

An Improved Spectral Clustering Algorithm Based on Local Neighbors in Kernel Space¹

Xinyue Liu^{1,2}, Xing Yong² and Hongfei Lin¹

¹ School of Computer Science and Technology, Dalian University of Technology,
116024 Dalian, China

² School of Software, Dalian University of Technology,
116620 Dalian, China
xyliu@dlut.edu.cn

Abstract. Similarity matrix is critical to the performance of spectral clustering. Mercer kernels have become popular largely due to its successes in applying kernel methods such as kernel PCA. A novel spectral clustering method is proposed based on local neighborhood in kernel space (SC-LNK), which assumes that each data point can be linearly reconstructed from its neighbors. The SC-LNK algorithm tries to project the data to a feature space by the Mercer kernel, and then learn a sparse matrix using linear reconstruction as the similarity graph for spectral clustering. Experiments have been performed on synthetic and real world data sets and have shown that spectral clustering based on linear reconstruction in kernel space outperforms the conventional spectral clustering and the other two algorithms, especially in real world data sets.

Keywords: Spectral Clustering, Kernel Space, Local Neighbors, Linear Reconstruction.

1. Introduction

In recent years, spectral clustering has become one of the most popular clustering algorithms due to its high performance in data clustering and simplicity in implementation compared to the traditional clustering methods. It is based on the spectral analysis methods and solves the data clustering by graph partitioning problems [1], [2]. Spectral clustering algorithms consist of two steps: (1) construct a similarity graph with some kind of similarity function; (2) find an optimal partition of the graph and cluster the data points. The former which reflects the intrinsic structure of the data plays an important

¹ This paper is supported by “the Fundamental Research Funds for the Central Universities” and “the National Natural Science Foundation of China under Grant No. 61070016”.

role in spectral clustering [3], [4], thus a great deal of effort has been carried out to address it [5], [6].

Spectral clustering makes use of information achieved from an appropriately defined affinity matrix. Its primary strength is the ability to treat complex patterns where traditional methods (such as k -means) either cannot be applied, or fail. The similarity matrix should be built in such a way that reflects the topological characteristics of the data sets. The most commonly used similarity measure is Gaussian function which is defined as

$G(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$, where x_i and x_j represent two point respectively [7].

Although Gaussian function is simple to implement, the selection of the parameter σ is still an open issue. Generally it is non-trivial to find a good σ and spectral clustering is sensitive to the value of σ , especially in a multi-scale data sets. In practice, σ is often set by an empirical value, such as it is set as 0.05 of the maximal pair wise Euclidean distance among the data points in normalized cut algorithm [2]. What's more, sparsity is another desired property, since it can offers computational advantages [7].

In this paper, we focus on how to construct an optimal similarity graph. We propose a novel method to obtain the similarity matrix based on linear reconstruction with local neighbors in kernel space. The idea of linear reconstruction derives from the manifold learning which assumes data points are locally linear and each point can be reconstructed by a linear combination of its neighbors [8]. It has been applied to semi-supervised learning [9] and spectral clustering [10]. The kernel method has become popular largely due to many successes in applying kernel methods such as kernel PCA [11] and spectral regularization [6]. In our algorithm, we introduce the kernel methods for constructing the similarity matrix which increase the linear separability by mapping the data into high dimensional space. Experimental results on both artificial data sets and real world data sets indicate that spectral clustering based on local neighborhood in kernel space (SC-LNK) outperforms the traditional spectral clustering (SC), self-tuning spectral clustering (SSC) and locality spectral clustering (LSC).

The rest of this paper is organized as follows: some related work is discussed in Section 2. We begin with a brief overview of NJW method [12] in Section 3. Then introduce the method of linear reconstruction in Section 4 and mercer kernel in Section 5 on details of constructing the similarity matrix for our algorithm. In Section 6, we summarize our algorithm and experimental results are then presented in section 7. Finally we conclude the discussions and point out further works in the last section.

