Postprint of Spectral Clustering Based on Natural Nearest Neighbor Similarity Graph

Authors: Liu Youchao, Zhang Xihuang

Date: 2018-11-29T00:00:00+00:00

Abstract

Spectral clustering is a clustering algorithm based on spectral graph partitioning theory that has gained widespread popularity due to its superior performance on non-convex datasets. However, traditional spectral clustering algorithms often produce unsatisfactory results when handling datasets with complex structures, and parameter selection for similarity matrix construction typically depends on extensive experimentation and personal experience. To address this issue, we propose a spectral clustering algorithm based on natural nearest neighbor similarity graphs (NSG-SC). Natural nearest neighbor is a novel nearest neighbor concept that can effectively avoid the disadvantage of manual parameter tuning required by K-nearest neighbor and -nearest neighbor methods. This algorithm conducts searches based on the intrinsic characteristics of the dataset itself when constructing the similarity matrix, thereby avoiding the effects of improper parameter selection and outliers while more accurately reflecting the structural relationships within the dataset. Experimental results demonstrate the feasibility and effectiveness of the proposed NSG-SC algorithm.

Full Text

Preamble

Spectral Clustering Based on Natural Nearest Neighbor Similarity Graph

Liu Youchao, Zhang Xihuang
(School of Internet of Things Engineering, Jiangnan University, Wuxi, Jiangsu 214122, China)

Abstract: Spectral clustering is a clustering algorithm grounded in spectral graph partitioning theory that has gained widespread popularity due to its superior performance on non-convex datasets. However, traditional spectral clustering algorithms often produce unsatisfactory results when processing structurally

complex datasets, and parameter selection for similarity matrix construction typically relies on extensive experimentation and personal experience. To address these limitations, this paper proposes a spectral clustering algorithm based on natural nearest neighbor similarity graphs (NSG-SC). Natural nearest neighbor is a novel nearest neighbor concept that effectively avoids the parameter-tuning drawbacks of K-nearest neighbor and -nearest neighbor methods. The algorithm constructs the similarity matrix by leveraging the intrinsic characteristics of the dataset itself, thereby avoiding the adverse effects of improper parameter selection and outliers while more faithfully reflecting the structural relationships within the data. Experimental results demonstrate the feasibility and effectiveness of the proposed NSG-SC algorithm.

Keywords: spectral clustering; natural nearest neighbor; similarity graph; affinity matrix

0 Introduction

Clustering analysis represents a crucial branch of machine learning and an effective means for understanding and exploring inherent relationships within data. As an important unsupervised learning method, clustering aims to partition data into multiple disjoint clusters according to specific criteria, such that intra-cluster data exhibit high similarity while inter-cluster data show low similarity [1]. Numerous clustering algorithms have been proposed to date, including partition-based methods such as K-means [2] and K-medoids [3]; density-based methods such as DBSCAN (density-based spatial clustering of applications with noise) [4] and OPTICS (ordering points to identify the clustering structure) [5]; grid-based methods such as STING (statistical information grid) [6]; hierarchical methods such as ROCK (A hierarchical clustering algorithm for categorical attributes) [7] and CURE (clustering using representatives) [8]; model-based approaches; and graph-theoretic spectral clustering methods.

Spectral clustering algorithms are built upon spectral graph theory and essentially transform the clustering problem into an optimal graph partitioning problem through spectral relaxation. Compared with traditional clustering algorithms, spectral clustering can perform clustering on sample spaces of arbitrary shapes and converge to global optimal solutions. Consequently, spectral clustering has been widely applied in bioinformatics [9], pattern recognition [10], image segmentation [11], and text mining [12]. Classic spectral clustering algorithms include the k-way partitioning NJW algorithm proposed by Ng et al. [13] and the self-tuning spectral clustering algorithm proposed by Zelnik-Manor [14]. Current research on spectral clustering primarily focuses on similarity matrix construction, eigenvector selection, automatic determination of cluster numbers, Laplacian matrix selection, and application to massive datasets [15].

