A Review on Clustering with Sparse Representation and Low Rank Recovery

Yi Han
Department of Electrical and
Computer Engineering
Rutgers, the State University of New Jersy

***Abstract*—In this report I explored two novel clustering techniques based on spectral clustering. The sparse subspace clustering utilizes the concept of sparse representation and acquires a representation that is able to reveal the subspace memberships of the data. The clustering via low rank recovery work on a similar fashion but instead seeks a low rank representation of the data. Both methods fit in the framework of spectral clustering. I evaluate the two techniques on some popular applications such as motion segmentation and face categorization.**

I. Introduction

Clustering is a major problem in machine learning, pattern recognition and all other related fields, it has applications in some of the most popular fields including computer vision, bio-informatics, etc. Clustering tries to separate data form a mixture of classes, reveal their true memberships in an unsupervised way. A common choice of modeling the mixture of data is the linear subspaces, since it's of lower computational complexity and appears to be rather effective in practice. linear Subspace technique has gained much attention in recent years, from the works in compressed sensing and matrix completion that model the data by a low-dimension linear subspace, to the sparse subspace clustering and clustering based on low rank recovery that is to be discussed in this paper.

A variety of works exists on the topic of subspace clustering. Iterative approaches such as K-subspace [1] iterate between assigning subspace memberships to data points and fitting subspaces to each cluster. Statistical approaches model the data to be generated from a mixture of probability models [2], and try to estimate the parameters of each individual distribution as well as the mixture coefficients by applying EM algorithm to the model. The aforementioned approaches require the number and dimension of subspaces to be known, and is sensitive to initialization. Algebraic methods, such as Generalized Principal Component Analysis [3], fit the data with a polynomial whose gradient at a point gives a vector normal to the subspace containing that point. Factorization-based methods [4] find an initial segmentation by thresholding the entries of a similarity matrix built from the factorization of the matrix of data points. Spectral clustering related approaches, which is the underlying framework of the two methods to be discussed, utilize the similarity matrix and then apply spectral clustering from the spectral graph theory to the matrix.

In this paper I explored two popular clustering techniques based on the framework of spectral clustering, which utilize sparse representation and low rank recovery respectively. Sparse subspace clustering(SSC) [5] tries to form the similarity matrix with sparse representations, it claims that by seeking the sparsest representation of data points with respect to the rest of the dataset, subspace membership can be revealed. Low rank recovery based method [6] work in a similar fashion, instead of seeking sparse representation, it tries to find the low rank representation of the whole dataset based on some specific 'dictionary', again, it has been proved that the result optimal low rank matrix is able to reveal the subspace membership of each data point [7] [8].

This rest of the paper is organized as such: In section I explain the concepts of techniques discussed in the paper, including the outline of spectral clustering, introduction on the theories of Sparse representation(SR) and Low rank recovery(LRR), as well as how they fit into the spectral clustering framework. In section II I test these two clustering techniques on two popular applications, motion segmentation and face categorization. Section III gives the conclusion and possible future work.

II. Techniques

A. Spectral Clustering

Spectral clustering [9] utilizes concepts from the spectral graph theory. Several steps including computing the similarity matrix, doing eigenvalue decomposition as performed to apply spectral clustering to the data.

1) Similarity Gragh
   We can form a similarity graph $G = (V, E)$ according to the data points. The vertices v_i represents the data points while links represents the connection between two data points, higher weight on the link means more similarity between two different data points. Common choice of computing the similarity graph including *The ε-neighborhood graph* which considers the neighbor points within the distance of ε, or *k-nearest neighbor graph* which take only the k nearest neighbor points into consideration. Each similarity gragh corresponding to a similarity matrix, in which the ij entry of the matrix represents the weight of the link between v_i and v_j.
2) Gragh Laplacians
   Graph laplacians is the main idea lying under spectral clustering. According to the spectral graph theory, some

linear algebras of the graph laplacian matrix reveal important properties of the data distribution. We can compute graph laplacians in many ways, for example we compute normalized laplacian matrix with:

$$L = D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{-1/2} \quad (1)$$

The multiplicity of the zero eigenvalue revealed the connected components in the similarity in the graph, which is actually the true number of clusters in the data. The eigenspace is spanned by $D^{-1/2}\mathbf{1}_{A_i}$, where A_i is the corresponding connected components.

