TW-k-Means: Automated Two-Level Variable Weighting Clustering Algorithm for Multiview Data

Xiaojun Chen, Xiaofei Xu, Joshua Zhexue Huang, and Yunming Ye

Abstract—This paper proposes TW-k-means, an automated two-level variable weighting clustering algorithm for multiview data, which can simultaneously compute weights for views and individual variables. In this algorithm, a view weight is assigned to each view to identify the compactness of the view and a variable weight is also assigned to each variable in the view to identify the importance of the variable. Both view weights and variable weights are used in the distance function to determine the clusters of objects. In the new algorithm, two additional steps are added to the iterative k-means clustering process to automatically compute the view weights and the variable weights. We used two real-life data sets to investigate the properties of two types of weights in TW-k-means and investigated the difference between the weights of TW-k-means and the weights of the individual variable weighting method. The experiments have revealed the convergence property of the view weights in TW-k-means. We compared TW-k-means with five clustering algorithms on three real-life data sets and the results have shown that the TW-k-means algorithm significantly outperformed the other five clustering algorithms in four evaluation indices.

Index Terms—Data mining, clustering, multiview learning, k-means, variable weighting

1 Introduction

Multiview data are instances that have multiple views (representations/variable groups) from different feature spaces. It is the result of integration of multiple types of measurements on observations from different perspectives and different types of measurements can be considered as different views. For example, the variables of the nucleated blood cell data [1] were divided into views of density, geometry, "color" and texture, each representing a view of particular measurements on the nucleated blood cells. In a banking customer data set, variables can be divided into a demographic view representing demographic information of customers, an account view showing the information about customer accounts, and the spending view describing customer spending behaviors. Web pages can be represented with three views: a term vector view whose elements correspond to the occurrence of certain words in the web page text, a hyperlink graph view that shows other web pages which each web page points to, and a term vector view for the words in the anchor text. In the past decade, multiview data has raised interests in the so-called multiview clustering [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. Different from the traditional clustering methods which take multiple views as a flat set of variables and ignore the differences among different views, multiview clustering exploits the information from multiple views and take the differences among different views into consideration in order to produce a more accurate and robust partitioning of the data.

Variable weighting clustering has been important research topic in cluster analysis [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27]. It automatically computes a weight for each variable, and identify important variables and insignificant variables through variable weights. The multiview data could be considered as have two levels of variables. In clustering the multiview data, the difference of views and the importance of individual variables in each view should be taken into account. The traditional variable weighting clustering methods only compute weights for individual variables and ignore the differences in views in the multiview data. Therefore, they are not suitable for multiview data.

To our knowledge, SYNCLUS is the first variable weighting multiview clustering algorithm which uses weights for both views and individual variables in the clustering process [28]. But it only computes variable weights automatically and the view weights are given by users. Recently, Tzortzis and Likas [9] proposed a weighted combination of exemplar-based mixture models (WCMM)

- X. Chen is with the Shenzhen Graduate School, Harbin Institute of Technology, C202#, HIT Campus at Xili University Town, Shenzhen 518055, P.R. China, and the College of Computer Science and Software, Shenzhe University, Shenzhen 518060, P.R. China. E-mail: xjchen.hitsz@gmail.com.
- X. Xu is with the Department of Computer Science and Engineering, Harbin Institute of Technology, 315#, 92 West Dazhi Street, Nan Gang District, Harbin 150001, P.R. China. E-mail: xiaofei@hit.edu.cn.
- J.Z. Huang is with the College of Computer Science and Software, Shenzhe University, Shenzhen 518060, P.R. China, and the Shenzhen Key Laboratory for High Performance Data Mining, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, 1068 Xueyuan Avenue, Shenzhen University Town, Shenzhen 518055, P.R. China. E-mail: zhexuehuang@gmail.com.
- Y. Ye is with the Shenzhen Graduate School, Harbin Institute of Technology, C202#, HIT Campus at Xili University Town, Shenzhen 518055, P.R. China. E-mail: yeyunming@hit.edu.cn.

Manuscript received 7 Apr. 2011; revised 23 Nov. 2011; accepted 9 Dec. 2011; published online 16 Dec. 2011.
Recommended for acceptance by M. Ester.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2011-04-0188.
Digital Object Identifier no. 10.1109/TKDE.2011.262.

that assigns different weights to the views and learns those weights automatically, but their method does not consider variable weights. The two algorithms have a big weakness that they are not scalable to large data sets.

In this paper, we propose TW-k-means, a novel two-level variable weighting k-means clustering algorithm for multiview data. In the TW-k-means algorithm, to distinguish the impacts of different views and different variables in clustering, the weights of views and individual variables are introduced to the distance function. The view weights are computed from the entire variables, whereas the weights of variables in a view are computed from the subset of the data that only includes the variables in the view. Therefore, the view weights reflect the importance of the views in the entire data, while the variable weights in a view only reflect the importance of variables in the view. We present an optimization model for the TW-k-means algorithm and introduce the formulae, derived from the model, for computing both view weights and variable weights. We define the TW-k-means algorithm as an extension to the standard k-means clustering process with two additional steps to compute view weights and variable weights in each iteration. Since the two steps do not require intensive computation, the new clustering algorithm remains efficient in clustering large high dimensional multiview data. Compared with SYNCLUS and WCMM, TW-k-means can automatically compute both view weights and individual variable weights. Moreover, it is a fast clustering algorithm which has the same computation complexity as k-means.

Two sets of experiments on five real-life data sets have been conducted, one was used to investigate the properties of two types of weights in TW-k-means, the other was used to verify the performance of TW-k-means in classification. With the first experiment, we discuss how to control two types of weight distributions, illustrate the differences of the weights in TW-k-means and the individual variable weighting method, and demonstrate the convergence property of view weights in TW-k-means. In the second experiment, we compared TW-k-means with five clustering algorithms and the results have shown that the TW-k-means algorithm significantly outperformed the other five in four evaluation indices.

The rest of this paper is organized as follows. Section 2 gives a brief survey of related work on multiview clustering and variable weighting clustering. We state the problem of two-level variable weighting in Section 3. Section 4 presents the TW-k-means model and an iterative algorithm to solve this model. To verify the effectiveness of TW-k-means, several experiments on real-life were carried out. Section 5 presents experiments to investigate the properties of two types of weights in TW-k-means and Section 6 presents experiments to verify the performance of TW-k-means in classification. Conclusions and future work are given in Section 7.

2 Related Work

2.1 Multiview Clustering

There exist two approaches in multiview learning [10]: *centralized* and *distributed*. *Centralized* algorithms make use of multiple representations simultaneously to discover hidden patterns from the data. Most of the existing work in multiview clustering follows the *Centralized* approach with extensions to existing clustering algorithms [3], [4], [5], [6], [7], [8], [9]. *Distributed* algorithms first cluster each view independently from others using an appropriate single-view algorithm, and then combine the individual clusterings to produce a final partitioning [10], [11].

Bickel and Scheffer [3] proposed the General multiView EM algorithm based on the co-EM algorithm and developed a two-view multinomial EM algorithm and a two-view spherical k-means algorithm. However, their methods can not guarantee to converge so they are hard for a user to decide when to stop.

Kailing et al. [4] proposed a multiview version of DBSCAN algorithm. In their method, DBSCAN is first employed on each view to produce several small clusters and a large amount of noise. Then the final clusters are determined using union and intersection of local neighborhoods.

De Sa [5] proposed a two-view spectral clustering algorithm which assumes that the views are independent. Their method is to cluster the data in each view so as to minimize the disagreement between the clusterings in each view.

Zhou and Burges [6] developed multiview spectral clustering via generalizing the usual single view normalized cut to the multiview data. The multiview normalized cut is to find a cut which is close to optimal on each graph, and it can be approximately optimized via a real-valued relaxation. The relaxation leads to vertex-wise mixture of Markov chains associated with different graphs.

Blaschko and Lampert [7] proposed a clustering algorithm for two-view data based on kernel canonical correlation analysis (CCA), called correlational spectral clustering. It uses separate similarity measures for each data representation, and allows for projection of previously unseen data that are only observed in one representation (e.g., images but not text).

Chaudhuri et al. [8] proposed a clustering algorithm which performs clustering on lower dimensional subspace of the multiple views of the data, projected via canonical correlation analysis. Two algorithms for mixtures of Gaussians and mixtures of log concave distributions were developed.

Long et al. [10] proposed a general model for multiview clustering under a distributed framework. The proposed model introduces the concept of mapping function to make the different patterns from different pattern spaces comparable and hence an optimal pattern can be learned from the multiple patterns of multiple views.

