Multimodality fusion for node classification in D2D communications

ZUFAN ZHANG1, XIELIANG LI1, CHENQUAN GAN1

1School of Communication and Information Engineering, Chongqing University of Post and Telecommunications, Chongqing 400065, China

Corresponding author: Xieliang Li (xieliangli2016@163.com).

This work is supported by Chongqing Research Program of Basic Research and Frontier Technology (Grant Nos. cstc2017jcyjAX0256 and cstc2018jcyjAX0154)

ABSTRACT With the rapid development of 5G technology, D2D communications will be widely used in content distribution scenarios. An important prerequisite for realizing data transmission in D2D scenarios is to correctly classify each node in the network. However, most of existing node classification methods need huge computational overhead in large-scale networks. Inspired by network structure and attribute information, a multimodality fusion based node classification scheme is proposed to obtain the tradeoff between classification accuracy and costs. First, in order to summarize label dependencies to perform node classification, structure and information features are constructed. Structure features are obtained by embedding network data into a low dimensional representation space and information features are extracted from attribute information. Second, a multimodality fusion model is applied to integrate knowledge learned from different modalities, which can further improve the overall performance. Besides, the multilayer neural network based on softmax regression is used for multi-label classification. Finally, extensive experiments on two public datasets are conducted to verify the validity of the proposed scheme. Specifically, experimental results demonstrate the superior performance of our method over existing solutions in terms of accuracy and overhead.

INDEX TERMS D2D communications, node classification, multimodality fusion, network embedding, attribute information.

I. INTRODUCTION

With the rapid development of 5G technology technology, various types of devices generate a tremendous amount of network data [1]. To reduce network load and improve network resource utilization, D2D communications have a wide application range in content distribution scenarios [2]. The precondition of content distribution is to correctly classify each node in the network, and then select seed nodes from the community to forward the content information. There is a lot of interactions between devices and the device that have been generating data. Hence, the network consisting of massive devices can be modeled as a graph or graph-like structure, where devices are represented as nodes associated with information [3]. With the advent of social platforms such as Twitter, Weibo and Weixin, social networks have been integrated into people’s daily lives, and people are engaging in social activities almost anytime and anywhere [4]. Since most of devices are carried by people, the subjectivity of people will largely affect the communication process and the network composed of many devices can also be regarded as a social network. However, in real world, only part of nodes are labeled due to the high costs of labeling. The target of node classification is to predict the labels of unlabeled nodes by utilizing partial labeled samples.

The key idea for node classification is to construct node features that summarize label dependencies to perform classification tasks. In general, previous work on node classification is divided into three methods. Node classification using network structure only is [5]- [6]. These methods are based on the homophily assumption [7] that linked nodes are inclined to share similar labels. Since the degree distribution of nodes in social networks follows a power law, the major drawback of these methods is that the classification only relies on network structure, which might get poor classification accuracy when the link sparsity problem exists. Node classification with attribute information is [8]- [10]. For social applications like Weibo, user information such as gender, age, location, and other preference is often regarded as node features. These methods apply common classifiers, such as the multilayer neural network and support
vector machine, to the attribute information of each node. The drawback of attribute information based methods is they neglect the network structure information when establishing the classifier. Node classification based on network structure and attribute information. These methods [11]- [15] are a subfield of collective classification methods, which utilize both network structure and attribute information in social networks. A lot of these studies have shown that blended methods are usually superior to use network structure or attribute information alone. However, a discrete adjacency matrix is often utilized to represent the network. Since the degree distribution of nodes in social networks follows a power law, nodes are represented by a high dimensional sparse vector. Although the node classification of network data has been extensively studied, such a traditional routine often makes node classification tasks have huge computational complexity and sometimes, even infeasible over large-scale networks [16].

Recently, many network embedding methods have been proposed [17], [18], [19], [20], which obtain a potential low-dimensional representation for each node and preserve the network proximity. Authors in [21] proposed that node classification tasks, applying common vector-based machine learning algorithm in the low-dimensional representation space, are easily efficiently conducted. The above network embedding methods can show promising performance in their specific network structure scenario, but all of these methods only focus on utilizing network structure rather than integrating attribute information. In addition, matrix factorization based methods like text Associated DeepWalk (TADW) [22] and Homophily, structure, and content augmented network representation (HSCA) [23] explore the possibility of integrating attribute information to learn informative representations. However, matrix operation like SVD decomposition needs computational expensive overhead, which prevents these methods from handling large-scale network [16].

