Pattern Recognition 53 (2016) 25–35

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Weighted Multi-view Clustering with Feature Selection

Yu-Meng Xu [a], Chang-Dong Wang [b,*], Jian-Huang Lai [a]

[a] School of Information Science and Technology, Sun Yat-sen University, Guangzhou, PR China
[b] School of Mobile Information Engineering, Sun Yat-sen University, Zhuhai, PR China

ARTICLE INFO

Article history:
Received 12 November 2014
Received in revised form
24 September 2015
Accepted 16 December 2015
Available online 23 December 2015

Keywords:
Data clustering
Multi-view
Feature selection
Weighting

ABSTRACT

In recent years, combining multiple sources or views of datasets for data clustering has been a popular practice for improving clustering accuracy. As different views are different representations of the same set of instances, we can simultaneously use information from multiple views to improve the clustering results generated by the limited information from a single view. Previous studies mainly focus on the relationships between distinct data views, which would get some improvement over the single-view clustering. However, in the case of high-dimensional data, where each view of data is of high dimensionality, feature selection is also a necessity for further improving the clustering results. To overcome this problem, this paper proposes a novel algorithm termed Weighted Multi-view Clustering with Feature Selection (WMCFS) that can simultaneously perform multi-view data clustering and feature selection. Two weighting schemes are designed that respectively weight the views of data points and feature representations in each view, such that the best view and the most representative feature space in each view can be selected for clustering. Experimental results conducted on real-world datasets have validated the effectiveness of the proposed method.

1. Introduction

Clustering is one of the most important methods to explore the underlying (cluster) structure of data [1]. The basic idea is to partition a set of data objects according to some criterion such that similar objects can be grouped into the same cluster, and dissimilar objects are separated into different clusters. To achieve this goal, we usually conduct clustering by maximizing the intra-cluster similarity and the inter-cluster dissimilarity. After several decades' development, a number of clustering algorithms have been developed [1], such as k-means clustering [2], spectral clustering [3], kernel-based clustering [4], graph-based clustering [5] and hierarchical clustering [6].

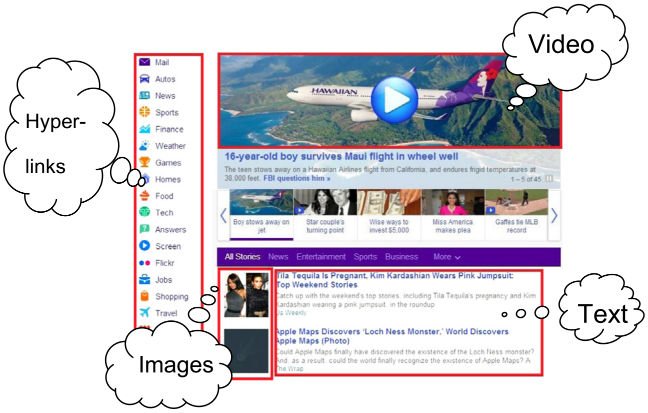
With the development of hardware technology, a huge amount of multi-view data with various representations have been generated in real-world applications [7–14]. For example, in web clustering, different types of data, such as images, videos, hyperlinks and texts, can be taken into consideration as they are different views of web pages (as shown in Fig. 1). In multi-view data, different views are different representations of the same set of instances. It is a significant research challenge to combine together multiple views or sources of the same set of instances to get a better clustering performance. The existing clustering algorithms designed for single-source data cannot be applied directly to the data consisting of multiple views or in various representations as they often vary greatly from traditional single-source data. Data in different views or sources are always not comparable to each other due to their dimensions and semantic representations are always different.

In addition, some views of data may be of high dimensionality which leads to high computational complexity and possibly low clustering accuracy. For example, when it comes to biomedicine, we can get different types of information for a patient, including magnetic resonance images, cerebrospinal fluid test data, blood test data, protein expression data, and genetic data, each of which is taken as a distinct view of patient data. However, some view of data may be of high dimensionality which would lead to a large amount of calculation. For some specific views, only a portion of features are needed for improving the clustering results. In other words, feature selection is a way which can both simplify the calculation and help to get an accurate data model in data clustering [15,13,16].

In order to solve this problem, we propose a novel algorithm, termed Weighted Multi-view Clustering with Feature Selection (WMCFS), which can simultaneously perform multi-view data clustering and feature selection. A global objective function is proposed, which takes into consideration both of the multi-view learning and feature selection in the process of data clustering. In the global objective function, two weighting schemes are designed

[*] Corresponding author. Tel.: +86 20 84110175.
E-mail addresses: yumengxu@hotmail.com (Y.-M. Xu), changdongwang@hotmail.com (C.-D. Wang), stsljh@mail.sysu.edu.cn (J.-H. Lai).

http://dx.doi.org/10.1016/j.patcog.2015.12.007
0031-3203/© 2015 Elsevier Ltd. All rights reserved.



16-year-old boy survives Maui flight in wheel well

The teen stows away on a Hawaiian Airlines flight from California, and endures frigid temperatures at 38,000 feet. FBI questions him »

Boy stows away on jet

Star couple's turning point

Wise ways to invest $5,000

Miss America makes plea

Gaffes tie MLB record

Tila Tequila Is Pregnant, Kim Kardashian Wears Pink Jumpsuit: Top Weekend Stories

Catch up with the weekend's top stories, including Tila Tequila's pregnancy and Kim Kardashian wearing a pink jumpsuit, in the roundup

Us Weekly

Apple Maps Discovers 'Loch Ness Monster,' World Discovers Apple Maps (Photo)

Could Apple Maps finally have discovered the existence of the Loch Ness monster? And, as a result, could the world finally recognize the existence of Apple Maps? A

The Wrap

Fig. 1. Multi-view data of web page.

that respectively weight the views of data points and feature representations in each view, such that the best view and the most representative feature space in each view can be selected for clustering. To solve the objective function, we design an EM (Expectation Maximization)-like iteration, which can converge to the acceptable clustering results. Experimental results conducted on real-world datasets have validated the effectiveness of the proposed method.

The rest of the paper is organized as follows. Section 2 briefly overviews the previous work on multi-view data clustering. The proposed WMCFS algorithm and its foundations are described in detail in Section 3. To demonstrate the performance of our algorithms, we have conducted extensive experiments, the experimental results of which are reported in Section 4. The conclusion is drawn in Section 5.

2. Related work

For clustering multi-view or multi-source datasets, some algorithms have been proposed recently which take different factors into consideration, e.g. the differences and relationships between data from various views. Most of the earlier methods extend the traditional single-source clustering algorithms to the multi-view situation by simply minimizing the disagreement between different views, i.e., by minimizing the difference of the clustering results generated from different views. Two early works [17,18] developed two-view algorithms by combining EM, *k*-means and spectral clustering algorithms simultaneously. In [19,20], Kumar et al. used the spectral embedding from one view to conduct clustering of the other views which enforces the clustering results in different views to agree with each other. Wang et al. designed a multi-view spectral clustering, which relies on Pareto optimization to find the best common cuts across all views [21]. However, the above methods only focus on the relationships between various views and ignore the characteristics of distinct views in data. Tzortzis and Likas [22] proposed an multi-view kernel *k*-means (MVKKM) algorithm which assigns a weight for each view according to the view's contribution to the clustering result and then combines the kernels derived from the weighted views together. However, it is based on the inner product kernels for all views, and has no explicit mechanism for feature selection.

To address the above issues, there are some other efforts that investigate feature selection in multi-view data clustering. A framework was proposed in [14], which constructs models respectively for the multi-source learning and feature selection. However, this work is designed for supervised learning and cannot deal with the unsupervised situation. In particular, a model is first trained based on the supervision information, during which the relatively more important features for each cluster can be selected. In this way, feature selection can be accomplished under the criterion to enforce the correct class labels and the important features discovered by this process will be assigned with high weights. However, when it comes to the unsupervised situation, where the labeled samples are not available, this method is no longer applicable, since the importance of features cannot be evaluated due to the lack of the ground-truth labeling. Similarly, Zhao et al. [23] proposed an algorithm combining LDA with co-training, i.e., exploiting labels learned in one view to learn discriminative features in another view. In [24], Wang et al. developed an algorithm to do feature learning for multi-view clustering. However, this method cannot deal with the noisy data in each view. When some of the views are noisy, the result might become unsatisfactory. Chen et al. [25] proposed an automated two-level variable weighting clustering algorithm for multi-view data termed TW-*k*-Means, which can simultaneously compute weights for views and individual variables. However, the same weighting scheme is used for both view weighting and feature selection, which is not able to explore more possibilities. Cai et al. [26] also focused on multi-view clustering based on *k*-means which would be applicable for multi-view data but did not really do feature selection so that their clustering model will degenerate in the case of high dimensionality.