Greene and Cunningham [11] proposed a clustering algorithm for multiview data using a late integration strategy. In their method, a matrix that contains the partitioning of every individual view is created and then decomposed to two matrices using matrix factorization approach: the one showing the contribution of those partitionings to the final multiview clusters, called metaclusters, and the other assigning instances to the metaclusters.

The current multiview clustering methods take both multiple views and individual variables into consideration. However, most of them are extensions to EM or spectral clustering so they are not scalable to large data sets.

2.2 Variable Weighting Clustering

Variable weighting clustering has been important research topic in cluster analysis [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27].

Huang et al. [20] proposed the W-k-means clustering algorithm that can automatically compute variable weights in the k-means clustering process. W-k-means extends the standard k-means algorithm with one additional step to compute variable weights at each iteration of the clustering process. The variable weight is inversely proportional to the sum of the within-cluster variances of the variable. As such, noise variables can be identified and their affects on the clustering result are significantly reduced. The new algorithm we propose in this paper weights both views and individual variables and is an extension to W-k-means.

Domeniconi et al. [21] have proposed the locally adaptive clustering (LAC) algorithm which assigns a weight to each variable in each cluster. They use an iterative algorithm to minimize its objective function. Jing et al. [22] pointed out that "*the objective function of LAC is not differentiable because of a maximum function. The convergence of the algorithm is proved by replacing the largest average distance in each dimension with a fixed constant value*."

Jing et al. [22] have proposed the entropy weighting k-means (EWKM) which also assigns a weight to each variable in each cluster. Different from LAC, EWKM extends the standard k-means algorithm with one additional step to compute variable weights for each cluster at each iteration of the clustering process. The weight is inversely proportional to the sum of the within-cluster variances of the variable in the cluster.

Hoff [23] proposed a multivariate Dirichlet process mixture model which is based on a Pólya urn cluster model for multivariate means and variances. The model is learned by a Markov chain Monte Carlo process. However, its computational cost is prohibitive. Bouveyron et al. [24] proposed the GMM model which takes into account the specific subspaces around which each cluster is located, and therefore limits the number of parameters to estimate. Tsai and Chiu [25] developed a variable weights self-adjustment mechanism for k-means clustering on relational data sets, in which the variable weights are automatically computed by simultaneously minimizing the separations within-clusters and maximizing the separations between clusters. Deng et al. [26] proposed an enhanced soft subspace clustering algorithm (ESSC) which employs both within-cluster and between-cluster information in the subspace clustering process. Cheng et al. [27] proposed another weighted k-means approach very similar to LAC, but allowing for incorporation of further constraints.

The traditional variable weighting clustering methods only compute weights for individual variables and ignore the differences in views in the multiview data. Therefore, they are not suitable for clustering of multiview data.

2.3 Variable Weighting Multiview Clustering

The variable weighting multiview clustering, as a combination of variable weighting method clustering and multiview clustering method, is a new direction for clustering of multiview data.

To our knowledge, SYNCLUS is the first clustering algorithm that uses weights for both views and individual variables in the clustering process [28]. The SYNCLUS clustering process is divided into two stages. Starting from an initial set of variable weights, SYNCLUS first uses the k-means clustering process to partition data into k clusters. It then estimates a new set of optimal weights by optimizing a weighted mean-square, stress-like cost function. The two stages iterate until the clustering process converges to an optimal set of variable weights. SYNCLUS only computes variable weights automatically and the view weights are given by users. Another weakness of SYNCLUS is that it is time-consuming [29] so it cannot process large data sets.

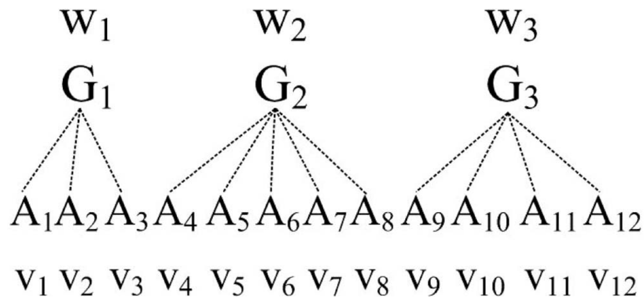


Fig. 1. Example of two-level variable weighting.

Tzortzis and Likas [9] proposed a weighted combination of exemplar-based mixture models for clustering multiview data that assigns different weights to the views and learns those weights automatically. In each view, the data is modeled using exemplar-based mixture models, called convex mixture models (CMMs) [30]. However, this method does not consider individual variable weights so it can not capture the differences among variables in a view. Moreover, WCMM employs two nested iterations and has an overall complexity of $O(N^2 V\tau\tau')$, where N is the number of objects, V is the number of views, and τ and τ' are the numbers of two nested iterations. Therefore, this method is not scalable to large data sets.

To sum up, the current two variable weighting multiview clustering methods cannot automatically compute weights for both views and individual variables. Moreover, they are not scalable to large data sets. The proposed method can automatically compute two types of weights and it retains the efficiency of the k-means algorithm.

3 Problem Statement

Fig. 1 illustrates an example of two-level variable weighting method. Let $\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\}$ be a set of n objects represented by the set $\boldsymbol{A}$ of m variables. Assume $\boldsymbol{A}$ is divided into T views $\{G_t\}_{t=1}^T$ where $G_t \bigcap G_s = \emptyset$ for $s \neq t$ and $\bigcup_{t=1}^T G_t = \boldsymbol{A}$. Let $W = \{w_1, w_2, \ldots, w_T\}$ be a set of T view weights, where w_t indicates the weight that is assigned to the tth view and $\sum_{t=1}^T w_t = 1$. Let $V = \{V_j\}$ be a set of m variable weights, where v_j indicates the weight that is assigned to the jth variable and $\sum_{j\in G_t} v_j = 1$ $(1 \leq t \leq T)$, $\sum_{j=1}^m v_j = T$. Assume that $\boldsymbol{X}$ contains k clusters. We want to discover the set of k clusters from $\boldsymbol{G}$. We also want to identify the important views from the view weight matrix $W = [w_t]_T$ and identify the important variables from the variable weight matrix $V = [v_j]_m$. If we consider $\boldsymbol{G}$ as the set of individual variables in data $\boldsymbol{X}$, this problem is equivalent to the individual variable weighting. Therefore, we can consider this two-level variable weighting method as a generalization of the current variable weighting methods.

In the two-level variable weighting method, the variable weights V are used to identify the important variables in each view, and the view weights W are used to identify compact cluster structures within these views. If the view contains compact cluster structures, a large view weight is assigned so as to enhance the effect of such view; on the contrary, if the view contains loose cluster structures, a small view weight is assigned to eliminate the effect of such view. Compared with the traditional variable weighting method, the new method can take both individual variables and multiple views into consideration and capture the differences among different views and variables.

Moreover, the traditional variable weighting methods suffer from unbalanced phenomenon: the view with more variables will play more important role than the view with less variables. In the two-level variable weighting method, the view weights will be only determined in the view level, while the variable weights will be only determined in a view. Therefore, the two levels of variable weights will eliminate the unbalanced phenomenon and compute more objective weights.

SYNCLUS can be considered as the first two-level variable weighting clustering algorithm, but it only computes variable weights automatically and the view weights are given by users. In this paper, we propose an automated two-level variable weighting clustering algorithm.

4 The TW-k-Means Clustering Algorithm

4.1 The Optimization Model

The clustering process to partition $\boldsymbol{X}$ into k clusters with weights for both views and individual variables is modeled as minimization of the following objective function:

$$\begin{aligned} P(U,Z,V,W) = &\sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{t=1}^{T}\sum_{j\in G_t} u_{i,l} w_t v_j d(x_{i,j}, z_{l,j}) \\ &+ \eta\sum_{j=1}^{m} v_j \log(v_j) + \lambda\sum_{t=1}^{T} w_t\log(w_t) \end{aligned} \tag{1}$$

subject to

$$\begin{cases} \sum_{l=1}^{k} u_{i,l} = 1, \quad u_{i,l}\in\{0,1\}, \quad 1\le i\le n, \\ \sum_{t=1}^{T} w_t = 1, \quad 0\le w_t\le 1, \\ \sum_{j\in G_t} v_j = 1, \quad 0\le v_j\le 1, \quad 1\le t\le T, \end{cases} \tag{2}$$

where

- U is a $n\times k$ partition matrix whose elements $u_{i,l}$ are binary where $u_{i,l}=1$ indicates that object i is allocated to cluster l;
- $Z=\{Z_1,Z_2,\ldots,Z_k\}$ is a set of k vectors representing the centers of the k clusters;
- $W=\{w_1,w_2,\ldots,w_T\}$ are T weights for T views;
- $V=\{v_1,v_2,\ldots,v_m\}$ are m weights for m variables;
- $\lambda>0$, $\eta>0$ are two given parameters;
- $d(x_{i,j},z_{l,j})$ is a distance or dissimilarity measure on the jth variable between the ith object and the center of the lth cluster. If the variable is numerical, then

$$d(x_{i,j},z_{l,j}) = (x_{i,j}-z_{l,j})^2. \tag{3}$$

If the variable is categorical, then

$$d(x_{i,j},z_{l,j}) = \begin{cases} 0, & (x_{i,j}=z_{l,j}), \\ 1, & (x_{i,j}\neq z_{l,j}). \end{cases} \tag{4}$$

The first term in (1) is the sum of the within cluster dispersions, the second and the third terms are two negative weight entropies. Two positive parameters λ and η control the strengths of the incentive for clustering on more views and variables.

