# Node Classification in Dynamic Social Networks

Yulong Pei, Jianpeng Zhang, George H.L. Fletcher, and Mykola Pechenizkiy

Department of Mathematics and Computer Science
Eindhoven University of Technology, Eindhoven, the Netherlands
{y.pei.1,j.zhang.4,g.h.l.fletcher,m.pechenizkiy}@tue.nl

**Abstract.** Classifying nodes in networks is different from traditional classification tasks since the *i.i.d.* assumption does not hold. In a dynamic scenario, nodes/edges may change during time which makes node classification more difficult. There have been a number of studies on node classification in networks in recent years but one limitation exists: how to make use of the temporal information. In this paper, we propose the *dynamic Factor Graph Model* (dFGM), which is an extension of Factor Graph Models, for the problem of node classification in dynamic social networks. *dFGM* can capture not only node attributes and correlations but also the temporal information for node classification. We conduct experiments on a real-world data set and the experimental results demonstrate the effectiveness of our proposed method.

## 1 Introduction

Network structures are ubiquitous nowadays and more and more data can be organized in networks with dependency relationships. Generally, nodes in networks can be associated with labels and these labels may come in many forms, e.g., demographic labels, interests, affiliations, etc. Assigning labels to unlabeled nodes in the graph is the *node classification problem.* However, the increasing number of the network applications and the complicated relationships between graph nodes have made the labels of the graph data expensive and/or difficult to obtain. Therefore, the problem of node classification in networks has attracted extensive attention recently.

Different from traditional classification tasks, the independent and identically distributed (*i.i.d.*) assumption does not hold for node classification in networks and methods should take the structure dependency into account. There have been a number of studies on node classification in networks in recent years [7, 9] and these methods can can be categorized into two types [1]: (1) methods based on iterative application of traditional classifiers using structural properties as features and (2) methods which propagate the labels via random walks. However, there is one major limitation in existing studies. In specific, most of these studies focus on static networks. In fact, many real-world networks are dynamic and nodes/edges in the networks may change during time. In such dynamic scenario, temporal information can also play an important role in classifying nodes.

There are some methods which have been proposed to classify nodes in dynamic networks [2, 8]. Li et al. propose a method which can learn the latent feature representation and capture the dynamic patterns [2]. However, this method requires data from all the historical snapshots to classify nodes in next snapshot while in practice some labels in previous data may be missing or incorrect. Yao et al. uses SVM to classify nodes in each snapshot and combines the support vector from last snapshot and current training data for classification [8]. But this operation depends heavily on the performance of SVM and only using support vector from previous snapshot may also lose useful dynamic information.

Aiming to overcome the limitations, in this paper we propose the *dynamic Factor Graph Model* (dFGM) for node classification in dynamic social networks. In detail, the dynamic graph data is organized in the format of a series of graph snapshots and to model the graph snapshots, three types of factors, named node factor, correlation factor and dynamic factor, are designed in the *dFGM* based on node features, node correlations and temporal correlations, respectively. Node factor and correlation factor can capture the global and local properties of the graph structures while the dynamic factor can make use of the temporal information. To validate the effectiveness of the *dFGM*, a real-world data set DBLP is used for the experiments.

The main contributions of our work can be summarized as follows:

– We propose the *dynamic Factor Graph Model* (dFGM) for node classification in dynamic social networks and this model can capture node attributes, correlations and temporal information.
– We evaluate the proposed *dFGM* on a real-world data set and the experimental results demonstrate the effectiveness of our model compared with other methods on two evaluation metrics: *accuracy* and *error* in probability.

The rest of this paper is organized as follows. Section 2 introduces the related work and Section 3 formally defines the problem. Section 4 introduces the proposed *dynamic Factor Graph Model*. And then in Section 5 we discuss the experiments and analysis. Finally, in Section 6 we draw the conclusions.

## 2 Problem Definition

In this section, several necessary definitions are introduced and then the formal definition of the node classification problem is presented. In this paper, we assume edges to be undirected and also assume the nodes are fixed and the edges may change over time.

