Abstract

Recently, research on analyzing networks with deep learning has received widespread attention. In particular, graph convolutional networks (GCNs) (Defferrard et al., 2016; Kipf and Welling 2017), which obtain node embeddings through the propagation and aggregation of the features on network topology, have achieved great success. While the success of GCNs and their variants, a key issue with them is that the accuracy of multi-typed features classification varies greatly. To address this problem, (Yang et al., 2015) proposed text-associated DeepWalk (TADW), which incorporates text features of vertices into network representation learning under the framework of matrix factorization. But the model can only handle the text attributes. (Cui et al., 2020) presented an adaptive graph encoder (AGE), a novel attributed graph embedding framework which applied a carefully-designed Laplacian smoothing filter. Nevertheless, all the above methods are designed to handle the single-typed feature and the node features only serve as an initial solution of embeddings.

GCNs obtain node features from a local mixing state of propagation by limiting the number of propagations to two or three layers. However, this will further make GCNs rely heavily on the local homophily of topology, that is, neighborhoods should be similar, a very strong assumption in many real-world text-rich networks. A lot of methods are developed to handle the topological limitations of GCNs. For example, several studies attempted to utilize the self-supervised learning methods (Zhu et al., 2020; You et al., 2020) which use several highly credible labels derived from GCNs to optimize the topological channels in the following propagation of GCNs. These existing methods have achieved reasonable results at handling the topological limitations of GCNs and thus improved the performance of GCNs. However, from the level of model architectures, an ideal way

1 Introduction

Content-rich networks are graphs with node features and network structures widely applied in academic citation networks, recommendation systems, etc. However, because of the complex non-Euclidean graph structure, capturing structure and feature information is a challenging task on machine learning approaches.
may be that the convolutions of features (on topology) and topology (on features) play together in the same system. By jointly training the BERT and GCN modules within Bert-GCN (Lin et al., 2021), this model is able to leverage the advantages of both worlds: large-scale pretraining which takes called text-rich networks, and have been the advantage of the massive amount of raw data extensively studied (Li et al., 2017; Zhang et al., 2018; Zhou et al., 2018; Veličković et al. 2019; Meng et al. 2019). Their goal is to preserve not only the network structure, but also the node attribute and transductive learning. However, the model was trained with the BERT feature of node text and it topological structure features. The training process of attributed network embedding lies in simultaneously capturing node attributes, network preserving the information of multi-typed features, and the relation-ship into hidden network topologies, and node classification representations. Our work is inspired by the work of using graph neural networks to fusion node information by mining the first-order features (Zhang et al., 2020). Existing works that second-order proximities between nodes; the combine BERT and GNNs uses graph to model relationships between tokens within a single document sample (Lu et al., 2020), which fall into patterns. Also, our model combines different node features at an evidence level, which produces a stable and reasonable uncertainty estimation. Figure 1 shows the illustration of our implementation of EGCN. The main contributions of this work are summarized as follows.

(1) We propose a unified deep model (EGCN) to learn the embedding vector for each node of the network by considering both multi-typed features and the graph semantic relationships simultaneously.

(2) We develop a novel multi-typed features classification method aiming to provide trusted and interpretable decisions in an effective and efficient way.

(3) We run extensive experiments which validate the superior accuracy and robustness of our model thanks to the promising uncertainty estimation and multi-typed features integration strategy.

2 Related Work

2.1 Enriching Graph Embeddings with External Text

Graph convolutional networks (GCN) is connectionist models that fusion dependencies and

2.2 Uncertainty-based Learning

The history of learning uncertainty-aware predictors is concurrent with the advent of modern Bayesian approaches to machine learning. Bayesian neural networks (BNNs) (Neal, 2012) endow deep models with uncertainty by replacing the deterministic weight parameters with distributions. Because BNNs need to consume a lot of computing power when performing inference calculations, a more stable and effective method, MC-dropout (Gal et al., 2016), was proposed. The inference calculation in this model is done by dropout sampling from the training and test weights. Ensemble based methods (Lakshminarayanan et al., 2017) train and integrate multiple deep networks and also achieve promising performance. Instead of indirectly modeling uncertainty through net-work weights, the algorithm (Sensoy et al., 2018) introduces the subjective logic theory to directly model uncertainty without ensemble or Monte Carlo
s with me time, the text features of encoders such as BERT or SimCSE are all available evidence.

