



# Multi-view Proximity Learning for Clustering

Kun-Yu Lin, Ling Huang, Chang-Dong Wang<sup>(✉)</sup>, and Hong-Yang Chao

School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China  
kunyulin14@outlook.com, huanglinghl@hotmail.com, changdongwang@hotmail.com,  
isschhy@mail.sysu.edu.cn

**Abstract.** In recent years, multi-view clustering has become a hot research topic due to the increasing amount of multi-view data. Among existing multi-view clustering methods, proximity-based method is a typical class and achieves much success. Usually, these methods need proximity matrices as inputs, which can be constructed by some nearest-neighbors-based approaches. However, in this way, neither the intra-view cluster structure nor the inter-view correlation is considered in constructing proximity matrices. To address this issue, we propose a novel method, named *multi-view proximity learning*. By introducing the idea of representative, our model can consider both the relations between data objects and the cluster structure within individual views. Besides, the spectral-embedding-based scheme is adopted for modeling the correlations across different views, i.e. the view consistency and complement properties. Extensive experiments on both synthetic and real-world datasets demonstrate the effectiveness of our method.

**Keywords:** Multi-view clustering · Proximity learning  
Representative · Spectral embedding

## 1 Introduction

Recently, multi-view data, whose data features are collected from multiple heterogeneous but related views, have arisen in a number of fields [1–8], such as pattern recognition, data mining, natural language processing, etc. For instance, a web page can be described in two views, one contains the words occurring in the page and the other contains the words occurring in the hyperlinks pointing to that page [4]. Another example is the multilingual document, which is available in several languages such that each language is taken as a separate view [5]. In these fields, data clustering is a basic but widely used technique [9]. Considering clustering the multi-view data, it is difficult to produce good results by using only one view of feature, since usually each view only provides partial information [10]. Therefore, it is necessary to properly combine information from all views together to improve the clustering performance. This leads to the emergence of multi-view clustering.

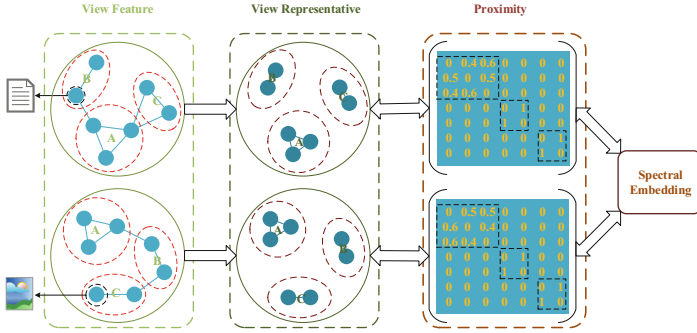
Proximity-based method is a kind of typical method for multi-view clustering. These methods integrate the information from different views by making use of the predefined proximity matrices together. One of the most straightforward scheme for view integration is weighted combination, which combines the proximity matrices from all views together by weighted addition via an adaptive weighting parameter for each view [11, 12]. Besides, some other useful methods are developed. In [13, 14], co-training based approaches are adopted to share information among views, which improves the proximity matrices to fit multi-view data. Co-regularized approaches are also effective approaches to view integration [15, 16]. Wang et al. propose a method which considers the neighborhood consistency of different views [17], while Xia et al. consider the low-rank and sparse properties of proximity matrices [18].

Despite the success of the aforementioned proximity-based methods, they suffer from some common problems. First, proximity matrices are needed as inputs for these methods, while usually data features are given rather than proximity matrices. In this case, some nearest-neighbors-based methods are applied on data features to construct proximity matrices, such as  $k$ -nearest neighbors, Gaussian proximity [19] and self-tuned Gaussian [20]. However, these proximity construction methods do not consider the underlying cluster structures, such that the constructed proximities may not exhibit good properties for clustering. Moreover, these methods only consider separately the information in individual views, leading to the loss of the inter-view correlations.

In order to address these problems, we propose a new *multi-view proximity learning* (MVPL) method for multi-view clustering. In the multi-view proximity learning, both the relations between data objects in individual views and the correlations across different views are considered. For the intra-view relations, a novel idea of data representative is adopted, such that the cluster structure is also taken into account during the learning process. Besides, spectral-embedding-based scheme is designed for modeling the inter-view correlations, such that both the view consistency and complement properties can be utilized for improving the clustering performance. Accordingly, an objective function is designed and an alternative iteration scheme is proposed to optimize the objective. Extensive experiments conducted on both synthetic and real-world datasets demonstrate the effectiveness of the proposed model.

## 2 The Proposed Model

In order to address the proximity learning problem for multi-view data, our model should consider two parts. One is the intra-view learning quality, which means that the learning process should consider the relations between data objects within each view. Inspired by [21], the proposed model discovers these relations based on the idea of representative. It can transform the original view feature into a more suitable representation for proximity learning, by which the cluster structures are also considered. In particular, in each view, each feature vector has a dedicated representative that is very similar to itself, and representatives of data objects with higher proximity should be similar to each other.



**Fig. 1.** Illustration of the main idea of our method. In this simple example, the dataset contains a document view and an image view.

The other part is to model the correlations across different views, such that both the view consistency and complement properties can be utilized for improving clustering performance. The view consistency property implies that the proximities learnt from different views will reach a certain degree of consistency, while the view complement property implies that one view will provide complementary information for the other views. Accordingly, a well-designed inter-view criterion function is proposed based on spectral embedding. For clarity, Fig. 1 illustrates the main idea of our method by a two-view example. From the figure, we find that the data representatives are determined by both view features and learnt proximities. Similarly, the learnt proximities are derived from intra-view data representatives and further mutually affected in a inter-view manner by spectral embedding. In what follows, we will introduce the model in detail.

