Structure and Features Fusion with Evidential Graph Convolutional Neural Network for Node Classification

Anonymous ACL submission

Abstract

Recently, text-enhanced network representation learning has achieved great success by taking advantage of rich text information and network structure information. However, content-rich network representation learning and quantifying classification uncertainty are challenging when it comes to integrating complex structural dependencies and rich content features at an evidence level. In this paper, we propose an evidential graph representation learning model (EGCN), which can not only fuse network structure and content information into a more complete and powerful representation for each node, but also assess the quality of graph node features to improve classification accuracy. To achieve better fusion, we integrate the node's features representation into structure-aware representation through a delivery operator. Besides, to overcome the difficulty of predicting node classification confidence, we employ a novel module based on Dirichlet distribution theory of evidence and subject opinion learning to collect the evidence of the class probabilities. Experimental results on three real-world networks show that our model can improve both node classification accuracy and robustness as compared to all baselines.

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.

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 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 the advantage of the massive amount of raw data and transductive learning. However, the model was trained with the BERT feature of node text and it cannot utilize multi-typed features like term frequency-inverse document frequency (TF-IDF) or SimCSE(Gao et al., 2021).

To the best of our knowledge, no work has been devoted to exploring both the multi-typed features and the semantic graph relationships in an efficient way. In this paper, we develop a unified deep model (EGCN) to capture both text-rich information and topology structure features. The training process of EGCN consists of three parts which focus on preserving the information of multi-typed features, network topologies, and node classification confidence, respectively. The graph structure information is mined by modeling the first-order and second-order proximities between nodes; the text features are processed with TF-IDF, BERT, and SimCSE methods to capture the different pattens. 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 relations between graph nodes (Hamilton et al., 2017; Keyulu et al., 2018). Besides structural properties, nodes in a graph are often affiliated with various contents, such as abstract or title text in the academic citation network. Such networks are called text-rich networks, and have been extensively studied (Li et al., 2017; Zhang et al., 2018; Zhou et al., 2018; Veliˇckovic´ et al. 2019; Meng et al. 2019). Their goal is to preserve not only the network structure, but also the node attribute proximity in learning representations. Recently, much efforts have been made to gain insights from attributed networks (Liao et al., 2018; Yang et al., 2018; Li et al., 2017). Some approaches (Huang et al., 2017; Xiao et al., 2017) simply take the label information into consideration, while others utilize more detailed attribute information. The key point of attributed network embedding lies in simultaneously capturing node attributes, network structure and their relation-ship into hidden representations. Our work is inspired by the work of using graph neural networks to fusion node features (Zhang et al., 2020). Existing works that combine BERT and GNNs uses graph to model relationships between tokens within a single document sample (Lu et al., 2020), which fall into the category of inductive learning. But different from these works, we focus on combining, and show that multi-typed features can significantly benefit from uncertainty-based learning model.

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 et al., 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 sampling. Building upon RBF networks, the distance between test samples and prototypes can be used as the agency for deterministic uncertainty (van Amersfoort et al., 2020). Benefiting from the learned weights of different tasks with homoscedastic un-certainty learning, (Kendall et al., 2018) achieves impressive performance in multi-task learning. (Han et al. 2021) utilized multiple views to promote both classification relia-bility and robustness by integrating evidence from each view. Dempster-Shafer Evidence Theory is about the theoretical method of the confidence function, which directly models uncertainty. DST allows beliefs from different sources to be combined with various fusion operators to obtain a new belief that considers all available evidence (Jøsang et al., 2012).

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.

3.2 GCN Module

The pre-training models can obtain the expression of the semantic information of the text. In the section, we will introduce how to use the GCN module to propagate these representations generated by the pre-training models. Once all the multi-typed representations learned by BERT and SimCSE are integrated into GCN, then the output feature of GCN will be able to accommodate for two different kinds of information, i.e., data itself and relationship. In particular, with the weight matrix W, the representation learned by the ℓ-th layer of GCN, $Z(\ell)$, can be obtained by the following convolutional operation:

$$Z^{(\ell)} = \phi\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}Z^{(\ell-1)}W^{(\ell-1)}\right) \quad (1)$$

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

$$\tilde{Z}^{(\ell-1)} = (1-\epsilon)Z^{(\ell-1)} + \epsilon H \quad (2)$$

where ϵ is a balance coefficient, H represents text feature.