In this paper, inspired by the multi-view kernel *k*-means algorithm proposed by Tzortzis and Likas [22], we design an algorithm termed Weighted Multi-view Clustering with Feature Selection (WMCFS), that can simultaneously perform multi-view data clustering and feature selection. Instead of integrating a feature selection mechanism into multi-view kernel *k*-means, we use a simple yet effective formula based on the original *k*-means algorithm. This is because multi-view kernel *k*-means relies on a kernel mapping in which the kernel selection itself is a challenging issue in the unsupervised learning case.

3. Weighted Multi-view Clustering with Feature Selection

To make this paper clear, Table 1 summarizes the symbols used in this paper.

3.1. Problem formulation

Consider a dataset consisting of N instances represented by V views. Let $\mathcal{X} = \{\mathbf{x}_1^1, \mathbf{x}_2^1, \ldots, \mathbf{x}_N^V\}$ denote the dataset, where $\mathbf{x}_i^v$ is the i-th instance from the v-th view. In this way, the multi-view data

Table 1
Symbols used in this paper.

Symbol	Meaning
$\mathcal{X}$	The whole dataset
X^v	The v-th view dataset
$\mathbf{x}_i^v$	The i-th instance of the v-th view dataset and $\mathbf{x}_i^v \in \mathbb{R}^{d^v}$
$\mathbf{m}_k^v$	The cluster center of the k-th cluster in the v-th view
N	Number of instances in each view
M	Number of clusters
l^v	Number of features in the v-th view
ε_H	Objective function which denotes the sum of intra-class distances
ω_v	Weight for the v-th view
τ_l^v	Weight for the l-th feature of the v-th view
δ_{ik}	Indicator variable showing whether the i-th instance belongs to the k-th cluster
p	Exponential parameter controlling the sparsity of view weight vector
β	Parameter controlling the sparsity of feature weight vectors

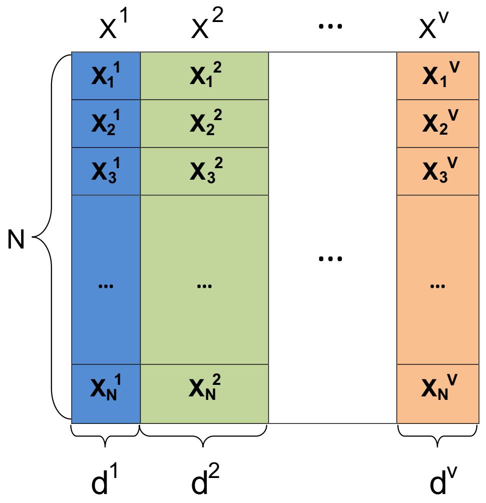


Fig. 2. Illustration of multi-view data. The same color represents data from the same view and the width of each column denotes the dimension of the corresponding view. It is often the case that the dimensions would vary in different views.

can be represented as follows (shown in Fig. 2):

$$\mathcal{X} = \{\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^V\}, \tag{1}$$

$$\mathbf{X}^1 = \{\mathbf{x}_1^1, \mathbf{x}_2^1, \ldots, \mathbf{x}_N^1\}, \tag{2}$$

$$\vdots$$

$$\mathbf{X}^V = \{\mathbf{x}_1^V, \mathbf{x}_2^V, \ldots, \mathbf{x}_N^V\}, \tag{3}$$

where $\mathbf{X}^v$ denotes the set of instances from the v-th view in $\mathcal{X}$, and $\mathbf{x}_i^v \in \mathbb{R}^{d^v}$ is the i-th instance in $\mathbf{X}^v$ with d^v being the dimension of the v-th view.

The goal of multi-view clustering is to cluster the N instances into M clusters according to their semantic similarity between each other in all views.

3.2. Global objective function

In what follows, we will describe in detail the proposed global objective function, which takes into consideration both of the multi-view learning and the feature selection in the process of data clustering. Both of the multi-view learning and the feature selection are realized by weighting. To this end, two weighting schemes are designed, which respectively weight the views of data points and feature representation in each view, such that the best view and the most representative feature subspace in each view can be selected for clustering. For clarity, we will describe the objective function from the viewpoint of multi-view k-means, followed by view weighting and feature selection.

3.2.1. Multi-view k-means

We will describe our objective function from the viewpoint of k-means for clarity. The goal of k-means is to choose M cluster centers such that the sum of the squared distance of each instance to the corresponding cluster center is minimized. Let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ denote the dataset, the objective function of k-means to be minimized is as follows:

$$\varepsilon_H = \sum_{i=1}^{N} \sum_{k=1}^{M} \delta_{ik} \| (\mathbf{x}_i - \mathbf{m}_k) \|^2, \tag{4}$$

where $\mathbf{m}_k$ is the center of cluster k and δ_{ik} denotes the cluster assignment of instances such that δ_{ik} equals 1 when the i-th instance is assigned to cluster k and 0 otherwise. Obviously, each instance must be assigned to one and only one cluster, i.e., $\sum_{k=1}^{M} \delta_{ik} = 1, \forall i, \delta_{ik} \in \{0, 1\}$.

In the case of multi-view data, the objective function becomes

$$\varepsilon_H = \sum_{v=1}^{V} \sum_{i=1}^{N} \sum_{k=1}^{M} \delta_{ik} \| (\mathbf{x}_i^v - \mathbf{m}_k^v) \|^2, \tag{5}$$

where $\mathbf{m}_k^v$ is the center of cluster k in view v. In the multi-view k-means objective function, each instance is assigned to the same cluster in all views, but the cluster centers of the same cluster vary in different views. This is because the data representations in distinct views are different, which leads to different cluster center representations.

3.2.2. View weighting and feature selection

To simultaneously perform multi-view learning and feature selection in the process of data clustering, two weighting schemes are designed that respectively weight the views of data points and feature representation in each view.

The first weighting scheme is to weight the data of each view. Let ω_v denote the weight for data from the v-th view, satisfying $\sum_{v=1}^{V} \omega_v = 1, \omega_v \geq 0$. Therefore, ω is the view weight vector. The second weighting scheme is to weight the features of each view. Let τ^v denote the feature weight vector of length d^v, with each entry $\tau_l{}^v$ representing the weight for the l-th feature in view v, satisfying $\sum_{l=1}^{d^v} \tau_l^v = 1, \tau_l^v \geq 0$.

Based on the above notation, we get the sum of the weighted squared distance with regularization term as follows:

$$\varepsilon_H = \sum_{v=1}^{V} (\omega_v)^p \sum_{i=1}^{N} \sum_{k=1}^{M} \delta_{ik} \| diag(\tau^v)(\mathbf{x}_i^v - \mathbf{m}_k^v) \|^2 + \beta \sum_{v=1}^{V} \| \tau^v \|^2, \tag{6}$$

where $\mathbf{m}_k^v$ is the center of cluster k in view v:

$$\mathbf{m}_k^v = \frac{\sum_{i=1}^{N} \delta_{ik} \mathbf{x}_i^v}{\sum_{i=1}^{N} \delta_{ik}}, \tag{7}$$

and $diag(\tau^v)$ is a diagonal matrix with the elements of the vector τ^v on the diagonal.

The last component of this objective function $\beta \sum_{v=1}^{V} \| \tau^v \|^2$ is used to control the sparsity of the feature weight vectors $\tau^v, \forall v$ so as to avoid the situation that only a few features are selected in getting a very small but meaningless objective value. The parameters p and β are the exponential and balancing parameters, which are selected according to the priori knowledge of data so as to help controlling the sparsity of the view weight vector ω and the feature weight vectors $\tau^v, \forall v = 1, \ldots, V$ respectively. Experimental analysis shows that there exists a relatively wide range of values that can generate satisfactory clustering results.

The major difference between the proposed objective function (6) and the one proposed by Tzortzis and Likas [22] is that, the objective in [22] only considers view weighting and there is no strategy for feature learning, as shown below:

$$\varepsilon_H = \sum_{v=1}^{V} (\omega_v)^p \sum_{i=1}^{N} \sum_{k=1}^{M} \delta_{ik} \| \phi^v(\mathbf{x}_i^v) - \mathbf{m}_k^v \|^2, \tag{8}$$

where $\mathbf{m}_k^v$ is the center of cluster k in view v:

$$\mathbf{m}_k^v = \frac{\sum_{i=1}^{N} \delta_{ik} \phi^v(\mathbf{x}_i^v)}{\sum_{i=1}^{N} \delta_{ik}}. \tag{9}$$

The feature in each view is pre-specified by a kernel mapping ϕ as input to their algorithm. However, in our method, not only view weighting but also feature weighting are automatically learned. Although it is possible to integrate the weighting-based feature selection mechanism into (8), the kernel selection itself is a challenging issue in the unsupervised learning case. Therefore, a simple yet effective formula is used in our model.

