

Similarity Learning via Kernel Preserving Embedding

Zhao Kang,^{1*} Yiwei Lu,^{1,2} Yuanzhang Su,³ Changsheng Li,¹ Zenglin Xu^{1*}

¹School of Computer Science and Engineering, University of Electronic Science and Technology of China, China

²Department of Computer Science, University of Manitoba, 66 Chancellors Cir, Winnipeg, MB R3T 2N2, Canada

³School of Foreign Languages, University of Electronic Science and Technology of China, Sichuan 611731, China

Zkang@uestc.edu.cn, luy2@cs.umanitoba.ca, syz@uestc.edu.cn, lichangsheng@uestc.edu.cn, zlxu@uestc.edu.cn

Abstract

Data similarity is a key concept in many data-driven applications. Many algorithms are sensitive to similarity measures. To tackle this fundamental problem, automatically learning of similarity information from data via self-expression has been developed and successfully applied in various models, such as low-rank representation, sparse subspace learning, semi-supervised learning. However, it just tries to reconstruct the original data and some valuable information, e.g., the manifold structure, is largely ignored. In this paper, we argue that it is beneficial to preserve the overall relations when we extract similarity information. Specifically, we propose a novel similarity learning framework by minimizing the reconstruction error of kernel matrices, rather than the reconstruction error of original data adopted by existing work. Taking the clustering task as an example to evaluate our method, we observe considerable improvements compared to other state-of-the-art methods. More importantly, our proposed framework is very general and provides a novel and fundamental building block for many other similarity-based tasks. Besides, our proposed kernel preserving opens up a large number of possibilities to embed high-dimensional data into low-dimensional space.

Introduction

Nowadays, high-dimensional data can be collected everywhere, either by low-cost sensors or from the internet (Chen et al. 2012). Extracting useful information from massive high-dimensional data is critical in different areas like text, images, videos and more. Data similarity is especially important since it is the input for a number of data analysis tasks, such as spectral clustering (Ng et al. 2002; Chen et al. 2018), nearest neighbor classification (Weinberger, Blitzer, and Saul 2005), image segmentation (Li et al. 2016), person re-identification (Hirzer et al. 2012), image retrieval (Hoi, Liu, and Chang 2008), dimension reduction (Passalis and Tefas 2017), and graph-based semi-supervised learning (Kang et al. 2018a). Therefore, similarity measure is crucial to the performance of many techniques and is a fundamental problem in machine learning, pattern recognition, and data mining communities (Gao et al. 2017;

Towne, Rosé, and Herbsleb 2016). A variety of similarity metrics, e.g., Cosine, Jaccard coefficient, Euclidean distance, Gaussian function, are often used in practice for convenience. However, they are often data-dependent and sensitive to noise (Huang, Nie, and Huang 2015). Consequently, different metrics lead to a big difference in the final results. In addition, several other similarity measure strategies are popular in dimension reduction techniques. For example, in the widely used locally linear embedding (LLE) (Roweis and Saul 2000), isomeric feature mapping (ISOMAP) (Tenenbaum, De Silva, and Langford 2000), and locality preserving projection (LPP) (Niyogi 2004) methods, one has to construct an adjacency graph of neighbors. Then, k -nearest-neighborhood (knn) and ϵ -nearest-neighborhood graph construction methods are often utilized. These approaches also have some inherent drawbacks, including 1) how to determine neighbor number k or radius ϵ ; 2) how to choose an appropriate similarity metric to define neighborhood; 3) how to counteract the adverse effect of noise and outliers; 4) how to tackle data with structures at different scales of size and density. Unfortunately, all these factors heavily influence the subsequent tasks (Kang et al. 2018b).

Recently, automatically learning of similarity information from data has drawn significant attention. In general, it can be classified into two categories. The first one is adaptive neighbors approach. It learns similarity information by assigning a probability for each data point as the neighborhood of another data point (Nie, Wang, and Huang 2014). It has been shown to be an effective way to capture the local manifold structure.

The other one is self-expression approach. The basic idea is to represent every data point by a linear combination of other data points. In contrast, LLE reconstructs the original data by expressing each data point as a linear combination of its k nearest neighbors only. Through minimizing this reconstruction error, we can obtain a coefficient matrix, which is also named similarity matrix. It has been widely applied in various representation learning tasks, including sparse subspace clustering (Elhamifar and Vidal 2013; Peng et al. 2016), low-rank representation (Liu et al. 2013), multi-view learning (Tao et al. 2017), semi-supervised learning (Zhuang et al. 2017), nonnegative matrix factorization(NMF) (Zhang et al. 2017).

However, this approach just tries to reconstruct the origi-

*Corresponding author.

nal data and has no explicit mechanism to preserve manifold structure information about the data. In many applications, the data can display structures beyond simply being low-rank or sparse. It is well-accepted that it is essential to take into account structure information when we perform high-dimensional data analysis. For instance, LLE preserves the local structure information.

