



Multi-view Similarity Learning of Manifold Data

Rui-rui Wang¹, Si-bao Chen^{1,2(✉)}, Bin Luo¹, and Jian Zhang²

¹ Key Lab of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230601, China
sbchen@ahu.edu.cn

² Peking University Shenzhen Institute, Shenzhen 518057, China

Abstract. In recent years, multi-view learning methods have developed rapidly where graph-based approaches have achieved good performance. Usually, these learning methods construct information graph for each view or fuse different views into one graph. In this paper, a novel multi-view learning model that learns one similarity matrix for all views named Multi-view Similarity Learning (MSL) is proposed, where adaptive weights are learned for each view. The multi-view similarity learning method is further extended to kernel space. Experiments of classification, clustering and semi-supervised classification on different real-world datasets show the effectiveness of the proposed method.

Keywords: Laplacian Eigenmaps · Multi-view learning · Similarity learning

1 Introduction

Usually, in the field of machine learning and pattern recognition, the dimension of data is very high, so how to map high-dimensional data to low-dimensional data and preserve the topology of data is an important issue we are studying now. Then, some classical linear dimensionality reduction methods are proposed, such as Principle Component Analysis (PCA) [5], Linear Discriminant Analysis (LDA) [13], Multidimensional Scaling (MDS) [4]. But the linear dimension reduction method can not represent the manifold structure of data well, so in recent years, many nonlinear dimensionality reduction methods are proposed, such as isometric feature mapping (ISOMAP) [14], local linear embedding (LLE) [8], and Laplacian Eigenmaps (LE) [1]. Laplacian Eigenmaps is a nonlinear dimensionality reduction method for manifold data learning, it can well preserve the nonlinear structure of data space.

This work was supported in part by NSFC Key Projects of International (Regional) Cooperation and Exchanges under Grant 61860206004 and in part by Shenzhen Science & Research Project under Grant JCYJ20170817155854115.

However, in many real world applications, the representation of actual data is not a single form, but can have many forms of expression, such as person's different angles of faces, which can collect different information from different angles. Usually, for each thing, we observe things from different angles that cannot be observed from another angle, so in order to understand more comprehensive information, we need to observe the problem from multiple angles. Nowadays, we can get different feature from one dataset, we think that compared with single feature, multi-view features can get better results. So for classification, clustering and semi-supervised, we think compared with the single-view similarity learning algorithm, multi-view similarity learning can have the higher accuracy.

Today, in many areas of science, such as pattern recognition, computer vision, genetics, and data mining, we can more easily get data that contain heterogeneous features from samples from different perspectives. In visual data, images can be represented by different descriptors. For example, gray features, gabor features [10], local binary patterns (LBP) features [12], and so on. In image processing, the gabor function is a linear filter for edge extraction. Local binary patterns (LBP) were first proposed as an effective texture description operator, and have been widely used due to their excellent rendering ability for image local texture features. LBP features have significant advantages such as gray invariance and rotation invariance. Because LBP features are simple to calculate and have good effects, LBP features have been widely used in many fields of computer vision.

One of the methods to solve the multi-view problem is to connect vectors from different perspectives into a vector and then on the cascaded vector, directly apply the single view clustering algorithm. However, this connection results in overfitting on small samples and has no meaning to the multi-view problem. Our solution to multi-view is to learn a similarity matrix by adding weights to each view. The weights can be updated through each iteration. Our method can be used for classification, clustering and semi-supervised classification.

The k -nearest neighbor (k -NN) [7] classification algorithm is one of the simplest machine learning algorithms. And k is a very important parameter in the k -NN algorithm. The selection of the k value will affect the classification result of the sample to be classified. However, it is hard to choose a proper neighbor number k beforehand, because if the value of k is too small, the model is easily interfered by noisy data, and if k is too large, the prediction ability of the model is greatly weakened. Generally we use cross-validation to select an appropriate k value.