2. Related Work

An appropriate adjacency matrix is conducive to make a good partition for clustering algorithms. Much research has been conducted on the problem of

how to construct an optimal similarity matrix for spectral clustering [6], [13], [14]. Most existing works on constructing similarity matrix were built on Gaussian kernel function which is limited to its sensitive scale parameter and the real world data in some situation. Little attention has been paid to the other methods to measure the similarity in spectral clustering.

As mentioned above, the scale parameter of Gaussian function is very sensitive, especially in data sets with multiple scales. In order to deal with the problem, Manor et al. [13] proposed a new algorithm called self-tuning to use a local scale parameter for each data point, i.e., they use the distance of k -th nearest neighbor of each point as the scale parameter. The parameter k is set to 7 in their experiments. However, as have shown in our experiments, it fails on many real data sets although performs well on some synthetic data sets.

Gong et al. proposed a spectral clustering method [10] which called LSC algorithm. Instead of using the pairwise relationship, they considered the linear neighbors relationship based on the idea of local linear embedding in manifold learning [8]. With this novel relationship in hand, LSC achieved the similarity matrix which represents the local neighbors information well. In particular, the matrix obtained is sparse which offers computational advantages. However, the LSC method has some limitations in synthetic and real world data sets as shown in our experiments.

In [15], the linear Reconstruction in kernel space had been described. DeCoste et al analysis the local linear embedding technique in kernel space and applied to specific application. Through the introduction of kernel methods, it can increase the linear separability by mapping the data into high dimensional space. Therefore, the kernel trick method conduces to deal with the multiple and real data sets. But the authors only focus on the classification problem.

3. Review of Spectral Clustering

Give a set of data points $X = \{x_1, x_2, \dots, x_n\}$ with each point $x_i \in R^d$, then we can construct an undirected graph $G = (V, E)$ in which every vertex v_i represents the data point x_i . According to a similarity function, we obtain the similarity graph and let W be its weighted adjacency matrix. Note that $w_{ij} \geq 0$ and W is symmetric. Each edge $(i, j) \in E$ carries a weight w_{ij} which represents the similarity between the point x_i and x_j . We expect to find a good partition of the graph such that points with low similarities should be clustered into different groups. Many spectral clustering algorithms formalize this partitioning problem in different ways [2], [12], [14], [16-17]. In this paper, we adopt the normalized cuts (Ncut) [12] described as follows:

4. Analysis of Linear Reconstruction

The similarity graph plays an important role in a spectral clustering algorithm. Before, we need to define a similarity function on the data. In the common case, a reasonable candidate is the Gaussian similarity function, which is

defined as $G(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$. However, it indicates that there is no reliable way to choose the parameter σ when there are very few or even there are no labeled examples according to [18].

Instead of using pairwise relationship to construct the similarity graph in conventional spectral clustering, we propose to use the neighbor information of each point to construct the graph.

Algorithm 1: Normalized Spectral Clustering

Input: Data set X , number k of clusters

Output: A partitioning S_1, S_2, \dots, S_k

begin

Construct similarity matrix $W \in R^{n \times n}$

Define a diagonal matrix $D(i, j) = \sum_{j=1}^n W(i, j)$

Form the normalized Laplacian matrix $L = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$

Compute the k largest eigenvectors of L and

construct the matrix $U \in R^{n \times k}$ with the eigenvectors as its columns

Form matrix $Y \in R^{n \times k}$ from U by normalizing the rows to norm 1.

Group Y into clusters S_1, S_2, \dots, S_k with k -means

end.

For convenience of calculation, we assume the sample space is locally linear, that is, each point can be optimally reconstructed by a linear combination of its neighbors [8]. Then our objective is to minimize

$$\mathcal{E} = \|x_i - \sum_{j: x_j \in N(x_i)} w_{ij} x_j\|^2 \quad (1)$$

where $N(x_i)$ represents the neighbors of x_i , and w_{ij} is the contribution of x_j to x_i . According to [8], w_{ij} should satisfy the constraint

