Node Clustering with Traditional and Neural Network Approaches

Fa ZHANG
Department of Mathematics
HKUST
fzhangat@connect.ust.hk

Ruizhe XIA
Department of Mathematics
HKUST
rxiaac@connect.ust.hk

Abstract

The graph characterized by the adjacency matrix is a widely existing type of data. A common task is to perform clustering for nodes without labels. Many methods have been proposed, we compare some traditional methods and neural network methods on Zachery's Karate club dataset and political blogs dataset to demonstrate their strengths and weaknesses. We find traditional methods are easy to interpret but do not achieve a promising performance. The neural network methods are very flexible and computationally efficient and outperform the traditional methods.

1 Introduction

In traditional machine learning tasks, the typical data is the feature matrix with rows as samples and columns as features. We train models to perform different tasks such as clustering and classification. Then we can handle new samples by plugging in their feature vectors. However, this paradigm doesn't take the dependency between samples into consideration. For example, people who are friends with each other on social networks are more likely to have similar hobbies, and stocks in the same industry are more likely to have similar performance in the market.

We can use the graph to capture the dependency between samples. Let $G = (V, E)$ be an undirected graph with vertex set $V = \{v_1, \cdots, v_n\}$, we use $i \sim j$ to denote that $v_i \in V$ is a neighbor of $v_j \in V$, i.e. $(i, j) \in E$. The adjacency matrix $\mathbf{A}$ of the graph G is the matrix

$$A_{ij} = \begin{cases} 1 & i \sim j \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

The degree of a vertex $v_i \in V$ is defined as

$$d_i = \sum_{j=1}^{n} A_{ij}, \tag{2}$$

and then we define a diagonal matrix $\mathbf{D} = \mathrm{diag}(d_i)$.

The Normalized Graph Laplacian is defined as

$$\mathbf{L}_{sym} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}, \tag{3}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the unnormalized graph Laplacian matrix.

Since the graph models the dependency between samples, we consider a naive situation, given the graph, we wish to perform clustering of nodes. Many methods have been proposed to handle this problem. Traditional methods, usually, they are model-based, which means an elegant model will be specifically designed for this task or this dataset. Their results are transparent and interpretable.

However, the model assumptions are not always met by the real data, which means they lack flexibility. Another is the data-driven neural network methods, they usually provide a general framework and let the neural network adapt to the data. They are very flexible and computationally efficient, however, they are blamed black boxes.

We choose four representative methods: spectral clustering, transition path (traditional methods), deep walk, and Node2Vec (neural network methods) to perform node clustering on two datasets: Zachery's Karate club dataset and political blogs dataset to demonstrate their strengths and weaknesses.

2 Background and Related Work

2.1 Spectral Clustering

Spectral clustering is one of the most popular clustering algorithms. Donath and Hoffman first suggest constructing graph partitions based on eigenvectors of the adjacency matrix. Fiedler discovers that bi-partitions of a graph are closely connected with the second smallest eigenvector of the graph Laplacian [1]. Shi and Malik [4] propose the normalized cut criterion and an efficient computational technique to minimize the criterion by computing the second smallest eigenvector of the normalized graph Laplacian. However, spectral clustering is computationally expensive since it runs an eigenvalue decomposition. The complexity is super-quadratic to the number of nodes.

A graph $G = (V, E)$ can be partitioned into two disjoint sets A, B, where $A \cup B = V$ and $A \cap B = \emptyset$. Shi and Malik propose the normalized cut(Ncut), a measure of disassociation between two groups:

$$\mathrm{Ncut}(A, B) = \frac{\mathrm{cut}(A, B)}{\mathrm{assoc}(A, V)} + \frac{\mathrm{cut}(A, B)}{\mathrm{assoc}(B, V)} \tag{4}$$

where $\mathrm{cut}(A, B) = \sum_{u \in A, v \in B} A_{uv}$ and $\mathrm{assoc}(A, V) = \sum_{u \in A, t \in V} A_{ut}$.

