GenPR: Generative PageRank framework for Semi-Supervised Learning on citation graphs *

Mikhail Kamalov[1] and Konstantin Avrachenkov[1]

INRIA Sophia Antipolis, France,
{mikhail.kamalov, k.avrachenkov}@inria.fr

Abstract. Nowadays, Semi-Supervised Learning (SSL) on citation graph data sets is a rapidly growing area of research. However, the recently proposed graph-based SSL algorithms use a default adjacency matrix with binary weights on edges (citations), that causes a loss of the nodes (papers) similarity information. In this work, therefore, we propose a framework focused on embedding PageRank SSL in a generative model. This framework allows one to do joint training of nodes latent space representation and label spreading through the reweighted adjacency matrix by node similarities in the latent space. We explain that a generative model can improve accuracy and reduce the number of iteration steps for PageRank SSL. Moreover, we show that our framework outperforms the best graph-based SSL algorithms on four public citation graph data sets and improves the interpretability of classification results.

Keywords: semi-supervised learning · generative model · PageRank · citation graphs · neural networks

1 Introduction

The main idea of SSL is to solve a classification task with an extremely low number n_l of labeled data points in comparison with the number n_u of unlabeled data points ($n_l \ll n_u$). Therefore, with regard to citation graphs with a huge amount of nodes (e.g. Pubmed, MS Academic) SSL is a good technique to avoid preparing data points for supervised learning. The area of SSL focusing on the classification of nodes in citation graphs, in particular, citation graphs is called a graph-based SSL. The standard input for graph-based SSL algorithms is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $n = n_l + n_u = |\mathcal{V}|$ nodes (papers), $e = |\mathcal{E}|$ edges (citations), $A \in \mathbb{R}^{n \times n}$ is the adjacency matrix and $X = (x_{i,j})_{i,j=1}^{n,d}$ is a matrix of nodes where each node $x_i = (x_{i,1}, \ldots, x_{i,d}) \in \mathbb{R}^d$ has a feature representation in d-space. In the context of citation graphs X is a bag-of-words representation for the nodes (papers). Moreover, each node belongs to one of c classes $\{\mathcal{C}_1, \ldots, \mathcal{C}_c\}$. Also we have the labels matrix $Y = (y_{i,j})_{i,j=1}^{n,c} \in \mathbb{R}^{n \times c}$ such that $y_{i,j} = 1$ if $x_i \in C_j$ and $y_{i,j} = 0$ otherwise. Nowadays, the area of the graph-based SSL consists of two main research directions:

* Supported by MyDataModels company.

- the classical diffusion-based linear algorithms that spread the class information through the adjacency matrix A: Label Propagation (LP) [9], PageRank SSL (PRSSL) [1];
- the graph convolution-based neural network (NN) algorithms that apply the dot product of adjacency matrix A with NN nonlinear transformation of features X for the classification. The recently proposed: approximated Personalized graph NN (APPNP) [6], Graph Attention Network (GAT)[8], Graph Convolution Network (GCN) [5].

Regarding graph-based SSL algorithms, one can notice that they use A with binary weights on edges (citations), which can cause a loss of information about node similarities and further may negatively affect label diffusion through A. We address this issue through the following contributions:

- We propose a novel graph-based SSL inductive (I)/ transductive (T) framework, created by embedding PRSSL [1] in generative model (GenPR);
- We show that the generative model can be used to reweight A to further improve PRSSL label spreading;
- We show that GenPR improves the interpretability of NN classification results based on the information about nodes similarity in the latent space;
- We show that GenPR outperforms the recently proposed algorithms for graph-based SSL and reduces the number of steps of PageRank [7] to obtain more accurate classification results.

2 Related work

Our framework is based on the combination of the following two ideas:

1. PRSSL [1] gives a PowerIteration based explicit solution for the graph classification: $F^t = \alpha D^{-\sigma} A D^{\sigma-1} F^{t-1} + (1-\alpha)Y;\ t \geq 0$ where F^t is a result of the t-th iteration and α is a regularization parameter in the range $[0, 1]$ and σ is a power of $D_{i,i} = \sum_{j=1}^{n} A_{i,j}$;
2. generative semi-supervised model (M2) [4] : $p(x, \hat{y}, z) = p(x|z, \hat{y})p(\hat{y})p(z)$ where $p(z) = \mathcal{N}(z|0, I)$, $p(\hat{y})$ is a categorical distribution of latent class variable and $p(x|z, \hat{y})$ is a nonlinear transformation of the latent variables z and $\hat{y}$.