3.2.3. *Final objective function*

The goal of the Weighted Multi-view Clustering with Feature Selection is to find the optimal cluster assignment, view weighting and feature weighting simultaneously such that the objective function is minimized. That is

$$\min_{\{\delta_{ik}\}_{k=1}^{M},\{\omega_v\}_{v=1}^{V},\{\tau^v\}_{v=1}^{V}} \varepsilon_H,$$

$$\text{subject to } \sum_{k=1}^{M}\delta_{ik}=1, \quad \forall i, \delta_{ik}\in\{0,1\},$$

$$\sum_{v=1}^{V}\omega_v=1, \omega_v\geq 0,$$

$$\sum_{l=1}^{d^v}\tau_l^v=1, \tau_l^v\geq 0, \quad \forall v. \tag{10}$$

3.3. *Optimization*

To search for the optimal cluster assignment, view weighting and feature weighting, we design an EM-like iteration, which contains three iteration stages. In each stage, one of the three variables is updated, with the other two variables being fixed. We will describe the three stages one by one in the following sections.

3.3.1. *Updating the cluster assignment*

By fixing the view weight vector and feature weight vectors, we can update the cluster assignment δ_{ik} by performing k-means. That is, in the objective function (6), fixing ω and $\tau^v, \forall v$ results in the objective function the same as k-means.

In the first round of iteration, the view weight vector ω and feature weight vectors $\tau^v, \forall v = 1, \ldots, V$ are respectively initialized evenly as $\omega_v = \frac{1}{V}, \forall v$ and $\tau_l^v = \frac{1}{d^v}, \forall l$.

Then we can apply k-means by minimizing ε_H in (6).

For convenience, some notations are introduced here. Let

$$G=\sum_{v=1}^{V}\omega_v^p G_v, \tag{11}$$

where

$$G_v(i,j)=(\mathbf{x}_i^v)^T diag(\tau^v) diag(\tau^v)\mathbf{x}_j^v, \tag{12}$$

which can be written as

$$G_v(i,j)=(\mathbf{y}_i^v)^T\mathbf{y}_j^v, \tag{13}$$

where $\mathbf{y}_i^v$ is the projection of instance $\mathbf{x}_i^v$ after feature selection (i.e., weighted by τ^v).

By substituting (7) into (6), we can get

$$\varepsilon_H=\sum_{v=1}^{V}\omega_v^p\sum_{i=1}^{N}\sum_{k=1}^{M}\delta_{ik}\parallel diag(\tau^v)\left(\mathbf{x}_i^v-\frac{\sum_{i=1}^{N}\delta_{ik}\mathbf{x}_i^v}{\sum_{i=1}^{N}\delta_{ik}}\right)\parallel^2+\beta\sum_{v=1}^{V}\parallel\tau^v\parallel^2,$$

$$\Rightarrow \varepsilon_H=\mathrm{tr}(G)-\mathrm{tr}(\Delta^T G\Delta)+\beta\sum_{v=1}^{V}\parallel\tau^v\parallel^2, \tag{14}$$

where Δ is

$$\Delta_{ik}=\frac{\delta_{ik}}{\sqrt{\sum_{j=1}^{N}\delta_{jk}}}, \tag{15}$$

and the last regularization term is a constant given fixed $\tau^v, \forall v = 1, \ldots, V$.

In this way, we can take full advantage of the information from various views to improve the clustering results.

3.3.2. *Updating the view weighting*

In this stage of iteration, by fixing the cluster assignment δ_{ik} and the feature weight vectors $\tau^v, \forall v$, we can update the view weight vector ω as follows.

First, we get the Lagrangian formula of (10) w.r.t. ω_v as follows:

$$L(\omega,\lambda)=\varepsilon_H(\omega)+\lambda\left(\sum_{v=1}^{V}\omega_v-1\right). \tag{16}$$

Taking derivative of both sides w.r.t. ω_v gives

$$\frac{\partial L(\omega,\lambda)}{\partial\omega_v}=\frac{\partial\varepsilon_H(\omega)}{\partial\omega_v}+\lambda. \tag{17}$$

Setting the derivation to zero, we can get

$$p\omega_v^{(p-1)}D_v+\lambda=0 \Rightarrow \omega_v=\left(\frac{-\lambda}{pD_v}\right)^{1/(p-1)}, \tag{18}$$

where

$$D_v=\sum_{i=1}^{N}\sum_{k=1}^{M}\delta_{ik}\parallel diag(\tau^v)(\mathbf{x}_i^v-\mathbf{m}_k^v)\parallel^2, \tag{19}$$

when $p > 1$.

Furthermore, we can get the following formula by substituting (18) into the constraint $\sum_{v'=1}^{V}\omega_{v'}=1$:

$$\sum_{v'=1}^{V}\left(\frac{-\lambda}{pD_{v'}}\right)1/(p-1)=1 \Rightarrow (-\lambda)^{1/(p-1)}=\frac{1}{\sum_{v'=1}^{V}\left(\frac{1}{pD_{v'}}\right)^{1/(p-1)}}. \tag{20}$$

Based on this, we can get the formula to update ω_v by substituting (20) into (18) as follows:

$$\omega_v=\frac{1}{\sum_{v'=1}^{V}\left(\frac{D_v}{D_{v'}}\right)^{1/(p-1)}}, \quad p>1. \tag{21}$$

When $p=1$, the weights are less than 1 according to $\sum_{v=1}^{V}\omega_v=1$, and we can get $D_{v*}\leq\sum_{v'=1}^{V}\omega_{v'}D_{v'}$, where $v*=\arg\min_{v'}D_{v'}$. Therefore, we can get the following formula for ω_v if $p=1$:

$$\omega_v=\begin{cases}1, & v=\arg\min_{v'} D_{v'}\\ 0, & \text{otherwise}\end{cases} \quad p=1. \tag{22}$$

In the above formulas, p is a exponential parameter that can help adjusting the sparsity of the view weight vector ω. If we get some priori knowledge of the input data, we can set p to a better value which can improve the result. That is, according to (21), if most of the views are useful, larger p is more suitable. Nevertheless, our experimental results show that there exists relatively a wide range of p values that can generate satisfactory clustering results.

The underlying rationale of the above updating formula is that the instances closer to the cluster centers are considered to be more useful. The more useful one view is, the larger weight this view will be assigned to.

3.3.3. *Updating the feature weighting*

In this stage of iteration, by fixing the cluster assignment δ_{ik} and the view weight vector ω, we can update the feature weight vectors $\tau^v, \forall v$ as follows.

Similar to the updating of the view weighting, the Lagrangian formula of (10) w.r.t. τ^v can be obtained as follows:

$$L(\tau,\lambda)=\varepsilon_H(\tau)+\lambda\left(\sum_{l=1}^{d^v}\tau_l^v-1\right), \quad \forall v. \tag{23}$$

Taking derivative of both sides w.r.t. τ_l^v gives

$$\frac{\partial L(\tau,\lambda)}{\partial \tau_l^v} = \frac{\partial \varepsilon_H(\tau)}{\partial \tau_l^v} + \lambda. \tag{24}$$

To obtain a simpler formula of the first term on the righthand side, we rewrite the formula of ε_H as follows:

$$\varepsilon_H(\tau) = \sum_{v=1}^{V}(\omega_v)^p \sum_{i=1}^{N}\sum_{k=1}^{M}\delta_{ik}\sum_{l=1}^{d^v}(\mathbf{x}_i^v - \mathbf{m}_k^v)_l^2(\tau_l^v)^2 + \beta\sum_{v=1}^{V}\sum_{l=1}^{d^v}(\tau_l^v)^2, \tag{25}$$

where $(\mathbf{x}_i^v - \mathbf{m}_k^v)_l$ means the l-th element of $(\mathbf{x}_i^v - \mathbf{m}_k^v)$. Based on the above formula, we can get

$$\frac{\partial \varepsilon_H(\tau)}{\partial \tau_l^v} = 2(\omega_v)^p \sum_{i=1}^{N}\sum_{k=1}^{M}\delta_{ik}(\mathbf{x}_i^v - \mathbf{m}_k^v)_l^2\tau_l^v + 2\beta\tau_l^v. \tag{26}$$