Let $\mathbf{x} \in \mathbb{R}^n$ be an indicator vector such that $x_i = 1$ if $v_i \in A$ and $x_i = -1$ if $v_i \in B$. They show that the minimum of the Ncut

$$\min_{\mathbf{x}} \mathrm{Ncut}(\mathbf{x}) \quad \text{s.t} \quad x_i = \pm 1. \tag{5}$$

is the same as the minimum of the following Rayleigh quotient,

$$\min_{\mathbf{y}} \frac{\mathbf{y}(\mathbf{D} - \mathbf{W})\mathbf{y}}{\mathbf{y^T D y}} \quad \text{s.t.} \quad y_i \in \{1, -\frac{\sum_{x_i > 0} d_i}{\sum_{x_i < 0} d_i}\}, \mathbf{y}^T \mathbf{D1} = 0. \tag{6}$$

If $\mathbf{y}$ can take real values, then let $\mathbf{z} = \mathbf{D}^{1/2}\mathbf{y}$ and $\mathbf{z}_0 = \mathbf{D}^{1/2}\mathbf{1}$. We can minimize (6) by minimizing the following Rayleigh quotient

$$\min_{\mathbf{z^T z}_0 = 0} \frac{\mathbf{z}^T \mathbf{L}_{sym} \mathbf{z}}{\mathbf{z^T z}}. \tag{7}$$

We can easily verify that $\mathbf{L}_{sym}$ is symmetric positive semidefinite and 0 is an eigenvalue of $\mathbf{L}_{sym}$ with eigenvector $\mathbf{z}_0$. Therefore, the second smallest eigenvector $\mathbf{z}_1$ is the minimizer of (7) and $\mathbf{y}_1 = \mathbf{D}^{-1/2}\mathbf{z}_1$ is the solution of (6). Then we can use $\mathbf{y}_1$ to bipartition the graph.

2.2 Transition Paths

The spectral method can apply to a directed graph viewed as a Markov chain. The adjacency matrix $\mathbf{A}$ of a graph can be converted to a transition probability matrix $\mathbf{P}$ to generate a random walk on a graph. That means we can understand the clustering under the framework of state clustering on Markov chains [2].

From the probabilistic framework, one can use the transition path theory(TPT). TPT is developed in the context of continuous-time Markov chains and later adopted to discrete-time Markov chain with transition probability matrix $\mathbf{P}$ [3]. The committor function tells the success rate of the transition and leads to a natural graph decomposition. It also provides statistical properties to characterize the importance of edges and nodes.

Assume that an irreducible Markov Chain on graph $G = (V, E)$ admits the decomposition $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$. Let $V = V_l \cup V_u$ be a partition of V, where $V_l = V_0 \cup V_1$ denotes the labeled vertices with source set V_0 and target set V_1, and V_u is the unlabeled vertex set.

The committor function gives the probability that a trajecory start at the state $i \in V$ will hit V_1 before V_0. Let the first hitting time of $V_k, k = 0, 1$ be

$$\tau_i^k = \inf\{t \geq 0 : x(0) = i, x(t) \in V_k\}, \quad k = 0, 1. \tag{8}$$

Define the committor function as

$$q(i) = \mathbb{P}(\tau_i^1 < \tau_i^0). \tag{9}$$

We compute the value q_i by solving the following boundary value problem,

$$\begin{cases} [(\mathbf{Id} - \mathbf{P})\mathbf{q}](i) = 0 & i \in V_u \\ q(i) = 0 & i \in V_0 \\ q(i) = 1 & i \in V_1 \end{cases} \tag{10}$$

The set $\{v_i : q(i) < 0.5\}$ consists of the points more likely to hit V_0 first. Thus, the committor function provides a decomposition of graph G via $V = \{v_i : q(i) < 0.5\} \cup \{v_i : q(i) \geq 0.5\}$.

The committor function can also characterize the transition mechanism from V_0 to V_1. Given an equilibrium trajectory $\{X(t)\}_{t\in\mathbb{N}}$, a sequence $(x_{t_1}, x_{t_1+1}, \cdots, x_{t_2})$ satisfies $x_{t_1} \in V_0$, $x_{t_2} \in V_1$ and $x_{t_k} \in (V_0 \cup V_1)^c$, then it is a reactive trajectory from V_0 to V_1. Let R be the union of such trajectories, it is a random set whose statistical properties are induced by those of the ensemble of equilibrium trajectories.