2 Learning Multi-view Similarity in Laplacian Eigenmaps

In this section, we first introduce a nonlinear dimensionality reduction method LE. And then simply illustrate our multi-view similarity learning algorithms.

2.1 Laplacian Eigenmaps

For the success of the graph-based approach, preserving local manifold structures is an important factor. Laplacian Eigenmaps (LE) is a graph-based dimensionality reduction algorithm and it constructs the relationship between data uses a local angle.

Given a set of data points $\{X_1, X_2, \dots, X_n\}$, denote data matrix $X \in \mathcal{R}^{n \times p}$, where n is the number of data points and p is the dimension of features, LE pursues their low dimensional representation $Y_1, Y_2, \dots, Y_n \in \mathcal{R}^q (q < p)$, which constructs a weighted graph with n points as nodes, and a set of weighted edges connecting neighboring points. If the two data instances i and j are very similar, then i and j should be as close as possible after dimensionality reduction.

The steps are as follows:

1. **Constructing the Adjacency Graph:** Construct neighborhood graph \mathcal{G} through k -nearest neighbors algorithm, k is a preset value. Given n data points $\{X_1, X_2, \dots, X_n\}$. Nodes i and j are connected if X_i is among k nearest neighbors of X_j or X_j is among k nearest neighbors of X_i .
2. **Choosing the weights:** Choose edge weights using heat kernel or simply set edge weight to be 1 if connected and 0 otherwise, or we can get similarity matrix S with heat kernel by:

$$S_{ij} = \exp \left\{ -\frac{d_{ij}^2}{2r} \right\} \quad (1)$$

where $d_{ij} = \|X_i - X_j\|$, $r > 0$ is a suitable constant.

3. **Eigenmaps:** Calculate the eigenvectors and eigenvalues of the Laplacian matrix L by:

$$Lv = \lambda Dv \quad (2)$$

where D is diagonal matrix and its entries are row sums of S , $D_{ii} = \sum_j S_{ij}$, Laplacian matrix $L = D - S$. We omitting the eigenvector v_0 and use the next q eigenvectors for embedding in q -dimensional Euclidean space: $X_i \mapsto Y_i = (v_1(i), v_2(i), \dots, v_q(i))^T$.

2.2 Learning New Multi-view Similarity

For multi-view data, the representation X^1, X^2, \dots, X^m is the data matrix for each view. $X^v \in R^{n \times p^v} (v = 1, 2, \dots, m)$, where n is the number of data and p^v is the feature dimension of the v -th view. For graph-based methods, each view can build a similar graph and maximize performance by itself. The similarity between two data points in 1 does not reflect the local popular structure of manifold data, so we add a locally linear reconstruction to sample point by its neighbor points. Then for each view, we propose an effective method is to combine these views

with the appropriate weights $w_v(v = 1, 2, \dots, m)$, so our objective function can be written as

$$\begin{aligned} \min_{S,w} \sum_{v=1}^m w_v^2 \|X^v - X^v S\|_F^2 \\ \text{s.t. } S_{ij} = S_{ji} \geq 0, S_{ii} = 0 \end{aligned} \tag{3}$$

If the distance between sample points are larger, the corresponding reconstruction weight is smaller, and vice versa. We limited reconstruction weight S_{ij} is non-negative, and reconstruction weight is symmetry. Therefore, between sample points we add linear reconstruction constraints to learn the new similarity, so the objective function can be written as

$$\begin{aligned} \min_{S,w} \sum_{v=1}^m w_v^2 \|X^v - X^v S\|_F^2 + \alpha_v \|S - S_0\|_F^2 \\ \text{s.t. } S_{ij} = S_{ji} \geq 0, S_{ii} = 0 \end{aligned} \tag{4}$$