\[ Z^{(t)} = \phi \left( \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} Z^{(t-1)} W^{(t-1)} \right) \] (1)

where \( \tilde{A} = A + I \), \( \phi(.) \) is an activation function such as ReLU. As can be seen from Eq. 1, the representation \( Z^{(t-1)} \) will propagate through the normalized adjacency matrix \( \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} \) to obtain the new representation \( Z^{(t)} \). Considering that the representation learned by BERT or SimCSE, we combine the two representations \( Z^{(t-1)} \) and \( H \) together to get a more complete and powerful representation as follows:

\[ Z^{(t-1)} = (1 - \epsilon)Z^{(t-1)} + \epsilon H \] (2)

where \( \epsilon \) is a balance coefficient, \( H \) represents text feature.

3 EGCN Model

3.1 Problem Statement

A text-rich network is a network \( G = \{V, R, X\} \), where \( R \) is the set of relations, \( V \) is the set of nodes, \( X \) is a matrix that encodes node attributes information for \( n \) nodes. Given an attributed network \( G \) and the set of adjacency matrices \( A \), the task of structure and features fusion with EGCN for node classification is to learn the multi-type features, and determine the confidence of each type of feature for the node classification.

Figure 1 Visual illustration of our implementation of EGCN. Text-rich network obtains the graph structure information through GCN module, and at the same time, the text features of encoders such as BERT or SimCSE are fused with GCN with Eq. 2. The text features such as TF-IDF, BERT and SimCSE and the features output by the GCN module are mapped using Dirichlet distribution to calculate the confidence and uncertainty of each feature through Uncertainty and the Theory of Evidence, combining them with Dempster–Shafer theory, and obtain the confidence and classification uncertainty after all the features are combined.

3.3 Uncertainty and the Theory of Evidence

Bayesian theory to subjective probabilities.

The Dempster–Shafer Theory of Evidence (DST) formalizes DST’s notion of belief assignments over a frame of discernment as a theory on belief functions, was first proposed a Dirichlet Distribution. Hence, we introduce a principle of evidential theory-based uncertainty.
In this paper, we design and train neural networks to form multi-view opinions for the classification of a given sample $i$ as a Dirichlet distribution $D(p|\alpha)$, where $p_i$ is a simple representing class assignment probabilities.

### 3.4 Dempster’s Rule of Combination for Multi-typed features Classification

Having introduced evidence and uncertainty for the single-feature case, we use the Dempster–Shafer theory of evidence to combine multi-typed features arriving at a degree of belief (represented by a mathematical object called the belief function) that focus on all the available evidence (see Figure 2). Specifically, we combine $V$ independent sets of probability mass assignments $\{M^v\}_v$, where $M^v = \{b^v_k, u^v\}$, $b$ refers to the confidence probability, $v$ is the feature type, $k$ is the node category, and $u^v$ is the uncertainty of the node classification for the feature $v$. The combined calculation of confidence and uncertainty among multiple types of features is as follows:

$$M = M^1 \oplus M^2 \oplus \cdots \oplus M^V$$

The more specific calculation rule can be formulated as follows:

$$b_k = \frac{1}{1-c} (b_k b_k^2 + b_k^1 u^1 + b_k^2 u^2), u = \frac{1}{1-c} u^1 u^2$$

where $C = \sum_{i \neq j} b_i b_j^2$ is a measure of the amount of conflict between the two mass sets, and the scale factor $1-C$ is used for normalization.

