Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, Baoding, 12-15 July 2009

# Improved Blog Clustering through Automated Weighting of Text Blocks*

**Hongbo Li[1], Yunming Ye[1], Joshua Zhexue Huang[2]**

[1]Shenzhen Graduate School, Harbin Institute of Technology, China
[2]E-Business Technology Institute, The University of Hong Kong, Hong Kong
E-MIAL: wave118@gmail.com, yeyunming@hit.edu.cn, jhuang@eti.hku.hk

## Abstract

**In this paper, a new clustering algorithm is proposed for blog data clustering. Considering the structure information of text blocks in blog data, we group the features of blog data into three groups and extend the $k$-means clustering algorithm to automatically calculate a weight for each feature group in the clustering process. We introduce a new objective function with group weight variables and present the Lagrangian method to derive the formula to calculate the group weights. This formula is added as a new step in the standard $k$-means iterative clustering process to automatically compute the group weights according to the distribution of features. This new process guarantees the convergency of the clustering process to a local optimal solution. The experimental results have shown that this new algorithm performed better than $k$-means without group feature weighting on different blog data sets.**

## Keywords:

**Blog; clustering; auto-weighted; web mining**

## 1 Introduction

In the past few years, blog mining has become an important research area in data mining because of the fast growth of blog websites and people using blogs as a place to express their opinions on various matters and communicate with others on the Internet. In blog mining, clustering is an important method for searching and extracting useful knowledge from massive blog text data in a huge number of the blog websites spread in the world. Currently, studies on blog clustering mostly follow the conventional text clustering methods, for instance $k$-means text clustering [1], hierarchical document clustering [3], $k$-means clustering on principal components [4], document clustering using SOM [5]. These methods simply represent blog pages as flat feature vectors in the vector space model and ignore the structure information of blog pages which is important in presenting the themes or topics of the blog documents (e.g., author's opinions on a matter).

A blog document consists of three text blocks, i.e., *title*, *body* and *comments*. Importantly, they play different roles in presenting the topics and opinions of blog pages. For a blog data, the feature groups are formed through extracting the features from each text block of the three. In clustering, the feature groups should be treated differently to reflect their roles in page characterization. One method is to group the features in each text block and assign a weight to each group to differentiate the importance of each feature group in clustering. This method is used in [6] where a large weight is manually assigned to the comments group to make the comment features significant in clustering. The weight value is determined by experiments. This arbitrary manual method did not reflect the true importance of the text blocks. For example, to different blogs, comments are not always more important than title and body. In fact, many comments are only affective words such as "very good", "great", and "garbage". These features do not present what the blog page talks about. Therefore, a more meaningful method to choose block weights should consider the distribution of the features in a specific data set.

In this paper, we propose a new clustering algorithm that automatically calculates a weight for each feature group of text blocks in the clustering process. The algorithm is called the feature group weighting $k$-means algorithm or FGW-$k$-means. The new algorithm is designed as follows: Firstly, we define a new objective function by introducing group weights to the objective function of the standard $k$-means clustering process. Then, using the Lagrangian method, we derive the formula to calculate the group weights. Finally, we add the formula as a new step in the standard $k$-means iterative clustering process to automatically compute

*This research is supported in part by NSFC under grant No.60603066 and China National High-tech Program under grants No.2007AA01Z436.

978-1-4244-3703-0/09/$25.00 ©2009 IEEE

the group weight values according to the distribution of features in the current clustering. This new process guarantees the convergency of the clustering process to a local optimal solution. Since only one step is added to the clustering process, the new algorithm is still efficient in clustering large blog data. We conducted experiments on different blog data sets. The experiment results have shown that this new algorithm performed better than the $k$-means algorithms without group feature weighting.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 discusses the blogosphere document model that is used in this paper. Section 4 describes our proposed algorithm — FGW-$k$-means. Section 5 presents experimental studies on real world blog data. We give some concluding remarks in Section 6.

## 2 Related Work

If the blog pages are viewed as common web pages, blog clustering is the same as web page clustering. Besides the text clustering methods mentioned in Section 1, there are also other methods for text clustering, for example, constructing adaptive context trees for text clustering [7], the hybrid algorithm for web document clustering based on frequent item sets and $k$-means[8], and text clustering with feature selection using statistical data [9]. These are the traditional text clustering methods that represent text documents as flat vectors in the vector space model.

