Using Graph Convolutional Networks to Compute Approximations of Dominant Eigenvectors

Ping-En Lu and Cheng-Shang Chang
Institute of Communications Engineering
National Tsing Hua University
Hsinchu, 30013 Taiwan
j94223@gmail.com, cschang@ee.nthu.edu.tw

ABSTRACT

Graph Convolutional Networks (GCN) have been very popular for the network embedding problem that maps nodes in a graph to vectors in a Euclidean space. In this short paper, we show that a special class of GCNs compute approximations of dominant eigenvectors of symmetric matrices with zero column sums.

1. THE EMBEDDING PROBLEM

Given a set of n data points (called nodes in the paper) $\{u_1, u_2, \dots, u_n\}$, the embedding problem is to map the n data points to vectors in a Euclidean space so that points that are "similar" to each other are mapped to vectors that are close to each other. Such a problem is an ill-posed problem [1] as people might have different interpretations of "similarity" between two points. One commonly used "similarity" measure is a bivariate distribution $p_{U,W}(u, w)$ that measures the probability that the two points u and w are "sampled" together. One can further subtract the product of the marginal distributions to construct a zero mean matrix $Q = (q(u, w))$ with

$$q(u, w) = p_{U,W}(u, w) - p_U(u)p_W(w). \tag{1}$$

Such a matrix is known as the (generalized) *modularity* matrix in the literature [2, 3, 4].

In order to map data points that are similar to each other to vectors that are close to each other, the embedding problem can be formulated as the optimization problem that minimizes the following weighted distance:

$$\sum_{u=1}^{n}\sum_{w=1}^{n} q(u, w)||h_u - h_w||^2, \tag{2}$$

where $h_u = (h_{u,1}, h_{u,2}, \dots, h_{u,K})^T$ is the vector mapped by point u in $\mathbb{R}^K$, and $||h_u - h_w||^2$ is the squared Euclidean distance between u and w. To understand the intuition of the minimization problem in (2), note that $-1 \le q(u, w) \le 1$. Two points with a positive (resp. negative) "covariance" should be mapped to two points with a small (resp. large) distance. The embedding vector $h_u = (h_{u,1}, h_{u,2}, \dots, h_{u,K})^T$ can be viewed as the "feature" vector of point u and $h_{u,k}$ is its k^{th} feature. In practice, it is preferable to have uncorrelated features. For this, we add the constraints

$$\sum_{u=1}^{n} h_{u,k_1}h_{u,k_2} = 0, \tag{3}$$

for all $k_1 \neq k_2$. Also, to have bounded values for these features, we also add the constraints

$$\sum_{u=1}^{n} h_{u,k}h_{u,k} = 1, \tag{4}$$

for all k.

For such an embedding problem, the following equivalent statements were shown in the book chapter [1].

THEOREM 1. *([1], Theorem 3) Let $\mathbf{Q} = (q_{u,w})$ be an $n \times n$ symmetric matrix with all its row sums and column sums being 0, and H be the $n \times K$ matrix with its u^{th} row being h_u.*

(i) *The embedding problem in (2) with the constraints in (3) and (4) is equivalent to the following optimization problem:*

$$\max \quad tr(H^T\mathbf{Q}H) \tag{5}$$
$$s.t. \quad H^TH = \mathbf{I}_K, \tag{6}$$

where $\mathbf{I}_K$ is the $K \times K$ identify matrix.

(ii) *The embedding problem in (2) with the constraints in (3) and (4) is equivalent to the following optimization problem:*

$$\min \quad ||\mathbf{Q} - HH^T||_2^2 \tag{7}$$
$$s.t. \quad H^TH = \mathbf{I}_K, \tag{8}$$

where $||A||_2$ is the Frobenius norm of the matrix A.

From Theorem 1(i), we know that solving the embedding problem is equivalent to solving the trace maximization problem in (5). As stated in [5], a version of the Rayleigh-Ritz theorem shows that the solution of the trace maximization problem in (5) can be found by solving the dominant eigenvectors of the matrix $\mathbf{Q}$. This is stated in the following corollary.