In view of this issue with the current approaches, we propose to learn the similarity information through reconstructing the original data kernel matrix, which is supposed to preserve overall relations. By doing so, we expect to obtain more accurate and complete data similarity. Considering clustering as a specific application of our proposed similarity learning method, we demonstrate that our framework provides impressive performance on several benchmark data sets. In summary, the main contributions of this paper are threefold:

- Compared to other approaches, the use of the kernel-based distances allows to work on preserving the sets of overall relations rather than individual pairwise similarities.
- Similarity preserving provides a fundamental building block to embed high-dimensional data into low-dimensional latent space. It is general enough to be applied to a variety of learning problems.
- We evaluate the proposed approach in the clustering task. It shows that our algorithm enjoys superior performance compared to many state-of-the-art methods.

Notations. Given a data set $\{x_1, x_2, \dots, x_n\}$, we denote $X \in \mathcal{R}^{m \times n}$ with m features and n instances. Then the (i, j) -th element of matrix X are denoted by x_{ij} . The ℓ_2 -norm of a vector x is represented by $\|x\| = \sqrt{x^T \cdot x}$, where T denotes transpose. The ℓ_1 -norm of X is defined as $\|X\|_1 = \sum_{ij} |x_{ij}|$. The squared Frobenius norm is represented as $\|X\|_F^2 = \sum_{ij} x_{ij}^2$. The nuclear norm of X is $\|X\|_* = \sum \sigma_i$, where σ_i is the i -th singular value of X . I is the identity matrix with a proper size. $\vec{1}$ represents a column vector whose every element is one. $Z \geq 0$ means all the elements of Z are nonnegative. Inner product $\langle x_i, x_j \rangle = x_i^T \cdot x_j$.

Related Work

In this section, we provide a brief review of existing automatic similarity learning techniques.

Adaptive Neighbors Approach

In a similar spirit of LPP, for each data point x_i , all the data points $\{x_j\}_{j=1}^n$ can be regarded as the neighborhood of x_i with probability z_{ij} . To some extent, z_{ij} represents the similarity between x_i and x_j (Nie, Wang, and Huang 2014). The smaller distance $\|x_i - x_j\|^2$ is, the greater the probability z_{ij} is. Rather than prespecifying Z with the deterministic neighborhood relation as LPP does, one can adaptively learn

Z from the data set by solving an optimization problem:

$$\min_{z_i} \sum_{j=1}^n (\|x_i - x_j\|^2 z_{ij} + \alpha z_{ij}^2) \text{ s.t. } z_i^T \vec{1} = 1, 0 \leq z_{ij} \leq 1, \quad (1)$$

where α is the regularization parameter. Recently, a variety of algorithms have been developed by using Eq. (1) to learn a similarity matrix. Some applications are clustering (Nie, Wang, and Huang 2014), NMF (Huang et al. 2018), and feature selection (Du and Shen 2015). This approach can effectively capture the local structure information.

Self-expression Approach

The so-called self-expression is to approximate each data point as a linear combination of other data points, i.e., $x_i = \sum_j x_j z_{ij}$. The rationale here is that if x_i and x_j are similar, weight z_{ij} should be big. Therefore, Z also behaves like the similarity matrix. This shares the similar spirit as LLE, except that we do not predetermine the neighborhood. Its corresponding learning problem is:

$$\min_Z \frac{1}{2} \|X - XZ\|_F^2 + \alpha \rho(Z) \text{ s.t. } Z \geq 0, \quad (2)$$

where $\rho(Z)$ is a regularizer of Z . Two commonly used assumptions about Z are low-rank and sparse. Hence, in many domains, we also call Z as the low-dimensional representation of X . Through this procedure, the individual pairwise similarity information hidden in the data is explored (Nie, Wang, and Huang 2014) and the most informative “neighbors” for each data point are automatically chosen.

Moreover, this learned Z can not only reveal low-dimensional structure of data, but also be robust to data scale (Huang, Nie, and Huang 2015). Therefore, this approach has drawn significant attention and achieved impressive performance in a number of applications, including face recognition (Zhang, Yang, and Feng 2011), subspace clustering (Liu et al. 2013; Elhamifar and Vidal 2013), semi-supervised learning (Zhuang et al. 2017). In many real-world applications, data often present complex structures. Nevertheless, the first term in Eq. (2) simply minimizes the reconstruction error. Some important manifold structure information, such as overall relations, could be lost during this process. Preserving relation information has been shown to be important for feature selection (Zhao et al. 2013). In (Zhao et al. 2013), new feature vector f is obtained by maximizing $f^T \hat{K} f$, where \hat{K} is the refined similarity matrix derived from original kernel matrix K with element $K(x, y) = \phi(x)^T \phi(y)$. In this paper, we propose a novel model to preserve the overall relations of the original data and simultaneously learn the similarity matrix.

Proposed Methodology

Since our goal is to obtain similarity information, it is very necessary to retain the overall relations among the data samples when we build a new representation. However, Eq. (2) just tries to reconstruct the original data and does not take overall relations information into account. Our objective is

finding a new representation which preserves overall relations as much as possible.