$\sum_{j: x_j \in N(x_i)} w_{ij} = 1, w_{ij} \geq 0$. Intuitively, the more similar x_j to x_i , the larger w_{ij} would be. Thus we can use w_{ij} to measure the similarity between x_j and x_i . Then we can rewrite Eq. (1) as

$$\begin{aligned}
 \mathcal{E}_i &= \left\| x_i - \sum_{j: x_j \in N(x_i)} w_{ij} x_j \right\|^2 \\
 &= \left\| \sum_{j: x_j \in N(x_i)} w_{ij} (x_i - x_j) \right\|^2 \\
 &= \sum_{j, k: x_j, x_k \in N(x_i)} w_{ij} w_{ik} (x_i - x_j)^T (x_i - x_k) \\
 &= \sum_{j, k: x_j, x_k \in N(x_i)} w_{ij} C_{jk}^i w_{ik}
 \end{aligned} \tag{2}$$

where C_{jk}^i represents the (j, k) -th entry of the local Gram matrix at point x_i and $C_{jk}^i = (x_i - x_j)^T (x_i - x_k)$. Then we can obtain the reconstruction weights of each data point by solving the following n standard quadratic programming problems

$$\begin{aligned}
 \min_{w_{ij}} & \sum_{j, k: x_j, x_k \in N(x_i)} w_{ij} C_{jk}^i w_{ik} \\
 s.t. & \sum_{j: x_j \in N(x_i)} w_{ij} = 1, w_{ij} \geq 0
 \end{aligned} \tag{3}$$

Using the above method, we will construct a sparse matrix W . It is worth mentioning that we set the weights $W_{ij} = w_{ij}$ and $W_{ji} = w_{ij}$. If there are any overlap, the later weight will overlap the former one so that we can guarantee the symmetric of the reconstruction matrix W . Then the W can be treated as the similarity matrix.

5. Kernelized Linear Reconstruction

5.1. Mercer Kernel

Combining the kernel method can optimize the performance of clustering algorithm [19], [20]. Through the introduction of kernel methods, it can increase the linear separability by mapping the data into high dimensional space. DeCoste has analyzed the linear Reconstruction in kernel space [15]. Consider a $l \times D$ matrix X of data points. A mercer kernel $K(x_i, x_j)$ will projects two given points from D -dimensional original space into some (possibly infinite-dimensional) feature space and return their dot product in that feature space. That is

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) = \phi(x_i)' \phi(x_j) \tag{4}$$

where ϕ is some mapping function which need not explicitly to compute the coordinates of the projected vectors. Thus the kernel method can avoiding curse of dimensionality when explored large non-linear feature spaces.

One of the most popular mercer kernels is the radial basis function (RBF) kernel:

$$K(u, v) = e^{-\frac{\|u-v\|^2}{2\sigma^2}} \quad (5)$$

where the parameter σ controls a scale of the dot product of the two data points. We will use this kernel when mapping the input space to its feature space.

5.2. Linear Reconstruction in Kernel Space

In general, the selection of k nearest neighbors use the Euclidian distance in original space. In this paper, we use the distance based on Mercer kernels. The distance between data point x_i and x_j in kernel space is defined as:

$$d_{ij} = \text{dist}(\phi(x_i), \phi(x_j)) = \sqrt{\|\phi(x_i) - \phi(x_j)\|^2} \quad (6)$$

Through Mercer kernel, distances can be calculated directly from kernel value as follows

$$d_{ij} = \sqrt{K_{ii} - 2K_{ij} + K_{jj}} \quad (7)$$

For each $\phi(x_i)$ in kernel space, denote its k nearest neighbors in kernel space as $N(\phi(x_i))$, then the kernelized reconstruction error is:

$$\mathcal{E} = \|\phi(x_i) - \sum_{j: \phi(x_j) \in N(\phi(x_i))} w_{ij} \phi(x_j)\|^2 \quad (8)$$

To find the optimal weights for reconstructing, we must compute the covariance matrix C_{jk} in kernel space which is:

$$C_{jk}^i = (\phi(x_i) - \phi(x_j))^T (\phi(x_i) - \phi(x_k)) \quad (9)$$

Then replacing each resulting term of form $\phi(x_a) \cdot \phi(x_b)$ in Eq. (4) with the corresponding kernel value $K_{ab} = K(x_a, x_b)$. The kernelized covariance matrix can be obtained by

$$\forall x_j, x_k \in N(x_i) \quad C_{jk}^i = K_{ii} - K_{ij} - K_{ik} + K_{jk} \quad (10)$$

We can compute the weights matrix W of kernel space and use it as the similarity matrix for spectral clustering.