4.2 The TW-k-Means Clustering Algorithm

We can minimize (1) by iteratively solving the following four minimization problems:

1. Problem P_1: Fix $Z=\hat{Z}$, $V=\hat{V}$, and $W=\hat{W}$, and solve the reduced problem $P(U,\hat{Z},\hat{V},\hat{W})$;
2. Problem P_2: Fix $U=\hat{U}$, $V=\hat{V}$, and $W=\hat{W}$, and solve the reduced problem $P(\hat{U},Z,\hat{V},\hat{W},)$;
3. Problem P_3: Fix $U=\hat{U}$, $Z=\hat{Z}$ and $W=\hat{W}$, and solve the reduced problem $P(\hat{U},\hat{Z},V,\hat{W})$;
4. Problem P_4: Fix $U=\hat{U}$, $Z=\hat{Z}$, and $V=\hat{V}$, and solve the reduced problem $P(\hat{U},\hat{Z},\hat{V},W)$.

Problem P_1 is solved by

$$\begin{cases} u_{i,l}=1, & \text{if} \quad D_l\le D_s \text{ for } 1\le s\le k, \\ & where \quad D_s = \sum_{t=1}^{T}\sum_{j\in G_t} w_t v_j d(x_{i,j},z_{s,j}), \\ u_{i,s}=0, & \text{for} \quad s\neq l, \end{cases} \tag{5}$$

and problem P_2 is solved for the numeric variables by

$$z_{l,j} = \frac{\sum_{i=1}^{n} u_{i,l}x_{i,j}}{\sum_{i=1}^{n} u_{i,l}} \quad \text{for } 1\le l\le k. \tag{6}$$

If the variable is categorical, then

$$z_{l,j} = a_j^r, \tag{7}$$

where a_j^r is the mode of the variable values of the jth variable in cluster l [31].

The solution to problem P_3 is given by Theorem 1.

Theorem 1. *Let $U=\hat{U}$, $Z=\hat{Z}$, and $W=\hat{W}$ be fixed. $P(\hat{U},\hat{Z},V,\hat{W})$ is minimized iff*

$$v_j = \frac{\exp\{\frac{-E_j}{\eta}\}}{\sum_{h\in G_t}\exp\{\frac{-E_h}{\eta}\}}, \tag{8}$$

where

$$E_j = \sum_{l=1}^{k}\sum_{i=1}^{n} \hat{u}_{i,l}\hat{w}_t d(x_{i,j},\hat{z}_{l,j}). \tag{9}$$

Here, t is the index of the view that the jth variable is assigned to

Proof. We minimize the objective function (1) with respect to v_j, the variable weight of the jth variable. Since there exist a set of T constraints $\sum_{j\in G_t} v_j = 1$ (for $1\le t\le T$), we form the Lagrangian by isolating the terms which contain $\{v_1,\ldots,v_m\}$ and adding the appropriate Lagrangian multipliers as

$$\begin{aligned} L_{[v_1,\ldots,v_m]} = \sum_{t=1}^{T}\Bigg[&\sum_{j\in G_t} v_j E_j + \eta\sum_{j\in G_t} v_j\log(v_j) \\ &+ \gamma_t\Bigg(\sum_{j\in G_t} v_j - 1\Bigg)\Bigg], \end{aligned}$$

where E_j is a constant of the jth variable for fixed $\hat{U}$, $\hat{Z}$, and $\hat{W}$, each of which is defined in (9).

By setting the gradient of $L_{[v_1,\ldots,v_m]}$ with respect to γ_t and v_j to zero, we obtain

$$\frac{\partial L_{[v_1,\ldots,v_m]}}{\partial \gamma_t} = \sum_{j\in G_t} v_j - 1 = 0, \tag{10}$$

and

$$\frac{\partial L_{[v_1,\ldots,v_m]}}{\partial v_j} = E_j + \eta(1+\log(v_j)) + \gamma_t = 0, \tag{11}$$

where t is the index of the view that the jth variable is assigned to.

From (11), we obtain

$$\begin{aligned} v_j &= \exp\left\{\frac{-E_j - \gamma_t - \eta}{\eta}\right\} \\ &= \exp\left\{\frac{-E_j - \eta}{\eta}\right\}\exp\left\{\frac{-\gamma_t}{\eta}\right\}. \end{aligned} \tag{12}$$

Substituting (12) into (10), we have

$$\sum_{j\in G_t} \exp\left\{\frac{-E_j-\eta}{\eta}\right\}\exp\left\{\frac{-\gamma_t}{\eta}\right\} = 1. \tag{13}$$

It follows that

$$\exp\left\{\frac{-\gamma_t}{\eta}\right\} = \frac{1}{\sum_{j\in G_t}\exp\{\frac{-E_j-\eta}{\eta}\}}.$$

Substituting this expression back into (12), we obtain

$$v_j = \frac{\exp\{\frac{-E_j}{\eta}\}}{\sum_{h\in G_t}\exp\{\frac{-E_h}{\eta}\}}.$$

□

The solution to problem P_4 is given by Theorem 2.

Theorem 2. *Let $U = \hat{U}$, $Z = \hat{Z}$, and $V = \hat{V}$ be fixed. $P(\hat{U}, \hat{Z}, \hat{V}, W)$ is minimized iff*

$$w_t = \frac{\exp\{\frac{-D_t}{\lambda}\}}{\sum_{h=1}^{T}\exp\{\frac{-D_h}{\lambda}\}}, \tag{14}$$

where

$$D_t = \sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{j\in G_t} \hat{u}_{i,l}\hat{v}_j d(x_{i,j}, \hat{z}_{l,j}). \tag{15}$$

Proof. We minimize the objective function (1) with respect to w_t, the weight of the tth view. Since there exists a constraint $\sum_{t=1}^{T} w_t = 1$, we form the Lagrangian by isolating the terms which contain $\{w_1, \ldots, w_T\}$ and adding the appropriate Lagrangian multipliers as

$$L_{[w_1,\ldots,w_T]} = \sum_{t=1}^{T} w_t D_t + \lambda \sum_{t=1}^{T} w_t \log(w_t) + \gamma\left(\sum_{t=1}^{T} w_t - 1\right),$$

where D_ts are T constants for fixed $\hat{U}$, $\hat{Z}$, and $\hat{V}$, each of which is defined in (15).

Taking the derivative with respect to w_t and setting it to zero yields a minimum at (where we have dropped the argument γ)

$$w_t = \frac{\exp\{\frac{-D_t}{\lambda}\}}{\sum_{h=1}^{T}\exp\{\frac{-D_h}{\lambda}\}}.$$

□

The TW-k-means algorithm that minimizes the objective function (1), using (5), (6), (7), (8), (9), (14), and (15), is given as Algorithm 1.

Algorithm 1. Algorithm: TW-k-means
Input: The number of clusters k and two positive real parameters λ, η;
Output: Optimal values of U, Z, V, and W;
Randomly choose k cluster centers Z^0;
for $t = 1$ to T **do**
  $w_t^0 \leftarrow 1/T$
  for all $j \in G_t$ **do**
    $v_j^0 \leftarrow 1/|G_t|$
  end for
end for
$r \leftarrow 0$
repeat
  Update U^{r+1} by (5);
  Update Z^{r+1} by (6) or (7);
  Update V^{r+1} by (8) and (9);
  Update W^{r+1} by (14) and (15);
  $r \leftarrow r + 1$
until the objective function (1) obtains its local minimum value;

The input parameters η and λ are used to control the distribution of the two types of weights V and W. We can easily verify that the objective function (1) can be minimized with respect to V and W iff $\eta \geq 0$ and $\lambda \geq 0$. Moreover, they are used as follows:

- $\eta > 0$. In this case, according to (8), v is inversely proportional to E. The smaller E_j, the larger v_j, the more important the corresponding variable.
- $\eta = 0$. In this case, according to (8), $\eta = 0$ will produce a clustering result with only one important variable in a view. It may not be desirable for high-dimensional data.
- $\lambda > 0$. In this case, according to (14), w is inversely proportional to D. The smaller D_t, the larger w_t, the more compact the corresponding view.
- $\lambda = 0$. In this case, according to (14), $\lambda = 0$ will produce a clustering result with only one important view. It may not be desirable for multiview data.