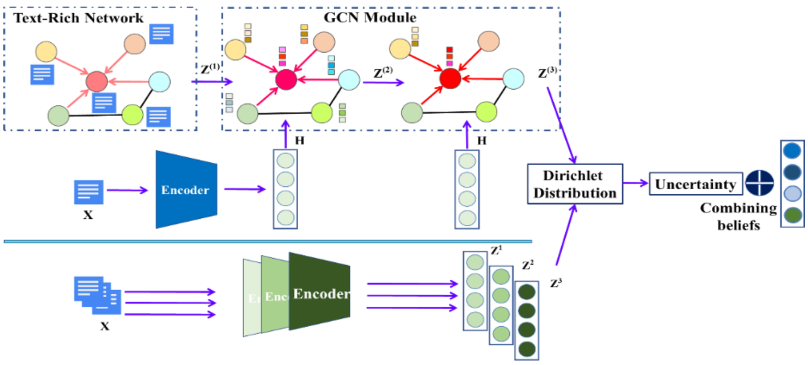

Figure 1 Visual illustration of our implementation of EGCN. Text-rich network obtains the graph structure information through GCN module, and at the same time, the text features of encoders such as BERT or SimCSE are fused with GCN with Eq. 2. The text features such as TF-IDF, BERT and SimCSE and the 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

The Dempster–Shafer Theory of Evidence (DST) is a theory on belief functions, was first proposed by Dempster and is a generalization of the Bayesian theory to subjective probabilities. Subjective Logic (SL) formalizes DST's notion of belief assignments over a frame of discernment as a Dirichlet Distribution. Hence, we introduce a principle of evidential theory-based uncertainty estimation technique which can provide more accurate uncertainty and allow us to flexibly integrate multi-typed features for trusted classification decision making. More specifically, SL considers a frame of K mutually exclusive singletons (e.g., class labels) by providing a belief mass b_k for each singleton k = 1, …, K and providing an overall uncertainty mass of u. These K + 1 mass values are all non-negative and sum up to one. Note that evidence e_k refers to the metrics collected from the multi-typed features to support the classification in Figure 2.

$$u + \sum_{k=1}^{K} b_k = 1 \tag{3}$$

where $u \geq 0$ and $b_k \geq 0$ for k = 1, . . . , K.

A belief mass b_k for a singleton k is computed using the evidence for the singleton. Let $e_k \geq 0$ be the evidence derived for the kth singleton, then the belief b_k and the uncertainty u are computed as:

$$b_k = \frac{e_k}{S} \ and \ u = \frac{K}{S} \tag{4}$$

where $S = \sum_{k=1}^{K} (e_k^v + 1)$.

Eq. 4 actually describes the phenomenon where the more evidence observed for the k-th category, the greater the probability assigned to the k-th class. A belief mass assignment, i.e., subjective opinion, corresponds to a Dirichlet distribution with parameters $\alpha_k = e_k + 1$. That is, a subjective opinion can be derived easily from the parameters of the corresponding Dirichlet distribution using $b_k = (\alpha_k - 1)/S$. However, a Dirichlet distribution parametrized over evidence represents the density of each such probability assignment; hence it models second-order probabilities and uncertainty (Han et al. 2021). The Dirichlet distribution is a probability density function for possible values of the probability mass function p. It is characterized by K parameters $\alpha = [\alpha_1, \ldots, \alpha_K]$ and is given by

$$D(p \mid \alpha) = \begin{cases} \frac{1}{B(\alpha)} \prod_{i=1}^{K} p_i^{\alpha_i - 1} & for p \in S_K \\ 0 & otherwise \end{cases} \tag{5}$$

where S_K is the K-dimensional unit simplex, $B(\alpha)$ is the K-dimensional multinomial beta function.

$$S_K = \{p \mid \sum_{i=1}^{K} p_i = 1 \ and 0 \leq p_1, \ldots, p_K \leq 1\} \tag{6}$$

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 $D(p_i|\alpha_i)$, 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 $\{M^v\}_1^V$, where $M^v = \{\{b_k^v\}_{k=1}^K, u^v\}$, 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:

$$\mathcal{M} = \mathcal{M}^1 \oplus \mathcal{M}^2 \tag{7}$$

The more specific calculation rule can be formulated as follows:

$$b_k = \frac{1}{1-C}(b_k^1 b_k^2 + b_k^1 u^2 + b_k^2 u^1), u = \frac{1}{1-C} u^1 u^2 \tag{8}$$

where $C = \sum_{i \neq j} b_i^1 b_j^2$ 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 α_i (i.e., $\alpha_i^v = e^i + 1$) 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 $Mult(y_i|p_i)$ and obtain the negated logarithm of the marginal likelihood by integrating out the class probabilities

$$\mathcal{L}_{ace}(\alpha_i) = \int \left[\sum_{j=1}^{K} -y_{ij} log(p_{ij})\right] \frac{1}{B(\alpha_i)} \prod_{j=1}^{K} p_{ij}^{\alpha_{ij}-1} dp_i = \sum_{j=1}^{K} y_{ij} \left(\psi(S_i) - \psi(\alpha_{ij})\right) \tag{9}$$

