NMF-GNN: Full Global Structure Enhancement of Structural-sensitive Message Passing in Graph Neural Networks*

Xinnan Dai
School of Information Science and Technology
ShanghaiTech University
Shanghai, China
201210
Chenyu He
School of Information Science and Technology
ShanghaiTech University
Shanghai, China
201210

I. Introduction

A. Current work

Currently typical message passing mechanisms of graph neural networks (GNN) are highly related to the graph structure. For example, graph attention networks (GAT) [7] passes information via attention sharing on global nodes, graph convolution network (GCN) [8] performs Laplacian approximations on spectral convolution. More specifically, message passing process of GCN follows the eigenvalue and eigenvector of Laplacian matrix, which is strong structure-related. However, there still exists information which are not position-aware, which indicates they are unable to be shared during the position aware message passing. For example, in relation networks like social networks and citation networks, there may exists orphan nodes which contains features not similar to those of connected nodes. Typical solution is random sampling on the whole node set of the graph, regardless of the edge connections. Thus successful extraction of these structure-independent features may significantly boosts the representation performance of the node. Typical solution is random sampling on the whole node set of the graph, regardless of the edge connections. For example, P-GNN [1] randomly samples several subsets containing arbitrary

Identify applicable funding agency here. If none, delete this.

count of nodes on the original graph, and stacks as a feature matrix. All subsets further aggregate into a shared vector by aggregation (subset-wise mean), and fused to each node feature vector. While GraphSTONE [2] divides the graph via random walk and find shared information from different samples on the unique structure topic. However, random sampling cannot perfectly extract the global information, which performance may influenced by the randomly sampled result. Mechanics with direct operation on the whole graph is required.

Non-negative matrix factorization (NMF) is a matrix factorization mechanism which is able to learn parts-based representations by 2 low rank matrices, as the linear combination of the data-based basis and weights [3], like other typical factorization methods like principle component analysis (PCA) and vector quantization (VQ). Among which, eigen-based methods PCA and VQ mainly extracts holistic features, where basis components manifesting building blocks of the original matrix may missing [4]. NMF makes the computation more efficient and has achieved wide application in engineering, such as data mining and image processing, to name a few. In image processing, [3] and [4] uses NMF to calculate the basis representation of human face data in facial recognition and classification tasks. In data mining, NMF is able to calculate the semantic feature representations in articles [3]. NMF has also become a basic baseline algorithm in recommendation system as representing the individual-item relation by refined individual-topic and topic-item relations. Besides, GraphSTONE [2] also uses NMF to factorize the key structure topic of sampled results from different random walks, and lead to the following principal component calculation. However typically, NMF is applied in end to end approaches as feature extractor, where potential distribution shift may exists between the learned representation by NMF and the original node features. Thus separated implement of NMF may still cause performance decreasing in contrast with the approach without NMF.

B. Our contribution

We purpose a novel approach enhancing the message passing of GNNs via non-negative matrix factorization (NMF), for leading the structure-invariant components into the node features. Inspired by above researches, we directly perform NMF on the feature matrix and gained a global basis matrix and the corresponding weight matrix. The operation on the whole matrix imposes the global feature distribution into the factorized basis, which is further embedded into a feature vector and send into message passing together with original node features. Furthermore, to better integrate NMF into message passing framework, the basis matrix is randomly initialized and update via backward propagation with the whole graph rather than explicitly perform NMF on the original feature matrix, to achieve distribution consistency between the feature of basis representation and node features.

II. METHODS

A. Non-negative matrix factorization

Non-negative matrix factorization (NMF) is a matrix factorization method wildly used in recommend systems and text mining. In the text mining task, we use this method to analyze text topics. For any text sets with n texts and m words, given an nonnegative matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$, it can find a nonnegative matrix $\mathbf{W} \in \mathbb{R}^{n\times t}$ to represent the relevance of each text to t topics, and a nonnegative matrix $\mathbf{H} \in \mathbb{R}^{t\times m}$ to show how the topics relate to the words,

Algorithm 1 NMF

Input: X, max iteration number $iter$ **Output:** W, H

random initial W matrix and H matrix
for i in $iter$ **do**
    $H' \leftarrow H \frac{W^T X}{W^T W H}$
    $W' \leftarrow W \frac{X H^T}{W H H^T}$
    $H \leftarrow H'$
    $W \leftarrow W'$
return W, H

which satisfies the condition $\mathbf{X} = \mathbf{WH}$, and decomposes a nonnegative matrix into the product of left and right nonnegative matrices. The text $\mathbf{X}$ is restructured by $\mathbf{W}$ and $\mathbf{H}$, consequently, the goal is $\mathbf{min}||\mathbf{X} - \mathbf{WH}||^{\mathbf{2}}$. The update rule for NMF algorithm is in Algorithm 1 according to [9].

