Two Modified Sparse Subspace Clustering

XIAOYAN WANG
North University of China
School of Information
and Communication Engineering
Xueyuan Road 3, 030051 Taiyuan
CHINA
xywang00@126.com

CAIXIA ZHANG
North University of China
School of Science
Xueyuan Road 3, 030051 Taiyuan
CHINA
18535152702@163.com

YANPING BAI
North University of China
School of Science
Xueyuan Road 3, 030051 Taiyuan
CHINA
baiyp666@163.com

Abstract: Sparse Subspace Clustering constructs a sparse similarity graph by using the coefficient of sparse representation to subspace clustering. Based on sparse representation techniques, the algorithm gets the sparse coefficient by using l^1-minimization and gets clusters' data by spectral clustering algorithm. The spectral clustering algorithm depends on k-means algorithm for data clustering, while k-means algorithm is sensitive to the choice of initial starting conditions and it needs iterations. In order to avoid the drawbacks of k-means algorithm, we propose two modified Sparse Subspace Clustering algorithm, then the results are not be affected by the centers or the iterations. In one of the method, we get the clusters by comparing the positions of nonzero elements in the sparse adjacent matrix of similarity graph and the eigenvector. And in the second method, we use SOM algorithm instead of k-means algorithm. The experiment results show our proposed algorithm outperforms the initial Sparse Subspace Clustering.

Key–Words: subspace clustering,spectral clustering,k-means,SOM

1 Introduction

Clustering is one of classic problems in pattern recognition, image processing, machine learning and statistics [1, 2], which aims to partition a collection of patterns into disjoint clusters, such that patterns in the same cluster are similar, however patterns belong to two different clusters are dissimilar. In many areas of machine learning, image processing and computer vision applications require processing and representation of high-dimensional data. In fact, these high-dimensional data often can be represented by a low-dimensional subspace. For example, face images of a object under varying illumination conditions can be well approximated by 9-dimensional linear subspace [3]. Subspace clustering [4] separates data according to their underlying subspaces and is applied to image processing [5] and computer vision [6].

Recently, Elhamifar and Vidal [7, 8] have introduced an approach which constructed a sparse similarity graph by using the coefficient of sparse representation to subspace clustering, called Sparse Subspace Clustering(SSC). The algorithm based on sparse representation techniques and got the sparse coefficient by using l^1-minimization, and clustered data by spectral clustering. The spectral clustering algorithm employed k-means algorithm for data clustering, however the k-means algorithm is sensitive to the choice of initial starting conditions [9, 10] and the process of the algorithm need iterations. In the literatures, some modified algorithms are available: Scalable Sparse Subspace Clustering(SSSC) [11] and Reweighted Sparse Subspace Clustering [12]. Kernel version [13, 14] can also be gotten. Patel et al. [15] introduces an algorithm Latent Space Sparse Subspace Clustering based on SSC.

In this paper, two modified SSC algorithms are proposed in order to avoid the drawbacks of k-means which gets uncertain results and needs iterations. We also improve the efficiency of the algorithm. By observing the structure of the adjacent matrix, we find that the nonzero element of columns of matrix is in the same position as the corresponding eigenvector, therefore we can get the clusters by comparing nonzero elements in the columns of adjacent matrix and the eigenvectors. k-means algorithm also can be instead of by SOM algorithm which can gets better results. We do experiments on synthetic data sets and face clustering data sets, the results show that our proposed algorithms are better than original SSC [7, 8].

The rest of paper is organized as follows. We briefly describe the sparse subspace clustering in section 2. In section 3 we analysis SSC algorithm and propose our algorithms. Experimental evaluation is presented in section 4. Finally section 5 conclude our work.

2 SSC

Elhamifar and Vidal [7, 8] proposed the SSC algorithm for subspace clustering based on sparse representation technique. The main idea of SSC algorithm is to solve the optimization problem

$$\min \|c_i\|_1, \;\; y_i = Yc_i, \;\; c_{ii} = 0 \tag{1}$$

where $c_i \in R^N (i = 1, 2, ..., N)$ is the sparse representation of data point $y_i \in R^d$ as a linear combination of $Y = [y_1, ..., y_i, ..., y_N]$.

After solving the optimization problem, using the sparse coefficient formes the similarity graph and then clusters the data by spectral clustering.

Algorithm1 SSC

Step1 Input a set of points $y_i \in R^d (i = 1, 2, ..., N)$.

Step2 Solve the optimization problem 1 to attain the sparse coefficient $c_i (i = 1, 2, ..., N)$, and normalize $c_i = \frac{c_i}{\|c_i\|_\infty}$.

Step3 Form a adjacent matrix W of the similarity graph defined by C, where $C = [c_1, c_2, ..., c_N]$ and $W = |C| + |C|^T$.

Step4 Apply spectral clustering to the similarity graph.