By substituting (26) into (24) and setting (24) to 0, we can get

$$\tau_l^v = \frac{-\lambda}{2\beta + 2(\omega_v)^p \sum_{i=1}^{N}\sum_{k=1}^{M}\delta_{ik}(x_i^v - m_k^v)_l^2}. \tag{27}$$

Substituting (27) into the constraint $\sum_{l'=1}^{d^v}\tau_{l'}^v = 1$, we can get

$$-\lambda = \frac{1}{\sum_{l'=1}^{d^v}\frac{1}{2\beta + 2(\omega_v)^p \sum_{i=1}^{N}\sum_{k=1}^{M}\delta_{ik}(x_i^v - m_k^v)_{l'}^2}}. \tag{28}$$

At last, we can get the formula for updating the feature weight vectors $\tau^v, \forall v$ by substituting (28) into (27) as follows:

$$\tau_l^v = \frac{\frac{1}{\sum_{l'=1}^{d^v}\frac{1}{2\beta + 2(\omega_v)^p \sum_{i=1}^{N}\sum_{k=1}^{M}\delta_{ik}(x_i^v - m_k^v)_{l'}^2}}}{2\beta + 2(\omega_v)^p \sum_{i=1}^{N}\sum_{k=1}^{M}\delta_{ik}(x_i^v - m_k^v)_l^2}, \quad \forall l, \tag{29}$$

which can be further simplified as

$$\tau_l^v = \frac{1}{\sum_{l'=1}^{d^v}\frac{B_l^v}{B_{l'}^v}}, \quad \forall l, \tag{30}$$

where

$$B_l^v = \beta + (\omega_v)^p \sum_{i=1}^{N}\sum_{k=1}^{M}\delta_{ik}(x_i^v - m_k^v)_l^2. \tag{31}$$

3.4. The complete algorithm

Algorithm 1. Weighted Multi-view Clustering with Feature Selection.

```
Input: $\mathcal{X} = \{\mathbf{x}_1^1, \mathbf{x}_2^1, \ldots, \mathbf{x}_N^V\}, p, \beta, M, t_{max}$.
Output: $\delta_{ik}$: the cluster assignment; $\omega_v$: the view weighting; ${\tau_l}^v$: the feature weighting.
Initialize $\omega_v = \frac{1}{V}$ and $\tau_l^v = \frac{1}{d^v}, \forall l = 1, 2, \ldots, d^v, \forall v = 1, 2, \ldots, V$. $t = 0$.
Repeat
    Update the cluster assignment $\delta_{ik}$ by performing $k$-means w.r.t. (6).
    Update the view weight vector $\omega$ via (21) or (22).
    Update the feature weight vectors $\tau^v, \forall v$ via (30).
    $t = t + 1$.
Until Convergence or $t > t_{max}$
```

For clarity, Algorithm 1 summarizes the proposed Weighted Multi-view Clustering with Feature Selection (WMCFS) algorithm.

To prove the convergence of Algorithm 1, we set the changes of weighted sum of intra-class distances between the t-th and the $(t+1)$-th iterations as $CH(t)$ which can be computed as follows:

$$CH(t) = \varepsilon_H^t - \min_{\hat{\omega}}(\min_{\hat{\tau}}(\min_{\hat{\delta}}(\varepsilon_H^t))), \tag{32}$$

where $\hat{\omega}$, $\hat{\tau}$ and $\hat{\delta}$ are the updated parameters in the $(t+1)$-th iteration and ${\varepsilon_H}^t$ is the weighted sum of intra-class distances after the t-th iteration. Obviously, $CH(t) \geq 0$ which means that the sum of intra-class distances updated in each step of iterations is strictly decreasing. Algorithm 1 converges to a local minimum.

In our experiments, the iteration stops when the number of iterations reaches the maximum number of iterations t_{max} (we always set $t_{max} = 10$ in our experiments) or the iteration converges, i.e., when the sum of intra-class distances keeps stable. Here we set a threshold ($e = 0.00001$) and when the gap of the sum of intra-class distances between the two consecutive iterations is less than the threshold, we stop the iteration and output the final results.

4. Experimental results

In order to demonstrate the effectiveness of the proposed method, extensive experiments have been conducted on three real-world datasets. We first analyze the performance sensitivity to the two parameters p and β. Then, several state-of-the-art multi-view clustering methods have been performed and compared with the proposed method, which shows the significant improvement achieved by our method. For experimental purpose, we only perform the parameter analysis on two of the three datasets and then report all of the comparison results on the three datasets.

4.1. Datasets and experimental settings

Three real-world datasets are used in our experiments, namely Mfeat, Reuters and Corel.

The Multiple Features (abbr. Mfeat) dataset is a dataset consisting of handwritten digits (0–9) [27]. There are 6 views of each instance, namely, Fourier coefficients of the character shapes (following the UCI Machine Learning Repository website[1], we use abbreviation mfeat-fou), profile correlations (mfeat-fac), Karhunen-Love coefficients (mfeat-kar), pixel averages in 2 × 3 windows (mfeat-pix), Zernike moments (mfeat-zer) and 6 morphological features (mfeat-mor). Here we take the first five representative views of this dataset to form a five-view dataset. The detailed information of this dataset is shown in Table 2.

The Reuters RCV1/RCV2 Multilingual (abbr. Reuters) dataset is a dataset consisting of machine translated documents [28]. It has been widely used for evaluating the performances of multi-view learning algorithms. The dataset contains documents originally written in five different languages, namely English (EN), French (FR), German (GR), Italian (IT) and Spanish (SP). Each document, originally written in one language, is translated to the other four languages using the Portage system [29]. The documents are categorized into six different topics. The dataset is summarized in Table 3. More detail can be found on the dataset website.[2] In our experiments, we choose one language, namely English (EN), as the original language source and take the translated documents in the other four languages as the other four sources. This means that we conduct our experiments on the five-view dataset, with the views being EN, FR, GR, IT and SP respectively.

[1] http://archive.ics.uci.edu/ml/datasets/Multiple+Features

[2] http://multilingreuters.iit.nrc.ca/ReutersMultiLingualMultiView.htm

The Corel dataset is extracted from a Corel image collection [27], and we randomly get 2000 instances of 5 classes. Four sets of features are available based on the color histogram (abbr. Col-h), color histogram layout (abbr. Col-hl), color moments (abbr. Col-m), and co-occurrence texture (abbr. Coo-t). These features are treated as the 4 views of samples whose information is shown in Table 4.

All the experiments are conducted in MATLAB 2012a (7.14) 64-bit edition on a workstation (Windows 64 bit, 8 Intel 2.00 GHz processors, 16 GB of RAM).

For clustering performance evaluation, two widely used measurements, i.e. classification rate (CR) [30] and normalized mutual information (NMI) [31], are used based on the ground-truth labels of the instances. When computing the classification rate, each obtained category is firstly associated with the "ground-truth" category which accounts for the largest number of samples in the learned category. Then the classification rate (CR) are computed as the ratio of the number of correctly classified samples to the size of the dataset. That is

$$CR = \frac{\#\text{Correctly classified samples}}{\#\text{Samples in the dataset}}. \tag{33}$$

Given the clustering labels π of c clusters and the actual class labels θ of $\hat{c}$ classes, we build a confusion matrix where entry (i,j) defines the number $N_i^{(j)}$ of data points in cluster i and class j. Then NMI can be computed from the confusion matrix [31],

$$NMI = \frac{2\sum_{l=1}^{c}\sum_{h=1}^{\hat{c}} \frac{N_l^{(h)}}{N} \log \frac{N_l^{(h)} N}{\sum_{i=1}^{c} N_i^{(h)} \sum_{i=1}^{\hat{c}} N_l^{(i)}}}{H(\pi)+H(\theta)}, \tag{34}$$

where $H(\pi) = -\sum_{i=1}^{c} \frac{N_i}{N} \log \frac{N_i}{N}$ and $H(\theta) = -\sum_{j=1}^{\hat{c}} \frac{N^{(j)}}{N} \log \frac{N^{(j)}}{N}$ are the Shannon entropy of cluster labels π and class labels θ respectively, with N_i and $N^{(j)}$ denoting the number of data points in cluster i and class j. Obviously, a higher classification rate (CR) (also normalized mutual information (NMI)) indicates a more accurate clustering result.

Table 2
Detailed information of the Mfeat dataset.