The L_1 paradigm can produce relatively sparse solutions, and has the ability to select features. It is useful when solving high-dimensional feature space. Then the minimization of the final objective function of our learning new similarity turns into

$$\begin{aligned} \min_{S,w} \sum_{v=1}^m (w_v^2 \|X^v - X^v S\|_F^2 + \alpha_v \|S - S_0\|_F^2) + \beta \|S\|_1 \\ \text{s.t. } S_{ij} = S_{ji} \geq 0, S_{ii} = 0 \end{aligned} \tag{5}$$

2.3 Learning Weight for Each View

Where each view shares the same similarity matrix, so we can get a more accurate similarity matrix by adding appropriate weights to each view. We want the distance between the data points in the same class to be as small as possible, so the objective function of weight can be written as

$$\begin{aligned} \min \sum_{v=1}^m w_v^2 d_v \\ \text{s.t. } \sum_{v=1}^m w_v = 1. \end{aligned} \tag{6}$$

The Lagrange function of Eq. (6) can be written as

$$\min \sum_{v=1}^m w_v^2 d_v - \lambda \left(\sum_{v=1}^m w_v - 1 \right). \tag{7}$$

Taking the derivative of Eq. (7) and setting the derivative to zero, we get the iterative formula for w_v is

$$w_v = \frac{(d_v)^{-1}}{\sum_{v=1}^m (d_v)^{-1}} \tag{8}$$

where $d_{ij} = \|X_i - X_j\|$.

3 Algorithms and Analyses

In this section, we respectively analyze the multi-view similarity learning algorithm for mix-signed data and non-negative data.

Algorithm 1. Algorithm of Learning Multi-view Similarity for Mix-Signed Data

Input: $X = \{X^1, X^2, \dots, X^m\}$, $X^v \in R^{n \times p^v}$, $v \in \{1, 2, \dots, m\}$, positive tuning parameter α and β .

Output: An n -by- n similarity matrix S among n training samples.

Initial The weight for each view, $w_v = \frac{1}{v}$, for compute initial similarity matrix S_0^v , with its elements being heat kernels $S_0^v_{ij} = \exp\{-d_v^2_{ij}/2r\}$.

Set $S_{ij}^{(0)} = 1$, $S_{ii}^{(0)} = 0(i, j = 1, 2, \dots, n)$, $t = 0$.

repeat

For each $i, j = 1, 2, \dots, n$, update $S_{ij}^{(t+1)}$ as in (9).

Update weight by $w_v = \frac{(d_v)^{-1}}{\sum_{v=1}^m (d_v)^{-1}}$.

$t = t + 1$.

until $t > \text{MaxIterNum}$.

return $S^{(t)}$.

3.1 Algorithm for Mix-Signed Data

When the elements of the cell array X are mixed with symbols (some are positive and some are negative), we learn the similarity matrix S by the Eq. (9) iterative update formula. Algorithm 1 summarizes the overall similarity learning iterative update algorithm for mix-signed data. In this algorithm, we set $Q = \sum_{v=1}^m (w_v^2 X^{v\top} X^v)$. And in addition to the input cell array X , there are two tuning parameters α and β . In practice, we found that our algorithm is robust to both parameters α and β , so in all experiments in this paper, we only set $\alpha = \beta = 1$.

Theorem 1. *The objective function in Eq. (5) monotonically decreases (ie, does not increase) under the update rule Eq. (9) of Algorithm 1.*

$$S_{ij}^{(t+1)} = S_{ij}^{(t)} \sqrt{\frac{[S^{(t)}Q^- + Q^-S^{(t)}]_{ij} + 2[Q^+ + \alpha_v S^v]_{ij}}{[S^{(t)}Q^+ + Q^+S^{(t)} + 2\sum_{v=1}^m \alpha_v S^{(t)}]_{ij} + 2Q^-_{ij} + \beta}} \quad (9)$$

For proof of Theorem 1, refer to article [2].

3.2 Algorithm for Nonnegative Data

For nonnegative data, we propose a more efficient multi-view similarity learning algorithm to learn the similarity matrix S as in Eq. (10), and we also set $Q = \sum_{v=1}^m (w_v^2 X^{v\top} X^v)$. We summarize the multi-view similarity algorithm for nonnegative data in Algorithm 2. And we only set $\alpha = \beta = 1$ in all the experiments of this paper.