The text block structure information is considered in blog clustering in [6]. In this work, a big weight is manually assigned to the comments feature group before clustering. However, in practice, it is very difficult to manually select a proper weight. In our work, the weights for three feature groups are automatically calculated in clustering process.

Automated feature weighting for clustering has been an active research topic in recent years. The variable weighting algorithm W-$k$-means is proposed in [10]. Jing and et al. proposed an entropy weighting $k$-means algorithm for subspace clustering of high dimensional data [11]. The above methods assign weights to individual features and do not consider the feature groups. However, in some applications, group behaviors of features are more important than individual feature behaviors, such as blog clustering. If features can be grouped, the group behaviors can be identified by group weights which is the motivation of this work.

## 3 Text Block Based Representation Model for Blog Data

A blog page is composed of three parts, *title*, *body* and *comments*. Usually, the blog title specifies the topic of the blog page. The blog body presents the content of the blog page. The blog comments part shows the comments given by authors/readers on the blog page. The feature term distribution is very different in these three text blocks. Therefore, the discriminative capability of features in each block is different. We need to consider this difference in the clustering process. Our idea is to group features in each block and assign different weights to them in clustering.

We can extend the vector space model (VSM) to a structured vector space model to encode the blog data. In the extended VSM, we divide vectors into three groups of sub vectors, $V_t$, $V_b$ and $V_c$, where $V_t$, $V_b$ and $V_c$ contain the features of the title, body and comments blocks, respectively. The elements of a sub vector are the frequency of the corresponding terms in that block. As such, a blog page is represented as a vector $V = [V_t, V_b, V_c]$, where $V_t$, $V_b$ and $V_c$ are the sub vectors for three blocks.

Let $W = \{w_t, w_b, w_c\}$ denote the weights to the three feature groups. The word frequency matrix of blog data can be defined as follows:

$$\mathbf{V} = (v_{ljі})_{3\times(m_1+m_2+m_3)\times n} \quad (1)$$

where $v_{ljі}$ denotes the frequency of the word of the $j$th feature in the $l$th group in blog document $i$. The three blocks of title, body and comments are ordered as group 1, group 2 and group 3, respectively. The numbers of features in these groups are denoted as $m_1$, $m_2$ and $m_3$.

### 3.1 Similarity Measure

In text mining and information retrieval, consine similarity is widely used [2]. In this paper, we also use the distance based on the consine similarity, because blog is represented based on the vector space model.

Let $V_x$,$V_y$ denote the vectors of two blog documents, i.e., two columns in (1). The similarity between them is defined as

$$sim(V_x, V_y) = \frac{V_x \cdot V_y}{|V_x|\,|V_y|}$$

where $V_x \cdot V_y$ denotes the dot product of $(V_x, V_y)$, i.e., $\sum_{j=1}^{m} v_{xj} v_{yj}$ for all features. The vector norm $|V_x|$ is defined as $|V_x| = \sqrt{V_x \cdot V_x}$.

The distance between $V_x$ and $V_y$ is defined as

$$dis(V_x, V_y) = 1 - sim(V_x, V_y) \quad (2)$$

Clearly, the smaller the distance $dis(V_x, V_y)$, the larger the similarity between $V_x$ and $V_y$.

## 4 A $k$-means Type Algorithm with Automatic Weighting of Feature Groups

In this section, we present a $k$-means type algorithm with automatic weighting of feature groups. After a brief

overview of the $k$-means technique, we discuss the new technique to calculate the feature group weights in the $k$-means clustering process. Based on the new technique, we introduce the feature group weighting $k$-means algorithm FGW-$k$-means. We also briefly discuss the complexity and convergence of the new algorithm.

### 4.1 The Feature Group Weighting Technique in $k$-means

In text mining, the $k$-means algorithm is widely used because of its efficiency in clustering large text data. In $k$-means clustering, the number of clusters $k$ is specified by the user and a set of $k$ initial cluster centers are selected from the input data. Then, the $k$-means clustering process iteratively moves the cluster centers to minimize the sum of the within cluster distances.

