Improvement on Incremental Spectral Clustering

Nil Ayday[1], Debarghya Ghoshdastidar[1]

[1] *Technical University of Munich, Germany*

Abstract

Spectral clustering (SC) is one of the most popular algorithms for finding communities in a static graph. The algorithm derives the communities through clustering the leading eigenvectors of the graph Laplacian. Many practical problems involve networks that evolve over time (for instance, the Facebook friendship network) and communities need to be updated dynamically. Since spectral clustering is an offline algorithm, any change in the graph requires a new computation of the eigenvectors of the graph Laplacian matrix, which is highly inefficient for real-time clustering of most large dynamic networks.

Incremental spectral clustering (ISC) extends the method to efficiently cluster dynamic graphs [1], by incrementally updating the eigenvectors and eigenvalues, but can sometimes lead to sub-optimal performance due to approximation errors. In this paper, a modified incremental spectral algorithm (MISC) is introduced by considering higher-order approximations of the eigenvector update rule. We compared the performance of ISC and MISC through experiments on random graphs generated from dynamic stochastic block model, from Barabasi-Albert model with planted communities and real-word network data.

Keywords

spectral clustering, dynamic network, incremental spectral clustering, random graph models

1. Introduction

Spectral clustering (SC) is an algorithm for finding communities and has been the center of attention for a few years, due to its simplicity in implementation [2] and its application in many areas such as image segmentation [3], bioinformatic [4], computer vision and VLSI design [5]. The idea of spectral clustering relies on standard linear algebra methods and studies the spectrum of the Adjacency and the Laplacian matrices to find communities. When the Laplacian matrix is used to represent the static graph, one can see that the leading eigenvectors relate well to the embedding of the nodes, and if we cluster these embeddings using algorithms such as K-means we can find the communities.

Real-world networks often change over time [6]. For example, social networks like Facebook or LinkedIn evolve every millisecond. In such cases, spectral clustering fails to efficiently cluster these large dynamic networks in real-time, as the eigenvectors of the graph Laplacian matrix have to be computed for every change on the network.

In dynamic networks [7], wherein the graphs change over time, incremental spectral clustering (ISC) [8] develop upon the methods of standard spectral clustering by tracking the changes

LWDA'21: Lernen, Wissen, Daten, Analysen September 01–03, 2021, Munich, Germany

nil.ayday@tum.de (N. Ayday); ghoshdas@in.tum.de (D. Ghoshdastidar)
https://www.in.tum.de/tfai/people/debarghya-ghoshdastidar/ (D. Ghoshdastidar)
0000-0003-0202-7007 (D. Ghoshdastidar)

in the eigenvectors of the matrix and generating cluster labels, as the graph evolves. This is more efficient than computing the eigenvectors and clusters from scratch after each iteration. However, the incremental update provides only an approximation of the actual eigenvectors. Since the error accumulates over time the loss of information in the approximation causes the algorithm to perform more poorly than SC as the number of changes on the graph increases.

In this paper, we propose a modified version of the existing ISC algorithm (Section 2.2). The proposed modified ISC (MISC) algorithm is initialized by the standard incremental update of ISC and then the eigenvalue-eigenvector pair are iteratively improved. This improvement aims to have an algorithm that still performs more efficiently than spectral clustering but is also more accurate than incremental spectral clustering. The comparison between MISC and ISC according to their accuracy has been shown in Section 3 through experiments on graphs produced by the Barabasi-Albert model with planted communities, dynamic stochastic block model and real-word network data.

2. Background and Methods

Consider a network represented as an unweighted undirected graph $G = (E, V)$ with node set V, edge set E and adjacency matrix $A \in \{0, 1\}^{n \times n}$, where $a_{ji} = a_{ij} = 1$ means that the vertices v_i and v_j are connected by an edge. The total number of edges connected to the vertex i (v_i) is the degree of the vertex i ($degree(i)$) and denoted by the degree matrix $D \in \mathbb{R}^{n \times n}$, which is a diagonal matrix with $D_{ii} = degree(i)$. The graph Laplacian matrices, such as the unnormalized Laplacian, $L = D - A$, and the random walk Laplacian, $L_{rw} = D^{-1}L$, capture the intrinsic of geometry of a graph. For instance, the leading eigenvalues of the Laplacians are closely related to the optimal ratio or normalised cuts [9].