View	# Samples	# Features	# Classes
mfeat-fou	2000	76	10
mfeat-fac	2000	216	10
mfeat-kar	2000	64	10
mfeat-pix	2000	240	10
mfeat-zer	2000	47	10

Table 3
Detailed information of the Reuters Dataset.

View	# Docs	# Words
EN	18,758	21,513
FR	26,648	24,839
GR	29,953	34,279
IT	12,342	11,547
SP	24,039	11,506
Topic	# Docs	per(%)
C15	18,816	16.84
CCAT	21,426	19.17
E21	13,701	12.26
ECAT	19,198	17.18
GCAT	19,178	17.16
M11	19,421	17.39

4.2. Parameter analysis

In this subsection, we demonstrate the performance of our method by using different parameters p and β. In the process of analyzing one of the two parameters, the other parameter is fixed.

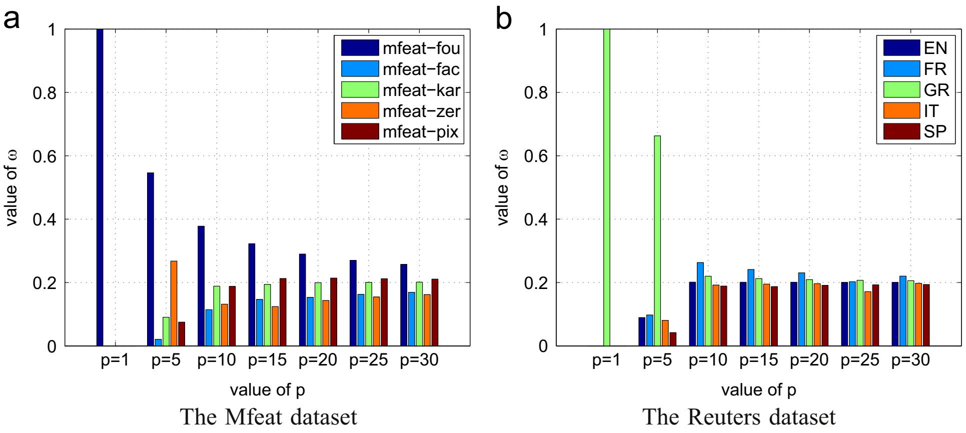
4.2.1. The exponential parameter p

The exponential parameter p is used to adjust the sparsity of the view weight vector ω, which would affect the performance of our method, i.e., different p will lead to different distribution of the view weight vector ω and hence different clustering results will be generated. To this end, the effect of the parameter p is analyzed from two perspectives, namely on the distribution of the view weight vector ω and on the final clustering results, i.e., classification rate (CR).

First, we analyze the effect of the parameter p on the distribution of the view weight vector ω. Fig. 3 shows the distribution of ω as a function of p on the Mfeat and Reuters datasets, by

Table 4
Detailed information of the Corel dataset.

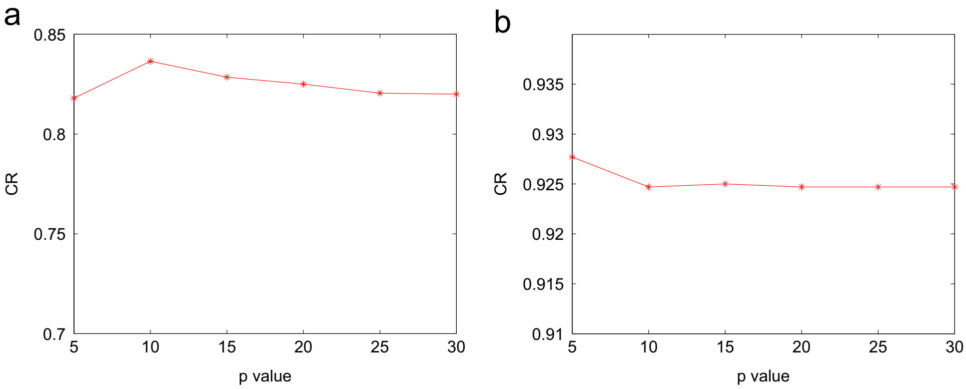
View	# Samples	# Features	# Classes
Col-h	2000	32	5
Col-hl	2000	32	5
Col-m	2000	9	5
Coo-t	2000	16	5



The Mfeat dataset

The Reuters dataset

Fig. 3. Analysis on the exponential parameter p: distribution of the view weight vector ω as a function of p.



The Mfeat dataset The Reuters dataset

Fig. 4. Analysis on the exponential parameter p: the clustering performance (in terms of CR) as a function of p.

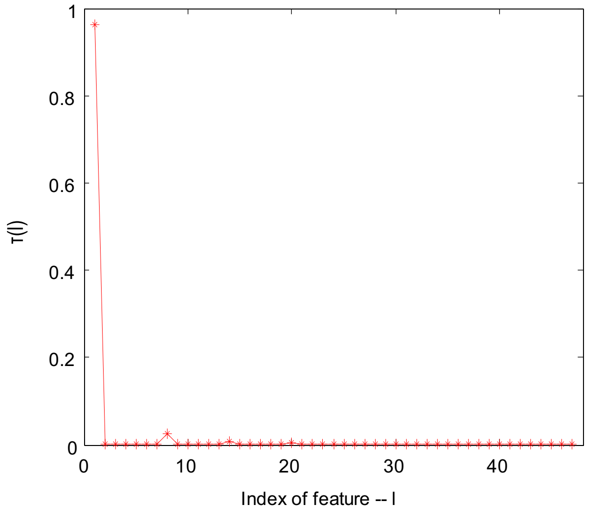


Fig. 5. Analysis on the balancing parameter β: distribution of the feature weight vector τ^v with $\beta=0$ in the mfeat-zer view on the Mfeat dataset.

ranging p from 1 to 30. Clearly, from (21), the smaller p is, the sparser the view weight vector ω will be. The results reported in Fig. 3 have confirmed this fact. If we know some priori knowledge about whether most of the data views are useful or not, we can select a relatively suitable p.

Secondly, we analyze the effect of the parameter p on the clustering performance in terms of classification rate (CR), as shown in Fig. 4. By fixing the other parameter β as 0.1 on both of the two datasets, we perform our method by using different p ranging from 5 to 30 (we do not start with $p=1$ here as we can see in Fig. 3 that setting $p=1$ degenerates into single view clustering). The clustering performance in terms of CR is plotted in Fig. 4 (a) and (b) respectively. From the figures, we can see that, the clustering performance is relatively stable in a wide range of p. For various datasets, a proper p can be set widely ranging from 5 to 30. We can figure out that the clustering results keep stable and are always better than those generated by the compared algorithms. In particular, the best clustering results can be obtained with $p=10$ on the Mfeat dataset and $p=5$ on the Reuters dataset.

4.2.2. The balancing parameter β

The balancing parameter β is used to control the sparsity of the feature weight vectors $\tau^v, \forall v$, which would also affect the performance of our method. To this end, the effect of the parameter β is also analyzed from two perspectives, namely on the distribution of the feature weight vectors $\tau^v, \forall v=1,\dots,V$ and on the final clustering results in terms of classification rate (CR).

To begin with, we first analyze the effect of parameter β on the sparsity of τ^v. Obviously, if the balancing parameter β is set to 0, there is no more regularization term for controlling the distribution of the feature weight vectors $\tau^v, \forall v=1,\dots,V$. In this case, in order to get a smaller objective function value during the iterations, only a small number of features will be used (with nonzero weight), as shown in Fig. 5, which plots the distribution of τ^v in one representative view of the Mfeat dataset. Therefore we need to set β to some proper value in order to balance the distribution of entries of τ^v as shown in Figs. 6 and 7. According to the experimental analysis, we suggest to choose β from $(0,1]$.