### 3.5 Learning to Form Opinions

The loss over a batch of training samples can be computed by summing the loss for each sample in the batch. During training, the model mine patterns in the node text and generate evidence for specific class labels based on these patterns to minimize the overall loss. For our model, given the evidence of the $i$-th sample obtained through the evidence network, we can get the parameter $\alpha_i$ (i.e., $\alpha_i = \alpha^i + 1$) of the Dirichlet distribution and form the multinomial opinions $D(p_i|\alpha_i)$, we can treat $D(p_i|\alpha_i)$ as a prior on the likelihood $\text{Mult}(y|p_i)$ and obtain the negated logarithm of the marginal likelihood by integrating out the class probabilities

$$L_{ace}(\alpha_i) = \int \sum_{j=1}^K \psi(\psi^{-1}(\alpha_i) - \log(p_{ij})) \frac{1}{\beta(\alpha_i)} \prod_{j=1}^K p_{ij}^{\alpha_i-1} dp_i = \sum_{j=1}^K \psi(S_j) - \psi(\alpha_i)$$

where $\psi(\cdot)$ is the digamma function.

Eq. 9 is the integral of the cross-entropy loss for classes, however, it cannot guarantee that less function on the simplex determined by $\alpha_i$. The above loss function ensures that the correct label of each sample generates more evidence than other class labels. That is to say, in our model, we expect the evidence...
for incorrect labels to shrink to 0. To this end, the following KL divergence term is introduced:

$$KL[D(p_i | \tilde{a}_i) \parallel D(p_i | 1)] =$$

$$\log \left( \frac{r_{\alpha K}(\tilde{a}_i)}{r(\tilde{a}_i) \prod_{k=1}^{K} r(\tilde{a}_{ik})} \right) +$$

$$\sum_{k=1}^{K} (\tilde{a}_{ik} - 1) \left[ \psi(\tilde{a}_{ik}) - \psi \left( \sum_{j=1}^{K} \tilde{a}_{ij} \right) \right]$$

where \( \tilde{\alpha} \equiv y_i + (1 - y_i) \hat{\alpha}_i \) is the adjusted parameter of the Dirichlet distribution which can avoid penalizing the evidence of the ground-truth class to 0, and \( \Gamma(\cdot) \) is the gamma function.

Let us consider that a Dirichlet distribution with zero total evidence, i.e., \( S = K \), corresponds to the uniform distribution and indicates total uncertainty, i.e., \( u = 1 \). We achieve this by incorporating a Kullback-Leibler(KL) divergence term into our loss function that regularizes our predictive distribution by penalizing those divergences from the "I do not know" state that do not contribute to data fit. The loss with this regularizing term reads:

$$L(\alpha_i) = L_{\text{ac}}(\alpha_i) + \lambda_i KL[D(p_i \mid \tilde{a}_i) \parallel D(p_i \mid 1)]$$

where \( \lambda_i = \min(1.0, t/10) \in [0, 1] \) is the annealing coefficient, \( t \) is the index of the current training epoch, \( D(p_i \mid h_1, \ldots, h_i) \) is the uniform Dirichlet distribution, to prevent the network from paying too much attention to the KL divergence in the initial stage of training. To ensure that all views can simultaneously form reasonable opinions and thus improve the overall opinion, we use a multi-task strategy with following overall loss function:

$$L_{\text{overall}} = \sum_{i=1}^{N} [L(\alpha_i) + \sum_{v=1}^{V} L(\alpha^v_i)]$$

Figure 2 Illustration of trusted multi-typed features classification. The evidence of each feature is obtained using BERT, TF-IDF, SimCSE and GCN in Figure 1. The obtained evidence parameterizes the Dirichlet distribution to induce the classification probability and uncertainty. The overall uncertainty and classification probability are inferred by combining the beliefs of multiple views based on the DST. The combination rule and an example are shown in Eq.7 and Eq. 8, respectively.

4 Experiments

In this section, we run experiments on three real-world da-tasets: Cora, Citeseer and DBLP. We compare EGCN to the following models: BERT (Devlin et al. 2018), SimCSE (Gao et al. 2021), GCN (Kipf et al. 2017), GAT (Veličković et al. 2017), GraphSage (Hamilton et al. 2017), TADW (Yang et al. 2015), BertGCN (Lin et al. 2021) to demonstrate the effectiveness of proposed model. We also prove EGCN model can produce trusted classification decisions on different types of attributed information.

