A Novel Fuzzy Kernel C-Means Algorithm for Document Clustering

Yingshun Yin[1], Xiaobin Zhang[1], Baojun Miao[2], and Lili Gao[1]

[1] School of computer science, Xi'an polytechnic university, Shaanxi, China
[2] Schol of mathematical Science, Xuchang University, Henan, China
yinyingshun@yahoo.com.cn, xbzhangcn@gmail.com

Abstract. Fuzzy Kernel C-Means (FKCM) algorithm can improve accuracy significantly compared with classical Fuzzy C-Means algorithms for nonlinear separability, high dimension and clusters with overlaps in input space. Despite of these advantages, several features are subjected to the applications in real world such as local optimal, outliers, the c parameter must be assigned in advance and slow convergence speed. To overcome these disadvantages, Semi-Supervised learning and validity index are employed. Semi-Supervised learning uses limited labeled data to assistant a bulk of unlabeled data. It makes the FKCM avoid drawbacks proposed. The number of cluster will great affect clustering performance. It isn't possible to assume the optimal number of clusters especially to large text corps. Validity function makes it possible to determine the suitable number of cluster in clustering process. Sparse format, scatter and gathering strategy save considerable store space and computation time. Experimental results on the Reuters-21578 benchmark dataset demonstrate that the algorithm proposed is more flexibility and accuracy than the state-of-art FKCM.

Keyword: Text clustering, Semi-supervised Learning, Fuzzy Kernel C-Means, Kernel Validity Index.

1 Introduction

Fuzzy Kernel C-Means (FKCM) algorithm [2] extends kernel methods to Fuzzy C-Means algorithm which is introduced by Bezdek [1]. FKCM algorithm achieves better performance than classical FCM for nonlinearly separable data and clusters with overlaps. FCM algorithm often minimizes the sum of square of Euclidean distance between samples and centroids. The assumption behind this measure is the belief that the data space consist of isolated elliptical regions. However, such an assumption is not always held on real world applications. Mapping the data to higher dimension space satisfies the requirement of the optimization measure.

Though FKCM algorithms have excellent performance in many applications, it suffer from several drawbacks: The c parameter specified in advance and fuzzier m, significant computation time and memory space for introducing kernel function, local optimal and bad convergence speed, These drawbacks restrict application in real world especially for large scale document clustering.

H. Li et al. (Eds.): AIRS 2008, LNCS 4993, pp. 418–423, 2008.

2 Adaptive Semi-supervised Fuzzy Kernel C-Means Algorithm

The major challenges in using FKCM algorithm on text clustering lie not in performing the clustering itself, but rather in choosing the number of clusters and tacking with the high dimensional, sparse document vectors. Worse, kernel functions always consume significant computation time and store space especially for large text collection. More unfortunately, extremely sparse feature vectors and the large difference size of clusters make some vectors be merged into the larger clusters. All these drawbacks are subject to generalize in practical applications.

2.1 Kernelised Validity Index

It's often unfeasible to predefine the number of clusters in advance for large, high-dimensional text data. With an exponential increase in the complexity and volume of data, it is blind to assign labels to document without knowing any information about categories. Many researches have been conducted. Validity indexes find the optimal c cluster that can measure the description of the data structure. It is tradeoff of the compactness and separation [4, 5, 6].

In order to save computation cost, Gauss kernel is extended to the validity index introduced by Bensaid [6]. Then the validity index function can be rewritten as

$$V_{KBszid}(U,V;c)=\sum_{i=1}^{c}\left[\frac{\sum_{j=1}^{n}u_{ij}(1-K(v_i,x_j))}{n_j\sum_{i=1}^{c}(1-K(v_i,v_j))}\right] \qquad \text{where, } n_j=\sum_{j=1}^{n}u_{ij}$$

Let $\sum_{j=1}^{n}u_{ij}(1-K(v_i,x_j))$ be compactness and be separation $n_j\sum_{i=1}^{c}(1-K(v_i,v_j))$

It's readily shown that the following advantages are true. First, it is able to observing the same size of clusters but distinct partitions. Second, by measuring the average compactness sum of each cluster, it's not sensitive to the size of cluster.

2.2 Sparse Format and Scatter-and-Gathering

Each document vector generally has small percentage of nonzero elements. Therefore, storing the data set in sparse format may not only reduce the computational time but also reduce considerable space to store it.

Inherent drawbacks of the kernel do lie in dot production computations consume significant time and kernel function need to marked memory space. To alleviate the expensive kernel computational and store cost, we introduce Scatter-and-Gathering strategy to further enhance performance [9].