Among these research directions, similarity matrix construction is paramount because it directly influences eigenvector acquisition and ultimately affects clus-

tering results. Research on similarity matrix construction in spectral clustering has been ongoing. In 2011, Yang et al. proposed a density-sensitive distance metric [16] that defines an adjustable-length segment adapting to different density regions—shortening in high-density areas and lengthening in low-density areas. Although this algorithm can handle multi-scale clustering problems and is relatively insensitive to parameter selection, it suffers from unstable clustering performance and suboptimal results on real datasets. In 2012, Li et al. proposed a new similarity matrix construction method based on neighbor propagation [17], which increases similarity between point pairs within the same cluster to better detect data structures. In 2013, Blekas et al. proposed a spectral clustering algorithm based on Newton' s equations of motion [18], establishing a potential orbital analysis interaction model and using Newton' s preprocessing method to obtain valuable similarity information, thereby enriching the similarity matrix. In 2015, Inkaya et al. proposed an adaptive similarity graph construction algorithm based on density and connectivity [19], which constructs a similarity graph from the dataset and then derives the similarity matrix, resulting in a new spectral clustering algorithm [20]. While this algorithm can find local characteristics of clusters with arbitrary shapes and variable densities and demonstrates stable performance, it exhibits weak noise handling capability and imprecise proximity relationships between data points in mixed clustering scenarios.

Natural nearest neighbor [21] is a novel nearest neighbor concept belonging to the scale-free nearest neighbor method category. This approach searches based on the dataset' s intrinsic characteristics, effectively avoiding the manual parameter-setting requirements of K-nearest neighbor and -nearest neighbor methods. This paper adopts the natural nearest neighbor concept to replace the K-nearest neighbor, -nearest neighbor, or fully connected methods traditionally used in spectral clustering algorithms for similarity matrix construction. By utilizing the natural nearest neighbor search algorithm to construct a parameter-free similarity graph and then applying a Gaussian kernel function to build the similarity matrix from this graph, we obtain a spectral clustering algorithm based on natural nearest neighbor similarity graphs (NSG-SC). Experiments demonstrate that the proposed NSG-SC algorithm achieves superior clustering performance.

1.1 Natural Nearest Neighbor

The concept of nearest neighbors was first proposed in 1951 and has since received extensive attention and research, finding widespread application in artificial intelligence, data mining, and pattern recognition. The two most widely used nearest neighbor concepts, proposed by Stenvens, are K-nearest neighbor and -nearest neighbor. K-nearest neighbor identifies the K objects with shortest distances around each object in the dataset, where parameter K requires manual setting. -nearest neighbor identifies objects within radius around each object, where parameter requires manual setting. Both approaches heavily rely on parameter settings rather than searching according to the dataset' s intrinsic

characteristics.

Natural nearest neighbor is a recently proposed novel concept that belongs to the scale-free nearest neighbor method category and requires no manual parameter setting—representing its primary distinction from K-nearest neighbor and -nearest neighbor. Inspired by human social relationships, this method can effectively determine neighborhoods in datasets without given parameters, dynamically selecting different numbers of nearest neighbors for each data point based on the dataset' s attributes.

The fundamental principle of natural nearest neighbor is based on density partitioning: data points in high-density regions naturally have more nearest neighbors, while those in low-density regions have fewer. Outlier points in the dataset have only a few or no nearest neighbors at all. Because noise and anomalous points lack nearest neighbors, normal points will not consider them as neighbors.

Definition 1 (r-Neighborhood). The r-neighborhood is defined as:

$$r_i(x) = \{(x_i, x_j) \mid x_j \in \text{findKNN}(x_i, r)\}$$

where $\text{findKNN}(x_i, r)$ is a search function returning the r-th nearest neighbor of x_i, and X represents a subset of the original dataset.

Definition 2 (Natural Nearest Neighbor). Based on the r-neighborhood, if point x_i is in the r-neighborhood of point x_j and point x_j is also in the r-neighborhood of point x_i, then x_i and x_j are natural nearest neighbors:

$$x_i \in NN_r(x_j) \iff x_i \in KNN_r(x_j) \wedge x_j \in KNN_r(x_i)$$

Definition 3 (Stable Search State). The natural nearest neighbor algorithm reaches a stable search state if and only if:

$$\forall x_i, x_j \in X, x_i \neq x_j \rightarrow x_i \in KNN_r(x_j) \wedge x_j \in KNN_r(x_i)$$

where r is the search round number. If formula (3) is satisfied, the current round of search is stable and does not require early termination.