**Definition 1** ***Partially labeled network***. *Given a fixed finite non-empty label set $R$, a partially labeled network is $G_L = \{V_L, V_U, E, X, r\}$ where (1) $V_L$ is a set of labeled nodes and $V_U$ is a set of unlabeled nodes with $V_L \cup V_U = V$ and $V_L \cap V_U = \emptyset$; (2) $E$ is the set of edges, i.e., $E \subseteq V \times V$; (3) $X$ is an attribute matrix associated with nodes in $V$ where each row corresponds to a node $v$, each column an attribute, and an element $x_{ij}$ denotes the value of the $j^{th}$ attribute of node $v_i$; and (4) $r$ is a mapping function which maps each labeled node to a label, i.e., $r : V_L \rightarrow R$.*

**Definition 2** ***Partially labeled dynamic network.*** *Let $V$ be a finite set of nodes, a partially labeled dynamic network $\mathcal{G} = \{G^t | t = 1, ..., T\}$ consists of a series of graph snapshots $G^t = \{V_L^t, V_U^t, E^t, X^t, r^t\}$, where each snapshot $G^t$ is a partially labeled network as defined in Definition 1, and $V_L^t \cup V_U^t = V$.*

Given a partially labeled dynamic network $\mathcal{G}$, our goal is to infer the labels of all the unlabeled nodes in the network. Formally,

**Problem 1** ***Node classification in dynamic networks.*** *Given a partially labeled dynamic network $\mathcal{G} = \{G^t | t = 1, ..., T\}$, learn a set of predictive functions $\{f^1, ..., f^T\}$ where for $1 \leq t \leq T$, $f^t : \mathcal{V} \rightarrow R$ such that (1) $\forall v \in V_L^t$, $f^t(v) = r(v)$, and (2) $\forall v \in V_U^t$, $f^t(v) \in R$ is the correct label assignment.*

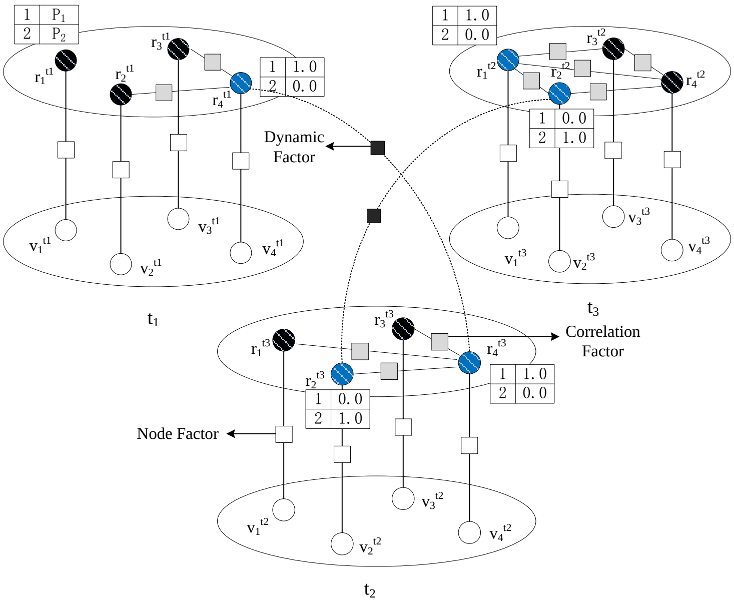

1 P₁
2 P₂
$r_1^{t1}$ $r_2^{t1}$ $r_3^{t1}$ $r_4^{t1}$
1 1.0
2 0.0
$v_1^{t1}$ $v_2^{t1}$ $v_3^{t1}$ $v_4^{t1}$
$t_1$
Dynamic Factor
1 1.0
2 0.0
$r_1^{t2}$ $r_2^{t2}$ $r_3^{t2}$ $r_4^{t2}$
1 0.0
2 1.0
$v_1^{t3}$ $v_2^{t3}$ $v_3^{t3}$ $v_4^{t3}$
$t_3$
Correlation Factor
$r_1^{t3}$ $r_2^{t3}$ $r_3^{t3}$ $r_4^{t3}$
1 0.0
2 1.0
1 1.0
2 0.0
Node Factor
$v_1^{t2}$ $v_2^{t2}$ $v_3^{t2}$ $v_4^{t2}$
$t_2$