2.1. Spectral Clustering (SC) and Incremental Spectral Clustering (ISC)

The goal of spectral clustering (SC) is to divide the nodes of a network into several communities such that the nodes in the same community have many edges with each other and nodes in different communities have few. More formally, SC solves a relaxation of cut minimisation problems [2] by clustering the leading eigenvectors of the graph Laplacians. In this paper, we focus on SC and its variants based on the random walk Laplacian L_{rw} (see [2]). The subsequent mathematical derivations and methods can be generalised to other graph Laplacians. SC based on the random walk Laplacian consider the solution (eigenvalue λ and eigenvector q) of the generalized eigenvalue system

$$Lq = \lambda Dq. \tag{1}$$

To extend SC to dynamic graphs, one considers the following equation, where ΔL and ΔD denote the changes in Laplacian and degree matrices [1] and $(\Delta\lambda, \Delta q)$ denote the increments in eigenpair that one needs to compute,

$$(L + \Delta L)(q + \Delta q) = (\lambda + \Delta\lambda)(D + \Delta D)(q + \Delta q). \tag{2}$$

[1] in the case of a new vertex the old matrices L and D are extended by a row and column of zeros to calculate ΔL and ΔD

By expanding the equation and using the fact that $Lq = \lambda Dq$, (2) can be written as:

$$L\Delta q+\Delta Lq+\Delta L\Delta q = \lambda D\Delta q+\lambda\Delta Dq+\lambda\Delta D\Delta q+\Delta\lambda Dq+\Delta\lambda D\Delta q+\Delta\lambda\Delta Dq+\Delta\lambda\Delta D\Delta q \tag{3}$$

In ISC [8], the authors propose to approximate the increments of the eigenvalues and eigenvectors $(\tilde{\Delta}\lambda, \tilde{\Delta}q)$ by removing the higher-order terms $\Delta L\Delta q, \lambda\Delta D\Delta q, \Delta\lambda D\Delta q, \Delta\lambda\Delta Dq, \Delta\lambda\Delta D\Delta q$:

$$L\tilde{\Delta}q + \Delta Lq = \lambda D\tilde{\Delta}q + \lambda\Delta Dq + \tilde{\Delta}\lambda Dq \tag{4}$$

One can solve (4) for $\tilde{\Delta}\lambda$ by multiplying both sides with q^T and eliminating $q^T L\tilde{\Delta}q = \lambda q^T D\tilde{\Delta}q$:

$$\tilde{\Delta}\lambda = \frac{q^T\Delta Lq - \lambda q^T\Delta Dq}{q^T Dq} \tag{5}$$

After calculating $\tilde{\Delta}\lambda$ with (5), one can reformulate (4) in order to obtain an update rule for $\tilde{\Delta}q$ as

$$\tilde{\Delta}q = (K^T K)^{\dagger} K^T(\lambda\Delta D + \tilde{\Delta}\lambda D - \Delta L)q \ , \tag{6}$$

where $K = L - \lambda D$ and $(K^T K)^{\dagger}$ denotes the pseudoinverse of $K^T K$ since the matrix is singular.[2] In [8], the rule in (6) is used to update the leading eigenvectors and the updated eigenvectors are subsequently clustered using K-means to obtain the new clusters in the graph.

2.2. Proposed Modified Incremental Spectral Clustering (MISC) Method

The increments of the eigenvectors in (6) are approximations of the true increments of eigenvectors in (2). The accuracy of the calculated eigenvectors degrade as the number of changes on the graph increases due to the accumulated error resulting from the removal of the higher-order terms. We propose a new method based on the ISC that aims to minimize the number of approximations needed. With some manipulations, one can write (2) as

$$(L + \Delta L - \lambda D - \lambda\Delta D - \hat{\Delta}\lambda D - \hat{\Delta}\lambda\Delta D)\hat{\Delta}q = (-\Delta L + \lambda\Delta D + \hat{\Delta}\lambda D + \hat{\Delta}\lambda\Delta D)q. \tag{7}$$

Note that the higher-order terms are still part of (7). From (7), we can solve for $\hat{\Delta}\lambda$ as

$$\hat{\Delta}\lambda = \frac{q^T\Delta Lq + q^T\Delta L\hat{\Delta}q - \lambda q^T\Delta Dq - \lambda q^T\Delta D\hat{\Delta}q}{q^T Dq + q^T D\hat{\Delta}q + q^T\Delta Dq + q^T\Delta D\hat{\Delta}q} \ . \tag{8}$$

(8) cannot be solved independently from $\hat{\Delta}q$ unlike (5). Therefore, we need an additional initialization step for $\hat{\Delta}\lambda$ and $\hat{\Delta}q$. The main intuition for the proposed modified ISC (MISC) algorithm is to start with the initialisation of $\hat{\Delta}\lambda$ from (5), similar to ISC, and update $(\hat{\Delta}\lambda, \hat{\Delta}q)$ through repeated iterations of (7) and (8). The steps of MISC algorithm is listed below in Algorithm 1.