Since our framework is the graph convolution-based NN algorithm with PageRank, we need to define the main difference with APPNP [6]. The difference is that GenPR jointly trains a redefined generative model [4] and PRSSL [1] with a linear combination of A and similarity matrix in latent space, while APPNP applies PageRank with default A as a preprocessing step for output of multilayer perceptron (MLP).

3 Generative PageRank (GenPR)

3.1 Intuition of GenPR

Before we go into details of our framework let us define the motivation and the intuition behind GenPR. The main idea of GenPR is to resolve the following issues:

1. $A_{i,j} = 1$ does not provide the information about impact of cited paper j on the citing one i;
2. $A_{i,j} = 0$ may show that author i did not cite the paper j, but he could have used some information from it.

Let us define some useful notation for the GenPR intuition: $x_i \in X$ is a i.i.d. samples of some continuous random variable x, then an output of MLP $Y^* = (y_i^*)_{i=1}^n \in \mathbb{R}^{n \times c}$ is a sample from random variable y^* given x as an input; $Z = (z_i)_{i=1}^n \in \mathbb{R}^{n \times d'}$ where z_i is a latent representations of each node x_i sampled from latent random variable z in d'-space; $W = (w_{i,j})_{i,j=1}^n \in \mathbb{R}^{n \times n}$ is a similarity matrix where each element $w_{i,j} = h(z_i, z_j)$, $\forall z_i, z_j \in Z$ is an output of some positively defined kernel h; $A' = A + \gamma W$ is a reweighted adjacency matrix A with a parameter $\gamma \in [0, 1]$ for W; $D'_{i,i} = \sum_{j=1}^n A'_{i,j}$ is a diagonal matrix.

Let us redefine the recurrent formula of PRSSL [1] using Y^* at each training epoch as a replacement of real labels $Y = Y^*$:

$$F^t = \alpha D'^{(-\sigma)} A' D'^{(\sigma-1)} F^{t-1} + (1 - \alpha) Y^*; \tag{1}$$

where $F^t = (y_{i,j}^{pr})_{i,j=1}^{n,c} \in \mathbb{R}^{n \times c}$. Here $y_i^{pr} = (y_{i,1}^{pr}, \ldots, y_{i,c}^{pr})$ is a sample from random variable y^{pr} since (1) is a transformation of the random variable y^* and $F^0 = Y^*$.

Then assume that F^t will improve the accuracy of Y^* by using the information of nodes similarity in latent space during the t-th iterations. We named it the PageRank spreading assumption. Moreover, we propose to use Y^* as a new labels. Let us notice that and $y^* \sim y^{pr}$ due to the PowerIteration PageRank property $||F^t - Y^*||_1 \leq \frac{1}{1-\alpha}||F^1 - Y^*||_1$ [2](Property 12). This allows us consistently use the aforementioned PageRank spreading assumption in training process of the generative model:

$$p(x, y^*, z) \approx p(x|z, y^{pr})p(y^{pr})p(z) \tag{2}$$

where $p(\cdot)$ is a PDF of a random variable.

3.2 Objective function of GenPR

In this subsection we consider the inductive regime of GenPR which allows us to train jointly the generative model (2) and PRSSL (1). Now let us define GenPR objective function. It is obtained by maximizing the variational lower bound of the data log-likelihood of (2) with variance ϕ and generative θ parameters [3]:

$$\begin{aligned} &\log p(x, y^*) \geq \mathbb{E}_{q_\phi(z|x,y^*)}\big[\log p_\theta(x|z, y^{pr})\big] \\ &+ \mathbb{E}_{q_\phi(z|x,y^*)}\big[\log p_\theta(y^{pr})\big] - D_{KL}(p(z)||q_\phi(z|x, y^*)) \end{aligned} \tag{3}$$

where $q_\phi(z|y^*,x) = \mathcal{N}(z|\mu(y^*,x),\sigma^2(x))$ is a multivariate Gaussian distribution parameterized by $\mu(y^*,x)$ and $\sigma(x)$ that are inferred from NN layers for expectation and variance respectively; $p_\theta(x|z,y^{pr}) = f_\theta(z,y^{pr})$ is a nonlinear transformation of z and y^{pr} by NN layer; $p_\theta(y^{pr}) = PR(y^*,\mu(y^*,x),A)$ is a linear transformation of y^* by (1) (the NN layer version will be defined in the next subsection 3.3), $p(z) = \mathcal{N}(z|0,I)$ is a multivariate Gaussian distribution and $D_{KL}(\cdot||\cdot)$ is the Kullback-Leibler divergence.