**Fig. 1.** A four-node example in three snapshots of the *dFGM*. The white circles in the lower layers denote the nodes, the colored circles in the upper layers denote the labels and the squares are the factors. For these colored circles in the upper layer, the blue circles mean that these nodes are labeled and the black ones are unlabeled. The white, grey and black squares denote the node, correlation and dynamic factors, respectively.

## 3 Dynamic Factor Graph Model

In this section, we present the *dynamic Factor Graph Model* (dFGM), which builds on the traditional factor graph models, for node classification in dynamic social networks. Intuitively the class/label of a node in a dynamic social network will be determined by three factors including: (1) node attributes, i.e., the node's global and local characteristics at current time step. This factor corresponds to the node features extracted from the network; (2) node correlations, i.e., the relationships or interactions between nodes in a snapshot. This factor corresponds to the social relations of nodes in the networks; (3) previous labels. We assume that the label of a node will not change abruptly. This factor can capture the dynamic information in the network. Based on these intuitions, we propose the *dFGM* which consists of three factors, named node factor, correlation factor and dynamic factor, and they correspond to the node attributes, node correlations and previous labels, respectively. In detail, these three factors are defined as follows.

**Node factor** $g(r_i, \boldsymbol{x}_i)$. This factor represents the posterior probability of the label $r_i$ given the feature $\mathbf{x}_i$ of node $v_i$.

**Correlation factor** $c(r_i, N(r_i))$. This factor reflects the correlation between nodes, where $N(r_i)$ is the set of correlated labels to $r_i$. There are multiple ways to define the set of correlated labels and in this study, $N(r_i)$ denotes the labels of neighbors of node $v_i$.

**Dynamic factor** $d(r_i^t, r_i^{t-1})$. This factor denotes the correlation between the labels of one node in two consecutive snapshots.

An example of the *dFGM* with three factors is shown in Fig 1. In this example, there are two labels: 1 and 2. For the labeled nodes, the probability values of the ground label will be 1.0 or 0.0. For the unlabeled nodes, the probability values will be real numbers. Based on all the factors introduced above, the joint distribution of labels $R$ given the graph $\mathcal{G}$ can be defined as

$$P(R|\mathcal{G}) = \prod_t \prod_i g(r_i, \boldsymbol{x}_i) c(r_i, N(r_i)) d(r_i^t, r_i^{t-1}) \tag{1}$$

These factors can be instantiated in different ways. We choose the exponential-linear function to instantiate the factors because it can simplify the model learning. In detail, the node factor is defined as

$$g(r_i, \boldsymbol{x}_i) = \frac{1}{Z_1} \exp\{\alpha^T \boldsymbol{\phi}(r_i, \boldsymbol{x}_i)\} \tag{2}$$

where $Z_1$ is the normalizing factor, $\alpha$ is the weighting vector, and $\boldsymbol{\phi}$ is a vector of feature function. Similarly, the edge factor is defined as

$$c(r_i, N(r_i)) = \frac{1}{Z_2} \exp\{\sum_{r_j \in N(r_i)} \beta^T \mathbb{I}_{corr}(r_i, r_j)\} \tag{3}$$

where $Z_2$ is the normalizing factor, $\beta$ is the weighting vector, and $\mathbb{I}_{corr}$ is the indicator function for node correlations and defined as

$$\mathbb{I}_{corr}(r_i, r_j) = \begin{cases} 0, & \text{if } e_{ij} \notin E \\ 1, & \text{if } e_{ij} \in E \end{cases} \tag{4}$$