Let $V = \{V_1, V_2, ..., V_n\}$ be a set of $n$ objects. Object $V_i = (v_{i1}, v_{i2}, ..., v_{im})$ is characterized by a set of $m$ features. The $k$-means clustering process searches for a partition of $V$ into $k$ clusters that minimizes the following objective function $F$ with unknown variables $U$ and $C$:

$$\min F(U, C) = \sum_{i=1}^{n} \sum_{l=1}^{k} u_{il} dis(V_i, C_l) \tag{3}$$

$$s.t. \begin{cases} \sum_{l=1}^{k} u_{il} = 1, 1 \le i \le n \\ u_{il} \in \{0, 1\}, \quad 1 \le i \le n, 1 \le l \le k \end{cases} \tag{4}$$

where

- $U$ is an $n \times k$ partition matrix, $u_{il}$ is a binary variable, and $u_{il} = 1$ indicates that object $i$ is allocated to cluster $l$;
- $C = \{C_1, C_2, ..., C_k\}$ is a set of $k$ vectors representing the centers of $k$ clusters;
- $dis(V_i, C_l)$ is the distance between object $i$ and the center of cluster $l$.

In blog clustering, we divide $m$ features into 3 groups, $\{V_{m_1}, V_{m_2}, V_{m_3}\}$. A blog page $V_i$ is represented as

$V_i = \{V_{m_1 i}, V_{m_2 i}, V_{m_3 i}\}$
$= \{\{v_{11i}, v_{12i}, ..., v_{1m_1 i}\}, \{v_{21i}, v_{22i}, ..., v_{2m_2 i}\}, \{v_{31i}, v_{32i}, ..., v_{3m_3 i}\}\}$.

Let $W = \{w_1, w_2, w_3\}$ denote the weight vector of the three text blocks. Combining with the $k$-means algorithm, we can translate the blog clustering problem into the optimization problem as follows:

$$\min F(U, C, W) = \sum_{i=1}^{n} \sum_{l=1}^{k} \sum_{t=1}^{3} u_{il} w_t^{\beta} \cdot dis(V_{m_t i}, C_{m_t l}) \tag{5}$$

$$s.t. \begin{cases} \sum_{l=1}^{k} u_{il} = 1, 1 \le i \le n \\ u_{il} \in \{0, 1\}, \quad 1 \le i \le n, 1 \le l \le k \\ \sum_{t=1}^{3} w_t = 1 \end{cases} \tag{6}$$

where

- $dis(V_{m_t i}, C_{m_t l})$ is the distance between object $i$ and the center of cluster $l$ in the feature group $t$, i.e., $dis(V_{m_t i}, C_{m_t l}) = 1 - \frac{V_{m_t i} \cdot C_{m_t l}}{|V_{m_t i}||C_{m_t l}|}$;
- $w_t$ is the weight of the feature group $t$.

To solve this optimization problem, we follow the optimization method in [10]. We minimize (5) by iteratively solving the following three minimization problems:

1. Problem $F_1$: Fix $C = \widehat{C}$ and $W = \widehat{W}$, and solve the reduced problem $F(U, \widehat{C}, \widehat{W})$;
2. Problem $F_2$: Fix $U = \widehat{U}$ and $W = \widehat{W}$, and solve the reduced problem $F(\widehat{U}, C, \widehat{W})$;
3. Problem $F_3$: Fix $C = \widehat{C}$ and $U = \widehat{U}$, and solve the reduced problem $F(\widehat{U}, \widehat{C}, W)$.

Problem $F_1$ is solved as follows:

$$\begin{cases} u_{i,l} = 1 & \text{if } \sum_{t=1}^{3} w_t^{\beta} dis(V_{m_t i}, C_{m_t l}) \le \\ & \quad \sum_{t=1}^{3} w_t^{\beta} dis(V_{m_t i}, C_{m_t h}), \text{for } 1 \le h \le k \\ u_{i,l} = 0 & \text{for } h \ne l \end{cases} \tag{7}$$

Problem $F_2$ is solved by the following formula

$$c_{tjl} = \frac{\sum_{i=1}^{n} u_{il} v_{tji}}{\sum_{i=1}^{n} u_{il}} \quad for \ 1 \le j \le m_t, 1 \le l \le k, t \in \{1, 2, 3\} \tag{8}$$

The formula for solving problem $F_3$ is derived as follows:

When fixing $C = \widehat{C}$ and $U = \widehat{U}$, the Lagrangian function of the optimization problem $F(\widehat{U}, \widehat{C}, W)$ is

$$\varphi(W, \lambda) = \sum_{t=1}^{3} w_t^{\beta} D_t + \lambda(\sum_{t=1}^{3} w_t - 1) \tag{9}$$

where

$$D_t = \sum_{i=1}^{n}\sum_{l=1}^{k} u_{il} \cdot dis(V_{m_t i}, C_{m_t l}) \quad (10)$$