## 2.1 The Objective Function

Given a dataset containing  $n$  objects whose features are collected from  $m$  views, the features in the  $v$ -th view are represented by matrix  $X^v = [\mathbf{x}_1^v, \mathbf{x}_2^v, \dots, \mathbf{x}_n^v] \in \mathbb{R}^{d^v \times n}$ , where  $d^v$  is the dimensionality of the  $v$ -th view and  $\mathbf{x}_i^v$  is the feature vector for the  $i$ -th object in the  $v$ -th view. The goal of the multi-view proximity learning is to learn a proximity set  $\{S^1, S^2, \dots, S^m\}$ , where  $S^v = [s_{ij}^v]_{n \times n}$  is the proximity matrix for the  $v$ -th view with  $s_{ij}^v$  representing the proximity between the  $i$ -th and  $j$ -th objects in the  $v$ -th view. According to the discussion above, the learning process should consider both intra-view and inter-view criteria.

**Intra-view Criterion.** In order to discover the relations between data objects in individual views, we introduce the idea of data representatives, which are better representations with clearer cluster structures for data objects. Intuitively in this process, original data point is moved to a better position for clustering according to its relations with other data points. We use  $U^v = [\mathbf{u}_1^v, \mathbf{u}_2^v, \dots, \mathbf{u}_n^v] \in \mathbb{R}^{d^v \times n}$  to denote the representative matrix where  $\mathbf{u}_i^v$  is the representative for feature vector  $\mathbf{x}_i^v$  in the  $v$ -th view. Treating the original feature as important basis

for learning representative,  $\mathbf{u}_i^v$  should not be far from  $\mathbf{x}_i^v$ , otherwise the topological structure will be destroyed. Besides, the learning of representative should also consider the proximities between objects. If two data objects have higher proximity in one view then their representatives should be relatively closer. Similarly, the proximity learning should consider the relations between data representatives. If two data representatives  $\mathbf{u}_i^v$  and  $\mathbf{u}_j^v$  are close in the  $v$ -th view, then  $s_{ij}^v$  should be relatively large. In other words, the learning processes of representatives and proximities are mutually affected by each other. According to the above discussion, the intra-view criterion is defined as follows,

$$\begin{aligned} \Phi^v(U^v, S^v) &= \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i^v - \mathbf{u}_i^v\|_2^2 + \frac{\alpha}{n^2} \left( \sum_{i=1}^n \sum_{j=1}^n s_{ij}^v \|\mathbf{u}_i^v - \mathbf{u}_j^v\|_2^2 + \beta \|S^v\|_F^2 \right) \\ \text{s.t. } &\sum_{j=1}^n s_{ij}^v = 1, s_{ij}^v \geq 0, \forall i, j \end{aligned} \quad (1)$$

where  $\|\cdot\|_2$  is the  $L_2$  norm of vector,  $\|\cdot\|_F^2$  is the Frobenius norm of matrix and  $\alpha, \beta > 0$  are trade-off parameters. In our paper, the probabilistic proximities are used. Therefore constraint  $\sum_{j=1}^n s_{ij}^v = 1$  and  $s_{ij}^v \geq 0$  should be introduced. The term  $\beta \|S^v\|_F^2$  is adopted for controlling the sparsity of learnt proximity. If  $\beta$  is large, the learnt proximity matrix will be relatively dense, while a smaller  $\beta$  will make the matrix sparser.

**Inter-view Criterion.** The inter-view criterion considers both the view consistency and view complement properties. We design such criterion by introducing the concept of spectral embedding. Spectral embedding is a low-dimensional representation of data object, which is obtained through spectral decomposition on specific matrix. In our model, spectral embedding is the representation integrating information from all views. By denoting the embedding matrix as  $F = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_n] \in \mathbb{R}^{c \times n}$  with  $\mathbf{f}_i$  being the  $c$ -dimensional spectral embedding of the  $i$ -th data object, the relation between  $F$  and the learnt proximity  $S^v$  can be modeled by

$$\frac{1}{2n^2} \sum_{i=1}^n \sum_{j=1}^n s_{ij}^v \|\mathbf{f}_i - \mathbf{f}_j\|_2^2 \quad \text{s.t. } FF^T = I \quad (2)$$

where  $I$  is the identity matrix. Here  $FF^T = I$  is a widely used constraint for weakening the relations between the features of embedding, which makes  $F$  a better representation [19]. If the distance between  $\mathbf{f}_i$  and  $\mathbf{f}_j$  is small, it implies that  $i$ -th and  $j$ -th data objects may have higher proximity in all views. If the value of (2) is smaller, the learnt proximity of the  $v$ -th view is more consistent with the spectral embedding  $F$ . Since the spectral embedding  $F$  carries information from all views, the high consistency between  $F$  and  $S^v$  implies that information of other views is transferred to the  $v$ -th view, which reflects the view complement property. Moreover, proximities from different views can reach

a certain degree of consistency through  $F$ . Here  $F$  is regarded as a medium for inter-view interactions, which reflects the view consistency property. Considering all views together, we get the inter-view criterion as follows,

$$\Psi(\{S^v\}, F) = \frac{1}{2n^2} \sum_{v=1}^m \sum_{i=1}^n \sum_{j=1}^n s_{ij}^v \|\mathbf{f}_i - \mathbf{f}_j\|_2^2 \quad \text{s.t. } FF^T = I \quad (3)$$

which models the inter-view correlations through the spectral embedding.

**The Overall Objective Function** According to the discussion above, we can use  $\Phi^v(U^v, S^v)$  to measure the intra-view learning quality and  $\Psi(\{S^v\}, F)$  to measure the inter-view consistency and complement properties. By integrating them together, we can get the overall objective function as follows,

$$\begin{aligned} & \min_{\{U^v\}, \{S^v\}, F} \sum_{v=1}^m \Phi^v(U^v, S^v) + \gamma \Psi(\{S^v\}, F) \\ & \text{s.t. } \sum_{j=1}^n s_{ij}^v = 1, s_{ij}^v \geq 0, \forall i, j, v, FF^T = I \end{aligned} \quad (4)$$

where  $\gamma > 0$  is the trade-off parameter balancing the intra-view criterion and the inter-view criterion. By minimizing the objective function (4), both the learning quality of proximities in all views and the interactions between different views are considered, such that suitable proximities for multi-view data can be obtained. Following the convention of spectral clustering, the dimensionality of spectral embedding can be set as the predefined number of clusters [19].