In this paper, a generic node classification scheme utilizing multimodality fusion is first proposed in this paper. The scheme mainly includes four parts: network embedding, attribute information extraction, multimodality fusion and label prediction. In the first part, network embedding methods learn low-dimensional representations of nodes only from network structure. In the second part, information features are extracted by representing attribute information with quantified feature vectors. Furthermore, a multimodality fusion model is exploited to integrate knowledge learned from these two modalities to improve the overall performance. Finally, the multilayer neural network based on softmax regression is used to predict labels of nodes. The scheme adopts the stochastic gradient descent (SGD) optimization strategy, which is scalable and naturally parallelized, and can be applied to large-scale social networks. Specifically, experimental results indicate that the proposed scheme significantly outperforms other baselines on two public datasets.

In summary, the main contributions of this paper are as follows:

1. We propose a multimodality fusion based node classification scheme, which integrates knowledge learned from both network structure and attribute information in the low dimensional feature space. The experimental results show that it is superior to most existing solutions in terms of accuracy and overhead.

2. The proposed scheme directly takes low dimensional structure features as the input of multimodality fusion model, so it can be apply to most of purely structure-based network embedding methods, such as DeepWalk, node2vec, Line and so on.

The rest of this paper is organized as follows. Problem definition and preliminaries are mentioned in Section II. Section III introduces a detailed description of the proposed scheme, including network embedding, attribute information extraction, multimodality fusion and label prediction. Section IV discusses the experiment results and Section V concludes this paper.

II. PROBLEM DEFINITION AND PRELIMINARIES

This section formally defines the research problem and gives a preliminary overview of network embedding methods related to our work.

A. PROBLEM DEFINITION

Definition 1. Social networks: As indicated in Fig. 1, social networks are represented as \( G = (V, E, A, Y) \), where \( V = \{ v_i \}_{i=1}^{N} \) consists of a series of nodes, \( E \) denotes a series of edges connecting nodes, \( a_i \in A \) is the attribute information associated with each node \( v_i \), and \( Y \) is related with a class label \( y_i \in Y \). As we know, links might have multiple forms in the different situations. For example, links are defined as interactions between devices in D2D communications, and in citation networks, each paper is considered as a node, links are defined as reference relationships between papers. In real word, it is difficult to spend a lot of time and effort labeling each node, hence the input of node classification is partially labeled networks.
Definition 2. Partially labeled networks: Partially labeled networks are defined as \( G = (V_L, V_U, E, A, Y_L) \), where \( V_L \cup V_U = V \), \( V_L \) and \( V_U \) denote a series of labeled nodes and unlabeled nodes, respectively. \( Y_L \) is the labels of labeled nodes \( V_L \). According to above concepts, the problem of node classification in social networks can be defined as follows.

Problem definition. Node classification in social networks: Formally, given partially labeled networks \( G = (V_L, V_U, E, A, Y_L) \), the target of node classification is to predict labels of unlabeled nodes \( V_U \) with network structure and attribute information.

B. NETWORK EMBEDDING

The proposed scheme directly takes low-dimensional structure features as the input of multimodality fusion model, hence it is apply to many network embedding methods. And this part gives a preliminary overview of network embedding methods related to our work.

![Adjacency matrix](image1)

As illustrated in Fig. 2, adjacency matrix is the most intuitionistic representation of network data. But for large-scale social networks, adopting the adjacency matrix usually makes node classification tasks computationally expensive and intractable. Recent works concentrate on embedding large-scale networks into a low-dimensional representation space and then implementing node classification tasks. DeepWalk [17] firstly generalized the neural language model Word2vec [24] to deal with a special sentence consisting of a series of truncated random walks. It can learn low-dimensional social representations of network nodes and preserve neighborhood similarity [25]. Following this line, the authors in node2vec [26] used biased random walk to obtain a flexible neighborhood sampling and achieved performance improvements. Different from the network embedding methods that performing random walks on the networks, [18] introduced probability distribution models to preserve the first-order proximity and second-order proximity of nodes. GraRep [27] further tried to consider different k-step proximity (in general \( k > 2 \)) to construct a better global representation.