6. Proposed Algorithm

In this section, we will introduce our algorithm called spectral clustering algorithm based on local neighbors in kernel space (SC-LNK). The main procedure of SC-LNK is summarized in Algorithm 2.

```
Algorithm 2: The SC-LNK Algorithm
Input: Data set  $X$ , number  $k$  of clusters
Output: A partitioning  $S_1, S_2, \dots, S_k$ 
begin
    Construct similarity matrix  $W \in R^{n \times n}$  according to
    Eq. (3) and Eq. (9)
    Use the affinity matrix to execute the algorithm 1
end.
```

The main advantage of our algorithm is that SC-LNK can effectively and correctly discover the underlying cluster structure by taking the advantage of the smooth graph W in kernel space. When using the Mercer kernel, the selection of parameter σ adopts the concept of local scale in [13] and σ_i equals the distance of the 15th neighbor of point x_i . Furthermore the parameter k in SC-LNK algorithm is more stable and more accuracy than former methods.

7. Experiments

In this section, we will report our results on several synthetic data sets and the real data sets.

7.1. Experiments on Synthetic Data Sets

We applied the SC-LNK algorithm and LSC method to five synthetic data sets that has been mentioned in [10], [13]. The results are show in Fig. 1 and Fig. 2.

As we can see, both SC-LNK and LSC can recognize the non-convex cluster and reliably finds groups consistent with what a human's intuitive solution in Fig. 1. However, the LSC method fails to find the real structure

hidden in data sets which are multiscale in Fig. 2 while the SC-LNK algorithm can distinguish it after mapping to kernel space.