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

Eq. 9 is the integral of the cross-entropy loss function on the simplex determined by α_i. The above loss function ensures that the correct label of each sample generates more evidence than other classes, however, it cannot guarantee that less evidence will be generated for incorrect 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:

$$KL[D(p_i \mid \tilde{\alpha}_i) \parallel D(p_i \mid 1)] = \log\left(\frac{\Gamma(\sum_{k=1}^{K}\tilde{\alpha}_{ik})}{\Gamma(K)\prod_{k=1}^{K}\Gamma(\tilde{\alpha}_{ik})}\right) + \sum_{k=1}^{K}(\tilde{\alpha}_{ik}-1)\left[\psi(\tilde{\alpha}_{ik}) - \psi\left(\sum_{j=1}^{K}\tilde{\alpha}_{ij}\right)\right] \quad (10)$$

where $\tilde{\alpha}_i = y_i + (1 - y_i) \odot \alpha_i$ 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:

$$\mathcal{L}(\alpha_i) = \mathcal{L}_{ace}(\alpha_i) + \lambda_t KL[D(p_i \mid \tilde{\alpha}_i) \parallel D(p_i \mid 1)] \quad (11)$$

where $\lambda_t = \min(1.0, t/10) \in [0, 1]$ is the annealing coefficient, t is the index of the current training epoch, $D(p_i|h_1, . . . , h_i)$ 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:

$$\mathcal{L}_{\text{overall}} = \sum_{i=1}^{N}\left[\mathcal{L}(\boldsymbol{\alpha}_i) + \sum_{v=1}^{V}\mathcal{L}(\boldsymbol{\alpha}_i^{v})\right] \quad (12)$$

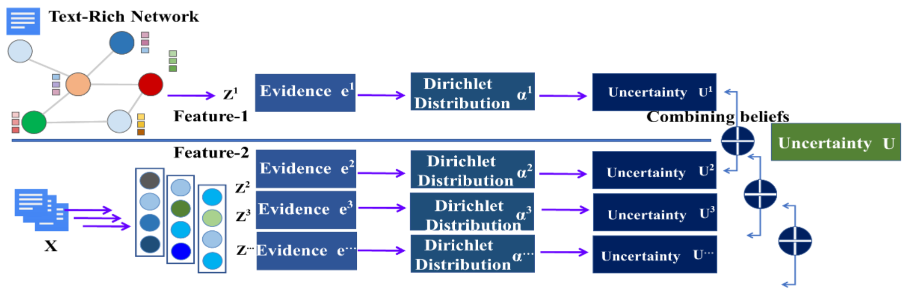

Figure 2 Illustration of trusted multi-typed features classification. The evidence of each feature is obtained using BERT, TF-IDF, SimCSE and GCN in Figure 1. The obtained evidence parameterizes the Dirichlet distribution to induce the classification probability and uncertainty. The overall uncertainty 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.

Dataset	Cora	Citeseer	DBLP
# Nodes	2211	4610	13404
# Edge	5214	5923	39861
# Text	169	10	10
# Classes	7	10	4

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

HuggingFace Transformers library3. We then fine-tune it using masked language model objective on the three real-world datasets: Cora, Citeseer and DBLP with a 10^{-5} learning rate. We set the number of layers to 2, And the hidden layer dimension is equal to 768, in order to be consistent with the dimension of the graph structure data and the BERT feature data. For the SimCSE method in the article, the temperature constant of the contrast loss function is set to 0.05.

As for our model, for different data sets, because the number of words in the document is different, the ability to represent different text features is not the same. Therefore, it is necessary to effectively fuse different features with confidence. The main idea is to choose the single-typed feature with higher classification capabilities. We choose multi-typed features from BERT, TF-IDF, and SimCSE methods and incorporate GCN output features to obtain the confidence and uncertainty of each feature for classification. In the fusion processing of multi-typed features, we set the fusion ratio λ to be 0.3.

