A Robustness Analysis of Personalized Propagation of Neural Prediction

Changkyu Choi, Michael Kampffmeyer, and Robert Jenssen

UiT The Arctic University of Norway

1 Introduction

Data without annotation are easy to obtain in the real-world, however, established supervised learning methods are not applicable to analyze them. Several learning approaches have been proposed in recent years to exploit the underlying structure of the data without requiring annotations [1, 2]. Semi-supervised learning aims to improve the predictive performance of these unsupervised approaches, by exploiting partially acquired annotations in the dataset. One recent promising line of work in this scheme makes use of graph neural networks (GNN) [3]. The data is expressed as a graph, where vertices are data samples and edges, given by an adjacency matrix $\mathbf{A}$, represent pairwise relationships between data points. Although these approaches achieve promising performance, they have so far been limited to applications, where the graph, in form of the adjacency matrix, is available. This is a severe limitation, as most available datasets do not include a predefined graph structure. To address this shortcoming, we investigate if the adjacency matrix $\mathbf{A}$ can be replaced with affinity matrices obtained directly from the data. As a first step into this direction, and in order to analyze its potential, we provide an analysis of how the current state-of-the-art semi-supervised approach, Personalized Propagation of Neural Predictions(PPNP)[4], is affected by changes in the affinity matrix.

2 Background

A popular concept of exploiting structured datasets is neighborhood aggregation, where large node neighborhoods are combined to achieve a more comprehensive representation. However, this often tends to cause over-smoothing and leads to a loss of the local structure in the neighborhood as the neighborhood size increases [5, 4]. To improve the over-smoothing issue commonly found in previous graph-based approaches, Klicpera et al.[4] suggest PPNP by adopting an idea from Personalized PageRank(PPR)[6]. Their model is given as,

$$\begin{aligned} \mathbf{Z^{(0)}} &= \mathbf{H} = f_\theta(\mathbf{X}) \\ \mathbf{Z^{(k+1)}} &= (1-\alpha)\hat{\tilde{\mathbf{A}}}\mathbf{Z^{(k)}} + \alpha\mathbf{H} \\ \mathbf{Z^{(K)}} &= \sigma\left((1-\alpha)\hat{\tilde{\mathbf{A}}}\mathbf{Z^{(K-1)}} + \alpha\mathbf{H}\right) \end{aligned} \tag{1}$$

where f_θ denotes a neural network, $\mathbf{H} = \left\{\mathbf{h}_i\right\}_{i=1}^{N}$ is the network prediction, α is the teleport probablity, σ is the softmax, and $\mathbf{X} = \left\{\mathbf{x}_i\right\}_{i=1}^{N}$ is an input feature matrix, where each data point is represented as a vertex in the graph. The adjacency matrix $\mathbf{A} = \left\{a_{ij} \in \left\{0,1\right\}\right\}_{i,j=1}^{N}$ represents the pairwise relationship of the points in $\mathbf{X}$ and $\hat{\tilde{\mathbf{A}}} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$, where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ and $\tilde{\mathbf{D}}_{ii} = \Sigma_j \tilde{\mathbf{A}}_{ij}$. The main idea of PPR is to assign the restart state $\mathbf{h}_i$ for the node i and to aggregate neighborhood using the matrix $\hat{\tilde{\mathbf{A}}}$ with restart at any random propagation layer k. In addition, shown in Eq. (1), approximated PPNP has a separate two-step architecture with individual functionality; **(a)** neural network $f_\theta(\mathbf{X})$ which is related to the learning procedure; and **(b)** a K-layer propagation stack which exploits $\mathbf{A}$.

3 Methodology

For the robustness analysis of the PPNP framework, we define an ideal affinity matrix $\mathbf{A}_{ide}$ and analyze the effect of reducing the quality of the affinity matrix. We do this by replacing $\mathbf{A}$ in the framework with degenerative versions of $\mathbf{A}_{ide}$. This analysis aims to observe the change in accuracy with respect to the degree of degeneration.

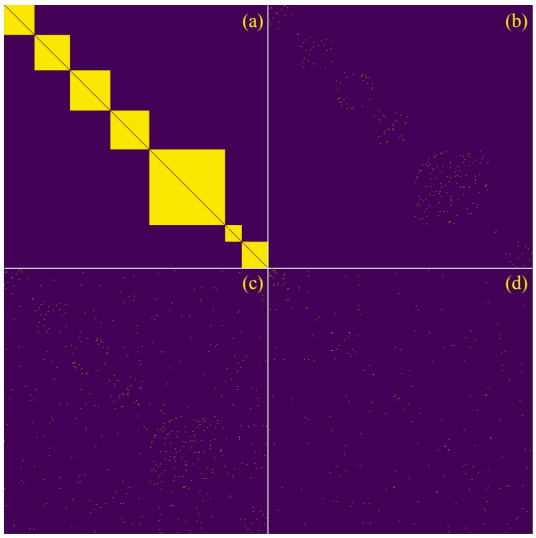

Figure 1: Degenerative versions of sorted $\mathbf{A}_{ide}$ of Cora-ML dataset, where the number of the classes $C = 7$, and the sample size in each class is $[354, 402, 452, 442, 857, 193, 295]$ (a) $\beta = \gamma = 0.00$ (b) $\beta = 0.99, \gamma = 0$ (c) $\beta = 0.99, \gamma = 2.32e^{-3}$ (d) observed/given graph $\mathbf{A}_{obs}$. Best viewed in electronic format (zoomed in).