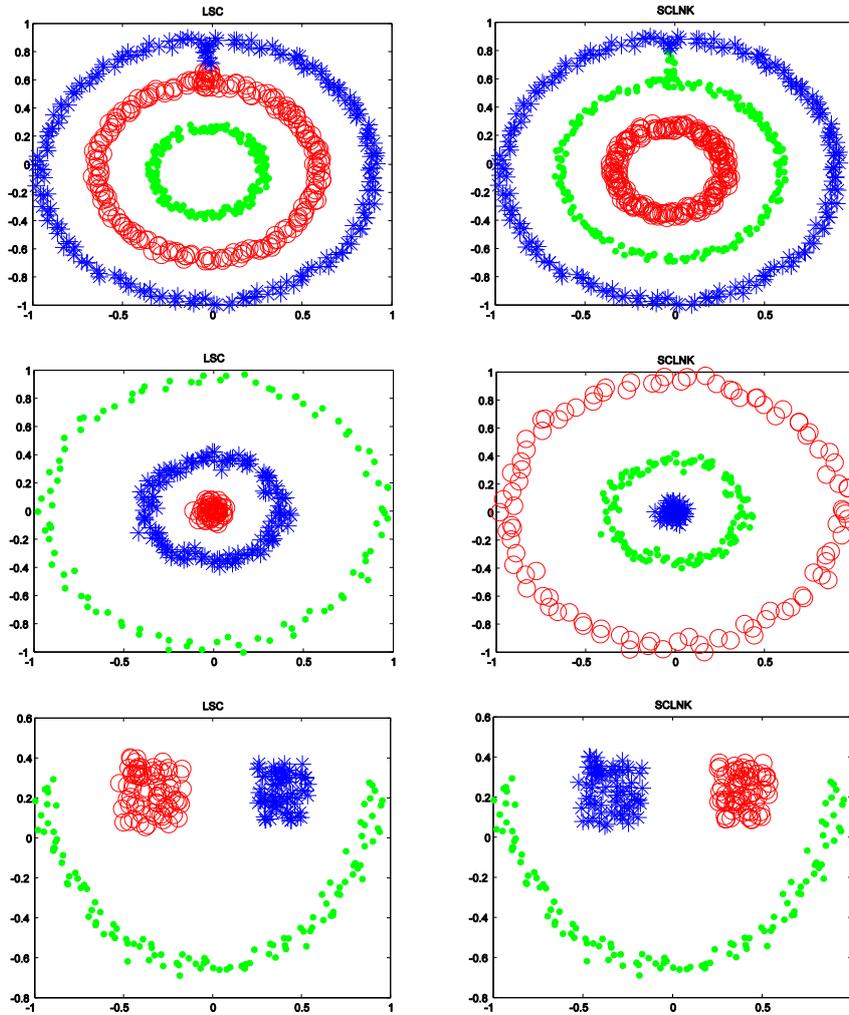


Fig. 1. Contrast results on Synthetic data. The left column shows the LSC algorithm results; The right column presents the SC-LNK clustering results. As shown above, both algorithms can recognize the non-convex clusters.

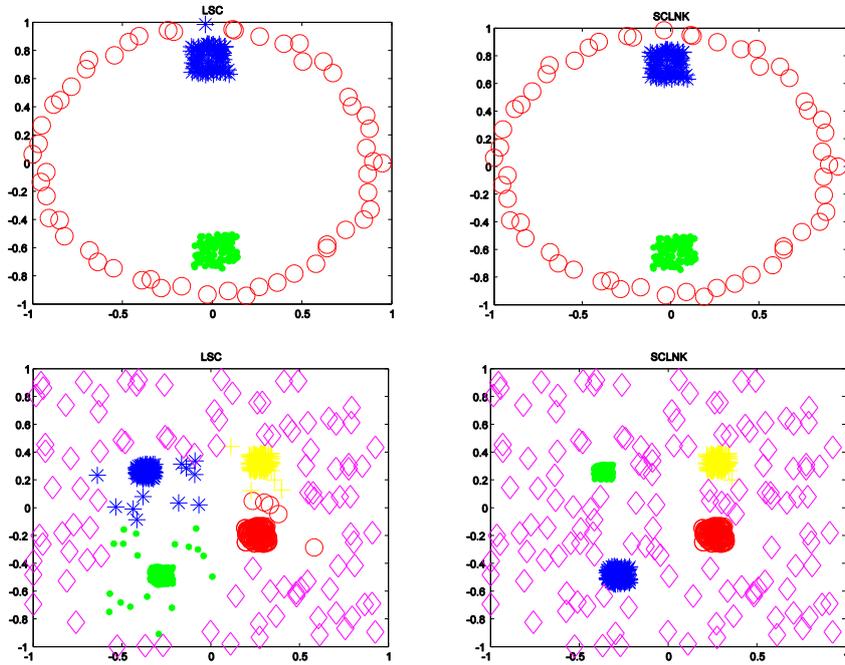


Fig. 2. Contrast results on Synthetic data. The left column shows the LSC algorithm results; The right column presents the SC-LNK clustering results. As shown above, LSC fails to find the real structure hidden in data sets while SC-LNK can distinguish.

7.2. Experiments on Real Data Sets

In order to extensively examine the effectiveness of the SC-LNK algorithm, we further compare the performances of our method with the other three clustering algorithms which include the SC algorithm, SSC method and the LSC algorithm. To evaluate the performance of different clustering algorithms, two different metrics are used: one is Rand index, and the other is the Normalized Mutual Information.

Evaluation metrics. Rand index (RI) is widely used to evaluate the clustering performance. A decision is considered correct if the clustering algorithm agrees with the real clustering. RI [21] is defined as

$$RI = \frac{\#CD}{n(n-1)/2} \quad (11)$$

where CD denotes the number of correct decisions. A larger RI value ($RI \in [0,1]$) signifies a better clustering result.