Given a data matrix X , one of the most commonly used relation measures is the inner product. Specifically, we try to minimize the inconsistency between two inner products: one for the raw data and another for reconstructed data XZ . To make our model more general, we build it in a transformed space, i.e., X is mapped by ϕ (Xu et al. 2009). We have

$$\min_Z \|\phi(X)^T \cdot \phi(X) - (\phi(X)Z)^T \cdot (\phi(X)Z)\|_F^2 \quad (3)$$

(3) can be simplified as

$$\min_Z \|K - Z^T K Z\|_F^2. \quad (4)$$

With certain assumption about the structure of Z , our proposed Similarity Learning via Kernel preserving Embedding (SLKE) framework can be formulated as

$$\min_Z \frac{1}{2} \|K - Z^T K Z\|_F^2 + \gamma \rho(Z) \quad s.t. \quad Z \geq 0, \quad (5)$$

where $\gamma > 0$ is a tradeoff parameter and ρ is a regularizer on Z . If we use the nuclear norm $\|\cdot\|_*$ to replace $\rho(\cdot)$, we have a low-rank representation. If the ℓ_1 -norm is adopted, we obtain a sparse representation. It is worth pointing out that Eq. (5) enjoys several nice properties:

- 1) The use of kernel-based distance preserves the sets of overall relations, which will benefit the subsequent tasks;
- 2) This learned low-dimensional representation or similarity matrix Z is general enough to be utilized to solve a variety of different tasks, where similarity information is needed;
- 3) The learned representation is particularly suitable to problems that are sensitive to data similarity, such as clustering (Kang et al. 2018c), classification (Wright et al. 2009), recommender systems (Kang, Peng, and Cheng 2017a);
- 4) Its input is the kernel matrix. This is also desirable, as not all types of data can be represented in numerical feature vectors form (Xu et al. 2010). For instance, we need to group proteins in bioinformatics based on their structures and to divide users in social media based on their friendship relations.

In the following section, we will show a simple strategy to solve problem (5).

Optimization

It is easy to see that Eq. (5) is a fourth-order function of Z . Directly solving it is not so straightforward. To circumvent this problem, we first convert it to the following equivalent problem by introducing two more auxiliary variables

$$\min_Z \frac{1}{2} \|K - J^T K W\|_F^2 + \gamma \rho(Z) \quad (6)$$

$$s.t. \quad Z \geq 0, \quad Z = J, \quad Z = W.$$

Now we resort to the alternating direction method of multipliers (ADMM) method to solve (6). The corresponding augmented Lagrangian function is:

$$\mathcal{L}(Z, J, W, Y_1, Y_2) = \frac{1}{2} \|K - J^T K W\|_F^2 + \gamma \rho(Z) + \frac{\mu}{2} \left(\|Z - J + \frac{Y_1}{\mu}\|_F^2 + \|Z - W + \frac{Y_2}{\mu}\|_F^2 \right), \quad (7)$$

where $\mu > 0$ is a penalty parameter and Y_1, Y_2 are the Lagrangian multipliers. The variables Z, W , and J can be updated alternately, one at each step, while keeping the other two fixed.

To solve J , we observe that the objective function (7) is a strongly convex quadratic function in J which can be solved by setting its first derivative to zero, we have:

$$J = (\mu I + K W W^T K^T)^{-1} (\mu Z + Y_1 + K W K^T), \quad (8)$$

where $I \in \mathcal{R}^{n \times n}$ is the identity matrix.

Similarly,

$$W = (\mu I + K^T J J^T K)^{-1} (\mu Z + Y_2 + K^T J K). \quad (9)$$

For Z , we have the following subproblem:

$$\min_Z \gamma \rho(Z) + \mu \left\| Z - \frac{J + W - \frac{Y_1 + Y_2}{\mu}}{2} \right\|_F^2. \quad (10)$$

Depending on different regularization strategies, we have different closed-form solutions for Z . Define $H = \frac{J + W - \frac{Y_1 + Y_2}{\mu}}{2}$, we can write its singular value decomposition (SVD) as $U \text{diag}(\sigma) V^T$. Then, for low-rank representation, i.e., $\rho(Z) = \|Z\|_*$, we have,

$$Z = U \text{diag}(\max\{\sigma - \frac{\gamma}{2\mu}, 0\}) V^T. \quad (11)$$

For sparse representation, i.e., $\rho(Z) = \|Z\|_1$, we can update Z element by element as,

$$Z_{ij} = \max\{|H_{ij}| - \frac{\gamma}{2\mu}, 0\} \cdot \text{sign}(H_{ij}). \quad (12)$$

For clarity, the complete procedures to solve the problem (5) are outlined in Algorithm 1.

Algorithm 1: The algorithm of SLKE

Input: Kernel matrix K , parameters $\gamma > 0, \mu > 0$.
Initialize: Random matrix W and $Z, Y_1 = Y_2 = 0$.
REPEAT

- 1: Calculate J by (8).
- 2: Update W according to (9).
- 3: Calculate Z using (11) or (12).
- 4: Update Lagrange multipliers Y_1 and Y_2 as

$$Y_1 = Y_1 + \mu(Z - J),$$

$$Y_2 = Y_2 + \mu(Z - W).$$

UNTIL stopping criterion is met.