All of these methods only focus on utilizing network structure rather than incorporating attribute information, and this paper focus on incorporating existing purely structure-based network embedding methods with attribute information to further improve the accuracy of node classification.

III. THE PROPOSED SCHEME FOR NODE CLASSIFICATION

In this section, the framework of the proposed scheme is described. As shown in Fig. 3, the proposed scheme mainly includes four parts: network embedding, attribute information extraction, multimodality fusion and label prediction. In the first part, network embedding methods are used to obtain latent low-dimensional representations of social network nodes. And then, attribute information is represented by feature vectors. It is considered that the results of different modalities should be similar in dealing with the same task [28], [29]. Hence, penalties between predicted label distributions of these two modalities are added to the loss function for implementing multimodality fusion in the third part. Finally, the multilayer neural network based on softmax regression is utilized to predict the label of nodes.

A. NETWORK EMBEDDING

DeepWalk that proposed by Perozzi et al. is the most influential method of network embedding. In this paper, it is adopted to obtain potential low-dimensional representations of nodes only from network structure. DeepWalk for the first time introduced Skip-Gram [30], an extensively used distributed word representation method, to learn structure features. It is similar to the word vector training model in natural language processing, which regards truncated random walks as special sentences and utilizes local information obtained from these sentences to learn potential node representations. These special sentences \( S = (v_1, v_2, ..., v_L) \) are generated by random walks, and the nodes \( v \in \{v_{-t}, ..., v_{+t}\} \) are viewed as the context of center node \( v_t \), where \( L \) is the walk length and \( t \) is the window size.

As shown in Fig. 4, following the idea of Skip-Gram, the node \( v_i \) is utilized to predict its context nodes, which is equal to yield the optimization problem:

\[
\min_{j} - \log \Pr \left( \{v_{-t}, ..., v_{+t}\} \setminus v_i \mid f(v_i) \right)
\]

And it uses a conditional independence assumption, the probability \( \Pr \left( \{v_{-t}, ..., v_{+t}\} \setminus v_i \mid f(v_i) \right) \) is approximated as

\[
\Pr \left( \{v_{-t}, ..., v_{+t}\} \setminus v_i \mid f(v_i) \right) = \prod_{j=i-t, j \neq i}^{i+t} \Pr (v_j \mid f(v_i))
\]
It can be seen that DeepWalk uses truncated random walks to implement the local search. Since context nodes in the stream of short random walks actually describe the neighborhood structure, nodes that have similar context nodes tend to be represented closely in the low-dimensional vector space. Furthermore, Hierarchical Softmax and Negative Sampling are adopted to simplify process and speed up the training time.

B. ATTRIBUTE INFORMATION EXTRACTION

The representation of information and selection of its feature items are a basic problem in data mining and information retrieval. Since text information widely exists in the social networks, this paper mainly focus on the feature extraction of text information. To transform an unstructured original text into structured data that computer can process, TF-IDF uses quantified feature words extracted from text to represent text information. TF-IDF is calculated as follows:

\[ \text{TF}_{i,j} = \frac{N_{i,j}}{\sum_{k=0}^{n} N_{k,j}} \]  \hspace{1cm} (3)

where \( N_{i,j} \) is the number of occurrences of the \( i \) -th word in text \( t_j \), and the denominator indicates the total number of words in text \( t_j \).

\[ \text{IDF}(c_i) = \log \left( \frac{|T|}{|\{t_j \in T : c_i \in t_j\}|} \right) \]  \hspace{1cm} (4)

where \(|T|\) is the total number of texts in the corpus, and \( \{t_j \in T : c_i \in t_j\} \) indicates the number of texts containing the characteristic \( c_i \).

\[ \text{TF} - \text{IDF} = \text{TF} \times \text{IDF} \]  \hspace{1cm} (5)

In this way, attribute information is represented by quantified feature vectors. In general, feature vectors are sparse since the corpus contains a large number of texts.