The probability distribution of reactive trajectories π_R gives the equilibrium probability that the system in state x is reactive at time t, and it can be computed as

$$\pi_R(x) = \pi(x)q(x)(1 - q(x)). \tag{11}$$

The reactive current from V_0 to V_1 is given by

$$J(xy) = \begin{cases} \pi(x)(1 - q(x))P_{xy}q(y) & x \neq y \\ 0, & \text{otherwise} \end{cases}. \tag{12}$$

$J(xy)$ gives the average rate that the reactive trajectories flow from state x to state y. Based on the reactive current, one can define effective current and transition current to characterize the importance of an edge and a node respectively.

The effective current of an edge xy is defined as

$$J^+(xy) = \max(J(xy) - J(yx), 0); \tag{13}$$

The transition current through a node $x \in V$ is defined as

$$T(x) = \begin{cases} \sum_{y\in V} J^+(xy) & x \in V_0 \\ \sum_{y\in V} J^+(yx) & x \in V_1 \\ \sum_{y\in V} J^+(xy) & x \in V_u \end{cases} \tag{14}$$

2.3 DeepWalk

DeepWalk, introduced by Perozzi, Al-Rfou, and Skiena in 2014, is a method for learning embeddings of nodes in a graph. The basic idea behind DeepWalk is to use random walks on a graph to generate sequences of nodes, and then use these sequences to learn embeddings for each node using a neural network.

To be more specific, we first generate a set of random walks of fixed length from each node in the graph. These random walks are essentially paths through the graph that are generated by moving from one node to a neighboring node at each step. Then we transform the graph structure into sequences, and nodes that are close to each other in the graph will also be close to each other in the sequence. Then the skip-gram model, which is typically used for natural language processing tasks, can be directly applied to learn the co-occurrence of nodes, nodes that are close to each other in the graph will also have a close embedding. By representing nodes as vectors, DeepWalk allows us to leverage the power of machine learning algorithms for graph analysis tasks.

2.4 Node2Vec

Node2Vec introduced by Grover and Leskovec in 2016 is very similar to deep walk. It also generates random walks on the graph and uses these walks to learn node embeddings. The main difference is Node2Vec uses a biased random walk strategy that captures both the local and global structure of the graph. The strategy is controlled by two parameters, p, and q, which determine the likelihood of returning to a previous node or exploring a new neighborhood. The resulting random walks are also used to train a Skip-gram model to output latent embeddings for each node in the graph.

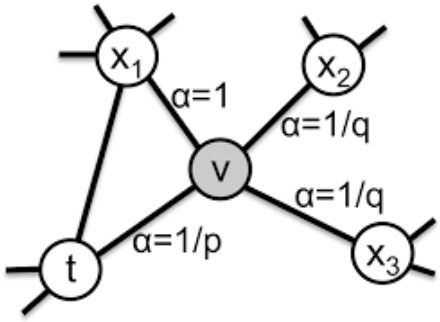


Figure 1: The walk just transitioned from t to v and is now evaluating its next step.

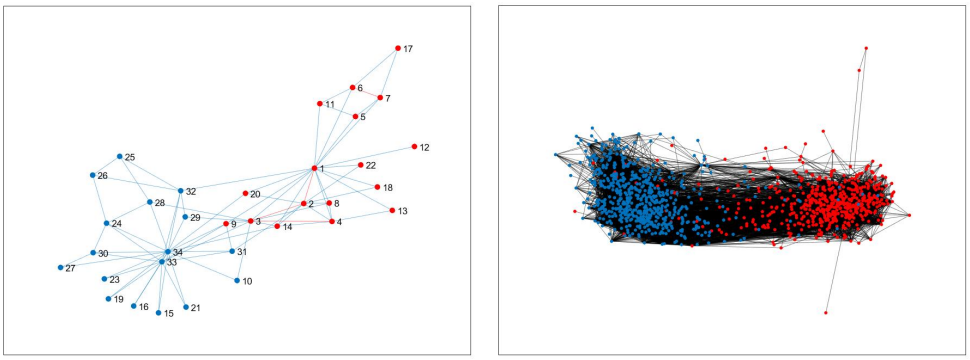
3 Data

3.1 Zachary's Karate Club Dataset

Zachary's Karate club dataset is a social network of friendships between 34 members of a karate club. The *karate.mat* contains a adjacency matrix $\mathbf{A} \in \mathbb{R}^{34\times34}$ of the club network and a one-hot vector. $\mathbf{c} \in \mathbb{R}^{34\times1}$ of group label.