4.3 Main Results

Table 2 presents the test accuracy of each model. We can see that EGCN outperforms other models in the three data sets. For the pre-training features of text data, the accuracy on the three data sets is low, the accuracy on the GCN, GAT and GraphSage models is improved to a certain ex-tent. Our method has a higher advantage. Competing with the strongest baseline BertGCN, our model outperforms it by 3% on Cora, by 9% on Citeseer, by 6% on DBLP.

Models	Cora	Citeseer	DBLP
BERT	0.55	0.60	0.63
SimCSE	0.71	0.67	0.67
GCN	0.78	0.72	0.65
GAT	0.79	0.80	0.69
GraphSage	0.78	0.81	0.71
BertGCN	0.83	0.88	0.84
EGCN	**0.86**	**0.97**	**0.90**

Table 2: Experimental results of node classification.

From the text, graph structure, text and graph structure fusion of Table 2 to multi-typed features trusted node classification, the following conclusions can be drawn: for graph node classification, Algorithms using both feature and graph information achieve better performance than methods leveraging information from single source. 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.

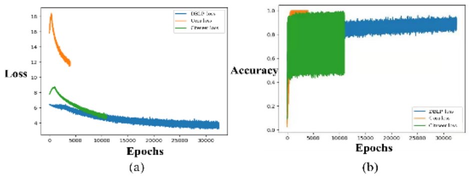

Figure 3 Illustration of the training process.

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 out-put dimension is 768, and BERT512 and BERT256 refer to the slave the 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.

MLP	Cora	Citeseer	DBLP
TF-IDF	0.85	0.84	0.79
SimCSE	0.71	0.67	0.67
Bert768	0.55	0.6	0.63
Bert512	0.49	0.41	0.58
Bert256	0.29	0.33	0.5

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

GCN+MLP	Cora	Citeseer	DBLP
TF-IDF	0.8381	0.9397	0.8715
SimCSE	0.6524	0.7397	0.6044
Bert768	0.7841	0.7511	0.5288
Bert512	0.7984	0.8519	0.8265
Bert256	0.7714	0.8153	0.7892

Table 4 Experimental results of classification of GCN nodes with multiple types of features

Table 4 shows the node classification results. We utilize the 2-layer GCN to aggregate the node feature neighbor in-formation after the text node has been learned by different feature expressions. It can be seen that, except for the feature vector 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. λ=1 means the node feature classification using GCN, and λ=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.

λGCN+(1-λ) Features	Cora	Citeseer	DBLP
TF-IDF	0.8048	0.8785	0.8436
SimCSE	0.7952	0.8715	0.8407
Bert768	0.8126	0.8708	0.8467
Bert512	0.7952	0.8769	0.8441
Bert256	0.7984	0.8769	0.8449

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

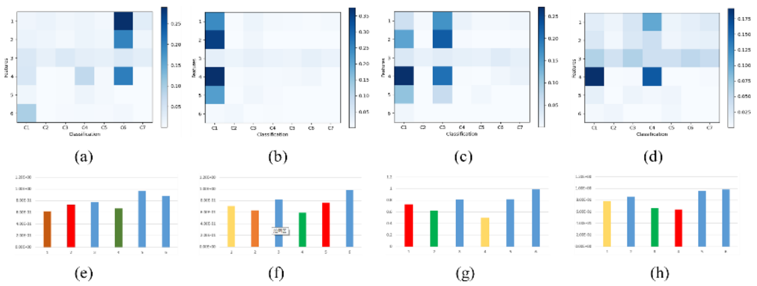

Figure 4 Illustration of node multi-type features classification confidence

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. This shows the effectiveness of our proposed model in modeling node features and network topology. By comparing EGCN with baseline methods, we can further infer the advantage of aggregating the multi-type 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 third type. The color distribution in Figure (d) is messy and the final classification is wrong. You can see the third row of main features are all dark in color, 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 confusion, and the experimental results are wrong. Figure (e) (f) (g) (h) represents the classification uncertainty of 6 different types of features for 4 samples. The lower the histogram, the lower the uncertainty and the higher the certainty. Figure (e) (f) (g) (h) respectively correspond to Figures (a) (b) (c) (d), in which the uncertainty of the six features for sample classification can also effectively reflect the confidence of the classification results.

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 pro-duce trusted classification decisions on multi-typed features and can jointly learn low-dimensional representations of both nodes and features for text-rich networks. Our al-gorithm 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.

Defferrard, M; Bresson, X.; Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems,3844-3852.

