} : generate\, x, X = g^x;$$
$$\hat{B} : generate\, y, Y = g^y;$$
$$\hat{A} \rightarrow \hat{B} : \hat{A}, \hat{B}, X$$
$$\hat{B} \rightarrow \hat{A} : \hat{B}, \hat{A}, Y$$
$$\hat{A} : \sigma_{\hat{A}} = (YB^e)^{x+da}; K = H(\sigma_{\hat{A}})$$
$$\hat{B} : \sigma_{\hat{B}} = (XA^d)^{y+eb}; K = H(\sigma_{\hat{B}})$$
$$MQV : d = 2^l + X \bmod 2^l, e = 2^l + Y \bmod 2^l;$$
$$l = \lceil \log_2 q \rceil / 2$$
$$HMQV : e = H(X \parallel \hat{B}), d = H(Y \parallel \hat{A})$$

Fig. 2 Matrix-based analogue of HMQV

$$\hat{A}: generate\ x = (X_1, X_2), X = X_1 G X_2;$$
$$\hat{B}: generate\ y = (Y_1, Y_2), Y = Y_1 G Y_2;$$
$$\hat{A} \rightarrow \hat{B}: \hat{A}, \hat{B}, X$$
$$\hat{B} \rightarrow \hat{A}: \hat{B}, \hat{A}, Y$$
$$\hat{A}: \sigma_{\hat{A}} = eX_1 B X_2 + dA_1 Y A_2; K = H(\sigma_{\hat{A}});$$
$$\hat{B}: \sigma_{\hat{B}} = eB_1 X B_2 + dY_1 A Y_2; K = H(\sigma_{\hat{B}})$$
$$e = H(X \parallel \hat{B}), d = H(Y \parallel \hat{A})$$

X, Y, only, their static private, a, b, and public, A, B, keys are used in MQV and HMQV (Fig. 1) thus not allowing an opponent to apply intruder-in-the-middle attack.

Matrix analogue of HMQV may be represented by Fig. 2 assuming that public keys, $A = A_1 G A_2, B = B_1 G B_2$ (generated from respective private keys, $a = (A_1, A_2), b = (B_1, B_2)$) are known to the both communicating parties in advance. They generate their ephemeral secret keys, calculate respective public keys, exchange them, and use both static and ephemeral keys to calculate common session key K (Fig. 2).

Hence, our protocol opens a door for a variety of similar secure protocols which are however less computationally intensive than DLP- or Elliptic Curve-based protocols since they apply few simple matrix multiplications only.

5 Conclusion

The proposed DH-like matrix protocol is based on few matrix multiplications and does not use exponentiation as other known DH protocol modifications do. The concept of the proposed protocol is the same as that of DH: it allows two-way arrival at the same common key that is provided by the use of private key matrices mutually commuting as conjugates to diagonal invertible matrices. The public key is obtained by multiplication of the private key matrices with a publicly known zero-determinant matrix. The non-invertibility of this matrix defines the trap-door property of our protocol. For 16×16 matrices with 7-bit integer entries it ensures a substantial security of 2^{112} search space size. The proposed protocol, contrary to previous ones, may operate with short numbers and is computationally simple thus assuring its high performance and wide applicability. The proposed matrix-based DH-like protocol bares the same deficiencies as the original DH key-exchange protocol, e.g., it is susceptible to the intruder-in-the-middle attack [11]. Secure protocols extending DH original protocol are considered, e.g., in [11–13]. These algorithms (e.g., MQV, HMQV) use some information known in advance to the both communicating parties (passwords, static keys) and may be exploited in multiplicative fields (for the Discrete Logarithm Problem), or in additive fields (Elliptic Curves) [13, p. 3]. Herein, we

propose a protocol similar to HMQV that is based on our DH-like protocol. Hence, our protocol opens a door for a variety of similar secure protocols which are however less computationally intensive than DLP- or Elliptic Curve-based protocols since they use few simple matrix multiplications.

References

1. Diffie W, Hellman M (1976) New directions in cryptography. IEEE Trans Inform Theory 22(6):644–654
2. Odoni RWK, Varadharajan V, Sanders PW (1984) Public key distribution in matrix rings. Electron Lett 20(9):386–387
3. Varadharajan V, Odoni R (1986) Security of public key distribution in matrix rings. Electron Lett 22(1):46–47
4. Alvarez R, Martinez FM, Vicent JF, Zamora A (2008) A matricial public key cryptosystem with digital signature. WSEAS Trans Math 7(4):195–204
5. Alvarez R, Tortosa L, Vicent JF, Zamora A (2009) Analysis and design of a secure key exchange scheme. Inform Sci 179:2014–2021
6. Vasco MIG, del Pozo ALP, Duarte PT (2009) Cryptanalysis of a key exchange scheme based on block matrices. http://eprint.iacr.org/2009/553.pdf Accessed 29 May 2013
7. Yang J, Yang X (2008) A new variant of the Diffie-Hellman key exchange protocol based on block triangular matrix groups. International Conference Intelligent Information Hiding and Multimedia Signal Processing, pp 1277–1281. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4604276 Accessed 29 May 2013
8. Nelson JB (2003) The Diffie-Hellman key exchange protocol in matrices over a field and a ring. MS. Thesis, Dept. Math., Texas Technical University. http://repositories.tdl.org/ttu-ir/bitstream/handle/2346/22169/31295017074922.pdf?sequence=1 Accessed 29 May 2013
9. Anshel I, Anshel M, Goldfeld D (1999) An algebraic method for public-key cryptography. Math Res Lett 6:1–5
10. Grigoriev D, Ponomarenko I, (2005) Constructions in public-key cryptography over matrix groups. Contemp Mathe 418:103–119. http://arxiv.org/PS_cache/math/pdf/0506/0506180v1.pdf Accessed 29 May 2013
11. Diffie W, van Oorschot PC, Wiener MJ (1982) Authentication and authenticated key exchange. Des Codes Crypt 2(2):107–125
12. Law L, Menezes A, Qu M, Solinas J, Vanstone S (2003) An efficient protocol for authenticated key agreement. Des Codes Crypt 28(2):119–134
13. Krawczyk H (2005) HMQV: A High-Performance Secure Diffe-Hellman Protocol, In Proc. Of Advances in Cryptology - CRYPTO 2005. 25^{th} Annual International Cryptology Conference, Santa Barbara, California, USA, August 14–18, 2005. Lecture Notes in Computer Science 3621:546–566

Anonymity in Multi-Instance Micro-Data Publication

Osman Abul

Abstract In this paper we study the problem of anonymity in multi-instance (MI) micro-data publication. The classical k-anonymity approach is shown to be insufficient and/or inappropriate for MI databases. Thus, it is extended to MI databases, resulting in a more general setting of MI k-anonymity. We show that MI k-anonymity problem is NP-Hard and the attack model for MI databases is different from that of single-instance databases. We make an observation that the introduced MI k-anonymity is not a strong privacy guarantee when anonymity sets are highly unbalanced with respect to instance counts. To this end a new anonymity principle, called p-certainty, which is unique to MI case is introduced. A clustering algorithms solving the p-certainty anonymity principle is developed and experimentally evaluated.

1 Introduction

Many micro-data datasets carry sensitive personal information. Broadly speaking, third parties can potentially benefit from micro-data sharing as the shared data may contain valuable knowledge once analyzed. However, institutions having the right to publish micro-data may be reluctant of doing so because of privacy concerns. The canonical privacy enhancing solution when publishing micro-data datasets is *anonymization*, the process of preventing identifiability at the cost of data quality loss. So, the challenge is preserving the data quality as much as possible so that valid and useful data analysis are still possible with the distorted data.

The simplest approach to providing anonymity is removing identifier (key) attributes from micro-data prior to data publication. Unfortunately, this simple approach does not guarantee anonymity most of the time due to existence of

O. Abul (✉)
Department of Computer Engineering, TOBB University of Economics and Technology, Ankara, Turkey
e-mail: osmanabul@etu.edu.tr

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_32,

so-called quasi-identifiers, a set of attributes that plays the role of an identifier when joined to a public database such as phone directories. Having the fact noticed, the traditional approach to providing anonymity in data publishing is *k-anonymity*, which ensures for every individual to be indistinguishable from at least k-1 other individuals based on quasi-identifier equality.

The k-anonymity is extensively studied in various contexts with varying formulations. However, the anonymity problem in general and k-anonymity problem in particular are not directly addressed in the literature for multi-instance (MI) datasets, where more than one records (tuples) may belong to the same subject (individual). The k-anonymity literature makes the assumption that no subject has more than one tuple in the dataset. Sweeney [22] states that this assumption is not a restriction but a simplification without loss of generality. The idea is to collapse all tuples of single individuals into one virtual tuple and solve the single-instance anonymization problem afterwards. Although this works in general, beyond this solution we claim that MI datasets need a special treatment for more effective anonymization and may need stronger privacy requirements. Moreover, the attack models are quite different. So, we directly address the anonymity problem and related issues in MI datasets.

There are many domains where MI micro-datasets with private/sensitive information naturally arise. An example domain is patient visits to hospitals, like the example given in Fig. 1a. The collected data is MI as any patient can visit the same hospital in multiple occasions. Another example domain is the web site browsing history pivoted by visitors. Clearly, in these domains collected data may contain certain sensitive information.

1.1 Related Work

Chronologically the first approach in privacy preserving micro-data publishing was originated from statistical databases research [2]. Statistical disclosure control aims at avoiding the identification of the original database rows while at the same time allowing the reconstruction of the data distribution at an aggregate level, and thus the production of valid mining models [5, 6, 20].

The traditional k-anonymity framework [21] focuses on the anonymization of relational tables: the basic assumptions are that the table to be anonymized contains entity-specific information, that each tuple in the table corresponds uniquely to an individual, and that attributes are divided in *quasi-identifiers* (i.e., a set of attributes whose values in combination can be linked to external information to reidentify the individual to whom the information refers); and *sensitive attributes* (that we want to keep secret).

Although it has been shown that the k-anonymity model presents some limitations [16] and optimal k-anonymization is NP-hard [4], the k-anonymity model is still practically relevant. Several variants and extensions of k-anonymity principle are proposed to suit different privacy requirements. These include l-diversity [16], (α, k)-anonymity [23], t-closeness [15], m-invariance [24] and (c, k)-safety [17].

Theoretical results and solution strategies are provided for the mentioned privacy principles in the respective papers. The k-anonymity problem is also shifted from relational tables to moving object databases [1].

K-anonymization models can be dichotomized based on (1) generalization versus suppression, (2) local versus global recoding, and (3) hierarchy-based versus partition-based [12]. The solution strategies of models differ too; ranging from greedy algorithms [13] to approximate algorithms [4]. It has been shown [7] that in an optimal k-anonymity solution, every anonymity set has size between k and $2k - 1$. The idea of using *k-member clustering for k-anonymity* has recently been studied in [3], and extended in [14] to deal with attributes that have a hierarchical structure.

MI learning is extensively studied by machine learning and data mining communities. Applications using MI datasets include MI kernels for support vector machines [11] and clustering using expectation-maximization [10]. Though there is a growing literature on MI database mining and learning, to the best of our knowledge, the problem of anonymity issues in MI databases is not exclusively addressed yet. The closest work to ours is by Nergiz et al. [19] where they study k-anonymity in the multi-relational setting.

2 Anonymity in Multi-Instance Databases

Definition 1 (*Multi-instance database*) An MI database (MID) is a 3-tuple (P, A, M) where;

- P is a subject set where each subject is identified with an identifier and a number of quasi-identifier features,
- A is the union of sensitive and insensitive attributes, with feature space $Dom(A)$,
- $M : P \rightarrow 2^{Dom(A)} - \emptyset$ is a feature set mapping.

For the subject $p \in P$; $M(p)$ is interpreted as the set of sensitive/insensitive feature vectors of p.

In the special case of $|M(p)| = 1$ for all $p \in P$, then the *MID* becomes single-instance database (*SID*), hence *MID*s generalize *SID*s. For convenience, a *MID* is represented by an MI database as in Fig. 1a, where the database has the schema $D(Name, Birth, Zip, Service\ quality, Disease)$, where $Name$ is the identifier (ID) attribute, the columns $Birth$ and Zip constitute the quasi-identifier (QI) attributes, $Service\ quality$ is the insensitive attribute and $Disease$ is the sensitive attribute.

The classical k-anonymity principle is based on anonymity of tuples. In single-instance databases, tuple anonymity implies subject anonymity due to one to one correspondence between tuples and subjects. However, in MI databases tuple anonymity does not necessarily imply subject anonymity as any subject may have multiple tuples. The classical k-anonymization provides tuple anonymity but not subject anonymity. The approach of collapsing all tuples of a subject into one is not appropriate as shown in this study.

(a)

ID	QI		Insensitive	Sensitive
Name	Birth	Zip	Serv. qual.	Disease
John	1971	35100	High	HIV
John	1971	35100	Low	Flu
John	1971	35100	Low	Cancer
Mary	1977	35200	Medium	Flu
Mary	1977	35200	High	Fever
Hans	1980	32300	Medium	Flu
Kane	1982	32300	High	Fever
Paul	1984	32800	Low	HIV
Eric	1985	32700	High	Hepatitis
Eric	1985	32700	Low	Tetanus

(b)

QI		ID	# oftuples
Birth	Zip	Name	nbrtuple
1971	35100	John	3
1977	35200	Mary	2
1980	32300	Hans	1
1982	32300	Kane	1
1984	32800	Paul	1
1985	32700	Eric	2

(c)

QI		Insensitive	Sensitive
Birth	Zip	Serv. qual.	Disease
1971-80	3****	High	HIV
1971-80	3****	Low	Flu
1971-80	3****	Low	Cancer
1971-80	3****	Medium	Flu
1971-80	3****	High	Fever
1971-80	3****	Medium	Flu
1982-85	32***	High	Fever
1982-85	32***	Low	HIV
1982-85	32***	High	Hepatitis
1982-85	32***	Low	Tetanus

Fig. 1 **a** Hospital visit data. **b** Attacker's background knowledge. **c** MI 3-anonymous version

2.1 The Attack Model

The standard attack model in classical k-anonymity consists of a background knowledge which links quasi-identifiers to identifiers. The attacker's job is linking the tuples in published data to records in the phone directory. Since published data do not contain identifiers, the attacker leverages quasi-identifiers to do so. The responsibility of data publisher is then making inference of "identification of which tuple belongs to which subject" impossible even the background knowledge is exploited. This is usually achieved with quasi-identifier distortion.

In MI data publication, in the worst-case, we assume that attackers know all the subjects and additionally the number of tuples every subject has in the published data, as a part of the background knowledge. So, the data publisher should take into account this background knowledge B of the attacker to prevent any privacy breaches. The B can be modeled as a table where there is a record for each subject, containing (1) the quasi-identifiers, (2) the identifier, (3) the number of tuples (instances) the subject has in the released data. The background knowledge, B, of the attacker is given in Fig. 1b for the running example presented in Fig. 1a.

Let D be a MID and D' be another MID such that, $|D| = |D'|$ and D' is D's anonymized (quasi-identifiers generalized) and released (identifiers removed) version. To prevent any privacy breach, data publisher must guarantee relatively high number of feasible mappings (from D' to B) and relatively high number of every tuples participating in feasible mappings.

2.2 Multi-Instance k-anonymity Principle

The bottomline of the k-anonymity principle is to build a number of anonymity sets (equivalence sets w.r.t. QIs) where each anonymity set has size at least k. The classical k-anonymity builds anonymity sets based on tuples. However, in MI databases, the anonymity sets should be based on anonymity of subjects as given in Definition 2.

Definition 2 (*MI k-anonymity*) An MI database is k-anonymous if for every subject p there exists at least k-1 other subjects having the same QI features as p.

Lemma 1 *MI k-anonymity is a stronger property than classical k-anonymity, i.e. satisfaction of MI k-anonymity property implies satisfaction of classical k-anonymity property but not necessarily vice versa.*

Proof Suppose an anonymity set with size k tuples satisfy classical k-anonymity property but the subject set which these tuples refer to has size less than k. Then the anonymity set does not satisfy MI k-anonymity property according to Definition 2. On the other hand, suppose an anonymity set satisfies MI k-anonymity property, then there is at least k tuples in the anonymity set. □

The utility of Lemma 1 is that one can not simply treat a MID as a SID and apply algorithms providing classical k-anonymity. But the opposite is true. For instance, Fig. 1c is MI 3-anonymous, and also satisfies classical 3-anonymity property.

Problem 1 (*MI k-anonymous data publication*) Given the background knowledge B and an MI database D; transform D to database D', which is to be published, such that;

- for every $t \in D$ there is a corresponding $t' \in D'$,
- the transformation from D to D' is feasible w.r.t. B,
- D' has the same columns as D except the identifier column removed,
- D' has MI k-anonymity property (i.e. equivalently D' is MI k-anonymous), and
- the similarity between D and D' is maximized.

The last condition in Problem 1 is a maximization problem under the constraints preceding it. To obtain a concrete definition of the problem, the similarity measure between D and D' needs to be concretized. It has been proven that the concretized single-instance variants of Problem 1 are NP-Hard [1, 4, 18]. Using this, we prove that the MI variants are also NP-Hard as given in Theorem 1.

Theorem 1 *MI k-anonymous publication problem is NP-Hard.*

Proof The restricted (special case) version of MI k-anonymity, where every subject has exactly one instance, is equal to the classical k-anonymity problem setting. The proof by restriction for NP-Hardness [8] implies that MI k-anonymity is NP-Hard for every setting, which is NP-Hard, of the classical k-anonymity problem. □

Equipped with NP-Hardness of the problem, some approximate solutions or heuristics need to be devised. As an initial attempt we explore possibility of using algorithms developed for classical k-anonymity problem. The framework presented in the next subsection is an example of how multi-instances can be collapsed into virtually single instance so that classical k-anonymity works. The major benefit of the framework is to enable us to solve MI k-anonymity using already available algorithms developed for classical k-anonymity.

2.3 MI k-anonymity as Classical k-anonymity

Suppose that we are provided with an algorithm, $AlgKAnon$, solving the classical k-anonymity problem. The algorithm can be input with a SIT D and anonymity parameter k, to obtain the output D', the k-anonymized version of D. The method presented in Algorithm 1 gives the framework that constructs a solution to MI k-anonymity problem exploiting the generic algorithm of $D' \leftarrow AlgKAnon(D, k)$.

Algorithm 1 MI k-anonymity through classical k-anonymity

Input: D: A MID, k: anonymity parameter, $AlgKAnon$: An algorithm which solves the classical k-anonymity problem
Output: D': An MI k-anonymous MID where the identifier columns is removed
1: $D'' \leftarrow D$ but duplicate tuples (w.r.t the identifier) removed
2: $D''' \leftarrow AlgKAnon(D'', k)$
3: $D' \leftarrow D$ but quasi-identifiers of subjects are replaced with respective values from D'''
4: $D' \leftarrow D'$ but the identifier column removed

Theorem 2 *The generic method presented in Algorithm 1 correctly solves the MI k-anonymity problem.*

Proof D'' is same as D except duplicate tuples (w.r.t. the identifier feature) are removed. Due to the construction every subject has exactly one tuple in D''. As a result providing classical k-anonymity on D'' is same as the providing MI k-anonymity on D''. So, D''' has both classical k-anonymity and MI k-anonymity properties. The quasi-identifiers of left out tuples from D are then replaced with the quasi-identifiers of their respective anonymity sets and finally merged with D'''. Since adding new tuples to anonymity sets does not reduce its anonymity level the proof is complete. □

Let $O(CA)$ be the time complexity of the $AlgKAnon$, then the complexity of the method is $max\{O(|D|), O(CA)\}$. When D and D'' are of comparable size then the complexity becomes that of $AlgKAnon$. As a result, the practicality of the method reduces to the practicality of $AlgKAnon$.

Beyond the simple approach presented in Algorithm 1, one can devise methods that effectively uses tuple counts for better anonymization. Indeed, to this end we developed a method which is not provided here due to space limitations.

3 P-Certainty Anonymity Principle

Consider an MI 3-anonymity set of three subjects, s_1, s_2 and s_3; where there are 100 tuples of s_1, 2 tuples of s_2 and 1 tuple of s_3. Consider a randomly selected tuple out of the 103 tuples; clearly the tuple belongs to s_1 with probability $\frac{100}{103} \approx 1$. Thus the attacker is almost sure that the tuple belongs to s_1, as a result the privacy of s_1

is risked. The main problem unique to MI datasets here is the highly unbalanced instance count distribution within the anonymity set. Motivated with this, we define a new anonymity principle, called *p-certainty*.

Definition 3 (*p-certainty*) Given background knowledge B; a MID D' satisfies p-certainty anonymity principle (p-certain) iff;

- none of the tuples in D' can be linked, using B, to any subject with probability higher than p $(0 < p < 1)$.

Problem 2 (*p-certain data publication*) The problem is exactly the same as Problem 1 except the released table D' is required to have the p-certainty property rather than the MI k-anonymity property.

Let an anonymity set be formed of tuple set, RS with $|RS| = n$, belonging to subject set SS with $|SS| = m$. Let $IS(s)$ to denote the set of tuples belonging to the subject $s \in SS$, and $P(r, s)$ to denote the probability that $r \in RS$ belongs to the subject s. Clearly, $P(r, s) = 0, \forall r \in RS, \forall s \notin SS$ and $\sum_{s \in SS} P(r, s) = 1, \forall r \in RS$. This observation allows us to check and satisfy p-certainty for each anonymity set independently as articulated in the next proposition. So, starting from Lemma 2 we focus on providing p-certainty for a single anonymity set.

Proposition 1 *An MI database D satisfies p-certainty property iff every anonymity set satisfies p-certainty property.*

Lemma 2 *Let* $maxis = \max\{|IS(s)| : s \in SS\}$ *and* $|RS| = n$ *then p-certainty is satisfied in RS iff* $\frac{maxis}{n} \leq p$.

Proof This is simply because for any $s \in SS$, $\frac{|IS(s)|}{n} \leq \frac{maxis}{n} \leq p$. Hence, the probability that any randomly selected tuple in RS belongs to any subject SS is not greater than p. This implies that RS is p-certain. □

From the lemma it is very straightforward to check p-certainty property on any anonymity set.

Lemma 3 *Let* $maxis = \max\{|IS(s)| : s \in SS\}$, $|RS| = n$ *and* $\frac{maxis}{n} > p$ *then p-certainty is satisfied if at least* $t = \lceil \frac{maxis - p \cdot n}{p} \rceil$ *new tuples (from other subjects) are added to the anonymity set.*

Proof The new t tuples increases n by t and solving for the requirement $\frac{maxis}{n+t} \leq p$ gives the minimum t in the claim. □

Proposition 2 *From Lemma 2 and 3 we conclude that checking and minimally providing p-certain anonymity sets only require knowledge of* $maxis = \max\{|IS(s)| : s \in SS\}$ *and* $|RS| = n$. *In other words, maxis and n are sufficient.*

Proposition 2 is exploited in the next subsection to design an algorithm for p-certainty.

3.1 A Clustering Algorithm

We devise a clustering-based algorithm that can accommodate arbitrary subject distance functions. Let $SDist(s_1, s_2)$ be a distance function measuring the distance between subjects s_1 and s_2. The algorithm selects a pivot subject among free subjects in each iteration. The subject with the highest instance count is the heuristic for pivot selection. The objective with this heuristic is reducing average distortion per instance. An anonymity set around the pivot is created until the set has p-certainty property. This is done by including always the closest subjects (to the pivot) into the growing anonymity set. After p-certain anonymity sets are created, the final stage is generalizing (lines 27–28) every anonymity sets. Note that no generalization is done during the clustering process (lines 1–25) as this is not required since only the instance counts are enough to check p-certainty (Proposition 2). The algorithm runs in $O(n^2)$ time where n is the number of subjects. The unavoidable quadratic computational complexity originates from pairwise subject distances computation.

Algorithm 2 Clustering-based p-certainty

Input: D: A MID, p: anonymity parameter
Output: D': A MID satisfying p-certainty property

1: $PivotSet \leftarrow \emptyset$
2: $SS \leftarrow$ *subject set in* D
3: $Assigned(s) \leftarrow false, \forall s \in SS$
4: **while** $\wedge_{s \in SS} Assigned(s) = false$ **do**
5: $pivot \leftarrow arg\max_{s \in SS \wedge Assigned(s) = false} |IS(s)|$
6: $minTupleSize \leftarrow \lceil \frac{|IS(pivot)|}{p} \rceil$
7: $CandSubjs \leftarrow \{s \in SS \mid Assigned(s) = false\} \setminus \{pivot\}$
8: **if** $|IS(pivot)| + \Sigma_{s \in CandSubjs} |IS(s)| \geq minTupleSize$ **then**
9: $PivotSet \leftarrow PivotSet \cup \{pivot\}$
10: $Assigned(pivot) \leftarrow true$
11: $AnonymitSet(pivot) \leftarrow \{pivot\}$
12: $currsize \leftarrow |IS(pivot)|$
13: **while** $currsize < minTupleSize$ **do**
14: $closer \leftarrow arg\min_{s \in CandSubjs} SDist(pivot, s)$
15: $CandSubjs \leftarrow CandSubjs \setminus \{closer\}$
16: $Assigned(closer) \leftarrow true$
17: $AnonymitSet(pivot) \leftarrow AnonymitSet(pivot) \cup \{closer\}$
18: $currsize \leftarrow currsize + |IS(closer)|$
19: **else**
20: **if** $PivotSet = \emptyset$ **then**
21: **return** with no solution
22: **for each** $s \in SS \wedge Assigned(s) = false$ **do**
23: $closestpivot \leftarrow arg\min_{pivot \in PivotSet} SDist(pivot, s)$
24: $AnonymitSet(closestpivot) \leftarrow AnonymitSet(closestpivot) \cup \{s\}$
25: $Assigned(s) \leftarrow true$
26: $D' \leftarrow \emptyset$
27: **for each** $pivot \in PivotSet$ **do**
28: $D' \leftarrow D' \cup Generalize(AnonymitSet(pivot))$
29: **return** D'

4 Experimental Evaluation

4.1 Distortion Measure

Our approach to create anonymity sets is through quasi-identifier generalization. In this approach, a domain hierarchy for every quasi-identifier feature is needed and, by using the domain hierarchies, quasi-identifiers of subjects are generalized so as to obtain anonymity sets. Let $QI_D = \{qid_1, qid_2, \ldots, qid_n\}$ be the set of quasi-identifier attributes of D. The domain hierarchy on qid_i, $i = 1, 2, \ldots, n$ is denoted by DH_i. In domain hierarchies, the topmost level (the most general value, usually denoted by *) has depth of 0 and the level of other internal and leaf values are the tree depth at the value. By definition, the most specific values are located at the leaves and internal nodes generalize immediate children.

Definition 4 (*tuple distortion*) Let $t \in D$ be a tuple and $t' \in D'$ be its generalized version. The tuple distortion, denoted by $TDC(t, t')$, is the distance between quasi-identifiers of t and t' as given next.

$$TDC(t, t') = \Sigma_{qid_i \in QI_D} level(t[qid_i]) - level(t'[qid_i])$$

where $level(t[qid_i])$ denotes the level number of the value for t projected on quasi-identifier attribute qid_i.

Definition 5 (*database distortion*) Let D be a MID and D' be its anonymous version. The database distortion, denoted by $Distortion(D, D')$, is defined as;

$$Distortion(D, D') = \Sigma_{t \in D} TDC(t, t')$$

where $t' \in D'$ is the corresponding tuple of $t \in D$.

4.2 Datasets

Two datasets are used for experimentation. The first dataset is a synthetically created dataset, hereafter called $\mathcal{SYNTHETIC}$, having the same schema as the running example given in Fig. 1a. There are 5,000 subjects each having (uniformly randomly assigned) one to ten instances in the dataset. This resulted in 25,132 tuples in $\mathcal{SYNTHETIC}$. The domain of $Birth$, Zip, $Service\,quality$ and $Disease$ attributes contain 50, 50, 3 and 6 distinct values, respectively. Each subject is assigned a random $Birth$ and Zip values from the respective domains and each tuple (instance) is similarly assigned random $Service\,quality$ and $Disease$ values. For each of the quasi-identifier attributes a 3-level domain generalization hierarchies are built.

The other dataset is obtained from the *Adults dataset* (a.k.a. *Census income dataset*) [9]. Each tuple in the dataset refers to an individual and contains some identifiers removed sensitive/insensitive information. The dataset contains 48,842 tuples with 14 attributes but some of the tuples contain missing values. After removing tuples with missing values we end up with a complete dataset of 30,162 subjects. Since it is a MID, we further pre-process the dataset to obtain a MID. Our procedure is as follows. For each subject in the dataset, we generate one to five instance counts (uniformly randomly), resulting in 78,395 tuples (on average 2.6 instances per subject). We call this dataset $\mathcal{MI}-\mathcal{ADULTS}$ hereafter. We use 8 (age, work-class, education, marital-status, occupation, race, sex and native-country) of the 14 attributes as quasi-identifiers and the remaining 6 as sensitive/insensitive attributes. Following [13], we build domain generalization hierarchies as follows: 4-level on age, 3-level on work-class, 3-level on education, 3-level on marital-status, 2-level on occupation, 2-level on race, 2-level on sex and 3-level on native-country.

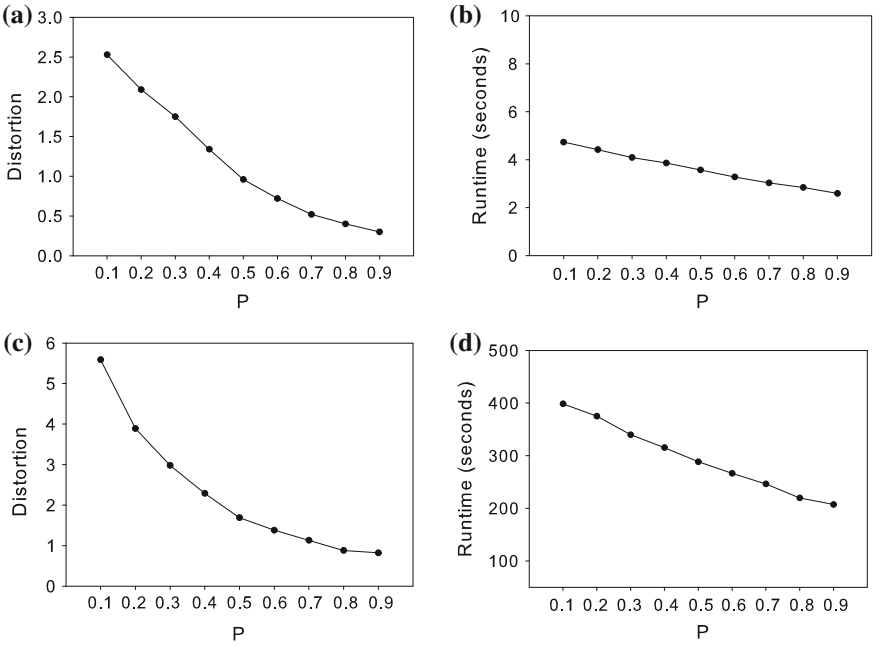

Fig. 2 Distortion empirical evaluation of p-certainty: **a** effectiveness on $\mathcal{SYNTHETIC}$, **b** efficiency on $\mathcal{SYNTHETIC}$, **c** effectiveness on $\mathcal{MI}-\mathcal{ADULTS}$, **d** efficiency on $\mathcal{MI}-\mathcal{ADULTS}$

4.3 Results

The results in Fig. 2 show the performance of Algorithm 2. The distortion gradually decreases as the p increases. This is an expected behavior as smaller values of p requires large size anonymity sets and hence potentially comes with much distortion. The runtime (the test computer is a PC with 3.0 GHz CPU-clock) also decreases with increasing p due to less number of iterations required.

5 Conclusion

Anonymization is a canonical solution to sensitive micro-data publication and k-anonymity is theoretically sound, well-defined and yet a simple anonymization principle. Because of this, it has been applied and studied in several contexts with extensions. The motivation of the current paper is the need for the anonymous data publication problem for MI databases. It is shown that the classical k-anonymity can not directly apply to MI datasets. Hence, a variant of (in fact a generalization to) classical k-anonymity is introduced. The proposed anonymity principle, called MI k-anonymity, is aimed at publishing MI databases such that every anonymity set has cardinality at least k subjects.

Some theoretical properties of MI k-anonymity are studied. The issue of solving MI k-anonymity problem in terms of classical k-anonymity is explored. It is shown that any algorithm solving classical k-anonymity problem can be used as a building block to solve the MI k-anonymity problem. Although the anonymity problem in MI datasets can be reduced to classical k-anonymity problem, there are more rooms for new algorithms that utilize instance counts.