Then dynamic factor is defined in the same way

$$d(r_i^t, r_i^{t-1}) = \frac{1}{Z_3} \exp\{\gamma^T \mathbb{I}_{dyn}(r_i^t, r_i^{t-1})\} \tag{5}$$

where $Z_2$ is the normalizing factor and $\gamma$ is the weighting vector. $\mathbb{I}_{dyn}(r_i^t, r_i^{t-1})$ is the indicator function for the dynamic information and defined as

$$\mathbb{I}_{dyn}(r_i^t, r_i^{t-1}) = \begin{cases} 0, & \text{if } r_i^t \neq r_i^{t-1} \\ 1, & \text{if } r_i^t = r_i^{t-1} \end{cases} \tag{6}$$

To learn this model, we write the joint probability defined in Eq (1) as

$$P(R|\mathcal{G}) = \frac{1}{Z}\prod_t \prod_i \exp\{\theta^T s_i^t\} = \frac{1}{Z}\exp\{\theta^T \sum_t \sum_i s_i^t\} = \frac{1}{Z}\exp\{\theta^T \mathbf{S}\} \tag{7}$$

where $Z = Z_1 Z_2 Z_3$ is the normalizing factor, $\theta$ is the parameter configuration, i.e., $\theta = (\alpha, \beta, \gamma)$ and $\mathbf{S}$ is the concatenation of the factor functions, i.e., $\mathbf{S} = (\boldsymbol{\phi}(r_i, \boldsymbol{x}_i)^T, \mathbb{I}_{corr}(r_i, r_j)^T, \mathbb{I}_{dyn}(r_i^t, r_i^{t-1})^T)^T$. Thus, model learning is to estimate the parameter configuration $\theta$. To solve this problem, we use the labeled data to infer the unknown labels. In specific, we use $R|R^L$ to denote the predicted labels inferred from the known labels and define the log-likelihood function $\mathcal{O}(\theta)$ as

$$\mathcal{O}(\theta) = \log p(R^L|\mathcal{G}) = \log \sum_{R|R^L} \frac{1}{Z}\exp\{\theta^T \mathbf{S}\} \tag{8}$$

$$= \log \sum_{R|R^L} \exp\{\theta^T \mathbf{S}\} - \log Z = \log \sum_{R|R^L} \exp\{\theta^T \mathbf{S}\} - \log \sum_R \exp\{\theta^T \mathbf{S}\}$$

Then gradient descent method is used to solve this optimal problem:

$$\frac{\partial \mathcal{O}(\theta)}{\partial \theta} = \frac{\partial(\log \sum_{R|R^L} \exp\{\theta^T \mathbf{S}\} - \log \sum_R \exp\{\theta^T \mathbf{S}\})}{\partial \theta} \tag{9}$$

$$= \frac{\sum_{R|R^L} \exp\{\theta^T \mathbf{S}\} \cdot \mathbf{S}}{\sum_{R|R^L} \exp\{\theta^T \mathbf{S}\}} - \frac{\sum_R \exp\{\theta^T \mathbf{S}\} \cdot \mathbf{S}}{\sum_R \exp\{\theta^T \mathbf{S}\}} = \mathbb{E}_{p_\theta(R|R^L, \mathcal{G})}\mathbf{S} - \mathbb{E}_{p_\theta(R, \mathcal{G})}\mathbf{S}$$

Since the graphical structure in *dFGM* can be arbitrary and may contain cycles, we use Loopy Belief Propagation (LBP) [3] for the model learning in this paper similar to [6] due to its effectiveness in handling graphs with cycles. The learning algorithm is summarized in Algorithm 1.

## 4 Experiments

We conduct experiments to validate the performance of *dFGM* on a subset of DBLP[1] data set. Conferences from six research communities, including artificial intelligence and machine learning, algorithm and theory, database, data mining, computer vision, information retrieval, have been extracted. In specific, we extract the co-author relations in these conferences from 2001 to 2010 and data in each year is organized in a graph snapshot. Each author represents a node in the network and if two authors collaborated on a paper, there will be an edge between these two nodes. The features are extracted from each snapshot using DeepWalk [4] due to its generalization in graph mining tasks.