`https://github.com/yao-lab/yao-lab.github.io/blob/master/data/karate.mat`

As shown in Figure 2(a), the club eventually breaks up into two groups, where node 1 (in red) represents the coach of the club and node 34 (in blue) is the owner of the club.



(a) Zachary's Karate Club Dataset (b) Political Blogs Dataset

Figure 2: Visualization

3.2 Political Blogs Dataset

The Political Blogs Dataset is a collection of blogs and hyperlinks between them, which has been widely used as a benchmark dataset for graph algorithms. The dataset was compiled by Lada A.

Adamic and Natalie Glance in 2005. The nodes represent political blogs and their corresponding political orientation are either "liberal" or "conservative". The edges represent the hyperlinks between blogs. As shown in Figure 2(b), the red nodes represent the Liberal while the blue nodes represent the Conservative.

4 Experiments and Results

4.1 Node Clustering

We perform clustering of nodes on the graph of Zachary's Karate Club and Political Blogs mentioned above using four methods: Spectral Clustering(SC), Transition Path Theory(TPT), DeepWalk, and Node2Vec, and use the following metric

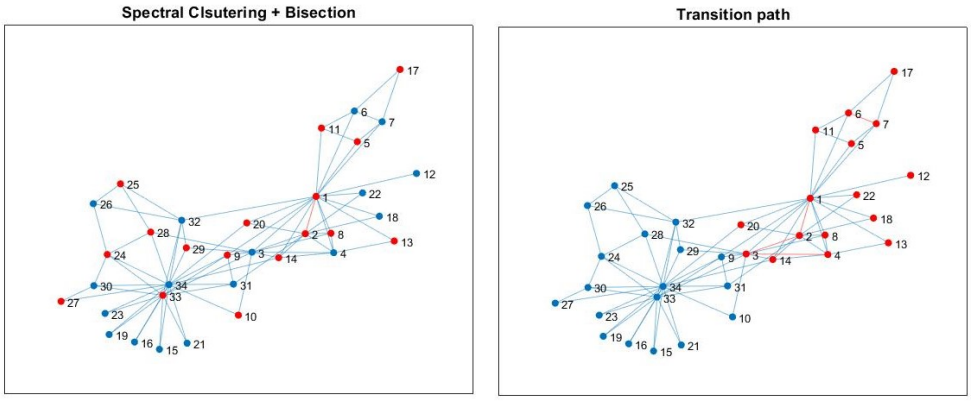
$$\text{Accuarcy} = \frac{\text{Number of correct predictions}}{\text{Total number of prediction}}. \tag{15}$$

to compare their performance.

The results are shown below:

Accuracy % Methods / Dataset	SC	TPT	DeepWalk	Node2Vec
Zachary's Karate Club Dataset	58.82	97.06	97.06	97.06
Political Blogs Network	51.80	94.68*	95.51	96.16

Table 1: Accuracy



(a) Bisection (b) Transition Path

Figure 3: Traditional Methods

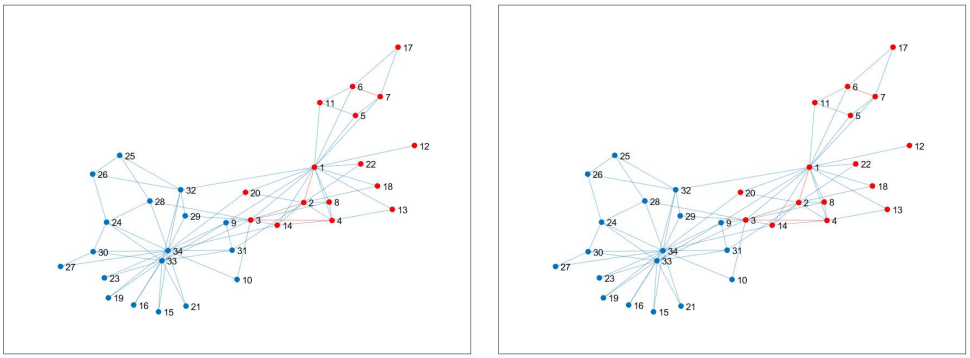
We can see neural network methods outperform traditional methods in terms of classification accuracy. The main reason is neural network methods do not specify the form of models, instead, they use neural networks to adapt to data to allow flexibility.