Algorithm 1: Natural Nearest Neighbor Search Algorithm

```
Input: dataset X
r = 1, flag = 0, NaNEdge =  , NaNNum(x_i) = 0 for all x_i   X
while flag = 0 do
    for all x_i   X do
        knn(x_i) = findKNN(x_i, r)
        KNN_r(x_i) = KNN_{r-1}(x_i)   knn(x_i)
        if x_i   KNN_r(knn(x_i)) then
            NaNEdge = NaNEdge   {(x_i, knn(x_i))}
            NaNNum(x_i) = NaNNum(x_i) + 1
            NaNNum(knn(x_i)) = NaNNum(knn(x_i)) + 1
        end if
```

```
    end for
    cnt = count{x_i | NaNNum(x_i) = 0}
    if all NaNNum(x_i) $\neq$ 0 for all x_i then
        flag = 1
    end if
    r = r + 1
end while
Output: NaNEdge
```

1.2 NJW Spectral Clustering

The NJW spectral clustering algorithm is a classic multi-way partitioning algorithm proposed by Ng et al. [13]. It constructs the similarity matrix using a fully connected method based on the Gaussian kernel function. The basic steps are as follows:

Input: Initial dataset $X = \{x_1, x_2, ..., x_n\}$
Output: Clustering results $\{C_1, C_2, ..., C_k\}$

a) Construct similarity matrix A, defined as:

$$A_{ij} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}$$

where σ is a scaling parameter requiring manual setting.

b) Compute the degree matrix D and Laplacian matrix L using A and D, where:

$$D_{ij} = \begin{cases} \sum_{j=1}^{n} A_{ij} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

$$L = D^{-1/2} A D^{-1/2}$$

c) Compute the top k eigenvectors $z_1, z_2, ..., z_k$ of L, construct matrix $Z \in \mathbb{R}^{n \times k}$, and normalize it to obtain matrix Y:

$$Z = [z_1, z_2, ..., z_k]$$

$$Y_{ij} = \frac{Z_{ij}}{\sqrt{\sum_{j=1}^{k} Z_{ij}^2}}$$

d) Treat each row of Y as a point in $\mathbb{R}^k$ and cluster them using K-means to obtain the final clustering results.

2 NSG-SC Algorithm

2.1 Algorithm Idea

The proposed algorithm first constructs the natural nearest neighbor relationship set for the dataset using Algorithm 1, obtaining the edge set $NaNEdge$. Each data point in the original dataset is then treated as a vertex to form the vertex set V. Using $NaNEdge$ as the edge relationship set and V as the vertex set, we construct an undirected weighted graph called the Natural Nearest Neighbor Similarity Graph (NSG), defined as:

$$NSG = (V, NaNEdge)$$

The graph NSG consists of many connected components, where each component represents a potential cluster. Any two points within the same connected component are connected by one or more edges either directly or indirectly. We determine the number of connected components g in NSG and compare it with the target number of clusters k. If $g > k$, too many independent clusters exist. In this case, we insert a new edge (v_i, v_j) between components, defined as:

$$(v_i, v_j) = \arg\min\{d(v_i, v_j) \mid v_i \in C_p, v_j \in C_q, p \neq q\}$$

where C_p and C_q represent connected components. After insertion, we update the connected component count g and similarity graph NSG, repeatedly comparing and merging components until g equals k.

[Figure 1: see original paper] illustrates this process, where red edges represent existing edges before merging and blue edges represent newly added edges after merging.

2.2 Algorithm Description

Input: Initial dataset $X = \{x_1, x_2, ..., x_n\}$
Output: Clustering results $\{C_1, C_2, ..., C_k\}$

Step 1: Obtain the relationship set $NaNEdge$ using Algorithm 1.

Step 2: Construct the undirected weighted graph $NSG(V, NaNEdge)$ according to equation (9).

Step 3: Determine the connected components of NSG and their count g, then merge components until the count reduces to k.