When the instance counts of subjects within an anonymity set is highly unbalanced, it is shown that a new anonymity principle beyond the MI k-anonymity principle is needed. A novel anonymity principle, called p-certainty, is unique to MI databases as it is based on instance count distribution. A clustering-based algorithm is developed and experimentally evaluated on two datasets.

The future research will take the challenge of (1) developing more efficient/effec tive algorithms for both of the MI k-anonymity and p-certainty, (2) new privacy principles for MI micro-data publication, (3) customization of the proposed methods to other databases including spatio-temporal, sequential databases etc. beyond relational databases.

References

1. Abul O, Bonchi F, Nanni M (2008) Never walk alone: uncertainty for anonymity in moving objects databases. In: Proceedings of 24th IEEE international conference on data, engineering (ICDE'08)

2. Adam NR, Wortmann JC (1989) Security-control methods for statistical databases: a comparative study. ACM Comput Surv 21(4):515–556
3. Aggarwal G, Feder T, Kenthapadi K, Khuller S, Panigrahy R, Thomas D, Zhu A (2006) Achieving anonymity via clustering. In: Proceedings of 25rd ACM symposium on principles of database systems (PODS'06)
4. Aggarwal G, Feder T, Kenthapadi K, Motwani R, Panigrahy R, Thomas D, Zhu A (2005) Anonymizing tables. In: Proceedings of 10th international conference on database theory (ICDT'05)
5. Agrawal D, Aggarwal CC (2001) On the design and quantification of privacy preserving data mining algorithms. In: Proceedings of 20th ACM symposium on principles of database systems (PODS'01), pp 247–255
6. Agrawal R, Srikant R (2000) Privacy-preserving data mining. In: Proceedings of 2000 ACM SIGMOD international conference on management of data (SIGMOD'00), pp 439–450
7. Domingo-Ferrer J, Mateo-Sanz JM (2002) Practical data-oriented microaggregation for statistical disclosure control. IEEE Trans Knowl Data Eng 14(1):189–201
8. Garey MR, Johson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. Freeman, New York
9. Kohavi R (1996) Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid. In: Proceedings of 2nd international conference on knowledge discovery and data mining (KDD'96)
10. Kriegel H-P, Pryakhin A, Schubert M (2006) An EM approach for clustering multi-instance objects. In: Proceedings of 10th Pacific-Asia conference on knowledge discovery and data mining (PAKDD'06)
11. Kwok JT, Cheung P-M (2007) Marginalized multi-instance kernels. In: Proceedings of 20th international joint conference on artificial intelligence (IJCAI'07)
12. LeFevre K, DeWitt DJ, Ramakrishnan R (2005) Incognito: efficient full-domain k-anonymity. In: Proceedings of 2005 ACM SIGMOD international conference on management of data (SIGMOD'05), pp 49–60
13. LeFevre K, DeWitt DJ, Ramakrishnan R (2006) Mondrian multidimensional k-anonymity. In: Proceedings of 22nd IEEE international conference on data, engineering (ICDE'06)
14. Li J, Wong RC-W, Fu AW-C, Pei J (2006) Achieving k-anonymity by clustering in attribute hierarchical structures. In: Proceedings of 8th international conference on data warehousing and knowledge, discovery (DaWaK'06)
15. Li N, Li T (2007) *t*-closeness: privacy beyond k-anonymity and l-diversity. In: Proceedings of 23rd IEEE international conference on data, engineering (ICDE'07)
16. Machanavajjhala A, Gehrke J, Kifer D, Venkitasubramaniam M (2006) *l*-diversity: privacy beyond *k*-anonymity. In: Proceedings of 22nd IEEE international conference on data, engineering (ICDE'06)
17. Martin DJ, Kifer D, Machanavajjhala A, Gehrke J (2007) Worst-case background knowledge for privacy-preserving data publishing. In: Proceedings of 23rd IEEE international conference on data engineering (ICDE'07)
18. Meyerson A, Willliams R (2004) On the complexity of optimal k-anonymity. In: Proceedings of the 23rd ACM symposium on principles of database systems (PODS'04)
19. Nergiz M, Clifton C, Nergiz A (2007) Multirelational k-anonymity. In: Proceedings of data engineering, 2007. ICDE 2007, IEEE 23rd international conference on, pp 1417–1421
20. O'Leary DE (1991) Knowledge discovery as a threat to database security. In Piatetsky-Shapiro G, Frawley WJ (eds) Knowledge discovery in databases. AAAI/MIT Press, Cambridge, pp 507–516
21. Samarati P, Sweeney L (1998) Generalizing data to provide anonymity when disclosing information (abstract). In: Proceedings of 17th ACM symposium on principles of database systems (PODS'98)
22. Sweeney L (2002) k-anonymity: a model of protecting privacy. Int J Uncertainty Fuzziness Knowl Based Syst 10(5):557–570

23. Wong R, Li J, Fu A, Wang K (2006) (α, k)-anonymity: an enhanced k-anonymity model for privacy-preserving data publishing. In: Proceedings of 12th ACM SIGKDD international conference on knowledge discovery and data mining (KDD'06)
24. Xiao X, Tao Y (2007) m-invariance: towards privacy preserving re-publication of dynamic datasets. In: Proceedings of 2007 ACM SIGMOD international conference on management of data (SIGMOD'07)

Homomorphic Minimum Bandwidth Repairing Codes

Elif Haytaoglu and Mehmet Emin Dalkilic

Abstract To store data reliably, a number of coding schemes including Exact-Minimum Bandwidth Regenerating codes (exact-MBR) and Homomorphic Self Repairing Codes (HSRC) exist. Exact-MBR offers minimum bandwidth usage whereas HSRC has low computational overhead in node repair. We propose a new hybrid scheme, Homomorphic Minimum Bandwidth Repairing Codes, derived from the above coding schemes. Our coding scheme provides two options for node repair operation. The first option offers to repair a node using minimum bandwidth and higher computational complexity while the second one repairs a node using fewer nodes, lower computational complexity and higher bandwidth. In addition, our scheme introduces a basic integrity checking mechanism.

1 Introduction

The volume of digital data grows substantially. The nodes of the data storage systems are prone to failure due to disk failures, network failures, etc. Less costly maintenance of this huge amount of storage is a crucial problem. Lots of techniques with different cost saving policies exist to protect the data against failures. The most naive solution to this problem is replication. Basic weakness of replication is its storage inefficiency. Another solution, erasure codes, provides high data reliability with a small storage overhead. However, traditional erasure codes [7] use bandwidth as high as the original data size for a single node repair.

Bandwidth, I/O and computation costs are among the main design aspects for distributed storage systems. Different systems may emphasize one cost factor over

E. Haytaoglu (✉) · M. E. Dalkilic
International Computer Institute, Ege University, Izmir, Turkey
e-mail: elif.acar@ege.edu.tr

M. E. Dalkilic
e-mail: mehmet.emin.dalkilic@ege.edu.tr

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_33,

the others. For instance, the number of helper nodes affects the I/O overhead proportionally for cloud storage systems [5]. Thus, reducing the number of helper nodes and downloading more data from each can be more efficient in some cases [2] whereas in systems where bandwidth is the main bottleneck, a coding scheme which connects to more helper nodes and downloads less for a node repair can be more appropriate. Optimising only one cost parameter of a distributed storage system may be not enough, since many systems are not static and their requirements may change dynamically. If a storage system receives many requests in a certain period of time, then a coding scheme with less I/O overhead and low computational complexity can be more effective to achieve fast response time. The same system, in another time, may be not busy with user requests, then saving bandwidth can make sense.

We propose a new hybrid coding scheme that can provide either fewer helper node usage with lower computational complexity or lower bandwidth usage in node repair considering system needs at the service request time. In a way, we combine exact-MBR's [6] bandwidth efficiency and HSRC's [4] low computational complexity and fewer helper node usage in a node repair. The new scheme can require higher finite field size (can lead usage of more storage) than exact-MBR and HSRCs. However, the new coding scheme offers flexibility for node repair and thus can be more effective for storage systems in overall than using the other two coding schemes separately. This hybrid coding scheme can provide a basic integrity check mechanism for node repair against malicious nodes or bit errors, if necessary conditions exist.

2 Related Work

In [1], regenerating codes which can reduce the bandwidth for repairing a node by including more nodes in a repairing session are proposed. Regenerating codes provide an optimal trade-off between the minimum storage usage and the minimum bandwidth usage. Optimal exact-regenerating codes for $[n, k, d \geq 2k - 2]$ MSR (Minimum Storage Regenerating) and $[n, k, d]$ MBR codes through product-matrix construction are shown in [6]. These constructions are the first instances of exact-MBR and exact-MSR codes such that n does not depend on the other parameters. A new class of coding scheme namely SRC (Self-Repairing Codes) are proposed in [4]. An SRC node can be repaired from small number of nodes (2 to k nodes). An explicit implementation of such codes: HSRCs (Homomorphic Self Repairing Codes) are also provided in [4]. HSRCs are constructed based on the homomorphism property of the weakly linearized polynomials. The analyses in that work state that these codes provide almost the same static resilience as the traditional erasure codes. In [8], two exact-MBR constructions are proposed. Although these new constructions using linearized polynomials achieve low computational complexity in node repair, they have higher complexity in data reconstruction than ours and exact-MBR codes in [6].

3 Homomorphic Minimum Bandwidth Repairing Codes

In this section, we explain in detail the new hybrid coding scheme, Homomorphic Minimum Bandwidth Repairing codes (HMBR). An $[n, k, d]$ HMBR code encodes $B = \binom{k+1}{2} + k(d-k)$ symbols (finite field elements) and produces n fragments that each of them contains d symbols. n denotes the total node count, k denotes the required node count for data reconstruction and d is the number of nodes used in node regeneration in the regenerating codes. n, k, d values satisfy $k \leq d \leq n-1$. Node repair can be accomplished in two ways. In one, the newcomer node connects to d nodes and downloads one symbol from each of them. In the other, the newcomer node connects to a subset of nodes containing τ nodes where $\tau \in \{2, 3, \ldots, k\}$ and downloading d symbols from each. The code construction, node repairing, integrity checking, data reconstruction processes of HMBR codes are explained in this section. We use some notation that we have adopted from exact-MBR [6] and HSRCs [4]. A finite field $\mathbb{F}_\rho$ is used where maximum of $(2^d, (2^{(\lceil log_2 n \rceil + 1 - (\lceil log_2 n \rceil - \lfloor log_2 n \rfloor))}))$ for ρ is sufficient for HMBR codes.[1]

The message matrix construction is the same as in exact-MBR [6]. We have a symmetric $d \times d$ message matrix, M. It contains four different sub-matrices: S, T, T^t and a zero matrix. The first $\binom{k+1}{2}$ symbols of the original data are placed in the upper-triangular half of S. The lower-triangular elements of S are filled with the reflection of the upper-triangular half of S. The remaining $k(d-k)$ symbols are located on T. T^t matrix is transpose of T and placed between $(k+1)_{th} - (d)_{th}$ rows of M. The remaining $(d-k)^2$ elements of M are filled with zero. Finally, M is constructed as:

$$M = \begin{bmatrix} S & T \\ T^t & 0 \end{bmatrix} \tag{1}$$

In HMBR, we would like to efficiently use the homomorphism property, so the encoding matrix generation is related to the HSRCs' [4] encoding matrix which is based on weakly linearized polynomials.

Definition 1 Oggier and Datta [4] describe a weakly linearized polynomial, p(X), as follows:

$$p(X) = \sum_{i=0}^{k-1} p_i X^{2^i} \ , \ p_i \in \mathbb{F}_q, p(X) \in \mathbb{F}_q \text{ and } q = 2^m. \tag{2}$$

Lemma 1 *Oggier and Datta* [4] *also show that if* $a, b \in \mathbb{F}_{2^m}$ *and* $p(X)$ *be a weakly linearized polynomial, then the following equation is hold:*

[1] Field size of 2^d is enough for finding d linearly independent symbols (and also linearly independent rows). Also encoding matrix requires a finite field with order at least $n+1$ and for using the homomorphism property the field size must be a power of 2.

$$p(a+b) = p(a) + p(b)\,. \tag{3}$$

The proof of Lemma 1 is also provided in [4]. We construct d different weakly linearized polynomials using M's elements. Then, we evaluate these polynomials with n different inputs. The generic polynomial is as follows:

$$p_j(x) = \sum_{i=1}^{d} x^{2^{i-1}} M_{ij}\ ,\ \text{where } j \in \{1, 2, \ldots, d\} \text{ and } M_{ij} \in \mathbb{F}_\rho\ . \tag{4}$$

M_{ij} corresponds to the symbol in ith row and jth column of M. We evaluate $p_j(x)$'s for different inputs: $w_{[1]}, w_{[2]}, \ldots, w_{[n]} \in \mathbb{F}_\rho$. After evaluating d different $p_j(x)$s with n different inputs, the encoding process of HMBR codes is completed.

Evaluating $p_j(x)$s with n inputs, for all $j \in \{1, 2, \ldots, d\}$, corresponds to the multiplying M with an encoding matrix which is shown below:

$$E = \begin{bmatrix} w_{[1]} & {w_{[1]}}^2 & {w_{[1]}}^4 & \ldots & {w_{[1]}}^{2^{d-1}} \\ w_{[2]} & {w_{[2]}}^2 & {w_{[2]}}^4 & \ldots & {w_{[2]}}^{2^{d-1}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{[n]} & {w_{[n]}}^2 & {w_{[n]}}^4 & \ldots & {w_{[n]}}^{2^{d-1}} \end{bmatrix} \tag{5}$$

The first k columns of $(n \times d)$ matrix E is denoted as matrix Φ and the remaining $(d-k)$ columns of the encoding matrix is denoted as matrix Δ, as in [6] ($E = \begin{bmatrix}\Phi\ \Delta\end{bmatrix}$). Notice that, every subset of E with d rows is not linearly independent. Thus, E differs from the encoding matrix in exact-MBR codes [6]. This is HMBR's trade-off for providing locality in node repair. HMBR codes' encoding process is given by: $C = E \times M$. Here, each row of C is sent to a different node.

A failed node can be repaired in two different ways in this scheme. The first is an adaptation of regenerating process of exact-MBR codes in [6], named AMBR. The latter is an adaptation of repairing process of HSRCs [4], named ASRC. AMBR repairs a node by downloading one symbol from each of the d helper nodes such that these nodes' corresponding encoding matrix rows are linearly independent.

Let $(d \times d)E_{helpers}$ matrix be a subset of the rows of E such that only rows corresponding to the helper nodes are included.

Theorem 1 (AMBR Repair) *If a non-singular $E_{helpers}$ matrix can be constructed, then the failed node can be repaired by downloading d symbols in total using AMBR.*

Proof The code fragment stored in $node_i$ is shown with C_{node_i} (row vector) and $node_i$'s corresponding row of E is denoted as $E^t_{node_i}$. Let us call the failed node as f and the f's corresponding row of encoding matrix as E^t_f. The newcomer connects to d helper nodes ($node_x, node_y, \ldots, node_z$) where $E^t_{node_x}, E^t_{node_y}, \ldots, E^t_{node_z}$ are linearly independent. Each helper node i calculates the $C_{node_i}E_f$ and sends the result to the newcomer node. Then, the newcomer node constructs a matrix from those received

results. After receiving d symbols, the newcomer has:

$$E_{helpers}ME_f \ , \text{ where } E_{helpers} = \begin{bmatrix} E^t_{node_x} \\ E^t_{node_y} \\ \vdots \\ E^t_{node_z} \end{bmatrix} . \tag{6}$$

The failed node f has stored $C_f = E_f^t M$. So, if the newcomer node eliminates the $E_{helpers}$ from (6), it can regenerate C_f. Due to the linear independence of related encoding matrix rows, $E_{helpers}$ is invertible, then we can calculate the ${E_{helpers}}^{-1}E_{helpers}ME_f$ and get the value ME_f. Since M is symmetric, $(ME_f)^t = E_f^t M$ which is the node f's lost content. Therefore, AMBR downloads d symbols in total for repairing a node. Notice that, this repair method works only if any d nodes exist in the system with linearly independent encoding matrix rows.

The second approach, ASRC, is an adaptation of the repairing process of HSRCs. This approach requires downloading τd symbols in total.

Theorem 2 (ASRC Repair) *We can repair the failed node by connecting τ special nodes (satisfying (7)) and downloading τd symbols in total where $\tau \in \{2, 3, \ldots, k\}$.*

Proof In this repairing method, we have to find special τ nodes which satisfy (7).

$$\sum_{i=1}^{\tau} w_{[node_i]} = w_{[f]} \text{ where } k \geq \tau \geq 2 \ . \tag{7}$$

In (7), $w_{[node_i]}$ denotes the polynomial input of $node_i$ and $w_{[f]}$ denotes the polynomial input of the failed node. Assume that $\tau = 2$. The failed node f has d $p_j(x)$s; with the input of $w_{[f]}$. If any pair of nodes $(node_a, node_b)$ satisfying (7) exists, the newcomer node downloads all symbols stored in these nodes (d symbols from each of them). Upon receiving a total of $2d$ symbols from $node_a$ and $node_b$, the newcomer node has $2d$ polynomials of the form:

$$p_j(w_{[l]}) = \sum_{i=1}^{d}((w_{[l]})^{2^{i-1}}M_{ij}) \ , \text{ where } j \in \{1, 2, \ldots, d\} \text{ and } l \in \{node_a, node_b\} \ . \tag{8}$$

$C_f = E_f^t M$ and jth symbol of C_f is $E_f^t M_{*,j} = p_j(w_{[f]})$ where $M_{*,j}$ is jth column of M. Since $w_{[f]} = w_{[node_a]} + w_{[node_b]}$, then $p_j(w_{[f]}) = p_j(w_{[node_a]}) + p_j(w_{[node_b]})$ from Lemma 1. If there are no such two nodes , we search a subset of nodes with minimum cardinality satisfying (7) ($\tau > 2$). Thus, the repairing process becomes evaluating $p_j(w_{[f]}) = \sum_{i=1}^{\tau} p_j(w_{[node_i]})$ for all $j \in \{1, 2, ..., d\}$. Through ASRC, we can repair the failed node content by downloading d symbols from each of the special τ helper nodes (τd bandwidth usage) and adding them over $\mathbb{F}_\rho$. □

Theorem 3 *The probability of a successful repair in* $[n, k, d]$ *HMBR coding scheme is at least the probability of a successful repair in* $[n, k]$ *HSRCs when both schemes use the same field size and probabilities of the node availabilities in both systems are same and independent and identically distributed (i.i.d).*[2]

Proof In ASRC, the probability of a successful repair depends on the probability of finding τ nodes which can construct the failed node content. If the HMBR and HSRC codes use the same finite field size, then HSRC's one fragment and HMBR's one symbol are represented in the same number of bits. Thus, probability of repairing one symbol in ASRC is the same as the probability of repairing one fragment in HSRCs. Since they have the same number of bits and n, k; finding the probabilities of such τ nodes is the same for both. In ASRC, if τ helper nodes can repair the failed node's one symbol, these helper nodes can also repair the failed nodes' remaining $d-1$ symbols, since polynomials' inputs of one node are the same. Additionally in HMBR, if ASRC fails, the lost node may be repaired by AMBR. □

Let node f fails and the newcomer node connects to d helper nodes, downloads $C_{node_i}E_f$ from each helper $node_i$ and repairs content $C_f = E_f^t M$ through AMBR.

Theorem 4 (Integrity Checking) *After repairing* C_f *with AMBR, the newcomer node can check* C_f*'s integrity by confirming the validity of* $E_f^t C_f^t = \sum_{i=1}^{\tau}(C_{node_i}E_f)$*, if any subset of the helper nodes with cardinality* τ *satisfies* $\sum_{i=1}^{\tau} w_{[node_i]} = w_{[f]}$ *where* $w_{[node_i]}$ *is* $node_i$*'s polynomial input.*

Proof $C_f^t = ME_f$ since M is a symmetric matrix and

$$\sum_{i=1}^{\tau} C_{node_i}E_f = \sum_{i=1}^{\tau} E_{node_i}^t ME_f = \left(\sum_{i=1}^{\tau} E_{node_i}^t\right) ME_f = E_f^t ME_f = E_f^t C_f^t. \quad (9)$$

$\sum_{i=1}^{\tau} E_{node_i}^t = E_f^t$ due to the homomorphism. Thus, the newcomer node can check C_f's integrity by selecting a set of helper nodes which contains such τ nodes. This integrity scheme can be used only in AMBR if such τ nodes exist in the helper nodes. □

HMBR codes' data reconstruction process is very similar to that of exact-MBR codes in [6].

Theorem 5 (Data-Reconstruction) *If the data-collector connects to* k *nodes such that these nodes' corresponding rows of* Φ *are linearly independent, then all* B *symbols can be reconstructed.*

Proof Let the data-collector selects k nodes such that these nodes' corresponding rows of Φ are linearly independent and downloads all symbols stored in these nodes. After completion of all downloads, the data-collector node has the following matrix:

[2] Notice that, $[n, k, d]$ HMBR encodes and stores more amount of data than $[n, k]$ HSRC, when they use the same field size.

$$E_{DC}M = \left[\Phi_{DC}S + \Delta_{DC}T^t \ \Phi_{DC}T\right]. \tag{10}$$

E_{DC} is consisted of these k nodes' corresponding encoding matrix rows and Φ_{DC} is consisted of the first k columns of E_{DC}. Since Φ_{DC} is a non-singular matrix (due to the selection of connected nodes), we can obtain the Φ_{DC}^{-1}. The data-collector node calculates the $\Phi_{DC}^{-1}\Phi_{DC}T$ and gets T, then computes $\Delta_{DC}T^t$ and subtracts it from $\Phi_{DC}S + \Delta_{DC}T^t$. The data-collector has $\Phi_{DC}S$, by calculating $\Phi_{DC}^{-1}\Phi_{DC}S$, it can get S. The data reconstruction can be achieved if the data-collector can find k nodes such that these nodes' corresponding rows of Φ are linearly independent. □

Theorem 6 *The probability of achieving data reconstruction in* $[n, k, d = k \geq 2]$ *HMBR code is not less than that of* $[n, k \geq 2]$ *HSRCs provided that the coding schemes use the same field size and probabilities of the node availabilities in both schemes are the same and i.i.d.*[3]

Proof The polynomials contain original data symbols as coefficients, in both coding schemes. In this case ($d = k$), both schemes have polynomials of degree 2^{k-1}. A polynomial having degree 2^{k-1} can be reconstructed from $2^{k-1} + 1$ points using Lagrange interpolation. If we can find k linearly independent code symbols (over $\mathbb{F}_2$), we can construct $2^k - 1$ different points in total using homomorphism as stated in [4]. $2^{k-1} + 1$ different points can be obtained from k points $(x, p_j(x))$ with linearly independent inputs (x) over $\mathbb{F}_2$. $R(x, d, r)$ [4] [4] function gives the same result in both schemes. In addition, HMBR code has to reconstruct d different polynomials whereas HSRC has to reconstruct only one. In HMBR, if the data-collector node can reconstruct the polynomial $p_1(x)$ using k different node's $(x, p_1(x))$ points, it can also reconstruct the remaining $d - 1$ $p_j(x)$s using the same k node's $(x, p_j(x))$ points where $j \in \{2, 3, ..., d\}$, since each of the connected k nodes stores d symbols which are encoded with the same inputs. data-collector node can generate $2^{k-1} + 1$ points for $p_1(x)$ with k different x, $p_1(x)$ points, it can also generate $2^{k-1} + 1$ different combinations for other $d-1$ $p_j(x)$s where $j \in \{2, 3, ..., d\}$, since each of the connected k nodes stores d symbols which are encoded with the same inputs. Therefore, if k nodes exist such that their polynomial inputs are linearly independent over $\mathbb{F}_2$, all symbols stored in these nodes guarantees the reconstruction of d polynomials. Thus, both schemes guarantee the data reconstruction, if any k nodes exist in the system such that their polynomial inputs are linearly independent. □

HMBR codes' data reconstruction process is very similar to that of exact-MBR codes in [6].

[3] Notice that, $[n, k, d = k \geq 2]$ HMBR encodes and stores more data than $[n, k]$ HSRC, when they use the same field size.

[4] Here, d denotes the symbol size in bits and $R(x, d, r)$ function counts the number of $x \times d$ binary sub-matrices having rank r [4]. In HMBR, $R(x, d, r)$ can be used for counting all possible live node permutations having at least k linearly independent polynomial inputs.

4 Computational Complexity Analysis

To analyse the computational complexities, we encode the same number of symbols in the three coding schemes using minimum required field sizes (considering symbols' vector representations over $\mathbb{F}_2$). Base field sizes of exact-MBR [6], HSRC [4] and HMBR are $2^{\lceil logn \rceil}$, $2^{((\frac{k+1}{2}+(d-k))\lceil logn \rceil)}$, $\rho = max(2^d, (2^{(\lceil logn \rceil+1-(\lceil logn \rceil-\lfloor logn \rfloor))}))$, respectively. Due to the lack of space, we show below only the complexity of the one step that dominates the order of data reconstruction and node repair operations.

The data reconstruction process of $[n, k, d]$ exact-MBR codes [6] requires getting the inverse of $k \times k$ Φ_{DC} having $O(k^3)$ additions and multiplications. Reconstruction of T requires multiplying $k \times k$ matrix with $k \times (d-k)$ matrix having $O(k^2(d-k))$ additions and multiplications. In the node repair process of $[n, k, d]$ exact-MBR codes, the newcomer node gets the inverse of $(d \times d)$ matrix, Ψ_{repair} [6], having $O(d^3)$ additions and multiplications. In $[n, k]$ HSRCs [4], the data reconstruction can be achieved as in the traditional erasure codes, if there are k nodes having linearly independent encoding matrix rows. This process has $O(k^3)$ additions and multiplications, since an inversion on a $k \times k$ matrix is conducted. In $[n, k]$ HSRCs, to repair a lost node, $O(\tau)$ additions on a set of τ fragments are performed. τ can be at most k so $O(\tau)$ is $O(k)$. These operations are performed over $\mathbb{F}_{2^{\frac{B\lceil log_2 n \rceil}{k}}}$, since the same bits with exact-MBR coding scheme, $B\lceil log_2 n \rceil$ bits are encoded then the minimum required field size is $2^{\frac{B\lceil log_2 n \rceil}{k}} = 2^{((\frac{k+1}{2}+(d-k))\lceil log_2 n \rceil)}$.

In $[n, k, d]$ HMBR, B symbols are also encoded. But, in HMBR the symbol size can be more than $\lceil log_2 n \rceil$ bits, due to the required field size.HMBR data-collector node gets the inverse of a $k \times k$ matrix (Φ_{DC}) which has $O(k^3)$ multiplications and additions, then operates a matrix multiplication between $k \times k$, ${\Phi_{DC}}^{-1}$, and $k \times (d-k)$, $\Phi_{DC}T$, matrices having $O(k^2(d-k))$ additions and multiplications. Since HMBR codes have two repairing methods, we analyse them separately. In AMBR, the newcomer node calculates the ${E_{helpers}}^{-1}$ matrix having $O(d^3)$ additions and multiplications. As for ASRC, the newcomer node connects to τ nodes and gets d symbols from each of them. Upon getting all fragments, the newcomer node makes $O(\tau d)$ additions. τ is also at most k. In $\mathbb{F}_{2^m}$, adding two symbols requires $O(m)$ $\mathbb{F}_2$ operations and multiplying two symbols requires $O(m^2)$ $\mathbb{F}_2$ operations [3]. We include these complexities in the analyses of the coding schemes in Table 1.[5] In data reconstruction, HMBR codes' computational complexity can be equal to or higher than that of exact-MBR codes. HMBR codes' computational complexity can be equal to, or lower or higher than that of HSRCs, in data reconstruction. One can easily find appropriate $[n, k, d]$ values for the above three cases. HMBR codes' ASRC method has lower computational complexity than exact-MBR codes' repair process in all cases. The complexity of HMBR codes' AMBR method can be equal to or higher than the complexity of exact-MBR codes' repair process according to $O(log_2 n)$ and $O(d)$ values. In node repair, HMBR codes' ASRC method's complexity can be equal

[5] If the same finite field size is used in all coding schemes, while comparing the complexities of the coding schemes, we can ignore the time taken by the finite field operations shown in Table 1.

Table 1 Complexity of node repair and data reconstruction operations

	Reconstruction	Repair	
Exact-MBR [6]	Multip. $O(k^3(logn)^2) + O(k^2(d-k)(logn)^2)$	$O(d^3(logn)^2)$	
	Addition $O(k^3(logn)) + O(k^2(d-k)(logn))$	$O(d^3(logn))$	
HSRC [4]	Multip. $O(k^3((\frac{k+1}{2} + (d-k))(logn))^2)$	0	
	Addition $O(k^3((\frac{k+1}{2} + (d-k))(logn)))$	$O(\tau((\frac{k+1}{2} + (d-k))(logn)))$	
		AMBR	ASRC
HMBR	Multip. $O(k^3(log\varrho)^2) + O(k^2(d-k)(log\varrho)^2)$	$O(d^3(log\varrho)^2)$	0
	Addition $O(k^3log\varrho) + O(k^2(d-k)log\varrho)$	$O(d^3log\varrho)$	$O(d\tau log\varrho)$

to or higher than that of HSRCs' with respect to the relative orders of $O(log_2 n)$ and $O(d)$. Also, HSRCs always have lower computational complexity than HMBR codes' AMBR method, in node repair.

5 Conclusions

Complexity analyses show that HMBR codes are more efficient in systems where node repairs occur more frequently than data reconstructions. HMBR codes enable repairing a node in two different ways and to check the integrity of repaired data as explained. This coding scheme can be appropriate for systems consisting of heterogeneous nodes that have different bandwidth and computational capacities. Therefore, nodes having low bandwidth and high computational capacity may use AMBR, whereas nodes having high bandwidth and low computational capacity may use ASRC method for repair. If many node failures exist, but the probability of losing all data is still low, repairing the lost nodes with less bandwidth consumption through AMBR method may be advantageous, since it can prevent congestions. But, if the system has lots of failed nodes such that the probability of losing all data is significant, the lost fragments must be repaired as fast as possible before entire data is lost. In this case, bandwidth saving is of secondary importance and the lost fragments must be repaired before more failures occur. Hence, using ASRC approach for repairing nodes may be advantageous due to its low computational complexity.

Acknowledgments We would like to thank Frédérique Oggier and Rashmi K. Vinayak for kindly answering our many questions. This study was supported by ÖYP research fund of Turkish Government No:05-DPT-003/35.

References

1. Dimakis AG, Godfrey PB, Wainwright MJ, Ramchandran K (2007) Network coding for distributed storage systems. In: INFOCOM 2007. 26th IEEE international conference on computer communications. IEEE, pp 2000–2008
2. Gopalan P, Huang C, Simitci H, Yekhanin S (2012) On the locality of codeword symbols. Inf Theor IEEE Trans 58(11):6925–6934
3. Menezes AJ, Vanstone SA, Oorschot PCV (1996) Handbook of applied cryptography, 1st edn. CRC Press, Inc., Boca Raton
4. Oggier F, Datta A (2011) Self-repairing homomorphic codes for distributed storage systems. In: INFOCOM, 2011 proceedings IEEE, pp 1215–1223
5. Papailiopoulos DS, Dimakis AG (2012) Locally repairable codes. In: information theory proceedings (ISIT), 2012 IEEE international symposium on, pp 2771–2775
6. Rashmi KV, Shah NB, Kumar PV (2011) Optimal exact-regenerating codes for distributed storage at the msr and mbr points via a product-matrix construction. Inf Theor IEEE Trans 57(8):5227–5239
7. Reed IS, Solomon G (1960) Polynomial codes over certain finite fields. J Soc Ind Appl Math 8(2):300–304
8. Xie H, Yan Z (2012) Exact-repair minimum bandwidth regenerating codes based on evaluation of linearized polynomials. CoRR abs/1203.5325

Recreating a Large-Scale BGP Incident in a Realistic Environment

Enis Karaarslan, Andres Garcia Perez and Christos Siaterlis

Abstract The Internet has become a critical asset for both the economy and the society. Realistic experimentation environments are needed to study and improve the resilience and the stability of the Internet. In this paper, we propose a methodology that allows to: (1) model an Internet-like topology, (2) recreate the model with realistic parameters and conditions, (3) reproduce large-scale incidents, and (4) test various what-if scenarios. As a prood of concept, a valid abstraction of the Europe Internet backbone is created where Network Service Providers (NSP) are connected to each other in various Internet Exchange Points (IXP). This topology is emulated on a Emulab-based testbed. A well-known BGP-route hijacking incident is replayed and studied under hypothetical scenarios of network operators reactions and collaboration. The results of the experiments are then analysed showing the potential value of the proposed methodology.