Since we can trade the quality generation of x for the quality of y_i^{pr} and estimate y_i^{pr} using the information from n_l, we can use $\beta \in [0,1]$ as a weight parameter for $p_\theta(x|z,y^{pr})$ and the categorical crossentropy $\mathcal{U}(F^t,Y) = \sum_{i=1}^{n_l}\sum_{j=1}^{c}(y_{i,j}\cdot \log(y_{i,j}^{pr}))$ for y_i^{pr} estimation. Thus, we obtain from (3) the final inductive (I) GenPR objective function:

$$\begin{aligned}\mathcal{L}(\theta,\phi,x,Y) &= \beta\mathbb{E}_{q_\phi(z|x,y^*)}\big[\log p_\theta(x|z,y^{pr})\big] + \log p_\theta(y^{pr}) \\ &- D_{KL}(p(z)||q_\phi(z|x,y^*)) - \mathcal{U}(F^t,Y)\end{aligned} \tag{4}$$

The difference between inductive (I) and transductive (T) regimes of GenPR is that transductive GenPR does not use the proposition that y^* is a new labels and an objective function looks as follows:

$$\begin{aligned}\mathcal{L}_T(\theta,\phi,x,Y) &= \beta\mathbb{E}_{q_\phi(z|x)}\big[\log p_\theta(x|z)\big] \\ &- D_{KL}(p(z)||q_\phi(z|x)) - \mathcal{U}(F^t,Y)\end{aligned} \tag{5}$$

.

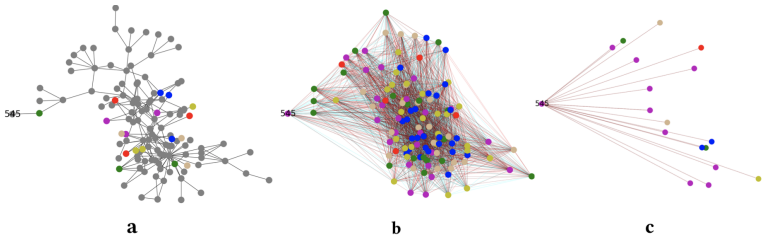

Fig. 1. Sample nodes from Citeseer data set: a - A before GenPR, where colored nodes are labeled and grey are unlabeled, straight black edges are citations between nodes (papers); b - A' after GenPR, where all colored nodes are result from F^t, and color of an edges by weights from A' (cyan is a lower weights, maroon is a higher weights); c - the result of filtering lower weight edges for the node 545.

3.3 Architecture of GenPR

Since we have defined the objective function of GenPR (4) we can explain the GenPR layers architecture. The part of z inference contains the following layers:

$$Y^* = \pi_\theta(X);\ \pi_\theta(X) = h_1(XW_1 + B_1) \tag{6}$$

$$\mu(X, Y^*) = h_\mu(concat(X, Y^*)W_\mu + B_\mu) \tag{7}$$

$$\sigma(X) = h_\sigma(XW_\sigma + B_\sigma) \tag{8}$$

where $h_\cdot$ and $B_\cdot$ are activation functions and biases for NN layers respectively; $W_1 \in \mathbb{R}^{d\times c}$, $W_\mu \in \mathbb{R}^{(d+c)\times d'}$ and $W_\sigma \in \mathbb{R}^{d\times d'}$ are trainable weight matrices of MLP (6), expectation (7) and variance (8) for a NN layer respectively; $(m_i)_{i=1}^n = \mu(X, Y^*)$ is an output of (7) layer with $m_i \in \mathbb{R}^{d'}$; $concat(\cdot,\cdot)$ is a matrix concatenation column-wise.

To avoid the issues with high variance of the gradient estimation of $\mathbb{E}_{q_\phi(z|x,y^*)}\big[\log p_\theta(x|z, y^{pr})\big]$ by Monte Carlo method, we follow [3] in using the reparameterization trick to compute a low-variance gradient estimator for $q_\phi(z|x, y^*)$:

$$q_\phi(z|x, y^*) \sim Z, \quad Z = \mu(X, Y^*) + \sigma(X) \odot \epsilon, \epsilon \sim \mathcal{N}(0, I)$$

where $\odot$ is an element-wise product and ϵ is a random variable.