Algorithm 2. Algorithm of Learning Multi-view Similarity for Nonnegative Data

Input: $X = \{X^1, X^2, \dots, X^m\}$, $X^v \in R^{n \times p^v}$, $v \in \{1, 2, \dots, m\}$, positive tuning parameter α and β .

Output: An n -by- n similarity matrix S among n training samples.

Initial The weight for each view, $w_v = \frac{1}{v}$, for compute initial similarity matrix S_0^v , with its elements being heat kernels $S_{ij}^v = \exp\{-d_{ij}^2/2r\}$.

Set $S_{ij}^{(0)} = 1$, $S_{ii}^{(0)} = 0(i, j = 1, 2, \dots, n)$, $t = 0$.

repeat

For each $i, j = 1, 2, \dots, n$, update $S_{ij}^{(t+1)}$ as in (10).

Update weight by $w_v = \frac{(d_v)^{-1}}{\sum_{v=1}^m (d_v)^{-1}}$.

$t = t + 1$.

until $t > \text{MaxIterNum}$.

return $S^{(t)}$.

Theorem 2. For nonnegative data, the objective function in Eq. (5) decreases monotonically (i.e. it is non-increasing) under the update rule Eq. (10) in Algorithm 2.

$$S_{ij}^{(t+1)} = S_{ij}^{(t)} \sqrt{\frac{2[Q + \alpha_v S^v]_{ij}}{[S^{(t)}Q + QS^{(t)} + 2\sum_{v=1}^m \alpha_v S^{(t)}]_{ij} + \beta}} \tag{10}$$

For proof of Theorem 2, refer to article [2].

4 Learning Multi-view Similarity in Kernel Spaces

The role of the kernel function is to imply a mapping from low-dimensional space to high-dimensional space \mathcal{F} , then in \mathcal{F} space, we learn new similarity for data.

For each view, we use a nonlinear map: $\phi : \mathcal{R}^{p^v} \rightarrow \mathcal{F}^v$, $X_i^v \rightarrow \phi(X_i^v)$, the mapped data matrix is $\phi(X)^v = [\phi(X_1^v), \phi(X_2^v), \dots, \phi(X_n^v)]$. So the minimization objective function becomes

$$\min_{S, w} \sum_{v=1}^m (w_v^2 \|\phi(X)^v - \phi(X)^v S\|_F^2 + \alpha_v \|S - S_0^v\|_F^2) + \beta \|S\|_1 \tag{11}$$

In this paper, four kernel functions are mainly used, include linear kernel, gaussian kernel, cosine kernel and polynomial kernel. In implementation, the mapping ϕ does not need to be computed explicitly. By choosing a proper kernel function k , The ϕ mapping and \mathcal{F} space can determined implicitly by the dot product between two mapped data samples $\phi(X_i)^v$ and $\phi(X_j)^v$ in \mathcal{F} space by

$$k(X_i^v, X_j^v) = (\phi(X_i)^v \cdot \phi(X_j)^v) \tag{12}$$

Note that most of kernel functions are nonnegative, such as gaussian kernel and cosine kernel. So we replacing $\phi(X)^v \top \phi(X)^v$ with kernel matrix K^v in the iterative updating (10) for non-negative data, then we can get

$$S_{ij}^{(t+1)} = S_{ij}^{(t)} \frac{2[\sum_{v=1}^m ((w_v^2 K^v) + \alpha_v S_0^v)]_{ij}}{[S^{(t)} \sum_{v=1}^m (w_v^2 K^v) + \sum_{v=1}^m (w_v^2 K^v) S_0^v + 2 \sum_{v=1}^m \alpha_v S_0^v]_{ij} + \beta} \quad (13)$$

We replacing (10) in Algorithm 2 for nonnegative data with (13), we can obtain the multi-view similarity learning algorithm in kernel space.