4.1 Datasets

Cora data is an open citation network data set containing 7 types of papers. The network contains 2211 paper nodes and 5214 citation relationships. Each paper contains an average of 169 words, and the vocabulary of the entire data set contains a total of 12619 words. The Citeseer data set consists of papers from 10 interdisciplinary research fields, it contains 4610 nodes and 5923 edges. The DBLP data set is a comprehensive data set covering 4 types of papers, the network contains 13,404 nodes and 39861 edges.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Cora</th>
<th>Citeseer</th>
<th>DBLP</th>
</tr>
</thead>
<tbody>
<tr>
<td># Nodes</td>
<td>2211</td>
<td>4610</td>
<td>13404</td>
</tr>
<tr>
<td># Edge</td>
<td>5214</td>
<td>5923</td>
<td>39861</td>
</tr>
<tr>
<td># Text</td>
<td>169</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td># Classes</td>
<td>7</td>
<td>10</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 1 Dataset statistics

Table 1 illustrates the details of datasets used in our experiment. #Text denotes the average number of words contained in each text node.

4.2 Experiment Setups

For all methods using the BERT model, we use BERT-base architecture with pre-trained weights from the original authors and adapted by
This investigation demonstrates that features and graph structure contribute to classification from different perspectives.

Figure 3 shows the loss (a) and the corresponding accuracy (b) of the model training process on the three data sets. It can be seen from the figure that due to the large amount of DBLP data, the loss and accuracy curve is longer. At the same time, (b) reflects the convergence of our model and shows higher accuracy. For the accuracy curve of (b), there is a jitter phenomenon, which can be explained by (d) and (h) in Figure 4. The classification error is caused by the conflict of multiple types of features.

### 4.4 Ablation study

Table 3 shows the classification results of text features. For the three data sets, learn feature expressions of the node's TF-IDF, SimCSE, and BERT. BERT768 refers to the output dimension is 768, and BERT512 and BERT256 refer to the slave first 512-dimensional and 256-dimensional features cropped from the 768-dimensional features. From Table 3, it can be seen that for node text classification, the TF-IDF feature performs the best for the classification results of the three data sets.

<table>
<thead>
<tr>
<th>Models</th>
<th>MLP</th>
<th>Cora</th>
<th>Citeseer</th>
<th>DBLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-IDF</td>
<td>0.85</td>
<td>0.84</td>
<td>0.79</td>
<td></td>
</tr>
<tr>
<td>SimCSE</td>
<td>0.71</td>
<td>0.67</td>
<td>0.67</td>
<td></td>
</tr>
<tr>
<td>Bert768</td>
<td>0.55</td>
<td>0.6</td>
<td>0.63</td>
<td></td>
</tr>
<tr>
<td>Bert512</td>
<td>0.49</td>
<td>0.41</td>
<td>0.58</td>
<td></td>
</tr>
<tr>
<td>Bert256</td>
<td>0.29</td>
<td>0.33</td>
<td>0.5</td>
<td></td>
</tr>
</tbody>
</table>

Table 3 Experimental results of node text multi-type features classification.

$$\text{Table 4 Experimental results of classification of GCN nodes with multiple types of features}$$

<table>
<thead>
<tr>
<th>Models</th>
<th>MLP</th>
<th>Cora</th>
<th>Citeseer</th>
<th>DBLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-IDF</td>
<td>0.8381</td>
<td>0.9397</td>
<td>0.8715</td>
<td></td>
</tr>
<tr>
<td>SimCSE</td>
<td>0.6524</td>
<td>0.7397</td>
<td>0.6044</td>
<td></td>
</tr>
<tr>
<td>Bert768</td>
<td>0.7841</td>
<td>0.7511</td>
<td>0.5288</td>
<td></td>
</tr>
<tr>
<td>Bert512</td>
<td>0.7984</td>
<td>0.8519</td>
<td>0.8265</td>
<td></td>
</tr>
<tr>
<td>Bert256</td>
<td>0.7714</td>
<td>0.8153</td>
<td>0.7892</td>
<td></td>
</tr>
</tbody>
</table>

Table 4 Experimental results of classification of GCN nodes with multiple types of features.
expressed by SimCSE, the node classification accuracy obtained a big improvement.