1 Introduction

Public authorities characterize the Internet as a Critical Information Infrastructure (CII) due to its importance for the economy and the society. The protection of CII can be studied by injecting faults and disruptions into real systems, software simulators or hardware emulators. Experimentation with production systems in extreme conditions entails the risk of side-effects. Emulation approaches, like those using the Emulab software [3, 9], are very popular in the field of security and resilience analysis [6], since in order to study those attributes a researcher has to

E. Karaarslan (✉)
Department of Computer Science, Mugla Sitki Kocman University, Mugla, Turkey
e-mail: enis.karaarslan@mu.edu.tr

A. Garcia Perez · C. Siaterlis
JRC Institute for the Protection and Security of the Citizen, Ispra, Italy
e-mail: andresperezgarcia@gmail.com and christos.siaterlis@jrc.ec.europa.eu

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_34,

expose the system-under-test to high load and extreme conditions, under which, software simulators fail to capture reality. The main challenge, even in such approaches, is that the experiments are not realistic enough in terms of network topology, network conditions and background traffic. In this study, we make a first step towards addressing this challenge by sketching a simple methodology of how to conduct a testbed-driven experiment in order to study real cyber-security incidents.

As a proof of concept we demonstrate the steps that we followed in order to form a realistic platform where different scenarios of a replayed security incident can be tested. First, an abstraction of the EU Internet backbone which includes the biggest Network Service Providers (NSP) and Internet Exchange Points (IXP) is defined. The topology is recreated on top of an Emulab-based testbed, including real conditions of link delays, routing protocols and dynamic updates. Then, the infamous YouTube BGP incident [7] was replayed. Using a simple network operator decision-process simulator, we study how operators countermeasures, like filtering, and the collaboration with peers, would affect the incident recovery process.

2 Developing Models for the Incident Environment

Recreating a real security incident inside an emulation testbed requires mainly two models: (a) an Internet topology abstraction; (b) simple model for network operators.

2.1 Related Work

Simulating internet or large scale networks is a very wide topic, which has been explored by numerous studies. Simulation of large scale networks is discussed in [8]. Large-scale testing of BGP and graph reduction on BGP graphs was discussed in [2]. Realistic, large scale, Internet-like BGP simulations were studied in previous studies but resilience, stability and security were not studied in detail. Lad et al. [5] conducted an interesting study on the resiliency of the Internet topology. IXPs were covered in academic studies like [1, 4, 11]. CAIDA has AS based graphs, Euro-IX shows IXP locations on maps. [1] To our knowledge, combining IXP, NSPs and making an emulation with these components was not done before.

2.2 Network Model

Internet is a network of interconnected networks. The Border Gateway Protocol (BGP) is used for inter-domain routing. BGP identifies networks which are under a common management as Autonomous Systems (AS) with a unique Autonomous

[1] https://www.euro-ix.net/location-of-ixps

System Number (ASN). According to the CAIDA dataset,[2] there are 36.878 ASN out of which 12.142 are in Europe. Network Service Providers (NSP) are organizations which provide direct access to the Internet. The biggest transit-free NSPs that can reach all other networks are called Tier-1 and they typically peer with every other Tier-1 network [10]. Internet Exchange Points (IXP) are the aggregation points where networks peer with each other with high speed links. According to Euro-IX the number of IXP is large, e.g., 125 in Europe. It is clear that the real topology cannot be simply recreated inside an emulation testbed.

The aim of the model is to abstract the complexity of a real topology so that the real incident environment can be recreated on an emulation testbed. The model construction process can be summarized as:

- *Step 1—Selecting and Aggregating Data Sources* The type of datasets that were selected as inputs are:
  - *AS-level Topology*: information about AS from the Internet BGP routing table were retrieved from the CAIDA AS Rank dataset and further refined by RIPE records[3] (e.g., correcting country labels).
  - *Geographical Topology*: the Euro-IX IXP Database[4] was used to define the European IXPs and the list of AS which they host.
  - *BGP Routing Data*: the dynamics of BGP routes, which are recorded mainly at IXPs in MRT format (RFC 6936), were obtained from the Routeviews Project and RIPE RIS.[5]
- *Step 2—Processing Data and scaling network model* Implementing realistic simulations does not mean that every real node should be used; since the testing platform will not have sufficient resources, a sample which represents the system should be selected [2]. The real topology data were organized in a database so that a topology can be constructed dynamically given the scaling requirements. Python scripts were written directly form an Emulab configuration.

Given the scale limitations on the available testbed, the model is focused in Europe and consists of 16 nodes, as shown in Fig. 1. The topology includes the 12 biggest Tier-1 NSPs which are present in the biggest six IXPs of Europe. Each NSP is presented as a separate node which has several network interfaces. Each interface of the node represents its connection to a specific IXP. Three nodes were added due to their role in the incident of our case study. These nodes are AS 5511, Youtube, Pakistan Telecom and AS3491 (PCCW Global). This topology represents 49 physical routers which are located in 6 different IXPs. IXPs are represented as Local Area Networks which have several NSPs connected to them. Each NSP peers with all NSPs in a common IXP, these IXPs are shown as connected to each other by the common AS they have.

---

[2] The CAIDA AS rank dataset http://as-rank.caida.org/data/2011.01/

[3] RIPE. ftp://ftp.ripe.net/ripe/stats/delegated-ripencc-latest

[4] Euro-IX IXP database, 2012. https://www.euro-ix.net/tools/asn_search

[5] RIPE. http://www.ripe.net/data-tools/stats/ris/ris-raw-data

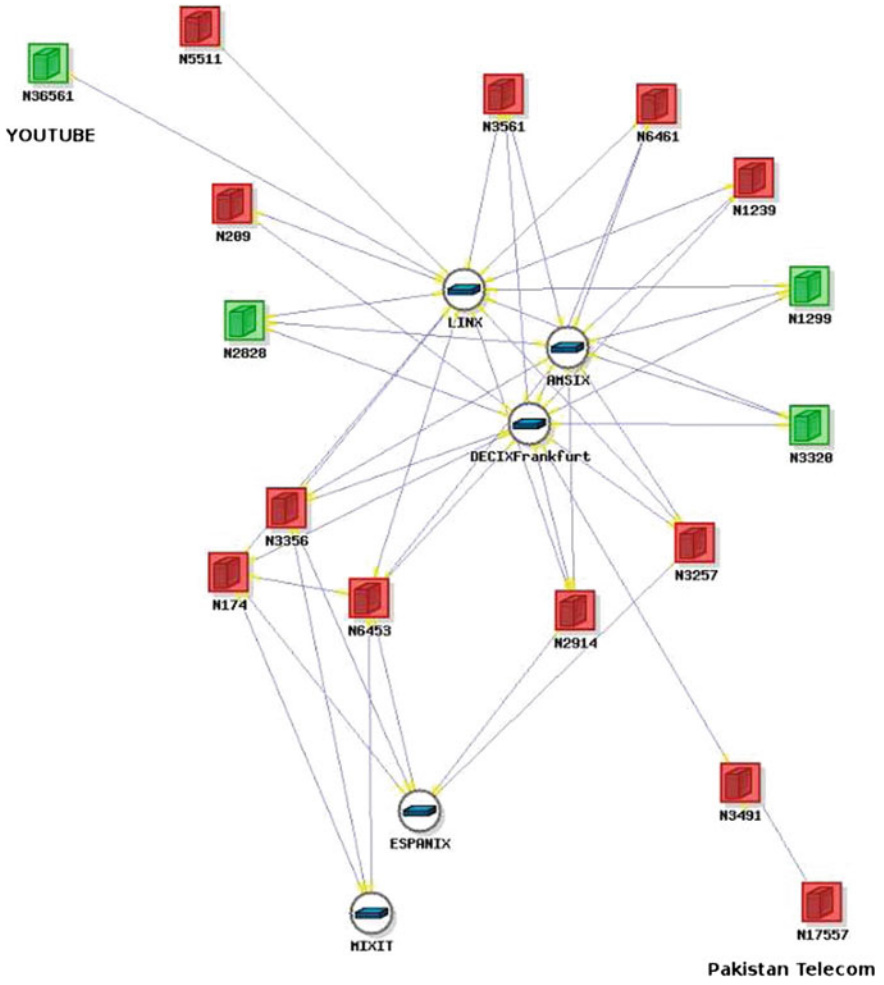

Fig. 1 Network topology model enhanced with a visualization of the attacked AS reachability

2.3 Network Operator Decision Model

Network operator decisions play an important role during security incidents. Beyond the various BGP vulnerabilities (RFC 4272), one of the most serious threats is the prefix hijack attack. In this attack, an attacker announces IP prefixes of another network into the global routing system either by a mistake or intentionally. Some hijack prevention/detection solutions are possible [5], but most of these solutions are complex to deploy and increase the workload of the routers. In our network operator model we consider this class of attacks and therefore we developed a simple program to simulate the operator's behavior during a BGP hijacking incident.

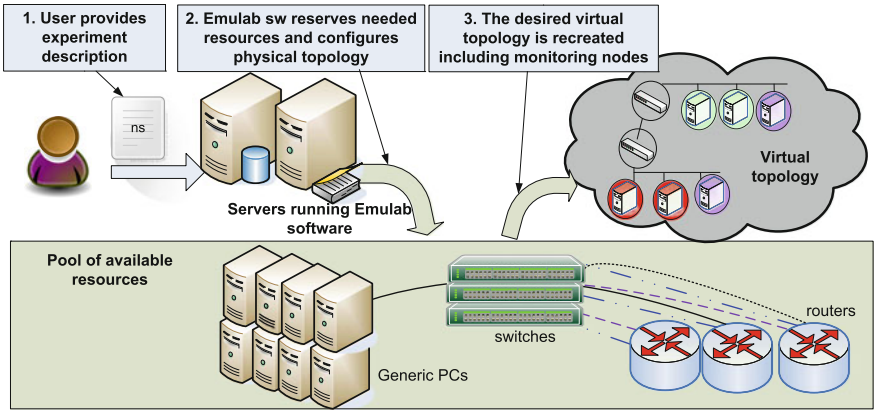

Fig. 2 Emulab environment

During normal work an operator may discover an incident by different means such as a complaint or information from a customer. We define this as Probability of Local Discovery (P1). Given that a network operator knows about the existence of the incident, the following three actions are considered:

- no further actions are taken;
- local countermeasures are applied with a Probability of Blocking (P2);
- the peers of the NSP are informed about the incident with probability of Notification (P3).

A communication channel is established so that simulated operators can communicate and share their knowledge. In the case of Peer—Remote Notification (P3) an operator writes the discovered event into the shared communication channel so that the other peers get notified. Finally in the case of a blocking action at the NSP, the simulated operator changes the local BGP configuration of the relevant router so that the wrong prefix learned from the peers gets blocked.

3 Recreating the Incident Environment on a Testbed

In this section, we describe how we recreate based on the previous models the incident environment on a testbed. An Emulab testbed [3] typically consists of two servers that run the Emulab software and a pool of physical resources used as experimental nodes. The Emulab software configures the physical components based on a experiment description, so that they emulate the virtual topology as transparently as possible (See Fig. 2). The important point in this case is that the resulting experiment environment consists of real interconnected devices, e.g. servers and routers.

The main steps to instantiate the incident environment are:

- *Network Setup* The topology is recreated based on the previously described network model. Each NSP node is a Quagga software router on a Freebsd server. The BGP configuration is generated by scripts during the node setup so that all the nodes have their ASN and peers configured automatically. Each node runs also an independent instance of the network operator simulator.
- *Delay Implementation* The network topology is enhanced with realistic delays. Fiber optic latency was considered as main source of delay under the assumption of direct fiber paths. Geographical distance between IXP locations was calculated and the delay was estimated by multiplying distance with the latency value. The delay between each IXP pair was then configured in each NSP node using the Dummynet software. Since the delay applied to a packed depends on the input and output interface, the interface tagging feature of ipfw was required.
- *Inject Full/Partial Internet Routing Table* The aforementioned network is made more realistic by replaying real routing data. The Internet routing table is injected in the beginning of the experiment. The MDFMT Suite and RIPE BGP update dataset were used to learn IP addresses of the NSPs. Using the Bgpdump tool the MRT formatted data are converted into readable form and the Bgpsimple perl script is used to inject the routing data (of the incident day) to the network. The whole Internet routing table (241,329 prefixes; 3,152,622 routes) was injected and the more than 100,000 route updates were injected through AS 5511.
- *Visualization and Monitoring* Visualization is an important element for experiment monitoring. In this case we monitored if the Youtube AS is reachable from each NSP. A dynamically updated network view enhanced with reachability information was created inside a Zabbix monitoring server (Fig. 1). If anode can reach YouTube, then it is shown as green otherwise it is colored red.

4 Replaying the BGP Hijacking Incident

The Youtube incident [7] is a good example where a misconfiguration by a single NSP, i.e., Pakistan Telecom, can lead to global reachability problem which may last several hours. In this section we show as a proof of concept an indicative set of simple questions that one could answer by replaying the real BGP incident data on the reconstructed incident environment. The experiments were carried out under three main assumptions: (1) both Youtube or Pakistan Telecom do not act; (2) there is no strict policy for operator reaction, i.e., they independently decide to block or inform their peers; (3) the probability of blocking the hijacking prefix (P2) is high.

In all experiments, we measured the Total Convergence Time (TCT), which is defined as the time needed for all NSPs (nodes) to learn about the incident and apply countermeasures (prefix filtering). Two particular scenarios were studied. Since our network operator model involves a random element (the probabilities P1, P2 and P3) for each scenario we repeat the same experiment 10 times. Our results then refer to the min, max and average values across these 10 independent runs.

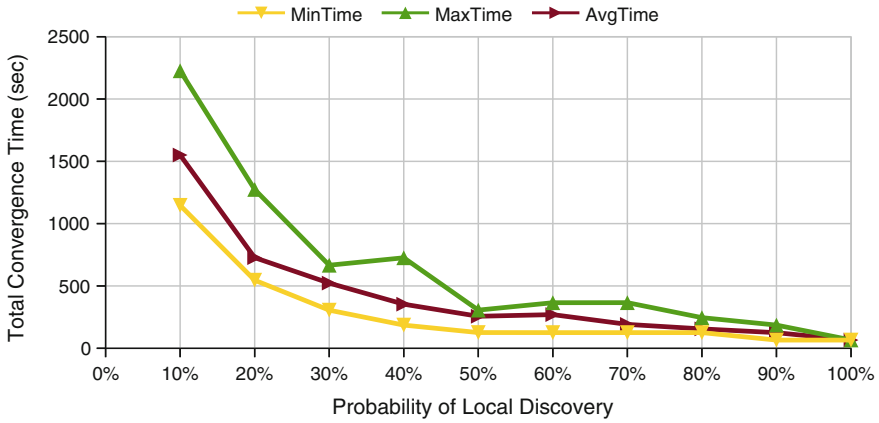

Fig. 3 Total convergence time with different P1 values (P2: 70 %, P3: 0 %)

In the first scenario, the nodes have to discover the disruption by themselves and there is no cooperation. Obviously, the probability of local discovery (P1) has then a big impact on TCT. When there is no probability of (peer—remote) notification (P3), the effect of P1 on TCT is shown in Fig. 3. It should be noted that in reality P1 depends on many factors such as operators qualifications and the existence of adequate discovery mechanisms (monitoring tools, etc.).

The second scenario assumes cooperation between NSPs. The NSPs (nodes) which get notified about the incident either by local discovery or remote notification, may notify their peers with probability P3. The effect of P3 on TCT when P1=10 % is shown in Fig. 4. When P3 increases, TCT decreases fast as expected. When P1 and P3 are low, there can be some nodes which respond late, and this results a longer TCT. When the P1 is low, P3 value should be at least 20 % for a fast convergence. Actually even a small change of P3 value makes a big difference on the average TCT. In the case of intense collaboration, for example with the existence of automated alert mechanisms, we can expect very fast responses. For example the network converges to the normal state at least 10 times quicker when P3 is 70 %.

Finally we show for this second scenario how the peer notification process evolves over time. When the probability of local discovery is low (P1=10 %) and given high probabilities of blocking (P2=70 %) and peer notification (P3=70 %), the NSPs (nodes) converge quickly to the normal state. Actually after the first 3 or 4 nodes realize the incident the network returns to the normal state. These results shows the important impact of peer notification to the incident handling. The experiment results showing the number of nodes in blocking state (in 10 different experiment runs) are given in Fig. 5.

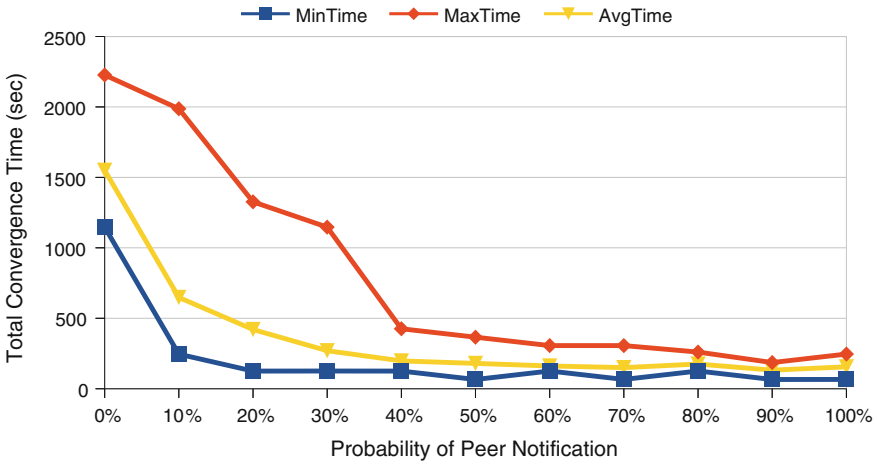

Fig. 4 Total convergence time with different P3 values (P1: 10 %, P2: 70 %)

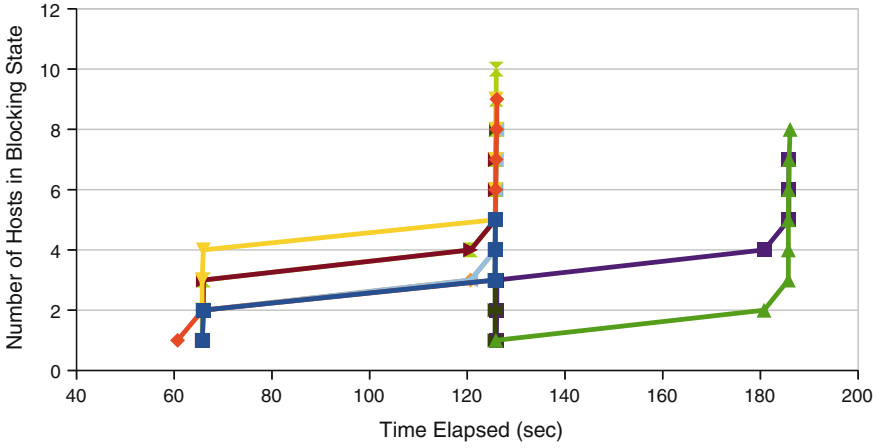

Fig. 5 Number of nodes in blocking state in 10 runs (P1: 10 %, P2: 70 %, P3: 70 %)

5 Conclusion and Future Work

This study constitutes a first small step towards the development of a methodology for the recreation of real world cyber-security incidents inside an emulation testbed with the long term goal of experimental what-if analysis. This goal entails many challenges since any experimental platform is bound to be far away from operational reality. Although addressing these challenges will require considerable future work, the following small steps forward were taken: a simple network abstraction was developed, this topology was then transfered on a Emulab-based testbed, and finally

a well-known BGP-route hijacking incident was recreated. The successful recreation of the incident validated the approach and the methodology.

Furthermore, the replayed BGP hijacking incident was studied under different hypothetical scenarios of operators reactions and coordination. The results of the experiments showed the huge effect of communication and coordination between NSPs in the event of such incidents, e.g., the importance of mechanisms for a fast discovery, the need for well trained operators and the positive effect of a trusted NSP communication channel. Although the results of this simple study are in-line with the expected outcomes, they are showing the potential value of the proposed methodology in order to experimentally test more complex what-if scenarios. Future work includes the increase of experiment fidelity with realistic background traffic, larger topologies and inclusion of other critical services such as DNS.

Acknowledgments This study was conducted during the stay of Mr. Karaarslan as a TUBITAK grant holder at the Joint Research Centre and was supported by TUBITAK BIDEB 2219.

References

1. Augustin B, Krishnamurthy B, Willinger W (2009) Ixps: mapped? In: Proceedings of the ICM 2009, pp 336–349
2. Carl G, Kesidis G (2008) Large-scale testing of the Internet's border gateway protocol (BGP) via topological scale-down. ACM Trans Model Comput Simul 18(3):11:1–11:30. doi:10.1145/1371574.1371577
3. Eide E, Stoller L, Lepreau J (2007) An experimentation workbench for replayable networking research. In: Proceedings of the 4th USENIX symposium on networked systems design & implementation, pp 215–228
4. Gregori E, Improta A, Lenzini L, Orsini C (2011) The impact of IXPs on the AS-level topology structure of the Internet. Comp Commun 34(1):68–82. doi:10.1016/j.comcom.2010.09.002
5. Lad M, Oliveira R, Zhang B, Zhang L (2007) Understanding resiliency of Internet topology against prefix hijack attacks. In: Proceedings of the DSN 2007 conference, IEEE, pp 368–377. doi:10.1109/DSN.2007.95
6. Mirkovic J, Hussain A, Fahmy S, Reiher P, Thomas R (2009) Accurately measuring denial of service in simulation and testbed experiments. IEEE Trans Dependable Secure Comput 6(2):81–95. doi:10.1109/TDSC.2008.73
7. NCC R (2008) Ripe ncc, youtube hijacking: a ripe ncc ris case study.
8. Nicol D, Liljenstam M, Liu J (2005) Advanced concepts in large-scale network simulation. In: Proceedings of the winter simulation conference, 2005, IEEE, pp. 153–166. doi:10.1109/WSC.2005.1574248
9. Siaterlis C, Garcia AP, Genge B (2012) On the use of emulab testbeds for scientifically rigorous experiments. IEEE Commun Surv Tutorials (Accepted)
10. Winter R (2009) Modeling the Internet routing topology in less than 24h. In: Proceedings of the PADS 2009 conference, IEEE, pp 72–79. doi:10.1109/PADS.2009.17
11. Xu K, Duan Z, Zhang Z, Chandrashekar J (2004) On properties of internet exchange points and their impact on as topology and relationship. In: Proceedings of the NETWORKING 2004 conference. Springer, pp 284–295

Uneven Key Pre-Distribution Scheme for Multi-Phase Wireless Sensor Networks

Onur Catakoglu and Albert Levi

Abstract In multi-phase Wireless Sensor Networks (WSNs), sensor nodes are redeployed periodically to replace nodes whose batteries are depleted. In order to keep the network resilient against node capture attacks across different deployment epochs, called *generations*, it is necessary to refresh the key pools from which cryptographic keys are distributed. In this paper, we propose Uneven Key Pre-distribution (UKP) scheme that uses multiple different key pools at each generation. Our UKP scheme provides self healing that improves the resiliency of the network at a higher level as compared to an existing scheme in the literature. Moreover, our scheme provides perfect local and global connectivity. We conduct our simulations in mobile environment to see how our scheme performs under more realistic scenarios.

1 Introduction

Wireless Sensor Networks (WSNs) are used to carry wide range of data for various kinds of applications such as military, security, smart homes, telehealth, environmental observation and industry automation. Information that is transferred via those networks may contain not only temperature readings for habitat monitoring but also classified military data for battlefield surveillance which should not be seen by an unauthorized person. Therefore, security should be prioritized for these applications. WSNs have very limited resources in terms of memory and computational power. Hence, symmetric key cryptography is mostly used for existing key management schemes. However, pre-distribution of the symmetric keys effectively and efficiently in terms of resource usage are always been a challenge in WSNs.

O. Catakoglu · A. Levi (✉)
Sabanci University, Orhanli, 34956 Istanbul, Tuzla, Turkey
e-mail: catakoglu@sabanciuniv.edu

A. Levi
e-mail: levi@sabanciuniv.edu

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_35,

An attacker can learn key rings that are inside of any node by corrupting the node and use these keys to compromise links between other sensor nodes. In Random Key Pre-distribution (RKP) scheme by Eschenauer and Gligor [5], an adversary corrupts sensor nodes of the network persistently (i.e. constant attacker) will eventually learn the whole key pool of the corresponding sensor network.

Most of the recent studies do not consider mobile environment i.e. they assume that sensor nodes are static. However, it is not a realistic assumption to make, because there are many types of applications in commercial, environmental and military studies such as housekeeping robots, service industry, wildlife tracking, patient tracking, autonomous deployment, shooter detection [1] which require a new network topology that takes mobility of nodes into consideration.

In this paper, we propose Uneven Key Pre-distribution (UKP) scheme for multi-phase wireless sensor networks in mobile environment. The main idea of our method is the uneven pre-distribution of keys, which are taken from distinct key pools. At every deployment, newly deployed nodes will have their keys not only from the previous key pools, but also from a new distinct key pool. Therefore, keys in the network will be renewed at each redeployment phase and this will provide *self healing* to the network. Our results showed that we have better resiliency than RoK scheme without decreasing the local connectivity of network and without adding any additional memory overhead. Differently from the most of schemes, UKP uses multiple distinct key pools to refresh keys instead of using forward and backward hash operations for the sake of resiliency. In our scheme, hash operation is used only for creating a session key between two nodes from common keys.

The rest of the paper is organized as follows: Section 2 explains the related work on WSN security. Section 3 provides a comparative overview of our scheme and explains it in more detail. Section 4 presents the performance evaluations and Sect. 5 concludes the paper.

2 Related Work

Because public key cryptography is a very costly option for WSNs, most of the studies use symmetric cryptography. *RKP* is the most popular scheme that is proposed by Eschenauer and Gligor [5]. In this work, each wireless node picks keys from the same key pool before the deployment and if two nodes have at least one common key, they can establish a secure communication. However, a constant attacker eventually learns all the keys in the key pool and he can compromise the whole network [3].

Chan et al. [4] improved *RKP* scheme by using a threshold value, $q > 1$, for the number of common keys that are used for establishing a connection. Yet it requires more keys to be stored before the deployment which causes memory overhead, or it requires fewer keys in the key pool that leads to increase chance of same key being used more than once.

In RoK scheme [3], Castelluccia and Spognardi improved *RKP* scheme by using forward and backward hash chains to form a resilient network against node capture attacks. RoK scheme will be explained in detail at the third section.

There are some other works inspired by RoK that focus on multiphase networks. RPoK [6] is a polynomial-based *RKP* scheme proposed by Ito et al. for multiphase WSNs. Using private sub-key that is indirectly stored into every each node, they are able to establish a resilient network. Yi et al. [7] separates work time of the nodes into phases. They proposed a hash chain based scheme (HM scheme) for multiphase WSNs by using different key matrices for every phase.

3 Uneven Key Pre-Distribution Scheme

In this section, we firstly explain preliminaries and definitions that are used in explaining RoK [3] and our UKP schemes. Then, we overview the RoK [3] scheme and finally we explain the proposed UKP scheme in detail.

3.1 Preliminaries and Definitions

Notation used for RoK and UKP is explained in Table 1.

Because sensor nodes are battery operated systems, they have to be redeployed periodically for the sake of connectivity of the network. These new nodes are assumed

Table 1 Symbols used for RoK and UKP

Symbol	Explanation
A	Sensor A
n	Last generation of the network
FKP^j	Forward key pool at gen
BKP^j	Backward key pool at gen
P^j	Key pool of gen. j
m^j	Number of keys that are taken from key pool j
p_A^j	Number of keys that are taken from key pool j for node A
P	Key pool size
FKR_A^j	Forward key ring of A at gen
BKR_A^j	Backward key ring of A at gen
KR_A^j	Key ring of A that deployed at gen. j
G_w	Generation window
fk_t^j	t–th forward key at gen. j
bk_t^j	t–th backward key at gen. j
$k_{t_u}^j$	t_u–th key of P^j
k_{AB}	Common secret key between sensor A and B
$H(.)$	Secure hash function
m	Key ring size

to be deployed at regular epochs which are called *generations*. Also, lifetime of a node is assumed to have an upper bound and it is determined by *generation window*,G_w. A newly deployed sensor node's battery at generation j will deplete before generation $j + G_w$.

In RoK [3], keys are updated and refreshed at the end of each phase. Therefore, two nodes which are from different generations can establish a secure channel with this update mechanism. UKP follows a different mechanism for that purpose. It is based on average age of nodes in the network i.e. keys are pre-distributed to a node according to its life time. Every sensor node from generation j can communicate with another sensor node from different generation in the range of $[j - G_w, j + G_w]$ as in the RoK. However in UKP, instead of taking m number of keys from a key pool, a node takes its keys from G_w number of key pools. In other words, our scheme pre-distributes the keys not just from the key pool of the current generation, but also from key pools of future generations.

3.2 Overview of RoK Scheme

In the RoK scheme [3], key pools evolve for each new generation and sensors update their key rings by hashing their keys. In other words, keys have lifetimes and they are refreshed when a new generation is deployed. This mechanism is achieved by using forward and backward hash chains. Each sensor node takes its keys from both forward and backward key pools, *FKP* and *BKP*, that are associated to its generation. Each key pool has $P/2$ random keys.

Forward key pool at generation j defined as $FKP^j = \{fk_1^j, fk_2^j, \ldots fk_{P/2}^j\}$ where $fk_t^{j+1} = H\left(fk_t^j\right)$. Similiarly backward key pool at generation j will be $BKP^j = \{bk_1^j, bk_2^j, \ldots bk_{P/2}^j\}$ where $bk_t^j = H(bk_t^{j+1})$. While a node at generation j takes its forward keys from FKP^j, it takes its backward keys from BKP^{j+G_w-1}. Therefore, key rings of the node will be formally represented as:

$$FKR_A^j = \left\{fk_u^j | u = h(id_A \| i \| j)\,, \quad i = 1, 2, \ldots, m/2\right\} \text{ and}$$
$$BKR_A^j = \left\{bk_u^{j+G_w-1} | u = h(id_A \| i \| j)\,, \quad i = 1, 2, \ldots, m/2\right\}$$

for forward and backward key ring respectively.

A sensor B deployed at generation i in the range of $[j - G_w, j + G_w]$ communicates with sensor A while their common keys' indices are $t_1, t_2, \ldots, t_z$ respectively as follows.

while $i \leq j$,

$$k_{AB} = H\left(fk_{t_1}^j \| bk_{t_1}^{i+G_w-1} \| fk_{t_2}^j \| bk_{t_2}^{i+G_w-1} \| \ldots fk_{t_z}^j \| bk_{t_z}^{i+G_w-1}\right)$$

If two neighboring nodes have multiple shared common keys, all of them are used for the session key, k_{AB}. An adversary cannot compute keys from past generations by using forward keys, and cannot compute keys from future generations by using backward keys. Therefore, this mechanism provides forward and backward secrecy.

3.3 Proposed Uneven Key Pre-Distribution Scheme

In this paper we propose Uneven Key Pre-distribution (UKP) scheme for multiphase wireless sensors in mobile environment. The main idea of UKP is to distribute keys considering nodes' average life time statistic that is also represented in RoK [3] as shown in the Fig. 1. According to the statistic, when $G_w = 10$ and average life time of a node is $G_w/2 = 5$ with Gaussian distribution, most of the nodes in the network are newly deployed or young. After the age of four, the number of old nodes decreases dramatically.

Pools and Key Assignments. In UKP, there are n distinct pools that cannot be associated with each other. A sensor node takes its keys from G_w number of consecutive key pools in terms of generations. In order to decide how much to take from a key pool, we use the statistic shown in Fig. 1. In that case, a sensor will have the most keys from its descendant key pool and takes fewer and fewer keys from key pools that belong to further generations. For instance, a node at generation j takes its keys from $P^j, P^{j+1} \ldots P^{j+G_w-1}$. Relationship between key counts is $m^j > m^{j+1} > \ldots > m^{j+G_w-1}$. Based on the statistic on Fig. 1, while the difference between m^j, m^{j+1} and m^{j+2} is smaller, difference between m^{j+4} and m^{j+5} is much greater and it will continue to grow in further generations until the generation window is reached. Hence, keys that are captured are only valid between $[j - G_w]$th and $[j + G_w]$th generations and this provides forward and backward secrecy. The key ring of node A, which is denoted as KR_A^j is composed of all the key sets coming from different generations of key pools as stated above. Similarly, if node B is at generation $j+1$, key ring of node B which is KR_B^{j+1} will take its keys from key pools $P^{j+1}, P^{j+2} \ldots P^{j+G_w}$. Node B does not have any keys from P^j pool. In other words,

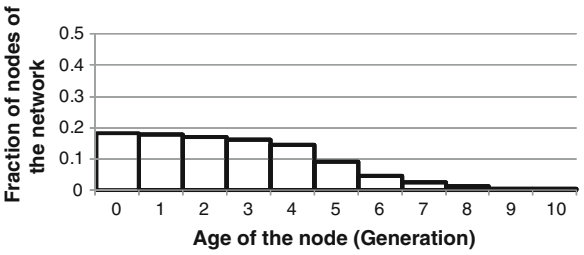

Fig. 1 Distribution of average age of the nodes

if there are G_w number of differences in terms of generations between two nodes, these two nodes will not share any common keys. In this way, we provide self healing because compromised keys become outdated in time. We can represent key ring of a node A at generation j as follows.

$$KR_A^j = \begin{cases} k_{t_u}^j | u = 1, 2, \dots \ m^j \\ k_{t_u}^{j+1} | u = 1, 2, \dots \ m^{j+1} \\ \dots \\ k_{t_u}^{j+G_w-1} | u = 1, 2, \dots \ m^{j+G_w-1} \end{cases}$$

where, $\mathrm{k}_{\mathrm{t_u}}^{\mathrm{i}}$, $i = j \dots j + G_w - 1$, are the keys selected from corresponding P^i using uniform random distribution with replacement.