Normalized Mutual Information (NMI) is another measure for determining the quality of clusters. For two random variable X and Y , the NMI is defined as [22]:

$$NMI(X, Y) = \frac{I(X, Y)}{\sqrt{H(X)H(Y)}} \quad (12)$$

where $I(X, Y)$ is the mutual information between X and Y , while $H(X)$ and $H(Y)$ are the entropies of X and Y respectively. Note that $NMI \in [0,1]$ and give a clustering result, the NMI is estimated as

$$NMI = \frac{\sum_{l=1}^k \sum_{h=1}^c n_{l,h} \log\left(\frac{n_{l,h}}{\hat{n}_h}\right)}{\sqrt{\left(\sum_{l=1}^k n_l \log \frac{n_l}{n}\right) \left(\sum_{h=1}^c \hat{n}_h \log \frac{\hat{n}_h}{n}\right)}} \quad (13)$$

where n_l denotes the number of data contained in the cluster $C_l (1 \leq l \leq k)$, \hat{n}_h is the number of data belonging to the h -th class and $n_{l,h}$ denotes the number of data which are in the intersection between the cluster C_l and the h -th class. The larger the NMI, the better the performance.

Results on uci data sets. We carry out the experiments on five data sets which come from the UCI data repository [23]. The properties of the datasets are summarized in Table 1.

Table 1. Properties of UCI Datasets

Dataset	Iris	Wine	Ionosphere	Glass	Segmentation
No. of instances	150	178	351	214	210
No. of attributes	4	13	34	9	19
No. of clusters	3	3	2	6	7

Experimental results are presented in Fig. 3 and Fig. 4. Especially, following [24], we set the parameter σ in SC as 0.05 of the maximal pairwise Euclidean distance among the dataset. We adopt an empirical value to estimate the parameter k and report the best result for the LSC and SC-LNK algorithm.

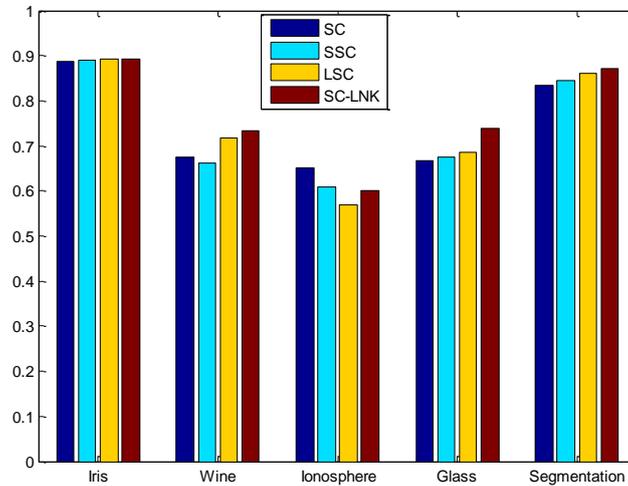


Fig. 3. Results on UCI data by RI

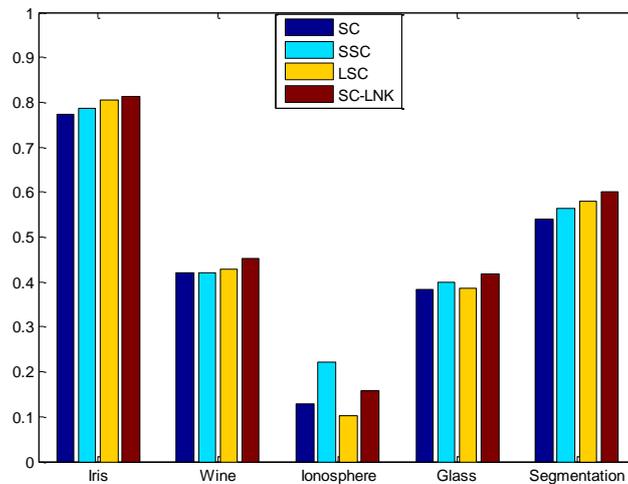


Fig. 4. Results on UCI data by NMI

From Fig. 3 and Fig. 4, we can see that the SSC algorithm has achieved limited success on real world data sets. SC-LNK outperforms the other algorithms on most data sets, especially in the wine, glass and segmentation dataset, the improvement is obvious. But on the ionosphere dataset, the performance is not as good as SC and self-tuning. Despite this, it can still be observed that SC-LNK works well on UCI datasets. With linear reconstruction

in kernel space, the local relationship is maintained and different clusters are more linearly separable, thus SC-LNK can distinguish the intrinsic structure of the data more correctly.