To further reduce the dimension of feature vectors, Principal Component Analysis (PCA) is used to map the \( n \) dimensional feature to the \( k \) dimension \( (k < n) \) and preserve orthogonal in the projection space. For a given set of data \( \{a_1, a_2, ..., a_n\} \), firstly, the average of vectors is calculated.

\[ \mu = \frac{1}{n} \sum_{i=1}^{n} a_i \]  \hspace{1cm} (6)

And then, all the samples are subtracted the corresponding mean.

\[ \{x_1, x_2, ..., x_n\} = \{a_1 - \mu, a_2 - \mu, ..., a_n - \mu\} \]  \hspace{1cm} (7)

The projection of sample points \( x_i \) in the hyperplane of new space is \( W^T x_i \). In order to make the projection of all the sample points as separate as possible, the total variance of the sample points after the projection should be maximized, so the optimized objective function can be written as:

\[ \max_W \text{tr} (W^T XX^T W) \]  \hspace{1cm} s.t. \( W^T W = I \)  \hspace{1cm} (8)

It uses the Lagrange multiplier method to get the following equation:

\[ XX^T W = \lambda W \]  \hspace{1cm} (9)

So only the covariance matrix \( XX^T \) needs eigenvalue decomposition. Furthermore, eigenvalues are sorted from large to small and the corresponding \( k \) feature vectors are utilized to form the eigenvector matrix. Finally, sample points are projected into \( k \)-dimensional vector space.

C. MULTIMODALITY FUSION

After obtaining structure and information features, it is considered that the results of different modalities should be similar in dealing with the same task, a multimodality fusion model is utilized to incorporate knowledge learned from these two modalities to further improve the overall performance. The label information of partial nodes directly indicates nodes’ group or class affiliation, which helps to improve the accuracy of node classification. Existing labeled nodes \( V_L \) are used to train the multilayer neural network, and then use it to predict the label of other nodes \( V_U \). The Multilayer neural network based on softmax regression has a
natural advantage in dealing with multi classification tasks, which can directly output the prediction probability of all classes. Softmax function is adopted to evaluate the probability distribution and the prediction probability of classifying input $x_i$ into class $j$ is

$$p(y_i = j| x_i; \theta) = \frac{e^{\theta_j^T x_i}}{\sum_{k=1}^{K} e^{\theta_k^T x_i}}$$

where $K$ is the total number of classes and $\theta_k$ is the k-th class’s parameter vector. Since the results of different modalities should be similar in dealing with the same task, consistent constrains are imposed across different modalities by adding penalties between the predicted label distributions. In the field of machine learning, cross-entropy is often utilized to measure the differences between the two distributions $p$ and $q$, which can be defined as

$$H(p||q) = -\sum_{k=1}^{K} p_k \ln q_k$$

Since cross-entropy is not symmetrical, $H_d(p||q)$ is defined as

$$H_d(p||q) = H(p||q) + H(q||p)$$

And L2 regularization is added to the loss function to prevent overfitting.

$$C = C_0 + \frac{\lambda}{2} \sum_{w} w^2$$

where $C_0$ is the original loss function and $\sum w^2$ is the sum of the squares of all the parameters. $\lambda$ is the hyper parameter to control the weights of $C_0$ and L2 regularization. As seen in Fig. 5, there are two modality features derived from network structure and attribute information, so $x^m_i$ (for $m \in \{1, 2\}$ ) are denoted by the m-th modality features of the i-th instance. And the loss function has two sets of parameters $\{\theta^1, \theta^2\}$ that need to be learned. The goal is

$$\min_{\Theta, \lambda, \beta_1, \beta_2, \beta_3} L(\Theta) = \frac{\beta_1}{N} \sum_{i=1}^{N} H_d(p_{\theta^1}(x^1_i)||y_i) + \frac{\lambda}{2} \theta^1^T \theta^1$$

$$+ \frac{\beta_2}{N} \sum_{i=1}^{N} H_d(p_{\theta^2}(x^2_i)||y_i) + \frac{\lambda}{2} \theta^2^T \theta^2$$

$$+ \frac{\beta_3}{N} \sum_{i=1}^{N} H_d(p_{\theta^1}(x^1_i)||p_{\theta^2}(x^2_i))$$

where $\{\beta^1, \beta^2, \beta^3\}$ are the hyper parameters, which control the relative weights of these three parts in the loss function. Gradient descent is adopted to optimize the objective function $L(\Theta)$ in Eqn. (14). And the gradient of the $L(\Theta)$ with respect to $\theta^1$ is

$$\frac{\partial L(\Theta)}{\partial \theta^1} = \frac{\beta_1}{N} \sum_{i=1}^{N} \frac{\partial H_d(p_{\theta^1}(x^1_i)||y_i)}{\partial \theta^1} + \lambda \theta^1$$