3) Spectral Clustering Algorithm
Instead of facing the whole dimension data, spectral clustering apply traditional clustering techniques on the eigenvectors of the zero eigenvalue of the laplacian matrix. Since according to the important property introduced above, these eigenvectors are able to reveal the truth subspace membership of the data points, the whole algorithm is described below: Here k-means

Algorithm 1 Normalized Spectral Clustering

Input: Data, number of clusters k.
- Compute the adjacent matrix W.
- Compute Laplacian matrix L.
- Solve the generalized eigenvalue decomposition problem $Lu = \lambda Du$.
- Take eigenvectors corresponding to the k bottom eigenvalues as U.
- Apply k-means algorithm to the rows of U.

Output Subspace membership of each sample.

algorithm is applied to the row of the eigenvectors, it is because according to the indicator function shape of those eigenvectors, each row is a valid low dimension representation of the data points, or we can take this as a way of encoding the data points.

4) Graph cut point of view
In a Graph cut perspective, spectral clustering can be interpret as cutting the whole graph into a number of connected components while at the same time keep the links been cut minimized:

$$\min_{A_1,\ldots,A_k} \frac{1}{2}\sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{\mathrm{vol}(A_i)} \quad (2)$$

where $W(,)$ is the function compute all links between two components, vol(A) is a way of describe the size of one component, it's the weight of all links with in the component. Normalizing the object function by the size of components can avoid trivial solutions. Solving the aforementioned optimization problem is NP-hard, by some transformation and relaxation we are able to get the solution. we define a indicated matrix:

$$\mathbf{H} = \begin{bmatrix}\mathbf{h_1}, & ..., & \mathbf{h_k}\end{bmatrix} \quad (3)$$

where

$$\mathbf{h_i} = \begin{bmatrix}0, & ..., & 1/\sqrt{\mathrm{vol}(A_j)}, & ..., & 0\end{bmatrix}^{\mathbf{T}}$$

apply $\mathbf{H}$ to the optimization problem and relax the discrete condition on the variable, we get the following standard trace minimization problem:

$$\min_{\mathbf{T}\in\mathbf{R}^{n\times k}} \mathrm{Tr}(\mathbf{T^T D^{-1/2} L D^{-1/2} T}) \quad \text{s.t. } \mathbf{T^T T = I}. \quad (4)$$

The solution is the matrix contains the first k eigenvectors of the laplacian matrix, which is consistent with the algorithm mentioned above.

B. Sparse Subspace Clustering

1) Sparse Representation: The idea of sparse representation comes from compressed sensing, which claims that a signal, when expressed in a proper basis, can be sparse, or most of its coefficients are zero, while preserve almost all the information in the original signal:

$$\min \ ||\boldsymbol{c_i}||_q \quad \text{s.t.} \quad \boldsymbol{y_i} = \mathbf{Y}\boldsymbol{c_i},\ \boldsymbol{c_{ii}} = \mathbf{0}. \quad (5)$$

by comparing different p, it can be noticed that as the value of p decreases, the sparsity of the representation increases. However since $l0$ norm minimization is NP-hard, relaxation is made to instead compute the $l1$ norm as and approximation.

