Uncovering the Formation of Triadic Closure in Social Networks

Zhanpeng Fang and Jie Tang
Department of Computer Science and Technology, Tsinghua University
Tsinghua National Laboratory for Information Science and Technology (TNList)
Jiangsu Collaborative Innovation Center for Language Ability, Jiangsu Normal University, China
fzp13@mails.tsinghua.edu.cn, jietang@tsinghua.edu.cn

Abstract

The triad is one of the most basic human groups in social networks. Understanding factors affecting the formation of triads will help reveal the underlying mechanisms that govern the emergence and evolution of complex social networks. In this paper, we study an interesting problem of *decoding triadic closure* in social networks. Specifically, for a given closed triad (a group of three people who are friends with each other), which link was created first, which followed, and which link closed. The problem is challenging, as we may not have any dynamic information. Moreover, the closure processes of different triads are correlated with each other. Our technical contribution lies in the proposal of a probabilistic factor graph model (DeTriad). The model is able to recover the dynamic information in the triadic closure process. It also naturally models the correlations among closed triads. We evaluate the proposed model on a large collaboration network, and the experimental results show that our method improves the accuracy of decoding triadic closure by up to 20% over that of several alternative methods.

1 Introduction

Group formation—the process by which people come together, seek new friends, and develop communities—is a central research issue in social science [Backstrom *et al.*, 2006]. A triad is a group of three people. It is one of the most basic human groups in social networks. Understanding the formation of triads can help reveal the complex and subtle mechanism that governs the dynamics of all social networks. In an undirected network, the formation of a *closed triad* $\Delta = (A, B, C)$ consists of two major steps: (1) Open triad: the three persons (A, B, C) are connected by two undirected links—e.g., e_{AB} and e_{AC} (formally, there is a path between any two persons in the triad); (2) Closed triad: any two persons are connected by an undirected link; i.e., we have e_{BC} in the above example. Uncovering the mechanism underlying the triadic closure process can benefit many applications, for example to explain the development of social communities and to predict the evolution of network structures [Newman, 2001].

Analysis of triad formation has attracted tremendous interest from both academic and industrial communities. Roughly speaking, prior work has studied how different patterns of triad formation influence the dynamic evolution of networks. For example, Milo et al. [2002] defined the recurring significant patterns of interconnections as "network motifs" and emphasized their importance. They found that triadic closure patterns can used to distinguish different types of networks, e.g., transportation networks and online social networks. Grindrod et al. [2012], and Kossinets and Watts [2006], focused on how the evolution of networks has been affected by the process of triadic closure. Opsahl [2011] used triadic closure to redefine global and local clustering coefficients. Sintos and Tsaparas [2014] used triadic closure to characterize social tie strength. Another thread of research studies how closed triads develop from open triads. Romero and Kleinberg [2010] studied the problem of triadic closure and developed a methodology based on preferential attachment for studying the directed triadic closure process. Lou et al. [2013] investigated how a reciprocal link was developed from a parasocial relationship and how the relationships further developed into triadic closure on a Twitter dataset. Huang et al. [2014] studied how closed triads are formed in a Chinese microblogging service. Dong et al. [2014] investigated how social triadic relationships are maintained by people with different demographic profiles. Zhang et al. [Zhang *et al.*, 2015] investigated how "following" links will trigger the formation of other neighboring links.

Problem and Solutions. In this paper, from a different perspective, we study a novel problem of *decoding triadic closure* in social networks. Suppose we are only given a static network $G = (V, E)$, with a goal of uncovering how each closed triad was formed step by step, i.e., which link was firstly created, which followed, and which one closed the triad. At high level, the problem can be viewed as a reverse engineering of the triadic closure process; but it is actually more challenging. First, in our setting, we do not have any dynamic information. This is the case in many mining tasks due to various reasons such as privacy. Second, technically, the closure processes of different triads are correlated with each. For example, the closure of $\Delta_1 = (A, B, C)$ may significantly increase the closure probability of $\Delta_2 = (A, B, D)$.

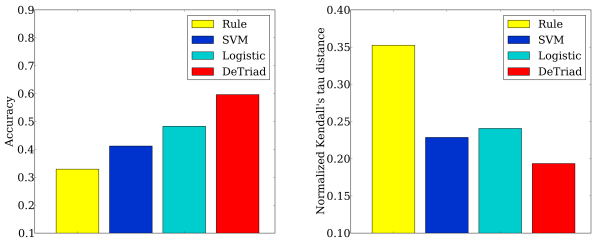

Figure 1: Performance comparison between DeTriad and the comparative methods. (in terms of accuracy and normalized Kendall's tau distance).

