Inductive Sparse Subspace Clustering

X. Peng, L. Zhang and Z. Yi

Sparse Subspace Clustering (SSC) has achieved state-of-the-art clustering quality by performing spectral clustering over a ℓ^1-norm based similarity graph. However, SSC is a transductive method which does not handle with the data not used to construct the graph (out-of-sample data). For each new datum, SSC requires solving n optimization problems in $O(n)$ variables, where n is the number of data points. Therefore, it is inefficient to apply SSC in fast online clustering and scalable grouping. In this letter, we propose an inductive spectral clustering algorithm, called inductive Sparse Subspace Clustering (iSSC), which makes SSC feasible to cluster out-of-sample data. iSSC adopts the assumption that high-dimensional data actually lie on the low-dimensional manifold such that out-of-sample data could be grouped in the embedding space learned from in-sample data. Experimental results show that iSSC is promising in clustering out-of-sample data.

Introduction: Spectral clustering is one of the most popular subspace clustering algorithms, which aims to find a cluster membership of data points and the corresponding low-dimensional representation by utilizing the spectrum of a Laplacian matrix. The entries in the Laplacian matrix depict the similarity among data points. Thus, the construction of similarity graph lies on the heart of spectral clustering. In a similarity graph, the vertex denotes a data point and the connection weight between two points represents their similarity.

Recently, Elhamifar and Vidal [1] constructed a similarity graph by using ℓ^1-minimization based coefficient and performed spectral clustering over the graph, named Sparse Subspace Clustering (SSC). It automatically selects the nearby points for each datum by utilizing the principle of sparsity without pre-determination of the size of neighborhood. SSC has achieved impressive performance in images clustering and motion segmentation. However, it requires solving n optimization problems over n data points and calculating the eigenvectors of a $n \times n$ matrix, resulting in a very high computational complexity. In general, the time complexity of SSC is proportion to the cubic of data size. Thus, any medium-sized data set will bring up the scalability issues with SSC. In addition, SSC is a transductive algorithm which does not handle with the data not used to construct the graph (out-of-sample data). For each new datum, SSC needs performing the algorithm over the whole data set, which makes SSC inefficient to fast online clustering and scalable grouping.

To address the scalability issue and the out-of-sample problem in SSC, we propose an inductive clustering algorithm which is called inductive Sparse Subspace Clustering algorithm (iSSC). Out motivation derives from a widely-accepted assumption in manifold learning that the high-dimensional data actually lie on the low-dimensional manifold. Therefore, we could obtain the cluster membership of out-of-sample data by assigning them to the nearest cluster in the embedding space learned from well-sampled in-sample data. In other words, we resolve the out-of-sample problem in SSC by using subspace learning method. On the other hand, for large scale data set, we randomly split it into two parts, in-sample data and out-of-sample data, such that scalability issue could be addressed as an out-of-sample problem.

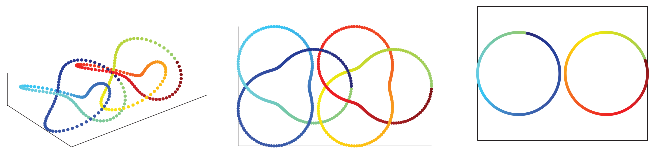
Inductive Sparse Subspace Clustering Algorithm: The basic idea of our approach is that: Suppose two data sets $\mathbf{Y} \in \mathbb{R}^{m\times p}$ (in-sample data) and $\mathbf{X} \in \mathbb{R}^{m\times n}$ (out-of-sample data) are drawn from multiple underlying manifolds of which each corresponds to a subspace. Provided $\mathbf{Y}$ is sufficient such that the manifolds are well-sampled, we expect to learn an embedding space with $\mathbf{Y}$ and group $\mathbf{X}$ in the embedding space since it is more compact and discriminative than the original space (See Fig. 1).

We make SSC feasible to cluster out-of-sample data in "subspace clustering, subspace learning and extension" manner. The first two steps are offline processes which only involve in-sample data, and the last one groups the out-of-sample data in online way.