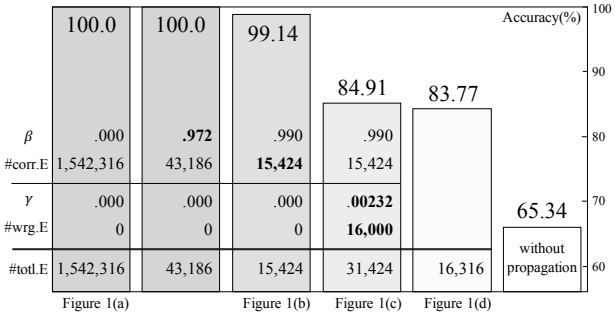

Figure 2: Result of the robustness analysis. Each bar-graph represents the accuracy from different affinity matrix.

Edge-reducing and Edge-activating Probabilities, β and γ: Let $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{N}$ be a set of one-hot encoded label information for classification. We define the ideal matrix $\mathbf{A}_{ide} = \mathbf{Y}\mathbf{Y}^T - \mathbf{I}_N$.

Note that the ideal graph consists of several subgraphs where each of them represents one class as shown in Fig. 1(a). All nodes in a subgraph are fully connected to each other, meaning that for a node all other nodes in the class are the one-hop neighborhood. Meanwhile, nodes between different classes are disconnected.

Two variables, β and γ, are introduced to degrade the ideal graph. $\beta = \frac{r}{M}$ and $\gamma = \frac{t}{N_2 - N - M}$, where M is the total number of edges in $\mathbf{A}_{ide}$, r corresponds to the number of reduced edges ($0 \leq r \leq M$), and t is the number of activated edges ($0 \leq t \leq N^2 - N - M$). Edge-reducing probability β implies the removal of the edges in the graph of $\mathbf{A}_{ide}$ as shown in Fig. 1(b). It destroys the structure within a subgraph but, on the other hand, makes the matrix sparse and may increase the efficiency.

4 Analysis and Insight

Edge-activating probability γ implies the addition of edges between nodes in different classes. It causes the matrix do become more dense and noisy (see Fig. 1(c)).

To enable comparisons, hyperparameters match the original PPNP paper [4], including $\alpha = 0.1$ and $K = 10$ in Eq. (1). The Cora-ML benchmark dataset [3] is chosen for the analysis. Input feature matrix $\mathbf{X}$ has $N = 2,905$ datapoints with $D = 2,819$ features each and the observed adjacency matrix $\mathbf{A}_{obs}$ has 16,316 edges. By varying the β and γ parameters and performing extensive experiments, we observe among others, that perfect accuracy can be achieved even if 97.2% of the edges in $\mathbf{A}_{ide}$ are removed ($\beta = 0.972$) by reducing the "ideal" number of edges $(1,542,316)$ to $43,186$ (see Figure 2). This is intuitive, as removing edges at random, still leaves the individual classes connected unless the graph is thinned too much. As long as there is a path that connects all nodes in the same class 100.0 % can be obtained. At the same time, this thinning reduces inference time approximately 15%. Further, the accuracy decreases as the number of wrong edges increase. Interestingly, thinning the graph to a similar size as the original Cora-ML dataset (by choosing $\beta = 0.99$) and doubling the number of edges by adding wrong edges ($\gamma = 0.0023$) still gives a performance of 84.91%. This is still more than the reported accuracy obtained by the PPNP approach, which is 83.77%.

5 Conclusion

We have analysed the state-of-the-art semi-supervised learning approach PPNP and provided insights into its robustness to the graph structure. This is done by replacing the adjacency matrix with degenerative versions of the ideal matrix $\mathbf{A}_{ide}$. In

future work, we will extend this framework to semi-supervised problems without adjacency matrix to learn the network representations and the affinity matrix simultaneously.

References

[1] B. Huang, K. Zhang, P. Xie, M. Gong, E. P. Xing, and C. Glymour. Specific and shared causal relation modeling and mechanism-based clustering. *NeurIPS*, 2019.

[2] M. Kampffmeyer, S. Løkse, F. M. Bianchi, L. Livi, A.-B. Salberg, and R. Jenssen. Deep divergence-based approach to clustering. *Neural Networks*, 113:91–101, 2019.

[3] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *ICLR*, 2017.

[4] J. Klicpera, A. Bojchevski, and S. Günnemann. Predict then propagate: Graph neural networks meet personalized PageRank. *ICLR*, 2019.

[5] Q. Li, Z. Han, and X.-M. Wu. Deeper insights into graph convolutional networks for semi-supervised learning. *AAAI*, 2018.

[6] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.