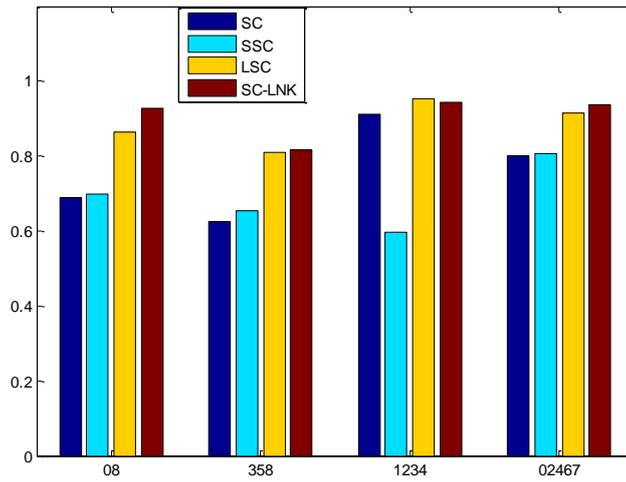


Fig. 5. Results on USPS data by RI

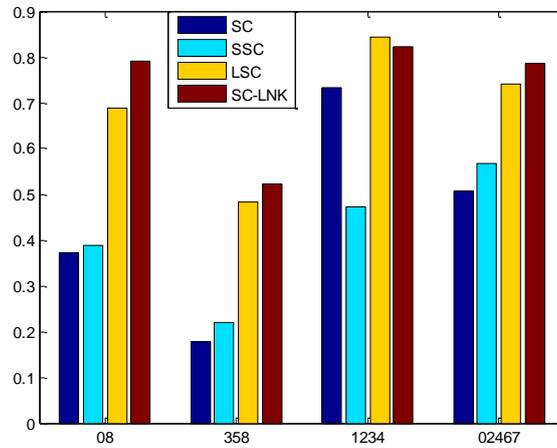


Fig. 6. Results on USPS data by NMI

Results on usps data sets. In this experiment, we consider the hand written digits from the well-known USPS database.

The digits have been normalized and centered to 16×16 gray-level images, thus the dimensionality of digit space is 256, as each sample image will be transformed to a vector as one column of the similarity graph. In the database it contains 7291 training instances and 2007 test instances.

We choose digits $\{0,8\}$, $\{3,5,8\}$, $\{1,2,3,4\}$ and $\{0,2,4,6,7\}$ as subsets and carry out the experiments separately. The clustering results by RI and NMI are presented in Fig. 5 and Fig. 6 separately.

According to the results in Fig. 5 and Fig. 6, we can see that the idea of linear reconstruction has a distinct advantage in the USPS data sets. The performance of LSC and SC-LNK algorithms are much better than the other two methods. Even on the challenging USPS subsets $\{1,2,3,4\}$, both of them still have improvement compared to SC and SSC though LSC is better than SC-LNK. But in most cases, the effectiveness of SC-LNK algorithm is better than LSC and the other two methods. It proves that the new similarity measure using linear reconstruction in kernel space is very helpful in detecting the real manifold of the digits.

8. Conclusion

In this paper, we propose a novel method of constructing similarity matrix using linear reconstruction. Based on this method we introduce the concept of kernel methods and propose an efficient spectral clustering algorithm called SN-LNK. Experimental results on five synthetic data sets and two groups of the real data sets show that the proposed algorithm achieves considerable improvements over traditional spectral clustering, locality spectral clustering and self-tuning spectral clustering algorithms.

There is still much work for us to undergo further research. The similarity matrix we obtained in this paper can be extended to other clustering algorithms based on the affinity matrix. Furthermore, the selection on the number of nearest neighborhoods k still remains to be further studied. We will pursue these research directions in our future work.