2) Clustering based on Sparse Representation: Sparse subspace clustering first seeks the sparsest representation C of each data points with respect to the rest of the dataset, note that the point itself is excluded in the dictionary to avoid trivial solution. By the theory of sparse representation, each point should be well represented by the data points within the same class, and not by the data points in other classes, or the coefficients corresponding to these points are close to zero.

$$\begin{aligned}&\min_{\mathbf{C}} \ ||\mathbf{C}||_{\mathbf{1}} \\ &\text{s.t. } \mathbf{Y} = \mathbf{YC},\ \mathbf{1^T C} = \mathbf{1^T},\ \mathbf{diag(C)} = \mathbf{0}.\end{aligned} \quad (6)$$

note that the constraint $\mathbf{1^T C} = \mathbf{1^T}$ is to deal with affine subspaces. Then by compute $W = |C| + |C|^T$, we get a valid similarity matrix that indicated the subspace membership of each data point. Actually by an appropriate permutation the matrix is block diagonal(Figure 1).

$$\mathbf{C} = \begin{bmatrix}\mathbf{C_1} & ... & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \ldots & \mathbf{C_k}\end{bmatrix}\mathbf{\Gamma}$$

Once get the similarity matrix, we are able to fit it to the spectral clustering, and cluster the data points to their true classes.

3) Robustness to Noise and Error: In the case when the data is contaminated by noise or gross error, we revise the optimization problem:

$$\begin{aligned}&\min_{\mathbf{C},\mathbf{R},\mathbf{Z}} \ ||\mathbf{C}||_{\mathbf{1}} + \lambda_{\mathbf{e}}||\mathbf{E}||_{\mathbf{1}} + \frac{\lambda_{\mathbf{z}}}{\mathbf{2}}||\mathbf{Z}||_{\mathbf{F}}^{\mathbf{2}} \\ &\text{s.t. } \mathbf{Y} = \mathbf{YC} + \mathbf{E} + \mathbf{Z},\ \mathbf{1^T C} = \mathbf{1^T},\ \mathbf{diag(C)} = \mathbf{0}.\end{aligned} \quad (7)$$

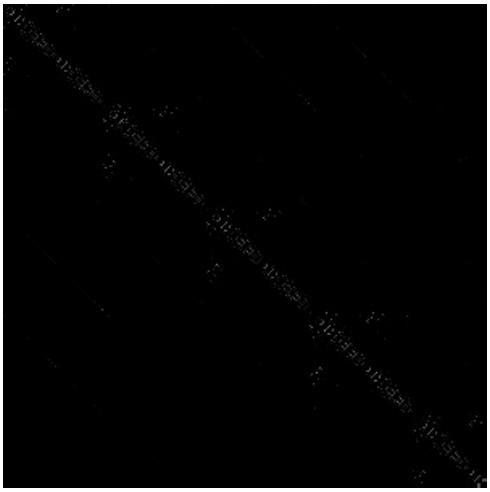

Fig. 1: The diagonal sparse coefficient matrix.

where the $l1$ norm term is for modeling the gross error while the Frobenius norm is for modeling the noise.

C. Subspace Clustering based on Low Rank Recovery

Similar to SSC, subspace clustering based low rank recovery works in a similar fashion of constructing a similar matrix based on the data points.

1) Low Rank Matrix Recovery: In low rank matrix recovery or completion [10], it is assumed that data lie in a subspace of much lower rank than what it likes, the noise ,corruption and gross error, should be account for the increase of rank. Low rank matrix recovery tries to reveal the underlying low rank structure while at the same time isolate the errors. PCA can be viewed as a way of low rank recovery, but it can only deal with small Gaussian noise. The low rank matrix recovery problem can be state as:

$$\min_{\mathbf{A}} \ \text{rank}(\mathbf{A}) + \lambda||\mathbf{E}||_{\mathbf{0}} \quad \text{s.t.} \quad \mathbf{D} = \mathbf{A} + \mathbf{E}. \tag{8}$$

the second term in the object function depends on what kind of corruption is modeled. Make relaxation on the object function, replace rank with nuclear norm, and $l0$ norm with $l1$ norm, we get a solvable optimization problem:

$$\min_{\mathbf{A}} \ ||\mathbf{A}||_* + \lambda||\mathbf{E}||_{\mathbf{1}} \quad \text{s.t.} \quad \mathbf{D} = \mathbf{A} + \mathbf{E}. \tag{9}$$