5 Experiments

In this section, we first introduce the data set we used. And then we will perform the proposed method on many benchmark datasets and compare it with other related graph-based multi-view learning methods. In the following experiments, we will learn new multi-view similarity matrix with Algorithm 2.

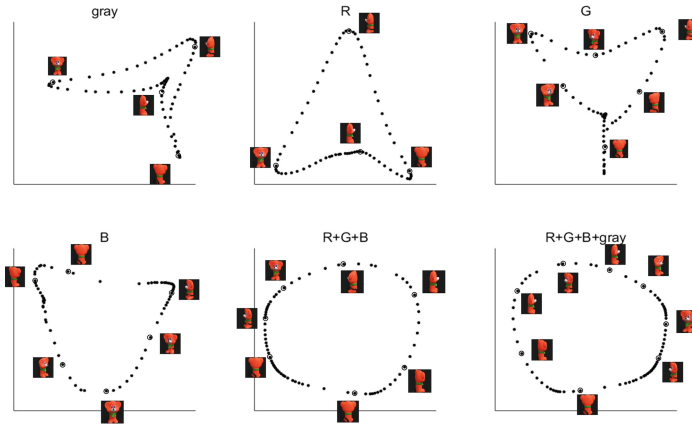


Fig. 1. Images of toy tiger mapped into the embedding space described by the two coordinates of MSL. Different angles of tiger are shown next to circled points in different parts of the space.

5.1 Brief Description of Data Sets

ORL¹ data set include 400 images with 40 classes. We extract three visual features from each image: gray feature with dimension 4,096, gabor feature with dimension 2,560, and LBP feature with dimension 3,776.

AR² data set include 3,120 images with 120 class. We extract three visual features from each image: gray feature with dimension 2,000, gabor feature with dimension 3,200, and LBP feature with dimension 4,720.

¹ <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>.

² <http://www.pudn.com/Download/item/id/2427991.html>.

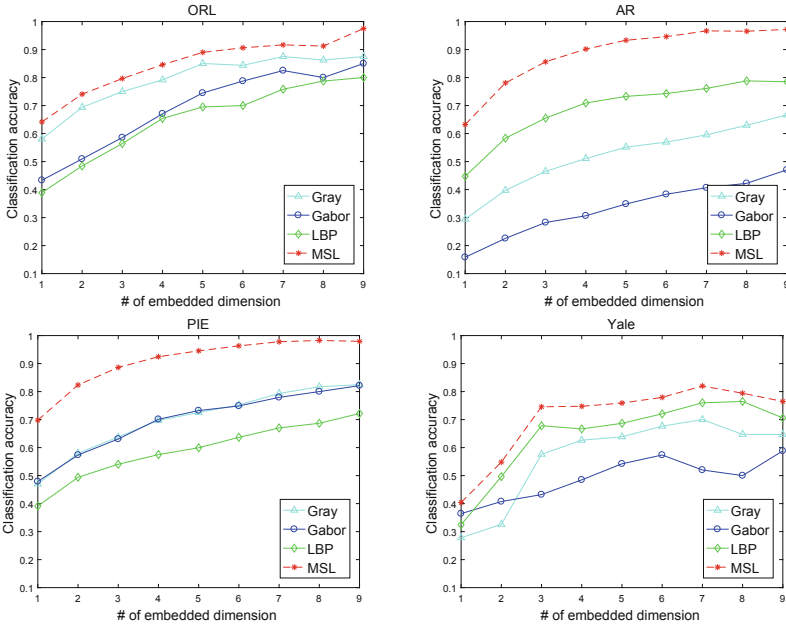


Fig. 2. Classification accuracy of MSL at four datasets.