Note that the proposed method aims to increase the quality of the approximation by using the previous values of $\hat{\Delta}\lambda$ and $\hat{\Delta}q$ in each iteration to keep the equation (2) intact, whereas in ISC all the higher order terms are ignored.

[2] In this paper numpy.linalg.pinv function with the condition number 0.001 is used to calculate the pseudoinverse (https://numpy.org/doc/stable/reference/generated/numpy.linalg.pinv.html)

Algorithm 1 Modified Incremental Spectral Clustering (MISC)

```
Given the initial graph, run SC with $L_{rw}$ [2] to obtain initial clusters. This also provides initial eigenvalues and eigenvectors $(\lambda, q)$ of the generalized eigenvalue system (1)
for every change in the graph do
    Use (5) to calculate $\hat{\Delta}\lambda$ and compute $\hat{\Delta}q$ without higher-order approximation using (7)
    for $k = 1, 2, \ldots, T$ do
        Recompute $\hat{\Delta}\lambda, \hat{\Delta}q$ using (7) and (8)
    end for
    Update the solution of eigensystem from $(\lambda, q)$ to $(\lambda + \hat{\Delta}\lambda, q + \hat{\Delta}q)$
    Cluster the nodes according to the updated eigenvectors using the K-means cluster centers from the previous iteration
    Update the K-means cluster centers according to the new labels
end for
```

3. Experimental Evaluation

In this section, we compare the performance of the proposed MISC algorithm with ISC [8]. Recall that the main objective of the incremental algorithms is to obtain efficient approximation of SC [2]. Hence, the basis for our empirical evaluation is to compare the alignment of the clusters from MISC and ISC with those obtained from SC. For small graphs, one may visually compare the obtained clusters (see Table 1), but for large graphs, it is more useful to use the adjusted Rand index (ARI) metric [10] to compute the alignment of MISC (or ISC) with SC. The ARI between any two clustering is a value between $[-1, 1]$ with 1 indicating that the two clustering are identical, and smaller values indicating less alignment.

The experiments are carried on random graph models with planted communities. Planted communities refer to the communities that are formed intentionally while generating a random graph, in order to analyse the performance of graph partitioning algorithms. A natural model for generating random undirected graphs with community structure is the stochastic block model (SBM) which is for static graphs. In order to construct SBM, one adds an edge (i, j) with probability p for every pair v_i , v_j in same community and an edge (i, j) with probability $q < p$ for every pair v_i , v_j from different communities. Since we need a dynamic graph model, we assign the new coming nodes randomly to one of the communities and then add edges according to the described rule.

Another random graph model we consider is the Barabasi-Albert model with planted communities (BA-planted) [11]. BA is the most popular model for preferential attachment, which means that the new nodes in a network connect more with high degree nodes. The principle of rich get richer, models natural networks more realistic. This behaviour is ensured by choosing the probability of having an edge between the new node (added at time t) and a node $v_u \in V_{t-1}$, proportional to d_u. Using this probability one adds edges between the new arriving node and m random nodes. BA-planted combines the community structure of SBM and preferential attachment process of BA model by initializing the graph with static SBM and adding nodes according to the principle of BA model.

In SBM, we chose $p = 0.7$ and $q = 0.3$. In BA-planted we chose $p = 0.9$, $q = 0.1$ and $m = 5$. In every time step a new node is added to the graph by adding edges between this new node and the existing nodes. For every added edge the spectral clustering algorithms determined the clusters without re-initialization.

Table 1 demonstrates how well each spectral clustering algorithm performs over time on BA-planted. We can clearly see that the MISC finds the same meaningful clusters as SC, while ISC delivers a poor result.

Figure 1a shows the ARI scores of ISC and MISC for graphs produced by BA-planted which starts with 100 nodes. The average of 100 runs is plotted with the $\%95$ confidence interval. The x-axis represents the percentage of added nodes with respect to n_0. We can see a slight decrease in the ARI score as the graph evolves. It should also be noted that the MICS achieves better performance compared to ISC, as expected. Figure 1b demonstrates the respective time taken per similarity change over the number of nodes and implicates that the both incremental methods are faster than the SC.

Figure 1g shows the average of the results on the SBM for 100 runs. Unsurprisingly, the earlier performances of both ISC and MISC are better than the latter, and the ARI score of the ISC stays below the score of the MISC. The corresponding runtime of the spectral clustering algorithms can be seen in Figure 1h.