For last term $\frac{\partial H_d(p_{\theta^1}(x^1_i)||p_{\theta^2}(x^2_i))}{\partial \theta^1}$, since $H_d(p||q)$ is symmetrical. Hence, it can be conclude as follows:

$$\frac{\partial H(p_{\theta^1}(x^1_i)||p_{\theta^2}(x^2_i))}{\partial \theta^1} = -\sum_{k=1}^{K} \left(p_{\theta^2}(k|x^2_i) + \ln(p_{\theta^2}(k|x^2_i))\right) \frac{\partial p_{\theta^1}(k|x^1_i)}{\partial \theta^1_{jl}}$$

where

$$\frac{\partial p_{\theta^1}(k|x^1_i)}{\partial \theta^1_{jl}} = \frac{\partial \exp(\theta^1_{jl}^T x^1_i)}{\sum_{k=1}^{K} \exp(\theta^1_{k}^T x^1_i)}$$

$\theta^1_{jl}$ is a sub-vector of $\theta^1$ and $\theta^1_{jl}$ is used to represent the l – th element of $\theta^1$. Since $H_d(p_{\theta^1}(x^1_i)||p_{\theta^2}(x^2_i))$ is symmetric in terms of $\theta^1$ and $\theta^2$, Eqn. (16) can also be used to calculate the gradient of the objective function $L(\Theta)$ with respect to $\theta^2$.

There are two groups of parameters $\{\theta^1, \theta^2\}$, which need to be iteratively learned in our implementation. Concretely, the two groups of parameters will try to be updated in each iteration. Since there are large datasets in experiments, mini-batch is adopted to speed up the process of learning parameters. The main steps for multimodality fusion are as following:

<table>
<thead>
<tr>
<th>TABLE 1: Algorithm 1 Multimodality Fusion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Algorithm 1 Multimodality Fusion</strong></td>
</tr>
<tr>
<td>Require:1. $X^1, X^2$ two different modality features on the object $X$</td>
</tr>
<tr>
<td>2. $Y = {y_1, y_2, \ldots, y_N}$ labels of $X$</td>
</tr>
<tr>
<td>1: Randomly divide the dataset into mini-batches</td>
</tr>
<tr>
<td>2: Randomly initialize $\Theta = {\theta^1, \theta^2}$</td>
</tr>
<tr>
<td>3: repeat</td>
</tr>
<tr>
<td>4: for i from 1 to 2 do</td>
</tr>
<tr>
<td>5: Randomly select a mini-batch $X_m$</td>
</tr>
<tr>
<td>6: Use gradient descent to update $\theta^0$ on $X_m$ with objective function in Eqn. (14) and derivative in Eqn. (16)</td>
</tr>
<tr>
<td>7: end for</td>
</tr>
<tr>
<td>8: until Results converge or the number of iterations reaches the threshold</td>
</tr>
<tr>
<td>9: return $\Theta$</td>
</tr>
</tbody>
</table>

**D. LABEL PREDICTION**

The target of multimodality fusion is to enforce the consistency of prediction results since results of different modalities for the same task should be similar. Similarly, in the process...
of label prediction, define \( p \) is the final label distribution, \( p \) is obtained by the following function:

\[
\min_p L(p|p_{\theta_1}, p_{\theta_2}) = \sum_{k=1}^{K} p(k) \ln \left( \frac{p(k)}{p_{\theta_1}(k)} \right) + \sum_{k=1}^{K} p(k) \ln \left( \frac{p(k)}{p_{\theta_2}(k)} \right) \quad \text{s.t.} \sum_{k=1}^{K} p(k) = 1
\]

(18)

And the Lagrange function of this problem is

\[
\Lambda(p|p_{\theta_1}, p_{\theta_2}) = \sum_{k=1}^{K} p(k) \ln \left( \frac{p(k)}{p_{\theta_1}(k)} \right) + \sum_{k=1}^{K} p(k) \ln \left( \frac{p(k)}{p_{\theta_2}(k)} \right) + \lambda \left( \sum_{k=1}^{K} p(k) - 1 \right)
\]