In general, η and λ are set as positive real values.

Since the sequence of $(P_1, P_2, \ldots)$ generated by the algorithm is strictly decreasing, Algorithm 1 converges to a local minimum.

TW-k-means algorithm is an extension to the k-means algorithm by adding two additional steps to calculate the view weights and individual variable weights in the iterative process. It does not seriously affect the scalability of the k-means clustering process in clustering large data. If

TW-k-means algorithm needs r iterations to converge, the computational complexity of the algorithm is $O(rknm)$. Therefore, TW-k-means has the same computational complexity as k-means.

5 Experiments on Properties of TW-k-Means

In this section, we present experiments to investigate the relationship of two types of weights w, v and three parameters k, η, λ in TW-k-means. We investigate the difference of the weights of TW-k-means and the weights of the individual variable weighting method, and demonstrate the convergence property of the view weights in TW-k-means.

5.1 Entropy-Based W-k-Means (EW-k-Means)

We consider a variant of W-k-means in [20], entropy-based W-k-means, with the following optimization function:

$$P(U,Z,W) = \sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} u_{i,l}\phi_j d(x_{i,j}, z_{l,j}) + \eta\sum_{j=1}^{m}\phi_j \log(\phi_j), \tag{16}$$

in which ϕ are a set of m variable weights.

We can easily verify that the above model has the same optimal values of u and z as TW-k-means. For the variable weights ϕ, the optima can be computed by

$$\phi_j = \frac{\exp\left\{\frac{-F_j}{\eta}\right\}}{\sum_{j=1}^{m}\exp\left\{\frac{-F_j}{\eta}\right\}}, \tag{17}$$

where F_j is the sum of the dispersions in the jth variable, defined as

$$F_j = \sum_{l=1}^{k}\sum_{i=1}^{n}\hat{u}_{i,l} d(x_{i,j}, \hat{z}_{l,j}). \tag{18}$$

EW-k-means and W-k-means both are individual variable weighting clustering methods, but they have different regularization terms. Comparing the formulae for computation of the variable weights in the two methods, (17) can avoid the computation problem caused by the zero value of F_j.

EW-k-means in (16) is very similar to TW-k-means. If we drop the view weights w in (1), TW-k-means will degenerate to (16). In the following, we will use EW-k-means to compare the distribution of the variable weights between the two-level variable weighting method and the individual variable weighting method.

5.2 Characteristics of Two Real-Life Data Sets

The *Water Treatment Plant* data set came from the daily measures of sensors in an urban waste water treatment plant [32]. This data set contains 527 instances and 38 features. The 38 features can be naturally divided into four views.

- **Input view.** Contains the first 22 features describing different input conditions.
- **Output view.** Contains the 23th-29th features describing output demands.
- **Performance input view.** Contains the 30th-34th features describing performance input demands.
- **Global performance input view.** Contains the 35th-38th features describing global performance input demands.

Here, we use G_1, G_2, G_3, and G_4 to represent the four views.

The *Yeast Cell Cycle* data set is microarray data from yeast cultures synchronized by four methods: α factor arrest, elutriation, arrest of a cdc15 temperature-sensitive mutant and arrest of a cdc28 temperature-sensitive mutant [33]. Further, it includes data for the B-type cyclin Clb2p and G1 cyclin Cln3p induction experiments. The data set is publicly available at http://genome-www.stanford.edu/cellcycle/. The original data contains 6,178 genes. In this investigation, we selected 6,076 genes on 77 experiments and removed those which had incomplete data. We used the following five views.

- G_1: contains four features from the B-type cyclin Clb2p and G1 cyclin Cln3p induction experiments;
- G_2: contains 18 features from the α factor arrest experiment;
- G_3: contains 24 features from the elutriation experiment;
- G_4: contains 17 features from the arrest of a cdc15 temperature-sensitive mutant experiment;
- G_5: contains 14 features from the arrest of a cdc28 temperature-sensitive mutant experiment.

In the following, we use the two real-life data sets to investigate the properties of two types of weights in TW-k-means.

5.3 Controlling Weight Distributions

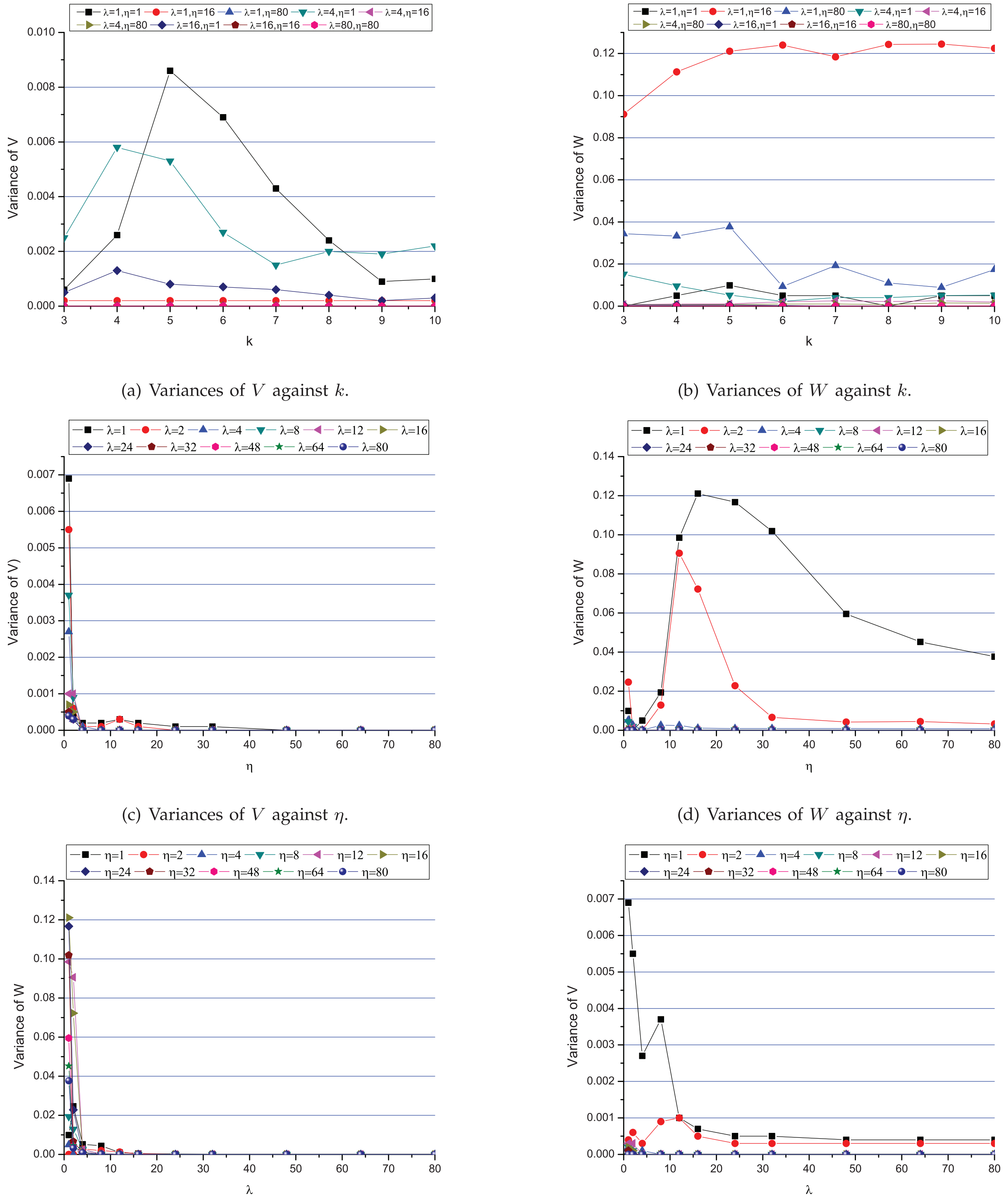
We set the number of clusters k as $\{3,4,5,6,7,8,9,10\}$, η and λ as $\{1,2,4,8,12,16,24,32,48,64,80\}$. For each combination of k, η, and λ, we ran TW-k-means to produce 100 clustering results and computed the average variances of the two types of weights V and W in the 100 results. Fig. 2 shows these variances.