3 The proposed algorithm

3.1 Analysis of SSC algorithm

In the last step of SSC, it uses spectral clustering for data cluster. The accurate procedure is as follows.

Algorithm 2 Spectral clustering

Step1 Form Laplacian matrix by $L = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, where $D = diag\{d_i\}$ with $d_i = \sum_{j=1}^{N} W_{ij}$.

Step2 Calculate multiplicity k of the zero eigenvalue of Laplacian matrix L and get the eigenvector $u_i (i = 1, 2, ..., k)$ of L corresponding to the zero eigenvalue.

Step3 Set $U = (u_1, u_2, ..., u_k)$ and normalize the rows of U to norm 1, that is set $u_{ij} = u_{ij} / (\sum_{l=1}^{k} u_{il}^2)^{1/2}$.

Step4 Get the clusters of data by performing k-means algorithm on the row of U.

The k-means algorithm is applied in the last step of the spectral clustering.

From literatures [7, 8] we knew that in a proper order adjacent matrix W can be written as a block diagonal form

$$W = \begin{bmatrix} W_1 & 0 & \cdots & 0 \\ 0 & W_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & W_k \end{bmatrix} \tag{2}$$

The eigenvector $u_i (i = 1, 2, ..., k)$ of L corresponding to the zero eigenvalue can be written as $u_i = [0, 0, ..., 1_{N_i}^T, 0, ..., 0]$ for $i \in \{1, 2, ..., k\}$ in a proper order and it is clear to see that nonzero elements of u_i locating at the position of N_i is in keeping with W_i.

3.2 MSSC

Through above discussion, we find that the nonzero value in eigenvector is relative to the block of adjacent matrix W. Therefore, in a proper order, we can estimate which cluster the data should be in by the positions of nonzero element of W. Hence we use nonzero position for clusters instead of k-means algorithm.

Given a set of data points $y_{i=1}^N$ drawn from a independent linear subspace $S_{i=1}^n$, where $d_i, \;\; N_i$ are the unknown dimension and number of the subspace S_i respectively. Let $Y = [y_1, ..., y_i, ..., y_N]$ and in a proper order, Y can be represent in the form of $Y = [Y_1, Y_2, ..., Y_n]$ where $Y_i \in S_i$. Each y_i has a representation by Y, and SSC algorithm needs the most sparse representation, that is

$$\min \|c_i\|_0, \;\; y_i = Yc_i, \;\; c_{ii} = 0 \tag{3}$$

where $\|c_i\|_0$ is the l_0 norm of c, i.e. the number of nonzero elements. Since 2 is an NP-hard problem, as the Basis Pursue (BP) [16] algorithm, it is replaced by l_1 optimization problem 1 based on the Basis Pursue (BP) [16] algorithm.

After getting solution C of 1, we form a similarity graph G, whose adjacent matrix is W, and $W = |C| + |C|^T$. The Laplacian matrix of G is $L = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, where $D = diag\{d_i\}$ with $d_i = \sum_{j=1}^{N} W_{ij}$. As the proposition in [7], the multiplicity of the zero eigenvalue of the Laplacian matrix L corresponding to the graph G is equal to the number of connected components of the graph. Also, the components of the graph can be determined from the eigenspace of the zero eigenvalue. More precisely, if the graph has n connected components, then $u_i = [0, 0, ..., 1_{N_i}^T, 0, ..., 0]$ is the $i - th$ eigenvector of L corresponding to the zero eigenvalue and it is easy to see that the position of N_i nonzero elements of u_i is in keeping with W_i. Therefore we use the eigenvectors as the center of clusters and cluster data by

just calculating the number of nonzero elements at the same position between the data and the centers.

Algorithm 3 Modified Sparse Subspace Clustering(MSSC)

Step1 Input a set of points $y_i \in R^d (i = 1, 2, ..., N)$.

Step2 Solve the optimization problem1 to attain the sparse coefficient $c_i (i = 1, 2, ..., N)$, and normalize $c_i = \frac{c_i}{\|c_i\|_\infty}$.

Step3 Form a adjacent matrix W of the similarity graph defined by C, where $C = [c_1, c_2, ..., c_N]$ and $W = |C| + |C|^T$.

Step4 Form Laplacian matrix by $L = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, where $D = diag\{d_i\}$ with $d_i = \sum_{j=1}^{N} W_{ij}$.

Step5 Calculate multiplicity k of the zero eigenvalue of Laplacian matrix L and get the eigenvector $u_i (i = 1, 2, ..., k)$ of L corresponding to the zero eigenvalue.

Step6 Set $U = (u_1, u_2, ..., u_k)$ and calculate the number of nonzero elements at the same position between the data and the centers.

Step7 Get the clusters of data by the maximum number of nonzero elements at the same position between the data and the centers.