4.2 Transition Paths

We also use effective and transition flux to analyze the importance of edges and nodes.

4.2.1 Zachary's Karate Club Dataset

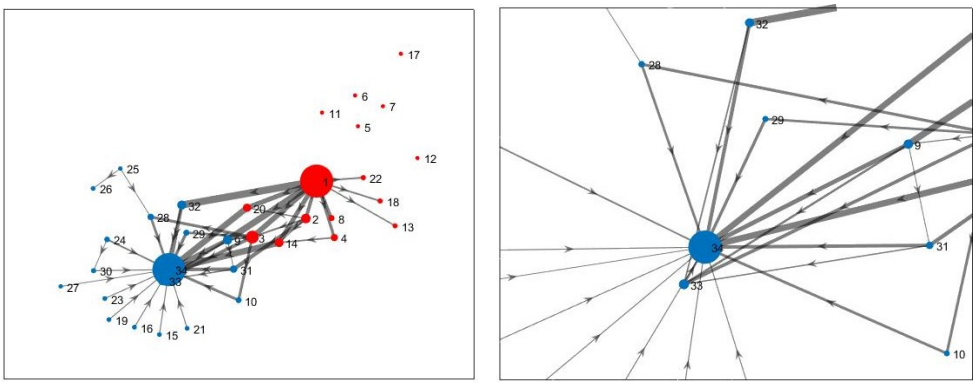
The transition flux shows that node 1 and node 34 are the most important, which coincides with the story: the conflicts between the coach and the owner finally lead to the fission of the club. Except for nodes 1 and 34, nodes 2,3,9,14,20,32,33 also have large transition flux. They lie in some paths



(a) DeepWalk (b) Node2Vec

Figure 4: Neural Network Methods

connecting nodes 1 and 34 with a large effective flux such as $1-20-34, 1-9-34, 1-32-34$. This indicates their relationships are crucial in this network. Almost all the blue nodes have effective flux to node 34 while there are some nodes 5,6,7,11,12,17 isolated in the graph. This indicates that there is still a hidden group in this network, the members in this group have little relation with the fission, though it finally stands with node 1.



(a) Transition Path (b) Zoom in Node 34

Figure 5: Effective/Transition Flux of Zachary's Karate Club Dataset

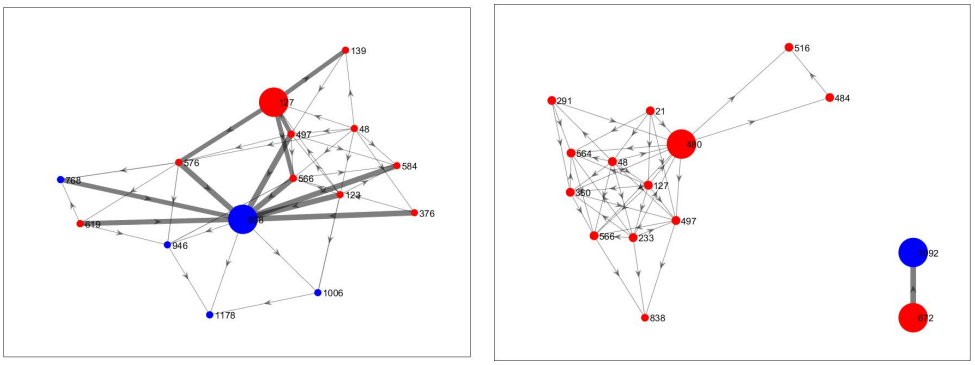
4.2.2 Political Blogs Network

We will encounter two important nodes 127 and 838. According to [7], node 127 is the top 1 Liberal blog *dailykos.com* and node 838 is the top 2 Conservative blog *instapundit.com*.