The size of the key ring produced in this way, m, is calculated as follows.

$$m = m^j + m^{j+1} \dots + m^{j+G_w-1}$$

The purpose of having an uneven key distribution, i.e., using more keys from closer key pools in terms of generation is to achieve higher local connectivity in network. Moreover, this will strengthen the *self healing* property, since a compromised key has less chance to exist in further generations. In other words, most of the keys will be outdated sooner than the remaining ones and resiliency will be enhanced by the arrival of the new nodes with fresh keys.

Session Key Establishment. Any two nodes, say node $\boldsymbol{A}$ and node $\boldsymbol{B}$, can establish a session key only if they share at least one common key in their key rings. The session key is computed as the hash of all common keys that nodes $\boldsymbol{A}$ and $\boldsymbol{B}$ share. This key is denoted as $\boldsymbol{k_{AB}}$. The common keys used in session key establishment are chosen irrespective of the generations of keys. In other words, even if the two nodes come from different generations, they use all of the keys in their key rings to find common keys for session key establishment. Let us say that node $\boldsymbol{A}$ comes from generation $\boldsymbol{j}$, node $\boldsymbol{B}$ comes from generation $\boldsymbol{i}$ and the condition $\mathbf{i} \leq \mathbf{j}$ holds for node generations. Then, if the common keys $\boldsymbol{A}$ and $\boldsymbol{B}$ share are denoted as

$$CK_{AB}^x = \{\forall k_v^x | k_v^x \in KR_A^j, k_v^x \in KR_B^i \ and \ k_v^x \in P^x \text{for} \quad i \leq j \leq x\}$$

then, the session key is computed as follows.

$$\boldsymbol{k_{AB} = H\left\{CK_{AB}^x | x = j \dots i + G_w - 1\right\}}$$

Example As an example, if node A comes from generation j, node B comes from generation $j - 2$ as shown in Fig. 2, and the set of common keys they share are $k_{t_1}^j, k_{t_2}^j, k_{t_1}^{j+1}, k_{t_2}^{j+1}, k_{t_9}^{j+4}, k_{t_3}^{j+5}$ and $k_{t_{11}}^{j+8}$, the session key is computed as follows.

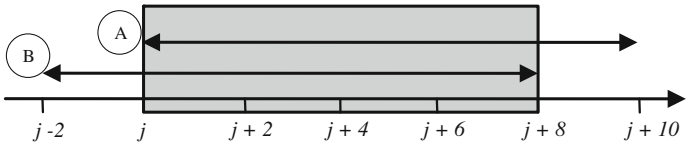

Fig. 2 Generation windows and overlapping generations of nodes *A* and *B*

$$k_{AB} = H\left(k_{t_1}^{j+2} \| k_{t_2}^{j} \| k_{t_1}^{j+1} \| k_{t_2}^{j+1} \| k_{t_9}^{j+4} \| k_{t_3}^{j+5} \| k_{t_{11}}^{j+8}\right)$$

Again, the common keys coming from different key pools (P^j, P^{j+1}, P^{j+4}, P^{j+5}, P^{j+8}) and consequently different generations are used together to form the session key. □

4 Performance Evaluation

We evaluated performance of our scheme with various simulations. In this section, we first explain performance metrics and then give simulation results together with the configuration and parameters.

4.1 Performance Metrics

Local connectivity is an important metric that shows the performance of the key distribution mechanism. It is defined as the probability of sharing a common key between two neighboring sensor nodes.

High local connectivity shows that a node can establish a secure communication with most of its neighbors. However, high local connectivity does not guarantee high global connectivity. Global connectivity is used to check if there are any nodes that are not reachable from the rest of the network. It is calculated as the ratio of the number of nodes in the largest isolated component to the number of nodes in the whole network.

In order to evaluate resiliency of the network we look at the ratio of additionally compromised links in the event of node capture. In other words, resiliency is computed as the number of indirectly corrupted channels divided by the number of all establishes links. We have better resiliency when the number is smaller. In our proposed scheme, we evaluated resiliency for active channels.

4.2 Evaluation by Simulation

For the sake of a fair comparison, we used similar setup as in the RoK [3] scheme for our simulation. The number of nodes in the network is taken as 1,000. We set the number of keys in each pool, P, as 10,000 and key ring size, m, as 500 for each node. Note that m value will be $m/2$ for forward and backward key rings of the RoK scheme. Generation window, G_w, is taken as 10 and we assume that a node's lifetime is determined according to a Gaussian distribution with mean $G_w/2$. and with standard deviation $G_w/6$. Deployment area is 500×500 m and sensor node's wireless communication range is 40 m. Nodes are distributed in that area with uniform random distribution. Speed of a node is decided randomly between 5–15 m per minute. Note that, we assumed network topology does not change over time for the sake of simplicity. In other words, nodes whose lifetimes expired will be replaced with new ones. We take average of 25 runs to get more realistic results.

We developed our simulation code in C# using MS Visual Studio 2010. Simulations are conducted on a computer with 64-bit Windows 7 running on Intel Core i7-2600 CPU, 8.00 GB RAM.

Instead of using a static environment, we run our simulation with mobile nodes to expand the usage of UKP. The mobility models that we use are explained next. After that we explain the attacker and the simulation results.

Mobility Models. In order to simulate node mobility, we used two models: (a) random walk mobility model, (b) reference point group mobility model. These models are summarized below and detailed information can be found in [2].

In *Random Walk Mobility Model*, a sensor node chooses a direction and speed randomly using uniform distribution. Then it moves in that direction for a fixed amount of time, which is taken as 1 minute in our simulations. When it finishes its movement, this process repeats itself with new direction and speed. Past location and speed information are not stored, so no memory usage is needed.

Reference Point Group Mobility Model covers both groups' random movement and random movement of individual nodes inside a group. Each group moves based on a node that is chosen as central node. Individual nodes pick a reference point randomly around the central node and move as in the Random Walk Mobility Model. Reference points are updated with the movement of central node.

Attacker Model. We assume that attacker can learn keys of a node by capturing it. As stated in RoK scheme, if a forward key captured in generation$j$, it is possible to compute key with same index for generations after j and it is also possible for backward keys for generations before $j + G_w$. Because key pools in the UKP scheme are distinct, there is no such association between keys of different generations. However, the attacker learns all the keys in a captured node including keys that belong to

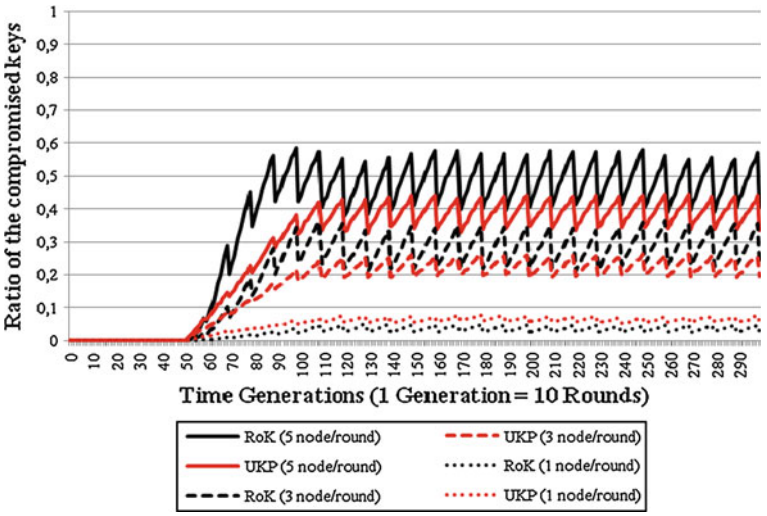

Fig. 3 Resiliency of RoK and UKP in case of an eager attacker with capture rates of 1, 3, and 5 nodes per round with Random Walk model

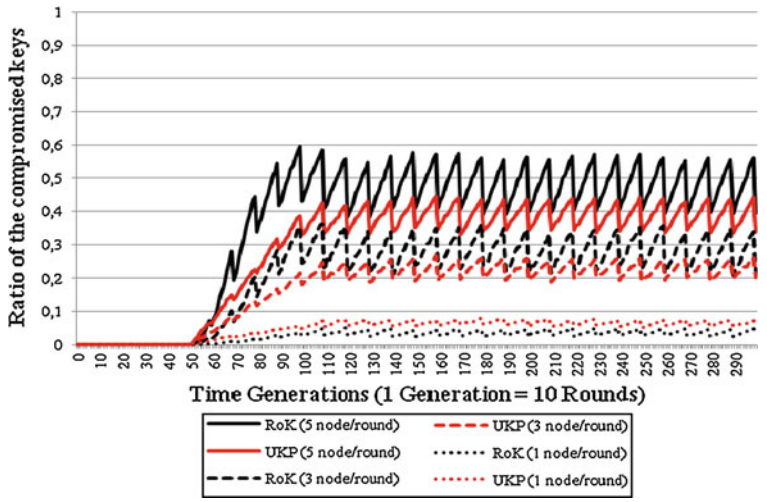

Fig. 4 Resiliency of RoK and UKP in case of an eager attacker with capture rates of 1, 3, and 5 nodes per round with Reference Point Group Mobility model

further generations. In our model, an *eager* attacker captures nodes at constant rate at each round and attack will not stop until the end of the simulation.

Simulation Results. We computed local connectivity, global connectivity and resiliency performance of our UKP scheme in comparison with RoK scheme [3]. Both RoK and our UKP schemes under both mobility models have perfect local and

global connectivity. In other words, every single node is reachable in the network and every node share at least one key with its neighbors.

For the evaluation of the network resiliency, we consider an attacker who captures sensor nodes with rates 1, 3 and 5 nodes per round ($1\,generation = 10\,rounds$). In our simulations, attacker starts compromising nodes at *generation* 5 in order to allow some time for network stabilization. Since keys in the network renewed by arrival of new nodes, rate of compromised keys sharply decreases at every new generation.

Figure 3 shows that in random walk model UKP scheme has almost same ratio of compromised keys under light attack (capture rate = 1). As attacker captures more nodes per round, our scheme outperforms RoK [3] model in terms of resiliency. Figure 4 also gives us similar results with reference point group mobility model. Compared to RoK, we have better results with capture rates 3 and 5.

5 Conclusions

In this paper, we proposed uneven key pre-distribution (UKP) scheme for multiphase wireless sensor networks in mobile environment. Our scheme is based on using different and distinct key pools at each generation. In this way, we improve the resiliency against heavy node capture attacks as compared to RoK scheme [3], while still maintaining perfect local and global connectivity.

Acknowledgments This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under grant 110E180.

References

1. Amundson I, Koutsoukos XD (2009) A survey on localization for mobile wireless sensor networks. In: Proceedings of the 2Nd international conference on mobile entity localization and tracking in GPS-less environments, Springer, Berlin, pp 235–254
2. Camp T, Boleng J, Davies V (2002) A survey of mobility models for ad hoc network research. Wireless Commun Mob Comput 2(5):483–502
3. Castelluccia C., Spognardi A (2007) RoK: a robust key pre-distribution protocol for multi-phase wireless sensor networks. In: SecureComm2007, Third International Conference on Security and Privacy in Communication Networks. Baltimore, MD, USA
4. Chan H, Perrig A, Song D (2003) Random key predistribution schemes for sensor networks. In: IEEE SP'03, pp 197–213
5. Eschenauer L, Gligor VD (2002) A key-management scheme for distributed sensor networks. In: Proceedings of the 9th ACM conference on computer and communications security. Washington, DC, USA, pp 41–47
6. Ito H, Miyaji A, Omote K (2010) RPoK: a strongly resilient polynomial-based random key predistribution scheme for multiphase wireless sensor networks. In: Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, pp 1–5
7. Yi S, Youngfeng C, Liangrui T (2012) A multi-phase key pre-distribution scheme based on hash chain. In: Fuzzy Systems and Knowledge, Discovery, pp 2061–2064

NEMESYS: Enhanced Network Security for Seamless Service Provisioning in the Smart Mobile Ecosystem

Erol Gelenbe, Gökçe Görbil, Dimitrios Tzovaras, Steffen Liebergeld, David Garcia, Madalina Baltatu and George Lyberopoulos

Abstract As a consequence of the growing popularity of smart mobile devices, mobile malware is clearly on the rise, with attackers targeting valuable user information and exploiting vulnerabilities of the mobile ecosystems. With the emergence of large-scale mobile botnets, smartphones can also be used to launch attacks on mobile networks. The NEMESYS project will develop novel security technologies for seamless service provisioning in the smart mobile ecosystem, and improve mobile network security through better understanding of the threat landscape. NEMESYS will gather and analyze information about the nature of cyber-attacks targeting mobile users and the mobile network so that appropriate counter-measures can be taken. We will develop a data collection infrastructure that incorporates virtualized mobile honeypots and a honeyclient, to gather, detect and provide early warning of mobile attacks and better understand the modus operandi of cyber-criminals that

E. Gelenbe (✉) · G. Görbil
Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ London , UK
e-mail: e.gelenbe@imperial.ac.uk

G. Görbil
e-mail: g.gorbil@imperial.ac.uk

D. Tzovaras
Centre for Research and Technology Hellas, Information Technologies Institute, 57001 Thessaloniki, Greece

S. Liebergeld
Technical University of Berlin, 10587 Berlin, Germany

D. Garcia
Hispasec Sistemas S.L, 29590 Campanillas, Malaga, Spain

M. Baltatu
Telecom Italia IT, 20123 Milan, Italy

G. Lyberopoulos
COSMOTE - Mobile Telecommunications S.A, 15124 Maroussi, Greece

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_36,

target mobile devices. By correlating the extracted information with the known patterns of attacks from wireline networks, we will reveal and identify trends in the way that cyber-criminals launch attacks against mobile devices.

1 Introduction

Smart devices have gained significant market share in the mobile handset and personal computer markets, and this trend is expected to continue in the near future. Smartphone shipments were 43.7 % of all handset shipments in 2012, and smartphones' share in the handset market is expected to grow to 65.1 % in 2016 [5]. Furthermore, while smartphones represented only 18 % of total global handsets in use in 2012, they were responsible for 92 % of total global handset traffic [6]. Therefore, smart devices evidently have a central role in the current and future mobile landscape. The growing popularity of smart mobile devices, Android and iOS devices in particular [22], has not gone unnoticed by cyber-criminals, who have started to address these smart mobile ecosystems in their activities. Smart mobiles are highly attractive targets since they combine personal data such as lists of contacts, social networks, and, increasingly, financial data and security credentials for online banking, mobile payments and enterprise intranet access, in a frequently used and always connected device. While most users are aware of security risks with using a traditional PC or laptop and therefore are more cautious, smart mobile devices have been found to provide a false sense of security [33], which exacerbates the mobile risk.

Smart devices also provide access to mobile networks for cyber-criminals and attackers to cross service and network boundaries by exploiting the vulnerabilities of the multiple communication technologies that smart devices have. Evolution of the mobile networks also introduces additional vulnerabilities, for example via the adoption of new radio access technologies such as femtocells [18]. Although the use of femtocells and other complementary access is recent and not yet widespread, their effect should not be underestimated. For example in 2012, 429 petabytes per month of global mobile data was offloaded onto the fixed network through Wi-Fi or femtocell radio access, which accounts for 33 % of total mobile traffic [6]. Mobile devices are also increasingly at the center of security systems for managing small or large emergencies in built environments, or during sports or entertainment events [14, 17], and they are used increasingly for online search of difficult-to-get sensitive information [2, 11]. Thus they will necessarily be targeted and breached in conjunction with other physical or cyber attacks, as a means of disrupting safety and confidentiality of individuals and emergency responders [19–21].

The ability of smart devices to install and run applications from official stores and third-party application markets has significantly increased the mobile malware threat [9, 36]. While the mobile malware threat is not new [7], it is decidedly evolving and growing as attackers experiment with new business models by targeting smart mobile users [28, 32]. For example, the number of detected malware was more

than 35, 000[1] in 2012, which reflects a six-fold increase from 2011 [32]. 2012 has also seen the emergence of the first mobile botnets [29]. A botnet is a collection of Internet-connected devices acting together to perform tasks, often under the control of a command and control server. Most malicious botnets are used to generate various forms of spam, phishing, and distributed denial-of-service (DDoS) attacks. In addition to giving cyber-criminals the advantages of control and adaptability, mobile botnets are also a significant threat to the mobile core network as they could be used to launch debilitating signaling-based DDoS attacks [23, 34]. In order to address the growing mobile threat, there is an urgent need to detect, analyze and understand the new vulnerabilities and threats in the smart mobile ecosystem. These new vulnerabilities and threats are a result of the evolution of mobile networks and smart devices, the changing way users interact with technology, the popularity of smart devices, and the heterogeneity of the wireless interfaces, supported platforms and offered services. We need to be proactive and work on predicting threats and vulnerabilities to build our defenses before threats materialize in order to advance in the fast moving field of cyber-security and to counter existing and potential mobile threats. In the next section, the approach adopted in the NEMESYS project for this purpose is described.

Thus the EU FP7 research project NEMESYS[2] will develop a novel security framework for gathering and analyzing information about the nature of cyber-attacks targeting mobile devices and the mobile core network, as well as the identification and prediction of abnormal behaviours observed on smart mobile devices so that appropriate countermeasures can be taken to prevent them. We aim to understand the modus operandi of cyber-criminals, and to identify and reveal the possible shift in the way they launch attacks against mobile devices through root cause analysis and correlation of new findings with known patterns of attacks on wireline networks.

2 The Data Collection Infrastructure

Figure 1 shows the system architecture that will be developed within the NEMESYS project. The core of the NEMESYS architecture consists of a data collection infrastructure (DCI) that incorporates a high-interaction honeyclient and interfaces with virtualized mobile honeypots (VMHs) in order to gather data regarding mobile attacks. The honeyclient and VMHs collect mobile attack traces and provide these to the DCI, which are then enriched by analysis of the data and by accessing related data from the mobile core network and external sources. For example, TCP/IP stack fingerprinting in order to identify the remote machine's operating system, and clustering of the traces are passive methods of data enrichment. DNS reverse name lookup, route tracing, autonomous system identification, and geo-localization are

[1] This number is for Android alone, which accounts for 99 % of all encountered malware in 2012 [29].

[2] http://www.nemesys-project.eu/nemesys/index.html

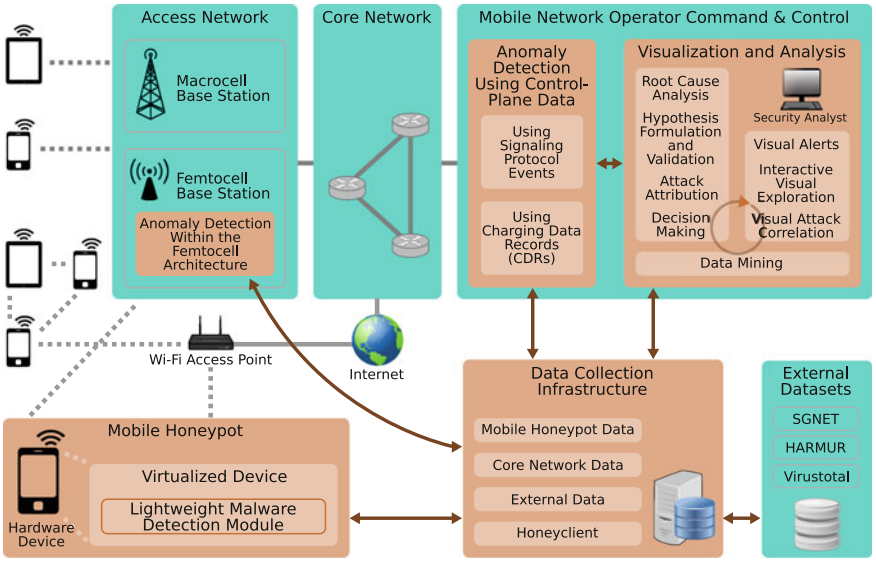

Fig. 1 The NEMESYS architecture

methods to improve characterization of remote servers which may require access to external sources, possibly in real time. The enriched mobile attack data is made available to anomaly detection module, and the visualization and analysis module (VAM) running on the mobile network operator site.

As an initial step in the design of the DCI, we are identifying available external data sources relating to wireline network attacks which will enable correlation of attack-related data coming from multiple heterogeneous sources. Different sources of information that NEMESYS partners maintain and have access to will be used for this purpose, for example SGNET [25], HARMUR [24], and VirusTotal. A source aggregator will be designed and developed to harvest and consolidate data from these sources, and the honeyclient and VMHs, in a scalable database. Scalable design of the database is important in order to be able to efficiently store and handle large heterogeneous data sets. As a final step, the DCI will help in the definition of the appropriate inputs representing normal and malicious network activity, which will then be used as the fundamental representation of information in the VAM.

The *honeyclient* being developed as part of the DCI is responsible for interacting with web servers to find websites with malicious content targeting mobile users, and for collecting related mobile threat data. The honeyclient consists of the crawler, client, and detector components. The crawler will generate a list of websites of interest for the client to visit. The client component will run an Android emulator which will generate, queue, and execute the requests corresponding to each discovered website, and record traces of changes in the system state that occur as a result.The

malware detector component will be used to detect malicious content. Data relating to identified malicious websites will be provided to the DCI by the honeyclient, which is described in more detail in [8].

3 Virtualized Mobile Honeypots

We adopt the high-interaction virtualized mobile client honeypot scheme in order to attract and collect mobile attack traces. Honeypots are networked computer system elements that are designed to be attacked and compromised so we can learn about the methods employed by the attackers [31]. Traditional honeypots are servers that passively wait to be attacked, whereas client honeypots are security devices that actively search for malware, compromised websites and other forms of attacks. High-interaction client honeypots are fully functional, realistic client systems which do not impose any limitations on the attacker other than those required for containing the attack within the compromised system. Despite their complexity and difficulty of maintenance, high-interaction client honeypots are effective at detecting unknown attacks, and are harder to detect by the attacker [31]. They also enable in-depth analysis of the attacks during and after the attack has taken place.

In NEMESYS, we are developing a high-interaction virtualized client honeypot for the Android mobile platform. We have chosen Android considering its popularity among mobile users and the extremely high ratio of malware targeting Android [4, 29]. We are developing a virtualization technology that addresses the problems we have identified in the system- and application-level security mechanisms of Android and enables secure support for new schemes of smart device use such as "bring your own device" [26]. Our virtualization technology logically partitions the physical device into two virtual machines (VMs): the *honeypot VM* and the *infrastructure VM*. The honeypot VM will host the largely unmodified mobile device operating system, and it will not have direct access to the device's communication hardware. The infrastructure VM will mediate all access to the communication hardware, and employ sensors to wiretap any communication and detect suspicious behaviour. It will also provide the event monitoring, logging and filesystem snapshot facilities, as well as transmit threat information to the DCI. It will host a *lightweight malware detection module* in order to identify malicious applications running on the honeypot VM. For this purpose, both signature-based and behaviour-based approaches will be considered. In order to improve the efficiency of malware detection, we will identify and prioritize the most important attributes in the system state space to monitor.

Our virtualization technology will ensure that an attack is confined within the compromised device so that it will not put other devices in the network at risk. It will also stop malware from using premium rate services and from subscribing the user to services without her knowledge. Thus, the user will be spared from any financial distress that may arise as a result of using the mobile honeypot. The virtualization solution also enables taking full snapshots of the honeypot VM filesystem for further forensic analysis of an attack, as well as improving honeypot maintenance since a compromised honeypot could be restored more quickly.

Current IP-based attacks encountered on mobile devices [35] have been found to be largely similar to non-mobile devices [10, 15], but we are more interested in the traits of attacks that are tailored specifically for mobile devices. Our initial research has shown that the infection vector of most mobile malware is social engineering, where users are "tricked" into installing the malware themselves. Upcoming malware will also employ attack vectors that require interaction with the user; for example, we have already witnessed the first malicious QR codes, which need to be scanned by the user for their activation. These observations have led us to the conclusion that the user should not be ignored in the construction of an effective mobile honeypot. To this end, we introduce the *nomadic honeypot* concept [27], which utilizes real smartphone hardware running the virtualization solution being developed by NEMESYS. We plan to deploy nomadic honeypots by handing them out to a chosen group of volunteers, who will use the honeypot as their primary mobile device. It will be up to these human users to get the honeypot infected by visiting malicious sites, installing dubious applications, etc. Traces from malware and other types of mobile attacks collected by the honeypots will be provided to the DCI.

4 Anomaly Detection Using Control Plane and Billing Data

The purpose of the anomaly detection mechanisms is to identify and predict deviations from normal behaviour of mobile users and the core network. These mechanisms will utilize Charging Data Records (CDR) of the users and control-plane protocol data, together with enriched mobile attack traces provided by the DCI. In addition to attacks targeting mobile users, mobile networks are vulnerable to a novel DoS attack called the signaling attack [23], which seeks to overload the control plane of the mobile network using low-rate, low-volume attack traffic by exploiting the structure and characteristics of mobile networks, for example by repeatedly triggering radio channel allocations and revocations. We will use control-plane protocol data such as traces of signaling events in order to identify such DoS attacks against the mobile network. Sanitized (anonymized) billing data will mostly be used to identify attacks targeting mobile users. For these purposes, we will use normal user behaviour statistics, as well as synthetic "typical" user playbacks, to create traces of signaling events and CDRs so as to characterize and extract their principal statistics such as frequencies, correlations, times between events, and possible temporal tendencies over short (milliseconds to seconds) and long (hours to days) intervals. We will employ Bayesian techniques such as maximum likelihood detection, neuronal techniques based on learning, and a combination of these two in order to design and develop robust and accurate change detection algorithms to detect the presence of an attack, and classification algorithms to identify with high confidence the type of attack when it is detected. Novel femtocell architectures provide a specific opportunity for user-end observation of network usage, while they also have specifics for attacks within the femtocells. To address femtocell-specific attacks, we will conduct a survey and evaluation of how users may be monitored and attacks detected within a femtocell, and how these are linked to overall mobile network events.

In these environments a number of novel ideas are being exploited. The structure of the signaling and billing network is being modeled as a queueing network [16] to capture the main events that involve hundreds of thousands of mobile calls and interactions among which only a few may be subject to an intrusion or attack at any given time. Detection of abnormalities is studied using learning techniques based on neural network models [12, 13] that can provide the fast low-order polynomial or linear detection complexity required from the massive amount of real-time data, and the need to detect and respond to threats in real-time. Such techniques can also benefit from distributed task decomposition and distributed execution for greater efficiency [3]. Our approach to anomaly detection is discussed in more detail in [1].

5 Root Cause Analysis, Correlation and Visualization

Enriched attack traces and mobile network data collected by the DCI, and the output of the anomaly detection modules are fed into the visualization and analysis module (VAM). The VAM's purpose is to aid the detection of existing and emerging threats in the mobile ecosystem through attack attribution, root cause identification, and correlation of observed mobile attacks with known attack patterns. The data provided to the VAM represents a large and heterogeneous data set that needs to be presented in a meaningful way to the security analyst without overwhelming her or restricting available views and actions on the data. In addition to mere representation of data, the VAM aims to provide visual analytics tools to the analyst. This task is compounded by different uses of visualization: (1) real-time monitoring of the status of mobile users and the mobile network, and (2) exploratory data analysis. For real-time monitoring, the security status of a large set of mobile users, and more importantly the mobile network, need to be presented. This includes providing early alerts for abnormal behaviour, DoS attacks, malware spreading among the users of the mobile network, etc. The VAM must also provide visual analytics tools so the analyst can perform hypothesis formulation and testing, attack attribution, and correlation analysis, with the help of algorithms running in the background.

In order to effectively visualize and explore large sets of heterogeneous, dynamic, complex data, it is necessary to create multiple coordinated views of the data that allow a multi-faceted perception and the discovery of any hidden attributes. The analysis methods also need to be scalable for early network alerts and fast access to the underlying data. We will therefore focus on enabling a real-time analysis framework by means of incremental analysis and visualization methods, such as multi-level hierarchical screen visualizations that update smoothly rather than showing abrupt changes. Visualization of mobile network data is discussed in more detail in [30].

6 Conclusions

In the NEMESYS Project, we will address and understand the new and potential vulnerabilities, threats, and operating methods of cyber-criminals, and provide new insight into next generation network security for the smart mobile ecosystem. We will contribute to the research novel security technologies for the identification and prediction of abnormal behavior observed on smart mobile devices, and to the gathering and analyzing of information about cyber-attacks that target mobile devices, so that countermeasures can be taken. We will develop virtualized honeypots for mobile devices, a data collection infrastructure, and novel attack attribution and visual analytics technologies for mining, presentation of large amounts of heterogeneous data regarding the smart mobile ecosystem.

Acknowledgments The work presented in this paper was supported by the EU FP7 collaborative research project NEMESYS (Enhanced Network Security for Seamless Service Provisioning in the Smart Mobile Ecosystem), under grant agreement no. 317888 within the FP7-ICT-2011.1.4 Trustworthy ICT domain.