References

1. Hagen, L., Kahng, B.: New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on computer aided design*, vol. 11, no. 9. (1992)
2. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, 888–905. (2000)
3. Jordan, F., Bach, F.: Learning spectral clustering. In *Advances in neural information processing systems 16: proceedings of the 2003 conference*, vol. 16. The MIT Press, 305. (2004)
4. Ozerem, U., Erdogmus, D., Jenssen, R.: Mean shift spectral clustering. *Pattern*

- Recognition, vol. 41, no. 6, 1924–1938. (2008)
5. Wang, X., Davidson, I.: Active spectral clustering. In *2010 IEEE International Conference on Data Mining*. 561–568. (2010)
 6. Li, Z., Liu, J., Tang, X.: Constrained clustering via spectral regularization. In *Computer Vision and Pattern Recognition*. 421–428. (2009)
 7. Luxbug, U.: A tutorial on spectral clustering. *Statistics and Computing*, vol. 17, no. 4, 395–416. (2007)
 8. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. *Science*, vol. 290, no. 5500, p. 2323. (2000)
 9. Wang, F., Zhang, C.: Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering*. 55–67. (2007)
 10. Gong, Y., Chen, C.: Locality Spectral Clustering. *AI 2008: Advances in Artificial Intelligence*. 348–354. (2008)
 11. Scholkopf, B., Smola, A., Muller, K.: Kernel principal component analysis. *Artificial Neural Networks-ICANN'97*. 583–588. (1997)
 12. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14: Proceeding of the 2001 Conference*. 849–856. (2001)
 13. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. *Advances in neural information processing systems*, vol. 17, no. 1601–1608, p. 16. (2004)
 14. Zhang, X., Li, J., Yu, H.: Local density adaptive similarity measurement for spectral clustering. *Pattern Recognition Letters*. (2010)
 15. DeCoste, D.: Visualizing Mercer kernel feature spaces via kernelized locally linear embedding. In *Proceedings of the Eighth International Conference on Neural Information Processing (ICONIP-01)*. (2001)
 16. Jianbo, S., Yu, S., Shi, J.: Multiclass Spectral Clustering. In *International Conference on Computer Vision*. Citeseer, 313–319. (2003)
 17. Meila, M., Shi, J.: A random walks view of spectral segmentation. *AI and Statistics (AISTATS)*, vol. 2001. (2001)
 18. Zhou, D., Bousquet, O., Lal, T., Weston, J., Schölkopf, B.: Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*. 595–602. (2004)
 19. Ben-Hur, A., Horn, D., Siegelmann, H., Vapnik, V.: Support vector clustering. *The Journal of Machine Learning Research*, vol. 2, p. 137. (2002)
 20. Girolami, M.: Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, vol. 13, no. 3. 780–784. (2002)
 21. Rand, W.: Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 846–850. (1971)
 22. Strehl, A., Ghosh, J.: Cluster ensembles--a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, vol. 3, 583–617. (2003)
 23. Asuncion, A., Newman, D.: UCI machine learning repository. (2007). [Online]. Available: <http://www.ics.uci.edu/~mllearn/MLRepository.html>
 24. Xia, T., Cao, J., Zhang, Y., Li, J.: On defining affinity graph for spectral clustering through ranking on manifolds. *Neurocomputing*, vol. 72, no. 13–15, 3203–3211. (2009)

An Improved Spectral Clustering Algorithm ased on Local Neighbors in Kernel Space

Xinyue Liu received the M.S degree in Computer Science and technology from Northeast Normal University, China, in 2006. She is currently working toward the Ph.D. degree in the School of Computer Science and Technology, Dalian University of Technology, Dalian, China. Her research interests include multimedia information retrieval, web mining and machine learning.

Xing Yong received his B.S. degree in Software Engineering from Dalian University of Technology in 2010, and currently is a master degree candidate in the same place. His major interests lie in machine learning.

Hongfei Lin received the Ph.D degree from Northeastern University, China. He is a professor in the School of Computer Science and Technology, Dalian University of Technology, Dalian, China. His professional interests lie in the broad area of information retrieval, web mining and machine learning, affective computing.

Received: April 15, 2011; Accepted: June 15, 2011.