To obtain the membership of in-sample data $\mathbf{Y}$, iSSC constructs a similarity graph by minimizing the following objective function,

$$\min \|\mathbf{c}_i\|_1 \qquad \text{s.t.} \quad \|\mathbf{y}_i - \mathbf{Y}_i\mathbf{c}_i\|_2 < \delta, \tag{1}$$

where $\mathbf{c}_i \in \mathbb{R}^p$ is the sparse representation of the data point $\mathbf{y}_i \in \mathbb{R}^m$ over the dictionary $\mathbf{Y}_i \triangleq [\mathbf{y}_1 \dots \mathbf{y}_{i-1}\ \mathbf{0}\ \mathbf{y}_{i+1} \dots \mathbf{y}_p]$, and $\delta \geq 0$ is the error tolerance.



(a) (b) (c)

Fig. 1 *A key observation. (a) some data points sampled from two 2-dimensional manifolds (trefoil-knots) which are embedded into a 3-dimensional space; (b) a plan view of the sampled data; (c) the embedding of the sampled data. It is easy to find that out-of-sample data points could be easily grouped into the correct cluster after they were projected into the embedding space.*

After getting the coefficients of $\mathbf{Y}$, we aim to group out-of-sample data $\mathbf{X}$ in the embedding space. In this letter, following the embedding program of Neighborhood Preserving Embedding algorithm (NPE) [3], we perform subspace learning to compute the projection matrix $\mathbf{W}$ via

$$\min_{\mathbf{W}} \left\|\mathbf{W}^T\mathbf{Y} - \mathbf{W}^T\mathbf{Y}\mathbf{C}\right\|_2^2, \quad \text{s.t. } \mathbf{W}^T\mathbf{Y}\mathbf{Y}^T\mathbf{W} = \mathbf{I}, \tag{2}$$

where $\mathbf{C} \in \mathbb{R}^{p\times p}$ is the collection of the sparse representation of $\mathbf{Y}$ produced by (1), and the constraint term aims at the scale-invariance.

The solution of (2) is given by solving the following generalized eigenvector problem:

$$(\mathbf{I} + \mathbf{C}^T\mathbf{C} - \mathbf{C} - \mathbf{C}^T)\mathbf{Y}^T\mathbf{W} = \lambda\mathbf{Y}^T\mathbf{W} \tag{3}$$

Once the optimal $\mathbf{W}$ is achieved, iSSC transforms out-of-sample data $\mathbf{X}$ in the embedding space via $\mathbf{W}^T\mathbf{X}$, and then assigns $\mathbf{X}$ to the nearest cluster in the space.

The steps of iSSC can be summarized as follows:

1. For in-sample data $\mathbf{Y}$, calculate the sparse representation coefficients $\mathbf{C}$ via solving
$$\min \|\mathbf{c}_i\|_1 \qquad \text{s.t.} \quad \|\mathbf{y}_i - \mathbf{Y}_i\mathbf{c}_i\|_2 < \delta.$$
2. Construct a Laplacian matrix $\mathbf{L} = \mathbf{S}^{-\frac{1}{2}}\mathbf{A}\mathbf{S}^{-\frac{1}{2}}$ by using the affinity matrix $\mathbf{A}$, where $\mathbf{S} = \mathrm{diag}\{s_i\}$ with $s_i = \sum_{j=1}^{p} a_{ij}$, a_{ij} is an entry of $\mathbf{A}$ and $\mathbf{A} = |\mathbf{C}| + |\mathbf{C}|^T$.
3. Obtain the matrix $\mathbf{V} \in \mathbb{R}^{p\times k}$ which consists of the first k normalized eigenvectors of $\mathbf{L}$ corresponding to its k smallest eigenvalues.
4. Get the segmentations of $\mathbf{Y}$ by performing k-means clustering algorithm on the rows of $\mathbf{V}$.
5. Suppose the desired dimensionality of embedding space is d, the projection matrix $\mathbf{W} \in \mathbb{R}^{m\times d}$ is given by the eigenvectors corresponding to d smallest nonzero eigenvalues of the following eigenvector problem:
$$\mathbf{M}\mathbf{Z}^T = \lambda\mathbf{Z}^T,$$
where $\mathbf{M} = \mathbf{I} + \mathbf{C}^T\mathbf{C} - \mathbf{C} - \mathbf{C}^T$ and $\mathbf{Z} = \mathbf{W}^T\mathbf{Y}$.
6. Project out-of-sample data $\mathbf{X}$ into the d-dimensional space via $\mathbf{W}^T\mathbf{X}$.
7. Search the nearest neighbor of $\mathbf{X}$ from $\mathbf{Y}$ in the embedding space, and assign $\mathbf{X}$ to the cluster that the neighbor belongs to.