Differentiating (9) by $W$ and $\lambda$ respectively, we obtain

$$\frac{\partial \varphi(W,\lambda)}{\partial w_t} = \beta w_t^{\beta-1} D_t + \lambda = 0, 1 \le t \le 3 \quad (11)$$

$$\frac{\partial \varphi(W,\lambda)}{\partial \lambda} = \sum_{t=1}^{3} w_t - 1 = 0 \quad (12)$$

Manipulating (11), we obtain

$$w_t = \left(\frac{-\lambda}{\beta D_t}\right)^{1/(\beta-1)} \quad (13)$$

Substituting (13) into (12), we have

$$w_t = \frac{1}{\sum_{s=1}^{3}\left(\frac{D_t}{D_s}\right)^{1/(\beta-1)}} \quad (14)$$

Finally, we obtain

$$w_t = \begin{cases} 0 & if\ D_t = 0 \\ 1/\sum_{s=1}^{3}\left(\frac{D_t}{D_s}\right)^{\frac{1}{\beta-1}} & if\ D_t \neq 0 \end{cases} \quad for\ t \in \{1,2,3\} \quad (15)$$

The above formula (15) is the optimal solution to problem $F_3$. Given a data partition, in order to adequately use the discriminative capability of different feature groups (e.g., the text blocks), the principal for feature group weighting is to assign a larger weight to a feature group that has a small sum of the within cluster distances and a small weight to a feature group that has a large sum of the within cluster distances. According to this principal and formula (15), it is easy to show that parameter $\beta$ must be in the regions of $\beta < 0 \quad or \quad \beta > 1$ .

### 4.2 The FGW-$k$-means Algorithm

Based on the above technique, we can define the feature grouping weighting $k$-means algorithm FGW-$k$-means as follows: Given a blog data, we extract the features from the three text blocks and group the features into three groups. In each group, we randomly select $k$ cluster centers as

$$C^0 = [\ C_t^0 \quad C_b^0 \quad C_c^0\ ]^T$$
$$= [\{C_{t1}, C_{t2}, ..., C_{tk}\}; \{C_{b1}, ..., C_{bk}\}; \{C_{c1}, ..., C_{ck}\}]^T$$

and randomly generate 3 initial weights for the three groups as $W^0 = [w_1^0, w_2^0, w_3^0]$ satisfying $\sum_{t=1}^{3} w_t^0 = 1$. Starting from the given blog data and the initial settings, we iteratively use the formulas (7),(8),(15) to solve the three problems $F_1$, $F_2$ and $F_3$, until the clustering process converges, i.e., the cluster centers do not change again in the subsequential iterations. The FGW-$k$-means algorithm is given in Table 1.

**Table 1. The Feature Group Weighting $k$-means algorithm FGW-$k$-means**

**Input**:

- $k$: the number of the clusters
- $D$: the data set including $n$ blog pages

**Output**: $k$ clusters
**Process**:

1. Choose $k$ objects randomly from the blogs data set $D$ as the initial centers of $k$ clusters, i.e., $C^0 = [\ C_t^0 \quad C_b^0 \quad C_c^0\ ]^T$; Randomly generate three initial group weights as $W^0 = [w_1^0, w_2^0, w_3^0]$ ($\sum_{t=1}^{3} w_t = 1$);
2. Repeat
   (a) Assign each object to the nearest cluster using formula (7);
   (b) Update the centers of clusters, i.e., calculate the center of each cluster using formula (8);
   (c) Update the weight vector, i.e., calculate the weight value of each block using formula (15).
3. Until the partition has no change

## 5 Experiments

To validate the FGW-$k$-means algorithm, we downloaded 356 blog files from Windows Live Spaces[1]. These files are manually assigned to six topics, "Olympic", "Stock", "Gun control", "Health", "Car" and "Terrorism". There are 56 blog files in "Car" topic and 60 blog files in each of the rest topics. By mixing up the blog files, we obtained 7 data sets as described in Table 2.