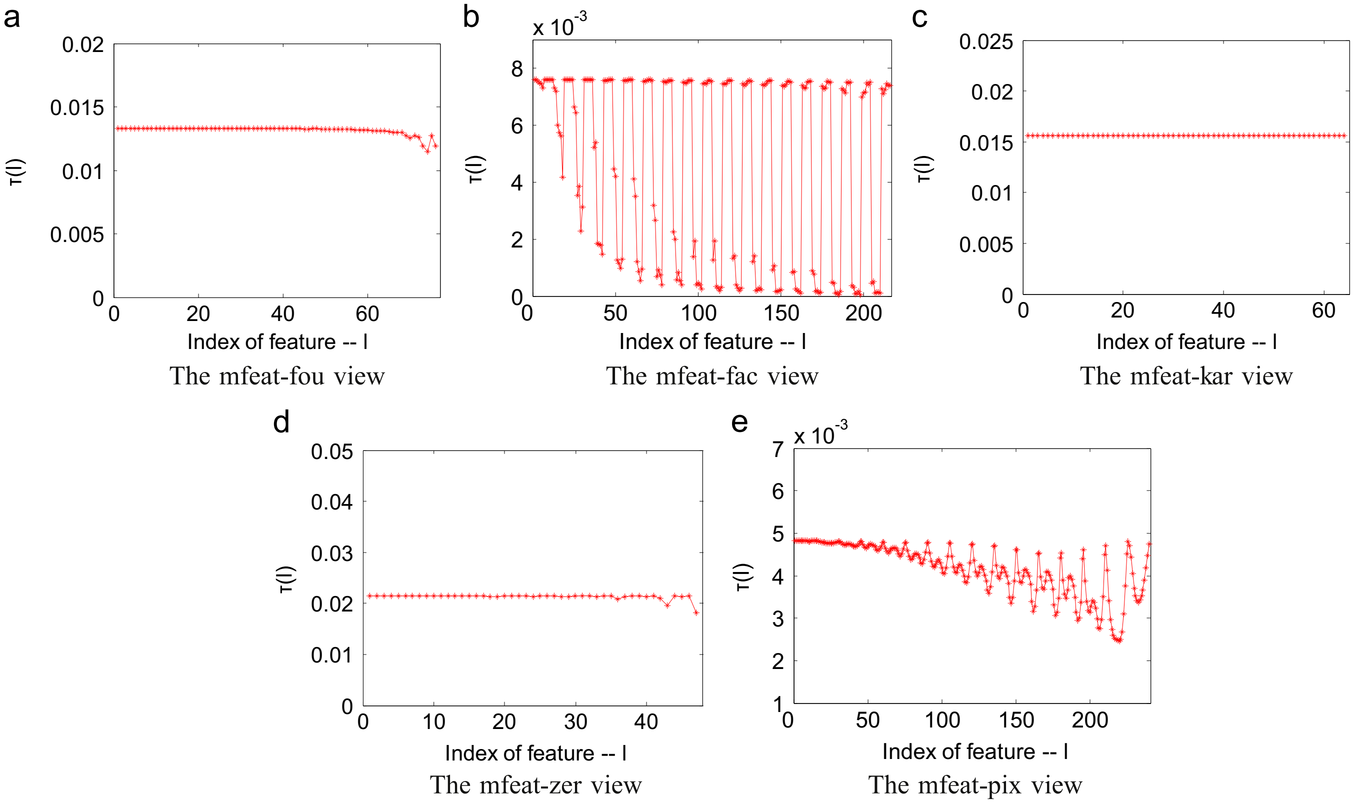
Additionally, we explore the effect of the parameter β on the clustering performance in terms of classification rate (CR). The results are reported in Fig. 8(a) and (b), with fixing the other parameter p as 10 and 5 on the two datasets respectively. When setting β in the range $[0.0005,1]$ and $[0.00005,1]$, the proposed method always generates the same clustering results on the Mfeat and Reuters datasets, i.e. the classification rate values 0.836 and 0.930. The only ranges where fluctuating clustering results are generated on Mfeat and Reuters are $[0,0.0005]$ and $[0,0.00005]$. Therefore we only plot the CR values as a function of β in these two ranges since it is meaningless to plot CR values when setting β in the range $[0.0005,1]$ and $[0.00005,1]$. Based on this analysis, we can get a wide range of β in which our method can generate very stable results. Therefore, based on the analysis in terms of both feature distribution and clustering performance, we suggest to set $\beta=0.1$ on all the datasets.

4.3. Comparison results

In this subsection, we compare the performance of the proposed method with some existing methods in terms of CR and NMI, namely traditional k-means [2], EM (Expectation Maximization) [32], MVKKM (multi-view kernel k-means) [22], Co-regspec (Co-regularized multi-view spectral clustering) [20] and LLC-fs (Local Learning-Based Clustering with Feature Selection) [33].

The optimal parameters analyzed in Section 4.2 are used in our WMCFS method. The parameter settings of the other five compared algorithms are summarized below:

1. *k-means*: We use the traditional k-means algorithm [2] as one of the compared algorithms. In our experiments, we apply the default k-means function of MATLAB to get the clustering result in each view on the three datasets.
2. *EM*: EM (Expectation Maximization) [32] is an iterative method for finding maximum likelihood or maximum a posteriori



The mfeat-fou view

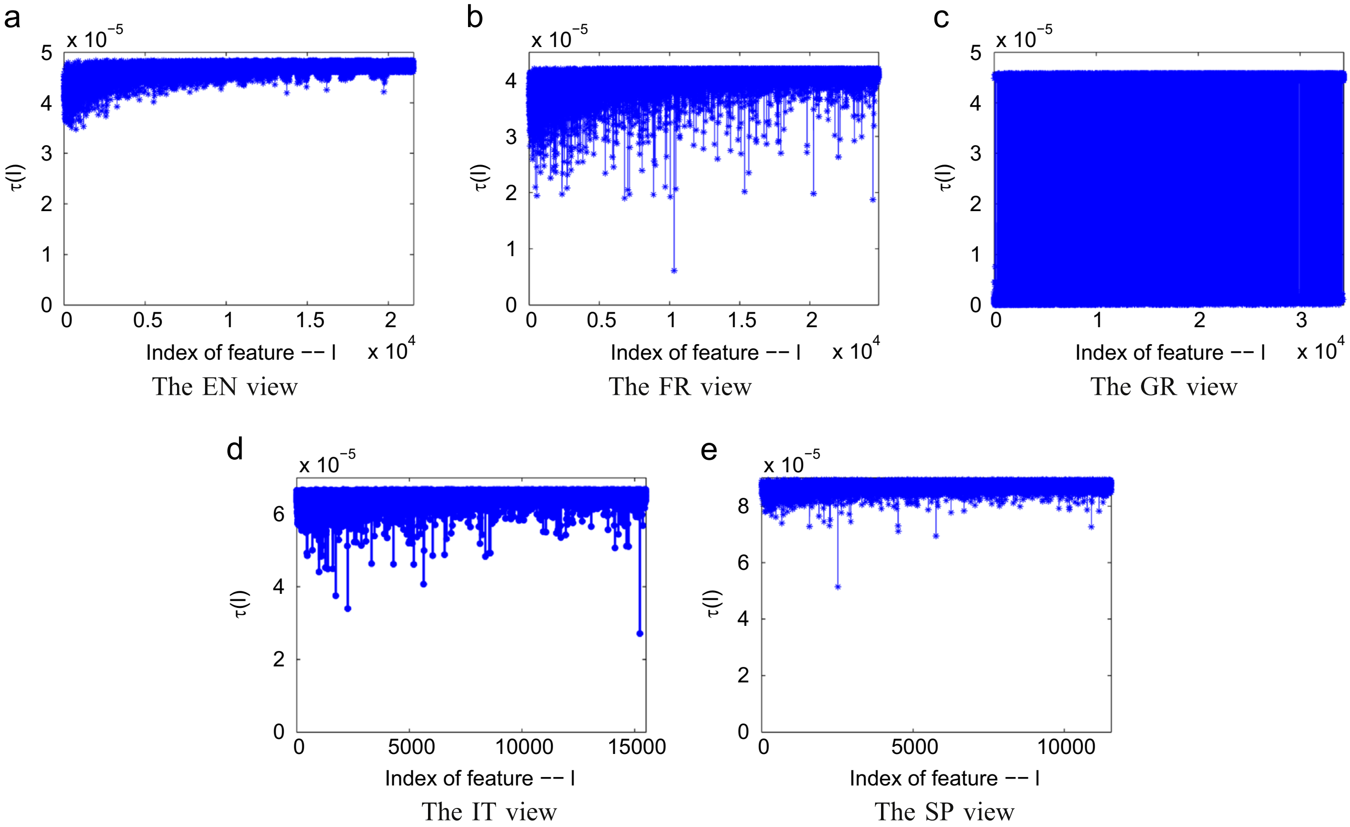
The mfeat-fac view

The mfeat-kar view

The mfeat-zer view

The mfeat-pix view

Fig. 6. Analysis of the balancing parameter β on the Mfeat dataset: distribution of the feature weight vector τ^v in each view with $\beta = 0.1$.