Now we can define (1) as a sequential sublayers in $PR(Y^*, \mu(Y^*, X), A)$:

1. the reweighting of A:

$$w_{i,j} = h\big(m_i, m_j\big);\ \forall\ w_{i,j} \in W; \tag{9}$$

$$A' = A + \gamma W; \tag{10}$$

   where γ is a parameter of involvement W in reweighting of A within the range $[0, 1]$. Here we compute the similarities between the outputs of (7) because we assume that the expectation of the latent variable z more correctly defines the differences between nodes in latent space.
2. the regularization of A':

$$\hat{A}' = D'^{(-\sigma)}A'D'^{(\sigma-1)};\ D'_{i,i} = \sum_{j=1}^{n} A'_{i,j} \tag{11}$$

   where σ is a parameter for selection of regularization type: $\sigma = 1$ is a Standard Laplacian; $\sigma = 0$ is a PageRank; $\sigma = 1/2$ is a Normalized Laplacian;
3. the redefined PRSSL [1]:

$$F^t = \alpha\hat{A}'F^{t-1} + (1 - \alpha)Y^*;\ t \geq 0; \tag{12}$$

   where $F^0 = Y^*$ (6) and F^t is a result of the t-th iterations, smoothly changing the node labels Y^* during iterations.

The final layer is the reconstruction of nodes (papers) $\hat{X} = f_\theta(Z, F^t)$ where $\hat{X} \in \mathbb{R}^{n\times d}$:

$$f_\theta(Z, F^t) = h_2(concat(Z, F^t)W_2 + B_2) \tag{13}$$

where $W_2 \in \mathbb{R}^{(d'+c)\times d}$, B_2 are weight and bias for x generation $p_\theta(x|z, y^{pr}) = f_\theta(z, y^{pr})$. We can turn to transductive regime of the aforementioned GenPR layers architecture by using modified loss as in (5). The Figure 2 presents the difference between inductive (I) and transductive (T) GenPR architectures.

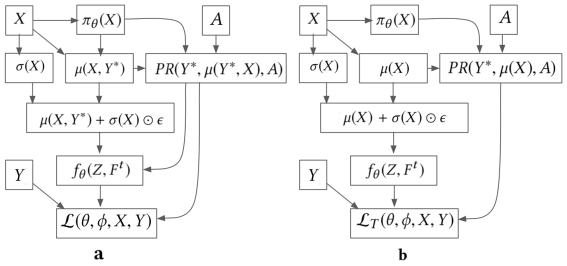

Fig. 2. The I-inductive (a) and T-transductive (b) architectures of GenPR.

4 Experimental setup

For conducting an experiment in the graph-based SSL area we have taken following citation-graph data sets: Citeseer, Cora-ML, Pubmed, MS-Academic (MSA). The description and statistics of data sets available in work APPNP [6]. These data sets are available here[1].

As a baseline algorithms we have considered: the classical diffusion - based linear algorithms (LP [9], PRSSL[1]); the recent graph convolution - based NN (GCN [5], GAT [8], APPNP [6]); the deep generative model (M2 [4]). To avoid overfitting issue we applied L_2 regularization with parameter $\lambda = 0.05$ for weights $W_\cdot$, dropout for $\hat{A}'$ with rate $dr = 0.5$ at each PowerIteration step and learning rate $l = 0.001$ for Adam optimizer. Moreover, we have used the random train-test-validation splitting strategy described in [6] and repeated experiments on each data set 500 times. For a fair model comparison we have made an architecture and parameters of GenPR that are very close to APPNP and GCN. In particular, for all data sets use the intermediate embedding layer $f_0(X) = relu(XW_0 + B_0)$ with $W_0 \in \mathbb{R}^{d \times \hat{d}}$ as the input for (6) with $\hat{d} = 64$, $W_1 \in \mathbb{R}^{\hat{d} \times c}$ and $h_1(\cdot) = softmax(\cdot)$, $d' = 64$ in (7) and (8), $\sigma = 0.5$ and $t = 4$ in (11), $B_\cdot = 0$. In (12) for MSA $\alpha = 0.8$, for Cora-ML, Pubmed and Citeseer $\alpha = 0.9$.

We have selected the specific parameters of GenPR by the 5 fold cross-validation grid search[2]. For all data sets use $h(m_i, m_j) = (m_i^T m_j)^3$ in (9) and $\beta = 0.001$ in (4) and (5). In particular, we have used: for Citeseer: $\gamma = 1$ in (10), $h_\mu(\cdot) = h_\sigma(\cdot) = relu(\cdot)$ in (7) and (8), $h_2(\cdot) = sigmoid(\cdot)$ in (13); for Cora-ML, Pubmed and MSA: $\gamma = 0.001$ in (10), and $h_\cdot(\cdot) = linear(\cdot)$.

