Postprint of the LPCA-Based Spectral Clustering Algorithm

Authors: Tong Tao, Wen Guoqiu, Malong Tan, Wu Lin, Du Tingting

Date: 2018-08-13T00:00:00+00:00

Abstract

To address the limitations of traditional spectral clustering—namely, its consideration of only global sample features while neglecting local features during relationship matrix construction, its typical requirement to specify the number of clusters a priori, and its inability to correctly partition intersection points—an improved spectral clustering algorithm based on local principal component analysis and connected graph decomposition is proposed. The algorithm first automatically learns and selects center points from the dataset, then employs local principal component analysis to obtain the dataset' s relationship matrix, and finally utilizes a connected graph decomposition algorithm to partition the relationship matrix. Experimental results demonstrate that the proposed improved algorithm outperforms existing classical algorithms.

Full Text

Preamble

Article URL: http://www.arocmag.com/article/02-2019-11-008.html
ChinaXiv Partner Journal: Computer Application Research

Title: Spectral Clustering Algorithm Based on LPCA

Authors: Tong Tao, Wen Guoqiu†, Tan Malong, Wu Lin, Du Tingting
Affiliation: Guangxi Key Laboratory of Multi-source Information Mining & Security, Guangxi Normal University, Guilin, Guangxi 541004, China

Abstract: Traditional spectral clustering algorithms suffer from three major limitations: (1) they consider only global sample structures while ignoring local structures when constructing correlation matrices; (2) they require prespecifying the number of clusters; and (3) they cannot correctly partition intersection points. This paper proposes an improved spectral clustering algorithm

based on local principal component analysis (LPCA) and connected graph decomposition. Specifically, the method automatically learns centroids from the dataset, obtains the correlation matrix using LPCA, and partitions this matrix through connected graph decomposition. Experimental results demonstrate that the proposed algorithm outperforms existing classical methods.

Keywords: local principal component analysis; spectral clustering; connected graph decomposition; intersection points

Classification: TP301.6
DOI: 10.3969/j.issn.1001-3695.2018.04.0283

0 Introduction

Clustering is a fundamental data processing technique that groups similar samples into the same cluster while assigning dissimilar samples to different clusters. Existing clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, and grid-based methods. Among these, K-means is widely adopted due to its simplicity and ease of implementation. However, K-means suffers from two prominent issues: initialization sensitivity of cluster centers and the requirement to pre-specify the number of clusters. Different initialization methods lead to different clustering results, while manual specification of cluster numbers demands prior experience or knowledge about data distribution.

To address these limitations, researchers have proposed various improved K-means algorithms. For instance, density-based methods determine cluster centers by calculating sample densities and assign nearby samples to these centers, simultaneously solving both problems of traditional K-means. However, such approaches require computing densities and distances for all samples, which is computationally expensive and still fails to handle intersection points correctly. Another approach introduces Gaussian distribution functions to enable clustering on non-convex datasets, but it still requires specifying the cluster number k.

This paper proposes a spectral clustering algorithm based on local principal component analysis and connected graph decomposition (SC-LPCA). The algorithm first automatically selects representative data points as centroids, then applies LPCA to these points' neighborhoods to construct a correlation matrix that captures local data characteristics, and finally partitions this matrix using connected graph decomposition. The resulting centroid clusters serve as anchors for assigning all remaining points based on proximity. Compared to traditional clustering methods, SC-LPCA offers four key advantages: (a) by clustering selected centroids rather than the entire dataset, it significantly reduces computational complexity; (b) LPCA enables the correlation matrix to better describe local data features, improving clustering performance; (c) the combination of

centroid selection and LPCA processing effectively handles intersection points in multi-manifold datasets; and (d) connected graph decomposition eliminates the need to pre-specify cluster numbers, reducing clustering complexity and difficulty.

1.1 Local Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction technique that projects high-dimensional data onto a lower-dimensional space. Given a sample dataset $\mathbf{X} \in \mathbb{R}^{d \times n}$ (where d is the number of attributes and n is the number of samples), PCA first centers the data ($\mathbf{x}'_i = \mathbf{x}_i -$), then computes the covariance matrix and performs eigendecomposition. The projected coordinates are sorted by eigenvalue magnitude, and the top d' dimensions are retained. The resulting representation $\mathbf{z}_i = (z_{i1}, z_{i2}, ..., z_{id'})^T$ captures the global structure of the dataset since covariance measures overall dispersion.

Local Principal Component Analysis (LPCA) improves upon PCA by focusing on local rather than global data distributions. Instead of analyzing the entire dataset, LPCA examines the relationship between individual samples and their neighborhoods. By performing covariance analysis and eigendecomposition on neighborhood subsets, LPCA better reflects relationships between nearby samples. The resulting correlation matrix more accurately represents the true data structure and preserves local features, which is crucial for enhancing clustering performance.