2) Clustering based on Low Rank Recovery: In the case of clustering, different from matrix completion, it is assumed that the underlying structure is not a single but a union of low rank subspaces, or

$$\min_{\mathbf{Z}} \ ||\mathbf{Z}||_* \quad \text{s.t.} \quad \mathbf{X} = \mathbf{XZ}. \tag{10}$$

it is proved that the solution to the above problem is the row space of the dictionary X, if let $U\Sigma V^T$ to be the singular value decomposition of X, then $Z = VV^T$, The solution matrix Z is block diagonal if arranged according to their groups:

$$\mathbf{Z}^* = \begin{bmatrix} \mathbf{Z_1^*} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{Z_k^*} \end{bmatrix} \tag{11}$$

By compute the optimal solution Z^*, we're able to construct the similarity matrix as:

$$W_{ij} = [(VV^T)_{ij}]^2 \tag{12}$$

and then fit into the framework of spectral clustering.

When facing sample specific corruption or outliers, the problem is revised as:

$$\min_{\mathbf{Z},\mathbf{E}} \ ||\mathbf{Z}||_* + \lambda||\mathbf{E}||_{\mathbf{2,1}} \quad \text{s.t.} \quad \mathbf{D} = \mathbf{XZ} + \mathbf{E}. \tag{13}$$

Note the in this case it is the column space of Z^* not Z^* itself that recoveries the row space of X, thus we compute the similarity matrix as:

$$W_{ij} = [(U^*)(U^*)^T)_{ij}]^2 \tag{14}$$

where $\mathbf{Z}^* = \mathbf{U}^*\mathbf{\Sigma}^*(\mathbf{V}^*)^{\mathbf{T}}$.

Different from SSC, LRR do the optimization over the whole matrix, not column by column, thus don't have the problem of resulting in a trivial solution.

III. Experiments

I evaluate the two algorithm on two of the popular problem, Motion Segmentation, which is a major task in video analysis, and face categorization, which is related to the biometrics.

A. Motion Segmentation

Motion segmentation tries to cluster trajectories in a video sequence according to their motions, rotation, translation, etc. The trajectories can be obtained by applying a trajectory tracker on the video sequence.

According to the affine camera model, the correspondence of 3D coordinates and coordinates on the image of an rigidly moving object can be expressed as:

$$\begin{bmatrix} x_{11} & \dots & x_{1p} \\ y_{11} & \dots & y_{1p} \\ \vdots & \ddots & \vdots \\ x_{f1} & \dots & x_{fp} \\ y_{f1} & \dots & y_{fp} \end{bmatrix} = \begin{bmatrix} A_1 \\ \vdots \\ A_F \end{bmatrix} \quad \begin{bmatrix} X_1 & \dots & X_p \\ Y_1 & \dots & Y_p \\ Z_1 & \dots & Z_p \\ 1 & \dots & 1 \end{bmatrix} \tag{15}$$

where A_i are the affine motion matrix. Thus we can see that the rank of the matrix on the right side, which is the product of the affine motion matrix and the structure matrix, is bounded by 4, and note that the last row of the structure matrix is actually all 1, thus we can conclude that trajectories of a single rigid motion lie in an affine subspace of dimension at most 3. By clustering trajectories, we are able to segment the various source motions.

Hopkins155 is a motion segmentation dataset which contains 155 sequences of two and three motions, the trajectories are extracted by a tracker. The trajectories in the dataset are

	LRR	SSC
ERROR(%)	12.23%	1.05%

TABLE I: Clustering error on Hopkins155 for both SSC and LLR

contaminated by noise, but there is no outliers or missing entries.