5 Experimental results

Table 1 presents performance of the classification based on the default adjacency matrix A or on the node features X leads to loss of classification quality because

[1] https://github.com/klicperajo/ppnp/tree/master/ppnp/data
[2] https://scikit-learn.org/stable/modules/grid_search.html

Table 1. Average accuracy (%) on citation graphs. △ and ▲ denote the statistical significance (t-test) of GenPR for $p < 0.05$ and $p < 0.01$, respectively, compared to the APPNP.

Input	Data set	Citeseer	Cora-ML	Pubmed	MSA
A	PRSSL	71.21	78.12	72.51	76.12
	LP	45.32	68.31	63.12	65.32
X	M2	70.81	79.22	77.6	86.12
X,A	APPNP	75.74	85.09	79.71	93.28
	GAT	75.43	84.41	77.73	91.18
	GCN	75.31	83.52	78.65	92.09
	GenPR (I)	**77.18** ▲	85.52 △	80.09 △	**94.08** ▲
	GenPR (T)	76.91△	**86.19** ▲	**81.13** ▲	93.81 △

we do not use all available information. In the case of the combination of X and A, GenPR significantly and consistently outperforms the others due to the intuition that default A contains incomplete information about nodes similarity. Since we have reached the best results with GenPR (I) and $\gamma = 1$ for Citeseer, it means that latent information is helpful for reweighting default adjacency matrix A (citation graph). In particular, Figure 1 (c) shows that GenPR can be used for the explanation of classification results, by filter the edges by weight and observe nodes with more influence on considered one (e.g. node 545).

Figure 3 shows the GenPR outperforms the APPNP not only in terms of accuracy, but also in number of PowerIteration steps, because GenPR takes less steps to converge for better accuracy than APPNP. Moreover, the GenPR is less complex than APPNP because it uses just one layer for MLP rather than 2 in APPNP.

6 Conclusion

In this work, we propose a graph-based SSL (I)/(T) framework created by embedding PRSSL in generative model. Based on the experimental results, we show that the generative model application for PRSSL can be used not only for the label spreading improvement, but also for interpretation of the classification results. We also show that GenPR significantly and consistently outperforms all other algorithms on every data set and requires less number of PageRank PowerIteration steps. Since GenPR produces complete weighted graph defined by A' we can use PageRank properties to split A' into batches that are complete subgraphs, which opens an opportunity to explore a distributed version of GenPR. The other interesting direction to investigate is an application of GenPR on data sets without default graph structure (e.g. images).

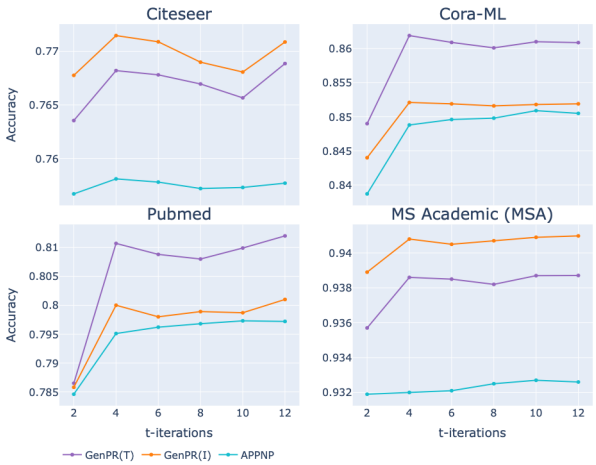

Fig. 3. Average accuracy of GenPR (I) inductive, GenPR (T) transductive and APPNP over the t-iteration steps.

References

1. Avrachenkov, K., Mishenin, A., Gonçalves, P., Sokol, M.: Generalized optimization framework for graph-based semi-supervised learning. In: Proceedings of the 2012 SIAM International Conference on Data Mining. pp. 966–974. SIAM (2012)
2. Brezinski, C., Redivo-Zaglia, M.: The pagerank vector: properties, computation, approximation, and acceleration. SIAM Journal on Matrix Analysis and Applications **28**(2), 551–575 (2006)
3. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Proceedings of the 2nd International Conference on Learning Representations. ICLR (2013)
4. Kingma, D.P., Mohamed, S., Rezende, D.J., Welling, M.: Semi-supervised learning with deep generative models. In: Advances in neural information processing systems. pp. 3581–3589 (2014)
5. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations. ICLR (2017)
6. Klicpera, J., Bojchevski, A., Günnemann, S.: Predict then propagate: Graph neural networks meet personalized pagerank. In: International Conference on Learning Representations. ICLR (2019)
7. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Tech. rep., Stanford InfoLab (1999)
8. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. In: 6th International Conference on Learning Representations. ICLR (2018)
9. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. CMU Technical report (2002)