(19)

Let the derivative of \( \Lambda \) with respect to \( p(k) \) equal 0, the final probability can be obtained by the above formula:

\[
p(k) = \sqrt{\prod_{m=1}^{2} p_{\theta_m}(k)} / \sqrt{\sum_{j=1}^{K} \prod_{m=1}^{2} p_{\theta_m}(j)}
\]

(20)

In this way, it incorporates knowledge learned from these two modalities to improve the overall performance of the proposed scheme. And the proposed scheme that mainly adopt SGD optimization strategy is much more efficient than network embedding methods based on matrix factorization, which is suitable for large-scale social networks.

E. COMPLEXITY ANALYSIS

For large-scale network data, the general approach is to embed it into a low dimensional representation space and then use a classifier to implement the task of node classification. To our best knowledge, the time complexity of multilayer neural network is not clearly defined yet. However, as a classifier, it has been proved to widely use in machine learning tasks, especially for large-scale datasets and multi-classification tasks. Furthermore, matrix factorization based methods like TADW and HSCA takes \( O(|V||E| + d|I| |E| + dmI|V| + d^2I|V|) \) time [16], where \( d \) is the dimension of obtained node representations, \( m \) is the dimension of attribute information, and \( I \) is the number of iterations. And note that, \( |V| \) is far greater than \( d \) and \( m \), which directly determines the size of the time complexity. Since most real-world networks are sparse, \( O(|E|) = O(|V|) \) [22], the time complexity of TADW and HSCA is at least \( O(|V|^2) \). And the time complexity of DeepWalk is \( O(d|V| \log |V|) \) [16]. Besides, the proposed scheme integrates knowledge of network structure and attribute information in the low-dimensional representation space, the time complexity of this part is related to \( d \) and \( m \), which are far less than the number of nodes \( |V| \). Hence in terms of time complexity, the proposed scheme is superior to TADW and HSCA. In addition, the proposed scheme adopts SGD optimization strategy, which is scalable and naturally parallelized, and can be applied to large-scale social networks.

IV. EXPERIMENTS

In this part, since there is no additional attribute information in the three datasets which are used in DeepWalk, experiments are conducted on two other public datasets to verify the validity of our method. Formally, it is to predict the labels of unlabeled nodes \( V_U \) with labeled nodes \( V_L \) based on node features. For a fair comparison, the multilayer neural network is used as the classifier in all cases. It is considered that partial labeled nodes are used as training sets, the training rate of the classifier varies from 10% to 50%. For each training ratio, the datasets are randomly divided into training sets and test sets. In order to reduce the error caused by the selection of the training sets, experiments are repeated for 10 times and the average accuracy is taken as the final result. In additional, two indicators, F1-micro and F1-macro, often used in text categorization to measure the effect of the experiments.

A. DATASETS

The two real-world datasets used in the experiments are Cora and Citeseer. They are citation networks and for each network, link directions and self-links are ignored. Hence, there is an edge between the two nodes, if any of them have a direct link to another. And the details of these two datasets are represented as follows:

Cora includes 2,708 machine learning papers from seven classes. And there are 5,429 links between them. These links are citation relationships between the papers. And the contents in Cora are short texts generated from titles and...
Citeser consists of 3, 312 scientific publications from six classes. Every publication in Citeser refers or is referred by at least one other publication. And the citation network contains 4, 732 links. Similar to Cora, the links are citation relationships between the publications. And it is also regarded as an undirected graph.

B. PARAMETER SETTINGS

DeepWalk [17]. It learns low dimensional representations of nodes utilizing network structure only. According to the original paper, the parameters are set as follows, walk length $L = 80$ and window size $t = 10$. To find the representation dimension $k$ with best performance in Cora and Citeser, experiments are conducted under different representation dimensions between 50 and 150. We find $k = 100$ is the best value, which is similar to Perozzi’s paper. So, the dimension of structure features is set to 100.

TF-IDF. All words in Cora and Citeser that appear less than 10 and stop words are removed. Each document in Cora and Citeser is described by a TF-IDF matrix indicating the presence of the key word.