COROLLARY 2. *For the embedding problem in (2) with the constraints in (3) and (4), let $\lambda_1, \lambda_2, \dots, \lambda_K$ be the K largest eigenvalues of the matrix $\mathbf{Q}$ and $v_k = (v_{k,1}, v_{k,2}, \dots, v_{k,n})^T$ be the eigenvector of $\mathbf{Q}$ corresponding to the eigenvalue λ_k. Then $h_u = (v_{1,u}, v_{2,u}, \dots, v_{K,u})^T$, $u = 1, 2, \dots, n$, are the optimal embedding vectors.*

2. GRAPH CONVOLUTIONAL NETWORKS

As mentioned in the previous problem, the embedding problem can be solved by finding the dominant eigenvectors of the matrix Q. When n is small, this can be done by the power (orthogonal iteration) method (see, e.g., [6]) with $O(n^3)$ computational complexity and $O(n^2)$ memory complexity [7]. In [7], a fast Chebyshev polynomial approximation algorithm was proposed to avoid the need for eigen decomposition of the matrix Q, and that motivated Kipf and Welling [8] to propose Graph Convolution Networks (GCN) for semi-supervised classification. A GCN obtains the embedding vectors by carrying out the following iterations:

$$H^{(\ell+1)} = \sigma(QH^{(\ell)}W^{(\ell)}), \tag{9}$$

where $W^{(\ell)}$'s are trainable weight matrices and σ is an activation function (used in a neural network). In this paper, we do not need the trainable weight matrices $W^{(\ell)}$' and they are removed from (9). This leads to the following simplified GCN:

$$H^{(\ell+1)} = \sigma(QH^{(\ell)}). \tag{10}$$

Instead of using the ReLU function in [8], we use the softmax function as our activation function. The softmax function σ with K inputs $z_1, z_2, \ldots, z_K$ generate the K outputs

$$\sigma(z_1, z_2, \ldots, z_K) = \frac{1}{\sum_{\ell=1}^{K} e^{\theta z_\ell}}(e^{\theta z_1}, e^{\theta z_2}, \ldots, e^{\theta z_K}), \tag{11}$$

where $\theta > 0$ is the inverse temperature. One nice feature of using the softmax function is that now every row of $H^{(\ell)}$ is a probability vector (with all its K nonnegative elements summing to 1). This leads to a probabilistic explanation of how the GCN in (10) works. Let

$$h_u^{(\ell)} = (h_{u,1}^{(\ell)}, h_{u,2}^{(\ell)}, \ldots, h_{u,K}^{(\ell)})^T \tag{12}$$

be the transpose of the u^{th} row of the matrix $H^{(\ell)}$. As pointed out in [8], one can view the GCN as a special case of the Weisfeiler-Lehman (WL) algorithm [9] that assigns the n points to K colors (or clusters). The probability $h_{u,k}^{(\ell)}$ then represents the probability that the u^{th} point is assigned with color k. The GCN starts from a non-uniform probability mass function for the assignment of each data point to the K colors. Then we repeatedly feed each point to the GCN to learn the probabilities $h_{u,k}^{(\ell)}$'s. When point u is presented to the GCN, its expected "covariance"

$$z_{u,k}^{(\ell)} = \sum_{w=1}^{n} q(u,w)h_{u,k}^{(\ell)} \tag{13}$$

to color k is computed for $k = 1, 2, \ldots, K$. Instead of assigning point u to the color with the largest positive "covariance" (the simple maximum assignment), GCN uses a softmax function to update $h_{u,k}^{(\ell)}$'s. Such a softmax update increases (resp. decreases) the confidence of the assignment of point u to colors with positive (resp. negative) "covariances." The "training" process is repeated until it converges. A sequential implementation of the GCN in (10) (like the usual training process of a neural network) is exactly the same as the the softmax embedding/clustering algorithm in Algorithm 1 of [1]. In Theorem 5 of [1], it was shown that the GCN converges to a local optimum of the objective value $\mathrm{tr}(H^T\mathbf{Q}H)$. This is stated in the following theorem.