From Figs. 2a, we can see that when η was small, the variances of V were unstable with the increase of k. When η was large, the variances of V became almost constant. From Fig. 2b, we can see λ has similar behavior.

To investigate the relationship of V, W with η, λ, we draw results with $k = 5$ in Figs. 2c, 2d, 2e, and 2f. From Fig. 2c, we can see that as η increased, the variance of V decreased rapidly. This result can be explained from (8): as η increases, V becomes flatter. Fig. 2d shows that the effect of η on the variance of W was not obvious. Fig. 2e shows that as λ increased, the variance of W decreased rapidly. This result can be explained from (14): as λ increases, W becomes flatter. We can see similar behavior of λ on V from Fig. 2f.

From above analysis, we summarize the following method to control two types of weight distributions in TW-k-means by setting different values of η and λ:

- Large η makes more variables contribute to the clustering while small η makes only important variables contribute to the clustering;
- Large λ makes more views contribute to the clustering while small λ makes only important views contribute to the clustering.



(a) Variances of V against k.

(b) Variances of W against k.

(c) Variances of V against η.

(d) Variances of W against η.

(e) Variances of W against λ.

(f) Variances of V against λ.

Fig. 2. The variances of two types of weights V and W against three parameters k, η, and λ in TW-k-means on the *Water Treatment Plant* data set.

5.4 Comparison of the Weights in TW-k-Means and EW-k-Means

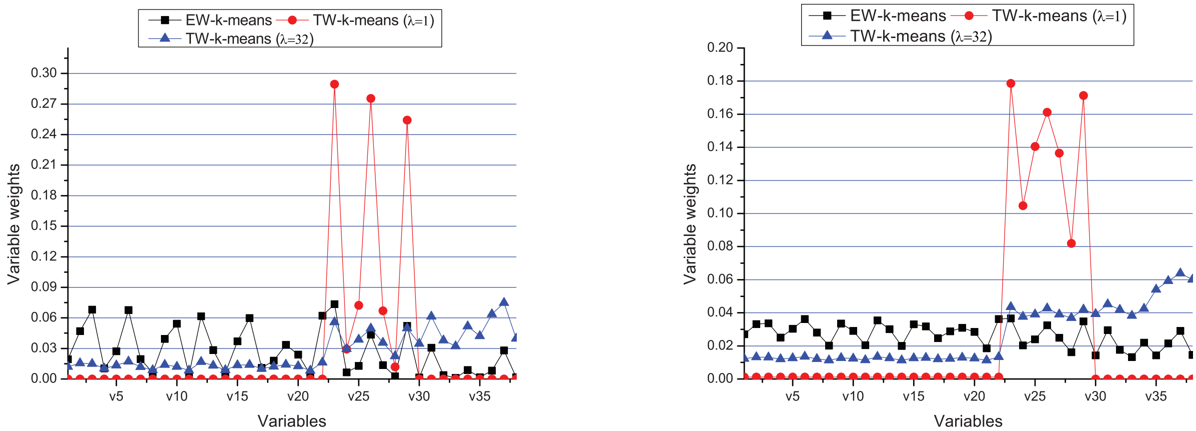
In TW-k-means, both view weights and individual variable weights act on a variable and produce on the jth variable the total variable weight φ_j as

$$\varphi_j = w_t v_j,$$

where t is the index of the view which the jth variable is assigned to. We can easily verify that

$$\sum_{j=1}^{m} \varphi_j = \sum_{t=1}^{T} \sum_{j \in G_t} w_t v_j = 1$$

From above analysis, we can see that the two-level variable weighting method can be considered as a special



(a) $\eta = 8$.

(b) $\eta = 32$.

Fig. 3. Comparison of the total variable weights in TW-k-means and EW-k-means on the *Water Treatment Plant* data set.

type of the variable weighting method. In the following, we compare the total variable weights of TW-k-means and EW-k-means.

We set the number of clusters $k = 5$, η in EW-k-means and TW-k-means as $\{8, 32\}$ and λ in TW-k-means as $\{1, 32\}$. For each setting, we ran EW-k-means and TW-k-means on the *Water Treatment Plant* data set to produce one clustering result, with the same initial cluster centers. Fig. 3 shows the total variable weights in these clustering results. From these figures, we can see that the total variable weights in TW-k-means were view-related: the overall weights of variables in compact views were enhanced, while the overall weights of variables in loose views were reduced. For example, in Fig. 3b, the total variable weights of variables in G_2 of TW-k-means were enhanced especially when λ is small, and the total variable weights of variables in G_1 of TW-k-means were reduced. We can observe similar phenomena in Fig. 3a. When λ becomes smaller, the enhancing and decreasing become more significant. Therefore, we can adjust the role of views in composing of the total variable weights in TW-k-means by adjusting λ: increasing λ to increase the role of views, while decreasing λ to decrease the role of views.

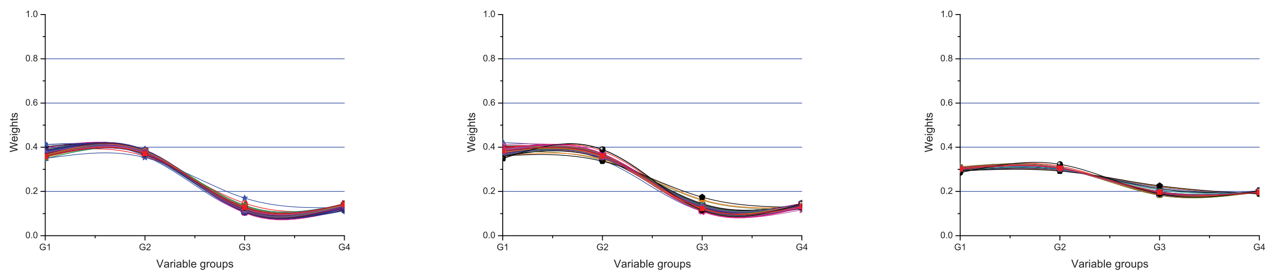
From Fig. 3, we also observe that the distributions of the total variable weights in TW-k-means and EW-k-means were different. The total variable weights in EW-k-means are only related to individual variables, while the total variable weights in TW-k-means are related to both views and individual variables. As λ increases, view weights w will become evener, and TW-k-means will degenerate to EW-k-means. Therefore, EW-k-means is just a special case of TW-k-means, and TW-k-means provides wider variable weight space than EW-k-means.

5.5 Convergence Property of View Weights

In the previous section, we have investigated the relationship between the view weights and three parameters of k, λ, and η. The results have shown that the view weights are weakly affected by k. In this experiment, we investigate the view weights against λ and η. This experiment was conducted on the *Water Treatment Plant* data set and the *Yeast Cell Cycle* data set.

We set $k = 5$ for the *Water Treatment Plant* data set and $k = 10$ for the *Yeast Cell Cycle* data set, and chose λ and η as $\{10, 20\}$. For each pair of λ and η, we randomly generated 100 sets of cluster centers, and used TW-k-means to cluster the two data sets with these settings.

Fig. 4 draws the final view weights of 300 clustering results on the *Water Treatment Plant* data set and Fig. 5 draws the final view weights of 300 clustering results on the *Yeast Cell Cycle* data set. From these figures, we can see that all results show that G_1 and G_2 were with higher view weights than G_3 and G_4. We can clearly observe the convergence property of the view weights in the clustering results: all clustering results converge to similar view weights. Under this property, TW-k-means can be used to do view selection.

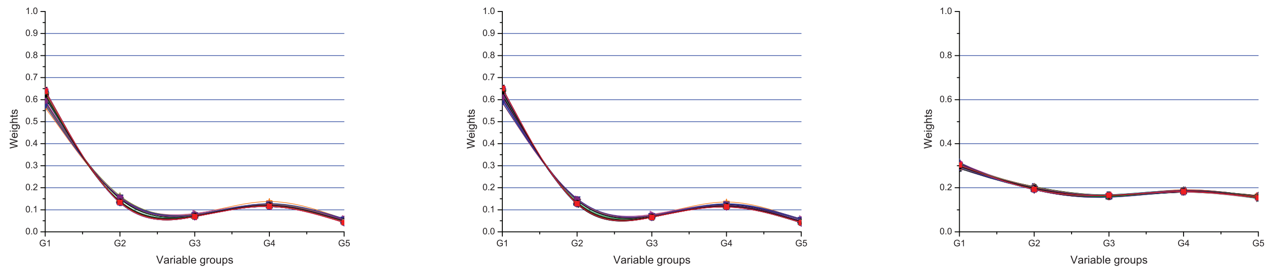


(a) $\lambda = 10$, $\eta = 10$.