3.3 SSCSOM

In this section, we modify the SSC based on self-organizing map(SOM). SOM algorithm was originally devised by Teuvo Kohonen [18, 19]. SOM operates in two phases: training and mapping. Training phase builds the map using input examples. It is a competitive process, also called vector quantization. Mapping automatically classifies a new input vector. SOM consists of components called nodes or neurons. Each node is associated with a weight vector of the same dimension as the input data vectors and a position in the map space. The procedure for placing a vector from data space onto the map is to first find the node with the closest weight vector to the vector taken from data space. Once the closest node is located it is assigned the values from the vector taken from the data space.

Algorithm 4 SOM

Step1 Initialize the map's nodes' weight vectors(w_{ij}) and the maximum iteration T.

Step2 Input training vector(X).

Step3 Use Euclidean distance formula to find similarity between the input vector and the map's node's weight vector.

Step4 Track the node that produces the smallest distance(this node is the best matching unit, BMU).

Step5 Update the nodes in the neighbourhood of BMU by pulling them closer to the input vector

$w_{ij}(t+1) = w_{ij}(t) + \eta(X_i - w_{ij})$

Step6 Let $t = t + 1$. Repeat from Step2 while $t < T$.

Algorithm5 SSCSOM

Step1 to Step5 are the same as Algorithm3.

Step6 Set $U = (u_1, u_2, ..., u_k)$. Let each row of U as a input vector of SOM.

step7 Get clusters of data by SOM algorithm.

4 Experiment Evaluation

In the following subsections we provide experimental results comparing the SSC algorithm with our algorithms.

4.1 Synthetic data sets

Four synthetic data sets S_1, S_2, S_3, S_4 are tested in this section. The results are compared from MSSC to SSC. They are all generated from pseudo random values drawn from the standard uniform distribution on the open interval(0,1). Generated a matrix of $10 * 36$ by function of rand, divided the matrix for three components and each 12 columns as a block embedded in a large matrix of $500 * 36$. The first block is in row 1 to row 10, the second is in row 101 to row 110, and the third is in row 301 to row 310, the other elements in the large matrix are zero. Regard each column of the matrix as a data point belonging to S_1, hence all columns in the large matrix composed data set of S_1. Clearly, there are three classes in S_1. S_2 data set has the same data as S_1, but the data are in a random orders.

Generated a matrix of $15 * 120$ by function of rand, divided the matrix for three components and each 20 columns as a block embedded in a large matrix of $1000 * 120$ and there are 6 blocks. The first block is in row 1 to row 15, the second is in row 101 to row 115, and the third is in row 201 to row 315, the $i - th$ block is in row $(i - 1) * 100 + 1$ to row $(i - 1) * 100 + 15$, the other elements in the large matrix are zero. Regard each column of the matrix as a data point belonging to S_3, hence all columns in the large matrix composed data set of S_3. S_4 data set has the same data point as S_3, but the data points are in a random orders.

4.2 Face clustering

In this section, we evaluate the clustering performance on the Extended Yale B database [17]. The data set contains of 192×168 pixel cropped face images of 38 individual, where there are 64 images for each subject acquired under various lighting conditions. In

Table 1: Comparative results on Synthetic data set

data set	method	clustering error	time
S_1	SSC	0	2.87 ± 0.93
S_1	MSSC	0	**1.72**
S_2	SSC	0	2.06 ± 0.07
S_2	MSSC	0	**1.73**
S_3	SSC	0	18.07 ± 1.87
S_3	MSSC	0	**14.88**
S_4	SSC	0	14.99 ± 0.17
S_4	MSSC	0	**14.79**

order to reduce the computational cost, we downsample the images to 48×42 pixel, then each face image is a 2016 dimension data. We do experiments on two and three subjects for five different groups.

Table 2: Comparative results on Extended Yale B

group	No. of subject	method	mean error	time
1	2	SSC	0	53.28
1	2	SSCSOM	0	**51.91**
2	2	SSC	0	53.93
2	2	SSCSOM	0	**53.80**
3	2	SSC	0	55.21
3	2	SSCSOM	0	**54.08**
4	2	SSC	0	58.42
4	2	SSCSOM	0	**54.73**
5	2	SSC	3.9	55.41
5	2	SSCSOM	1.6	**55.17**

4.3 Performance analysis

The comparison of the algorithms across the data sets is shown in Table 1-3. In Table 1, we find that both SSC algorithm and MSSC algorithm can attain the correct clustering, however, SSC algorithm is time consuming. In Table 2 and 3, the mean error of the SSCSOM algorithm is less than SSC algorithm, and also the time consuming. From the experiments, we see that our proposed algorithms can perform well in synthetic data sets and face clustering data sets than SSC algorithm.