1.2 Spectral Clustering

Spectral clustering transforms the clustering problem into a graph partitioning problem using graph theory. Given a graph $G = (V, E)$, each sample is treated as a vertex $v \in V$, and correlations between samples define edges $e \in E$. The correlation matrix $\mathbf{W}$ is typically constructed using Euclidean distances: $w_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2)$. The degree of a vertex $d_i = \sum_{j=1}^{n} w_{ij}$ represents the sum of weights for all edges connected to that point. The degree matrix $\mathbf{D}$ is a diagonal matrix containing all vertex degrees. The Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is then computed. Eigendecomposition of $\mathbf{L}$ yields the top k eigenvectors, which form a new feature matrix $\mathbf{F}$. Treating each row of $\mathbf{F}$ as a sample, K-means is applied to obtain final clusters. Spectral clustering elegantly handles non-convex datasets through graph representation, but still requires pre-specifying k.

1.5 Algorithm Pseudocode

SC-LPCA Algorithm Pseudocode

Input: Training set $\mathbf{X} \in \mathbb{R}^{d \times n}$, feature vector dimension d'.

Output: Clustering results.

1. Randomly select a sample $\mathbf{y}_1$ from $\mathbf{X}$. Let r be the neighborhood threshold. The neighborhood subset of $\mathbf{y}_1$ is $\mathcal{N}_r(\mathbf{y}_1) = \{\mathbf{x}_j \mid \|\mathbf{x}_j - \mathbf{y}_1\| \leq r, j \in [1, n]\}$.
2. Randomly select a sample $\mathbf{y}_2$ not in $\mathcal{N}_r(\mathbf{y}_1)$. Repeat this selection process n_0 times to obtain a new dataset $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_{n_0}\}$.
3. For each sample $\mathbf{y}_i$ in $\mathbf{Y}$, compute the covariance matrix $\mathbf{C}_i$ of its neighborhood subset $\mathcal{N}_r(\mathbf{y}_i)$:

$$\mathbf{C}_i = \frac{1}{|\mathcal{N}_r(\mathbf{y}_i)|} \sum_{\mathbf{x} \in \mathcal{N}_r(\mathbf{y}_i)} (\mathbf{x} - \mathbf{y}_i)(\mathbf{x} - \mathbf{y}_i)^T$$

4. Perform eigendecomposition on $\mathbf{C}_i$ and retain the top d' eigenvectors to form the projection matrix $\mathbf{Q}_i$.
5. Compute the spatial threshold ε and projection scale threshold η:

$$\varepsilon = \max_{i,j} \|\mathbf{y}_i - \mathbf{y}_j\|_2, \quad i \neq j, i, j \in [1, n_0]$$

$$\eta = \text{median}_{i,j} \|\mathbf{y}_i - \mathbf{y}_j\|_2$$

6. Calculate the correlation matrix $\mathbf{W}$ based on $\mathbf{Y}$ and projection results $\mathbf{Q}$:

$$w_{ij} = \exp\left(-\frac{\|\mathbf{y}_i - \mathbf{y}_j\|^2}{\varepsilon^2}\right) \cdot \exp\left(-\frac{\|\mathbf{Q}_i - \mathbf{Q}_j\|^2}{\eta^2}\right)$$

7. Binarize $\mathbf{W}$ using threshold δ to obtain $\mathbf{W}^*$:

$$\delta = \text{median}_{i,j} w_{ij}$$

$$w_{ij}^* = \begin{cases} 1, & w_{ij} > \delta \\ 0, & \text{otherwise} \end{cases}$$

8. Apply connected graph decomposition on $\mathbf{W}^*$ to partition the selected centroids.
9. Assign remaining samples to the nearest centroid cluster based on Euclidean distance.

2 Algorithm Description

The SC-LPCA algorithm preserves local data characteristics by introducing neighborhood subsets of centroid samples and LPCA processing. This makes samples within the same cluster more compact and strengthens intra-cluster relationships. While covariance processing alone can separate intersecting clusters to some extent, it may fail when cluster angles are small. As shown in [Figure 1: see original paper], when the angle θ between two clusters is sufficiently small,

the distance between samples $\mathbf{p}_1$ and $\mathbf{p}_2$ from different clusters may be smaller than that between $\mathbf{p}_1$ and $\mathbf{p}_0$ from the same cluster, leading to misclassification. To address this, projection is introduced alongside covariance analysis. Projection makes samples from the same cluster more compact while separating different clusters, effectively solving the intersection partitioning problem for clusters with small angles.