If one examines the graphs Figure 1a and Figure 1g, one can see that the gap between the ARI values of MISC and ISC increases as the graph gets larger. Hence it is expected that MISC to outperform ISC on a larger dataset.

The experiments carried out on real networks [12, 13], emphasizes the validity of previously presented results. Both in Figure 1e and Figure 1f the ARI scores of the ISC is under the ARI scores of the MISC.

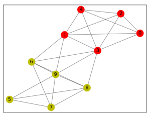
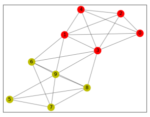
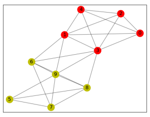
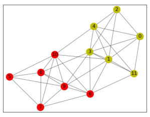
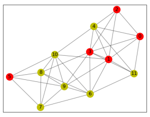
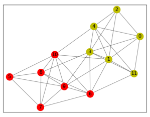
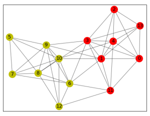
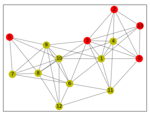
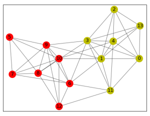
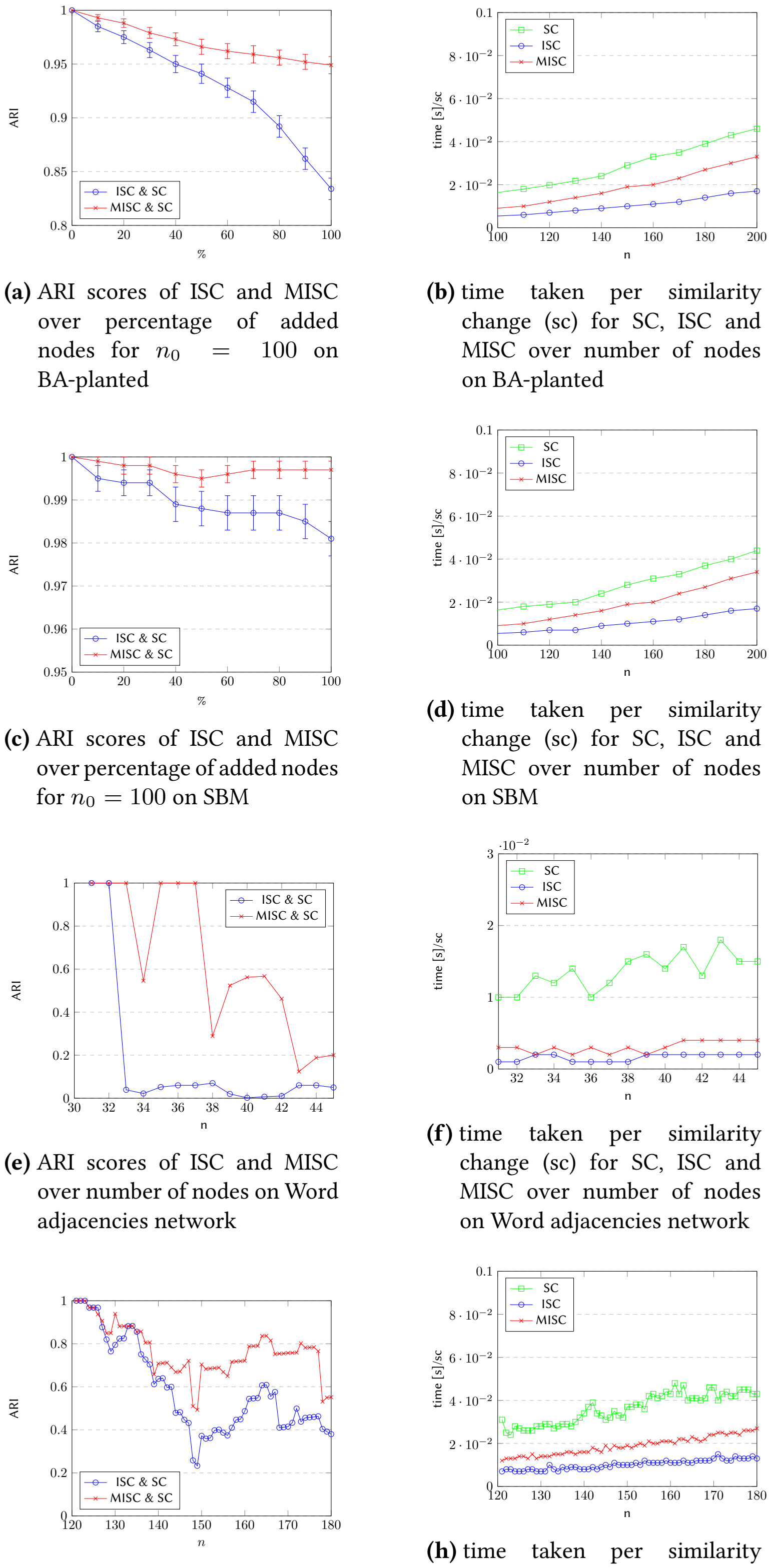
	SC	ISC	MISC
$t = 0$			
$t = 2$			
$t = 4$			