## 2.2 Determination of Parameter $\beta$

In the proposed model, three parameters are needed as inputs for proximity learning. Parameter  $\alpha$  is adopted to control the distances between data features and data representatives, while parameter  $\gamma$  is adopted for controlling the view consistency. Both parameters should be determined according to the properties of datasets. In comparison,  $\beta$  is adopted for controlling the sparsity of learnt proximities, which has less variability. Therefore, it is necessary to design a method for determining its value more easily.

Inspired by [22], we propose a method based on  $k$ -nearest neighbors to determine  $\beta$ . It also induces a method for constructing single-view proximity, which will be used in our experiments. Considering data feature in certain view, whose data matrix is  $X = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$  (here we ignore the superscript specifying view index for simplicity), we can learn the proximity vector  $\mathbf{w}_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$  associated with  $\mathbf{x}_i$  by solving the following model

$$\min_{\mathbf{w}_i} \frac{1}{2} \left\| \mathbf{w}_i + \frac{\mathbf{d}_i^{\mathbf{x}}}{2\beta_i} \right\|_2^2 \quad \text{s.t. } \mathbf{w}_i^T \mathbf{1} = 1, \mathbf{w}_i \geq 0, \quad (5)$$

where  $\beta_i > 0$  is the sparsity parameter,  $\mathbf{1}$  is the all-one vector and  $\mathbf{w}_i \geq 0$  means all elements of vector  $\mathbf{w}_i$  are not less than 0. We assume the original distance vector as  $\hat{\mathbf{d}}_i^{\mathbf{x}} = [\hat{d}_{i1}^{\mathbf{x}}, \hat{d}_{i2}^{\mathbf{x}}, \dots, \hat{d}_{in}^{\mathbf{x}}]^T$ , where  $\hat{d}_{ii}^{\mathbf{x}}$  is set as a very large number (i.e. ignoring  $\mathbf{x}_i$  itself) and  $\forall j \neq i, \hat{d}_{ij}^{\mathbf{x}} = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$ . The distance vector  $\hat{\mathbf{d}}_i^{\mathbf{x}}$  in (5) is defined by  $\mathbf{d}_i^{\mathbf{x}} = [d_{i1}^{\mathbf{x}}, d_{i2}^{\mathbf{x}}, \dots, d_{in}^{\mathbf{x}}]^T$ , which is the sorted vector of  $\hat{\mathbf{d}}_i^{\mathbf{x}}$  such that  $d_{i1}^{\mathbf{x}} \leq d_{i2}^{\mathbf{x}} \leq \dots \leq d_{in}^{\mathbf{x}}$ . In the model, the parameter  $\beta_i$  determines the number of nonzero elements in the proximity information vector  $\mathbf{w}_i$ . If  $\beta_i = 0$ , there will be only one nonzero element in the vector, corresponding to the nearest neighbor of object  $\mathbf{x}_i$ . If  $\beta_i \rightarrow \infty$ , all elements will be nonzero except the one corresponding to  $\mathbf{x}_i$ . Aiming to solve problem (5), we write down its Lagrangian function as

$$\mathcal{L}(\mathbf{w}_i, \eta, \mu_i) = \frac{1}{2} \left\| \mathbf{w}_i + \frac{\mathbf{d}_i^{\mathbf{x}}}{2\beta_i} \right\|_2^2 - \eta (\mathbf{w}_i^T \mathbf{1} - 1) - \mu_i^T \mathbf{w}_i \tag{6}$$

where  $\eta$  and  $\mu_i \geq 0$  are Lagrangian multipliers. According to the KKT condition, the optimal solution of  $\mathbf{w}_i$  is given by

$$w_{ij} = \max \left( -\frac{d_{ij}^{\mathbf{x}}}{2\beta_i} + \eta, 0 \right). \tag{7}$$

If there are exactly  $k$  nonzero elements in the vector  $\mathbf{w}_i$ , we get the value of Lagrangian multiplier  $\eta = \frac{1}{k} + \frac{1}{2k\beta_i} \sum_{j=1}^k d_{ij}^{\mathbf{x}}$  [22]. These  $k$  nonzero elements of  $\mathbf{w}_i$  correspond to the  $k$ -nearest neighbors of  $\mathbf{x}_i$  and the elements of  $\mathbf{w}_i$  satisfy  $\forall 1 \leq j \leq k, w_{ij} > 0$  and  $\forall j \geq k + 1, w_{ij} = 0$ . According to the constraint  $\mathbf{w}_i^T \mathbf{1} = 1$ , the sparsity parameter  $\beta_i$  can be set as

$$\beta_i = \frac{k}{2} d_{i,k+1}^{\mathbf{x}} - \frac{1}{2} \sum_{j=1}^k d_{ij}^{\mathbf{x}}, \tag{8}$$

such that the resulting  $\mathbf{w}_i$  will have exactly  $k$  nonzero elements. Considering all data objects, the sparsity parameter  $\beta$  can be set as the average of  $\beta_i$ , which is given by

$$\beta = \frac{1}{n} \sum_{i=1}^n \left( \frac{k}{2} d_{i,k+1}^{\mathbf{x}} - \frac{1}{2} \sum_{j=1}^k d_{ij}^{\mathbf{x}} \right). \tag{9}$$