[1] `http://dblp.uni-trier.de/xml/`

**Algorithm 1** Model learning for dFGM

```
Input: learning rate $\eta$
Output: learned parameters
  while not converge do
    Calculate $\mathbb{E}_{p_\theta(R|R^L,\mathcal{G})}\mathbf{S}$ and $\mathbb{E}_{p_\theta(R,\mathcal{G})}\mathbf{S}$ using LBP
    Calculate the gradient $\nabla_\theta$ of $\theta$ according to Eq. (9)
    Update parameters $\theta$ with the learning rate $\eta$ according to $\theta_{new} = \theta_{old} - \eta\nabla_\theta$
  end while
```

To compare our proposed *dFGM* with existing methods, three types of baseline methods have been used:

*Feature-based classification.* We use the Logistic Regression (LR) and Support Vector Machine (SVM) as the baseline in the feature-based classification.

*Link-based classification.* Two methods have been employed in the link-based classification type. The first method is majority voting method with dynamic information (MV+dynamic). In detail, if a node is labeled in previous snapshot, the predicted label is copied from previous one. Otherwise, the node is labeled by majority voting from neighbors in current snapshot. The second one is the collective classification (CC) [5]

*Factor graph models (FGM).* To validate the effectiveness of the temporal information, we also compare *dFGM* with FGM using only the features (FGM_feat) and FGM using both features and correlations (FGM_corr).

Note that features used in these methods are the same extracted using DeepWalk and the correlations used here are same to the link-based classification methods.

### 4.1 Evaluation Metrics

Two types of evaluation metrics have been used in the experiments: *accuracy* and *error* in probability. The *accuracy* is defined as $accuracy = \frac{n}{N}$ where $n$ is the number of instances correctly classified (the label with highest probability matches the ground-truth label), and $N$ is the total number of instances in the test data.

To better evaluate the performance, we also use another evaluation metric, namely the *error* in probability. This metric is beneficial in two aspects: (1) it can match the output of *dFGM* which is the probability of labels; (2) it can evaluate the prediction of multiple labels. The *error* in probability is defined as

$$error = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{c}|\hat{p}_i^j - p_i^j| \tag{10}$$

where $N$ is the number of instances in the test data and $c$ is the number of labels. $\hat{p}_i^j$ and $p_i^j$ are the predicted probability and ground probability of label $j$ for user $i$, respectively. The ground probability of label $j$ for user $i$ is the ratio of the number of papers published by user $i$ in community $j$ to the total number of papers published by user $i$.

**Table 1.** Comparison of node classification performance in DBLP data set.

| Methods | | *accuracy* | *error* |
|---|---|---|---|
| Feature-based classification | LR | 0.3157±0.0031 | ——— |
| | SVM | 0.4983±0.0029 | 1.2651±0.0027 |
| Link-based classification | MV+dynamic | 0.5049+0.0011 | 0.9068+0.0009 |
| | CC | 0.7935+0.0033 | 0.8642±0.0101 |
| Factor Graph Models | FGM_feat | 0.2684±0.0024 | 1.4917±0.0003 |
| | FGM_corr | 0.8360±0.0328 | 0.7865±0.0062 |
| | dFGM | **0.8410±0.0058** | **0.7360±0.0073** |

### 4.2 Results

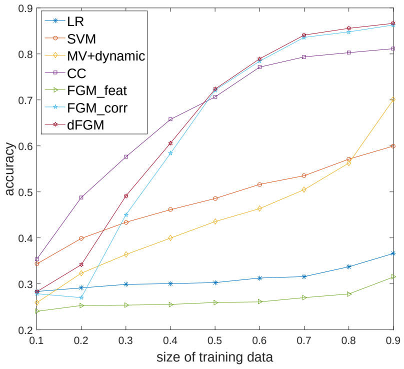
The performance of *dFGM* and other methods are shown in Table 1 and we use 70% data as the training set and 30% as the test set. From the results, some conclusions can be drawn: (1) the *dFGM* outperforms other methods in both evaluation metrics which shows the effectiveness of our proposed model and the importance of the dynamic information; (2) since FGM is used to model correlations in graphs, if the correlation information is removed, i.e., in FGM_Feat, the performance will be extremely poor even compared with traditional classification methods, e.g., SVM; (3) link-based classification methods (MV+dynamic and CC) perform better than feature-based methods (LR and SVM), and it demonstrates the importance of correlations in graph classification problem.