5 Conclusions

We modify the SSC algorithm and the experiments show MSSC algorithm can get the same correct rate and costs little time than original algorithm;

Table 3: Comparative results on Extended Yale B

group	No. of subject	method	mean error	time
1	3	SSC	6.8	83.43
1	3	SSCSOM	**5.7**	**82.86**
2	3	SSC	1.0	83.06
2	3	SSCSOM	1.0	**81.71**
3	3	SSC	3.1	83.78
3	3	SSCSOM	**1.5**	**83.05**
4	3	SSC	0.5	83.35
4	3	SSCSOM	0.5	**82.11**
5	3	SSC	0	82.50
5	3	SSCSOM	0	**82.31**

the SSCSOM algorithm can both get more accurate and little time than SSC algorithm. However, the real data sets may not satisfy the conditions of the disjoint sets, the MSSC algorithm will not perform well in those data sets. In SSCSOM algorithm, when we do experiments on five subjects, the mean error is bigger than SSC. Hence the SSCSOM is not suitable for more clusters.

As for future work, we plan to find a method to circumvent the uncertainty in SSC algorithm and save time as well. Another side, we would like to apply the MSSC algorithm into real data sets.

Acknowledgements: The research was supported by the National Natural Science Foundation of China(grant No. 61275120,61573016).

References:

[1] R. Xu, D. C. Wunsch, Survey of clustering algorithms, *IEEE Trans. Neural Netw.* 16(3), 2005, pp. 645-678.

[2] Anil,K. Jain, Data clustering: 50 years beyond K-means, *Pattern Recognition Letters.* 31, 2010, pp. 651-666.

[3] R. Basri, D. W. Jacobs, Subspace Clustering, *Signal Processing Magazine.* 28(2), 2011, pp. 52-68.

[4] R. Vidal, Lambertian reflectance and linear subspaces, *IEEE Trans. Pattern Anal. Mach. Intell.* 25(2), 2003, pp. 218-233.

[5] W. Hong, J. Wright and K. Huang, Multi-Scale Hybrid Linear Models for Lossy Image Representation, *IEEE Trans. Image Processing.* 15(12), 2006, pp. 3655-3671.

[6] K. Kanatani, Motion Segmentation by Subspace Separation and Model Selection, *Proc. IEEE In-t'l Conf. Computer Vision.* 2, 2001, pp. 586-591.

[7] E. Elhamifar, R. Vidal, Sparse Subspace Clustering, *in: IEEE Conference on Computer Vision and Pattern Recognition, CVRP.* 2009, pp. 2790-2797.

[8] E. Elhamifar, R. Vidal, Sparse Subspace Clustering: Algorithm, theory, and applications, *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 35(11), 2013, pp. 2765-2781.

[9] M. E. Celebi, H. A. Kingravi and P. A. Vela, A comparative study of efficient initialization methods for the k-means clustering algorithm, *Expert Systems with Applications.* 40, 2013, pp. 200-210.

[10] J. M. Peña, J. A. Lozano and P. Larrañaga, An empirical comparison of four initialization methods for the K-Means algorithm, *Pattern Recognition Letters.* 20, 1999, pp. 1027-1040.

[11] X. Peng, L. Zhang and Z. Yi, Scalable Sparse Subspace Clustering, *IEE Conference on Computer Vision and Pattern Recognition.* 9(4), 2013, pp. 430-437.

[12] J. Xu, K. Xu, K. Chen and J. Ruan, Reweighted sparse subspace clustering, *Computer Vision and Image Understanding.* 138, 2015, pp. 25-37.

[13] M. Yin, Y. Guo and J. Gao etl, Kernel Sparse Subsapce Clustering on Symmetric Positive Definite Manifolds, arXiv:1601.00414v1[cs.CV] 4 , 2016.

[14] M. Vishal Patel, V. René, Kernel Sparse Subspace Clustering, *Image Processing (ICIP), 2014 IEEE International Conference.* 2014, pp.2849-2853.

[15] V. M. Patel, H. V. Nguyen and R. Vidal, Latent Space Sparse Subspace Clustering, *Selected Topics in Signal Processing IEEE Journal of* 9(4), 2015, pp. 691-701.

[16] D. L. Donoho, For most large underdetermined systemms of linear equations the minimal l^1-norm solution is also the sparest solution, *Communications on Pure and Applied Mathematics.* 59(6), 2006, pp. 797-829.

[17] K. −C. Lee, J. HO, and D. Kriegman, Acquiring Linear Subspaces for Face Recognition under Variable Lighting, *IEEE Trans. Pattern Analysis and Machine Intelligence.* 27(5), 2005, pp. 684-698.

[18] T. Kohonen, Self−organized formation of topologically correct feature maps, Biol. Cybernet. *Biol. Cybernet.* 43, 1982, pp. 59-69.

[19] T. Kohonen, Analysis of a simple self-organizing process, *Biol. Cybernet.* 44, 1982, pp. 135-140.