On Constructing Affinity Matrix

Cong Mu

The Problem

Given a simple graph $G = (V, E)$ associated with a $n \times n$ adjacent matrix A which is

- Binary: For $i, j \in V$, $A_{ij} = 1$ if $(i, j) \in E$; 0 otherwise.
- Symmetric: $A = A^\top$.
- Hollow: $\text{diag}(A) = 0$.

The spectral graph clustering tries to partition the vertices into coherent clusters or communities by [1, 2, 6]

1. Embedding the graph into d-dimension Euclidean space, denoted by $n \times d$ matrix $\widehat{X}$, by singular value decomposition (SVD). For this step, we have two commonly used approaches

   - Adjacency Spectral Embedding (ASE). Decompose the adjacency matrix directly by $A = USU^\top$ and take $\widehat{X} = U_d S_d^{1/2}$ where S_d is the first d singular values and U_d is the associated singual vectors.
   - Laplacian Spectral Embedding (LSE). First calculate the normalized Laplacian matrix $L = D^{-1/2} A D^{-1/2}$ where D is the degree matrix. Then decompose $L = USU^\top$ and take $\widehat{X} = U_d S_d^{1/2}$ like ASE.

   Recent works [2, 5] have shown that ASE and LSE could discover different clustering truths, and neither would dominate the other. In particular, ASE could better capture the core-periphery structure while LSE could uncover the affinity structure.

2. Clustering $\widehat{X}$ by some claasical algorithms such as K-Means and Gaussian Mixture Model (GMM). Theoretical results [3, 4] suggest that GMM would be a better choice for both ASE and LSE.

Moreover, the powerful spectral clustering methods could be applied to a more general problem. Given a $n \times D$ data matrix X, if we could represent the original data by constructing a $n \times n$ affinity matrix A which could capture geometric relationships among multiple data points. Then we could apply spectral graph clustering to A and get the clustering results. This is appealing since classical clustering methods might not work in origial D-dimensional space [7], especially in high-dimensional setting (D is very large). While spectral clustering could project the data onto a low dimension space (usually $d \ll D$) and yield satisfying clusters.

$$X_{n \times D} \xrightarrow{\text{Construct}} A_{n \times n} \xrightarrow{\text{Transform}} T(A)_{n \times n} \xrightarrow{\text{Embed}} \widehat{X}_{n \times d} \xrightarrow{\text{Cluster}} \widehat{C}. \tag{1}$$

Here as suggested in [2, 5], we consider both ASE ($T(A) = A$) and LSE ($T(A) = D^{-1/2}AD^{-1/2}$). To choose the embedding dimension d, we consider profile likelihood method in [8]. As for clustering algorithms in last step, we use GMM and penalized likelihood via BIC in [9] to choose the number of clusters. So the problem is to find a best $n \times n$ affinity matrix A given a $n \times D$ data matrix X such that we could get the optimal clustering results. i.e. we want to find a tractable (in terms of computational complexity) constructing method T^* such that

$$T^* = \arg\max \mathrm{ARI}(C, \widehat{C}) \qquad \text{subject to some constraints,}$$

where $\widehat{C}$ is given in (1) and ARI (Adjusted Rand Index) is a well-known criteria that could measure the probability that two partitions of data points will agree for a randomly chosen pair of data points [2]. Note that here we assume to have knowledge on truth C so that we could compare the performance, while in practice C is neither known nor necessarily unique.

Related Work

Some traditional methods to construct affinity matrix for spectral clustering are introduced in [1], including ε-neighborhood graph, k-nearest neighbor graph and fully connected graph with similarity function like Gaussian similarity function. A general recommendation is to work with the k-nearest neighbor graph first since it is simple and will construct a sparse adjacency matrix. Experience also shows that it is less vulnerable to unsuitable choices of parameters than the other graphs [1]. However there are few theoretical justification or rules on the choice of these methods and their parameters.

In terms of practical consideration, there are some disscusion about computational complexity and outliers in [10]. One aspect that we need to pay attention to when we construct the affinity matrix is how to deal with outliers. Since an outlier is a point which has very low similarity with all other points (e.g., because it is far away from them). Thus it will produce a spurious eigenvalue very close to 1 with an eigenvector which approximates an indicator vector for the outlier. Which means k outliers in a data set will cause the k principal eigenvectors to be outliers, not clusters [10]. Therefore, it is strongly recommended that outliers be detected and removed before the decomposition. One easy way to do this is to remove all points for which $\sum_{j \neq i} S_{ij} < \epsilon$ for some ϵ which is small w.r.t. the average d_i. Here S_{ij} is the entry of the affinity matrix and d_i is the degree of node i.

An unsupervised approach to generating more robust affinity matrix via identifying and exploiting discriminative features was proposed in [11]. Instead of trusting all available features blindly for measuring pairwise similarities, they formulated a unified and generalised data similarity inference framework based on the unsupervised clustering random forest with some modifications: (1) Avoid less informative features by measuring between-sample proximity via discriminative feature subspaces; (2) Relax the Euclidean assumption for data similarity inference by following the information-theoretic definition of data similarity, which states that different similarities can be induced from a given sample pair if distinct propositions are taken or different questions are asked about data commonalities.

A modified k-nearest neighbor graph was proposed in [12]. To reduce the sensitivity to the choice of k in k-

nearest neighbor graph, they considered the union of the kNN graph and the minimum spanning tree of the negated similarity matrix (kNN-MST). By adding the MST to the kNN similarity graph, the sensitivity to the choice of k is reduced due to ensuring that the final similarity graph is connected.

References

[1] Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4), 395-416.

[2] Priebe, C. E., Park, Y., Vogelstein, J. T., Conroy, J. M., Lyzinskic, V., Tang, M., Athreya, A., Cape, J. & Bridgeford, E. (2018). On a 'Two Truths' Phenomenon in Spectral Graph Clustering. *arXiv preprint*, arXiv:1808.07801.

[3] Athreya, A., Priebe, C. E., Tang, M., Lyzinski, V., Marchette, D. J., & Sussman, D. L. (2016). A limit theorem for scaled eigenvectors of random dot product graphs. *Sankhya A*, 78(1), 1-18.

[4] Tang, M., & Priebe, C. E. (2018). Limit theorems for eigenvectors of the normalized laplacian for random graphs. *The Annals of Statistics*, 46(5), 2360-2415.

[5] Cape, J., Tang, M., & Priebe, C. E. (2018). On spectral embedding performance and elucidating network structure in stochastic block model graphs. *arXiv preprint*, arXiv:1808.04855.

[6] Ng, A., Jordan, M., and Weiss, Y. (2002). On Spectral Clustering: Analysis and An Algorithm. *Advances in Neural Information Processing Systems 14*, (pp. 849 - 856). MIT Press.

[7] Blum, A., Hopcroft, J., & Kannan R. (2015). Foundation of Data Science.

[8] Zhu, M., & Ghodsi, A. (2006). Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Computational Statistics & Data Analysis*, 51(2), 918-930.

[9] Fraley, C., & Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, 97(458), 611-631.

[10] Hennig, C., Meila, M., Murtagh, F., & Rocci, R. (Eds.). (2015). Handbook of cluster analysis. , CRC Press.

[11] Zhu, X., Change Loy, C., & Gong, S. (2014). Constructing robust affinity graphs for spectral clustering. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 1450-1457).

[12] Veenstra, P., Cooper, C., & Phelps, S. (2016). Spectral clustering using the kNN-MST similarity graph. *In Computer Science and Electronic Engineering (CEEC)*, 2016 8th (pp. 222-227). IEEE.