Table 5 shows that different features are linearly fused when GCN is used for feature aggregation. \( \lambda = 1 \) means the node feature classification using GCN, and \( \lambda = 0 \) means the node text feature classification is used. As can be seen from the table, for the three data Collection, linear fusion between multiple features, the output features are classified, and the accuracy is greatly improved.

<table>
<thead>
<tr>
<th>( \lambda \text{GCN}+(1-\lambda) )</th>
<th>Cora</th>
<th>Citeseer</th>
<th>DBLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF-IDF</td>
<td>0.8048</td>
<td>0.8785</td>
<td>0.8436</td>
</tr>
<tr>
<td>SimCSE</td>
<td>0.7952</td>
<td>0.8715</td>
<td>0.8407</td>
</tr>
<tr>
<td>Bert768</td>
<td>0.8126</td>
<td>0.8708</td>
<td>0.8467</td>
</tr>
<tr>
<td>Bert512</td>
<td>0.7952</td>
<td>0.8769</td>
<td>0.8441</td>
</tr>
<tr>
<td>Bert256</td>
<td>0.7984</td>
<td>0.8769</td>
<td>0.8449</td>
</tr>
</tbody>
</table>

Table 5 Experimental results of node classification of multi-type features with linear fusion of text features and structural features

Figure 4 Illustration of node multi-type features classification confidence

For the progressive experimental results in Tables 3, 4, 5 and Table 2, we have the following observations:

1. The basic observation is that our proposed EGCN framework achieves better results on three datasets compared with baseline methods and variants, expressing that this type of feature is confident in the seven classification results, and the dark colors in the first, third, and fourth columns cause confidence modeling node features and network topology. By comparing EGCN with baseline methods, we can further infer the advantage of aggregating the multiple features of nodes and structures to classify the confidence of nodes.

2. From node text feature classification, text feature input to GCN node classification, text feature and GCN structure feature linear aggregation node feature classification to text feature, text feature and graph structure feature multi-type features for category confidence classification, classification The accuracy continues to improve, showing the effective fusion of multiple types of features and the experimental verification of the confidence of each type of feature on the classification from different types of features.

In Figure 4, for the four samples of the Cora dataset, the rows in Figures (a) (b) (c) (d) represent six different types of features (including the text features and structural features of the graph nodes), and the list is up to 7 classification types, color intensity codes, the confidence of each feature for the 7 types of samples, the darker the color, the more confidence, the sixth column of Figure (a) has a darker color, expressing the confidence of multiple types of features for the sixth category. Figure (b) is a darker color in the first column. Express the confidence of multiple types of features for the first type. The colors in the first and third columns in Figure (c) are darker, but the color depth of the third type is evenly distributed. The experimental results show that it is classified as the

5 Conclusion

In this work, we propose a novel trusted multi-typed features graph node classification (EGCN) model which, based on the Dempster-Shafer evidence theory, can produce trusted classification decisions on multi-typed features and can jointly learn low-dimensional representations of both nodes and features for text-rich networks. Our algorithm focuses on decision-making by fusing the uncertainty of multi-typed features, which is essential for making trusted decisions. Furthermore, our model can produce the uncertainty of a current decision while making the final classification, providing interpretability. The empirical results validate the effectiveness of the proposed algorithm in classification accuracy.
References

Kipf, T. N.; Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In ICLR.


Yang, C.; Liu, Z.; Zhao; D.; Sun, M.; Chang, E. 2015. Network representation learning with rich text information. In Twenty-fourth international joint conference on artificial intelligence.


Han, Z., Zhang, C., Fu, H., & Zhou, J. T. 2021. Trusted Multi-View Classification. In ICLR.