(b) $\lambda = 10$, $\eta = 20$.

(c) $\lambda = 20$, $\eta = 10$.

Fig. 4. The final view weights of TW-k-means on the *Water Treatment Plant* data set.



(a) $\lambda = 10$, $\eta = 10$. (b) $\lambda = 10$, $\eta = 20$. (c) $\lambda = 20$, $\eta = 10$.

Fig. 5. The final view weights of TW-k-means on the *Yeast Cell Cycle* data set.

6 Experiments on Classification Performance of TW-k-Means

To investigate the performance of the TW-k-means algorithm in classifying real-life data, we selected three data sets from UCI Machine Learning Repository [32]: the *Multiple Features* data set, the *Internet Advertisement* data set and the *Image Segmentation* data set. With these data, We compared TW-k-means with four individual variable weighting clustering algorithms, i.e., W-k-means [20], EW-k-means (see Section 5.1), LAC [21] and EWKM [22], and a weighted multiview clustering algorithm WCMM [9].

6.1 Characteristics of Three Real-Life Data Sets

The *Multiple Features* data set contains 2,000 patterns of handwritten numerals that were extracted from a collection of Dutch utility maps. These patterns were classified into 10 classes ("0"-"9"), each having 200 patterns. Each pattern was described by 649 features that were divided into the following six views:

1. mfeat-fou view: contains 76 Fourier coefficients of the character shapes;
2. mfeat-fac view: contains 216 profile correlations;
3. mfeat-kar view: contains 64 Karhunen-Love coefficients;
4. mfeat-pix view: contains 240 pixel averages in 2×3 windows;
5. mfeat-zer view: contains 47 Zernike moments;
6. mfeat-mor view: contains 6 morphological variables.

Here, we use G_1, G_2, G_3, G_4, G_5, and G_6 to represent the six views.

The *Internet Advertisement* data set contains a set of 3,279 images from various web pages that are categorized either as advertisements or nonadvertisements (i.e., two classes). The instances are described in six sets of 1,558 features, which are the geometry of the images (width, height, aspect ratio), the phrases in the url of the pages containing the images (base url), the phrases of the imagesurl (image url), the phrases in the url of the pages the images are pointing at (target url), the anchor text, and the text of the images alt (alternative) html tags (alt text). All views have binary features, apart from the geometry view whose features are continuous. Details for the construction of the data set can be found in [34].

The *Image Segmentation* data set consists of 2,310 objects drawn randomly from a database of seven outdoor images. The data set contains 19 features which can be naturally divided into two views.

1. Shape view: contains nine features about the shape information of the seven images;
2. RGB view: contains 10 features about the RGB values of the seven images.

Here, we use G_1 and G_2 to represent the two views.

All three data sets are very complex, especially the first two data sets with very high dimension. In the following, we use the three real-life data sets to investigate the classification performance of the TW-k-means clustering algorithm.

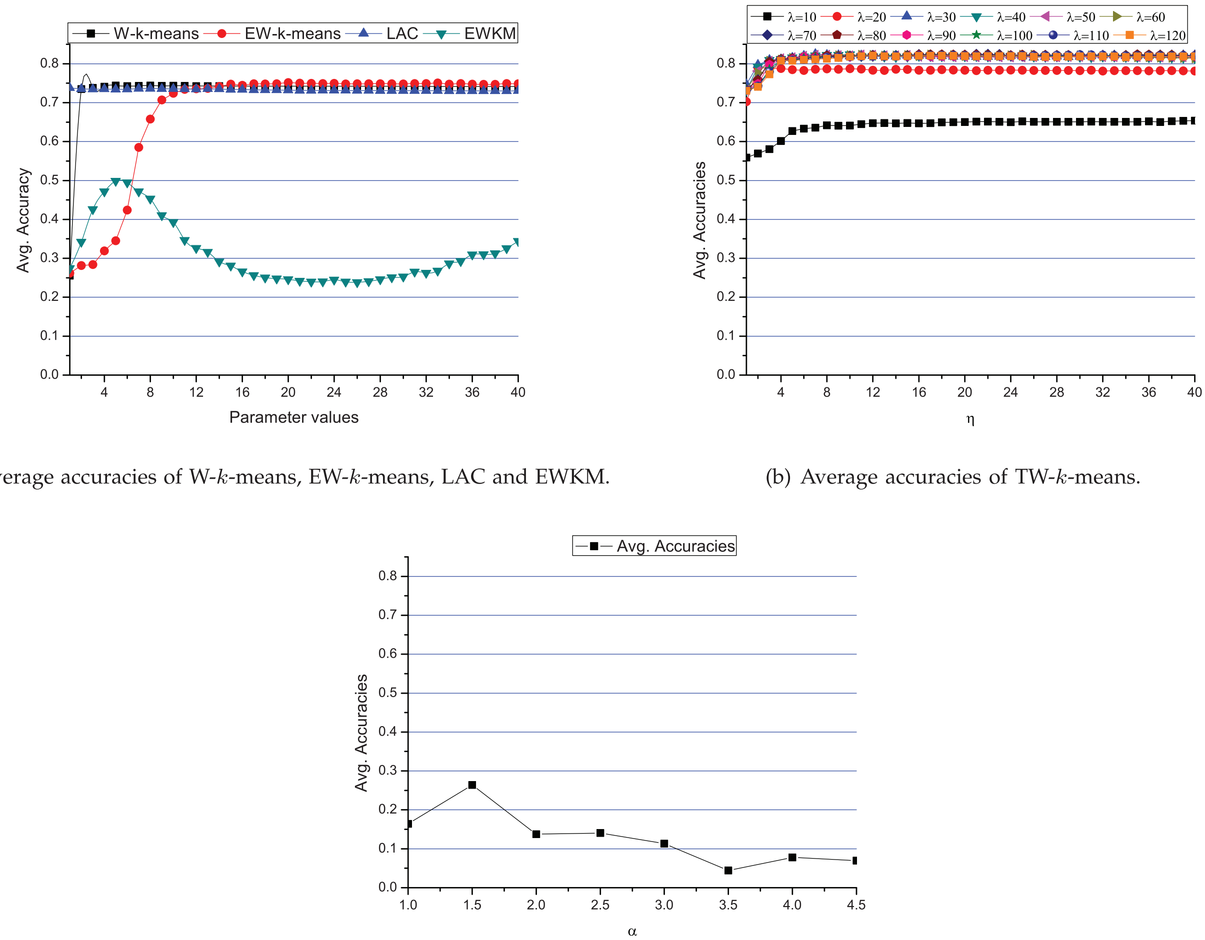
6.2 Experiment Setup

With the three real-life data sets introduced in the last section, we carried out two experiments to compare TW-k-means with five clustering algorithms, i.e., W-k-means, EW-k-means, LAC, EWKM and WCMM. The purpose of the first experiment was to select proper parameter values for comparing the clustering performance of six algorithms in the second experiment. In each experiment, the number of clusters for all clustering algorithms were set as the actual number of classes of the used data set.

In the first experiment, we set the parameter values of four clustering algorithms as 30 integers from 1 to 30 (β in W-k-means, η in EW-k-means, h in LAC, and γ in EWKM). For TW-k-means, we set η as 30 integers from 1 to 30 and λ as 12 values of $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120\}$. Since the clustering results of the five clustering algorithms excluding WCMM were affected by the initial cluster centers, we randomly generated 100 sets of initial cluster centers for each data set. For each parameter setting, we ran each of the five clustering algorithms to produce 100 clustering results on each of the three data sets. For WCMM, we set α as eight values $\{1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5\}$. Since WCMM can find global optima, we only ran WCMM once. In the second experiment, we first set the parameter values for six algorithms by selecting those with the best results in the first experiment. Similar to the first experiment, we produced 100 results for each of the five clustering algorithms excluding WCMM and 1 result for WCMM on each data set.

In order to compare the classification performance, we used precision, recall, f-measure and accuracy to evaluate the results [35]. Precision is calculated as the fraction of correct objects among those that the algorithm believes belonging to the relevant class. Recall is the fraction of actual objects that were identified. F-measure is the harmonic mean of precision and recall and accuracy is the proportion of correctly classified objects. All four indices use the corresponding actual classification as the reference classification.

To statistically compare the clustering performance, the paired t-test comparing TW-k-means with the other five



(a) Average accuracies of W-k-means, EW-k-means, LAC and EWKM.