The Scatter-and-Gathering strategy is an efficient way of computing vector dot production for sparse vectors. The main idea is to first scatter the sparse vector into a full length vector, then looping through the nonzero element of sparse matrix to evaluate the vector product. The strategy can explore the pipeline effect of the CPU to reduce the number of CPU cycles and lead to significant computing saving. Therefore, the strategy releases the expensive burden for kernel and makes it suitable for large-scale text data.

2.3 Semi-supervised Learning

Text feature vectors are always very high dimensional and extremely sparse, leading to the clusters with rarely data merging into large cluster. So, the performance of clustering suffers from great impact. The key advantage of incorporating prior knowledge into clustering algorithm lies in their ability to enforcing the correlations of feature vectors and enhancing the speed of convergence.

Semi-Supervised clustering falls into two general approaches that we call Constraint-based and Distance-based methods [8, 10].

We introduce the Semi-Supervised Learning into FKCM algorithm. Then, We briefly discusses the problem.

Let labeled vector B=[b_j], $j = 1,2,\ldots,n$

$$b_j = \begin{cases} 1, x_j labeled \\ 0, otherwise \end{cases} \tag{1}$$

In the help of labeled vectors, the better the degree of membership F is obtained F=$[f_{ij}]$ $1 \le i \le c, 1 \le j \le n$

For simultaneously obtaining the minimum of distance from clustering data to clustering center and the degree of prior membership, we rewrite

$$J(U,V) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m + \alpha \sum_{i=1}^{c}\sum_{j=1}^{n} (u_{ij}^m - f_{ij} b_j)^m d_{ij}^2$$

$$s.t. \begin{cases} \sum_{i=1}^{c} u_{ij} = 1, j = 1,2,\ldots,n \\ 0 < \sum_{j=1}^{n} u_{ij} < n, i = 1,2,\ldots,c \end{cases} \tag{2}$$

Here, α is coefficient that adjusts the proportion of unsupervised clustering and semi-supervised clustering. The larger α is, the more important of labeled data are. The α is proportional to labeled data. Due to labeled data are far smaller than unlabeled data, we calculate α with the equation $a = \frac{n}{M}$, where n and M represents the total number of objects and the number labeled data respectively.

Minimizing the objective function, we introduce the Lagrange multiplier λ .

$$L(U,V,\lambda) = \sum_{i=1}^{c} u_{ik}^m + \alpha \sum_{i=1}^{c} (u_{ik}^m - f_{ik} b_k)^m d_{ik}^2 - \lambda(\sum_{i=1}^{c} u_{ik} - 1) \tag{3}$$

Setting the partial derivative of the Lagrange λ and variable respectively of equation (3) to zero u_{st} yield

$$\frac{\partial L(U,V,c)}{\partial u_{st}} = mu_{st}^{m-1} d_{st}^{2} + 2\alpha m(u_{ik} - f_{ik}b_{k})^{m-1} d_{ik}^{2} - \lambda(\sum_{i=1}^{c} u_{ik} - 1) = 0 \tag{4}$$

From validity of clustering, we take the optimal value in interval [1.5, 2.5] [4]. Bezdek [1, 6] considered $m = 2$ is the optimal value. So we take $m = 2$ **,** then we have

$$u_{ij} = \frac{1}{1+\alpha}\left\{ \frac{\dfrac{1+\alpha(1-b_{j}\sum_{k=1}^{c} f_{kj})}{1-K(v_{i},x_{j})}}{\sum_{k=1}^{c}\dfrac{1}{1-K(v_{i},x_{j})}} \right\} \tag{5}$$

AS^{2}FKCM algorithm

Step1: Initial centroids using labeled data $v_i^{(0)}(i = 1,2,\ldots c)$ and termination value ε =0.01

Step2: Compute and update the degree of membership $u_{ij}^{(t)}$ using equation (5)

Step3: Compute and update kernel $K(v_i, x_j)$ using $K(v_i,x_j) = \dfrac{\sum_{j=1}^{n} u_{ij}^{m}.K(v_i,x_j)}{\sum_{j=1}^{n} u_{ij}^{m}}$

Compute kernel using Scatter-and-Gathering strategy. Update $u_{ij}^{(t)}$ to $u_{ij}^{(t+1)}$

Step4: If $\max_{i,j}\left|u_{ij}^{(t)} - u_{ij}^{(t+1)}\right| < \varepsilon$, then stop; otherwise, go to step 3

Step5: If $V_{KBszid}^{t+1} < V_{KBsaid}^{t}$, then $c = c+1$, go to step 2; otherwise, Validity index get the optimal the number of cluster, $c = c-1$ stop.

3 Experiments

3.1 Dataset

Experiments were carried out on the popular dataset which was evaluated perforance of text clustering algorithms.