In the first experiment, we choose the node with the largest degree to be the source state and the node with the second largest degree to be the target state. The source state happens to be node 127 and the target state is 838.

In the second experiment, we randomly choose the source state from the nodes with label 0, and the target state from label 1. Then we run 100 random trials and the accuracy is $50.32 \pm 2.05\%$. Figure 6(b) displays one of the results. It doesn't provide a reasonable explanation for the network structure, since the transition from the source state to the target state is isolated and it even classifies nodes 127 and 838 into one class.

The quality of the source/target state(known labels) will affect the performance of TPT dramatically. That means we need high-quality samples when applying the TPT methods.



(a) Good source/target state (Acc = 94.68%) (b) Random source/target state(Acc = 48.04%)

Figure 6: Subgraph of Top 15 Nodes in Political Blogs Network

4.3 Perturbation Analysis

In real applications, we don't have access to the exact information of a network. We analyze the performance of the above four methods when the edge structures are distorted. First, we randomly delete k edges from the original Zachary's karate club network, subject to the constraint that the distorted graph is still connected. Second, we randomly add k edges by randomly choosing k pairs of nodes in the network. We perform 50 random trials for $k = 1, 2 \cdots, 7$ respectively. Table 1,2 display the mean and standard derivation of the accuracy. We can find that DeepWalk and Node2Vec are more stable than SC and TPT.

	1	2	3	4	5	6	7
SC	50.92+10.60	47.86+10.68	46.90+10.26	46.20+10.14	44.40+9.75	44.61+9.98	44.62+9.99
TPT	96.39+1.54	96.13+1.68	95.75+1.92	95.60+2.15	95.56+2.22	95.33+2.35	95.14+2.47
DeepWalk	96.76+0.01	96.53+0.01	96.24+0.01	96.20+0.02	96.12+0.01	96.12+0.01	96.02+0.02
Node2Vec	96.88+0.01	96.59+0.01	96.82+0.01	96.41+0.01	96.41+0.02	96.82+0.01	96.12+0.02

Table 2: Missing Edges

	1	2	3	4	5	6	7
SC	50.82+10.21	49.29+10.23	49.06+10.03	47.46+9.31	47.48+9.07	46.06+8.54	46.42+8.50
TPT	96.71+0.95	96.45+1.28	96.32+1.30	96.22+1.52	96.06+1.63	95.96+1.75	95.86+1.72
DeepWalk	96.65+0.01	96.35+0.01	96.41+0.01	96.00+0.02	96.06+0.01	95.88+0.02	95.35+0.04
Node2Vec	96.76+0.01	96.76+0.01	96.71+0.01	96.82+0.01	96.76+0.01	96.76+0.01	96.71+0.01

Table 3: Noisy Edges

5 Conclusions

The advantage of traditional methods is their interpretability. For example, we can use TPT to analyze the importance of edges. However, it loses the flexibility to capture complex nonlinear relationships, and it's also computationally expensive. We can see neural network methods DeepWalk and Node2Vec outperform SC and TPT in two graph datasets. The data-driven approach makes neural network methods very flexible. Besides, they are scalable and efficient by only computing the gradient on GPU.

Contribution

Fa Zhang: literature review, results analysis.

Ruizhe Xia: code implementation, report writing.

References

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

[2] Meila, M. & Shi, J. (2000). Learning segmentation by random walks. *Advances in neural information processing systems*, 13.

[3] Jianfeng, L. & Yuan, Y. (2013). The landscape of complex networks: Critical nodes and a hierarchical decomposition. *Methods and Applications of Analysis*, 20(4), 383-404.

[4] Shi, J. & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8), 888-905.

[5] Perozzi, B., Al-Rfou, R., & Skiena, S. (2014, August). Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 701-710).

[6] Grover, A., & Leskovec, J. (2016, August). node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 855-864).

[7] Adamic, L. A., & Glance, N. (2005, August). The political blogosphere and the 2004 US election: divided they blog. In Proceedings of the 3rd international workshop on Link discovery (pp. 36-43).