Using the method above, we can determine the sparsity parameter according to the number of neighbors  $k$ , which is much easier to tune. Furthermore, the single-view weighted  $k$ -nearest neighbors proximity can be constructed after  $k$  is determined. For multi-view data, since different views may have different distance distributions, it is more reasonable to use different sparsity parameters for different views. Therefore, the modified intra-view criterion function is given by

$$\tilde{\Phi}^v(U^v, S^v) = \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i^v - \mathbf{u}_i^v\|_2^2 + \frac{\alpha}{n^2} \left( \sum_{i=1}^n \sum_{j=1}^n s_{ij}^v \|\mathbf{u}_i^v - \mathbf{u}_j^v\|_2^2 + \beta^v \|S^v\|_F^2 \right) \tag{10}$$

where  $\beta^v > 0$  is the sparsity parameter for the  $v$ -th view determined by the aforementioned method via the number of neighbors  $k$ . Finally, our objective is given by

$$\begin{aligned} \min_{U^v, S^v, F} \sum_{v=1}^m \tilde{\Phi}^v(U^v, S^v) + \gamma \Psi(\{S^v\}, F) \\ \text{s.t. } \sum_{j=1}^n s_{ij}^v = 1, s_{ij}^v \geq 0, \forall i, j, v, FF^T = I. \end{aligned} \quad (11)$$

Although more sparsity parameters are introduced to control the model in (11) compared with (4), they can be determined via the same number of nearest neighbors  $k$ .

### 2.3 Optimization

In this subsection, the alternative iteration scheme is used to solve problem (11).

**Update  $U^v$ .** When  $S^v$  and  $F$  are fixed, the subproblem with respect to  $U^v$  is given by

$$\min_{U^v} \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i^v - \mathbf{u}_i^v\|_2^2 + \frac{\alpha}{n^2} \sum_{i=1}^n \sum_{j=1}^n s_{ij}^v \|\mathbf{u}_i^v - \mathbf{u}_j^v\|_2^2. \quad (12)$$

In order to rewrite the subproblem into matrix form, we introduce the property [19] as

$$\text{Tr}(HL_G H^T) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n g_{ij} \|\mathbf{h}_i - \mathbf{h}_j\|_2^2 \quad (13)$$

where  $\text{Tr}(\cdot)$  is the trace operator for matrix,  $G = \{g_{ij}\} \in \mathbb{R}^{a \times a}$  and  $H = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_a] \in \mathbb{R}^{b \times a}$ .  $L_G$  is the unnormalized Laplacian matrix of  $G$  defined by  $L_G = D_G - G$ , where  $D_G$  is the degree matrix of  $G$ . Using the property (13), the subproblem can be transformed as

$$\min_{U^v} \|X^v - U^v\|_F^2 + \frac{2\alpha}{n} \text{tr}(U^v L_S^v U^{vT}) \quad (14)$$

where  $L_S^v$  is the unnormalized Laplacian matrix of  $(S^v + S^{vT})/2$ . Setting the derivative with respect to  $U^v$  to zero, we find that  $U^v$  satisfies the equation as follows

$$U^v \left( I + \frac{2\alpha}{n} L_S^v \right) = X^v, \quad (15)$$

which can be solved by matrix inversion. Besides, the problem is essentially a least-square problem, which can also be solved in many efficient ways.

**Update  $S^v$ .** When  $U^v$  and  $F$  are fixed, the subproblem with respect to  $S^v$  is given by

$$\begin{aligned} \min_{S^v} & \sum_{i=1}^n \sum_{j=1}^n s_{ij}^v \|\mathbf{u}_i^v - \mathbf{u}_j^v\|_2^2 + \beta^v \sum_{i=1}^n \sum_{j=1}^n s_{ij}^v + \frac{\gamma}{2\alpha} \sum_{i=1}^n \sum_{j=1}^n s_{ij}^v \|\mathbf{f}_i - \mathbf{f}_j\|_2^2 \\ \text{s.t.} & \sum_{j=1}^n s_{ij}^v = 1, s_{ij}^v \geq 0, \forall i, j, v. \end{aligned} \quad (16)$$

By denoting  $d_{ij}^v = \|\mathbf{u}_i^v - \mathbf{u}_j^v\|_2^2 + \frac{\gamma}{2\alpha} \|\mathbf{f}_i - \mathbf{f}_j\|_2^2$ ,  $\mathbf{d}_i^v = [d_{i1}^v, d_{i2}^v, \dots, d_{in}^v]^T$  and  $\mathbf{s}_i^v = [s_{i1}^v, s_{i2}^v, \dots, s_{in}^v]^T$ , we translate the problem into vector form as follows

$$\begin{aligned} \min_{\mathbf{s}_i^v} & \left\| \mathbf{s}_i^v + \frac{\mathbf{d}_i^v}{2\beta^v} \right\|_2^2 \\ \text{s.t.} & \mathbf{s}_i^{vT} \mathbf{1} = 1, \mathbf{s}_i^v \geq 0, \end{aligned} \quad (17)$$

which is equivalent to computing the Euclidean projection of point  $-\mathbf{d}_i^v/(2\beta^v)$  onto the probability simplex. The problem has a unique solution, which can be solved by using the method proposed in [23].

**Update  $F$ .** When  $U^v$  and  $S^v$  are fixed, the subproblem with respect to  $F$  is to solve a trace minimization problem as

$$\min_{FF^T=I} \text{Tr}(FL_S F^T) \quad (18)$$

where  $L_S = \sum_{v=1}^m L_S^v$ . The optimal  $F$  is a matrix formed by the  $c$  eigenvectors of  $L_S$  corresponding to the  $c$  smallest eigenvalues.

By alternatively update  $U^v$ ,  $S^v$  and  $F$ , the objective value will decrease and finally converge as the iteration goes, from which the solution of problem (11) can be obtained. The optimization algorithm is summarized in Algorithm 1. After learning the proximity matrices, the spectral clustering is applied on the proximity matrices to obtain the clustering results.

### 3 Experiment

In this section, extensive experiments are conducted to demonstrate the effectiveness of the proposed method on one synthetic dataset and four real-world datasets. On the synthetic dataset, we will show how the proposed method works. While on the real-world datasets, parameter analysis, convergence analysis and comparison experiments will be conducted. The code of our method and the testing datasets are available on dropbox<sup>1</sup>.

<sup>1</sup> The code is available on [https://www.dropbox.com/s/tj5zc7yry0ing3l/MVPL\\_PCode.zip?dl=0](https://www.dropbox.com/s/tj5zc7yry0ing3l/MVPL_PCode.zip?dl=0) and the password for decompression is ‘‘DASF AA2018’’.