Step 4: Construct the similarity matrix A using equation (11):

$$A_{ij} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|^2}{2h^2}\right) & \text{if } (i,j) \in NaNEdge \\ 0 & \text{otherwise} \end{cases}$$

where $h = \max\{d(i,h) \mid (i,h) \in NaNEdge\}$ represents the longest edge in the same connected component.

Step 5: Perform steps 2, 3, and 4 of the NJW spectral clustering algorithm from Section 1.2 to obtain the final clustering results.

2.3 Algorithm Time Complexity Analysis

Let n be the number of samples in the original dataset. According to Algorithm 1, the time complexity of the natural nearest neighbor search phase is determined by: (a) creating a k-d tree for storing the dataset, with time complexity $O(n \log n)$; (b) performing one round of natural nearest neighbor search with complexity $O(n \log n)$. With λ search rounds conducted, the total search complexity is $O(\lambda n \log n)$. Typically, $2 \leq \lambda \leq n$, and for high-dimensional or irregular datasets, $20 \leq \lambda \leq 30$.

According to the NSG-SC algorithm steps in Section 2.1, the time complexity of remaining steps is determined by: (1) constructing the similarity matrix, with complexity $O(n^2)$; (2) the K-means step, with complexity $O(nkt)$, where t is the number of iterations (generally not exceeding 300).

In summary, for large n, the time complexity of the NSG-SC algorithm is $O(n^2)$, which is the same as general spectral clustering algorithms.

Although the NSG-SC algorithm does not require parameters during the similarity graph construction process, it still needs manual setting of the target cluster number k in subsequent steps. Determining the target cluster number k is actually a universal problem for most clustering algorithms, for which various more or less successful methods have been proposed. The simplest and most common approach is to visualize the data and directly observe the appropriate number of clusters, though this is often ineffective. In model-based clustering, effective criteria for selecting k from data typically exist, usually based on data log-likelihood, which can then be processed using frequentist or Bayesian methods [22]. With few or no assumptions about the underlying model, various metrics are generally used to select k, such as ad-hoc measures like the ratio of intra-cluster to inter-cluster similarity, information-theoretic criteria [23], and gap statistics.

3 Experiments

3.1 Related Algorithms and Parameter Settings

The proposed algorithm is compared with K-means, NJW spectral clustering [13] (hereinafter NJW), Self-Tuning spectral clustering [14] (hereinafter ST-SC), and the algorithm proposed in [20] (hereinafter DAN). Parameter settings are as follows: K-means uses target cluster number k; NJW uses scaling parameter sigma (empirically set to 0.005 in these experiments) and target cluster number k; ST-SC uses parameter K (number of nearest neighbors per point, set to the authors' recommended value of 7) and target cluster number k; DAN uses target cluster number k; the proposed NSG-SC algorithm uses target cluster number k.

3.2 Artificial Dataset Experiments and Analysis

To validate the effectiveness of NSG-SC, we first compare it with K-means, NJW, and ST-SC on four synthetic datasets shown in [Figure 2: see original paper]: ChainLink, Sticks, ThreeCircles, and UnbalanceSpiral. Dataset details are provided in . The final clustering results are shown in [Figure 2: see original paper] through [Figure 5: see original paper], with K-means in the upper left, NJW in the upper right, ST-SC in the lower left, and NSG-SC in the lower right.

TABLE:1 Four Types of Artificial Datasets

Dataset	Instances	Dimensions
ChainLink	-	-
Sticks	-	-
ThreeCircles	-	-
UnbalanceSpiral	-	-

Comparative analysis reveals that NSG-SC correctly clusters all four datasets. On ChainLink, ST-SC achieves correct clustering while K-means and NJW fail to correctly cluster the two adjacent manifold clusters. On Sticks and UnbalanceSpiral, K-means, NJW, and ST-SC exhibit varying degrees of incorrect clustering, while NSG-SC correctly handles these cases. On ThreeCircles, ST-SC achieves correct clustering, whereas K-means and NJW produce completely incorrect results. These experiments demonstrate that when inter-cluster similarity is high, the similarity matrices constructed by NJW and ST-SC cannot accurately reflect dataset structure, leading to misclassification. In contrast, the similarity matrix built from the natural nearest neighbor similarity graph enables NSG-SC to correctly capture the true data structure even for complex datasets with high inter-cluster similarity, yielding superior clustering results.