B. Spatial Graph

Graph neural network is a method of connecting information between samples. This method can predict the information of neighbor nodes according to the properties of the points. Among them, most graph networks are based on spectral graph theory. A graph is defined as $G = (X, E)$, where $X = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}$ is the node set and $\mathbf{E} \in \mathbb{R}^{\mathbf{n} \times \mathbf{n}}$ represent whether exist the edges in the graph. To simulation the fusion on the graph, we use the Laplacian matrix which is $\mathbf{L} = \mathbf{D} - \mathbf{E} = \mathbf{dI} - \mathbf{E} = \mathbf{U \Lambda U^T}$, where $\mathbf{U}$ is eigenvector matrix and $\mathbf{\Lambda}$ is the eigenvalue diagonal matrix. According to the definition of the matrix, the motion direction of the vector can be changed by the matrix A. For each vector in the graph, the transition is $\lambda \mathbf{x} = \mathbf{A x}$. [10] In the Laplacian transmition, $A \in \mathbb{R}^{n \times n}$ is an approximate matrix based on Chebyshev Theory and $\theta \in \mathbb{R}^{n \times n}$. Thus, the transition is

$$\mathbf{g_\theta x} = \mathbf{U g_\theta U^T x} = \mathbf{U g_\theta(\Lambda) U^T x} = \mathbf{A x},$$

where $\mathbf{g_\theta}$ is the a diagonal matrix with Chebyshev characteristic of λ, and $\mathbf{g_\theta} = \mathbf{g_\theta(\Lambda)}$ [11]. Thus, the Laplasian transition is in the Chebyshev form where $\tilde{\mathbf{L}} = \mathbf{U \tilde{L} U^T}, \tilde{\mathbf{L}} = \mathbf{A}$ and since the $\mathbf{g_\theta x} = \mathbf{A x} = \tilde{\mathbf{L}} \mathbf{x}$. In GCN [8], the transition is approximate to,

$$\mathbf{g_\theta x} = \theta_{\mathbf{0}} \mathbf{x} - \theta_{\mathbf{1}} \mathbf{D^{-1/2} E D^{-1/2} x}.$$

For simplicity, supposing $\theta = \theta_0 = -\theta_1$, the form is,

$$\mathbf{g_\theta x} = \theta(\mathbf{I} + \mathbf{1 D^{-1/2} E D^{-1/2}})\mathbf{x}.$$

After the normalization,

$$\hat{\mathbf{X}} = \tilde{\mathbf{D}}^{\mathbf{-1/2}} \mathbf{E} \tilde{\mathbf{D}}^{\mathbf{-1/2}} \mathbf{\Theta X},$$

where $\mathbf{\Theta} \in \mathbb{R}^{\mathbf{n} \times \mathbf{m}}$.

As shown below, the GCN describ the information transition based on the edges. Similarly, the graph attention network (GAT) is in the form of $\hat{\mathbf{X}} = \mathbf{E W_b X}$, where $\mathbf{W_b}$ is the weight matrix from attention coefficient [7].