Complexity Analysis

With our optimization strategy, the complexity for J is $\mathcal{O}(n^3)$. Updating W has the same complexity as J . Both J and W involve matrix inverse. Fortunately, we can avoid it by resorting to some approximation techniques when we face large-scale data sets. Depending on the choice of regularizer, we have different complexity for Z . For low-rank representation, it requires an SVD for every iteration and

its complexity is $\mathcal{O}(n^3)$. Since we seek a low-rank matrix and so only need a few principle singular values. Package like PROPACK can compute a rank k SVD with complexity $\mathcal{O}(n^2k)$ (Larsen 2004). To obtain a sparse solution of Z , we need $\mathcal{O}(n^2)$ complexity. The updating of Y_1 and Y_2 cost $\mathcal{O}(n^2)$.

Experiments

To assess the effectiveness of our proposed method, we apply the learned similarity matrix to do clustering.

Data Sets

Table 1: Description of the data sets

	# instances	# features	# classes
YALE	165	1024	15
JAFFE	213	676	10
ORL	400	1024	40
COIL20	1440	1024	20
BA	1404	320	36
TR11	414	6429	9
TR41	878	7454	10
TR45	690	8261	10
TDT2	9394	36771	30

We conduct our experiments with nine benchmark data sets, which are widely used in clustering experiments. We show the statistics of these data sets in Table 1. In summary, the number of data samples varies from 165 to 9,394 and feature number ranges from 320 to 36,771. The first five data sets are images, while the last four are text data.

Specifically, the five image data sets contain three face databases (ORL, YALE, and JAFFE), a toy image database COIL20, and a binary alpha digits data set BA. For example, COIL20 consists of 20 objects and each object was taken from different angles. BA data set contains images of digits of “0” through “9” and letters of capital “A” through “Z”. YALE, ORL, and JAFFE consist of images of the person. Each image represents different facial expressions or configurations due to times, illumination conditions, and glasses/no glasses.

Data Preparation

Since the input for our proposed method is kernel matrix, we design 12 kernels in total to fully examine its performance. They are: seven Gaussian kernels of the form $K(x, y) = \exp(-\|x - y\|_2^2 / (td_{max}^2))$ with $t \in \{0.01, 0.05, 0.1, 1, 10, 50, 100\}$, where d_{max} denotes the maximal distance between data points; a linear kernel $K(x, y) = x^T y$; four polynomial kernels $K(x, y) = (a + x^T y)^b$ of the form with $a \in \{0, 1\}$ and $b \in \{2, 4\}$. Besides, all kernels are rescaled to $[0, 1]$ by dividing each element by the largest element in its corresponding kernel matrix. These kernels are commonly used types in the literature, so we can well investigate the performance of our method.

Comparison Methods

To fully examine the effectiveness of the proposed framework on clustering, we choose a good set of methods to compare. In general, they can be classified into two categories: similarity-based and kernel-based clustering methods.

- **Spectral Clustering (SC)** (Ng et al. 2002): SC is a widely used clustering method. It enjoys the advantage of exploring the intrinsic data structures. Its input is the graph Laplacian, which is constructed from the similarity matrix. Here, we directly treat kernel matrix as the similarity matrix for spectral clustering. For our proposed SLKE method, we employ learned Z to do spectral clustering. Thus, SC serves as a baseline method.
- **Robust Kernel K-means (RKKM)**¹ (Du et al. 2015): Based on classical k-means clustering algorithm, RKKM has been developed to deal with nonlinear structures, noise, and outliers in the data. RKKM demonstrates superior performance on a number of real-world data sets.
- **Simplex Sparse Representation (SSR)** (Huang, Nie, and Huang 2015): SSR method has been proposed recently. It is based on adaptive neighbors idea. Another appealing property of this method is that its model parameter can be calculated by assuming a maximum number of neighbors. Therefore, we don’t need to tune the parameter any more. In addition, it outperforms many other state-of-the-art techniques.
- **Low-Rank Representation (LRR)** (Liu et al. 2013): Based on self-expression, subspace clustering with low-rank regularizer achieves great success on a number of applications, such as face clustering, motion segmentation.
- **Sparse Subspace Clustering (SSC)** (Elhamifar and Vidal 2013): Similar to LRR, SSC assumes a sparse solution of Z . Both LRR and SSC learn similarity matrix by reconstructing the original data. In this aspect, SC, LRR, and SSC are baseline methods w.r.t. our proposed algorithm.
- **Clustering with Adaptive Neighbor (CAN)** (Nie, Wang, and Huang 2014). Based on the idea of adaptive neighbors, i.e., Eq.(1), CAN learns a local graph from raw data for clustering task.
- **Twin Learning for Similarity and Clustering (TLSC)** (Kang, Peng, and Cheng 2017b): Recently, TLSC has been proposed and has shown promising results on real-world data sets. TLSC does not only learn similarity matrix via self-expression in kernel space, but also have optimal similarity graph guarantee. Besides, it has good theoretical properties, i.e., it is equivalent to kernel k-means and k-means under certain conditions.
- **SLKE**: Our proposed similarity learning method with overall relations preserving capability. After obtaining similarity matrix Z , we use spectral clustering to conduct clustering experiments. We test both low-rank and sparse

¹<https://github.com/csliangdu/RMKKM>

regularizer. We denote them as SLKE-R and SLKE-S, respectively².