Table 1. The number of labeled and unlabeled documents in five categories

	corn	grain	wheat	sugar	ship
Labeled	10	10	10	1	1
Unlabeled	90	90	90	9	9

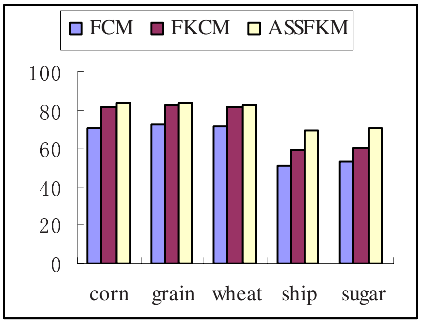

Fig 1. The performances of three algorithms on Reuters-21578

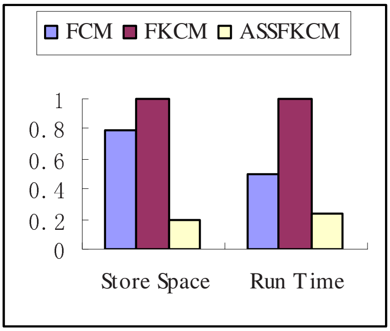

Fig 2. Comparison of store space and run time for three algorithms on the benchmark dataset

Reuters-21578: We constructed the subset of Reuter-21578 by sampling 320 documents from the original dataset. Features were extracted by removing stopwords by stoplist, performing stemming. The 1 percentage documents of each category have been not removed topics. The subset embodies the feature characteristics of a typical text collection, which are high dimensional, sparse, some significantly overlapping and skewed. The briefly describes as follows.

3.2 Results

The results on Reuter-21578 are shown in Figure 1. From the results, we see that the three algorithms perform well regarding balanced dataset with high overlapping. ASSFKCM obtains a small amount of improvement than FKCM for balanced dataset. FKCM gain better results indicate that the boundaries of text clusters are in possession of nonlinear relations. But the FCM and FKCM have no ability to tackle with small datasets overlapped with large dataset. Small datasets always be merged big cluster. Incorporating limited labeled data can marked improve in term of skewed dataset.

A few things need to be noted regarding the run time and the store space in Figure2. Generally, document vectors are all high-dimensional, extremely sparse. Storing the vector in sparse format, we save marked store space. We need not load big matrix into memory time after time. It is important to note that kernel computation and storing consume exponent power in term of the number of objects. Figure 2 indicates that sparse format and scatter-and –gathering strategy achieve sharp decrease.

4 Conclusion

FKCM algorithms which are based on minimizing the objective function have two drawbacks. The first one is sensitivity of algorithms to the initialization of the parameter c . The second is that local optimal and slow convergence speed for skewed clusters. In this paper, we first have introduced a kernelised validity index measures the fitness of clustering algorithms. The objective is to find optimal c clusters that can ensure the best description of the data structures. Then, semi-supervised learning has been explored to enhance the convergence and performance of clustering algorithms. Labeled data efficiently strengthen the correlations of unlabeled data and centroids. Last but least, avoiding the issue of high dimensionality, extremely sparse, we have introduced spare format and scatter-and-gathering strategy. In the end, the experiments on the popular benchmark datasets indicate that the algorithm proposed has high ability to automatic tackling with skewed and pronounced overlapping data clustering.

References

1. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
2. Wu, Z.-d., Xie, W.-x., Yu, J.-p.: Fuzzy C-means clustering algorithm based on kernel method. In: Proceedings of Fifth International Conference on Computational Intelligence and Multimedia Applications, pp. 49–56 (2003)
3. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis, pp. 327–338. Cambridge University Press, Cambridge (2004)
4. Pal, N.R., Bezdek, J.C.: On clustering for the fuzzy c-means model. IEEE Transaction on Fuzzy System 3(3), 370–379 (1995)
5. Xie, X.L., Beni, G.: A validity measure for fuzzy clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 841–847 (1999)
6. Bensaid, A.M., Hall, L.O., Bezdek, J.C.: Validity-guided (re)clustering with applications to image segmentation. IEEE Transactions on Fuzzy Systems, 112–123 (1996)
7. Li, K., Liu, Y.: KFCSA:A Novel clustering Algorithm for High-Dimension Data. In: Wang, L., Jin, Y. (eds.) FSKD 2005. LNCS (LNAI), vol. 3613, pp. 531–536. Springer, Heidelberg (2005)
8. Chapelle, O., Schölkopf, B., Zien, A.: Semi-Supervised Learning. MIT Press, Cambridge (2006)
9. Huang, T.-M., Kecman, V., Kopriva, I.: Kernel Based Algorithms for Mining Huge Data Sets: Supervised, Semi-supervised, and Unsupervised Learning. Springer, Berlin (2006)
10. Bouchachia, A., Pedrycz, W.: Data Clustering with Partial Supervision Data Mining and Knowledge Discovery 12, 47–78 (2006)