(b) Average accuracies of TW-k-means.

(c) Average accuracies of WCMM.

Fig. 6. The clustering results of six clustering algorithms versus their parameter values on the *Multiple Features* data set.

clustering methods was computed from each of the four evaluation indices. If the p-value was below the threshold of the statistical significance level (usually 0.05), then the null hypothesis was rejected in favor of an alternative hypothesis, which typically states that the comparing two distributions do differ. Thus, if the p-value of two approaches was less than 0.05, the difference of the clustering results of the two approaches was considered to be significant, otherwise, insignificant.

6.3 Results and Analysis

As an example, Fig. 6 draws the average clustering accuracies of six clustering algorithms on the *Multiple Features* data set in the first experiment. From these results, we can observe that TW-k-means produced better results with large value of λ than the other five algorithms. When λ was large, it produced relatively stable results with the change of η. WCMM produced the worst results, which indicates that WCMM failed to recover the clusters from this high-dimensional multiview data. EWKM produced unstable and worse results than W-k-means, LAC and TW-k-means. EW-k-means produced similar results as W-k-means, which indicates that the regularization term affects the result not too much.

In the second experiment, we set the parameter values of six clustering algorithms as shown in Table 1. Table 2 summarizes the total 1,503 clustering results. From these results, we can see that TW-k-means significantly outperformed the other five algorithms in almost all results, especially on the *Multiple Features* and *Internet Advertisement* data sets. Although TW-k-means is an extension to EW-k-means, the introduction of weights on views improved its results. WCMM produced the worst results on all three data sets.

To sum up, TW-k-means is superior to the other five clustering algorithms in clustering multiview data.

TABLE 1
Parameter Values of Six Clustering Algorithms in the Experiments on the Three Real-Life Data Sets

Algorithms	MF	IA	IS
W-k-means (β)	8	10	30
EW-k-means (η)	20	1	30
LAC (h)	1	15	30
EWKM (λ)	5	40	30
WCMM (α)	1.5	4	1
TW-k-means ((λ,η))	(30,7)	(80,25)	(70,40)

MF: the Multiple Features data set, IA:the Internet Advertisement data set, IS: the Image Segmentation data set.

TABLE 2
Summary of Clustering Results on Three Real-Life Data Sets by Six Clustering Algorithms

Data	Evaluation indices	W-k-means	EW-k-means	LAC	EWKM	WCMM	TW-k-means
MF	Precision	$-0.06(.10)*$	$-0.07(.10)*$	$-0.07(.09)*$	$-0.24(.08)*$	$-0.59(.00)*$	$0.79(.09)$
	Recall	$-0.09(.09)*$	$-0.09(.09)*$	$-0.09(.08)*$	$-0.36(.12)*$	$-0.56(.00)*$	$0.82(.08)$
	F-measure	$-0.08(.10)*$	$-0.08(.10)*$	$-0.08(.08)*$	$-0.41(.12)*$	$-0.59(.00)*$	$0.80(.09)$
	Accuracy	$-0.09(.09)*$	$-0.09(.09)*$	$-0.09(.08)*$	$-0.36(.12)*$	$-0.56(.00)*$	$0.82(.08)$
IA	Precision	$-0.16(.19)*$	$-0.16(.20)*$	$-0.14(.20)*$	$-0.22(.19)*$	$-0.56(.00)*$	$0.72(.12)$
	Recall	$-0.14(.04)*$	$-0.10(.07)*$	$-0.10(.08)*$	$-0.13(.06)*$	$-0.33(.00)*$	$0.72(.07)$
	F-measure	$-0.23(.04)*$	$-0.17(.12)*$	$-0.17(.12)*$	$-0.21(.09)*$	$-0.47(.00)*$	$0.69(.11)$
	Accuracy	$-0.14(.04)*$	$-0.10(.07)*$	$-0.10(.08)*$	$-0.13(.06)*$	$-0.33(.00)*$	$0.72(.07)$
IS	Precision	$-0.03(.07)*$	$-0.04(.08)*$	$-0.03(.07)*$	$-0.03(.09)*$	$-0.37(.00)*$	$0.62(.09)$
	Recall	$-0.03(.05)*$	$-0.03(.03)*$	$-0.03(.05)*$	$-0.03(.05)*$	$-0.41(.00)*$	$0.64(.05)$
	F-measure	$-0.01(.07)*$	$-0.02(.05)*$	$-0.01(.07)*$	$-0.02(.07)*$	$-0.40(.00)*$	$0.60(.07)$
	Accuracy	$-0.03(.05)*$	$-0.03(.03)*$	$-0.03(.05)*$	$-0.03(.05)$	$-0.41(.00)*$	$0.64(.05)$

The value of the TW-k-means algorithm is the mean value of 100 results and the other values are the differences of the mean values between the corresponding algorithms and the TW-k-means algorithm. The value in brackets is the standard deviation of 100 results. "" indicates that the difference is significant.*

6.4 Scalability Comparison

We used all five real-life data sets to compare the scalability of TW-k-means with the other five clustering algorithms. Fig. 7 draws the average time costs of six clustering algorithms. We can see that the execution time of TW-k-means was only more than EW-k-means, and significantly less than the other four clustering algorithms. This result indicates that TW-k-means scales well to high-dimensional data.

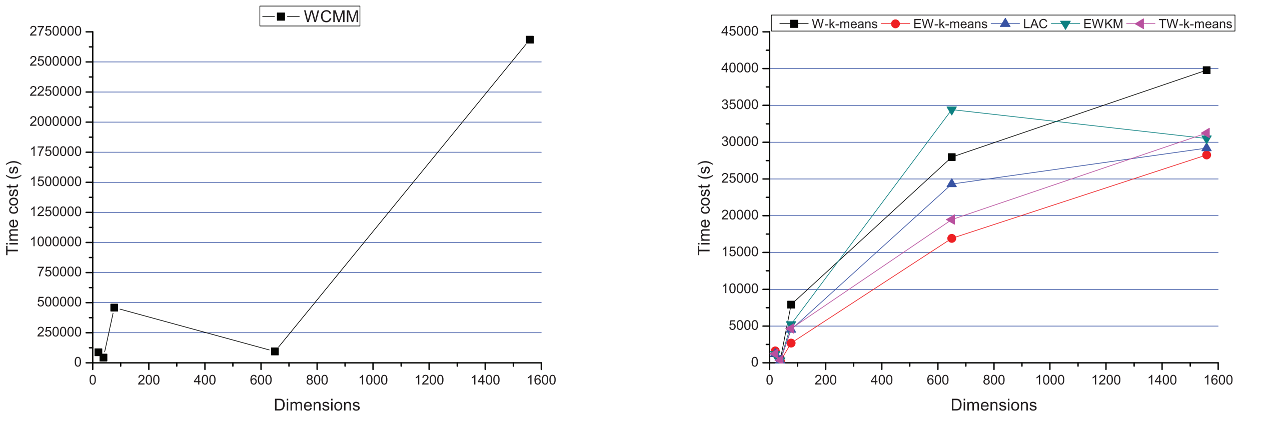
7 Conclusions

In this paper, we have presented TW-k-means, an innovative two-level variable weighting clustering algorithm for clustering of multiview data. Given multiple-view data, TW-k-means can compute weights for views and individual variables simultaneously in the clustering process. With the two types of weights, compact views and important variables can be identified and effect of low-quality views and noise variables can be reduced. Therefore, TW-k-means can obtain better clustering results than individual variable weighting clustering algorithms from multiview data. We used two real-life data sets to investigate the properties of two types of weights in TW-k-means. We discussed the difference of the weights between TW-k-means and EW-k-means algorithms. The experiments also revealed the convergence property of the view weights in TW-k-means. We compared TW-k-means with five clustering algorithms on three real-life data sets and the results have shown that the TW-k-means algorithm significantly outperformed the other five clustering algorithms in four evaluation indices. As such, it is a new variable weighting method for clustering of multiview data.

In the future, we will combine the two-level variable weighting method with other techniques such as fuzzy techniques, subspace clustering techniques, semi-supervised techniques etc. so as to apply our method to more applications. Moreover, we will investigate approaches that can automatically group variables in the clustering process.

Acknowledgments

This research is supported in part by NSFC under Grant no. 61073195, and Shenzhen New Industry Development Fund under Grant nos. CXB201005250024A and CXB201005250021A.



(a) Average time cost of WCMM. (b) Average time costs of the other five clustering algorithms.

Fig. 7. Average time costs of six clustering algorithms on five real-life data sets.