**Algorithm 1.** Multi-view proximity learning

**Input:** Data matrices of  $m$  views  $\{X^1, X^2, \dots, X^m\}$ , parameters  $\alpha, \gamma$  and  $k$ , number of clusters  $c$ .

1: Initialize representative matrix  $U^v$  as  $X^v$ .

2: Initialize  $S^v$  and determine  $\beta^v$  by the strategy in Section 2.2.

3: Initialize  $F$  by solving Eq. (18).

4: **repeat**

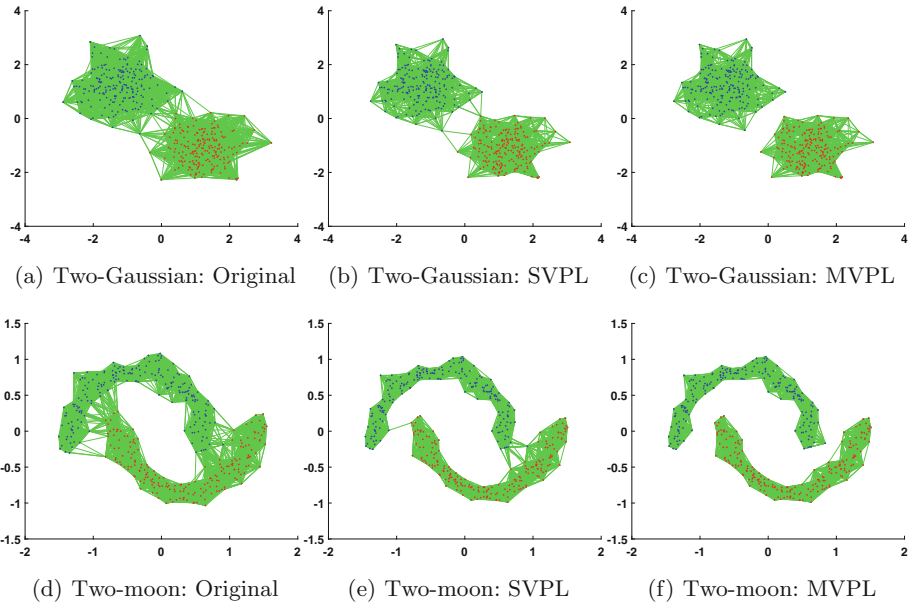
5:   Update  $U^v, \forall v$  by solving Eq. (15).

6:   Update  $S^v, \forall v$  by solving Eq. (17).

7:   Update  $F$  by solving Eq. (18).

8: **until** Convergence or reaching the maximum number of iterations.

**Output:** Proximity matrices  $\{S^1, S^2, \dots, S^m\}$ .



**Fig. 2.** Synthetic experiment. In the figures, points in the first class are in blue while those in the second class are in red. Green lines are edges representing the proximities between data objects, i.e., if the proximity between two data objects in certain view is larger than zero then there is an edge between them. (Color figure online)

### 3.1 Synthetic Experiment

A synthetic dataset consisting of two views, namely Two-Gaussian and Two-moon, is used for demonstrating how the proposed method works. Figure 2(a) and (d) plot the original data points in both views with edges representing the initial proximities learnt by the method introduced in Sect. 2.2. In order to show the significance of considering inter-view criterion, a variant of our method, called SVPL, is introduced by setting  $\gamma = 0$ . It is a single-view proximity learning

method which considers only the intra-view criterion. Figure 2(b) and (e) show the results of SVPL, where points denote the learnt representatives. From the subfigures, we find that SVPL transforms the original data view into a more suitable state for clustering based on representatives. It is essentially equivalent to making the data points move in such a manner that the intra-class connections are stronger and the inter-class connections are weaker. However, the learnt proximity is not good enough since there are still edges between the two clusters. Therefore, we need to consider the inter-view information. Figure 2(c) and (f) show the results of MVPL, where points denote representatives learnt by MVPL. From these two subfigures, we find that there are no edges between clusters in both views. This implies that much better proximities are learnt by considering both the intra-view and the inter-view criterion. The comparison results confirm the significance of considering inter-view criterion.

### 3.2 Real-World Datasets and Evaluation Measures

In this subsection, we will first introduce the four real-world datasets used in experiments.

#### 1. Handwritten numeral dataset

Multiple features (Mfeat) dataset is a handwritten numeral dataset from UCI machine learning repository [24]. The dataset contains handwritten digits from 0 to 9 and each category has 200 objects. In our experiment, we use three kinds of feature to represent images, namely 216 profile correlations, 76 Fourier coefficients and 47 Zernike moments, where each kind of features is regarded as a view.

#### 2. Multi-source news dataset

3Sources dataset<sup>2</sup> is a multi-source news dataset consisting of news collected from three sources, namely BBC, Guardian and Reuters. Although the original dataset contains 984 news articles covering 416 distinct news stories, there are only 169 stories reported by all three medias. In our experiment, we only use these 169 news objects so that each object has three views of features.

#### 3. Object image datasets

Caltech101 [25] is an image dataset consisting of 101 categories of images for object recognition problem. Following the previous work [26], two subsets are selected to generate two datasets for experimental purpose. The first subset is called Caltech101-7, containing 1474 images from 7 widely used categories. The second one is a larger subset called Caltech101-20, which contains 2386 images of 20 categories. Three kinds of features are extracted from the images to generate three views, namely 1984-dimensional HOG feature, 512-dimensional GIST feature and 928-dimensional LBP feature.

The statistic of the four real-world datasets is shown in Table 1.

In order to evaluate the clustering performance of the proposed method and the compared methods, three widely used measures are adopted in our experiments, namely accuracy (ACC), normalized mutual information (NMI) and

<sup>2</sup> <http://mlg.ucd.ie/datasets/3sources.html>.