Computational Complexity Analysis: Suppose in-sample data $\mathbf{Y} \in \mathbf{R}^{m\times p}$ drawn from k subspaces, we need $O(t_1mp^3 + t_2pk^2)$ to perform SSC over $\mathbf{Y}$, where t_1 and t_2 are the numbers of iteration of Homotopy optimizer [2] and k-means clustering algorithm, respectively. Moreover, we need $O(p^3)$ to compute the projection matrix $\mathbf{W}^T$. To group out-of-sample datum $\mathbf{X} \in \mathbf{R}^{m\times n}$, we need $O(dmn)$ to obtain its d-dimensional representation and $O(dpn)$ to search the nearest neighbor of $\mathbf{X}$ from $\mathbf{Y}$ in the embedding space. Note that, $p \ll n - p < n$.

Putting everything together, the computational complexity of iSSC is $O(t_1mp^3 + t_2pk^2 + dpn)$ owing to $d < m$ and $m < p$, where $m < p$ derives from the conditions of compressive sensing theory. Clearly, iSSC is more efficient than SSC whose time complexity is about $O(t_1mn^3 + t_2nk^2)$.

Baselines and Evaluation Metrics: We presented the experimental results of our approach over two real-world data sets, i.e., Extended Yale Database B (ExYaleB) and USPS. ExYaleB contains 2414 facial images of 38 subjects. We cropped the images from 192×168 to 48×42 and extracted 114 features by using PCA to retain 98% energy of the cropped data. USPS consists of 11000 handwritten digital images with 256 dimensionality over 10 classes.

We compared iSSC with three state-of-the-art inductive clustering algorithms, i.e., Nyström based spectral clustering [4], Spectral Embedding Clustering (SEC) [5] and Approximate Kernel K-means (AKK) [6]. Note that, Nyström based method and SEC have two variants, which are denoted as Nyström, Nyström_Orth, SEC_K and SEC_R, respectively. The approximate affinity matrix of Nyström is non-orthogonal, while that of Nyström-Orth is column-orthogonal. SEC_K performs k-means to get the clustering results and SEC_R adopts spectral rotation method to do it. Moreover, we reported the results of k-means clustering and SSC [1] over the whole database. The MATLAB code of iSSC can be downloaded at *http://www.machineilab.org/users/pengxi/*.

Accuracy and Normalized Mutual Information (NMI) are used to measure the clustering quality of the tested methods. The value of Accuracy or NMI is higher, the performance of the algorithm is better.

In all experiments, the tuned parameters for the algorithms were applied to achieve their best Accuracy. Specifically, iSSC and SSC adopted Homotopy optimizer [2] to solve ℓ^1-minimization problem. The optimizer needs two user-specified parameters, sparsity parameter λ and error tolerance parameter δ. We found a good value combination by setting $\lambda = (10^{-7}, 10^{-6}, 10^{-5})$ and $\delta = (10^{-3}, 10^{-2}, 10^{-1})$. Moreover, iSSC groups out-of-sample data in a low-dimensional space which preserves 98% energy of the embedding space learned from in-sample data. For the other competing methods, we set the value range for different parameters by following the configurations in [4, 5, 6].

Results: To examine the effectiveness of iSSC, Nyström, SEC and AKK, we randomly selected a half of images (1212) from ExYaleB and 1000 images from USPS as in-sample data, respectively. And the remaining samples are used as out-of-sample data. For k-means and SSC that cannot handle the out-of-sample data, we reported their results over the whole data set without data partition.

Table 1: Performance comparisons in different algorithms over ExYaleB.