The EN view

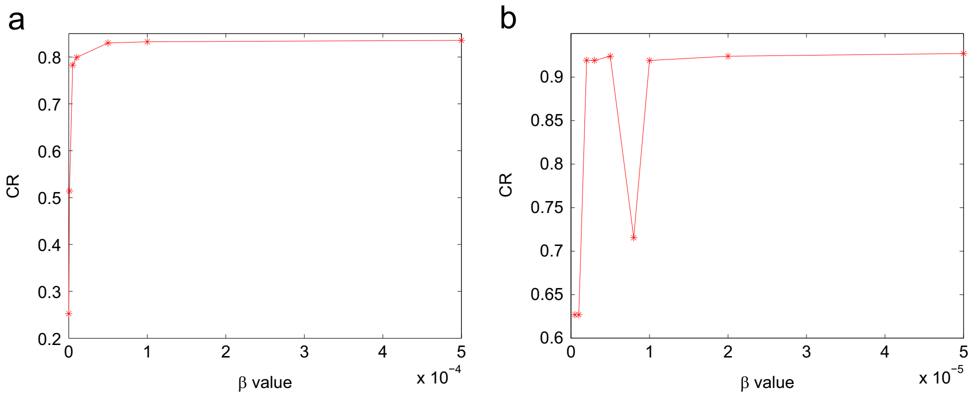
The FR view

The GR view

The IT view

The SP view

Fig. 7. Analysis of the balancing parameter β on the Reuters dataset: distribution of the feature weight vector τ^v in each view with $\beta = 0.1$.



The Mfeat dataset

The Reuters dataset

Fig. 8. Analysis on the balancing parameter β: the clustering performance (in terms of CR) as a function of β.

Table 5
The mean and standard deviations of classification rate (CR) generated by the six algorithms over 100 runs.

Datasets		Average of single view k-means	Average of single view EM	Average of single view LLC-fs	MVKKM	Co-regspec	**WMCFS**
Mfeat	mfeat-fac&mfeat-fou	0.639 ± 0.011	0.655 ± 0.010	0.776 ± 0.009	0.825 ± 0.012	0.792 ± 0.013	$\mathbf{0.835} \pm 0.010$
	mfeat-fac&mfeat-zer	0.610 ± 0.013	0.595 ± 0.009	0.708 ± 0.010	0.718 ± 0.015	0.652 ± 0.007	$\mathbf{0.794} \pm 0.009$
	All 5 views	0.678 ± 0.012	0.636 ± 0.010	0.815 ± 0.006	0.646 ± 0.014	0.735 ± 0.010	$\mathbf{0.836} \pm 0.010$
Reuters	EN&FR	0.698 ± 0.007	0.770 ± 0.008	0.682 ± 0.010	0.626 ± 0.009	0.915 ± 0.012	$\mathbf{0.925} \pm 0.008$
	EN&GR	0.762 ± 0.009	0.770 ± 0.011	0.782 ± 0.013	0.638 ± 0.008	0.907 ± 0.010	$\mathbf{0.926} \pm 0.009$
	All 5 views	0.748 ± 0.011	0.683 ± 0.010	0.825 ± 0.012	0.690 ± 0.007	0.925 ± 0.007	$\mathbf{0.927} \pm 0.007$
Corel	Col-h&Col-m	0.423 ± 0.007	0.553 ± 0.010	0.480 ± 0.011	0.502 ± 0.010	0.621 ± 0.011	$\mathbf{0.657} \pm 0.007$
	Coo-t&Col-hl	0.344 ± 0.013	0.495 ± 0.015	0.423 ± 0.012	0.513 ± 0.007	0.656 ± 0.010	$\mathbf{0.690} \pm 0.012$
	All 4 views	0.384 ± 0.013	0.505 ± 0.010	0.459 ± 0.013	0.508 ± 0.014	0.698 ± 0.008	$\mathbf{0.712} \pm 0.008$

(MAP) estimates of parameters in statistical models. In our experiments, we apply the default EM function of MATLAB to get the clustering result in each view on the three datasets.

3. *MVKKM*: In the MVKKM (multi-view kernel k-means) algorithm [22], a weighted combination of kernels is learned to conduct clustering. There is one parameter p used to control the sparsity of view weight vector with the best parameter value selected from $(1, 6]$ as suggested in [22]. In our experiments, we get the best performance with $p=3$, $p=1.5$ and $p=2$ on the Mfeat, Reuters and Corel datasets respectively. Additionally, we stop the iteration when the gap of objective between two consecutive iterations is less than 0.00001.
4. *Co-regspec*: Co-regspec (Co-regularized multi-view spectral clustering) [20] is a method aiming at minimizing the differences between various views. We use Gaussian kernel for constructing similarity matrix. The hyperparameter λ which trades off the spectral clustering objectives and the spectral embedding (dis)agreement term can be adjusted by the algorithm itself. We set the maximum number of iterations to 10 as suggested in the paper [20].
5. LLC-fs: LLC-fs (Local Learning-Based Clustering with Feature Selection) [33] is a single view data clustering algorithm with feature selection. In our experiment, we run the LLC-fs algorithm in each view of the three datasets. The size of neighborhood k and the trade off parameter β are chosen from the prespecified candidate as reported in the paper [33] and we only report the best performances with $k=30$ on the Mfeat dataset, $k=70$ on the Reuters dataset and $k=55$ on the Corel dataset. Additionally, we stop the iteration when the gap of objective between two consecutive iterations is less than 0.0001.

Tables 5 and 6 report the mean and standard deviation of CR and NMI respectively generated by the six algorithms on the three testing datasets over 100 runs, where different random initializations are used in performing clustering. Apart from performing clustering on the three original whole-view datasets, we also report the clustering results on the datasets formed by some randomly selected views. For instance, on the Mfeat dataset, apart from the five-view data, we also run on the 2 two-view datasets formed by mfeat-fac&mfeat-fou and mfeat-fac&mfeat-zer respectively.

By comparing the results of our algorithm and other compared algorithms, we can draw the conclusion that our proposed method performs the best in the case of multi-view clustering, since we consider the weights both for various views and for various features by measuring their contributions to the clustering result. In particular, on the five-view Mfeat dataset, a classification rate as high as 0.836 can be generated, achieving a significant improvement over the existing methods; we get the best NMI value at the same time. On the five-view Reuters dataset, the proposed WMCFS method is slightly better than the second best Co-regspec method but much better than the other methods. Even on the two-view datasets, the WMCFS method performs much better than the compared methods in terms of both CR and NMI. Another important result is that, for the multi-view clustering methods, the performance achieved on the five-view datasets is much better than that achieved on the two-view datasets. This confirms the fact that adding useful views will enhance the clustering performance of the multi-view clustering methods.

Table 6
The mean and standard deviations of normalized mutual information (NMI) generated by the six algorithms over 100 runs.

Datasets		Average of single view *k*-means	Average of single view EM	Average of single view LLC-fs	MVKKM	Co-regspec	**WMCFS**
Mfeat	mfeat-fac&mfeat-fou	0.594 ± 0.010	0.617 ± 0.010	0.731 ± 0.010	0.782 ± 0.011	0.759 ± 0.013	**0.790** ± 0.010
	mfeat-fac&mfeat-zer	0.562 ± 0.012	0.541 ± 0.008	0.653 ± 0.011	0.654 ± 0.016	0.610 ± 0.008	**0.753** ± 0.009
	All 5 views	0.626 ± 0.010	0.584 ± 0.010	0.765 ± 0.007	0.600 ± 0.014	0.695 ± 0.010	**0.794** ± 0.010
Reuters	EN&FR	0.654 ± 0.008	0.729 ± 0.008	0.635 ± 0.010	0.575 ± 0.009	0.846 ± 0.011	**0.859** ± 0.007
	EN&GR	0.707 ± 0.009	0.720 ± 0.011	0.726 ± 0.010	0.598 ± 0.009	0.845 ± 0.009	**0.862** ± 0.006
	All 5 views	0.703 ± 0.010	0.640 ± 0.009	0.788 ± 0.012	0.652 ± 0.007	0.864 ± 0.008	**0.866** ± 0.007
Corel	Col-h&Col-m	0.385 ± 0.008	0.502 ± 0.010	0.432 ± 0.011	0.450 ± 0.012	0.570 ± 0.011	**0.611** ± 0.007
	Coo-t&Col-hl	0.307 ± 0.012	0.442 ± 0.015	0.382 ± 0.013	0.470 ± 0.007	0.602 ± 0.011	**0.643** ± 0.013
	All 4 views	0.330 ± 0.012	0.460 ± 0.010	0.403 ± 0.012	0.461 ± 0.013	0.633 ± 0.008	**0.652** ± 0.008

5. Conclusion

In this paper, we have proposed a novel multi-view clustering methods, termed Weighted Multi-view Clustering with Feature Selection (WMCFS), which simultaneously performs feature selection and multi-view data clustering. A global objective function is proposed, which takes into consideration both of the multi-view learning and the feature selection in the process of data clustering. In the global objective function, two weighting schemes are designed that respectively weight the views of data points and feature representation in each view, such that the best view and the most representative feature space in each view can be selected for clustering. To solve the objective function, we design an EM-like iteration, which consists of three main stages and can converge to satisfactory results. Experiments have been conducted on three real-world datasets, the results of which validate the effectiveness of the proposed method. In the future work, we plan to extend our algorithm into multi-view clustering with missing data, and with the capability of selecting the number of clusters automatically. Moreover, we will also try the exploration of the automatical way to determine p and β which would not rely on our prior knowledge any more.