References

1. Abdelrahman O, Gelenbe E, Gorbil G, Oklander B (2013) Mobile network anomaly detection and mitigation: the NEMESYS approach. In: Proceedings of 28th international symposium on computer and information sciences (ISCIS'13) accepted for publication
2. Abdelrahman OH, Gelenbe E (2013) Time and energy in team-based search. Phys Rev E 87(3):032125
3. Aguilar J, Gelenbe E (1997) Task assignment and transaction clustering heuristics for distributed systems. Inf Sci 97(1–2):199–219
4. Baltatu M, D'Alessandro R, D'Amico R (2013) NEMESYS: first year project experience in telecom Italia information technology. In: Proceedings of 28th international symposium on computer and information sciences (ISCIS'13) accepted for publication
5. (2013) Mobile device market to reach 2.6 billion units by 2016. Canalys. [Online]. Available: http://www.canalys.com/newsroom/mobile-device-market-reach-26-billion-units-2016
6. (2013) Cisco visual networking index: global mobile data traffic forecast update, 2012–2017. White Paper. Cisco. [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf
7. Dagon D, Martin T, Starner T (2004) Mobile phones as computing devices: the viruses are coming! IEEE Pervasive Comput 3(4):11–15
8. Delosieres L, Garcia D (2013) Infrastructure for detecting Android malware. In: Proceedings of 28th international symposium on computer and information sciences (ISCIS'13) accepted for publication
9. Felt AP, Finifter M, Chin E, Hanna S, Wagner D (2011) A survey of mobile malware in the wild. In: Proceedings of 1st ACM workshop on security and privacy in smartphones and mobile devices (SPSM'11), pp 3–14
10. Gelenbe E (2009) Steps toward self-aware networks. Commun ACM 52(7):66–75
11. Gelenbe E (2010) Search in unknown random environments. Phys Rev E 82(6):061112
12. Gelenbe E (2012) Natural computation. Comput J 55(7):848–851
13. Gelenbe E, Fourneau J-M (1999) Random neural networks with multiple classes of signals. Neural Comput 11(4):953–963

14. Gelenbe E, Gorbil G, Wu J-F (2012) Emergency cyber-physical-human systems. In: Proceedings of 21st international conference on computer communications and networks (ICCCN), pp 1–7
15. Gelenbe E, Loukas G (2007) A self-aware approach to denial of service defence. Comput Netw 51(5):1299–1314
16. Gelenbe E, Muntz RR (1976) Probabilistic models of computer systems: part I (exact results). Acta Informatica 7(1):35–60
17. Gelenbe E, Wu F-J (2012) Large scale simulation for human evacuation and rescue. Comput Math Appl 64(2):3869–3880
18. Golde N, Redon K, Borgaonkar R (2012) Weaponizing femtocells: the effect of rogue devices on mobile telecommunication. In Proceedings 19th annual network and distributed system security, symposium (NDSS'12), pp 1–16
19. Gorbil G, Filippoupolitis A, Gelenbe E (2012) Intelligent navigation systems for building evacuation. In: computer and information sciences II. Springer, pp 339–345
20. Gorbil G, Gelenbe E (2011) Opportunistic communications for emergency support systems. Procedia Comput Sci 5:39–47
21. Gorbil G, Gelenbe E (2013) Disruption tolerant communications for large scale emergency evacuation. In: Proceedingds 11th IEEE international conference on pervasive computing and communications workshops
22. (2013) Android and iOS combine for 91.1 % of the worldwide smartphone OS market in 4Q12 and 87.6 % for the year, according to IDC. IDC. [Online]. Available: http://www.idc.com/getdoc.jsp?containerId=prUS23946013#.UTCOPjd4DlY
23. Lee PP, Bu T, Woo T (2009) On the detection of signaling DoS attacks on 3G/WiMax wireless networks. Comput Netw 53(15):2601–2616
24. Leita C, Cova M (2011) HARMUR: storing and analyzing historic data on malicious domains. In: Proceedings of 1st workshop on building analysis datasets and gathering experience returns for, security (BADGERS'11), pp 46–53
25. Leita C, Dacier M (2008) SGNET: a worldwide deployable framework to support the analysis of malware threat models. In: Proceedings 7th European dependable computing conference (EDCC'08), pp 99–109
26. Liebergeld S, Lange M (2013) Android security, pitfalls, lessons learned and BYOD. In: Proceedings of 28th international symposium on computer and information sciences (ISCIS'13) accepted for publication
27. Liebergeld S, Lange M, Mulliner C (2013) Nomadic honeypots: a novel concept for smartphone honeypots. In: Proceedings of W'shop on mobile security technologies (MoST'13), in conjunction with the 34th IEEE symposium on security and privacy, accepted for publication
28. (2008) State of mobile security (2012) Lookout mobile security. [Online]. Available: https://www.lookout.com/_downloads/lookout-state-of-mobile-security-2012.pdf
29. Maslennikov D, Namestnikov Y (2012) Kaspersky security bulletin 2012: the overall statistics for 2012. Kaspersky lab. [Online]. Available: http://www.securelist.com/en/analysis/204792255/Kaspersky_Security_Bulletin_2012_The_overall_statistics_for_2012
30. Papadopoulos S, Tzovaras D (2013) Towards visualizing mobile network data. In: Proceedings of 28th international symposium on computer and information sciences (ISCIS'13) accepted for publication
31. Provos N, Holz T (2007) Virtual Honeypots: from Botnet tracking to intrusion detection. Addison Wesley, Jul
32. Raiu C, Emm D (2012) Kaspersky security bulletin 2012: Malware evolution. Kaspersky lab. [Online]. Available: http://www.securelist.com/en/analysis/204792254/Kaspersky_Security_Bulletin_2012_Malware_Evolution
33. (2012) National cyber security alliance and McAfee release new cybercrime data for national cyber security awareness month. StaySafeOnline.org. [Online]. Available: http://www.staysafeonline.org/about-us/news/national-cyber-security-alliance-and-mcafee-release-new-cybercrime-data

34. Traynor P, Lin M, Ongtang M, Rao V, Jaeger T, McDaniel P, Porta TL (2009) On cellular botnets: measuring the impact of malicious devices on a cellular network core. In: Proceedings of 16th ACM conference on computer and communications, security (CCS'09), pp 223–234
35. Wahlisch M, Vorbach A, Keil C, Schonfelder J, Schmidt TC, Schiller JH (2013) Design, implementation, and operation of a mobile honeypot, arXiv computing research repository, vol abs/1301.7257
36. Zhou Y, Jiang X (2012) Dissecting Android malware: characterization and evolution. In: Proceedings of 2012 IEEE symposium on security and privacy, pp 95–109

Towards Visualizing Mobile Network Data

Stavros Papadopoulos and Dimitrios Tzovaras

Abstract This paper presents the research directions that the visualization in the NEMESYS project will follow, so as to visualize mobile network data and detect possible anomalies. These directions are based on the previous approaches on network security visualization and attack attribution techniques, while possible extensions are also discussed based on the presented approaches.

1 Introduction

Mobile malware can be used for many purposes, such as attack the mobile network and cause Denial of Service, monitor network traffic, or steal the users' personal information. Thus, it is very important to develop techniques that could detect this type of behavior so as to counter attack it efficiently. To this end, there are mainly two groups of techniques that can be used for attack detection, the use of algorithmic methods, and the use of visualization methods. The NEMESYS project will utilize both methods and combine them so as to detect and provide early warning of attacks on mobile devices and, eventually, to understand the modus operandi of cyber-criminals that target mobile devices.

This work has been partially supported by the European Commission through project FP7-ICT-317888-NEMESYS funded by the 7th framework program. The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the European Commission..

S. Papadopoulos (✉)
Department of Electrical and Electronic Engineering, Imperial College London, London, UK
e-mail: s.papadopoulos11@imperial.ac.uk

D. Tzovaras
Information Technologies Institute, Centre for Research and Technology Hellas, Thermi, Greece
e-mail: tzovaras@iti.gr

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_37,

This paper focuses on the research directions that will be followed, so as to provide efficient visual representations of the input information for the purpose of attack detection and attribution. To a certain extent, the visualization methods presented here are combined with algorithmic methods (as for example clustering or filtering) so as to take advantage of both approaches and enhance the analytical potential of the proposed techniques. Other contributions made in the context of the NEMESYS project are detailed in [1–5].

The rest of the paper is organized as follows. Section 2 presents the related work, while Sect. 2.1 presents the related research that has been carried out previously. The novel research directions that will be investigated are presented in Sect. 3. Finally, the paper concludes in Sect. 4.

2 Related Work

There are many visualization systems that deal with network security in general [6]. But there are very few approaches that specifically target mobile network security.

The most interesting approach is MobiVis [7]. MobiVis uses ontology graphs in which each vertex represents a specific ontology or object (as for example places or persons), while edges illustrate connectivity between the different ontologies. Furthermore, MobiVis uses radially lay-out, pie-shaped wedges around the vertices of the graph, so as to represent time-varying information, such as some particular activity including phone calls, or proximity relations. In addition, 2-dimensional plots are also utilized, so as to highlight periodic patterns and allow the analyst to perform temporal filtering.

The authors of [8] also utilized 2-dimensional plots and coloring, so as to represent the relationship between three features. As a result they were able to detect time varying patterns and behaviors. Furthermore, in [8] the use of entropy is suggested as a metric that quantifies the degree in which a specific person has a predictable behavior.

In [9] the use of probability spans is suggested for the purpose of visualizing and exploring mobile data. For each feature, the probability density function is calculated over time, and afterwards, using 1-dimensional plots this function is visualized onto the screen. Moreover, using multiple connected views of probability spans the analyst is able to detect patterns and extract meaningful information.

2.1 Attack Attribution

This section presents the research that has been carried out in VIS-SENSE regarding novel information visualization and Attack Attribution techniques. The Attack Attribution Graphs [10, 11] visualization method uses multiple views of linked graph views for the purpose of anomaly detection and attack attribution i.e., how to attribute

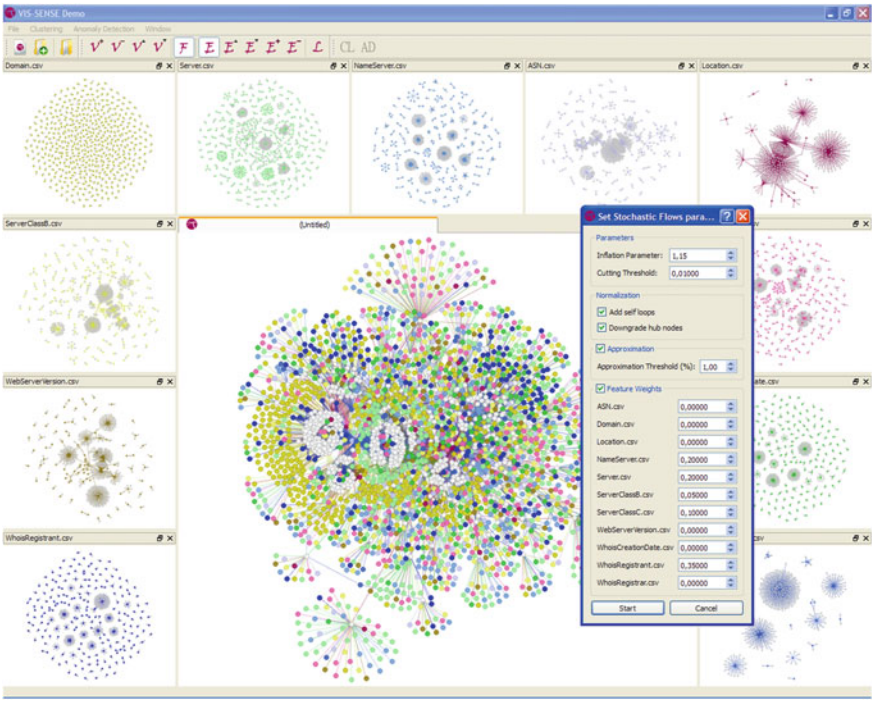

Fig. 1 The attack attribution graphs visual analytics framework for graph visualization. Eleven different features are visualized in the dock windows surrounding the main window. The graph corresponding to the interconnections of security events to the 11 features is visualized in the middle. The dialog window allows the security analyst to set the parameters of the stochastic flows clustering algorithm

(apparently) different attacks to a common root cause, based on the combination of all available evidence. The Attack Attribution Graphs visualizes network traffic and attack related data.

The first step towards achieving Attack Attribution, is the definition of the k-partite graph. This graph consist of k types of nodes, $k-1$ types of feature nodes and 1 type of event nodes, and is used to visualize the structure of the whole dataset (Fig. 1). A force directed algorithm is utilized to position the nodes of the graph onto the visual display, while the same algorithm is also utilized to position the nodes of all the individual bipartite feature graphs shown in the dock windows of Fig. 1.

On important aspect of the force directed algorithm, is that nodes that are positioned in close geometric proximity, have many similarities. In other words, two nearby vertices have highly similar feature vectors, whereas two distant points have nothing in common. Thus, by utilizing the force-based algorithm, the final layout groups together nodes of high similarity, and thus provides an intuitive way to detect clusters of attack events that are ascribed to the same root cause.

In addition, the Attack Attribution Graphs also utilizes an analytical clustering method based on stochastic flows, which helps combine multiple attack features in a more descriptive, intuitive and meaningful way. The clustering results are mapped onto the graphs by the use of coloring, thus leading to attack attribution and root cause identification. The knowledge that the analyst gains by this procedure, can be used so as to redefine the parameters of the clustering algorithm and as a result change the analytical procedure, so as to find any hidden relationships or phenomena.

Finally, an abstract graph visualization method is provided by the Attack Attribution Graphs so as to deal with the vast number of data dimensions, as well as the vast number of events. In this abstract graph view, the event nodes are omitted and the visualization depicts only the feature nodes. The size of the depicted node represents the degree of the corresponding feature node in the original k-partite graph. Thus the larger a node, the most events it is involved in. Furthermore, an edge is added between to feature nodes, if and only if an event occurred which is related to these features, while, the width of the edge is proportional to the number of such events.

2.2 BGP

An other visualization method developed in the context of the VIS-SENSE project is the BGP Routing Changes Graph [12]. BGP Routing Changes Graph is utilized for the purpose of visualizing the traffic changes caused by the Border Gateway Protocol (BGP) announcement messages. BGP is the de facto protocol used today in the internet for the exchange of reachability and routing information between the Autonomous Systems (AS).

As the name suggests, BGP Routing Changes Graph uses a graph metaphor to represent the topology of the ASes over the internet. The nodes of the graph represent ASes, while edges illustrate connectivity between ASes. The size of the nodes represents the amount of IP ownership change calculated within a time window, while the width of the edges represents the magnitude of traffic change that the corresponding edge serves. Furthermore, red color illustrates negative traffic change, while green positive traffic change.

The described graph approach suffers from scalability issues, as the large amount input data makes it very difficult to be visualized in the relatively small display screen. In the year 2012 the internet consisted of over 40,000 ASes and over 53,000 physical links between them. The large size of this graph renders it very difficult to visualize it entirely due to the cluttering that is introduced. As a result, in order to overcame this scalability issue, BGP Routing Changes Graph uses a hierarchical clustering approach that produces a hierarchy of coarse to fine graphs. As a consequence, the analyst is able to use of high level visualizations in the first steps of the analysis, and thus eliminate visual clutter. Moreover, the analyst is capable of seeing in more detail specific parts of the graph that are of particular interest.

One important advantage of the BGP Routing Changes Graph, is the introduction of a novel metric that quantifies the quality of the visualization. Using the proposed

quantification approach, BGP Routing Changes Graph is able to produce optimal visualization results. To achieve this, the proposed metric uses Information Theory to quantify the amount of information that is visualized using the graph metaphor, for the purpose of enabling comparisons and optimizations.

3 Research directions in NEMESYS

The Sections that follow provide an overview of the research directions that the visualization in NEMESYS will follow.

3.1 Visualization

In Sect. 2.1, two visualization methods were described, the Attack Attribution Graphs and the BGP Routing Changes Graph. These two approaches use complectly different metaphors, despite the fact that they are both graph based. In the Attack Attribution Graphs approach the nodes represent either events or features. The connection between the events and the feature values that they have, is illustrated using edge connectivity. On the other hand, the BGP Routing Changes Graph maps the feature value of each node or edge onto the corresponding radii or width.

As novel extension of both approaches, the user will be provided with the choice to select which features will be represented using edge connectivity and which features will be mapped onto the visual features of the graph nodes. The visual features of the nodes are the size, color, and opacity, while the possibility to use glyphs instead of nodes will also be investigated, so as map a vast amount of features onto it. As a result of this procedure, visual cluttering is decreased, while the visualization capacity is increased, so as to visualize a large amount of features.

The use of aggregation and abstraction will also be investigated within this task. As far as the aggregation is concerned, the user will be provided with the choice to aggregate nodes of the graph based on certain criteria. A possible example is to aggregate all the mobile user nodes on per mobile Cell basis, so as to reduce the size of the graph and detect patterns on per mobile Cell basis. On the other hand, as far as the abstraction is concerned, the Attack attribution graphs approach has introduced an abstract graph view, which will be also utilized here. As a further step, filtering will be applied onto the abstract graph view based on multiple criteria, so as to visualize only the most important nodes.

Entropy will be used for the purpose of increasing the information content of the generated visualizations and as a result present more information to the analyst. This enables the analyst to take more informed decisions about the data and reduce false positives that could be caused by high information loss. Furthermore, the use of entropy measures is useful in the selection of appropriate aggregation and abstraction criteria, so as to show the most interesting parts of the data, while omitting irrelevant parts.

3.2 Clustering

Clustering is the task of grouping a set of objects in such a way that objects in the same group (clusters) are more similar to each other than to those in other groups (clusters). Due to the properties of the clustering procedure, it can be used to address the so called attack attribution problem in network security; that is, to assist the security analyst in exploring the root causes of attack phenomena and revealing the modus operandi of cyber-criminals.

There are generally many clustering algorithms that use different approaches, as for example the k-means, density based [14], distance based [15], or entropy based [16]. All these algorithms operate on the feature N dimensional space. In addition to these algorithms, there is a possibility to use the k-partite graph that is generated from the features as a way to find clusters using graph based clustering algorithms [17]. In the Attack Attribution Graphs visualization approach [10], a graph based clustering algorithm was developed that utilized stochastic flows. As a further research, alternative clustering algorithms will be investigated so as find clusters based on the feature space or the graph structure.

After the construction of clusters, the appropriate visualization technique to visualize the clusters will be selected. In this case there are two choices, graph or glyph visualization, as illustrated in Fig. 2. Using the graph representation of clusters, along with the force directed layout (Fig. 2a), the analyst is able to see the actual distances between the different nodes and clusters, so as to verify the clustering results, or even find alternative clusters on his/her own. The disadvantage of this approach is that the use of coloring to discriminate between the different clusters makes it difficult to visualize a large number of clusters. On the other hand, the glyph representation of clusters (Fig. 2b) is a much less space greedy approach, and it is possible to visualize a larger number of clusters. Furthermore, specific properties of each cluster (such as size or density) can be mapped onto the corresponding glyph, in order to provide additional information.

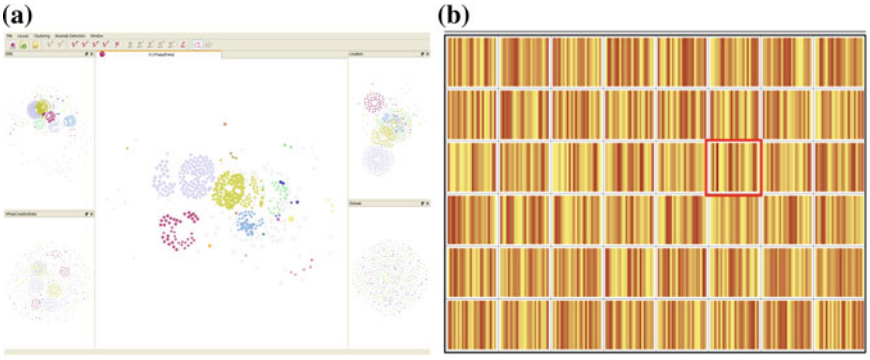

Fig. 2 Alternative visual representations of the cluster results. **a** Graph based representation of clusters. **b** Glyph based representation of clusters [13]

In future visualization approaches, both graph and glyph representations of clusters will be taken into account, so as to select the best one or even combine them using linked views and take advantage of both.

3.3 Filtering

The mobile data can be very large in scale. Thus, filtering techniques are necessary to support exploring such a large data set, and allow the analyst to select a subset of the data to focus on. Filtering can be performed in multiple ways. The most common methods are: (1) the threshold filtering; that is omit every feature value below a specific predefined threshold, (2) allow the analyst to select a subset of interesting data to focus on, based on multiple criteria, as for example specific nodes of the graph, or clusters, (3) select a subset of all the available features for visualization (feature selection), and finally (4) select all the data that occurred over a specific time window. This Section focuses only on cases 3 and 4.

The purpose of feature selection is to select a subset of the most representative features to perform further analysis. To achieve this goal the correlations between the different features must be found, so as to omit the most irrelevant or redundant ones. There are two categories of feature correlation-selection methods, algorithmic and visual. Well known algorithmic methods are the Mutual information and the Pearson coefficient metrics. On the other hand, visual feature selection methods are still a research area. Although there are some well know visual methods, such as Scatterplots and Parallel Coordinates, no new methods were presented recently. The goal of this task is to find new visual feature selection methods to assist the user in the analytical procedure.

One example of visual feature selection is illustrated in Fig. 3, which depicts three bipartite feature graphs. The utilization of the force directed layout has as a result the

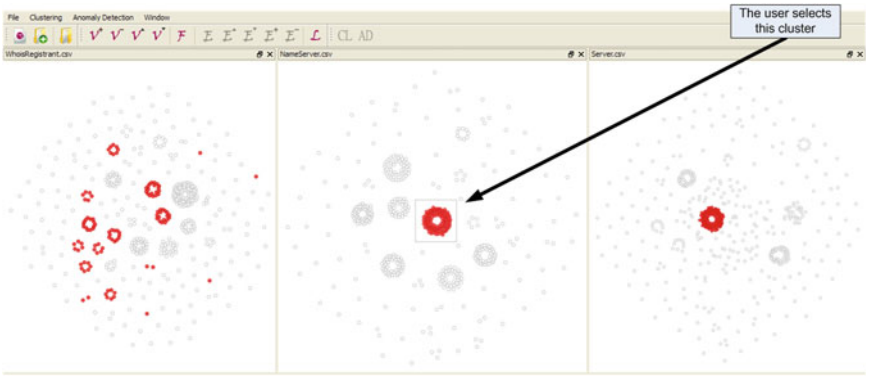

Fig. 3 Visual correlation based on linking and brushing

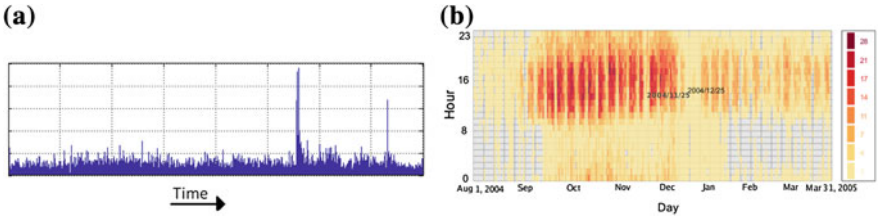

Fig. 4 1D and 2D plots used to provide an overview of the available data over a specific period of time. By utilizing these plots the user is able to filter the data and focus on the most interesting time periods. **a** 1D plot. **b** 2D plot taken from [7]

formation of visual clusters, by positioning closely related nodes in close proximity. Afterwards, linking and brushing is utilized so as to allow the user to select a cluster of nodes and highlight the corresponding nodes in all the feature graphs. Thus, by observing the clustering formation and the number of clusters it is possible to detect correlated features. In the example of Fig. 3, the user selects a cluster in the middle view, and detects that this feature is highly correlated with the feature at the right.

Time based filtering is common in many visualization systems today [7, 8]. As with the case of feature selection, there are two approaches for time filtering, use of algorithmic methods, or use of visual methods. Future research involves the investigation of two visual filtering methods, the use of 1D and 2D plots, as depicted in Fig. 4. In both these cases, one or two dimensions represent the time parameter, while the other, one feature, either selected from the available features, or created using different methods (as for example probability density function, entropy metrics, or feature extraction). Using this views, the analyst is able to observe the change of the data over time, detect patterns and finally focus on specific time periods for analysis using the available visualization methods.

4 Conclusions

This paper presented future research directions regarding the visualization of mobile data, towards the detection of anomalies in the NEMESYS project. More specifically, the k-partite graph visualization approach will be extended to map features onto the visual features of the nodes of the graph. Furthermore, novel abstraction and aggregation methods will be applied, so as to reduce the size of the graph and visualize only the most interesting events. New clustering approaches will be investigated in order to solve the attack attribution problem. Finally, novel filtering techniques will be developed based on visual methods so as to allow the analyst to focus on important parts of the data, reduce visual cluster, and ease the burden of the analysis of vast amounts of data.

References

1. Gelenbe E, Gorbil G, Tzovaras D, Liebergeld S, Garcia D, Baltatu M, Lyberopoulos G (2013) Nemesys: enhanced network security for seamless service provisioning in the smart mobile ecosystem. In: Proceedings of 28th international symposium on computer and information sciences (ISCIS'13), Oct 2013, accepted for publication
2. Delosieres L, Garcia D (2013) Infrastructure for detecting android malware. In: Proceedings of 28th international symposium on computer and information sciences (ISCIS'13), Oct 2013, accepted for publication
3. Baltatu M, D'Alessandro R, D'Amico R (2013) NEMESYS: first year project experience in Telecom Italia Information Technology. In: Proceedings of 28th international symposium on computer and information sciences (ISCIS'13), Oct. 2013, accepted for publication
4. Liebergeld S, Lange M (2013) Android security, pitfalls, lessons learned and BYOD. In: Proceedings of 28th international symposium on computer and information sciences (ISCIS'13), Oct. 2013, accepted for publication
5. Abdelrahman O, Gelenbe E, Gorbil G, Oklander B (2013) Mobile network anomaly detection and mitigation: the NEMESYS approach. In Proceedings of 28th international symposium on computer and information sciences (ISCIS'13), Oct. 2013, accepted for publication
6. Shiravi H, Shiravi A, Ghorbani AA (2011) A survey of visualization systems for network security. IEEE Trans Visual Comput Graphics 1(1):1–19
7. Shen Z, Ma KL (2008) Mobivis: a visualization system for exploring mobile data. In: Visualization symposium, 2008. PacificVIS'08. IEEE Pacific, pp 175–182, IEEE, 2008
8. Eagle N, Pentland A (2006) Reality mining: sensing complex social systems. Pers Ubiquit Comput 10(4):255–268
9. Lambert MJ (2005) Visualizing and analyzing human-centered data streams. PhD thesis, Massachusetts Institute of Technology
10. Tsigkas O, Tzovaras D (2012) Analysis of Rogue Anti-Virus Campaigns using hidden structures in k-Partite graphs. Cryptology and network Security, pp 114–125, Springer, Berlin
11. Tsigkas O, Thonnard O, Tzovaras D (2012) Visual spam campaigns analysis using abstract graphs representation. In: Symposium on visualization for cyber security, Seattle, WA, USA
12. Papadopoulos S, Moustakas K, Tzovaras D (2012) Hierarchical visualization of BGP routing changes using entropy measures. In: Bebis G, Boyle R, Parvin B, Koracin D, Fowlkes C, Wang S, Choi MH, Mantler S, Schulze J, Acevedo D, Mueller K, Papka M (eds) Advances in visual computing, vol 7432 of Lecture notes in computer science. Springer, Berlin Heidelberg, pp 696–705
13. Fuchs J, Fischer F, Mansmann F, Bertini E, Isenberg P et al (2013) Evaluation of alternative glyph designs for time series data in a small multiple setting. In: Proceedings of the conference on human factors in computing systems (CHI)
14. Zhao L, Yang J, Fan J (2009) A fast method of coarse density clustering for large data sets. In: 2nd international conference on biomedical engineering and informatics, 2009. BMEI'09, pp 1–5, IEEE
15. Adrienko N, Adrienko G (2011) Spatial generalization and aggregation of massive movement data. IEEE Trans Visual Comput Graphics 17(2):205–219
16. Vinh NX, Epps J (2010) Mincentropy: a novel information theoretic approach for the generation of alternative clusterings. In: IEEE 10th international conference on data mining (ICDM), pp 521–530, IEEE
17. Schaeffer SE (2007) Graph clustering. Comput Sci Rev 1(1):27–64

Infrastructure for Detecting Android Malware

Laurent Delosières and David García

Abstract Malware for smartphones have sky-rocketed these last years, particularly for Android platforms. To tackle this threat, services such as Google Bouncer have intended to counter-attack. However, it has been of short duration since the malware have circumvented the service by changing their behaviors. Therefore, we propose a malware taxonomy, a survey of attack vectors to better understand the Android malware, a survey of the modus-operandi of attackers for infecting the smartphones, and the design of components that are responsible for analyzing and detecting Android malware of the NEMESYS infrastructure. This infrastructure aims at understanding and detecting attacks both at the network and smartphone level.

1 Introduction

Malware represent a high threat for smartphone users since they affect all the plaftorms [1] by stealing personal data, banking data, transforming the smartphone in bots, etc. Most scaring, their number has tremendously exceeded the predictions for the year 2012 by a factor up to 300 % as shown by TrendMicro [2]. Furthermore, in 2012, their number has continuously increased as revealed by the F-Secure report [3].

Even though antiviruses for mobile phones and other services such as the service Google Bouncer for detecting Android malware exist, the number of infections is still very high, especially for Android platforms as shown by McAfee [4].

This paper is about introducing the infrastructure for analyzing and detecting malware, part of the NEMESYS infrastructure [5]. We propose a malware taxonomy

L. Delosières (✉) · D. García
Hispasec Sistemas S.L, 29590 Campanillas, Malaga, Spain
e-mail: ldelosieres@hispasec.com

D. García
e-mail: dgarcia@hispasec.com

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_38,

and a study of 38 Android malware samples to identify the most common Android families. We also survey the attack vectors i.e., the modus-operandi of attackers to infect the smartphones. Finally, we overview the NEMESYS infrastructure and show the current development of the infrastructure concerning the analysis and detection of Android malware.

The infrastructure comprises components both on the smartphone and on the server. The component on the smartphone logs the behavior of the Android programs which are instrumented by the user and uploads them to a server for an offload analysis, and early detects the malware. As for the component on the server side, it makes a more complete analysis of the malware and embeds a crawler in order to crawl the web searching for Android applications both goodware and malware.

The contributions of this paper are (1) proposing a taxonomy for the attack vectors, (2) surveying the different Android malware families, and (3) designing the components of the NEMESYS infrastructure that are responsible for analyzing Android malware.

The paper is structured as follows: Section 2 introduces the terms for understanding the article. Section 3 proposes a taxonomy of Android malware. Section 4 surveys the different Android malware families and a study of 38 samples is proposed in order to identify the main Android malware families. Finally, Sect. 5 overviews the NEMESYS infrastructure with an emphasis on the components for analyzing and detecting Android malware.

2 Background

GooglePlay [6], formerly called Android Market, is the official Android application store. It is accessible via the browser or via the Android application "Play Store". Both of them enable to browse Android applications on the official Google repository by their category, their name, and download them. Alternative stores for Android applications exist such as Amazon Appstore [7], GetJar [8], etc.

VirusTotal is a free service for analyzing samples by 44 different antivirus engines. The service issues a report containing the malware name if so, the hash ID of the sample (MD5, SHA1, and SHA-256), the initial filename, the type of file (e.g., an image), etc. For some samples, the sample behavior is also inserted in the report. By behavior, we intend the actions of the malware in the system such as the files that are read, written, the communications that are established, etc.

Recovery mode is a security added by each phone manufacturer in case the main filesystem gets corrupted. Recovery mode consists in dividing the media into two partitions: one for the system and the other one for the copy of the system with the factory settings. The recovery mode is also used for different purposes such as installing a custom image (e.g. CyanoGenMod [9]).

Exploits are used for installing automatically malware on the mobile device or crashing the device making it unavailable. For this, they need a software vulnerability, i.e., a flaw in a program that is exploited in order to inject and run external code.

An exploit can be used in every layer of a smartphone from the kernel layer up to the applications layer containing the browser application. By adding more functionalities to the device such as Near Field Communication (NFC), more code is added to the smartphone and therefore it is more likely to increase the number of exploits.

Every Android application is executed inside its own Dalvik Virtual Machine (VM) [10]. It is similar to Java VM but Dalvik byte codes are executed instead of Java byte codes. Unlike Java VM, Dalvik VM is a registered-machine. In other words, registers are used for passing functions' parameters while Java VM uses the stack instead. Dalvik VM has been used to keep the applications confined but mainly to keep the portability of Android applications between different smart-phones by abstracting their hardware.

Android Debug Bridge (ADB) is the application provided by Google in order to communicate with the phone from the computer. This application enables a user to install, remove, execute, and terminate applications.

3 Attack Vectors

A possible taxonomy for the attack vectors is as follows: physical attacks, social engineering, and exploits (applications, drivers and kernel).

3.1 Physical Attack

A physical attack consists in accessing physically the device in order to install or remove software. For the Android devices, an attacker can easily access it by connecting it through a USB[1] cable and use either the program *adb* provided by Google or the recovery mode of the phone to install or remove software.

In the former case, an attacker use the most common way for installing and removing software. The tool *adb* can be used for installing a program that will exploit a vulnerability of the Android OS. If the exploit is successful, the attacker can install a rootkit and therefore being stealthy.

In the latter case, the user can take advantage of the recovery mode of the phone to customize the phone firmware. By handcrafting a system image and overwriting the recovery image, the attacker can overwrite the system image by activating the recovery mode which will install the recovery image, i.e., the handcrafted system image. Nevertheless, the attacker needs to know both the manufacturer and the model version for crafting the correct image that will be accepted by the smartphone. Furthermore, this attack will wipe out the user data.

[1] Universal Serial Bus

3.2 Social Engineering

Social engineering is based on deception. It targets the human instead of the machine. The main techniques in the mobile environment are the phishing and pharming, and the application repacking attacks.

Attackers can use the phishing attack which consists in acquiring information such as usernames, passwords, and credit card details by masquerading as a trustworthy entity. For instance, the attacker sends an email to a user containing a link to a malicious server or to a repository [11]. Pharming [12] is also commonly used. It redirects the user to a fraudulent website by changing the victim's host file or compromising a DNS server, i.e. by poisoning it.

The application repacking is also very common. It consists in decompiling an application, injecting some malicious code, and re-compiling. Most people using mobile devices, even though aware of the presence of malware, cannot distinguish malware from goodware which benefits to the attacker. Therefore, the attackers can suggest malicious applications which look like good applications. For attracting users, generally the attackers propose a free version of an Android application.

3.3 Exploits

Since the beginning of Android, several vulnerabilities at every level of the smartphone have been revealed. For instance, vulnerabilities that were found in the drivers, in the NFC, SMS stack, and the Android browser.

Drivers added by the phone manufacturer turn out to be vulnerable. That is the case for instance of the buggy driver developed by Samsung [13]. The exploit uses a bug in the driver to gain root access to the phone i.e., the highest privileges.