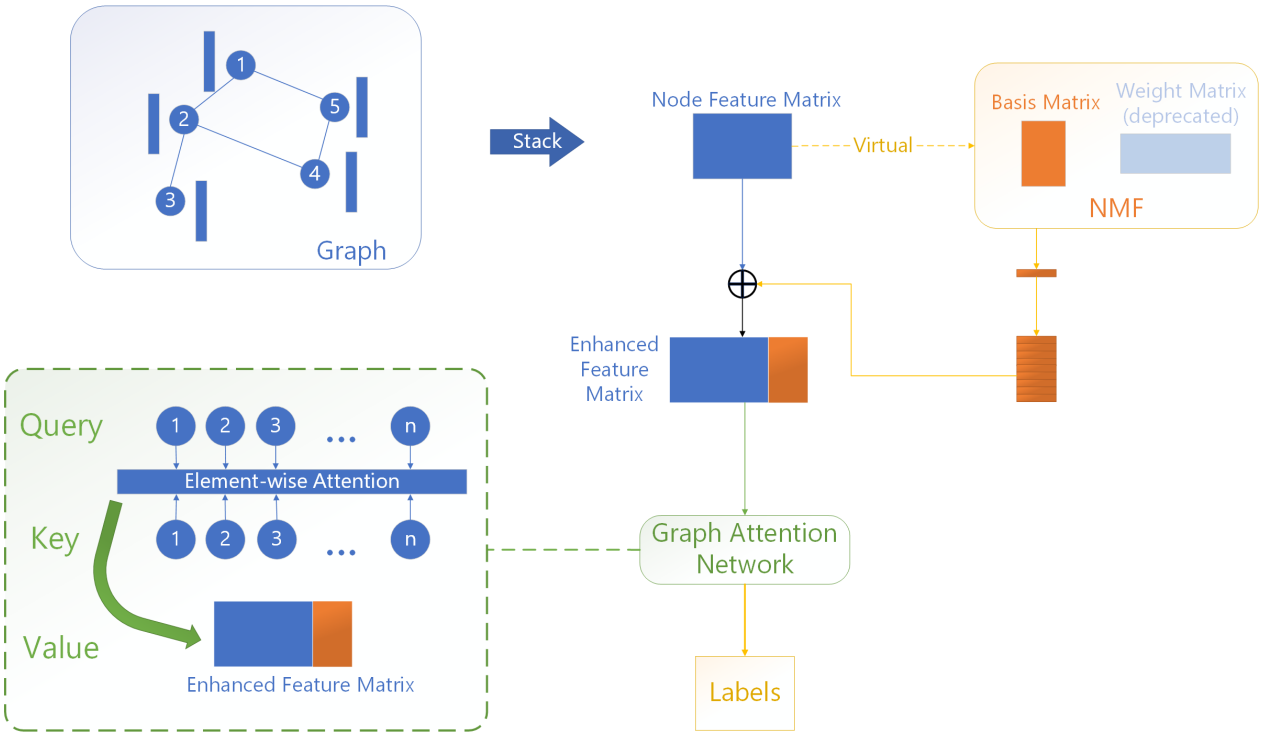

Figure 1. Illustrated general pipeline of NMF-GNN. Random initialized basis matrix is aggregated into a global feature vector and concatenated to each row of original feature matrix. Then the matrix is updated through GAT. Both shared weight matrix in GAT and basis matrix is updated through backpropagation

Both GCN and GAT defines the information transition through the edges with the information from neigbor nodes. However, the labels also depend on the similar contents among the whole dataset.

C. Pipeline

The general pipeline of our approach is shown in figure II-B. The original matrix of node features $\mathbf{X} \in \mathbb{R}^{n \times m}$ is factorized as $\mathbf{X} \approx \mathbf{WH}$, where $\mathbf{W} \in \mathbb{R}^{n \times k}$ and $\mathbf{H} \in \mathbb{R}^{k \times m}$ represents the basis and combination matrix respectively, where k is manually defined. Then basis matrix $\mathbf{W}$ is embedded into a feature vector $\mathbf{w}$ by aggregation operations. In our implementation the aggregation is chosen as row-wise mean, which is $\mathbf{w} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{W}_{i,:}$. These operation are all differentiable which can be optimized via backpropagation. Then $\mathbf{w}$ is concatenated with all node features $\mathbf{v}_i, i \in [1, N]$ into the new vector $\mathbf{v}'_i$, which is denoted as

$$\mathbf{v'}_i = \Big|\Big| \{\mathbf{v}_i, \mathbf{w}\}. \tag{1}$$

Then the fused vectors are taken into the following message passing defined by GAT. GAT calculates self-attention inside the node set, which attention score calculated between weighted key vectors $\mathbf{k}_i$ and query vectors $\mathbf{q}_j$ in the

node set, where the weight matrix $\mathbf{R}$ is shared in the message passing. The process can be mathematically expressed as

$$\mathbf{A} = \delta(\frac{\exp(\mathbf{RQK}^\intercal\mathbf{R}^\intercal)}{\sqrt{m}}) \tag{2}$$

where $\mathbf{A}_{ij}$ denotes the attention score between i-th query vector and j-th key vector, m denotes the feature dimension and $\delta()$ denotes a nonlinear activation function (LeakyRELU in the original approach). Then the vector is updated through attention-weighted fusing of other node features, which is

$$\mathbf{H}^{(t+1)} = \sigma(\mathbf{AWH}^{(t)}) \tag{3}$$

where $\sigma()$ is a nonlinear function.

D. Further work

1) Do the experiments on different datasets.
2) We need to prove that NMF solution is not unique.
3) If the solution is not unique, we should implement the multi-NMF to the GNN to capture different features.
4) Embed the NMF into the GAT network. NMF should share the parameters with GAT. Besides, we need to test the deep NMF model to show whether the heretical NMF would have fine-grained topics.

REFERENCES

[1] You, J., Ying, R., & Leskovec, J. (2019). Position-aware graph neural networks. arXiv preprint arXiv:1906.04817.

[2] Long, Q., Jin, Y., Song, G., Li, Y., & Lin, W. (2020, August). Graph Structural-topic Neural Network. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 1065-1073).

[3] Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755), 788-791.

[4] Feng, T., Li, S. Z., Shum, H. Y., & Zhang, H. (2002, June). Local non-negative matrix factorization as a visual representation. In Proceedings 2nd International Conference on Development and Learning. ICDL 2002 (pp. 178-183). IEEE.

[5] Huang, Z., Zhou, A., & Zhang, G. (2012, October). Non-negative matrix factorization: A short survey on methods and applications. In International Symposium on Intelligence Computation and Applications (pp. 331-340). Springer, Berlin, Heidelberg.

[6] Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., & Eliassi-Rad, T. (2008). Collective classification in network data. AI magazine, 29(3), 93-93.

[7] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.

[8] Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

[9] Lee, D. D., & Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In Advances in neural information processing systems (pp. 556-562).

[10] Trevisan, L. (2017). Lecture Notes on Graph Partitioning, Expanders and Spectral Methods. University of California, Berkeley, https://people. eecs. berkeley. edu/ luca/books/expanders-2016. pdf.

[11] Hammond, D. K., Vandergheynst, P., & Gribonval, R. (2011). Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis, 30(2), 129-150.