Evaluation Metrics

To quantitatively and effectively assess the clustering performance, we utilize the two widely used metrics (Peng et al. 2018), accuracy (Acc) and normalized mutual information (NMI).

Acc discovers the one-to-one relationship between clusters and classes. Let l_i and \hat{l}_i be the clustering result and the ground truth cluster label of x_i , respectively. Then the Acc is defined as

$$Acc = \frac{\sum_{i=1}^n \delta(\hat{l}_i, \text{map}(l_i))}{n},$$

where n is the sample size, Kronecker delta function $\delta(x, y)$ equals one if and only if $x = y$ and zero otherwise, and $\text{map}(\cdot)$ is the best permutation mapping function that maps each cluster index to a true class label based on Kuhn-Munkres algorithm.

Given two sets of clusters L and \hat{L} , NMI is defined as

$$NMI(L, \hat{L}) = \frac{\sum_{l \in L, \hat{l} \in \hat{L}} p(l, \hat{l}) \log\left(\frac{p(l, \hat{l})}{p(l)p(\hat{l})}\right)}{\max(H(L), H(\hat{L}))},$$

where $p(l)$ and $p(\hat{l})$ represent the marginal probability distribution functions of L and \hat{L} , respectively. $p(l, \hat{l})$ is the joint probability function of L and \hat{L} . $H(\cdot)$ is the entropy function. The greater NMI means the better clustering performance.

Results

We report the extensive experimental results in Table 2. Except SSC, LRR, CAN, and SSR, we run other methods on each kernel matrix individually. As a result, we show both the best performance among those 12 kernels and the average results over those 12 kernels for them. Based on this table, we can see that our proposed SLKE achieves the best performance in most cases. To be specific, we have the following observations:

1) Compared to classical k-means based RKKM and spectral clustering techniques, our proposed method SLKE has a big advantage in terms of accuracy and NMI. With respect to the recently proposed SSR and TLSC methods, SKLE always obtains better results.

2) SLKE-R and SLKE-S often outperform LRR and SSC, respectively. The accuracy increased by 8.92%, 8.76% on average, respectively. That is to say, kernel-based distance approach indeed performs better than original data reconstruction technique. This verifies the importance of retaining relation information when we learn a low-dimensional representation, especially for sparse representation.

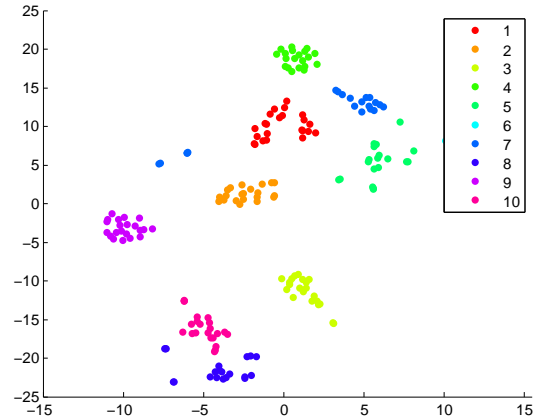
3) With respect to adaptive neighbors approach CAN, we also obtain better performance on those datasets except COIL20. For COIL20, our results are quite close to

²<https://github.com/sckangz/SLKE>

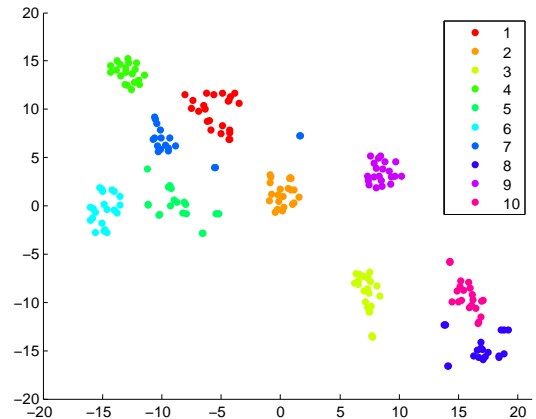
CAN's. Therefore, compared to various similarity learning techniques, our method is very competitive.

4) Regarding low-rank and sparse representation, it is hard to conclude which one is better. It totally depends on the specific data.

Furthermore, we run t-SNE (Maaten and Hinton 2008) algorithm on the JAFFE data X and the reconstructed data XZ from the best result of our SLKE-R. As shown by Figure 1, we can see that our method can well preserve the cluster structure of the data.



(a) Original data



(b) Reconstructed data

Figure 1: JAFFE data set visualized in two dimensions.

To see the significance of improvements, we further apply the Wilcoxon signed rank test (Peng, Cheng, and Cheng 2017) to Table 2. We show the p -values in Table 3. We note that the testing results are under 0.05 in most cases when comparing SLKE-S and SLKE-R to other methods. Therefore, SLKE-S and SLKE-R outperform SC, RKKM, SSC, and SSR with statistical significance.