Near Field Communication (NFC) has been shown as being vulnerable. Miller et al. [14] have injected and run some malicious code in an Android smartphone by means of the NFC. For this, they have exploited the memory corruption bug of an Android 2.3 to gain the control of the NFC daemon with a specific designed RFID tag. The MWR Labs hacking team [15] has used a zero day vulnerability on the NFC in order to hack the Samsung Galaxy S3. With the exploit, the team had access to the user data of the phone such as e-mails, SMS databases, the address book, the photo gallery, and access to third-party application data.

SMS can also be used to exploit a vulnerability of a smartphone [16]. For instance with the Android 1.5, a malformed SMS message can cause the crash of the system. The user is therefore forced to restart the mobile phone. A continuous reception of such SMS can be used as Denial of Service in which case the mobile phone becomes unavailable.

Android browser has also turned out to be vulnerable. Kurtz et al. [17] have presented a Remote Access Tool (RAT) which can infect the smartphones with Android

2.2 by exploiting a bug in the Android WebKit browser. Unlike RAT, BaseBridge [18] exploits a breach in ADB for elevating its privileges.

4 Malware Taxonomy

There is no standard taxonomy. For example, the taxonomy is different between the three main malware taxonomy standardization namely the Computer Antivirus Research Organization [19], Common Malware Enumeration [20], and Malware Attribute Enumeration and Characterization [21]. Therefore, we propose the following taxonomy w.r.t the malware behavior.

Bot transforms the smartphone in a robot waiting for commands from a Command and Control (C&C) server. Bot might be used to launch a Denial of Service (DoS), a spam campaign, etc.

SMS sender sends SMS which are mostly premium messages without the user awareness.

Infostealer steals private information such as personal user information, GPS location, or device technical data.

Privileges elevator elevates its privileges by exploiting a vulnerability. By gaining root privileges, malware may improve its stealthiness profile and operation capabilities on the resources of the device.

Rootkit uses stealth techniques for covering its traces. However, rootkits need root privileges.

Trojan bank steals banking data. The proliferation of online mobile bank application makes the device a target for scammers.

Adware displays ads, search results, application ratings or comments to the user.

Dropper installs a malware in either one stage or two stages. The one stage dropper consists in installing the malware that is camouflaged in the dropper whilst the second stage consists in downloading the malware from a remote source and installing it.

In order to know the common Android malware characteristics, we have analyzed a set of 38 Android malware. Out of these 38 samples, only 6 of them employ an exploit to elevate their privileges. Almost half of them send premium SMS as a fast way for earning money and transform the smartphone in a Bot. Most of the applications use Social Engineering for infecting the smartphones. Malware seen so far do not embed any anti-debugging or anti-virtual machines as the malware targeting Desktop PCs but we can expect changes in the near future.

5 Infrastructure

The infrastructure defined in the NEMESYS project aims at understanding and counterattacking attacks occurring in the network and in smartphones. The components in which we are involved in encompass a honeyclient and a data collector on the server

side, and a lightweight malware detector on the smartphone side. We shall first enumerate the role of each component, and then emphasize the current development of the honeyclient.

5.1 Components of the Infrastructure

The lightweight malware detector aims to detect Android malware on the smartphone. It will be efficient and lightweight in terms of memory and CPU consumptions. The detection will be based on two ways: signature and abnormality behavior based. The detector will also interact with the mobile honeypot and the data collection infrastructure in order to download the latest malware signatures and upload traces.

The high-interaction honeyclient development is responsible for interacting with web servers searching for attacks. It is consisted of three components: a queuer which is responsible for creating a list of servers for the client to visit, a client which will run an Android emulator and record traces, and a malware detector engine which will analyze the traces and detect if an attack has occurred. It will also interact with the data collection infrastructure so as to save the attack traces.

As for the data collector, it aims to collect/provide the data from/to the different entities. The data will be a collection of attack traces detected by the lightweight malware, the honeyclient, etc. This infrastructure will expose an interface so that all the entities can interact with it. It will be scalable and therefore will be able to process a highly number of requests for accessing and storing data. It will also enrich the raw data that has been stored (e.g., OS fingerprint, GeoIP, etc.).

5.2 Honeyclient

The honeyclient consists of a static and dynamic analyzer. The static analyzer is provided by Androguard which will extract the characteristics of an Android application whilst the dynamic analysis is done by DroidBox. Both analyzers will work jointly in order to extract as many Android characteristics as possible.

Androguard [22] provides a framework for analyzing statically Android applications. It can retrieve the Android permissions requested by an Android application, extract the list of called functions, re-assemble an application, detect the presence of ads and obfuscators making the analysis harder, etc.

Droidbox [23] automates the analysis of an Android application in the Android emulator [24]. When executed, the Android application calls Android functions that are provided by the Android framework. A subset of those functions are hooked by DroidBox and outputs a log when executed. This subset encompasses the functions that are sending out data, reading from, or writing into files. The output log is retrieved by means of the tool Android Debug Bridge (ADB) [25] and then parsed in order to extract the logs generated by DroidBox. Droibox also includes the module

TaintDroid [26] for tracking the information leaks. In other words, every sensitive information that is sent out (e.g., a phone number in a SMS [2]) is logged.

For hooking the functions, DroidBox w.r.t to its version provides two ways: the former one consists in modifying the Android firmware, i.e., modifying and recompiling the Android firmware while the latter one consists in modifying the Android applications in order to add the hooks inside this application. However like malware for Desktop PCs, Android malware can use techniques for hiding the functions and load them dynamically [27] and therefore bypass the Droidbox hooking. As for the second one, a malware can bypass the hooking at the Dalvik level by instantiating a C program which interacts directly with the native functions. However, for calling most of the native functions, the user must have the root access which is not set by default when an application is launched. As for the applications that use elevation of privileges, they represent a minority and generally exploit vulnerabilities of old Android platforms. Therefore, in order to limit the circumventions, we chose the latter way with a new Android emulator Ice Cream Sandwich (ICS). Nevertheless, in order to capture most information about the network traffic, the traffic will be captured outside the emulator.

In order to get an analysis as complete as possible of the malware, the honeyclient is instrumented by a framework integrating a virtual user simulator and a user actions recorder [28]. The simulator is built from a model which is based on eight different reproducible scenarios composed of about five user actions each. An action is a click on a button, a swipe, etc. We chose the application Facebook, Hotmail, Youtube, Calendar, Gallery ICS, Browser, Slide Box Puz, and talk.to since they encompass most of the user actions and are amongst the most downloaded. All those applications were running in the Android ICS emulator. Before recording the user actions, the Android has been set up, i.e., an account has been created for the applications Facebook, Hotmail, and talk.to.

The scenarios were adapted for each application in the respective order Facebook, Hotmail, Youtube, Calendar, Gallery ICS, Browser, Slide Box Puz, and talk.to:

- The user (1) signs in, (2) posts a new message, (3) goes to the list of messages by clicking on the top icon Messages, and then (4) opens the first message.
- The user (1) signs in, (2) clicks on the first message, (3) scrolls down, and then (4) presses the button “Home” to go back to the Android board.
- The user (1) scrolls down the list of videos, (2) plays one video, (3) increases the volume, (4) decreases the volume, (5) goes back to the list of videos by clicking on the button “Back”, (6) scrolls up the list, and then (7) plays another video.
- The user (1)signs in (2) selects one day with a long pressing, (3) types the name of an event, (4) clicks on Save, and then (5) goes back to the Android board by clicking on the button “Home”. (6) signs out.
- The user (1) selects one album, (2) selects one photo, (3) zooms in, and then (4) zooms out.
- The user (1) selects main menu in Android, (2) clicks on the browser icon, (3) selects keyboard, and then (4) types “www.peugeot.com”.

[2] Short Message Service

- The user (1) moves the ball left, right, up, and down for the 1st level.
- The user (1) selects a contact, (2) types a message, (3) presses the button Settings, and then (4) presses the item End chat.

Since future Android malware might embed virtual machines detectors by checking the GPS locations or track the accelerometers changes for instance, we will besides the previous scenarios, inject some random events so as to make the phone look more real to the Android applications. For instance, events of the accelerometer or the GPS locations will be injected into the Android emulator.

6 Related Works

To tackle the sky-rocketing number of malware these last years, Google has set up a service called Bouncer [29] for scanning and detecting the potential malware present on Google Play. This service analyzes the uploaded Android applications both statically and dynamically. With the former analysis, Google detects the malware by their signature while with the latter analysis, it detects the malware by their misbehavior. This service also prevents known attackers to submit other Android applications to Google Play. Nevertheless, this service can be easily circumvented as shown by Miller et al. [30]. For instance, a malware can mitigate its malicious behavior during the analysis by Google Bouncer.

Google has also introduced the notion of permissions for mobile devices in order to inform the user about the intentions of the program. When the application wants to access a resource of the system (e.g. network), it must ask the authorization of the system, i.e., mandatory access control policy. This mechanism prevents malware from using resources whose access rights are prohibited. The rights are granted by the user upon installing the application. However, most of the time the user does not understand those permissions [31].

Even though mobile antiviruses for mobile phone exist, they turn out to be inefficient as shown by Zhou et al. [32]. In their study that was based on 1,200 malware samples representing all the Android families from 2010 to 2011, they showed that the best antivirus engine detects at most 79.6 % of Android malware while the worst case detects only 20.2 %. This also reveals that methods to circumvent detections for Android malware evolve rapidly.

7 Conclusion

This paper states the state of the art in Android malware and describes the current development of the NEMESYS infrastructure aiming to understand and detect attacks both at the network and smartphone level. First, we have surveyed the different attack vectors in Android environment. Secondly, we have proposed a taxonomy of

Android malware. From the analysis of 38 malware, we have found that the social engineering is the main attack vector for infecting smartphone and that most malware were bots, infostealers, and SMS senders. Thirdly, we have overviewed the overall architecture of the NEMESYS infrastructure, with an emphasis on the components used for detecting Android malware, namely the honeyclient, the data collector, and the lightweight malware detector. Finally, we have shown the current developement of the honeyclient which is composed of both a static and dynamic analyzer in order to extract as many Android characteristics as possible. The honeyclient is instrumented by a virtual user in order to have an Android application analysis as complete as possible. In the future, we are planning to extract the characteristics of the malware and goodware datasets by the honeyclient, and continue the development of the framework, i.e., building the lightweight malware detector in the smartphone, the malware detector and the queuer in the honeyclient, and the data collector.

Acknowledgments The work presented in this paper is funded by the European Commission FP7 collaborative research project NEMESYS (Enhanced Network Security for Seamless Service Provisioning in the Smart Mobile Ecosystem), no. 317888 within the Trustworthy ICT domain.

References

1. Global smartphone installed base forecast by operating system for 88 countries: 2007 to 2017 (2012). https://www.strategyanalytics.com/default.aspx?mod=reportabstractviewer&a0=7834
2. Android under siege: Popularity comes at a price (2012). http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/reports/rpt-3q-2012-security-roundup-android-under-siege-popularity-comes-at-a-price.pdf
3. Mobile threat report q4 2012 (2012). http://www.f-secure.com/static/doc/labs_global/Research/MobileThreatReport Q4 2012.pdf
4. Mcafee threats report: Third quarter 2012 (2012). http://www.mcafee.com/it/resources/reports/rp-quarterly-threat-q3-2012.pdf
5. The nemesys project (2012). http://www.nemesys-project.eu/nemesys/index.html
6. Google play (2013). https://play.google.com/store
7. Amazon appstore for android (2013). http://www.amazon.com/mobile-apps/b?ie=UTF8&node=2350149011
8. Getjar (2013). http://www.getjar.com/
9. Cyanogenmod (2013). http://www.cyanogenmod.org/
10. D. Bornstein. Dalvik virtual machine internals (2008). http://de.youtube.com/watch?v=ptjedOZEXPM
11. Netcraft. Angry birds impersonated to distribute malware (2013). http://news.netcraft.com/archives/2013/04/12/angry-birds-impersonated-to-distribute-malware.html
12. Karlof C, Shankar U, Tygar JD, Wagner D (2007) Dynamic pharming attacks and locked same-origin policies for web browsers. In: Proceedings of the 14th ACM conference on Computer and communications security, CCS '07, ACM, New York, USA, pp 58–71. doi:10.1145/1315245.1315254, http://doi.acm.org/10.1145/1315245.1315254
13. The samsung exynos kernel exploit–what you need to know (2012). http://www.androidcentral.com/samsung-exynos-kernel-exploit-what-you-need-know
14. Black hat hacker lays waste to android and meego using nfc exploits (2012). http://www.extremetech.com/computing/133501-black-hat-hacker-lays-waste-to-android-and-meego-using-nfc-exploits

15. Naraine R (2012) Exploit beamed via nfc to hack samsung galaxy s3 (android 4.0.4) (2012). http://www.zdnet.com/exploit-beamed-via-nfc-to-hack-samsung-galaxy-s3-android-4-0-4-7000004510/
16. Google fixes sms crashing bug in mobile os (2009). http://www.theregister.co.uk/2009/10/12/google_android_security_update
17. Android smartphones infected via drive-by exploit (2012). http://www.h-online.com/security/news/item/Android-smartphones-infected-via-drive-by-exploit-Update-1446992.html
18. Revealed! the top five android malware detected in the wild (2012). http://nakedsecurity.sophos.com/2012/06/14/top-five-android-malware/
19. Caro naming scheme (2013). http://www.caro.org/naming/scheme.html
20. Cme common malware enumeration (2013). http://cme.mitre.org/about/faqs.html
21. Maec malware attribute enumeration and characterizacion (2013). http://maec.mitre.org/
22. Androguard (2013). https://code.google.com/p/androguard/
23. Droibox (2013). https://code.google.com/p/droidbox/
24. Using the android emulator (2013). http://developer.android.com/tools/devices/emulator.html
25. Android debug bridge - android developer documentation (2012). http://developer.android.com/tools/help/adb.html
26. Enck W, Gilbert P, Chun BG, Cox LP, Jung J, McDaniel P, Sheth AN (2010) TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones. In: Proceedings of the 9th USENIX conference on operating systems design and implementation, OSDI'10, USENIX Association, Berkeley, CA, USA, pp 1–6. http://dl.acm.org/citation.cfm?id=1924943.1924971
27. Tamada H, Nakamura M, Monden A, Matsumoto KI (2008) Introducing dynamic name resolution mechanism for obfuscating system-defined names in programs. In: Proceedings of the IASTED international conference on software engineering, SE '08, ACTA Press, Anaheim, CA, USA, pp 125–130. http://dl.acm.org/citation.cfm?id=1722603.1722627
28. P. Moschonas, N. Kaklanis, D. Tzovaras (2011) Novel human factors for ergonomy evaluation in virtual environments using virtual user models. In: Proceedings of the 10th international conference on virtual reality continuum and its applications in industry, VRCAI '11, ACM, New York, NY, USA, pp 31–40. doi:10.1145/2087756.2087760, http://doi.acm.org/10.1145/2087756.2087760
29. Android and security (2012). http://googlemobile.blogspot.com.es/2012/02/android-and-security.html
30. Circumventing google's bouncer, android's anti-malware system (2012). http://www.extremetech.com/computing/130424-circumventing-googles-bouncer-androids-anti-malware-system
31. Kelley PG, Consolvo S, Cranor LF, Jung J, Sadeh N, Wetherall D (2012) A conundrum of permissions: installing applications on an android smartphone. In: Proceedings of the 16th international conference on financial cryptography and data security, FC'12, Springer, Berlin, Heidelberg, pp 68–79. doi:10.1007/978-3-642-34638-56, http://dx.doi.org/10.1007/978-3-642-34638-5_6
32. Zhou Y, Jiang X (2012) Dissecting android malware: characterization and evolution. In: Proceedings of the 2012 IEEE symposium on security and Privacy, SP '12, IEEE Computer Society, Washington, DC, USA, pp 95–109. doi:10.1109/SP.2012.16, http://dx.doi.org/10.1109/SP.2012.16

NEMESYS: First Year Project Experience in Telecom Italia Information Technology

Madalina Baltatu, Rosalia D'Alessandro and Roberta D'Amico

Abstract In 2011, when mobile malware quadrupled in volume, it became clear that malware for mobile platforms would experience a rapid increase. During 2012 it continued to grow steadily, closely following the spreading of smartphones all over the world. For mobile network operators it is also clear that this growing phenomenon must be at least constantly monitored, if not prevented altogether. At the present moment there are no effective tools to achieve this purpose. NEMESYS comes to fill in this vacuum. This paper presents the vision, position and activities of Telecom Italia Information Technology as part of the NEMESYS project. The emphasis is given to the perception and practical experience of mobile threats and mobile security in Telecom Italia Mobile Network, and on the advances in the art that our company is hoping to achieve from its active participation to the project.

1 Introduction

Nowadays smartphones are ubiquitous, their usage continues to grow all over the world. With the International Telecommunication Union (ITU) estimating global mobile subscriptions at 6 billion at the end of 2011, it is calculated that global smartphones penetration is now 16.7 % [1]. Smartphones are devices built on full-fledged operating systems, with advanced computing capabilities and enhanced connectivity (3G/4G, Wi-fi, bluetooth). They are also personal digital assistants, media players,

M. Baltatu (✉)
Security Lab, Telecon Italia Information Technology, via Reiss Romoli 274,
10148 Turin, Italy
e-mail: madalina.baltatu@it.telecomitalia.it

R. D'Alessandro
e-mail: rosalia.dalessandro@it.telecomitalia.it

R. D'Amico
e-mail: roberta.damico@it.telecomitalia.it

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_39,

compact digital cameras, video cameras, GPS navigation devices, and even tuners for musical instruments. Smartphones are also beginning to be used for direct paying (like debit cards), and to get access to enterprise premises. They are all equipped with web browsers and other network applications that use high-speed Wi-fi data access and mobile broadband, or proximity bluetooth access. Mobile application developers have immediately understood the opportunity presented by smartphones deployment, and, at the present time, mobile applications stores are the major drivers of smartphones adoption in everyday life.

Unfortunately, smartphones are also becoming attractive for cyber criminals and malware developers. Ever since 2011, mobile malware has started to grow steadily. It seems that the trend is similar to that followed by malware developed for personal computers, but in a much faster way. Moreover, differently from traditional computer platforms, smartphones are natively a source of profit (since they have available the users' phone and data traffic credit), and this makes them a great target of attacks. The spreading of smartphones also imply that Mobile Network Operators (MNOs) are required to provide appropriate protection and security mechanisms to their core network and, if possible, to their customers' devices. Telecom Italia is aware of the potential threat compromised smartphones represent to mobile networks. This is the main rational that motivates our presence in the NEMESYS project. NEMESYS aims to respond to these challenges by designing a comprehensive security infrastructure able to offer protection to both devices and mobile networks. The material presented in this paper is organised as follows: Section 2 presents some statistics on mobile platforms market penetration and mobile malware spreading during the last years. Section 3 describes the main activities related to mobile security in Telecom Italia Information Technology. Section 4 presents the first year participation of our organization to the NEMESYS project, while the last Section contains the concluding remarks.

2 Mobile Platforms and Malware Statistics

In this section we present some significant statistics computed from worldwide data, on mobile platforms distribution and malware spreading, in order to understand the importance of mobile security.

2.1 Mobile Platforms Statistics

At the end of 2011 the smartphones market penetration worldwide has reached and surpassed one billion units [1]. In 2012 Andorid and iOS account for the significant majority of the global smartphone installed base: [2] shows that these platforms represented 92 % of global smartphone shipments in the fourth quarter of 2012. Furthermore, it appears that smartphones represent the technology that is spreading

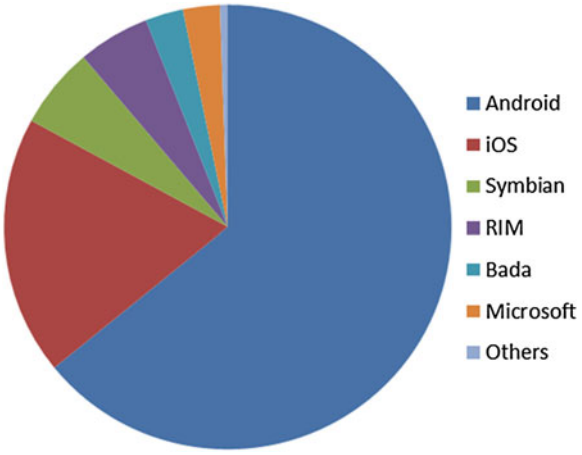

Fig. 1 The mobile market: the shares of the main mobile OSes in 2012

faster than any other technology in human history except for television [3], and this happens even in developing countries.

Figure 1 illustrates the market shares of the main mobile operating systems in the first half of 2012, as presented in [4]. We can see that in the last two years, the four most popular mobile platforms are, in order: Android, Apple iOS, Symbian OS and RIM (Blackberry).

Statistics computed from real data collected in the mobile network of Telecom Italia during a single day in 2012 (the first half of the year) shows the following distribution of the mobile operating systems of the devices registered to the network: most of the registered phones still run the Symbian OS, they are followed by iOS and Android. The distribution per OS of the traffic coming from these devices shows that 63 % of the network data traffic is generated by iOS smartphones, followed by Android devices.

By the end of 2012, the situation changed significantly: Android surpassed both Symbian and iOS, with 31 % of the terminals registered to the network running Android, 29 % Symbian OS and 21.6 % iOS. BlackBerry is at the 4th place, while a small number of devices run Nokia OS, Windows Phone, and Bada OS. The data traffic is still mainly generated by iOS phones, but Android is following up very fast.

According to a six-month study during 2012 presented in [5], 67 % of the measured web traffic during this time period came from iOS devices. Android accounted for about half of the overall traffic. As we may see, these findings are in line with the statistics performed on instantaneous data collected in Telecom Italia mobile network. In February 2013, another report [6] shows that Android took the lead from iOS in mobile data traffic.

2.2 Malware Statistics

Mobile malware spreading increased steadily all along 2012, overriding the predictions [7]. According to public statistics presented in [8], the malware volume doubled in the last quarter of 2012 if compared to the same period one year before.

An interesting view on the phenomenon is offered in [9], which provides an image of the mobile malware spreading during 2012, where Android is the incontestable 'leader': 98.96 % of all malware is Android malware! This situation is also illustrated in [8], which shows Android at 97 %, followed by Symbian and Java ME.

Even if the actual figures change slightly from one malware statistics to another, we can note that, Android always holds the leadership. Popularity comes at a price: the most open and the most spread mobile OS at the moment is also the preferred target of malware. Malware rates for the other platforms are so insignificant that, in the majority of malware reports, the statistics are only shown for Android.

2.3 Malware Classification

Usually, an application is classified as malware if it performs one or more of the following actions: leaks device or personal information (including user credentials), or spies on users activity; sends premium rate SMS messages, makes premium rate calls, makes subscription to paid services; exploits a vulnerability or software bug on the device to cause it to do something the user does not expects; roots (or jailbreaks) the device to give the attacker control over it; installs a backdoor or turns the device into a bot client; downloads a secondary piece of malicious code from a website (using the http/https channel) or an arbitrary remote server; is destructive to users device or data stored therein; sends spam messages via SMS or spam e-mails from the device; steals private users information and publish it on the Internet, demanding a price to delete it. It is also interesting to take into consideration malware classifications implicitly proposed by the major mobile antivirus companies in their periodical reports. For example, [10] estimates that, from more than six million people affected by Android malware from June 2011 to June 2012, many were affected by Toll Fraud applications. The prevalence of Toll Fraud malware grew from 29 % of the application-based threats in the third quarter of 2011 to more than 62 % in the second quarter of 2012. The classification proposed in [10] is: Toll fraud, Bot client, App Downloader, Infostealer, Contact Spammer, Rooter, Destructive.

In [11], while describing the mobile security trends in 2013, a malware classification is proposed, based on the main malware behaviour patterns: Info Stealers, Spyware, Adware, Premium SMS, Fraud, Exploit, Rooting Malware, Backdoor/Botnet, Hacktool, Downloader/Installer, Destructive, SMS Spam. The distribution in the wild of these typologies is also given for a long period of observation, from 2007 to 2012, where Info stealers, Spywares, SMS senders and Adware are placed at the top of the list.

3 Mobile Security in Telecom Italia Information Technology

In the followings, we offer an overview on some ongoing activities in Telecom Italia Information Technology (Security Lab) in the field of mobile security.

3.1 SMS Spam Reporting Service

During 2012 Security Lab developed a prototype spam reporting service, that helps the operator to identify the spam received by mobile users over the SMS channel. The service is specified by the GSMA [12], which states that a mobile network operator has to dedicate a specific short number in order for its customers to be able to report any SMS they received which they consider spam.

At the beginning of 2013 we started a trial of this service (implemented on Android platforms), dedicated exclusively to employees. The idea is to understand what is the actual level of SMS spam received by this category of users, and, also to investigate if such initiatives of participative security services are well accepted by the users. The next steps will be to extend the trial to other categories of customers, and, also to evaluate possible countermeasures to deploy in order to mitigate the problem.

3.2 Mobile Malware and Application Analysis

Security Lab started to study mobile malware in a systematic manner (and as a separate phenomenon from generic PC malware) since 2010/2011, when mobile malware displayed a significant growth (mainly for Android platforms).

Since Android is the preferred target of attacks, an automated applications analyser was developed to evaluate the potential danger of an Android application package (*apk*). The system implements static analysis techniques to obtain a detailed application's behaviour description together with a comprehensive risk value, and uses and extends the Androguard framework [13].

Briefly, the system looks at all APIs used by the application and maps them to the requested permissions (declared in the application's Manifest) [14], in order to detect incoherencies between them. Our work enhances a similar approach proposed in [15], by exhaustively checking whether the declared permissions are effectively used, and whether actions that are not explicitly permitted are performed (in order to avoid permission escalation). We also look for critical APIs usage and Intents abuse. Furthermore, we propose a risk taxonomy and a mapping between the application behaviour and this set of risks. Briefly, the most relevant risks are related to the root privileges escalation, the use of encrypted code, the presence of binary code and/or dynamic code loading, Internet activities, the presence of exploits, the use of dangerous APIs, SMS receipt, sending and interception, phone call activities, user privacy violations (leakage of device and user information), the presence of

critical system permissions, and the monitoring and/or modification of the device state (e.g., phone state, network state, active tasks, etc.). Some activities (SMS, calls and Internet) are also related to the economic loss risk.

A detailed analysis of the apk archive is also performed, in order to detect potential threats, like embedded applications, infected files (e.g., apk or elf binary libraries already classified as malicious), and shell scripts with potentially dangerous commands. Many malware applications attempt to conceal their purposes. Often, they alter files with some innocuous extension (e.g., png or jpg). Morover, the system is also able to look for URLs and phone numbers, which might be used by the application to communicate with C&C servers or spend the user's money by making calls or sending SMS messages to premium numbers.

The risk computation extends the original *risk.py* module implemented in Androguard by adding additional risks categories, which concur to compute the global risk score. This value is computed by combining all the risk values in a *fuzzy* [16] system.

During 2012, some interesting statistics have been computed, based on real apk data organized in two databases. The first database is a set of 41,000 free applications downloaded from GooglePlay, while the second contains 1,488 known malware samples classified in 90 distinct families, most of them available on ContagioMiniDump [17].

Figure 2 shows the risk scores distribution for these two datasets. We can see that free applications from GooglePlay are concentrated in the score intervals from 60 to 80, while malware in the intervals which goes from 70 to 90. There is a significant overlapping window which can imply both false positives and false negatives in anomaly based malware detection systems. In our experience, applications that obtain security risk scores major than 70 are to be considered potentially damaging for the device and its user.

Moreover, legitimate applications obtained unexpected high risk values on several categories like dynamic, exploit, root privileges, dangerous APIs, while the malware set obtained high values on economic loss, Internet, SMS, and privacy violation categories, and significant risks on their archive files. As far as malware is concerned,

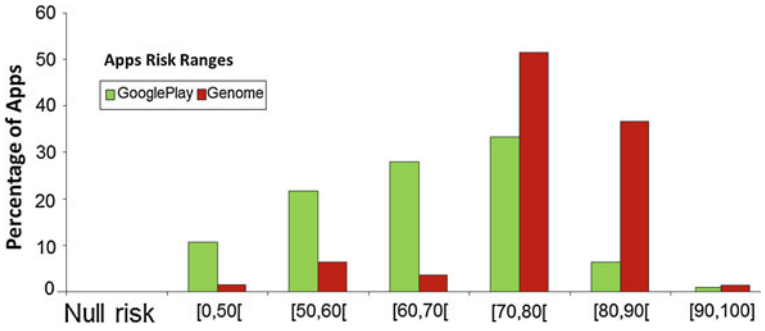

Fig. 2 Risk scores distribution for malware and legitimate apps

the privacy violation is the most significant risk encountered, while, for applications downloaded from GooglePlay, the dangerous API usage risk is the highest.

These results show that, quite often, a legitimate free application is not as innocuous as users may believe. This may be an effect of either poor programming or the presence of potentially unwanted code (mainly related to adware or due to recycled code).

4 The First Year Participation in NEMESYS

The participation in the NEMESYS European project is considered a great opportunity in Telecom Italia Information Technology, since this project can provide effective tools for mobile malware monitoring and infection prevention. The value of the NEMESYS approach if compared to the existing approaches nowadays consists in the fact that it takes into consideration a plethora of input information sources to offer a better response to incidents, together with a prevention mechanism. Current mobile security solutions are entirely reactive and non predictive. We envision that NEMESYS can become the starting point for MNOs to cooperate in providing an extended mobile malware response and prevention network.

The goal of NEMESYS is to create and develop new security technologies in mobile networks. These technologies are meant to protect both terminals (in particular smartphone devices) and the network core elements. Mobile security is a fast moving field, where new vulnerabilities and their exploits need to be detected and analyzed on a (quasi) real time basis. In order to advance in the field of mobile security, the new technologies must become proactive and work on predicting threats and vulnerabilities. Ideally, the defences must be built before threats materialize. Therefore, the NEMESYS's purpose is to gather and analyse information about the nature of attacks targeting smart mobile devices, so that appropriate countermeasures can be taken to prevent all potential damage (to the core network and devices themselves). NEMESYS will adopt the honeypot scheme for the most popular smartphone platforms. An infrastructure will be developed to collect all susceptible information (possible attack traces), detect and provide early warning of attacks on mobile devices and mobile networks. By correlating the extracted information with the known patterns of attacks extracted from wireline networks, NEMESYS plans to reveal and identify the possible synergies between the two ecosystems (wired and wireless).

The first activity related to this kind of realization is the compilation of a thorough state of the art in security threats and attacks against mobile devices and in the field of analysis of current practices. The state of the art and the trends in mobile malware are to be closely monitored during all the project life time.

An important activity that TIIT will continue to perform inside NEMESYS is the active monitoring of mobile malware spreading in its own mobile network. To this purpose, TIIT will leverage the deployment of mobile honeypots, in order to better understand the phenomenon of mobile malware spreading and to offer optimal protection to the mobile network and its users.

A honeypot is a computer system, built and deployed only for the goal of being attacked and compromised, in order to study new attacks and to serve as an early warning system [18]. A mobile honeypot is a new concept in network security. At the present moment, the majority of honeypots are PC-based, at best they only simulate a mobile environment (like Android and iOS). Security Lab already deploys a PC-based passive honeypots that emulate Android and iOS responses for some services.

Nevertheless, PC-based or emulated environments are to be considered far insufficient in order to have a real perception of mobile malware. To get the pulse of the situations the honeypot has to actually become mobile and collect all the activities that the users perform on their devices. Since we consider this approach of great importance, TIIT will have an active role in all the processes that are related to testing all practical instruments provided by NEMESYS (both mobile honeypots and the virtualization mechanisms proposed by our partners).

TIIT is also involved in the definition of both system requirements and framework architecture. We hope to bring a significant contribution to these two tasks, leveraging our experience on mobile network systems and our interaction with customers, which helps us to be very sensitive to our real end users' needs.

5 Conclusion

The NEMESYS project represents a great challenge and a great opportunity for Telecom Italia Information Technology. In our vision, NEMESYS can change the way people look at mobile security. This project tries to modify the paradigm that mobile security has adopted all along its first years of existence. Mobile security is basically deployed on terminals only (antiviruses, mobile security suites and the same). Some research work was proposed in order to begin the monitor at a more centralized level (through IP-based honeypost that emulate a mobile OS). But, at the present time, there is no correlation between these mechanisms, and no correlation between these and mobile network security mechanisms (if they exist at all). NEMESYS tries to combine the device-side security instruments with network side anomalies monitoring, in order to offer an effective global security to all mobile network components. In our opinion, both mobile users and MNOs can benefit from this approach.

Acknowledgments The work presented in this paper is part of the Project NEMESYS (Enhanced Network Security for Seamless Service Provisioning in the Smart Mobile Ecosystem) which has received funding from the European Union Seventh Framework Programme (FP7) under grant agreement 317888.