We used two evaluation metrics to evaluate the clustering results. The measure "entropy" gauges the distribution of each class of documents within each cluster. The measure

[1]http://spaces.live.com

**Table 2. 7 Data Sets and Their Topics**

| Blogs Data Set | Topics |
|---|---|
| **Data Set 1** | Olympic,Stock,Gun control |
| **Data Set 2** | Health, Car, Terrorism |
| **Data Set 3** | Olympic,Stock,Health |
| **Data Set 4** | Car,Stock, Health |
| **Data Set 5** | Olympic, Stock, Gun control, Health |
| **Data Set 6** | Olympic,Stock,Car ,Health, Terrorism |
| **Data Set 7** | Olympic, Stock, Gun control, Health, Car, Terrorism |

"purity" computes the extent to which each cluster contains documents from primarily one class [12]. Generally speaking, the smaller the entropy value and the larger the purity value, the better the clustering solution.

## 5.1 Experiment Results

We conducted experiments on the 7 data sets with three $k$-means type clustering algorithms, $k$-means, W-$k$-means and FGW-$k$-means. Table 3 shows the corresponding results.

Firstly,we consider the result for Data Set 1. From the result on Data Set 1 in Table 3, we can see that FGW-$k$-means got an entropy of 0.2977 and a purity of 0.8734. The two corresponding measures for $k$-means are 0.3387 and 0.7839, respectively. The two corresponding measures for W-$k$-means are 0.3439 and 0.8636 . Clearly, FGW-$k$-means performed better than $k$-means and W-$k$-means in this data set.

Similar results are obtained from other data sets as well. Except that W-$k$-means is little better than FGW-$k$-means on Data Set 3, FGW-$k$-means gets best results on the other six data sets. On the whole, it shows that FGW-$k$-means performs much better than $k$-means and W-$k$-means.

## 5.2 Weight Distribution

Fig. 1 shows the distribution of weights for the three text blocks (title, body and comments) calculated by FGW-$k$-means from Data Set 1. They are 32.79%, 25.68% and 41.53% respectively. These weight values indicated the importance of different block feature groups in the clustering result.

The difference of weights in different text blocks on this data set can be analyzed from the semantic characteristics of the three text blocks. Since the title of a blog is much shorter than the body and comments, if we mix the title with the body and comments in calculating the frequency of the words, the title contribution to the word frequency would

**Table 3. Comparison of Clustering Result on All 7 Data Sets**

| | | Entropy | Purity |
|---|---|---|---|
| **Data Set 1** | FGW-$k$-means | **0.2977** | **0.8734** |
| | $k$-means | 0.3387 | 0.7839 |
| | W-$k$-means | 0.3439 | 0.8636 |
| **Data Set 2** | FGW-$k$-means | **0.3215** | **0.8554** |
| | $k$-means | 0.3412 | 0.7714 |
| | W-$k$-means | 0.3390 | 0.8329 |
| **Data Set 3** | FGW-$k$-means | 0.2346 | 0.9111 |
| | $k$-means | 0.2942 | 0.9000 |
| | W-$k$-means | **0.2297** | **0.9164** |
| **Data Set 4** | FGW-$k$-means | **0.2223** | **0.9278** |
| | $k$-means | 0.2522 | 0.9056 |
| | W-$k$-means | 0.2436 | 0.9085 |
| **Data Set 5** | FGW-$k$-means | **0.2367** | **0.9125** |
| | $k$-means | 0.2456 | 0.9083 |
| | W-$k$-means | 0.2315 | 0.9103 |
| **Data Set 6** | FGW-$k$-means | **0.3262** | **0.8007** |
| | $k$-means | 0.3349 | 0.7635 |
| | W-$k$-means | 0.3235 | 0.7834 |
| **Data Set 7** | FGW-$k$-means | **0.2982** | **0.8343** |
| | $k$-means | 0.3353 | 0.8034 |
| | W-$k$-means | 0.3274 | 0.8157 |

be very small. However, the key words in the title usually have strong discriminative capability. The FGW-$k$-means algorithm calculates the word frequency of the three text blocks separately. The words appearing in the title are more standout in the title block.

The result shows that the title weight is 32.79%, which enhance the effect of the title block in the clustering result. The number of the words in the body block is much larger than the other two blocks, because it is the main content of the blog. However, the proportion of key words is low because of a lot of many other words that are not well related to the topic. The result shows that the body weight is 25.68%, which reduces the influence of other irrelevant words to the clustering result. The information quality of comments from different readers is intermingled with good and bad. Some comments reflect the topic information highly, others reflect little. Generally speaking, the proportion of words around the relative topic is larger. Comments has a strong discrimination capability and more words than the title block. Thus the comments weight is high, i.e., 41.53%.