The algorithm constructs a set of connected graphs from the correlation matrix $\mathbf{W}^*$. Each connected graph is recursively decomposed by computing a splitting threshold λ for the largest subgraph:

$$\lambda = \frac{e_{12}}{\min\{u_1, u_2\}}$$

where u_1 and u_2 represent the number of points in the two partitioned subgraphs, and e_{12} is the number of edges connecting them. The splitting threshold is compared against a tolerance threshold t:

$$t = \frac{e}{2b}$$

where e is the number of edges and b is the number of vertices in the largest subgraph. Decomposition continues recursively until $\lambda \leq t$ for all subgraphs, ensuring correct partitioning of all centroids. Finally, remaining samples are assigned based on distance to centroids.

By selecting centroids rather than clustering the entire dataset directly, SC-LPCA significantly reduces computational complexity. LPCA processing preserves local features and enhances clustering accuracy. The combination of centroid selection and LPCA effectively handles intersection points in multi-manifold datasets without requiring pre-specification of cluster numbers, making the algorithm more practical for real applications.

3 Experimental Results and Analysis

All algorithms were implemented in MATLAB 2014a and tested on a Windows 10 64-bit system with an Intel Core i7-7700 CPU @ 3.6 GHz and 8 GB RAM.

3.1 Comparison Algorithms and Evaluation Metrics

To evaluate SC-LPCA performance, we compared it against several clustering algorithms: K-means, LRR, LSR, SSQP, NCut, and SSC. K-means serves as the baseline. Parameter settings follow corresponding literature, with K-means using the true number of classes k where applicable. Although SC-LPCA does not require specifying k, results in tables are reported at the true class count for fair comparison.

Experiments were conducted on eight datasets from the UCI repository [http://archive.ics.uci.edu/ml/index.php]. Dataset details are shown in . Evaluation metrics include:

Accuracy (ACC):

$$\mathrm{ACC} = \frac{1}{n}\sum_{i=1}^{n}\delta(l_i, \hat{l}_i)$$

where l_i is the true label and $\hat{l}_i$ is the predicted label for sample i.

Normalized Mutual Information (NMI):

$$\mathrm{NMI} = \frac{\mathrm{MI}(U, V)}{\sqrt{H(U)H(V)}}$$

where $\mathrm{MI}(U, V)$ is mutual information between true labels U and predicted labels V, and $H(\cdot)$ denotes entropy.

3.3 Experimental Results and Analysis

Results across eight datasets are presented in , , and [Figure 2: see original paper]. shows that SC-LPCA achieves consistent improvements in accuracy over comparison algorithms. For the Cars dataset, SC-LPCA improves ACC by 26.02%, 22.45%, 24.49%, 17.09%, 19.39%, and 13.52% compared to K-means, LRR, LSR, SSQP, NCut, and SSC respectively, with an average improvement of 20.49%. These results demonstrate the algorithm' s effectiveness and validity.

shows NMI results, where SC-LPCA outperforms comparison algorithms on all datasets except Breast. For Cars, NMI improvements are 14.04%, 8.07%, 7.51%, 13.27%, 12.54%, and 11.01% respectively, averaging 11.07%. This indicates that SC-LPCA's correlation matrix better captures sample relationships. While SSC achieves good performance through sparse subspace constraints that capture global structures, it neglects local features. SC-LPCA' s LPCA processing preserves local characteristics and handles intersection points effectively, leading to superior NMI values.

Execution time comparisons in show SC-LPCA' s efficiency. For Cars, time reductions are 0.06s, 105.388s, 0.491s, and 0.428s compared to other algorithms. Centroid selection reduces dataset size, and LPCA projection decreases dimensionality, both contributing to faster computation.

[Figure 2: see original paper] illustrates clustering performance across different numbers of clusters. Results show that SC-LPCA does not always achieve optimal performance at the true class count. For example, the Auto dataset performs best at 4 clusters rather than the true 6 clusters. Since SC-LPCA automatically determines clusters through connected graph decomposition without manual specification, it can discover more natural groupings.

4 Conclusion

This paper proposes an improved spectral clustering algorithm that first selects representative samples for clustering, then generalizes to all data points. This

approach reduces computational complexity while LPCA processing preserves local features and improves accuracy. The algorithm handles intersection points effectively and eliminates the need to pre-specify cluster numbers, enhancing practical applicability. Comparative analysis across multiple datasets demonstrates superior performance, particularly on low-dimensional data, and shows that optimal clustering does not always occur at the true class count. Future work will focus on extending the algorithm to high-dimensional and large-scale datasets, as well as improving robustness to noise.

References

[1] Zhou Zhihua. Machine Learning [M]. Beijing: Tsinghua University Press, 2016: 197-217, 229-231.