(a) Accuracy(%)

Data	SC	RKKM	SSC	LRR	SSR	CAN	TLSC	SLKE-S	SLKE-R
YALE	49.42(40.52)	48.09(39.71)	38.18	61.21	54.55	58.79	55.85(45.35)	61.82(38.89)	66.24 (51.28)
JAFFE	74.88(54.03)	75.61(67.89)	99.53	99.53	87.32	98.12	99.83(86.64)	96.71(70.77)	99.85 (90.89)
ORL	58.96(46.65)	54.96(46.88)	36.25	76.50	69.00	61.50	62.35(50.50)	77.00 (45.33)	74.75(59.00)
COIL20	67.60(43.65)	61.64(51.89)	73.54	68.40	76.32	84.58	72.71(38.03)	75.42(56.83)	84.03(65.65)
BA	31.07(26.25)	42.17(34.35)	24.22	45.37	23.97	36.82	47.72(39.50)	50.74 (36.35)	44.37(35.79)
TR11	50.98(43.32)	53.03(45.04)	32.61	73.67	41.06	38.89	71.26(54.79)	69.32(46.87)	74.64 (55.07)
TR41	63.52(44.80)	56.76(46.80)	28.02	70.62	63.78	62.87	65.60(43.18)	71.19(47.91)	74.37 (53.51)
TR45	57.39(45.96)	58.13(45.69)	24.35	78.84	71.45	48.41	74.02(53.38)	78.55(50.59)	79.89 (58.37)
TDT2	52.63(45.26)	48.35(36.67)	23.45	52.03	20.86	19.74	55.74(44.82)	59.61(25.40)	74.92 (33.67)

(b) NMI(%)

Data	SC	RKKM	SSC	LRR	SSR	CAN	TLSC	SLKE-S	SLKE-R
YALE	52.92(44.79)	52.29(42.87)	45.56	62.98	57.26	57.67	56.50(45.07)	59.47(40.38)	64.29 (52.87)
JAFFE	82.08(59.35)	83.47(74.01)	99.17	99.16	92.93	97.31	99.35(84.67)	94.80(60.83)	99.49 (81.56)
ORL	75.16(66.74)	74.23(63.91)	60.24	85.69	84.23	76.59	78.96(63.55)	86.35 (58.84)	85.15(75.34)
COIL20	80.98(54.34)	74.63(63.70)	80.69	77.87	86.89	91.55	82.20(73.26)	80.61(65.40)	91.25(73.53)
BA	50.76(40.09)	57.82(46.91)	37.41	57.97	30.29	49.32	63.04(52.17)	63.58 (55.06)	56.78(50.11)
TR11	43.11(31.39)	49.69(33.48)	02.14	65.61	27.60	19.17	58.60(37.58)	67.63(30.56)	70.93 (45.39)
TR41	61.33(36.60)	60.77(40.86)	01.16	67.50	59.56	51.13	65.50(43.18)	70.89 (34.82)	68.50(47.45)
TR45	48.03(33.22)	57.86(38.96)	01.61	77.01	67.82	49.31	74.24(44.36)	72.50(38.04)	78.12 (50.37)
TDT2	52.23(27.16)	54.46(42.19)	13.09	64.36	02.44	03.97	58.35(46.37)	58.55(15.43)	68.21 (28.94)

Table 2: Clustering results obtained on benchmark data sets. The average performance of those 12 kernels is put in parenthesis. The best results among those kernels are highlighted in boldface.