References

1. mobiThinking, Global mobile statistics 2012 Part A, http://mobithinking.com
2. Strategy Analytics (2012) Android and Apple iOS capture a record 92 percent share of global Smartphone shipments in Q4 2012, http://blogs.strategyanalytics.com
3. Technology Review, Are SmartPhones spreading faster than any technology in human history? http://www.technologyreview.com
4. Gartner (2012) Market share: mobile devices, Worldwide. http://www.gartner.com/resId=2117915
5. Insights Chitika (2012) Study six-month Apple iOS users consume growing amount of Web traffic
6. InfoWorld (2013) Android takes the lead from iOS in mobile data traffic. http://www.infoworld.com
7. Trend Micro (2012) Mobile malware surged from 30K to 175K, Q3. http://www.trendmicro.com
8. McAfee Threats Report (2012) Fourth quarter. http://www.mcafee.com
9. Kaspersky Labs (2012) Kaspersky security bulletin 2012. The overall statistics for 2012. http://www.securelist.com/en/analysis/204792255
10. Lookout Inc. US (2012) State of mobile security 2012. https://www.lookout.com/resources/reports/state-of-mobile-security-2012
11. McAfee (2013) Mobile security: McAfee consumer trends report 2013. http://www.mcafee.com
12. Gsma, SPAM Reporting Services. http://www.gsma.com/technicalprojects/gsma-spam-reporting-services
13. Androguard, Reverse engineering, Malware and goodware analysis of Android applications. http://code.google.com/p/androguard/
14. Android Permissions. http://developer.android.com
15. Apvrille A, Strazzere T (2012) Reducing the window of opportunity for Android malware Gotta catch 'em all. J Comput Virol 8(1–2):61–71
16. Jan Jantzen (2008) Tutorial on fuzzy Logic. Technical University of Denmark, Oersted-DTU, Automation, Bldg 326, 2800 Kongens Lyngby, DENMARK. Technical report no 98-E 868 (logic). Revised 17 June 2008
17. Contagio malware, Mobile malware mini dump. http://contagiominidump.blogspot.it/
18. Mulliner C, Liebergeld S, Lange M (2011) Poster: HoneyDroid—creating a SmartPhone Honeypot. In: IEEE symposium on security and privacy, March
19. Portio Research Ltd. UK (2012) Mobile Factbook 2012. http://www.portioresearch.com
20. ZDNet, iOS users generate twice as much web traffic than Android users. http://www.zdnet.com
21. F-Secure Labs (2012) Mobile threat report Q3 2012. http://www.f-secure.com
22. Droidbox. Android application Sandbox. http://code.google.com/p/droidbox/

Android Security, Pitfalls and Lessons Learned

Steffen Liebergeld and Matthias Lange

Abstract Over the last two years Android became the most popular mobile operating system. But Android is also targeted by an over-proportional share of malware. In this paper we systematize the knowledge about the Android security mechanisms and formulate how the pitfalls can be avoided when building a mobile operating system.

1 Introduction

Smartphones are now very popular. Aside from calling and texting, people use them for connecting with their digital life–email, social networking, instant messaging, photo sharing and more. With that smartphones store valuable personal information such as login credentials, photos, emails and contact information. The confidentiality of that data is of paramount importance to the user because it might be abused for impersonation, blackmailing or else. Smartphones are very attractive for attackers as well: First, attackers are interested in the precious private information. Second, smartphones are constantly connected, which makes them useful as bots in botnets. Third, smartphones can send premium SMS or SMS that subscribe the victim to costly services, and thus directly generate money for the attacker. It is up to the smartphone operating system (OS) to ensure the security of the data on the device. In the last two years Android became the most popular mobile OS on the market. With over 1.5 million device activations per day Android is expected to cross the one billion active device barrier in 2013. Its world wide market share has reached 70 % of all smartphones. On the downside Android also became a major target for mobile malware [19]. Interestingly the share of mobile malware that targets Android is around 90 %, which is larger than its market share. The question is why is the Android platform so attractive for malware authors? In this paper we investigate the Android architecture and the security mechanisms it implements. Android and

S. Liebergeld · M. Lange
Berlin, Germany

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_40,

its weaknesses have already been well researched and we systematize the results and give advice for platform designers to avoid those pitfalls in the future. Our contributions are:

Android security mechanisms: We describe the Android architecture from a security point of view and give details on application and system security. We further detail the mechanisms of Android that are targeted at fending off attacks.

Android security problems: We identify the inherent security problems of the Android platform.

2 Android Platform Security

Android runs on a wide range of devices and Android's security architecture relies on security features that are embedded in the hardware. The security of the platform depends on a secure boot process.

Secure Boot The boot process of an Android device is a five-step process. First the CPU starts executing from its reset vector to which the initial bootloader (IBL) code from the ROM is wired. Then the IBL loads the bootloader from the boot medium into the RAM and performs a signature check to ensure that only authenticated code gets executed. The bootloader loads the Linux kernel and also performs a signature check. The Linux kernel initializes all the hardware and finally spawns the first user process called *init*. Init reads a configuration file and boots the rest of the Android user land.

Rooting In general, mobile devices are subject to strict scrutiny of the mobile operators. That is it employs secure boot to ensure that only code is being booted, that has received the official blessing in the form of a certification from the operators. This is being done to ensure that the mobile OS's security measures are implemented and the device does not become a harm to the cellular network.

Rooting involves a modification to the system partition. Modifications to the system partition require root permissions, which are not available by default. There are two ways of obtaining root permissions: Either the customer boots a custom system that gives him a root shell, or he exploits a vulnerability to obtain root permissions at runtime.

Rooting, voluntarily or involuntarily has repercussions on device security. Unsigned kernels can contain malware that runs with full permissions and is undetectable by anti-virus software (*rootkits*). Further, rooted devices do not receive over the air updates. If an application has received root permissions, it can essentially do as it pleases with the device and its data, including copying, modifying and deleting private information and even bricking the device by overwriting the bootloader.

3 Android System Security

The flash storage of an Android device is usually divided into multiple partitions. The system partition contains the Android base system such as libraries, the application runtime and the application framework. This partition is mounted read-only to prevent modification of it. This also allows a user to boot their device into a safe mode which is free of third party software.

Since Android 3.0 it is possible to encrypt the data partition with 128 bit AES. To enable filesystem encryption the user has to set a device password which is used to unlock the master key.

Data Security By default an application's files are private. They are owned by that application's distinct UID. Of course an application can create world readable/writable files which gives access to everybody. Applications from the same author can run with the same UID and thereby get access to shared files. Files created on the SD card are world readable and writable. Since Android 4.0 the framework provides a Keychain API which offers applications the possibility to safely store certificates and user credentials. The keystore is saved at `/data/misc/keystore` and each key is stored in its own file. A key is encrypted using 128-bit AES in CBC mode. Each key file contains an info header, the initial vector (IV) used for the encryption, an MD5 hash of the encrypted key and the encrypted data itself. Keys are encrypted using a master key which itself is encrypted using AES.

4 Android Application Security

In Android application security is based on isolation and permission control. In the picture you can see, that there are processes that run with root privileges. Zygote is the prototype process that gets forked into a new process whenever a (Java) application is launched. Each application runs in its own process with its own user and group ID which makes it a *sandbox*. So, by default applications cannot talk to each other because they don't share any resources. This isolation is provided by the Linux kernel which in turn is based on the decades-old UNIX security model of processes and file-system permissions. It is worth noting that the Dalvik VM itself is not a security boundary as it does not implement any security checks. In addition to traditional Linux mechanisms for inter-process communication Android provides the *Binder* [8] framework. Binder is an Android-specific IPC mechanism and remote method invocation system. Binder consists of a kernel-level driver and a userspace server. With Binder a process can call a routine in another process and pass the arguments between them. Binder has a very basic security model. It enables the identification of communication partners by delivering the PID and UID.

Android Permissions On Android services and APIs that have the potential to adversely impact the user experience or data on the device are protected with a

mandatory access control framework called *Permissions*. An application declares the permissions it needs in its AndroidManifest.xml[1] such as to access the contacts or send and receive SMS. At application install time those permissions are presented to the user who decides to grant all of them or deny the installation altogether. Permissions that are marked as *normal* such as wake-up on boot are hidden because they are not considered dangerous. The user however can expand the whole list of permissions if he wants to.

Memory Corruption Mitigation Memory corruption bugs such as buffer overflows are still a huge class of exploitable vulnerabilities. Since Android 2.3 the underlying Linux kernel implements `mmap_min_addr` to mitigate null pointer dereference privilege escalation attacks. `mmap_min_addr` specifies the minimum virtual address a process is allowed to mmap. Before, an attacker was able to map the first memory page, starting at address 0×0 into its process. A null pointer dereference in the kernel then would make the kernel access page zero which is filled with bytes under the control of the attacker. Also implemented since Android 2.3 is the eXecute Never (XN) bit to mark memory pages as non-executable. This prevents code execution on the stack and the heap. This makes it harder for an attacker to inject his own code. However an attacker can still use return oriented programming (ROP) to execute code from e.g. shared libraries. In Android 4.0 the first implementation of address space layout randomization (ASLR) was built into Android. ASLR is supposed to randomize the location of key memory areas within an address space to make it probabilistically hard for an attacker to gain control over a process. The Linux kernel for ARM supports ASLR since version 2.6.35. The Linux kernel is able to randomize the stack address and the brk memory area. The brk() system call is used to allocate the heap for a process. ASLR can be enabled in two levels by writing either a 1 (randomize stack start address) or a 2 (randomize stack and heap address) to `/proc/sys/kernel/randomize_va_space`. In Android 4.0 only the stack address and the location of shared libraries are randomized. This leaves an attacker plenty of possibilities to easily find gadgets for his ROP attack. In Android 4.1 Google finally added support for position independent executables (PIE) and a randomized linker to fully support ASLR. With PIE the location of the binary itself is randomized. Also introduced in Android 4.1 is a technique called read-only relocation (RELro) and immediate binding. To locate functions in a dynamically linked library, ELF uses the global offset table (GOT) to resolve the function. On the first call a function that is located in a shared library points to the procedure linkage table (PLT). Each entry in the PLT points to an entry in the GOT. On the first call the entry in the GOT points back to the PLT, where the linker is called to actually find the location of the desired function. The second time the GOT contains the resolved location. This is called lazy-binding and requires the GOT to be writable. An attacker can use this to let entries in the GOT point to his own code to gain control of the program flow.

[1] There are more than 110 permissions in Android. A full list is available at http://developer.android.com/reference/android/Manifest.permission.html.

RELro tells the linker to resolve dynamically linked functions at the beginning of the execution. The GOT is then made read-only. This way an attacker cannot overwrite it and cannot take control of the execution.

5 Android Security Enhancements

With Android 4.2 and the following minor releases Google introduced new security features in Android. We will present a small selection of these enhancements in the following paragraphs. The user now can choose to verify side-loaded applications prior to installation. This is also know as the on-device Bouncer. It scans for common malware and alerts the user if the application is considered harmful. So far the detection rates don't measure up with other commercial malware scanners [5]. With Android 4.2.2 Google introduced secure USB debugging. That means only authenticated host devices are allowed to connect via USB to the mobile device. To identify a host, adb generates an RSA key pair. The RSA key's fingerprint is displayed on the mobile device and the user can select to allow debugging for a single session or grant automatic access for all future sessions. This measure is only effective if the user has a screen lock protection enabled. Prior to Android 4.2 the optional `exported` attribute of a Content Provider defaulted to true which hurts the principle of least privilege. This lead to developers involuntarily making data accessible to other apps. With Android 4.2 the default behaviour is now "not exported".

SELinux on Android The SEAndroid project [15] is enabling the use of SELinux in Android. The separation guarantees limit the damage that can be done by flawed or malicious applications. SELinux allows OS services to run without root privileges. Albeit SELinux on Android is possible it is hard to configure and it slows down the device. Samsung Knox has been announced to actually roll-out SEAndroid on commercial devices.

6 Android Security Problems

According to F-Secure Response Labs 96 % of mobile malware that was detected in 2012 targets the Android OS [11]. In this chapter we want to shed light on the security weaknesses of Android that enabled such a vibrant market of malware. In short, Android has four major security problems: First, security updates are delayed or never deployed to the user's device. Second, OEMs weaken the security architecture of standard Android with their custom modifications. And third, the Android permission model is defective. Finally, the Google Play market poses a very low barrier to malware. We will now detail each of these problems.

Android Update Problem There are four parts of the system that can contain vulnerabilities: the base system containing the kernel and open source libraries, the stock

Android runtime including basic services and the Dalvik runtime, the Skin supplied by the OEM and the branding. The Android base system and runtime are published with full source by the AOSP. This code is the basis of all Android based smart phones. Any vulnerability found therein can potentially be used to subvert countless Android devices. In other terms, a vulnerability has a high *impact*. In practice, updates are very slow to reach the devices, with major updates taking more than 10 months [3]. Many vendors do not patch their devices at all, as the implementation of a patch seems too costly [4]. According to Google Inc.'s own numbers, the most recent version of Android is deployed to only 1.2 % of devices [2]. To remedy this problem, Google announced an industry partnership with many OEM pledging to update their devices for 18 months. This partnership is called the *Android Update Alliance*. However, there has been no mentioning of the alliance since 2012, and updates are still missing [3]. Bringing the updates to the devices is more involved however. Once the update reaches the OEMs, they incorporate it into their internal code repositories. For major updates, this includes porting their Skin forward. A faulty firmware update has very bad consequences for the OEM's reputation. Therefore the updated firmware is subject to the OEM's quality control. In summary, incorporating an update into a device firmware is therefore very costly to the OEM both temporal and financial. Cellular operators certify devices for correct behaviour. This is done to ensure that the device does not misbehave and therefore does not put the network and its users at risk. Updated firmwares need to be re-certified before they can be deployed. Depending on the operator this can take a substantial amount of time. For example re-certification at T-Mobile takes three to six months [12], other carriers opt out of the process and do not ship any updates at all.

Android Permission Model The Android permission model has been under criticism since Android was introduced. It has been extensively studied by researchers. Here we present the problems that stand out. Kelley et al. conducted a study and found that users are generally unable to understand and reason about the permission dialogues presented to them at application installation time [18]. In [16] Barrera et al. conducted an analysis of the Android permission model on a real-world data set of applications from the Android market. It showed that a small number of permissions are used very frequently and the rest is only used occasionally. It also shows the difficulty between having finer or coarser grained permissions. A finer grained model increases complexity and thus has usability impacts. The study also showed that not only users may have difficulties understanding a large set of permissions but also the developers as many over-requesting applications show. Felt et al. performed a study on how Android permissions are used by Apps. They found that in a set of 940 Apps about one-third are over-privileged, mostly due to the developers being confused about the Android permission system [17]. Another problem are combo permissions. Different applications from the same author can share permissions. That can be used to leak information. For example an application has access to the SMS database because it provides full text search for your SMS. Another app, say a game, from the same author has access to the Internet because it needs to load ads from an ad server.

Now through Android's IPC mechanism those two apps can talk to each other and essentially leak the user's SMS database into the Internet.

Insufficient Market Control Anybody can publish her applications to the official Android App market *Google Play* after paying a small fee. There are alternative App markets, e.g. the Amazon Appstore [7] and AndroidPit [9], but Google Play is the most important one because it is preinstalled on almost any Android device. Any App that is published via Google Play must adhere to the Google Play Developer Distribution Agreement (DDA) [13] and Google Play Developer Program Policies (DPP) [14]. However, Google Play does not check upfront if an uploaded App does adhere to DDA and DPP. Only when an App is suspected to violate DDA or DPP, it is being reviewed. If it is found to breach the agreements, it is suspended and the developer notified. If the App is found to contain malware, Google might even uninstall the App remotely. In 2012 Google introduced *Bouncer* [6]. Bouncer is a service that scans Apps on Google Play for known malware. It runs the Apps in an emulator and looks for suspicious behaviour. Unfortunately it didn't take long for researchers to show ways on how to circumvent Bouncer [1]. Malicious Apps have been found on Google Play repeatedly [10].

7 Lessons Learned

From our study of Android security problems we compile a set of lessons learned to educate future OP developers in avoiding these pitfalls.

Timely security updates are an absolute must for any secure system. This is especially important for open source systems where the code is public and bugs are easy to spot. For Smartphones an update system has to take all involved parties into account. We think that the key lies in clear abstractions and a modular system. That would enable the cellular operators to certify a device by looking on the radio stack alone.

Control platform diversity: The OS designer should enforce that third party modifications to the OS do not introduce security breaches by design. He should enforce contracts on security critical points in the system that third party code has to follow. For example Google should enforce that any device running Android must only contain code that enforces the Android permission system.

Ensure lock screen locks screen under all circumstances: Ensure that no third party can mess with the lockscreen.

Design permission system with user and developer in mind: A permission system should be designed such that the permissions it implements are understood by both the developer to avoid over-privileged Apps and the user, so that she can make an educated decision when granting permissions. Granting all permissions at installation time is problematic. Users ofter grant permissions just to be able to install an App. Also, it does not allow for fine-grained permissions. Maybe a better solution would be to ask the user to grant permissions on demand.

Ensure that the App market does not distribute malware: The App market is the most important distribution place for Apps. People trust in the App markets, and have no chance to determine the quality of an App by themselves. Aside from having a mandatory admission process, an App market should also scan for repackaged Apps.

8 Conclusion

In this work we investigated the security of the Android mobile OS. We described the Android security measures, and its problems. We derived a set of lessons learned that will help future mobile OS designers to avoid pitfalls.

Acknowledgments This work was partially supported by the EU FP7/2007-2013 (FP7-ICT-2011.1.4 Trustworthy ICT), under grant agreement no. 317888 (project NEMESYS).

References

1. Adventures in BouncerLand (2012) Failures of automated Malware detection within mobile application markets. http://media.blackhat.com/bh-us-12/Briefings/Percoco/BH_US_12_Percoco_Adventures_in_Bouncerland_WP.pdf, July 2012
2. Android Dashboard (2012) https://developer.android.com/about/dashboards/index.html, Dec 2012
3. Arstechnica (2012) The checkered, slow history of Android handset updates. http://arstechnica.com/gadgets/2012/12/the-checkered-slow-history-of-android-handset-updates/, Dec 2012
4. Arstechnica (2012) What happened to the Android Update Alliance? http://arstechnica.com/gadgets/2012/06/what-happened-to-the-android-update-alliance/, June 2012
5. An evaluation of the application verification service in android 4.2. http://www.cs.ncsu.edu/faculty/jiang/appverify/, Dec 2012
6. Google Mobile Blog (2012) Android and security. http://googlemobile.blogspot.de/2012/02/android-and-security.html, Feb 2012
7. Amazon Appstore (2013) http://www.amazon.com/mobile-apps/b/ref=sa_menu_adr_app?ie=UTF8&node=2350149011, April 2013
8. Android Developer Documentation (2013) Binder. http://developer.android.com/reference/android/os/Binder.html, Jan 2013
9. AndroidPit (2013) http://www.androidpit.com/, Apr 2013
10. Arstechnica (2013) More "BadNews" for Android: New malicious apps found in Google Play. http://arstechnica.com/security/2013/04/more-badnews-for-android-new-malicious-apps-found-in-google-play/, Apr 2013
11. F-Secure Mobile Threat Report Q4 2012. http://www.f-secure.com/static/doc/labs\global/Research/Mobile20Threat20Report20Q4202012.pdf, March 2013
12. Gizmodo (2013) Why Android Updates Are So Slow. http://gizmodo.com/5987508/why-android-updates-are-so-slow, March 2013
13. Google Play Developer Distribution Agreement. http://www.android.com/us/developer-distribution-agreement.html, Apr 2013
14. Google Play Developer Program Policies. http://www.android.com/us/developer-content-policy.html, Apr 2013
15. Seandroid wiki (2013). http://selinuxproject.org/page/SEAndroid, Apr 2013

16. Barrera, D, Kayacik HG, van Oorschot PC, Somayaji A (2010) A methodology for empirical analysis of permission-based security models and its application to android. In: Proceedings of the 17th ACM conference on computer and communications security, CCS '10, ACM, New York, NY, USA, pp. 73–84. http://doi.acm.org/10.1145/1866307.1866317
17. Felt AP, Chin E, Hanna S, Song D, Wagner D (2011) Android permissions demystified. In: Proceedings of the 18th ACM conference on Computer and communications security, CCS '11, ACM, New York, NY, USA, pp. 627–638. http://doi.acm.org/10.1145/2046707.2046779
18. Kelley P, Consolvo S, Lorrie C, Jung J, Sadeh N, Wetherall D (2012) An conundrum of permissions: installing applications on an android smartphone. Workshop on Usable, Security
19. Symantec (2013) Internet security threat report. Technical report, Apr 2013. http://www.symantec.com/content/en/us/enterprise/other_resources/b-istr_main_report_v18_2012_21291018.en-us.pdf

Mobile Network Threat Analysis and MNO Positioning

George Lyberopoulos, Helen Theodoropoulou and Konstantinos Filis

Abstract The dramatic increase of smart mobile devices and applications, the advent of Android OS, the increased number of wireless radios (incl. NFC) the support and the low awareness about security and privacy risks on the one hand, and the flatter, IP-based network architecture, the introduction of new radio technologies (femtocells, WiFi, LTE) and applications (M2M, NFC) on the other, have changed the mobile threats landscape and will change the way security will be dealt in the coming years. Mobile Network Operators (MNOs) have started to investigate the possibility to introduce additional measures to secure their networks, providing thus a defense before security threats materialize.

1 Introduction

The wide adoption of smart mobile devices (smartphones, tablets) encompassing personal data such as, contacts' list, photos, notes, financial data, credentials for online banking, while offering always-on capability to social networks, e-mail accounts and possibly access to corporate networks, has augmented the interest of cyber-criminals, not only due to possible financial gains, but because these devices could be utilized as stepping stones for launching attacks towards the mobile core network and other connected networks or for industrial espionage. In addition,as mobile networks become

G. Lyberopoulos · H. Theodoropoulou · K. Filis (✉)
COSMOTE-Mobile Telecommunications SA, Ikarou 1 and Ag, 19002Louka, Attica, Greece
e-mail: glimperop@cosmote.gr

H. Theodoropoulou
e-mail: etheodorop@cosmote.gr

K. Filis
e-mail: cfilis@cosmote.gr

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_41,

central part of our daily lives, they comprise attractive, high-profile, targets for hacktivists, whose aim is to promote a political or social agenda through disruption.

The support of multiple communication technologies such as Bluetooth, Wi-Fi, 2G, 3G, Long Term Evolution (LTE), along with the user's capability to install applications from "untrusted" sources and the extensive use of outdated operating system versions—esp. for Android devices [17]—have increased the vulnerability of smart devices by exposing them to heterogeneous attack vectors [18]. In addition, the introduction of new radio access technologies, such as femtocells, Mobile Network Operator (MNO)-operated WiFi, and LTE (4G), the transition to flatter and more open network IP-based architectures, the upcoming M2M and NFC applications and the exponential growth traffic, introduce additional vulnerabilities for both the mobile devices/users and the core network.

It is envisaged that security attacks will become more aggressive both in terms of frequency and severity. As such, MNOs, to prevent potential mobile cyber-attacks and protect its brand name, are investigating the possibility to introduce additional measures to secure their networks, providing thus a defense before security threats materialize. Currently, the research interest is focusing on the specification and development of an infrastructure based on honeypots being capable of collecting and analyzing attack traces coming from mobile devices, in order to understand the attack strategies and build appropriate countermeasures [15, 16, 20].

The material included in this paper is organized as follows: Sect. 2 sheds some light on the mobile telecommunications threats landscape. In Sect. 3 we present the currently available security techniques for the 3G and LTE mobile networks and for the femtocells. In Sect. 4 we elaborate on the MNO's role/strategy in addressing the emerging security threats, while in Sect. 5 we draw some concluding remarks.

2 Mobile Environment Threats Landscape

The proliferation of powerful smart devices, the dramatic increase in the number of applications (from "trusted" and "untrusted" sources), the advent of Android OS—which is more susceptible to malware due to its openness-, the increased number of wireless radios (incl. NFC) and the low awareness of users about security and privacy risks on the one hand, and the flatter, IP-based network architecture, the introduction of new radio technologies (femtocells, WiFi, LTE) and applications (M2M, NFC) on the other, have changed the mobile threats landscape and will change the way security will be dealt in the coming years.

Figure 1 illustrates the security threats landscape in a mobile telecommunications environment. Obviously, mobile devices play a crucial role since hackers may not only benefit from the information stored in the terminal, but because they may be utilized as enablers/facilitators to launch attacks towards the mobile core networks and/or other external networks. The most common method for spreading malware to

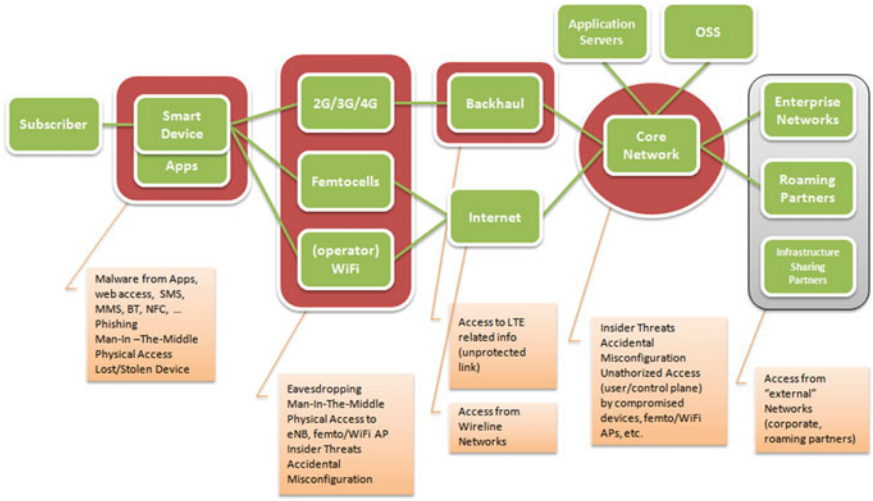

Fig. 1 Threats landscape in a mobile telecommunications environment

mobile devices is through the installation of an application (by the user's full consent), via SMS, MMS, Bluetooth, e-mail, the Internet, trusted and untrusted markets, etc.

By granting permission to collect and transmit information from the device, the hacker, may initiate voice calls and/or SMSs to premium numbers (that cost extra money), send e-mails and/or block incoming calls/SMSs, record conversations and send them to 3rd parties, steal and send personal info to 3rd parties, monitor the phone "status" (off-hook, ringing), take over control of the smartphone, turn the phone into mobile botnets so others can execute commands remotely, initiate attacks to mobile core/corporate network(s), (such as DDoS), reduce smartphone utility e.g., battery discharging, unusable smartphone (repeated reboots).

As far as the security threats towards the mobile core network are concerned (see Fig.1), these may originate from: (a) Compromised smart mobile devices, (b) The access network, (c) The backhaul network, (d) The core network and (e) External or 3rd party networks such as, the Internet, corporate networks, roaming partners' networks, other connected PLMNs, shared RAN, non-3GPP access network, external transport network, etc.

Note: Access to access/backhaul/core network necessitates physical access to the respective network nodes (by a "malicious" employee or a 3rd party) or the use of special equipment.

The security issues/challenges in 3G mobile core networks have been extensively discussed in the literature [7–10]. However, the evolution of the mobile networks towards the provision of higher Quality of Experience (QoE) to the end-users, as well as the simplification of the network architecture, has raised new security concerns for MNOs. More specifically:

- The proliferation of femtocells has introduced new points of attacks, including the air-interface, the FAP itself (which may reside at untrusted locations) and the backhaul [11, 12]. Third-party attacks may include man-in-the-middle (MITM), traffic snooping/redirection, fake base station attacks, authentication snooping, service disruption, and billing fraud.
- The incorporation of non-3GPP WiFi(s) owned by the MNO may necessitate access to LTE core network elements depending on the level of integration. As such, as in the case of femtocells, for a hacker it's easy to gain physical access to a WiFi access point that is now part of the MNO infrastructure.
- The introduction of the LTE (due to the new interfaces—such as X2, the Diameter protocol, the flatter architectures as well as the application related control plane traffic), is expected to have a tremendous impact on the signaling traffic that the network will have to cope with which may be leveraged for malicious activity.
- The transition of mobile networks to IP brings additional security threats, such as DoS and DDoS attacks, ping floods, SYN floods, replay attacks, DNS hijacking, IP port scanning [6], which may result in the interception of subscriber data, limit subscriber access (causing congestion), and/or compromise the overall network security of the network, since some of the core elements' functionality may be lost.[1]
- Apart from the upcoming VoLTE and the mandated introduction of IMS, it is envisaged that the trend toward virtualization and sw-defined networks will create new vulnerability sources, as both user and control plane traffic becomes more distributed across network and has to cross untrusted portions of it [6].

3 MNO Security Architecture

A threat and risk analysis for mobile communication networks in a qualitative way—see estimation of the likelihood of attacks, overall vulnerability of the assets, impact of successful attacks on the network—is presented in [14]. According to this study, the following threat categories can be identified:

(1) Flooding an interface (radio, backhaul), (2) Crashing a network element via a protocol or application flaw, (3) Eavesdropping (radio interface, backhaul, control plane, user plane), (4) Unauthorized data access to sensitive data on a network element via leakage, (5) Traffic modification (radio interface, backhaul, c-plane, u-plane), (6) Data modification on a network element, (7) Compromise of a network element via a protocol or application implementation flaw, (8) Compromise of a network element via management interface, (9) Malicious insider, (10) Theft of service.

MNO's security techniques include: advanced firewall and intrusion prevention systems, the addition of IPsec termination capabilities on platforms, and standardized features of network security architecture, including advanced message and entity

[1] In Japan, NTT DoCoMo experienced a signaling flood that disrupted network access in Jan/2012, caused by a VoIP OTT application running on Android phones [13].

authentication for both user and network, by using strong key-based cryptography, advanced encryption methods, mobile equipment identification, security gateways, ciphering mechanisms for signalling messages, location verification, prevention of unauthorized access, etc. However, the protection of the mobile core network from an attack coming from a user device still remains a challenge to be investigated and addressed, so that MNOs can protect their networks and provide their subscribers with a safe environment and advanced quality of service.

3.1 3G Security Architecture

Security protection in 3G-networks requires the consideration of several aspects and issues, such as the wireless access, the end-user mobility, the particular security threats, the type of information to be protected, and the complexity of the network architecture. The radio transmission is by nature more susceptible to eavesdropping than wired transmission. The user mobility and the universal network access certainly imply security treats. The different types of data, such as user data, charging and billing data, customer information data, and network management data, which are conveyed or are resident in mobile networks, require different types and levels of protection. Furthermore, the complex network topologies and the heterogeneity of the involved technologies increase the dependability challenge. Figure 2 presents an overview of the complete 3G security architecture [2].

There are 5 different sets of features that are part of the architecture:

(1) Network access security: Provides secure access to 3G services and protects against attacks on the radio interface link. (2) Network domain security: Allows nodes in the operator's network to securely exchange signaling data and protects

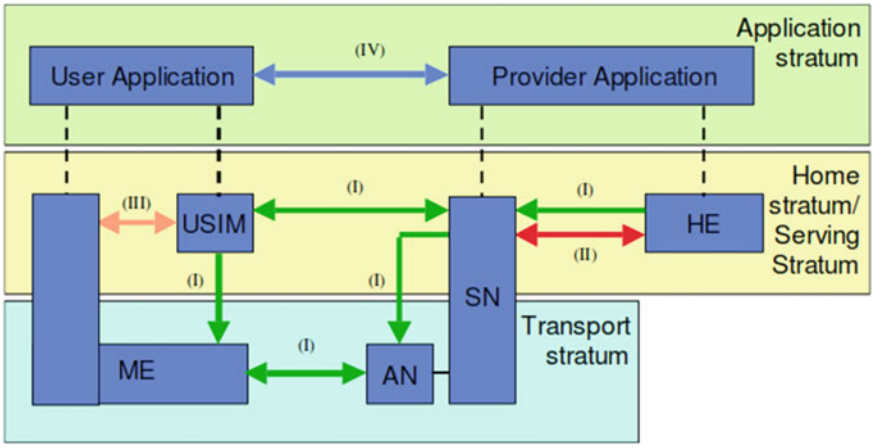

Fig. 2 Overview of the 3G network security architecture

against attacks on the wireline network. (3) User domain security: Secures access to mobile stations. (4) Application domain security: Enables applications in the user and in the provider domain to securely exchange messages and (5) Visibility and configurability of security: Allows the user to get information about the security features in operation and whether a service provision depends on the activation or not of a security feature.