An example of the trajectory points are shown in 2, different, Three portion of the frame are tracked. By apply the

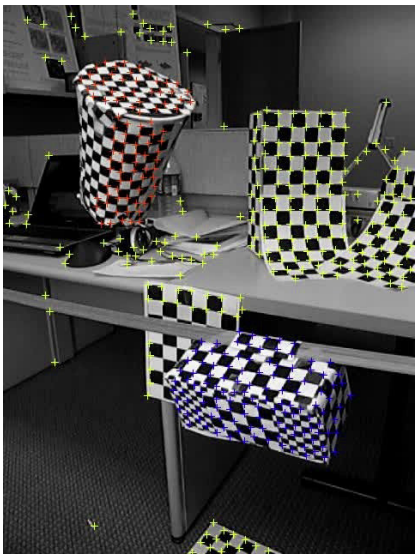

Fig. 2: Example trajectory points in the Hopkins155 dataset.

two algorithm to the dataset, as is shown in Figure 3, the overall subspace membership is recovered, The clustering error is shown in Table I, As can be seen, due to the tidiness of the whole dataset, the clustering error under both methods is low.

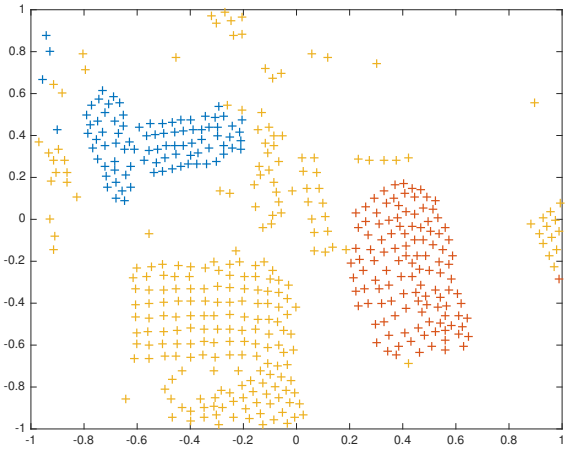

Fig. 3: Example trajectory points in the Hopkins155 dataset.

	LRR	SSC
ERROR(%)	53.17%	37.32%

TABLE II: Clustering error on YaleB for both SSC and LLR

B. Face Categorization

Face Categorization is a important problem in biometrics, it involves a lot in authentication and other issues related to security. Face Categorization consider each face image vector to be a data point to be clustered. It is said that face of a single individual under various illumination conditions and expressions lie in a 9 dimension linear subspace, thus we can apply subspace clustering technique to find the true class each data point belongs to.

I test the algorithms on the YaleB face dataset(Figure 4). YaleB contains 38 individual each of who have around 60 samples with different variations such the light condition. The clustering error is much higher than the Hopkins155 dataset(II), probably due to the high variation within the same class.

Fig. 4: Example faces in the YaleB dataset.

IV. Conclusion

In this paper I reviewed two popular subspace clustering methods, sparse subspace clustering and clustering via low rank recovery, by fitting the idea of sparse representation and low rank into the spectral clustering framework, they both achieved good performance. We evaluate these two techniques on the motion segmentation problem as well as the face categorization problems, the results shows their capability to handle the clustering job, however further adaptation or parameter adjust is needed to make better performance.

References

[1] Dingding Wang, Chris Ding, and Tao Li. K-subspace clustering. In *Machine learning and knowledge discovery in databases*, pages 506–521. Springer, 2009.

[2] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[3] Rene Vidal, Yi Ma, and Shankar Sastry. Generalized principal component analysis (gpca). *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(12):1945–1959, 2005.

[4] Amit Gruber and Yair Weiss. Multibody factorization with uncertainty and missing data using the em algorithm. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–707. IEEE, 2004.

[5] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2765–2781, 2013.

[6] Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *ICML*, pages 663–670, 2010.

[7] René Vidal and Paolo Favaro. Low rank subspace clustering (lrsc). *Pattern Recognition Letters*, 43:47–61, 2014.

[8] Vishal Patel, Hien Nguyen, and René Vidal. Latent space sparse subspace clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 225–232, 2013.

[9] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[10] John Wright, Arvind Ganesh, Shankar Rao, Yigang Peng, and Yi Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pages 2080–2088, 2009.