Information Features. Information features are obtained by implementing the PCA algorithm on the TF-IDF matrix. The dimension of information feature vector is also set to 100 for a fair comparison. In theory, the effect of the experiment will be better as the representation dimension increases, but it will lead to huge computational overhead, which has violated our original intention.

Early Fusion. Early fusion concatenates feature vectors of network structure and attribute information to form a single feature vector, hence the dimension of early fusion is 200.

HSCA [23]. It is an excellent algorithm that utilizes both network and attribute information to learn the network representation. According to [23], HSCA is relatively robust to the change of representation dimension $k$. Besides that, we perform experiments under different representation dimensions between 100 and 300, the experimental results have also proved it. For fair comparison, the dimension of HSCA is also set to 200 and other parameters are set as default values. Note that, SVM is used as the classifier in [23] and the multilayer neural network is utilized as the classifier in this paper. So, the final results may be slightly different.

Proposed Method. Features of the proposed method include structure and information features, so the dimension of proposed method is 200. And the setting of hyper parameters is manually adjusted according to the training results.

F1-macro and F1-micro. When dealing with multi-label classification tasks, micro-average and macro-average are often used to evaluate the effectiveness of these measures. The former weights equally all the documents, thus favouring the performance on classes with numerous documents. Note that, the value of micro-average is equal to the accuracy. And the latter weights equally all the classes, no matter how many documents belong to the class.

C. RESULT ANALYSIS

<table>
<thead>
<tr>
<th>Training Ratio</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepWalk</td>
<td>75.57</td>
<td>78.69</td>
<td>80.07</td>
<td>81.42</td>
<td>82.13</td>
</tr>
<tr>
<td>Information Features</td>
<td>57.62</td>
<td>66.59</td>
<td>70.51</td>
<td>72.39</td>
<td>73.27</td>
</tr>
<tr>
<td>Early Fusion</td>
<td>76.26</td>
<td>81.09</td>
<td>82.12</td>
<td>83.52</td>
<td>83.92</td>
</tr>
<tr>
<td>HSCA</td>
<td>83.46</td>
<td>84.39</td>
<td>85.06</td>
<td>85.89</td>
<td>86.42</td>
</tr>
<tr>
<td>Proposed method</td>
<td><strong>83.74</strong></td>
<td><strong>84.87</strong></td>
<td><strong>85.96</strong></td>
<td><strong>86.62</strong></td>
<td><strong>87.23</strong></td>
</tr>
</tbody>
</table>

Note that, although DeepWalk is an excellent method for embedding networks, it is designed to utilize only the network structure. For fairness of comparison with the proposed method that additionally exploits attribute information, early fusion and HSCA are adopted in the experiments. Table 3 and Table 4 show micro-F1 and macro-F1 results in Cora, respectively. Clearly, with the increase of training ratio, the classification accuracy for all methods will be enhanced. First and foremost, it can be seen that the proposed method consistently outperforms the other baselines in all cases. Furthermore, in the case of the same classifier, the proposed method outperforms DeepWalk and Information Features at least by 5% in Cora, which demonstrates that integrating attribute information can achieve a higher accuracy compared with only utilizing network structure.
TABLE 6: Micro-F1 (%) in Citeseer dataset.

<table>
<thead>
<tr>
<th>Training Ratio</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepWalk</td>
<td>52.98</td>
<td>57.07</td>
<td>60.98</td>
<td>62.82</td>
<td>64.29</td>
</tr>
<tr>
<td>Information Features</td>
<td>60.56</td>
<td>65.06</td>
<td>66.74</td>
<td>67.23</td>
<td>68.22</td>
</tr>
<tr>
<td>Early Fusion</td>
<td>62.07</td>
<td>66.42</td>
<td>69.26</td>
<td>70.09</td>
<td>71.26</td>
</tr>
<tr>
<td>HSCA</td>
<td>69.27</td>
<td>70.69</td>
<td>71.41</td>
<td>72.46</td>
<td>73.90</td>
</tr>
<tr>
<td>Proposed method</td>
<td>70.02</td>
<td>71.16</td>
<td>72.45</td>
<td>73.28</td>
<td>74.25</td>
</tr>
</tbody>
</table>

TABLE 7: Macro-F1 (%) in Citeseer dataset.