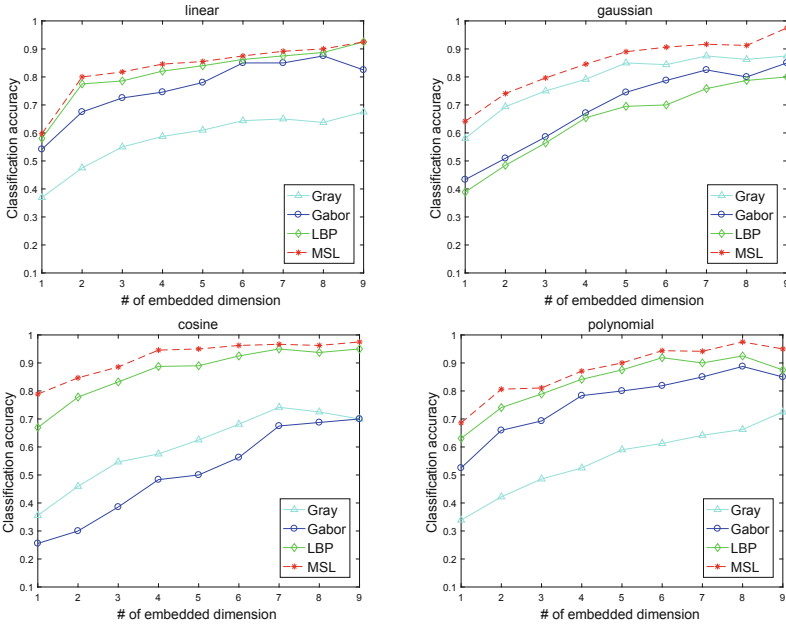


Fig. 3. Classification accuracy of MSL with different kernel functions at ORL dataset.

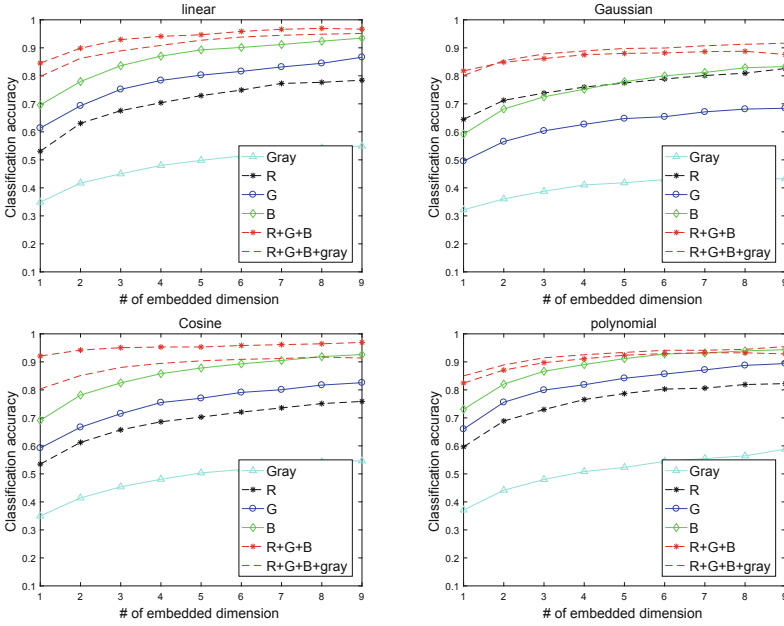


Fig. 4. Classification accuracy of MSL with different kernel functions at COIL-100 dataset.

PIE³ data set is a face data set, we select its subset pose27 that include 1,440 images with 20 class. We extract three visual features from each image: gray feature with dimension 1,024, gabor feature with dimension 640, and LBP feature with dimension 944.

Yale⁴ data set include 166 images with 15 class. We extract three visual features from each image: gray feature with dimension 4,096, gabor feature with dimension 2,560, and LBP feature with dimension 3,776.

COIL20⁵ data include 1,440 images with 20 class. We extract three visual features from each image: gray feature with dimension 1,024, gabor feature with dimension 640, and local binary pattern (LBP) with dimension 944.