It is worth noting that the improvement on *accuracy* is very small. This is because in *accuracy* calculation, the predicted labels are the labels with maximum probability. For example, assume **CV** is the correct label and the probability of label **CV** is 0.8 predicated by model A and 0.95 predicted by model B, although model B performs better (it gives a more precise prediction), A and B have the same predicted label for *accuracy* metric.

Furthermore, we analyze the influence of size of training data in *dFGM*. The size of training data is set from 10% to 90% and the results are shown in Fig. 2 and Fig. 3. Overall, better results can be obtained when more training data is given and they demonstrate the robustness of the *dFGM*. Moreover, note that when the size of training data is relatively small (e.g., less than 50% in Fig 2 and less than 30% in Fig 3), the performance of *dFGM* is not good because estimates of correlation and dynamic information become less reliable with decreased training set size which will influence the performance of *dFGM*.

## 5 Conclusions

In this paper, we proposed the *dFGM* method to classify nodes in dynamic social networks. To capture the temporal information, graph factors based on node attributes, node correlations and dynamic information are integrated in *dFGM*. Experiments have been conducted on a real-world data set which demonstrate the effectiveness of our method. We also analyzed the influence of feature dimension and size of training data. As future work, we will take the UGC information



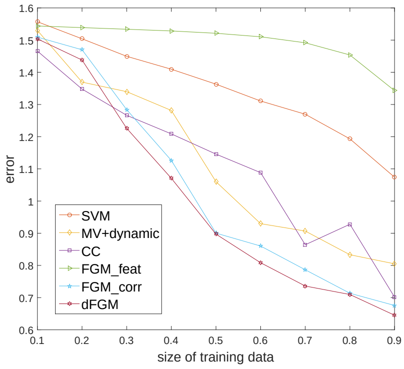


**Fig. 2.** *Accuracy* vs. size of training data.

**Fig. 3.** *Error* vs. size of training data.

into consideration for node classification in the dynamic scenario. In addition, with rapid increase of network size, it will be interesting to study more effective and efficient method for larger scale networks.

**Acknowledgement** The research of Pei is supported by the Netherlands Organisation for Scientific Research (NWO).

## References

1. Smriti Bhagat, Graham Cormode, and S Muthukrishnan. Node classification in social networks. In *Social network data analytics*, pages 115–148. Springer, 2011.
2. Kang Li, Suxin Guo, Nan Du, Jing Gao, and Aidong Zhang. Learning, analyzing and predicting object roles on dynamic networks. In *ICDM*, pages 428–437. IEEE, 2013.
3. Kevin P Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. In *UAI*, pages 467–475. Morgan Kaufmann Publishers Inc., 1999.
4. Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *KDD*, pages 701–710. ACM, 2014.
5. Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93, 2008.
6. Wenbin Tang, Honglei Zhuang, and Jie Tang. Learning to infer social ties in large networks. In *Machine Learning and Knowledge Discovery in Databases*, pages 381–397. Springer, 2011.
7. Huan Xu, Yujiu Yang, Liangwei Wang, and Wenhuang Liu. Node classification in social network via a factor graph model. In *Advances in Knowledge Discovery and Data Mining*, pages 213–224. Springer, 2013.
8. Yibo Yao and Lawrence Holder. Scalable svm-based classification in dynamic graphs. In *ICDM*, pages 650–659. IEEE, 2014.
9. Yuchen Zhao, Guan Wang, Philip S Yu, Shaobo Liu, and Simon Zhang. Inferring social roles and statuses in social networks. In *KDD*, pages 695–703. ACM, 2013.