<table>
<thead>
<tr>
<th>Training Ratio</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepWalk</td>
<td>47.77</td>
<td>52.39</td>
<td>55.52</td>
<td>56.33</td>
<td>58.45</td>
</tr>
<tr>
<td>Information Features</td>
<td>54.47</td>
<td>58.39</td>
<td>60.40</td>
<td>60.97</td>
<td>61.83</td>
</tr>
<tr>
<td>Early Fusion</td>
<td>56.06</td>
<td>60.73</td>
<td>63.13</td>
<td>64.47</td>
<td>65.12</td>
</tr>
<tr>
<td>HSCA</td>
<td>63.78</td>
<td>64.28</td>
<td>65.15</td>
<td>66.70</td>
<td>67.24</td>
</tr>
<tr>
<td>Proposed method</td>
<td>64.16</td>
<td>65.83</td>
<td>66.35</td>
<td>67.56</td>
<td>68.10</td>
</tr>
</tbody>
</table>

Similar to Cora, Table 5 and Table 6 show micro-F1 and macro-F1 results in Citeseer, respectively. In this dataset, the quality of DeepWalk is poor since it contains less link information (as illustrated in Citeseer introduction). In this way, with the training ratio decreases, the link sparse problem will become more serious. While, the proposed method exhibits more stability when the training ratio is small, which is credible to its effective incorporating of attribute information.

In addition, confusion matrices at 25% training ratio are investigated in order to further reveal the accuracy of each class. As illustrated by Fig. 7 and Fig. 8, the most of classes achieve a satisfactory accuracy. However, the accuracy of IR in Citeseer is at a low level. And it can be interpreted from two aspects. On the one hand, samples of this class account for a small proportion of the dataset. Since datasets are randomly divided into training sets and test sets, the probability of selecting training samples in this class is smaller than others. Hence it is easier to mistake this type of samples in testing process. On the other hand, the difference between samples of this class and others is indistinctive, so it is difficult to distinguish them.

From the above discussion, the proposed method consistently outperforms all the other baselines. It considers jointly from network structure and attribute information, which have a preferable performance in node classification task comparing with the single modality. Moreover, the proposed method can be applied into most of purely structure-based network embedding methods to improve the accuracy of node classification. And these experiments demonstrate that proposed method is effective and robust.

D. PARAMETER SENSITIVITY

There are three key hyper parameters \{β₁, β₂, β₃\} in Eqn. (14), which control the relative weights of different parts in our model. Since DeepWalk consistently outperforms information features in Cora, β₁ is set as 1, β₂ varies from 0.2
to 1.2 and $\beta_3$ varies from 0.2 to 1. Meanwhile, the training ratio is fixed to 25% and experiments are conducted with different $\beta_2$ and $\beta_3$. As can be seen from the Fig. 9 that in all cases, the classification accuracies reach the maximum when $\beta_3$ is 0.4. This can be explained by that, compared with other parts, penalties in Eqn. (14) ought to have a smaller weight. And the experimental results have also proved it. Furthermore, it achieves the best overall performance under the condition of $\beta_1 = 1$, $\beta_2 = 0.4$ and $\beta_3 = 0.4$. And with the growth of $\beta_2$, the classification accuracy continues to drop, which corresponds with the precondition that DeepWalk is superior to information features in Cora. As indicted in Fig. 10, similar to Cora, since information features beats DeepWalk in Citeseer, the classification accuracy reaches the maximum when $\beta_1 = 0.8$, $\beta_2 = 1$ and $\beta_3 = 0.6$. Moreover, as can be seen from the graphs, the classification accuracies range within 1% when adjusting hyper parameters in Cora and Citeseer. Hence, these experimental results demonstrate that the proposed scheme can keep stable when hyper parameters vary within a reasonable range.

V. CONCLUSIONS

In this paper, a generic node classification scheme based on multimodality fusion has been explored. It incorporates knowledge learned from network structure and attribute information to classify each node in social networks, which is beneficial for implementing content distribution in D2D communications. More importantly, the proposed scheme adopting SGD optimization strategy is scalable and naturally parallelized, which can be applied to large-scale social networks. Furthermore, the scheme directly takes low-dimensional structure features as the input of multimodality fusion model, hence it can be applied to most of purely structure-based network embedding methods, such as DeepWalk, node2vec, Line and so on.

REFERENCES