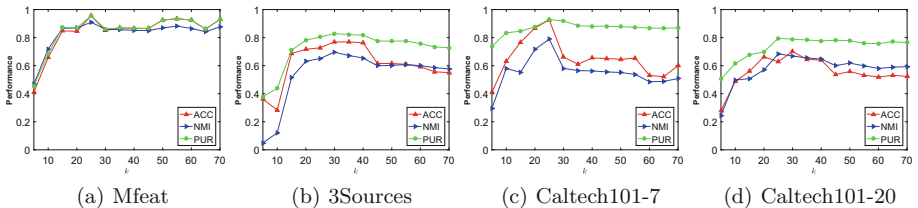
**Table 1.** Statistic of the four real-world datasets.

	Mfeat	3Sources	Caltech101-7	Caltech101-20
View1	fac(216)	BBC(3560)	hog(1984)	hog(1984)
View2	fou(76)	Guardian(3631)	gist(512)	gist(512)
View3	zer(47)	Reuters(3068)	lbp(928)	lbp(928)
# of objects	2000	169	1474	2386
# of classes	10	6	7	20

purity (PUR). For each measure, higher value indicates better performance [17]. In comparison experiments, following [21], the average rank of the performance obtained by each method is also reported across all datasets.

### 3.3 Parameter Analysis

In this subsection, parameter analysis is conducted to show the effect of the three parameters  $\alpha$ ,  $\gamma$  and  $k$ . The first parameter to be analyzed is  $k$ , which determines the value of  $\beta^v$ . By fixing  $\alpha = 1$  and  $\gamma = 0.001$ , we tune the value of  $k$  in range [5, 70] with step 5. The performance in terms of all three measures on the four datasets are reported in Fig. 3. From the figure, we find that the method performs not so well when  $k$  is too small due to the failure of preserving the neighborhood structures. As the value of  $k$  increases, the performance will gradually increase. After reaching the highest point (often around  $k = 30$ ), the value of curve will gradually decrease. Although the method may perform not so well with relatively larger  $k$ , it produces acceptable results. The main reason is that by introducing the idea of representative, which transforms the original data into a more suitable state for proximity learning, the negative impact caused by the noisy neighbors will be alleviated.

**Fig. 3.** Parameter analysis on number of neighbors  $k$ .

Next we analyze the effect of  $\alpha$  and  $\gamma$  by setting  $k = 30$ . According to the properties of datasets, different ranges of  $\gamma$  are used for different datasets while the same range of  $\alpha$  is used for all datasets. The experimental results are shown in Figs. 4, 5, 6 and 7 respectively. From the figures, we find our method has

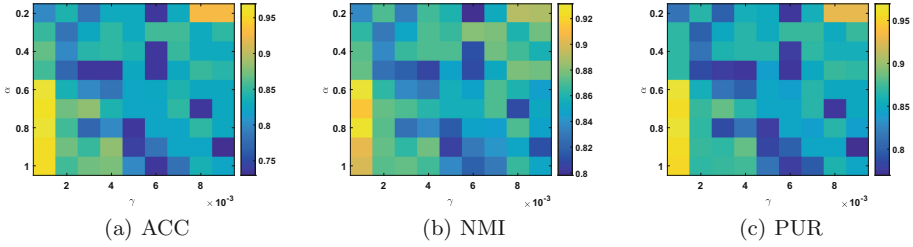


Fig. 4. Parameter analysis on  $\alpha$  and  $\gamma$  on Mfeat.

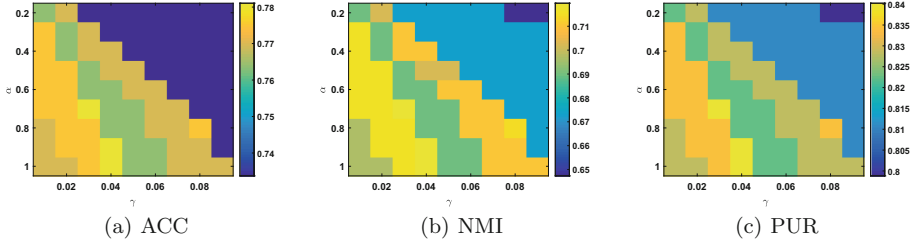


Fig. 5. Parameter analysis on  $\alpha$  and  $\gamma$  on 3Sources.

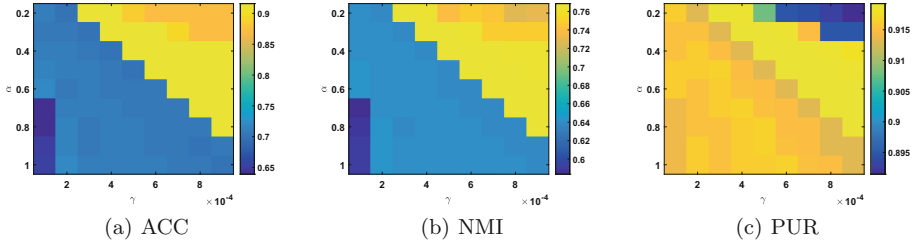


Fig. 6. Parameter analysis on  $\alpha$  and  $\gamma$  on Caltech101-7.

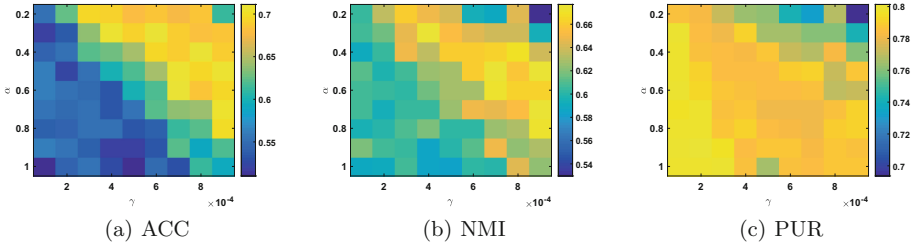


Fig. 7. Parameter analysis on  $\alpha$  and  $\gamma$  on Caltech101-20.

similar performance with similar  $\gamma/\alpha$ . What is more, the value of  $\alpha$  should not be set too large since it may lead to information loss in terms of topological structure. In practice, user can select the value of  $\alpha$  in  $[0.5, 1]$  and the value of  $\gamma$  from  $\{0.01, 0.001, 0.0001\}$  by which satisfactory performance can be obtained.