REFERENCES

[1] J. Mui and K. Fu, "Automated Classification of Nucleated Blood Cells Using a Binary Tree Classifier," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 2, no. 5, pp. 429-443, May 1980.

[2] J. Wang, H. Zeng, Z. Chen, H. Lu, L. Tao, and W. Ma, "ReCoM: Reinforcement Clustering of multiType Interrelated Data Objects," *Proc. 26th Ann. Int'l ACM SIGIR Conf. Research and Development in Informaion Retrieval,* pp. 274-281, 2003.

[3] S. Bickel and T. Scheffer, "Multi-view Clustering," *Proc. IEEE Fourth Int'l Conf. Data Mining,* pp. 19-26, 2004.

[4] K. Kailing, H. Kriegel, A. Pryakhin, and M. Schubert, "Clustering Multi-Represented Objects with Noise," *Proc. Eighth Pacific-Asia Conf. Knowledge Discovery and Data Mining,* H. Dai, R. Srikant, and C. Zhang, eds., vol. 3056, pp. 394-403, Springer Berlin/Heidelberg, 2004.

[5] V.R. de Sa, "Spectral Clustering with Two Views," *Proc. IEEE 22nd Int'l Workshop Learning with Multiple Views (ICML),* pp. 20-27, 2005.

[6] D. Zhou and C. Burges, "Spectral Clustering and Transductive Learning with Multiple Views," *Proc. 24th Int'l Conf. Machine Learning,* pp. 1159-1166, 2007.

[7] M.B. Blaschko and C.H. Lampert, "Correlational Spectral Clustering," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '08),* pp. 1-8, 2008.

[8] K. Chaudhuri, S. Kakade, K. Livescu, and K. Sridharan, "Multi-view Clustering via Canonical Correlation Analysis," *Proc. 26th Ann. Int'l Conf. Machine Learning,* pp. 129-136, 2009.

[9] G. Tzortzis and C. Likas, "Multiple View Clustering Using a Weighted Combination of Exemplar-Based Mixture Models," *IEEE Trans. Neural Networks,* vol. 21, no. 12, pp. 1925-1938, Dec. 2010.

[10] B. Long, P. Yu, and Z. Zhang, "A General Model for Multiple View Unsupervised Learning," *Proc. Eighth SIAM Int'l Conf. Data Mining (SDM '08),* 2008.

[11] D. Greene and P. Cunningham, "A Matrix Factorization Approach for Integrating Multiple Data Views," *Proc. European Conf. Machine Learning and Knowledge Discovery in Databases,* pp. 423-438, 2009.

[12] R. Gnanadesikan, J. Kettenring, and S. Tsao, "Weighting and Selection of Variables for Cluster Analysis," *J. Classification,* vol. 12, pp. 113-136, 1995.

[13] G. De Soete, "Optimal Variable Weighting for Ultrametric and Additive Tree Clustering," *Quality and Quantity,* vol. 20, pp. 169-180, 1986.

[14] G. De Soete, "OVWTRE: A Program for Optimal Variable Weighting for Ultrametric and Additive Tree Fitting," *J. Classification,* vol. 5, no. 1, pp. 101-104, 1988.

[15] E. Fowlkes, R. Gnanadesikan, and J. Kettenring, "Variable Selection in Clustering," *J. Classification,* vol. 5, pp. 205-228, 1988.

[16] V. Makarenkov and B. Leclerc, "An Algorithm for the Fitting of a Tree Metric According to a Weighted Least-Squares Criterion," *J. Classification,* vol. 16, pp. 3-26, 1999.

[17] V. Makarenkov and P. Legendre, "Optimal Variable Weighting for Ultrametric and Additive Trees and k-Means Partitioning: Methods and Software," *J. Classification,* vol. 18, no. 2, pp. 245-271, 2001.

[18] D. Modha and W. Spangler, "Feature Weighting in k-Means Clustering," *Machine Learning,* vol. 52, no. 3, pp. 217-237, 2003.

[19] J. Friedman and J. Meulman, "Clustering Objects on Subsets of Attributes," *J. Royal Statistical Soc.. Series B (Statistical Methodology),* vol. 66, no. 4, pp. 815-849, 2004.

[20] Z. Huang, M. Ng, H. Rong, and Z. Li, "Automated Variable Weighting in k-Means Type Clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 5, pp. 657-668, May 2005.

[21] C. Domeniconi, D. Gunopulos, S. Ma, B. Yan, M. Al-Razgan, and D. Papadopoulos, "Locally Adaptive Metrics for Clustering High Dimensional Data," *Data Mining and Knowledge Discovery,* vol. 14, no. 1, pp. 63-97, 2007.

[22] L. Jing, M. Ng, and Z. Huang, "An Entropy Weighting k-Means Algorithm for Subspace Clustering of High-Dimensional Sparse Data," *IEEE Trans. Knowledge and Data Eng.,* vol. 19, no. 8, pp. 1026-1041, 2007.

[23] P. Hoff, "Model-Based Subspace Clustering," *Bayesian Analysis,* vol. 1, no. 2, pp. 321-344, 2006.

[24] C. Bouveyron, S. Girard, and C. Schmid, "High Dimensional Data Clustering," *Computational Statistics and Data Analysis,* vol. 52, no. 1, pp. 502-519, 2007.

[25] C.-Y. Tsai and C.-C. Chiu, "Developing a Feature Weight Self-Adjustment Mechanism for a k-Means Clustering Algorithm," *Computational Statistics and Data Analysis,* vol. 52, no. 10, pp. 4658-4672, 2008.

[26] Z. Deng, K. Choi, F. Chung, and S. Wang, "Enhanced Soft Subspace Clustering Integrating Within-Cluster and Between-Cluster Information," *Pattern Recognition,* vol. 43, no. 3, pp. 767-781, 2010.

[27] H. Cheng, K.A. Hua, and K. Vu, "Constrained Locally Weighted Clustering," *Proc. VLDB Endowment,* vol. 1, pp. 90-101, Aug. 2008.

[28] W. DeSarbo, J. Carroll, L. Clark, and P. Green, "Synthesized Clustering: A Method for Amalgamating Clustering Bases with Differential Weighting Variables," *Psychometrika,* vol. 49, no. 1, pp. 57-78, 1984.

[29] P. Green, J. Kim, and F. Carmone, "A Preliminary Study of Optimal Variable Weighting in K-Means Clustering," *J. Classification,* vol. 7, no. 2, pp. 271-285, 1990.

[30] D. Lashkari and P. Golland, "Convex Clustering with Exemplar-Based Models," *Advances in Neural Information Processing Systems,* vol. 20, pp. 825-832, 2008.

[31] Z. Huang, "Extensions to the k-Means Algorithms for Clustering Large Data Sets with Categorical Values," *Data Ming and Knowledge Discovery,* vol. 2, no. 3, pp. 283-304, 1998.

[32] A. Frank and A. Asuncion, "UCI Machine Learning Repository," http://archive.ics.uci.edu/ml, 2010.

[33] P. Spellman, G. Sherlock, M. Zhang, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein, and B. Futcher, "Comprehensive Identification of Cell Cycle-Regulated Genes of the Yeast Saccharomyces Cerevisiae by Microarray Hybridization," *Molecular Biology of the Cell,* vol. 9, no. 12, pp. 3273-3297, 1998.

[34] N. Kushmerick, "Learning to Remove Internet Advertisements," *Proc. Third Ann. Conf. Autonomous Agents,* pp. 175-181, 1999.

[35] Wikipedia, "Plagiarism—Wikipedia, the Free Encyclopedia," http://en.wikipedia.org/wiki/Information_retrieval, 2011.

Xiaojun Chen received the PhD degree from Harbin Institute of Technology in 2011. He is now an assistant professor at Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. His research interests include subspace clustering, topic model, high-performance data mining, and business intelligence.

Xiaofei Xu received the BS, MS, and PhD degree in the Department of Computer Science and Engineering at Harbin Institute of Technology (HIT) in 1982, 1985, and 1988, respectively. He is now the assistant to the president of HIT. His research interests include enterprise computing, service computing and service engineering, enterprise interoperability, enterprise modeling, ERP and supply chain management systems, databases and data mining, knowledge management software engineering.

Joshua Zhexue Huang received the PhD degree from the Royal Institute of Technology in Sweden. He is now a professor and chief scientist at Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, and a honorary professor at the Department of Mathematics, The University of Hong Kong. His research interests include data mining, machine learning, and clustering algorithms.

Yunming Ye received the PhD degree in computer science from Shanghai Jiao Tong University. He is now a professor in the Shenzhen Graduate School, Harbin Institute of Technology. His research interests include data mining, text mining, and clustering algorithm.