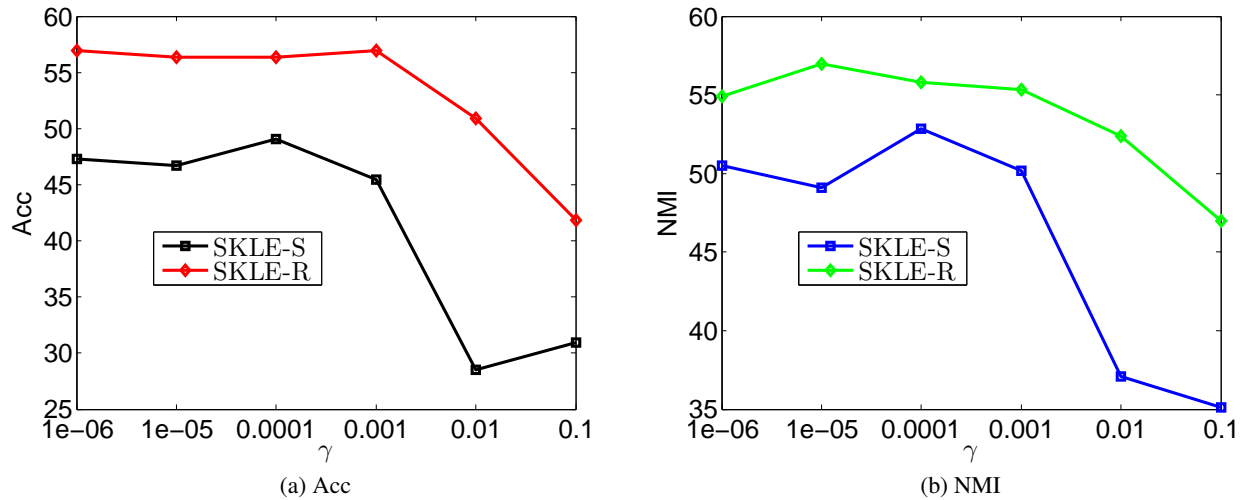


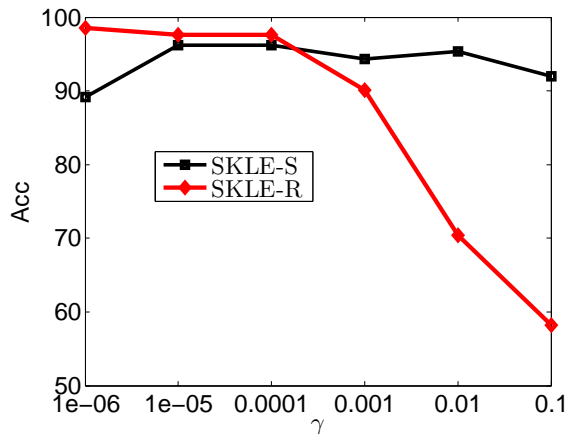
Figure 2: The effect of parameter γ on the YALE data set.

Table 3: Wilcoxon Signed Rank Test on all Data sets.

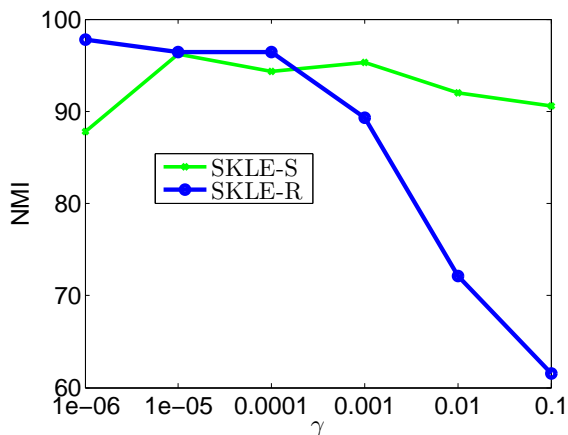
Method	Metric	SC	RKKM	SSC	LRR	SSR	CAN	TLSC
SLKE-S	Acc	.0039	.0039	.0117	.2500	.0078	.0391	.0391
	NMI	.0078	.0039	.0195	.6523	.0391	.0547	.3008
SLKE-R	Acc	.0039	.0039	.0039	.0977	.0039	.0039	.0117
	NMI	.0039	.0078	.0039	.0742	.0039	.0078	.0391

Parameter Analysis

In this subsection, we investigate the influence of our model parameter γ on the clustering results. Take Gaussian kernel with $t = 100$ of YALE and JAFFE data sets as examples, we plot our algorithm's performance with γ in the range $[10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}]$ in Figure 2 and 3, respectively. As we can see that our proposed methods work well for a wide range of γ , e.g., from 10^{-6} to 10^{-3} .



(a) Acc



(b) NMI

Figure 3: The effect of parameter γ on the JAFFE data set.

Conclusion

In this paper, we present a novel similarity learning framework relying on an embedding of kernel-based distance. Our model is flexible to obtain either low-rank or sparse representation of data. Comprehensive experimental results on real data sets well demonstrate the superiority of the proposed method on the clustering task. It has great potential to be applied in a number of applications beyond clustering. It has been shown that the performance of the proposed method is largely determined by the choice of kernel function. In the future, we plan to address this issue by develop-

ing a multiple kernel learning method, which is capable of automatically learning an appropriate kernel from a pool of input kernels.

Acknowledgment

This paper was in part supported by Grants from the Natural Science Foundation of China (Nos. 61806045 and 61572111), a 985 Project of UESTC (No. A1098531023601041) and two Fundamental Research Fund for the Central Universities of China (Nos. A03017023701012 and ZYGX2017KYQD177).

References

- Chen, X.; Ye, Y.; Xu, X.; and Huang, J. Z. 2012. A feature group weighting method for subspace clustering of high-dimensional data. *Pattern Recognition* 45(1):434–446.
- Chen, X.; Hong, W.; Nie, F.; He, D.; Yang, M.; and Huang, J. Z. 2018. Directly minimizing normalized cut for large scale data. In *SIGKDD*, 1206–1215.
- Du, L., and Shen, Y.-D. 2015. Unsupervised feature selection with adaptive structure learning. In *SIGKDD*, 209–218. ACM.
- Du, L.; Zhou, P.; Shi, L.; Wang, H.; Fan, M.; Wang, W.; and Shen, Y.-D. 2015. Robust multiple kernel k-means using ℓ_2 ; 1-norm. In *IJCAI*, 3476–3482. AAAI Press.
- Elhamifar, E., and Vidal, R. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence* 35(11):2765–2781.
- Gao, X.; Hoi, S. C.; Zhang, Y.; Zhou, J.; Wan, J.; Chen, Z.; Li, J.; and Zhu, J. 2017. Sparse online learning of image similarity. *ACM Transactions on Intelligent Systems and Technology (TIST)* 8(5):64.
- Hirzer, M.; Roth, P. M.; Köstinger, M.; and Bischof, H. 2012. Relaxed pairwise learned metric for person re-identification. In *ECCV*, 780–793. Springer.
- Hoi, S. C.; Liu, W.; and Chang, S.-F. 2008. Semi-supervised distance metric learning for collaborative image retrieval. In *CVPR*, 1–7. IEEE.
- Huang, S.; Wang, H.; Li, T.; Li, T.; and Xu, Z. 2018. Robust graph regularized nonnegative matrix factorization for clustering. *Data Mining and Knowledge Discovery* 32(2):483–503.
- Huang, J.; Nie, F.; and Huang, H. 2015. A new simplex sparse learning model to measure data similarity for clustering. In *IJCAI*, 3569–3575.
- Kang, Z.; Lu, X.; Yi, J.; and Xu, Z. 2018a. Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification. In *IJCAI*, 2312–2318.
- Kang, Z.; Peng, C.; Cheng, Q.; and Xu, Z. 2018b. Unified spectral clustering with optimal graph. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*. AAAI Press.
- Kang, Z.; Wen, L.; Chen, W.; and Xu, Z. 2018c. Low-rank kernel learning for graph-based clustering. *Knowledge-Based Systems*.