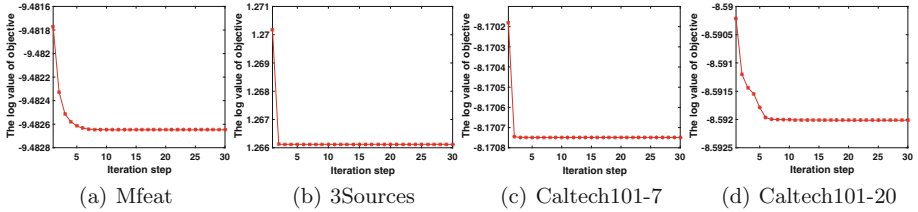


Fig. 8. Convergence analysis of optimization.

### 3.4 Convergence Analysis

In this subsection, convergence analysis is conducted to explore the convergence property of the proposed iterative algorithm by setting  $\alpha = 1$ ,  $\gamma = 0.001$  and  $k = 30$ . Figure 8 plots the log value of objective as a function of iteration step. From the subfigures, we find that the log values of objective decrease rapidly during the iterations on all four datasets. Usually, the algorithm will converge within 30 times of iteration.

### 3.5 Comparison Experiment

In this subsection, we compare the proposed MVPL method with several state-of-the-art algorithms. Two types of clustering methods are used for comparison, namely the traditional single-view clustering methods and the multi-view clustering methods. For the single-view methods, three representative algorithms are selected, namely  $k$ -means (KM) [27], normalized cut (NCut) [19] and robust continuous clustering (RCC) [21]. These single-view methods operate on each individual views from which the best results are reported. For multi-view clustering methods, five state-of-the-art algorithms are used, namely multi-view  $k$ -means (MVKM) [28], multi-view spectral clustering (MVSC) [12], co-training multi-view clustering (CoTrn) [13], co-regularized multi-view clustering (CoReg) [15] and multi-view learning with adaptive neighbors (MLAN) [29]. Following [13], for the methods that generate multiple view-specific clustering results (i.e. CoTrn, CoReg and MVPL), prior knowledge is used to select the most informative view for comparison purpose. For all the spectral-clustering-like compared methods, we use the method mentioned in Sect. 2.2 to construct the proximity matrices, which is shown to be a good method for proximity construction [30]. And the sparsity of the proximity matrices is determined by the number of neighbors  $k$ . We will tune  $k$  in the range of [10, 50] to select the best proximity according to the three measures for all the methods. For all spectral-clustering-like methods and  $k$ -means-like methods, we set the number of clusters  $c$  as the ground-truth number. Besides, for all the methods involving  $k$ -means, we run each algorithm 50 times in the same parameter setting and select the results with the smallest objective as the result for this setting. For all the methods, the best parameters are tuned as suggested by the authors.

**Table 2.** Clustering results in terms of ACC on all datasets.

Method	Mfeat	Caltech101-7	Caltech101-20	3Sources	Rank
KM	0.729	0.463	0.466	0.527	8.5
NCut	0.753	0.646	0.486	0.746	5.0
RCC	0.779	0.761	0.597	0.420	4.6
MVKM	0.738	0.754	0.516	0.775	4.0
MVSC	0.834	0.556	0.445	0.645	6.8
CoTrn	0.833	0.588	0.473	0.734	5.5
CoReg	0.844	0.586	0.497	0.675	4.8
MLAN	0.750	0.707	0.475	0.757	5.0
MVPL	<b>0.970</b>	<b>0.926</b>	<b>0.719</b>	<b>0.781</b>	<b>1.0</b>

**Table 3.** Clustering results in terms of NMI on all datasets.

Method	Mfeat	Caltech101-7	Caltech101-20	3Sources	Rank
KM	0.685	0.459	0.582	0.506	7.8
NCut	0.742	0.521	0.564	0.679	6.0
RCC	0.790	0.621	0.588	0.344	5.5
MVKM	0.650	0.616	0.619	0.587	5.3
MVSC	0.819	0.473	0.551	0.619	6.3
CoTrn	0.846	0.555	0.597	0.696	2.8
CoReg	0.830	0.489	0.596	0.690	4.3
MLAN	0.815	0.544	0.464	0.613	6.3
MVPL	<b>0.932</b>	<b>0.789</b>	<b>0.677</b>	<b>0.720</b>	<b>1.0</b>

**Table 4.** Clustering results in terms of PUR on all datasets.

Method	Mfeat	Caltech101-7	Caltech101-20	3Sources	Rank
KM	0.729	0.875	0.786	0.757	7.3
NCut	0.774	0.891	0.783	0.834	5.3
RCC	0.836	0.876	<b>0.866</b>	0.716	4.8
MVKM	0.738	0.899	0.810	0.781	4.8
MVSC	0.834	0.868	0.764	0.811	6.0
CoTrn	0.857	0.896	0.803	<b>0.846</b>	2.3
CoReg	0.844	0.860	0.788	0.811	5
MLAN	0.778	0.857	0.665	0.793	7.8
MVPL	<b>0.970</b>	<b>0.929</b>	0.803	0.840	<b>2.0</b>

The comparison results obtained by all the methods on the four real-world datasets in terms of ACC, NMI and PUR are reported in Tables 2, 3 and 4 respectively. In the tables, the best performance among all the methods is highlighted in bold. From the tables, we find that the proposed MVPL method outperforms

all the other methods on ACC and NMI. In particular, our method has achieved on average 14% percent improvement in terms of ACC and 12% improvement in terms of NMI on all the datasets. For PUR, although our method cannot reach the highest PUR on all the datasets, it still ranks the first on average. Overall, the comparison results have demonstrated the effectiveness of the proposed method.