The weights in different blocks can be different on other data sets, depending on the information and data quality in different blocks. Therefore, it is difficult to manually assign weights to different blocks as the method proposed in [6]. FGW-$k$-means has the advantage to adapt to inherent data

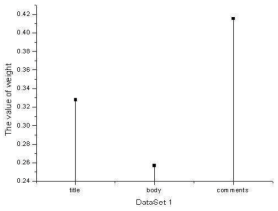


**Figure 1. The weight distribution to title, body and comments in Data Set 1**

distributions in calculating the weights in order to get the better clustering result.

### 5.3 Parameter Tuning

In the process of using FGW-$k$-means, $\beta$ can affect the clustering accuracy. We conducted sensitivity analysis on $\beta$. In Section 4, we have mentioned that $\beta$ must be $\beta < 0$ *or* $\beta > 1$ . Without loss of generality, we set $\beta$ to 2,4,6,8,10,12,14 and 16 and ran FGW-$k$-means on Data Set 1 several times. Figure 2 shows the relationship between $\beta$ and the values of entropy and purity. From this figure, we can see that the clustering accuracy approached the highest point when $\beta = 8$. Thus, in the rest of the above experiments, we choose $\beta = 8$.

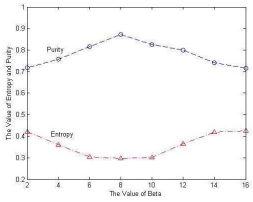


**Figure 2. The influence of $\beta$ on clustering accuracy**

## 6 Conclusions

In this paper, we have presented the FGW-$k$-means algorithm for clustering blog data. It's been shown that FGW-$k$-means automatically calculates the weights for different feature groups. This capability is very important in clustering complex data with diverse features. We have shown that FGW-$k$-means indeed performed much better in clustering real blog data than $k$-means and W-$k$-keams, a new $k$-means type algorithm that can automatically calculate weights for individual features. In the future work, we will experiment FGW-$k$-means on blog data with more feature groups such as links and time.

## References

[1] Berry M. W. , Survey of Text Mining: Clustering, Classification and Retrieval, Springer publisher, pp. 73–79, 2004

[2] Inderjit S. D. , and Dharmendra S. M. , Concept decompositions for large sparse text data using clustering, Machine Learning, v.42 n.1/2, pp. 143-175, Jan. 2001

[3] Fung B. , Wang K. , and Ester M., Hierarchical document clustering using frequent itemsets. Proceedings of the SIAM International Conference on Data Mining, San Francisco, pp. 59-70, May 2003

[4] Chris D. , and Xiaofeng H. , K-means clustering via principal component analysis, Proceedings of the twenty-first international conference on Machine learning, Banff, 2004

[5] Lagus K. , Honkela T. , Kaski S. , and Kohonen T. , Self-organizing maps of document collections: a new approach to interactive exploration, KDD'96, 1996

[6] Beibei L. , Shuting X. , and Jun Z. , Enhancing clustering blog documents by utilizing author/reader comments, Proceedings of the 45th Annual Southeast Regional Conference, Winston-Salem, North Carolina, pp. 94-99, March 2007

[7] Vert J. P. , Adaptive context trees and text clustering, IEEE Transactions on Information Theory **47**, pp. 1884-1901, 2001

[8] Le W. , Li T. , Yan J. , and Weihong H. , A hybrid algorithm for web document clustering based on frequent term sets and k-means, Lecture Notes in Computer Science **4537**, pp. 198-203, 2007

[9] Yanjun L. , Congnan L. , and Chung S. M. , Text clustering with feature selection by using statistical data, IEEE Transactions on Knowledge and Data Engineering, pp.212–217, 2008

[10] Joshua Z. H. , Michael K. Ng, Hongqiang R. , and Zichen L. , Automated variable weighting in k-means type clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence **27**, pp. 657-668, 2005

[11] Liping Jing, Michael K. Ng, and Joshua Z. H. , An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data, IEEE Transactions on Knowledge and Data Engineering **19**, pp. 1026-1041, 2007

[12] Zhao Y. , and Karypis G. , Criterion function for document clustering experiments and analysis, Technical report, Department of Computer Science and Engineering,University of Minnesota, Minneapolis, 2001