Table 1: SC, ISC and MISC over time



(a) ARI scores of ISC and MISC over percentage of added nodes for $n_0 = 100$ on BA-planted

(b) time taken per similarity change (sc) for SC, ISC and MISC over number of nodes on BA-planted

(c) ARI scores of ISC and MISC over percentage of added nodes for $n_0 = 100$ on SBM

(d) time taken per similarity change (sc) for SC, ISC and MISC over number of nodes on SBM

(e) ARI scores of ISC and MISC over number of nodes on Word adjacencies network

(f) time taken per similarity change (sc) for SC, ISC and MISC over number of nodes on Word adjacencies network

(g) ARI scores of ISC and MISC over number of nodes on Neural network

(h) time taken per similarity change (sc) for SC, ISC and MISC over number of nodes on Neural network

Figure 1: Performance analysis of the incremental spectral clustering methods on evolving graphs. Left column shows the ARI scores of ISC and MISC, right column shows time taken per similarity change (sc) for SC, ISC and MISC. Rows correspond respectively BA-planted, SBM, Word adjacencies network and Neural network.

4. Conclusion

In this short paper a Modified Incremental Spectral Clustering (MISC) method is proposed, which can be used for clustering the networks that evolve over time. MISC improves over existing Incremental Spectral Clustering [8] by adopting an iterative approach that takes into account higher-order approximations of the generalised eigenvalue problem corresponding to spectral clustering. Experimental results over dynamic stochastic block model, planted Barabasi-Albert model and real-word networks illustrate the improved performance of the proposed method. Further studies on the advantages and generality of the MISC approach will be considered in future. In particular, we plan to derive and compare MISC based on different graph Laplacians [2] and consider not just vertex and edge additions but also deletions. We also aim to replicate the discussed results on larger graphs. Furthermore, note that the iterative approach in MISC makes the method slightly slower than ISC, and the tradeoff of accuracy against efficiency needs to be extensively evaluated. Alternatives methods for solving linear equations instead of the computation of the pseudoinverse may be taken into consideration to increase the efficiency. Moreover, a theoretical analysis of the incremental update rules is envisioned for random graph models, particularly identifying the causes for the poor performance of ISC and how this is overcome by the proposed MISC updates.

Acknowledgments

This work is supported by the Baden-Württemberg Stiftung through the project “Clustering large evolving networks” in the framework of Baden-Württemberg Eliteprogram for postdocs.

References

[1] J. Li, H. Dani, X. Hu, J. Tang, Y. Chang, H. Liu, Attributed network embedding for learning in a dynamic environment, Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (2018).

[2] U. von Luxburg, A tutorial on spectral clustering, Statistics and Computing 17 (2007).

[3] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000).

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

[5] A. Y. Ng, M. I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, Advances in neural information processing systems (2002) 849–856.

[6] Y. Xie, C. Li, B. Yu, C. Zhang, Z. Tang, A survey on dynamic network embedding, arXiv preprint arXiv:2006.08093 (2020).

[7] G. Rossetti, R. Cazabet, Community discovery in dynamic networks: a survey, ACM Computing Surveys (CSUR) 51 (2018) 19–20.

[8] H. Ning, W. Xu, Y. Chi, Y. Gong, T. Huang, Incremental spectral clustering with application to monitoring of evolving blog communities, Proceedings of the 2007 SIAM International Conference on Data Mining (2007).

[9] F. R. Chung, Lectures on spectral graph theory, CBMS Lectures, Fresno 6 (1996) 17–21.

[10] L. Hubert, P. Arabie, Comparing partitions, Journal of classification 2 (1985) 193–218.

[11] H. Bruce, S. Sankagiri, Community recovery in a preferential attachment graph, IEEE Transactions on Information Theory 65 (2019) 6853–6874.

[12] M. E. Newman, Finding community structure in networks using the eigenvectors of matrices, Physical review E 74 (2006) 036104.

[13] D. J. Watts, S. H. Strogatz, Neural network, Nature 393 (1986) 440–442.