THEOREM 3. *([1], Theorem 5) Given a symmetric matrix $\mathbf{Q} = (q(u,w))$ with $q(u,u) = 0$ for all u, the following objective value*

$$tr(H^T\mathbf{Q}H) = \sum_{k=1}^{K}\sum_{u=1}^{n}\sum_{w=1}^{n} q(u,w)h_{u,k}h_{w,k} \tag{14}$$

is increasing after each update for the sequential implementation of the GCN in (10). Thus, the objective values converge monotonically to a finite constant.

For $K = 2$, we have $h_{u,1} + h_{u,2} = 1$ for all u. Using this in (14) yields

$$\begin{aligned} &\mathrm{tr}(H^T\mathbf{Q}H) \\ &= \sum_{u=1}^{n}\sum_{w=1}^{n} q(u,w)h_{u,1}h_{w,1} + \sum_{u=1}^{n}\sum_{w=1}^{n} q(u,w)h_{u,2}h_{w,2} \\ &= 2\sum_{u=1}^{n}\sum_{w=1}^{n} q(u,w)h_{u,1}h_{w,1}, \end{aligned} \tag{15}$$

where we use the assumption that all the row sums and column sums of the matrix $\mathbf{Q}$ are equal to 0. As a result of Theorem 3 and Theorem 1, the GCN with $K = 2$ obtains a local optimum solution for the one-dimensional trace maximization in (5). This implies that

$$\sum_{u=1}^{n}\sum_{w=1}^{n} q(u,w)\frac{h_{u,1}}{\sqrt{\sum_{\ell=1}^{n} h_{\ell,1}^2}}\frac{h_{w,1}}{\sqrt{\sum_{\ell=1}^{n} h_{\ell,1}^2}}$$

should be very close to the largest eigenvalue of the matrix $\mathbf{Q}$. This motivates us to consider the n-vector $x = (x_1, x_2, \ldots x_n)^T$, where

$$x_u = \frac{h_{u,1}}{\sqrt{\sum_{\ell=1}^{n} h_{\ell,1}^2}}. \tag{16}$$

We will show in the next section that x is close to the eigenvector corresponding to the largest eigenvalue of $\mathbf{Q}$.

3. THEORETICAL BOUNDS AND NUMERICAL RESULTS FOR $K = 2$

Recall that $\mathbf{Q} = (q(u,w))$ is a symmetric matrix with all its row sums and column sums being 0. It is well-known that a real symmetric matrix is diagonalizable by an orthogonal matrix. Specifically, let

$$\lambda_1 > \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_n$$

be the (ordered) n eigenvalues of $\mathbf{Q}$ and v_i be the normalized (column) eigenvector corresponding to the eigenvalue λ_i, $i = 1, 2, , \ldots, n$. Let $V = [v_1 v_2 \ldots v_n]$ be the $n \times n$ orthogonal matrix formed by grouping the n eigenvectors together. Then

$$V^T\mathbf{Q}V = D, \tag{17}$$

where D is diagonal matrix with the n diagonal elements, $\lambda_1, \lambda_2, \ldots, \lambda_n$.

For our analysis, we also assume that there is a spectral gap between the largest eigenvalue and the second largest eigenvalue magnitude (SLEM), i.e., $\lambda_1 > \max[\lambda_2, -\lambda_n]$. Since $\mathbf{Q}$ has an eigenvalue 0 with the eigenvector $\mathbf{e}$ (with all its elements being 1), we know from the spectral gap assumption that $\lambda_1 > \lambda_2 \geq 0$. Moreover, as two eigenvectors corresponding two different eigenvalues are orthogonal for a real symmetric matrix, we then have $v_1^T \mathbf{e} = 0$.

In this paper, we use the cosine similarity to measure the difference between two vectors.

Definition 4. *The cosine similarity between two n-vectors x and y, denoted by $COS(x, y)$, is*

$$COS(x, y) = \frac{x^T y}{\sqrt{x^T x}\sqrt{y^T y}}. \tag{18}$$

In particular, if both x and y are unit vectors, i.e., $x^T x = y^T y = 1$, then the Euclidean distance between these two vectors is

$$\sqrt{(x-y)^T(x-y)} = \sqrt{2(1 - COS(x, y))}.$$

As such, if the cosine similarity between two unit vectors is close to 1, then the Euclidean distance between these two unit vectors is close to 0.