## 4 Conclusion

In this paper, we propose a novel proximity learning method for multi-view clustering, called multi-view proximity learning. Through the method, proximities between data objects with multiple views of features can be obtained, which are suitable for multi-view clustering. Accordingly, our method adopts two criteria to fulfill the task, namely intra-view criterion and inter-view criterion. For the intra-view part, we not only make use of the relations between data objects but also take cluster structures into account within individual views. For the inter-view part, we model the correlations between views based on spectral embedding, which utilizes the view consistency and complement properties such that the learning performance is improved. Extensive experiments conducted on both synthetic and real-world datasets demonstrate the effectiveness of our method.

**Acknowledgments.** This work was supported by NSFC (61502543), Guangdong Natural Science Funds for Distinguished Young Scholar (2016A030306014), and Tip-top Scientific and Technical Innovative Youth Talents of Guangdong special support program (2016TQ03X542).

## References

1. Gao, Y., Gu, S., Li, J., Liao, Z.: The multi-view information bottleneck clustering. In: Kotagiri, R., Krishna, P.R., Mohania, M., Nantajeewarawat, E. (eds.) DASFAA 2007. LNCS, vol. 4443, pp. 912–917. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-71703-4\\_78](https://doi.org/10.1007/978-3-540-71703-4_78)
2. Müller, E., Assent, I., Sánchez, I.P., Mülle, Y., Böhm, K.: Outlier ranking via subspace analysis in multiple views of the data. In: 12th IEEE International Conference on Data Mining, pp. 529–538. IEEE (2012)
3. Chen, C., He, J., Bliss, N., Tong, H.: Towards optimal connectivity on multi-layered networks. *IEEE Trans. Knowl. Data Eng.* **29**(10), 2332–2346 (2017)
4. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the 11th Annual Conference on Computational Learning Theory, pp. 92–100 (1998)
5. Amini, M., Usunier, N., Goutte, C.: Learning from multiple partially observed views—an application to multilingual text categorization. *Adv. Neural Inf. Process. Syst.* **22**, 28–36 (2009)
6. Xu, Y.M., Wang, C.D., Lai, J.H.: Weighted multi-view clustering with feature selection. *Pattern Recogn.* **53**, 25–35 (2016)
7. Zhang, G.Y., Wang, C.D., Huang, D., Zheng, W.S.: Multi-view collaborative locally adaptive clustering with Minkowski metric. *Expert Syst. Appl.* **86**, 307–320 (2017)

8. Huang, L., Chao, H.Y., Wang, C.D.: Multi-view intact space clustering. In: Proceedings of the 4th Asian Conference on Pattern Recognition, pp. 500–505 (2017)
9. Xu, R., Wunsch, D.C.: Survey of clustering algorithms. *IEEE Trans. Neural Netw.* **16**(3), 645–678 (2005)
10. Xu, C., Tao, D., Xu, C.: A survey on multi-view learning. CoRR abs/1304.5634 (2013)
11. Xia, T., Tao, D., Mei, T., Zhang, Y.: Multiview spectral embedding. *IEEE Trans. Syst. Man Cybern. Part B* **40**(6), 1438–1446 (2010)
12. Tzortzis, G., Likas, A.: Kernel-based weighted multi-view clustering. In: 12th IEEE International Conference on Data Mining, pp. 675–684 (2012)
13. Kumar, A., Daumé, H.: A co-training approach for multi-view spectral clustering. In: Proceedings of the 28th International Conference on Machine Learning, pp. 393–400 (2011)
14. Son, J.W., Jeon, J., Lee, A., Kim, S.J.: Spectral clustering with brainstorming process for multi-view data. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence, pp. 2548–2554 (2017)
15. Kumar, A., Rai, P., Daumé, H.: Co-regularized multi-view spectral clustering. In: Advances in Neural Information Processing Systems, pp. 1413–1421 (2011)
16. Lu, C., Yan, S., Lin, Z.: Convex sparse spectral clustering: single-view to multi-view. *IEEE Trans. Image Process.* **25**(6), 2833–2843 (2016)
17. Wang, C.D., Lai, J.H., Yu, P.: Multi-view clustering based on belief propagation. *IEEE Trans. Knowl. Data Eng.* **28**(4), 1007–1021 (2016)
18. Xia, R., Pan, Y., Du, L., Yin, J.: Robust multi-view spectral clustering via low-rank and sparse decomposition. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence, pp. 2149–2155 (2014)
19. Luxburg, U.V.: A tutorial on spectral clustering. *Stat. Comput.* **17**(4), 395–416 (2007)
20. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. *Adv. Neural Inf. Process. Syst.* **17**, 1601–1608 (2005)
21. Shah, S.A., Koltun, V.: Robust continuous clustering. *Proc. Nat. Acad. Sci. U.S.A.* **114**(37), 9814 (2017)
22. Nie, F., Wang, X., Huang, H.: Clustering and projected clustering with adaptive neighbors. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 977–986 (2014)
23. Wang, W., Carreira-Perpinán, M.A.: Projection onto the probability simplex: an efficient algorithm with a simple proof, and an application. CoRR abs/1309.1541 (2013)
24. Bache, K., Lichman, M.: UCI machine learning repository (2013). <http://archive.ics.uci.edu/ml/index.php>
25. Li, F.F., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2004, p. 178 (2004)
26. Li, Y., Nie, F., Huang, H., Huang, J.: Large-scale multi-view spectral clustering via bipartite graph. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, pp. 2750–2756 (2015)
27. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297 (1967)



28. Cai, X., Nie, F., Huang, H.: Multi-view  $k$ -means clustering on big data. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, pp. 2598–2604 (2013)
29. Nie, F., Cai, G., Li, X.: Multi-view clustering and semi-supervised classification with adaptive neighbours. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence, pp. 2408–2414 (2017)
30. Nie, F., Wang, X., Jordan, M.I., Huang, H.: The constrained Laplacian rank algorithm for graph-based clustering. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, pp. 1969–1976 (2016)