- Kang, Z.; Peng, C.; and Cheng, Q. 2017a. Kernel-driven similarity learning. *Neurocomputing* 267:210–219.
- Kang, Z.; Peng, C.; and Cheng, Q. 2017b. Twin learning for similarity and clustering: A unified kernel approach. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*. AAAI Press.
- Larsen, R. M. 2004. Propack-software for large and sparse svd calculations. Available online. URL <http://sun.stanford.edu/rmunk/PROPACK> 2008–2009.
- Li, T.; Cheng, B.; Ni, B.; Liu, G.; and Yan, S. 2016. Multitask low-rank affinity graph for image segmentation and image annotation. *ACM Transactions on Intelligent Systems and Technology (TIST)* 7(4):65.
- Liu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; and Ma, Y. 2013. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(1):171–184.
- Maaten, L. v. d., and Hinton, G. 2008. Visualizing data using t-sne. *Journal of machine learning research* 9(Nov):2579–2605.
- Ng, A. Y.; Jordan, M. I.; Weiss, Y.; et al. 2002. On spectral clustering: Analysis and an algorithm. *NIPS* 2:849–856.
- Nie, F.; Wang, X.; and Huang, H. 2014. Clustering and projected clustering with adaptive neighbors. In *SIGKDD*, 977–986. ACM.
- Niyogi, X. 2004. Locality preserving projections. In *NIPS*, volume 16, 153. MIT.
- Passalis, N., and Tefas, A. 2017. Dimensionality reduction using similarity-induced embeddings. *IEEE transactions on neural networks and learning systems*.
- Peng, X.; Xiao, S.; Feng, J.; Yau, W.-Y.; and Yi, Z. 2016. Deep subspace clustering with sparsity prior. In *IJCAI*, 1925–1931.
- Peng, C.; Kang, Z.; Cai, S.; and Cheng, Q. 2018. Integrate and conquer: Double-sided two-dimensional k-means via integrating of projection and manifold construction. *ACM Transactions on Intelligent Systems and Technology (TIST)* 9(5):57.
- Peng, C.; Cheng, J.; and Cheng, Q. 2017. A supervised learning model for high-dimensional and large-scale data. *ACM Transactions on Intelligent Systems and Technology (TIST)* 8(2):30.
- Roweis, S. T., and Saul, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500):2323–2326.
- Tao, Z.; Liu, H.; Li, S.; Ding, Z.; and Fu, Y. 2017. From ensemble clustering to multi-view clustering. In *IJCAI*, 2843–2849.
- Tenenbaum, J. B.; De Silva, V.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500):2319–2323.
- Towne, W. B.; Rosé, C. P.; and Herbsleb, J. D. 2016. Measuring similarity similarly: Lda and human perception. *ACM Transactions on Intelligent Systems and Technology* 8(1).
- Weinberger, K. Q.; Blitzer, J.; and Saul, L. K. 2005. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, 1473–1480.
- Wright, J.; Yang, A. Y.; Ganesh, A.; Sastry, S. S.; and Ma, Y. 2009. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence* 31(2):210–227.
- Xu, Z.; Jin, R.; King, I.; and Lyu, M. 2009. An extended level method for efficient multiple kernel learning. In *NIPS*, 1825–1832.
- Xu, Z.; Jin, R.; Yang, H.; King, I.; and Lyu, M. R. 2010. Simple and efficient multiple kernel learning by group lasso. In *ICML-10*, 1175–1182. Citeseer.
- Zhang, L.; Zhang, Q.; Du, B.; You, J.; and Tao, D. 2017. Adaptive manifold regularized matrix factorization for data clustering. In *IJCAI*, 33999–3405.
- Zhang, L.; Yang, M.; and Feng, X. 2011. Sparse representation or collaborative representation: Which helps face recognition? In *ICCV*, 471–478. IEEE.
- Zhao, Z.; Wang, L.; Liu, H.; and Ye, J. 2013. On similarity preserving feature selection. *IEEE Transactions on Knowledge and Data Engineering* 25(3):619–632.
- Zhuang, L.; Zhou, Z.; Gao, S.; Yin, J.; Lin, Z.; and Ma, Y. 2017. Label information guided graph construction for semi-supervised learning. *IEEE Transactions on Image Processing* 26(9):4182–4192.