Handwritten numerals⁶ (HW) data set is comprised of 2,000 data points for 0 to 9 digit classes, 200 data points for each class. We extract three visual features from each image: gray feature with dimension 256, gabor feature with dimension 160, and LBP feature with dimension 236.

The COIL100⁷ data set contains 7,200 colorized images corresponding to 100 different objects in 72 different viewpoints. We extracted 7 features from each

³ <http://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html>.

⁴ <http://vision.ucsd.edu/content/yale-face-database>.

⁵ <http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php>.

⁶ <https://archive.ics.uci.edu/ml/datasets/Multiple+Features>.

⁷ <http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php>.

image, including the gray scale features features and other six channels in the RGB and HSV channels, all of which are characterized by 16,384 dimensions.

Caltech101⁸ data set is containing 101 categories of images. We select 1,474 images with 7 classes, include Dolla-Bill, Face, Garfield, Motorbikes, Snoopy, Stop-Sign and Windsor-Chair. Six features are extracted from all the images, include 48 dimension gabor feature, 40 dimension wavelet moments, 254 dimension CENTRIST feature, 1,984 dimension HOG feature, 512 dimension GIST feature, and 928 dimension LBP feature.

5.2 Low Dimensional Embedding

In this part, we use one class of COIL-100 datasets, this class containing 72 images at different angles, we embedding high-dimensional images data into low-dimensional space. We can see in Fig. 1, these 72 images are rotated at different angles, so when embedding to a two-dimensional space, the closer to a circle. We extracted the gray feature and RGB features of each channel, and found that the performance after fusion is better than that of a single channel features. Figure 1 showing the results of a single channel and channel fusion, and there is a corresponding picture next to each point.

Figure 1 shows the 2-*D* embedding results of single-view and multi-view with each row corresponding to one manifold benchmark. From the figure, we can see our method MSL is more robust and can effectively find the proper low-dimensional embedding.

5.3 Classification of MSL

In this section, in order to validate the performance of the proposed method, we apply our method into multi-view classification. We used the indicator accuracy (ACC) [6] to evaluate the performance of the algorithm on four benchmark datasets. The four datasets we used are ORL, Yale, AR, and PIE. Each database is randomly divided into a training set and a test set, with different numbers of images being used for training.

The neighbor number k in computing heat kernel similarity of MSL are tuned such that MSL reach its best classification performance. We set the parameters $\alpha = \beta = 1$. The classification accuracy is computed by the nearest neighbor classifier.

To test the performance of the multi-view similarity learning in kernel spaces computed in (13), we test the classification accuracy of single-view and multi-view new similarity learning method with different kernel functions on ORL database. In this paper, four kernel functions (linear, polynomial, cosine, and Gaussian) are adopted in the experiments. We test MSL with tenfold cross-validation [3] as different number of embedded dimension is chosen. Figure 2 shows the classification accuracy of the single-view feature on the four data sets and the classification accuracy after the fusion of the three features. Figure 3

⁸ <http://www.vision.caltech.edu/feifeili/Datasets.htm>.

Table 1. Clustering accuracy of MSL on two datasets

Database	Yale	Caltech101
SC(1)	0.3675	0.3460
SC(2)	0.3313	0.4480
SC(3)	0.3375	0.5290
SC(4)	–	0.6070
SC(5)	–	0.6720
SC(6)	–	0.5910
MVSC	0.6050	0.7250
MLAN	0.6325	0.7800
MSL	0.6506	0.8433

shows the classification performance variations of MSL with different kernel functions, and the classification performance variations of single-view and multi-view. And from the Fig. 3, We can see that classification performance of multi-view is better than classification performance of single-view on any kernel. From the Fig. 4, we extracted the four features include gray feature and features of the RGB three channels of the picture and then calculate the classification accuracy of each channel, and classification accuracy of fusion with multiple channels, we can see the classification accuracy after fusion is higher than the classification accuracy of a single channel.