3.3 Evaluation Metrics

For real dataset experiments, we employ ARI [24] (adjusted rand index) and AMI [25] (adjusted mutual information) to evaluate K-means, NJW, ST-SC, DAN, and NSG-SC. The Rand Index (RI) measures similarity between two clusterings from a statistical perspective, mathematically related to accuracy with range [0,1]. ARI further ensures the metric approaches zero for random clustering results, with range [-1,1] where larger values indicate better alignment with ground truth. Broadly, ARI measures the agreement between two data distributions.

Mutual Information (MI) measures useful information in information theory, quantifying the information one random variable contains about another or the uncertainty reduction. AMI improves upon MI for clustering applications, similar to how ARI corrects RI, and is closely related to information variation.

Like ARI, AMI ranges from [-1,1], with larger values indicating better clustering quality.

3.4 Real Dataset Experiments and Analysis

To evaluate practical significance, we conduct experiments on four UCI datasets: Iris, Wine, Vehicle, and Landsat. Dataset details are shown in .

TABLE:2 Four Types of UCI Data

Dataset	Instances	Dimensions
Iris	-	-
Wine	-	-
Vehicle	-	-
Landsat	-	-

ARI and AMI evaluation results are presented in and .

TABLE:3 ARI Index Comparison Across Algorithms

Algorithm	Iris	Wine	Vehicle	Landsat
K-means	-	-	-	-
NJW	-	-	-	-
ST-SC	-	-	-	-
DAN	-	-	-	-
NSG-SC	-	-	-	-

TABLE:4 AMI Index Comparison Across Algorithms

Algorithm	Iris	Wine	Vehicle	Landsat
K-means	-	-	-	-
NJW	-	-	-	-
ST-SC	-	-	-	-
DAN	-	-	-	-
NSG-SC	-	-	-	-

Analysis of and shows that on Iris, both ST-SC and NSG-SC perform well, with NSG-SC slightly outperforming ST-SC. On Wine, ST-SC and NSG-SC again show strong performance, with NSG-SC achieving higher ARI but lower AMI than ST-SC. On Vehicle, all algorithms perform poorly, though NSG-SC' s metrics are significantly higher than others. On Landsat, DAN and NSG-SC perform well, with NSG-SC showing lower ARI but higher AMI than DAN. Overall, NSG-SC demonstrates superior performance on real datasets, accurately reflecting dataset structural relationships to achieve better clustering results.

4 Conclusion

This paper proposes a spectral clustering algorithm based on natural nearest neighbor similarity graphs. Leveraging the advantages of natural nearest neighbor relationships—parameter-free operation, search based on dataset intrinsic characteristics, and robustness to outliers—the algorithm precisely determines each sample' s neighborhood to construct a similarity graph. This yields a similarity matrix that more faithfully reflects sample similarity relationships than traditional K-nearest neighbor, -nearest neighbor, or fully connected methods used in conventional spectral clustering. Experiments on both artificial and UCI real datasets demonstrate that NSG-SC better captures dataset structural relationships, particularly for structurally complex data, thereby achieving superior clustering results. Future research may consider introducing fuzzy nearest neighbor relationships for handling mixed clusters, incorporating pairwise constraint information to optimize clustering performance, or combining with heuristic algorithms.

References

[1] Xu Rui, Wunsch D. Survey of clustering algorithms [J]. IEEE Trans on Neural Networks, 2005, 16 (3): 645-678.

[2] Jain A K. Data clustering: 50 years beyond K-means [J]. Pattern Recognition Letters, 2010, 31 (8): 651-666.

[3] Park H S, Jun C H. A simple and fast algorithm for K-medoids clustering [J]. Expert Systems with Applications, 2009, 36 (2): 3336-3341.

[4] Yang Jing, Gao Jiawei, Liang Jiye, et al. An improved DBSCAN clustering algorithm based on data field [J]. Journal of Frontiers of Computer Science and Technology, 2012, 6 (10): 903-911.