Algorithms	Accuracy	NMI	Time(s)
iSSC (1e-6, 1e-3)	**59.69**%	**62.77**%	24.88
Nyström (12)	25.72%	46.57%	**9.33**
Nyström_Orth (2)	21.71%	41.74%	58.87
SEC_K (1e+12, 5, 1)	11.02%	11.09%	34.91
SEC_R (1e+9, 4, 1)	5.97%	4.31%	19.96
AKK (0.4)	8.00%	9.01%	9.94
SSC (1e-6, 1e-3)	**64.75**%	**68.10**%	310.30
k-means	9.03%	11.20%	37.05

Table 2: Performance comparisons in different algorithms over USPS.

Algorithms	Accuracy	NMI	Time(s)
iSSC (1e-7, 0.01)	**52.93**%	**52.90**%	41.52
Nyström (14)	47.66%	44.42%	**15.91**
Nyström_Orth (0.5)	50.70%	44.60%	183.37
SEC_K (1e-9, 3, 1)	47.63%	42.28%	43.38
SEC_R (1e-6, 4, 1)	11.70%	1.44%	19.78
AKK (0.3)	48.49%	46.79%	16.81
SSC (1e-7, 0.1)	**58.55**%	**59.76**%	7157.04
k-means	46.54%	45.61%	250.82

Tables 1-2 report the clustering quality and the time costs of the tested algorithms over the data sets. In the parenthesis, we also show the tuned parameters when the best Accuracy was achieved. From the results, we have the following observations:

- In all the tests, iSSC demonstrates an elegant balance between running time and clustering quality. Although iSSC is not the fastest algorithm, it outperforms the other tested methods with considerable performance margins in Accuracy and NMI. For example, iSSC achieved 33.97% gains in Accuracy and 16.20% gains in NMI over the second best algorithm when ExYaleB database was used to test.
- The clustering quality of iSSC is slight lower than that of SSC, but the speed of iSSC is 12.47 and 172.38 times faster than that of SSC over two data sets. Clearly, iSSC makes SSC feasible in large scale setting, which verifies the efficacy of our method.

Conclusion: In this letter, we have presented an inductive spectral clustering algorithm, called inductive Sparse Subspace Clustering (iSSC). The algorithm is an out-of-sample extension of Sparse Subspace Clustering algorithm (SSC) [1], which not only makes SSC feasible to cluster out-of-sample data, but also speed up SSC with hundred times. Experimental results with facial image and digital image clustering indicate the effectiveness of iSSC comparing with several state-of-the-art approaches.

Acknowledgment: The Authors would like to thank the anonymous reviewers for their valuable comments. This work was supported by National Basic Research Program of China 973 Program under Grant No. 2011CB302201, and Doctoral Fund of Ministry of Education of China under Grant No. 20110181110049.

X. Peng, L. Zhang and Z. Yi (*College of Computer Science, Sichuan University, Chengdu, 610065, P. R. China.*)

E-mail: pangsaai@gmail.com; leizhang@scu.edu.cn; zhangyi@scu.edu.cn

References

1 Elhamifar, E. and Vidal, R., 'Sparse Subspace Clustering: Algorithm, Theory, and Applications', To appear in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

2 Yang, A., Ganesh, A., Sastry, S., and Ma, Y., Fast L1-Minimization Algorithms and an Application in Robust Face Recognition: A Review', (EECS Department, University of California, Berkeley, 2010)

3 He, X., Cai, D., Yan, S., and Zhang, H., Neighborhood Preserving Embedding', *IEEE International Conference on Computer Vision*, (IEEE, 2005)

4 Chen, W.-Y., Song, Y., Bai, H., Lin, C.-J., and Chang, E.Y., 'Parallel Spectral Clustering in Distributed Systems', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011, **33**, (3), pp. 568-586.

5 Nie, F., Zeng, Z., W., T.I., Xu, D., and Zhang, C., 'Spectral Embedded Clustering: A Framework for in-Sample and out-of-Sample Spectral Clustering', *IEEE Transactions on Neural Networks*, 2011, **22**, (11), pp. 1796-1808.

6 Chitta, R., Jin, R., Havens, T.C., and Jain, A.K., 'Approximate Kernel K-Means: Solution to Large Scale Kernel Clustering', *ACM SIGMOD international conference on Knowledge discovery and data mining*, (2011)