5.4 Clustering of MSL

In this section, we compared our method with other two multi-view learning methods, one is Multi-view Spectral Clustering (MVSC) [9] and the other is Multi-view Learning with Adaptive Neighbours (MLAN) [11], Table 1 show the clustering result in terms of accuracy of different method in two different datasets, SC is single-view feature. We can see our clustering results are better than the other two methods.

5.5 Semi-supervised Classification of MSL

In this section, we compared our method with MLAN [11] method, and in terms of semi-supervised classification, we choose the front 20% data as labeled sample, Table 2 show the semi-supervised classification performance of different method in three different datasets, we can see that our method is better than other method.

Table 2. Semi-supervised classification accuracy of MSL on three datasets

Database	ORL	COIL20	HW
SC(1)	0.5500	0.7207	0.9250
SC(2)	0.7750	0.8690	0.8040
SC(3)	0.7813	0.5474	0.7630
SC(4)	–	–	0.6930
SC(5)	–	–	0.6910
SC(6)	–	–	0.4730
MLAN	0.8313	0.9716	0.9760
MSL	0.8700	0.9920	0.9813

6 Conclusion and Remarks

In this paper, we introduce a novel multi-view similarity learning method named MSL, and our method can preserve the manifold structure of the data. For multi-view learning, our method can automatically learn weights for each view. The experimental results on real world benchmark data sets demonstrate that the classification accuracy of multi-view features is higher than that of single-view feature. And in clustering and semi-supervised classification, the accuracy of our method is higher than that of other multi-view methods. These experimental results demonstrate the effectiveness of our method.

References

1. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.* **15**, 1373–1396 (2003)
2. Chen, S., Ding, C.H.Q., Luo, B.: Similarity learning of manifold data. *IEEE Trans. Cybern.* **45**(9), 1744–1756 (2015)
3. Chik, Z., Aljanabi, Q.A., Kasa, A., Taha, M.R.: Tenfold cross validation artificial neural network modeling of the settlement behavior of a stone column under a highway embankment. *Arab. J. Geosci.* **7**(11), 4877–4887 (2014)
4. Cox, M.A.A., Cox, T.F.: Multidimensional scaling. *J. R. Stat. Soc.* **46**(2), 1050–1057 (2001)
5. Debruyne, M., Verdonck, T.: Robust kernel principal component analysis and classification. *Adv. Data Anal. Classif.* **4**(2–3), 151–167 (2010)
6. Diebold, F.X., Mariano, R.S.: Comparing predictive accuracy. *J. Bus. Econ. Stat.* **13**(1), 134–144 (1995)
7. Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K.: KNN model-based approach in classification. In: Meersman, R., Tari, Z., Schmidt, D.C. (eds.) OTM 2003. LNCS, vol. 2888, pp. 986–996. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39964-3_62
8. Hsieh, P., Yang, M., Gu, Y., Liang, Y.: Classification-oriented locally linear embedding. *IJPRAI* **24**(5), 737–762 (2010)

9. Kumar, A., III, H.D.: A co-training approach for multi-view spectral clustering. In: Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, 28 June–2 July 2011, pp. 393–400 (2011)
10. Liu, C., Wechsler, H.: Independent component analysis of gabor features for face recognition. *IEEE Trans. Neural Netw.* **14**(4), 919–928 (2003)
11. Nie, F., Cai, G., Li, X.: Multi-view clustering and semi-supervised classification with adaptive neighbours. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, pp. 2408–2414 (2017)
12. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(7), 971–987 (2002)
13. Wang, X., Tang, X.: Dual-space linear discriminant analysis for face recognition. In: 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 564–569 (2004)
14. Yang, M.-H.: Discriminant isometric mapping for face recognition. In: Crowley, J.L., Piater, J.H., Vincze, M., Paletta, L. (eds.) ICVS 2003. LNCS, vol. 2626, pp. 470–480. Springer, Heidelberg (2003). <https://doi.org/10.1007/3-540-36592-3-45>