Network access security features can be further classified into:

(1) User authentication: The property that the network that provides the service (serving network) corroborates the identity of the user. (2) Network authentication: The property that the user corroborates that he is connected to a serving network that is authorized by the user's home network. (3) Cipher algorithm agreement: The property that the terminal and the serving network can securely negotiate the algorithm that they shall use subsequently. (4) Cipher key agreement: The property that the terminal and the serving network agree on a cipher key that they may use subsequently. (5) Confidentiality of user data: The property that user data cannot be overheard on the radio interface. (6) Confidentiality of signaling data: The property that signaling data cannot be overheard on the radio interface. (7) Integrity algorithm agreement: The property that the terminal and the serving network can securely negotiate the integrity algorithm that they shall use subsequently. (8) Integrity key agreement: The property that the terminal and the serving network agree on an integrity key they may use subsequently. (9) Data integrity and origin authentication of signaling data: The property that the receiving entity (terminal or serving network) is able to verify that signaling and/or its origin has not been modified in an unauthorized way.

3.2 LTE Security Architecture

The LTE/SAE (System Architecture Evolution) network consists of only two nodes: (1) The MME/S-GW (Mobility Management Entity/SAE gateway), which is a multi-standard access system behaving as the anchor point for the mobility between different access systems, and (2) The eNB, which gathers all the purely radio-oriented functionalities. Most of the security requirements for 3G networks hold also for the LTE, so as at least the same level of security (as in 3G) shall be guaranteed (Fig. 3).

The main changes that have been adopted to fulfill the required level of LTE security are summarized below:

- A new hierarchical key system has been introduced in which keys can be changed for different purposes.
- The LTE security functions for the Non-Access Stratum (NAS) and the Access Stratum (AS) have been separated. The NAS functions are responsible for the communications between the core network and the mobile terminal, while the AS functions encompass the communications between the network edges, i.e. the eNB and the terminal.
- The concept of forward security has been introduced for LTE.

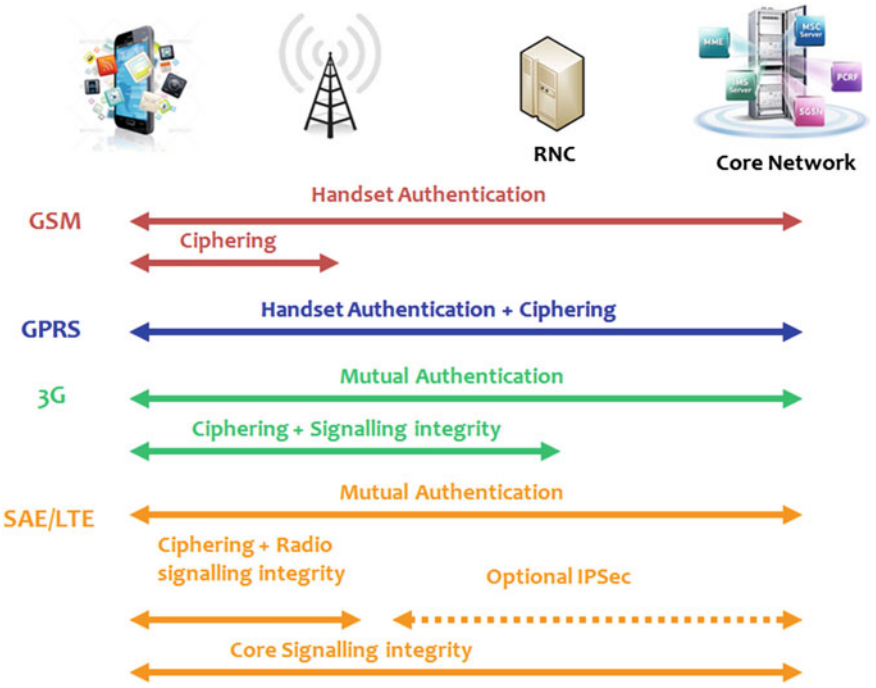

Fig. 3 Evolving security architecture towards LTE [19]

- LTE security functions have been introduced between the existing 3G network and the LTE.

In addition, since in LTE the mandated encryption from the mobile device terminates at the eNB, MNOs should secure the IP-based control/user plane transport to the core network using IPsec; although not mandatory according to the standards.

3.3 Femtocell Security Architecture

Femtocell Access Points (FAPs) are close-range, limited-capacity base stations that utilize residential broadband connections to connect to carrier networks [4]. The use of such distributed base station architecture, although it improves reception and allows the operators to deliver fast, seamless, high-bandwidth cellular coverage indoors, introduces new security concerns classified into three main categories: (1) Device and network authentication, (2) Data privacy and (3) Data integrity.

The security mechanisms designed to fulfill these concerns are the following [5]:

(1) FAP Physical Security, (2) FAP and Core Network mutual authentication and IPSec tunnel establishment, (3) Location Verification, (4) Access Control,

(5) Protection of traffic between FMS and FAP and (6) Measures for Clock Protection.

4 MNO Positioning

The mobile industry (equipment manufacturers, security systems vendors, etc.) is oriented towards a strict compliance approach aligned with the 3GPP specifications for securing networks' operation. In order for the MNOs to cope with the emerging security threats, they should develop a holistic, proactive, defensive and affordable security strategy (incl. policies, security processes, security risk management, business continuity, etc.), so as to secure both their networks and their subscribers. It is obvious that this objective is a real challenge, since the mobile network infrastructure is massive and extremely complex with multiple entities coordinating together. In case of a successful attack (e.g. DDoS attack) affecting part or the whole mobile network, the impact on the operator business will be negative, and therefore such incidents are highly undesirable.

Even in case the attack is directed to the user terminal, the impact could be significant if this happens on a large scale. In such cases, the customers' experience will be degraded and even if the "problem" originated by the customer itself (phone jailbreaking/rooting, installation of applications from untrusted sources, careless acceptance of application permissions), it is envisaged that the customers will blame the operator; especially if they have purchased their smartphones from the particular MNO. In the long term, if fraud via mobile malware gets out of control, it may lead to a trust loss which may slow down the growth of the overall mobile business. Towards this direction, the MNOs should:

- Apply all the latest security features (upon availability) to protect their networks end-to-end from possible malicious and/or accidental attacks.
- Facilitate the public awareness regarding the existence of malware and their impact as well as to inform the public on how they could be protected.
- Participate in R&D security-related activities, so as to be capable of setting the requirements from their own perspective, being informed on the latest developments on the security aspects and/or exploiting security research results.
- Closely cooperate with infrastructure and security vendors, security analysts, etc. to specify/develop new security features and toolsets.
- Establish a dedicated team to deal with network and terminal security, in terms of: identification of possible security gaps, conduction of specific experiments to reveal network vulnerabilities, monitoring how malicious attacks are evolving with time, disseminating the findings especially to those responsible for protecting the availability and integrity of the mobile network, coordination/decision making for actions required at the event of a successful DDoS signalling attack (e.g., load balancing, policy enforcement, validation of legitimate signalling traffic to minimize disruption, etc. [6]).

5 Conclusions

While current security techniques provide an adequate level of protection, MNOs need to take further actions to protect their networks from emerging threats, which may be caused either by malicious activity explicitly directed at the mobile network, or accidentally occurred. On the other hand, compromised mobile devices may constitute a quite substantial threat for the affected user, while when acting as mobile botnets, can easily endanger a whole mobile network operation. Therefore, MNOs should focus on building an effective proactive security strategy to protect both their network and subscribers, while in case of an attack, the MNOs should be prepared to respond immediately to defend their reputation and ensure the viability of mobile business.

Acknowledgments The current study is part of the Project NEMESYS (Enhanced Network Security for Seamless Service Provisioning in the Smart Mobile Ecosystem) which has received funding from the European Union Seventh Framework Programme (FP7) under grant agreement 317888. Disclaimer: The views expressed in this article are those of the authors and do not necessarily represent the views of the company.

References

1. http://www.lucent.com/enrich/v1i12007/article_c4a4.html
2. ETSI TS 133 102 V11.5.0 (2013–02) Digital cellular telecommunications system (Phase 2+); UMTS; LTE; 3G security; Security architecture (3GPP TS 33.102 v11.5.0 Rel. 11)
3. LTE; E-UTRA; E-UTRAN; Overall description; Stage 2 (3GPP TS 36.300 v11.4.0 Rel. 11).
4. Chen J, Wong M (2012) Security implications and considerations for femtocells. J Cyber Security Mobility 21(35)
5. ETSI TS 133 320 V11.6.0 (2012–11) UMTS; LTE; Security of Home Node B (HNB)/Home evolved Node B (HeNB) (3GPP TS 33.320 v11.6.0 Rel. 11)
6. Monica Paolini. Wireless Security in LTE Networks
7. Peng X, Wen Y, Zhao H (2011) Security issues and solutions in 3G core network. J Networks 6(5):823–830
8. Ahmed F et al. (2011) A data mining framework for securing 3g core network from GTP fuzzing attacks. Proceedings of the 7th International Conference on Information Systems Security
9. Checkpoint White Paper, Next Generation Security for 3G and 4G LTE Networks
10. Peng X et al. (2010) GTP security in 3G core network. 2010 2nd international conference on networks security, wireless communications and trusted computing
11. Bilogrevic I, Jadliwala M, Hubaux J.-P (2010) Security issues in next generation mobile networks: LTE and femtocells
12. Security of H(e)NB. Technical Report TR 33.820 v8.3.0, 3GPP, Dec. 2009
13. http://www.reuters.com/article/2012/01/27/us-docomo-idUSTRE80Q1YU20120127
14. ASMONIA Project: D5.1 - Threat and Risk Analysis for Mobile Communication Networks and Mobile Terminals, http://www.asmonia.de/deliverables/D5.1_II_ThreatAnd RiskAnalysisMobileCommunicationNetworksAndTerminals.pdf
15. NEMESYS Project, http://www.nemesys-project.eu/nemesys/
16. C. Dimitriadis Improving Mobile Core Network Security with Honeynets
17. Android Dashboard. https://developer.android.com/about/dashboards/index.html, Dec 2012
18. La Polla M et al. A survey on security for mobile devices

19. Cisco LTE Security Architecture—Session 2
20. Gelenbe E, Gorbil G, Tzovaras D, Liebergeld S, Garcia D, Baltatu M, Lyberopoulos G (2013) NEMESYS: Enhanced network security for seamless service provisioning in the smart mobile ecosystem. Proceedings of 28th International Symposium on computer and information sciences (ISCIS13), Paris, Oct 2013

Mobile Network Anomaly Detection and Mitigation: The NEMESYS Approach

Omer H. Abdelrahman, Erol Gelenbe, Gökçe Görbil and Boris Oklander

Abstract Mobile malware and mobile network attacks are becoming a significant threat that accompanies the increasing popularity of smart phones and tablets. Thus in this paper we present our research vision that aims to develop a network-based security solution combining analytical modelling, simulation and learning, together with billing and control-plane data, to detect anomalies and attacks, and eliminate or mitigate their effects, as part of the EU FP7 NEMESYS project. These ideas are supplemented with a careful review of the state-of-the-art regarding anomaly detection techniques that mobile network operators may use to protect their infrastructure and secure users against malware.

1 Introduction

Mobile malware is emerging as a significant threat due to the increasing popularity of smart phones and tablets, which now run fully-fledged operating systems (OSs) on powerful hardware and feature multiple interfaces and sensors. Personal computers (PCs) are no longer the dominant computing platform, and indeed the global shipments of smart phones alone have exceeded those of desktop computers since 2011 [7]. Further, with the accelerated adoption of 4G technologies including WiMAX and LTE, cellular devices will become the primary means of broadband Internet access for many users. In fact, while 4G capable devices represent only 0.9 % of all global mobile connections observed in the Internet during 2012, they already account for 14 % of the mobile data traffic [8]. As more and more people move from PCs to handheld devices, cyber criminals are naturally shifting their attention to the mobile environment, and this trend is fuelled by the availability of off-the-shelf

O. H. Abdelrahman (✉) · E. Gelenbe · G. Görbil · B. Oklander
Department of Electrical and Electronic Engineering, Imperial College,
London SW7 2BT, UK
e-mail: o.abd06@imperial.ac.uk

E. Gelenbe and R. Lent (eds.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7_42,

malware creation tools [5] as well as the proliferation of mobile application (shortly known as app) marketplaces, enabling the distribution of malicious apps to potentially millions of users [31]. Such mobile malware can attempt to snoop and steal saleable information, generate revenue by calling premium rate numbers, or perform other malicious activities.

Despite this growing challenge, operators continue to be reactive rather than proactive towards these security threats [2], and investments in detection and mitigation techniques specific to mobile networks are only implemented when a problem occurs. In this position paper regarding certain research activities of the EU FP7 project NEMESYS [21], we describe our approach to *the research and development of anomaly detection techniques so that mobile network operators may protect their own infrastructure and defend users against malware*. The techniques that operators may develop and deploy on their networks can constitute value-added services for communities of users, in the same way that banks use profiling to reduce credit card fraud. If deployed at the network rather than mobile devices, such services will also spare mobile device batteries and bandwidth (both computational and communication). In fact, most users are not aware of the growing security risks with mobile devices [3], but it is in the interest of operators to ensure that users are well protected since large scale malware infections pose a significant threat to the availability and security of cellular networks. Network level analysis also provides a broad view of malicious activities within an operator's network, is not vulnerable to exploits that allow malware to circumvent client-side security, and can be modified easily without requiring users to install patches and updates.

In the sequel, we first present our vision about how to address this field of research in the context of NEMESYS for a network-based security solution which combines modelling and learning, and uses network measurements as well as control-plane and billing data to detect and evaluate anomalies and attacks. Then we review the recent literature on attacks targeting services and users of mobile networks, and outline the merits and limitations of existing network- and cloud-based anomaly detection techniques. Finally we draw some conclusions.

2 The NEMESYS Model-based Approach

The research we are conducting with regard to anomaly detection and mitigation within the NEMESYS project uses a model-based approach that involves representing how the communication system functions at the level of each mobile connection. The model-based approach is motivated by several factors. *First*, the number of mobile users that we need to monitor and deal with in real time is very large. Thus a clear and understandable uniform approach is needed to deal with each individual mobile call, emphasising the similarities and common parameters. Anomalies can then be detected via deviations from normal parameters or behaviours. *Second*, the computational tools that are being developed for anomaly detection and mitigation need to be based on sound principles; mathematical models allow us to evaluate and validate such algorithms based on clear underlying model assumptions, even though

the use of these models in various practical situations will include conditions when some of the model assumptions are not satisfied. Thus mathematical models will need to be tested and validated through simulation and practical measurements. *Third*, due to the sheer size of the systems we need to deal with, the mathematical models will have to be decomposable, both in terms of obtaining analytical and numerical solutions, e.g. in terms of product forms [19], and in terms of distributed processing for reasons of scalability [1]. Again, the mathematical and decomposable structure also provides a handle for decomposing and distributing the computational tasks.

The focus of model construction is on identifying and modelling the individual steps that a mobile user makes regarding:

- Call establishment, including the connection to base stations, access points, and call management,
- Monitoring and billing, and the interactions between the mobile operator's resources and the network for monitoring and billing,
- Accesses that the call may make to sensitive resources such as web sites for privileged information interrogation,
- Call processing or service steps that may require that the mobile user identity itself to the network or external resources, or provide other sensitive information (e.g. personal addresses) at certain operational steps,
- The access to web sites that are used for purchasing and billing.

Indeed, in order to develop detection capabilities of a practical value it is vital to formulate a unified analytical framework which explicitly describes the main *internal resources* of the network architecture, including both the internal aspects regarding base station, access points and call management and billing, and the *sensitive external resources* that the mobile user may access during its call. Since our approach will have to be effective in situations where hundreds of thousands of mobile users may be active in a given network simultaneously, we need to address both:

- The case where only a small percentage of mobile users come under attack at a given time, but these attacks are nevertheless of high value to the attacker so that we must be able to detect relatively rare events in a very large ensemble, a little like detecting a small number of hidden explosive devices in a very large and potentially dangerous terrain, also
- Situations where attacks affect the signaling system and are significantly disturbing a large fraction of the ongoing mobile connections.

In all cases we will need to deal with real-time detection, mitigation and possibly attack elimination, as well as data collection for deferred ulterior analysis.

2.1 Modelling

Research in communication systems has a solid background of modelling methodologies such as stochastic processes, queueing systems, graph models, etc. These methods are routinely and successfully utilised to describe communication systems

and to analyse and improve their performance, but they are rarely used for security. Our research plan for anomaly detection incorporates modelling of the wireless communication network at different levels of abstraction to properly represent the components and the processes that are prone to anomalous behaviour.

The natural choice for this approach is multi-class queueing and diffusion models [10, 17] and related methods. Such models provide estimates of both averages and variances of the times that it would take to undertake signaling or call processing functions, as well as of access times to web sites and other services, in the presence of a large population of mobile users in the network, once the average service times and task sequences are known. For a given (small) subset of users which one wishes to monitor, if the estimated average quantities for a population scaled up to the current observed numbers in the network, for the same user set, deviates significantly from the current measured values in the network, then one can infer some level of anomaly. The estimates for the scaled up population can be calculated from the queueing models in a very fast and straightforward manner, leading to a useful modelling based anomaly detector. This performance based approach can also provide billing estimates based on the utilisation and durations related to internal and external resources, so that the queueing model results can map into billing information: once again, deviations from the expected values (even in the presence of large traffic loads) can be estimated by comparing model predictions with measured observations.

In addition to the above approach, some of the analysis may require more effort because contrary to standard approaches that obtain the steady-state behaviour, the detection of anomalies may require detection of change that is time dependent and thus requires transient or rare event analysis. In order to address the rise of the resulting computational complexity, we also plan to utilise the learning capabilities of the Random Neural Network (RNN) [11, 12, 14–16, 20], which has previously been applied to a variety of complex problems. The RNN's mathematical analogy to a queueing and communication network will ease the design of learning techniques that have to mimic the behaviour of mobile calls in a wireless network with its different resources and customers.

2.2 Simulation Tools

Our analytical models and anomaly detection algorithms will be augmented and validated with simulation tools. As an initial step, we are developing realistic simulations of UMTS and LTE networks using the OPNET simulator in order to extract data regarding control-plane events that take place during normal mobile communications. For this purpose, we are currently modelling a small-scale mobile network and all control-plane protocols in the packet-switched domain, including mobility, session and radio resource management. Characteristics of these control events will be used to drive the development of our analytical models. We will later increase the scale of our simulations to validate our mathematical results. For performance reasons in this stage, instead of simulating communications and events of all mobile

users, we will identify the generic characteristics of a large number of users and use these to generate background traffic on the network while explicitly simulating a smaller set of users, among which only a few may demonstrate anomalous behaviour. Another set of simulations will include billing system components to monitor monetary use of internal and external network resources, and based on data traces and parameters obtained from real mobile networks, these simulations will be used to generate synthetic data for the learning methods we plan to apply and to test the performance of our real-time and offline anomaly detection methods. Finally, we will employ simulation as a tool for the integration and validation of other system components which are developed by NEMESYS partners, such as the attack correlation and visualisation & analysis modules [34].

3 Prior Work on Network Threats and Mitigation

Mobile networks are vulnerable to a form of denial-of-service (DoS) attack known as *signaling attacks* in which the control plane is overloaded through low-rate traffic patterns that exploit vulnerabilities in the signaling procedures involved in paging [41], transport of SMS [9] and radio resource control [29] (see [39] for a review on the subject). In principle, signaling attacks can be carried out either by compromising a large number of mobile devices as in the case of distributed DoS attacks on wired networks [18] or from outside the network (e.g. the Internet) by targeting a hit list of mobile devices through carefully timed traffic bursts. In order to orchestrate such attacks, active probing can be used to infer the radio resource allocation policies of operational networks [4, 36] as well as to identify a sufficient number of IP addresses in a particular location [37]. Moreover, a study [37, 43] of 180 cellular carriers around the world revealed that 51 % of them allow mobile devices to be probed from the Internet. Despite their feasibility, signaling attacks are yet to be observed in practice, which is likely due to the lack of financial incentives for cyber criminals who would rather have the infrastructure functional in order to launch profitable attacks. A related threat known as *signaling storms* occur often as a result of poorly designed popular mobile apps that repeatedly establish and tear down data connections, generating huge levels of signaling traffic capable of crashing a mobile network. Thus signaling storms have the same effect as a DoS attack [13, 18], but without the malicious intent, and they are becoming a serious threat to the availability and security of cellular networks. For example, a Japanese mobile operator suffered a major outage in 2012 [38], which was attributed to an Android VoIP app that constantly polls the network even when users are inactive. Moreover, according to a recent survey of mobile carriers [2], many of them have reported outages or performance issues caused by non-malicious but misbehaving apps, yet the majority of those affected followed a reactive approach to identify and mitigate the problem. Note that unlike flash crowds, which normally happen and last for a short period of time coinciding with special occasions such as New Year's Eve, signaling storms are unpredictable and tend to persist until the underlying problem is identified and corrected. This

has prompted the mobile industry to promote best practices for developing network-friendly apps [22, 27]. However, the threat posed by large scale mobile botnets cannot be eliminated in this manner, as botmasters care more about increasing their revenue and stealthiness than the impact that their activities have on the resources of mobile networks.

Countermeasures signaling problems have a limited impact on the data plane and thus are difficult to detect using traditional intrusion detection systems which are effective against flooding type attacks. For Internet-based attacks, a change detection algorithm using the cumulative sum method has been proposed in [29], where the signaling rate of each remote host is monitored and an alarm is triggered if this rate exceeds a fixed threshold. The use of a single threshold for all users, however, presents a trade-off between false positives and detection time, which can be difficult to optimise given the diversity of users' behaviour and consumption. A supervised learning approach is used in [23] to detect mobile-initiated attacks whereby transmissions that trigger a radio access bearer setup procedure are monitored, and various features are extracted relating to destination IP and port numbers, packet size, variance of inter-arrival time, and response-request ratio. One problem with supervised learning techniques is that both normal and malicious behaviours need to be defined in advance, rendering them ineffective against new and sophisticated types of attacks. Detection of SMS flooding attacks is considered in [28], where low reply rate is used as the main indicator of malicious activities, which is likely to misclassify SMS accounts used for machine-to-machine (M2M) communications, such as asset tracking and smart grid meters [33].

3.1 Attacks Against Mobile Users

A recent report by Kaspersky [32] revealed that the most frequently detected malware threats affecting Android OS are SMS trojans, adwares and root exploits. Mobile botnets are also emerging as a serious threat because of their dynamic nature, i.e. they could be used to execute any action at the command of a botmaster. In the following we summarise the various approaches that have been proposed to enable mobile operators to detect attacks against users.

Network level analysis has been explored in a number of recent studies focusing on three aspects:

- Domain Name System (DNS): Since malware typically uses DNS to retrieve IP addresses of servers, detecting and blacklisting suspicious domains can be a first step towards limiting the impact of malware [25, 30]. However, detection should not be based solely on historical data (i.e. known malicious domains), but also on behavioural characteristics such as host changes and growth patterns which may differentiate normal and abnormal traffic.
- Call Charging Records (CDR): One of the key characteristics of mobile communications pertains to the fact that the whole extent of exchanged traffic load is

continuously monitored for billing and accounting purposes. Hence, it is expected that many malicious activities will have an evident impact on the CDR of the parties involved. In [33], communication patterns of SMS spammers are compared to those of legitimate mobile users and M2M connected appliances, showing evidence of spammer mobility, voice and data traffic resembling the behaviour of normal users, as well as similarities between spammers and M2M communication profiles. Fuzzy-logic is used in [42] to detect SMS spamming botnets by exploiting differences in usage profiles, while in [44] SMS anomalies are detected through building normal social behaviour profiles for users, but the learning technique fails to identify transient accounts used only for malicious purposes. Markov clustering of international voice calls [26] indicates that different fraud activities, such as those carried by malicious apps with automated dialler or via social engineering tactics, exhibit distinct characteristics.

- Content matching: Uncommon header flags and syntactic matches in HTTP messages can be used as indicators of data exfiltration attempts [25], but this approach is not effective when end-to-end encryption is used, as it relies on extracting information from plain-text transmissions.

Cloud-based detection offers a trade-off between network level analysis and on-device security: the former imposes zero-load on the device, but limits its scope to cellular data, while the latter is able to utilise internal mobile events for detection but is resource hungry. There are two main approaches, both of which offload intensive security analysis and computations to the cloud. The first uses a thin mobile client to extract relevant features from the device [40] including free memory, user activity, running processes, CPU usage and sent SMS count, which are then sent to a remote server for inspection. An example is Crowdroid [6] which collects system calls of running apps and sends them preprocessed to a server that applies a clustering algorithm to differentiate between benign and trojanised apps. Although this approach can offer heavy-weight security mechanisms to devices that may not otherwise have the processing power to run them, it still requires continuous monitoring, some processing and frequent communication with a cloud service, thus limiting its utility. In the second approach [24, 35, 45] an exact replica of the mobile device is stored in a virtual environment in the cloud, and the system consists of three modules: (1) a *server* applying heavy-weight security analyses on the replica, such as virus scanning [35, 45] and off-the-shelf intrusion detection systems [24, 35]; (2) a *proxy* duplicating incoming traffic to the mobile device, and forwarding it to the mirror server; and (3) a *tracer* on the mobile recording and offloading all necessary information needed to replay execution on the replica. The advantage of this approach is that it can leverage existing complex security solutions, but the processing and energy costs of synchronising the entire state of a device are prohibitive.

4 Conclusions

The goal of the NEMESYS project is to develop a novel security framework for gathering and analysing information about the nature of cyber-attacks targeting mobile devices and networks, and to identify abnormal events and malicious network activity [21]. Thus this paper summarises our proposed approaches to the analysis of network traffic and the development of anomaly detection algorithms, combining modelling and learning using network measurements, control-plane and billing data. Since network threats often map into *congestion* in the signaling system, we will use queueing models to understand bottlenecks in signaling protocols to identify and predict abnormal traffic patterns that arise in such phenomena. In addition, we will develop efficient algorithms to detect and classify attacks based on semi-supervised and unsupervised learning, that can process massive amounts of signaling and billing data in real-time, allowing the early warning of abnormal activities. Finally, we will use the OPNET simulator to run tests and identify possible issues, to contribute to the adjustment of network operational parameters and help mitigate such threats.

References

1. Aguilar J, Gelenbe E (1997) Task assignment and transaction clustering heuristics for distributed systems. Inf Sci 97(1–2):199–219
2. Arbor Networks (2012). http://www.arbornetworks.com/research/infrastructure-security-report
3. AVG (2011). http://mediacenter.avg.com/content/mediacenter/en/news/new-avg-study.html
4. Barbuzzi A, Ricciato F, Boggia G (2008) Discovering parameter setting in 3G networks via active measurements. IEEE Commun Lett 12(10):730–732
5. BBC (2012). http://www.bbc.co.uk/news/technology-20080397
6. Burguera I, Zurutuza U, Nadjm-Tehrani S (2011) Crowdroid: behavior-based malware detection system for android. In: Proceedins of SPSM '11, ACM, Chicago, pp 15–26
7. Canalys (2012). http://www.canalys.com/newsroom/smart-phones-overtake-client-pcs-2011
8. Cisco (2013). http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf
9. Enck W et al. (2005) Exploiting open functionality in SMS-capable cellular networks. In: Proceedings of CCS '05, ACM, Alexandria, pp 393–404
10. Gelenbe E (1979) Probabilistic models of computer systems. Acta Inf 12(4):285–303
11. Gelenbe E (1989) Random neural networks with negative and positive signals and product form solution. Neural Comput 1(4):502–510
12. Gelenbe E (1993) Learning in the recurrent random neural network. Neural Comput 5:154–164
13. Gelenbe E (2009) Steps towards self-aware networks. Commun ACM 52(7):66–75
14. Gelenbe E (2012) Natural computation. Comput J 55(7):848–851
15. Gelenbe E, Fourneau JM (1999) Random neural networks with multiple classes of signals. Neural Comput 11(4):953–963
16. Gelenbe E, Hussain K (2002) Learning in the multiple class random neural network. IEEE Trans. Neural Netw 13(6):1257–1267
17. Gelenbe E, Labed A (1998) G-networks with multiple classes of signals and positive customers. Eur J Oper Res 108(2):293–305
18. Gelenbe E, Loukas G (2007) A self-aware approach to denial of service defence. Comput Netw 51(5):1299–1314

19. Gelenbe E, Muntz RR (1976) Probabilistic models of computer systems: Part i (exact results). Acta Inform 7(1):35–60
20. Gelenbe E, Timotheou S, Nicholson D (2010) Fast distributed near-optimum assignment of assets to tasks. Comput J 53(9):1360–1369
21. Gelenbe E et al (2013) NEMESYS: Enhanced network security for seamless service provisioning in the smart mobile ecosystem. In: Proceedings of ISCIS (2013) LNEE. Springer, Berlin
22. GSMA (2012). http://www.gsma.com/technicalprojects/wp-content/uploads/2012/04/gsmasmarterappsforsmarterphones0112v.0.14.pdf
23. Gupta A et al (2013) Detecting MS initiated signaling DDoS attacks in 3G/4G wireless networks. In: Proceedings of COMSNETS'13, pp 1–6
24. Houmansadr A, Zonouz SA, Berthier R (2011) A cloud-based intrusion detection and response system for mobile phones. In: Proceedings of DSNW '11, IEEE Computer Society, Hong Kong, pp 393–404
25. Iland D, Pucher A, Schäuble T (2012) Detecting android malware on network level. Technical representation, UC Santa Barbara
26. Jiang N et al. (2012) Isolating and analyzing fraud activities in a large cellular network via voice call graph analysis. In: Proceedings of MobiSys '12, ACM, Lake District, UK, pp 253–266.
27. Jiantao S (2012) Analyzing the network friendliness of mobile applications. Technical representation, Huawei
28. Kim EK, McDaniel P, Porta T (2013) A detection mechanism for SMS flooding attacks in cellular networks. In: SecureComm'12, LNICST, vol 106, Springer, Berlin, pp 76–93.
29. Lee PPC, Bu T, Woo T (2009) On the detection of signaling DoS attacks on 3G/WiMax wireless networks. Comput Netw 53(15):2601–2616
30. Lever C et al. (2013) The core of the matter: analyzing malicious traffic in cellular carriers. In: Proceedings NDSS'13, San Diego, CA, pp 1–16.
31. Lookout Mobile Security (2013). https://blog.lookout.com/blog/2013/04/19/the-bearer-of-badnews-malware-google-play
32. Maslennikov D (2013) http://www.securelist.com/en/analysis/204792283/Mobile_Malware_Evolution_Part_6 Technical representaion, Kaspersky Lab
33. Murynets I, Jover RP (2012) Crime scene investigation: SMS spam data analysis. In: Proceedings of IMC '12, ACM, Boston, pp 441–452.
34. Papadopoulos S, Tzovaras D (2013) Towards visualizing mobile network data. In: Proceedings of ISCIS 2013. Springer, Berlin.
35. Portokalidis G et al. (2010) Paranoid Android: versatile protection for smartphones. In: Proceedings of ACSAC '10, ACM, Austin, pp. 347–356.
36. Qian F et al. (2010) Characterizing radio resource allocation for 3G networks. In: Proceedings of IMC '10, ACM, Melbourne, pp 137–150.
37. Qian Z et al. (2012) You can run, but you can't hide: exposing network location for targeted DoS attacks in cellular networks. In: Proceedings of NDSS'12, San Diego, pp 137–150.
38. Rethink Wireless (2012). http://www.rethink-wireless.com/2012/01/30/docomo-demands-googles-signalling-storm.htm
39. Ricciato F, Coluccia A, D'Alconzo A (2010) A review of DoS attack models for 3G cellular networks from a system-design perspective. Comput Commun 33(5):551–558
40. Schmidt AD et al (2009) Monitoring smartphones for anomaly detection. Mobile Netw Appl 14(1):92–106

41. Serror J, Zang H, Bolot JC (2006) Impact of paging channel overloads or attacks on a cellular network. In: Proceedings of WiSe '06, ACM, Los Angeles, pp 137–150
42. Vural I, Venter H (2010) Mobile botnet detection using network forensics. In: Proceedings of FIS'10, LNCS, vol 6369. Springer, Berlin, pp 57–67
43. Wang Z et al. (2011) An untold story of middleboxes in cellular networks. In: Proceedings of SIGCOMM 2011, ACM, Toronto, pp 57–67
44. Yan G, Eidenbenz S, Galli E (2009) SMS-watchdog: Profiling social behaviors of SMS users for anomaly detection. In: Proceedings of RAID '09, Springer, Saint-Malo, pp 202–223
45. Zhao B et al. (2012) Mirroring smartphones for good: A feasibility study. In: MobiQuitous'10, LNICST, vol 73. Springer, Berlin, pp 26–38

Author Index

E. Gelenbe and R. Lent (ed.), *Information Sciences and Systems 2013*,
Lecture Notes in Electrical Engineering 264, DOI: 10.1007/978-3-319-01604-7,