[5] Kalita H K, Bhattacharyya D K, Kar A. A new algorithm for ordering of points to identify clustering Structure based on perimeter of triangle: OPTICS (BOPT) [C]// Proc of the 15th International Conference on Advanced Computing and Communications. Piscataway, NJ: IEEE Press, 2007: 523-528.

[6] Ansari S, Chetlur S, Prabhu S, et al. An overview if clustering analysis techniques used in data mining [J]. International Journal of Emerging Technology and Advanced Engineering, 2013, 3 (12): 284-246.

[7] Agarwal P, Alam M A, Biswas R. A hierarchical clustering algorithm for categorical attributes [C]// Proc of the 2nd International Conference on Computer Engineering and Applications. Piscataway, NJ: IEEE Press, 2010: 112-119.

[8] Zhou Yajian, Xu chen, Li Jiguo. Unsupervised anomaly detection method based on improved CURE clustering algorithm [J]. Journal on Communications, 2010, 31 (7): 18-23.

[9] Higham D J, Kalna G, Kibble M. Spectral clustering and its use in bioinformatics [J]. Journal of Computational and Applied Mathematics, 2007, 204 (1): 25-37.

[10] Chih-Hsuan W. Recognition of semiconductor defect patterns using spatial filtering and spectral clustering [J]. Expert Systems with Applications, 2008, 34 (3): 1914-1923.

[11] Shan Zeng, Rui Huang, Zhen Kang, et al. Image segmentation using spectral clustering of Gaussian mixture models [J]. Neurocomputing, 2014, 144: 350-362.

[12] He Ruifang, Qin Bing, Liu Ting. A novel approach to update summarization using evolutionary manifold-ranking and spectral clustering [J]. Expert Systems with Applications, 2012, 39 (3): 2375-2384.

[13] Ng A Y, Jordan M I, Weiss Y. On spectral clustering: analysis and an algorithm [C]// Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2001: 849-856.

[14] Zelnik-Manor L, Perona P. Self-tuning spectral clustering [C]// Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2004, 17: 1601-1608.

[15] Jia Hongjie, Ding Shifei, Xu Xinzheng, et al. The latest research progress on spectral clustering [J]. Neural Computing and Applications, 2014, 24: 1477-1486.

[16] Yang Peng, Zhu Qingsheng, Huang Biao. Spectral clustering with density sensitive similarity function [J]. Knowledge-Based Systems, 2011, 24 (5): 621-628.

[17] Li Xinye, Guo Lijie. Constructing affinity matrix in spectral clustering based on neighbor propagation [J]. Neurocomputing, 2012, 97: 125-130.

[18] Blekas K, Lagaris I E. A spectral clustering approach based on Newton' s equations of motion [J]. International Journal of Intelligent Systems, 2013, 28 (4): 394-410.

[19] Inkaya T, Kayaligil S, Evin Ozdemirel N. An adaptive neighborhood construction algorithm based on density and connectivity [J]. Pattern Recognition Letters, 2015, 52: 17-24.

[20] Inkaya T. A parameter-free similarity graph for spectral clustering [J]. Expert Systems with Applications, 2015, 42: 9489-9498.

[21] Zhu Qingsheng, Feng Ji, Huang Jinlong. Natural neighbor: a self-adaptive neighborhood method without parameter K [J]. Pattern Recognition Letters, 2016, 80: 30-36.

[22] Fraley C, Raftery A E. Model-based clustering, discriminant analysis, and density estimation. [J] Journal of the American Statistical Association, 2002, 97: 611-631.

[23] Still S, Bialek W. How many clusters? An information-theoretic perspective. [J] Neural Computation, 2004, 16 (12): 2483-2506.

[24] Santos J M, Embrechts M. On the use of the adjusted rand index as a metric for evaluating supervised classification [C]// Proc of International Conference on Artificial Neural Networks. Berlin: Springer, 2009: 175-184.

[25] Cai Deng, He Xiaofei, Han Jiawei. Document clustering using locality preserving indexing [J]. IEEE Trans on Knowledge and Data Engineering, 2005, 17 (12): 1624-1637.

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