[2] Sun Jigui, Liu Jie, Zhao Lianyu. Clustering algorithms research [J]. Journal of Software, 2008, 19(1): 48-61.

[3] Li Yonggang, Su Yijuan, He Wei, et al. Hypergraph and self-representation for spectral clustering [J]. Application Research of Computers, 2017, 34(6): 1621-1625.

[4] Wang Junjie. Density-sensitive K-means clustering algorithm [D]. Jinan: Shandong Normal University, 2014.

[5] Yin Nan. Expectation maximization clustering algorithm based on Gauss mixture model [J]. Statistic and Decision, 2017(4): 87-89.

[6] Deng Jianshuang, Zheng Qilun, Peng Hong, et al. Clustering algorithm based on dynamic division of connected graph [J]. Journal of South China University of Technology: Natural Science Edition, 2007, 35(1): 118-122.

[7] Rodriguez A, Laio A. Clustering by fast search and find of density peaks [J]. Science, 2014, 344(6191): 1492.

[8] Mika S, Smola A, Scholz M. Kernel PCA and de-noising in feature spaces [C]// Proc of Conference on Advances in Neural Information Processing Systems II. Cambridge: MIT Press, 1999: 536-542.

[9] Arias-Castro E, Lerman G, Zhang T. Spectral clustering based on local PCA [J]. Journal of Machine Learning Research, 2017, 18(1): 253-309.

[10] Yu X S, Shi Jianbo. Multiclass spectral clustering [C]// Proc of the 9th IEEE International Conference on Computer Vision. 2003: 313-319.

[11] He Xin, Wang Jiabing, Zhang Zhongxian, et al. Clustering Web documents based on Multiclass spectral clustering [C]// Proc of International Conference on Machine Learning and Cybernetics. 2011: 1466-1471.

[12] Zhu Xiaofeng, He Wei, Li Yonggang, et al. One-step spectral clustering via dynamically learning affinity matrix and subspace [C]// Proc of the 31st AAAI Conference on Artificial Intelligence. 2007: 2963-2969.

[13] Shah S A, Koltun V. Robust continuous clustering [J]. Proceedings of the National Academy of Sciences of the USA, 2017, 114(37): 9814.

[14] Liu Guangcan, Lin Zhouchen, Yan Shuicheng, et al. Robust recovery of subspace structures by low-rank representation [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2013, 35(1): 171-184.

[15] Lin Zhouchen, Liu Risheng, Su Zhixun. Linearized alternating direction method with adaptive penalty for low-rank representation [C]// Advances in Neural Information Processing Systems. 2011: 612-620.

[16] Lu Canyi, Min Hai, Zhao Zhongqiu, et al. Robust and efficient subspace segmentation via least squares regression [C]// Computer Vision. Berlin: Springer, 2012: 347–360.

[17] Chun H, Keleş S. Sparse partial least squares regression for simultaneous dimension reduction and variable selection [J]. Journal of the Royal Statistical Society, 2010, 72(1): 3.

[18] Lu Canyi, Hai Min, Zhao Zhongqiu, et al. Robust and efficient subspace segmentation via least squares regression [C]. Berlin: Springer-Verlag, 2012.

[19] Abdi H. Partial least squares regression and projection on latent structure regression (PLS Regression) [J]. Wiley Interdisciplinary Reviews: Computational Statistics, 2010, 2(1): 97–106.

[20] Lin Liyuan, Chen Xiaoyun, Jian C. Subspace segmentation via least squares regression including information about distance [J]. Microcomputer & Its Applications, 2016.

[21] Wang Shusen, Yuan Xiaotong, Yao Tiansheng, et al. Efficient subspace segmentation via quadratic programming [C]// Proc of the 25th AAAI Conference on Artificial Intelligence. 2011: 519-524.

[22] Shi Jianbo, Malik J. Normalized cuts and image segmentation [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2000, 22(8): 888-905.

[23] Xu Linli, Li Wenye, Schuurmans D. Fast normalized cut with linear constraints [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2009: 2866-2873.

[24] Elhamifar E, Vidal R. Sparse subspace clustering: algorithm, theory, and applications [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2012, 35(11): 2765-2781.

[25] Wang YuXiang, Xu Huan. Noisy sparse subspace clustering [J]. Journal of Machine Learning Research, 2013, 17(1): I-89.

[26] Peng Xi, Zhang Lei, Yi Zhang. Scalable sparse subspace clustering [C]// Computer Vision and Pattern Recognition. 2013: 430-437.

[27] Zhu Xiaofeng, Zhang Shichao, Hu Rongyao, et al. Local and global structure preservation for robust unsupervised spectral feature selection [J]. IEEE Trans

on Knowledge and Data Engineering, 2018, 30(3): 517-529.

Note: Figure translations are in progress. See original paper for figures.