In the following theorem, we show a lower bound for the cosine similarity between the vector x and v_1. Its proof is given in Appendix A of this paper.

Theorem 5. *Let δ the ratio of the SLEM to the largest eigenvalue of $\mathbf{Q}$, i.e.,*

$$\delta = \frac{\max[\lambda_2, -\lambda_n]}{\lambda_1}. \tag{19}$$

Consider the vector x in (16). If

$$x^T \mathbf{Q} x \geq \lambda_1 (1 - \epsilon) \tag{20}$$

for some ϵ satisfying

$$0 \leq \epsilon \leq 1 - \delta, \tag{21}$$

then

$$COS(v_1, x) \geq \sqrt{\frac{1 - \epsilon - \delta}{1 - \delta}}. \tag{22}$$

Moreover, $COS(v_1, \mathbf{Q}x) \geq COS(v_1, x)$ and

$$COS(v_1, \mathbf{Q}x) \geq \sqrt{\frac{1 - \epsilon - \delta}{1 - \epsilon - \delta + \delta^2}}. \tag{23}$$

Theorem 5 shows that if the GCN obtains a good solution for the trace maximization problem in the sense of (20) and (21), then it is close to the dominant eigenvector v_1 in terms of the bound of the cosine similarity in (22). Moreover, the vector $\mathbf{Q}x$ is even closer to v_1, and it is even a better approximation of v_1.

In Figure 1, we show the dominant eigenvector and the unit vector of $\mathbf{Q}x$ for four different datasets. As shown in Figure 1, the differences are very small and the GCN indeed computes good approximations of the dominant eigenvectors.

4. CONCLUSION

In this paper, we proposed a class of GCNs to compute approximations of dominant eigenvectors of real symmetric matrices with zero column sums. In general, the power method computes the dominant eigenvector by the following iteration:

$$x^{(\ell+1)} = \frac{Qx^{(\ell)}}{||Qx^{(\ell)}||}. \tag{24}$$

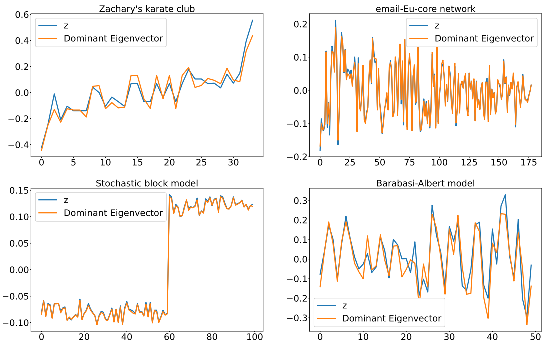

Figure 1: Comparison of the dominant eigenvector and the unit vector of $\mathbf{Q}x$ for four different datasets.

In comparison with the power method, the softmax function σ in our GCN that renormalizes the embedding vector back to a probability vector is a *local* method as it only requires local information from the node itself. On the other hand, the power method is a *global* method that requires information (from all the n nodes) for renormalization back to a unit vector. As such, our GCN approach is more scalable for large n.

5. REFERENCES

[1] C.-S. Chang, C.-C. Huang, C.-T. Chang, D.-S. Lee, and P.-E. Lu, "Generalized modularity embedding: a general framework for network embedding," *arXiv preprint arXiv:1904.11027*, 2019.

[2] M. E. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 6, p. 066133, 2004.

[3] C.-S. Chang, C.-J. Chang, W.-T. Hsieh, D.-S. Lee, L.-H. Liou, and W. Liao, "Relative centrality and local community detection," *Network Science*, vol. 3, no. 4, pp. 445–479, 2015.

[4] C.-S. Chang, D.-S. Lee, L.-H. Liou, S.-M. Lu, and M.-H. Wu, "A probabilistic framework for structural analysis and community detection in directed networks," *IEEE/ACM Transactions on Networking*, vol. 26, no. 1, pp. 31–46, 2018.

[5] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[6] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2012, vol. 3.

[7] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[8] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[9] B. Weisfeiler and A. A. Lehman, "A reduction of a graph to a canonical form and an algebra arising during this reduction," *Nauchno-Technicheskaya Informatsia*, vol. 2, no. 9, pp. 12–16, 1968.