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.

Cui, G.; Zhou, J.; Yang, C.; Liu, Z. 2020. Adaptive graph encoder for attributed graph embedding. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 976-985.

Zhu, Q.; Du, B.; Yan, P. 2020. Self-supervised training of graph convolutional networks. arXiv preprint arXiv:2006.02380.

You, Y.; Chen, T.; Wang, Z.; Shen, Y. 2020. When does self-supervision help graph convolutional networks?. In International Conference on Machine Learning (pp. 10871-10880). PMLR.

Lin Y; Meng Y; Sun X. 2021. BertGCN: Transductive Text Classification by Combining GNN and BERT.In Findings of the Association for Computational Linguistics: ACL-IJCNLP,1456–1462.

Gao, T.; Yao, X.; Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In Empirical Methods in Natural Language Processing. In EMNLP.

Hamilton W; Zhitao Y; and Jure L. 2017. Inductive representation learning on large graphs. In Advances in neural information processing systems,1024–1034.

Keyulu X; Weihua H; Jure L; and Stefanie J. 2018. How powerful are graph neural networks? arXiv preprint arXiv:1810.00826.

Li, J.; Dani, H.; Hu, X.; Tang, J.; Chang, Y.; and Liu, H. 2017.Attributed network embedding for learning in a dynamic environment. In CIKM. ACM.

Zhang, Z.; Yang, H.; Bu, J.; Zhou, S.; Yu, P.; Zhang, J.; Ester, M.;and Wang, C. 2018. Anrl: Attributed network representation learning via deep neural networks. In IJCAI.

Zhou, S.; Yang, H.; Wang, X.; Bu, J.; Ester, M.; Yu, P.; Zhang, J.;and Wang, C. 2018. Prre: Personalized relation ranking embed-ding for attributed networks. In CIKM. ACM.

Veliˇckovi´c, P.; Fedus, W.; Hamilton, W. L.; Li` o, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep graph infomax. ICLR.

Meng, Z.; Liang, S.; Bao, H.; and Zhang, X. 2019. Co-embedding attributed networks. In WSDM. ACM.

Liao L.; He X.; Zhang H.2018. Attributed social network embed-ding, IEEE Transactions on Knowledge and Data Engineering.

Yang H.; Pan, S; Zhang P; Chen L.; Lian D.; and Zhang C..2018. Binarized attributed network embedding," in ICDM. IEEE,1476–1481.

Li J.; Dani H.; Hu X.; Tang J.; Chang Y.; and Liu H. 2017. Attributed network embedding for learning in a dynamic environ-ment.in CIKM. ACM, 387–396.

Huang X.; Li J.; and Hu X. 2017. Accelerated attributed network embedding. In SIAM International Conference on Data Mining, 633–641.

Xiao H.; Jundong L.; and Xia H. 2017. Label informed attributed network embedding. In WSDM. ACM, 731–739.

Zhang H and Zhang X. 2020. Text graph transformer for document classification. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP),8322–8327.

Lu Z; Pan Du P; and Nie J. 2020. Vgcn-bert: augmenting bert with graph embedding for text classification. In European Conference on Information Retrieval, pages 369–382. Springer.

Radford M Neal. 2012 Bayesian learning for neural networks. Springer Science & Business Media.

Gal Y and Ghahramani Z. 2016. Dropout as a bayesian approxima-tion: Representing model uncertainty in deep learning. In International Conference on Machine Learning, 1050–1059.

Lakshminarayanan B; Pritzel A; and Blundell C. 2017.Simple and scalable predictiveuncertainty estimation using deep ensembles. In Advances in Neural Information Processing Systems, 6402–6413.

Sensoy M; Kaplan L and Kandemir M. 2018. Evidential deep learning to quantify classification uncertainty. In Advances in Neural Information Processing Systems, 3179–3189.

van Amersfoort J; Smith L; Whye Te ,Y and Gal Y. 2020. Uncer-tainty estimation using a single deep deterministic neural network. In International Conference on Machine Learning.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 7482–7491.

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

Jøsang A and Hankin R. 2012. Interpretation and fusion of hyper opinions in subjective logic. In 2012 15th International Conference on Information Fusion,1225–1232

Devlin, J.; Chang, M. W.; Lee, K; Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language under-standing. arXiv preprint arXiv:1810.04805.

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

Hamilton W, Ying Z, Leskovec J.2017.Inductive representation learning on large graphs. Advances in Neural Information Processing Systems, 1024-1034.