Conflict of interest

There is no conflict of interest.

Acknowledgments

This work was supported by NSFC (61173084 and 61502543), CCF-Tencent Open Research Fund (CCF-TencentRAGR20140112), the PhD Start-up Fund of Natural Science Foundation of Guangdong Province, China (2014A030310180), Guangdong Natural Science Funds for Distinguished Young Scholar (No. 16050000051). The authors would like to thank the associate editor and reviewers for their comments which are very helpful in improving the manuscript.

References

[1] R. Xu, D. Wunsch, et al., Survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (2005) 645–678.

[2] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[3] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 888–905.

[4] C.-D. Wang, J.-H. Lai, J.-Y. Zhu, A conscience on-line learning approach for kernel-based clustering, in: Proceedings of the 10th International Conference on Data Mining, 2010, pp. 531–540.

[5] Z. Wu, R. Leahy, An optimal graph theoretic approach to data clustering: theory and its application to image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 15 (1993) 1101–1113.

[6] E. Dahlhaus, Parallel algorithms for hierarchical clustering and applications to split decomposition and parity graph recognition, J. Algorithms 36 (2000) 205–240.

[7] E. Eaton, M.d. EsJardins, S. Jacob, Multi-view constrained clustering with an incomplete mapping between views, Knowl. Inf. Syst. 38 (2014) 231–257.

[8] E. Taralova, F. De la Torre, M. Hebert, Source constrained clustering, in: Proceedings of the 2011 IEEE International Conference on Computer Vision, 2011, pp. 1927–1934.

[9] L. Huang, J. Lu, Y.-P. Tan, Co-learned multi-view spectral clustering for face recognition based on image sets, IEEE Signal Process. Lett. 21 (2014) 875–879.

[10] X. Wang, B. Qian, I. Davidson, Improving document clustering using automated machine translation, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, 2012, pp. 645–653.

[11] X. Zhao, N. Evans, J.-L. Dugelay, A subspace co-training framework for multi-view clustering, Pattern Recognit. Lett. 41 (2014) 73–82.

[12] M. Fang, Y. Guo, X. Zhang, X. Li, Multi-source transfer learning based on label shared subspace, Pattern Recognit. Lett. 51 (2014) 101–106.

[13] H. Wang, X. Wang, J. Zheng, J.R. Deller, H. Peng, L. Zhu, W. Chen, X. Li, R. Liu, H. Bao, Video object matching across multiple non-overlapping camera views based on multi-feature fusion and incremental learning, Pattern Recognit. 47 (2014) 3841–3851.

[14] S. Xiang, L. Yuan, W. Fan, Y. Wang, P.M. Thompson, J. Ye, Multi-source learning with block-wise missing data for Alzheimer's disease prediction, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2013, pp. 185–193.

[15] J.A. Sáez, J. Derrac, J. Luengo, F. Herrera, Statistical computation of feature weighting schemes through data estimation for nearest neighbor classifiers, Pattern Recognit. 47 (2014) 3941–3948.

[16] Z. Chen, S. Xiong, Z. Fang, Q. Li, B. Wang, Q. Zou, A kernel support vector machine-based feature selection approach for recognizing Flying Apsaras' streamers in the Dunhuang Grotto Murals, China, Pattern Recognit. Lett. 49 (2014) 107–113.

[17] S. Bickel, T. Scheffer, Multi-view clustering, in: Proceedings of the 4th International Conference on Data Mining, 2004, pp. 19–26.

[18] V.R. de Sa, Spectral clustering with two views, in: ICML Workshop on Learning with Multiple Views, 2005, pp. 20–27.

[19] A. Kumar, H. Daumé, A co-training approach for multi-view spectral clustering, in: Proceedings of the 28th International Conference on Machine Learning, 2011, pp. 393–400.

[20] A. Kumar, P. Rai, H. Daumé III, Co-regularized multi-view spectral clustering, in: Neural Information Processing Systems (NIPS), 2011, pp. 1413–1421.

[21] X. Wang, B. Qian, J. Ye, I. Davidson, Multi-objective multi-view spectral clustering via Pareto optimization, in: SIAM International Conference on Data Mining (SDM), 2013, pp. 234–242.

[22] G. Tzortzis, A. Likas, Kernel-based weighted multi-view clustering, in: Proceedings of the 12th International Conference on Data Mining, 2012, pp. 675–684.

[23] X. Zhao, N. Evans, J.-L. Dugelay, A subspace co-training framework for multi-view clustering, Pattern Recognit. Lett. 41 (2014) 73–82.

[24] H. Wang, F. Nie, H. Huang, Multi-view clustering and feature learning via structured sparsity, in: Proceedings of the 30th International Conference on Machine Learning, 2013, pp. 352–360.

[25] X. Chen, X. Xu, J.Z. Huang, Y. Ye, Tw-(k)-means: automated two-level variable weighting clustering algorithm for multiview data, IEEE Trans. Knowl. Data Eng. 25 (2013) 932–944.

[26] X. Cai, F. Nie, H. Huang, Multi-view k-means clustering on big data, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Beijing, China, AAAI Press, 2013, pp. 2598–2604.

[27] A. Asuncion, D. Newman, UCI machine learning repository, ⟨http://archive.ics.uci.edu/ml/⟩, 2007.

[28] M.-R. Amini, N. Usunier, C. Goutte, Learning from multiple partially observed views—an application to multilingual text categorization, in: Neural Information Processing Systems (NIPS), 2009, pp. 28–36.

[29] N. Ueffing, M. Simard, S. Larkin, J.H. Johnson, NRC's PORTAGE system for WMT 2007, in: ACL-2007 Second Workshop on SMT, 2007, pp. 185–188.

[30] C.-D. Wang, J.-H. Lai, C.Y. Suen, J.-Y. Zhu, Multi-exemplar affinity propagation, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 2223–2237.

[31] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2002) 583–617.

[32] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the em algorithm, J. R. Stat. Soc. Ser. B (Methodological) (1977) 1–38.

[33] H. Zeng, Y.-m. Cheung, Feature selection and kernel learning for local learning-based clustering, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011) 1532–1547.

Yu-Meng Xu received her master degree in 2015 from Sun Yat-sen University, China. Her research interest is data clustering.

Chang-Dong Wang received his Ph.D. degree in computer science in 2013 from Sun Yat-sen University, China. He is currently an assistant professor at School of Mobile Information Engineering, Sun Yat-sen University. His current research interests include machine learning and pattern recognition, especially focusing on data clustering and its applications. He has published over 30 scientific papers in international journals and conferences such as IEEE TPAMI, IEEE TKDE, IEEE TSMC-C, Pattern Recognition, Knowledge and Information System, Neurocomputing, ICDM and SDM. His ICDM 2010 paper won the Honorable Mention for Best Research Paper Awards. He won 2012 Microsoft Research Fellowship Nomination Award. He was awarded 2015 Chinese Association for Artificial Intelligence (CAAI) Outstanding Dissertation.

Jian-Huang Lai received his M.Sc. degree in applied mathematics in 1989 and his Ph.D. in mathematics in 1999 from Sun Yat-sen University, China. He joined Sun Yat-sen University in 1989 as an assistant professor, where currently, he is a professor with the Department of Automation of School of Information Science and Technology and dean of School of Information Science and Technology. His current research interests are in the areas of digital image processing, pattern recognition, multimedia communication, wavelet and its applications. He has published over 150 scientific papers in the international journals and conferences on image processing and pattern recognition, e.g. IEEE TPAMI, IEEE TKDE, IEEE TNN, IEEE TIP, IEEE TSMC (Part B), Pattern Recognition, ICCV, CVPR and ICDM. Lai serves as a standing member of the Image and Graphics Association of China and also serves as a standing director of the Image and Graphics Association of Guangdong.