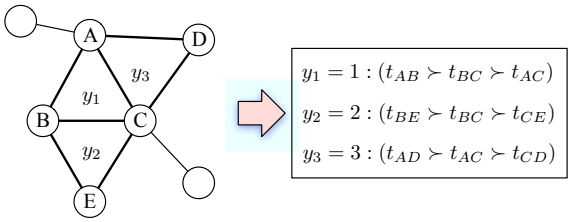

Figure 2: Illustration of decoding triadic closure. t_{AB} denotes the time when the link e_{AB} was created. y indicates a partial order of the time stamps associated with the three links in a closed triad.

To deal with these challenges, we propose a probabilistic factor graph model (DeTriad). The model is very general and flexible. It is able to recover the dynamic information in the triadic closure process. It can also incorporate different features into the model to help solve the task. More importantly, the model naturally models correlations among closed triads. We evaluate the proposed model on a large collaboration network consisting of over 1 million authors and 4 million coauthor relationships. There are in total 5,913,455 closed triads in the network. We consider the year when a collaboration started as the time information associated with each relationship. Figure 1 shows the performance comparison of different methods in terms of accuracy and normalized Kendall's tau distance [Kendall, 1938] on the dataset. It can be seen that our method improves the accuracy of decoding triadic closure by up to 20%, compared with several alternative methods. We also evaluate different strategies for modeling correlations among triads. All codes and data used in this paper are publicly available.[1]

Organization. In the rest of this paper, we first formulate the problem, and then present the proposed model and describe the algorithm for learning the model. After that, we give the experimental results, and finally, conclude the work.

2 Problem Definition

Let $G = (V, E)$ denote a static social network, where $V = \{v_1, v_2, ..., v_n\}$ represents a set of users and $E \subseteq V \times V$ represents a set of links connecting those users. Let $e_{ij} \in E$ represent a relationship between user v_i and user v_j. To begin with, we give the definitions of *closed triad* and *open triad* in a static social network.

Definition 1 ***Closed Triad:*** *For three users $\Delta = (A, B, C)$, if there is a link between every two users, i.e., $e_{AB}, e_{BC}, e_{AC} \in E$, then we say that Δ is a closed triad.*

Definition 2 ***Open Triad:*** *For three users $\Delta = (A, B, C)$, if we have and only have two links among them, e.g., $e_{AB}, e_{AC} \in E \wedge e_{BC} \notin E$, then we call the triad Δ as an open triad.*

In a static social network G, for each closed triad Δ, our general goal is to recover the time when each link was formed. More precisely, it is to associate each edge e_{ij} with a time stamp t_{ij}. However, the problem is extremely challenging, as the space of the time stamp might be large and continuous. In this paper, we focus on detecting the partial order of the time stamps associated with the three links of a closed triad. Specifically, for a closed triad $\Delta = (A, B, C)$, one example of the detected partial order could be $(t_{AB} \succ t_{AC} \succ t_{BC})$. Formally, we have the following problem definition.

Problem 1 Decoding Triadic Closure: *Given a social network $G = (V, E)$, we associate a variable y_i to each closed triad $\triangle_i$ to represent the partial order of time stamps associated with the three links in $\triangle_i$. Our goal is to use some available labeled information to train a function f, so that, for an unlabeled $\triangle$, we can predict its corresponding partial order y, i.e.,*

$$f : (\{\triangle\}^U | G, Y^L) \rightarrow Y^U$$

where Y^L indicates a set of variables corresponding to the available labeled triads for training, and $\{\triangle\}^U$ indicates a set of unlabeled triads; Y^U is the set of predicted results by function f corresponding to the unlabeled $\{\triangle\}^U$.

In an undirected network, there are six possible orders of the time stamps for a closed triad. Figure 2 shows an example. It is worth noting that the orders of different triads might be correlated. When designing the function f, it is necessary to consider the correlation. In addition, the order not only depends on some kinds of "local" information associated with the three users in the triad $\triangle$, but also is closely related to the topological position of the triad in the whole network G. The problem becomes more complicated for a directed network. In this paper, we will first focus on the undirected network. The study of the directed network is one of our ongoing projects and will be reported elsewhere.

3 DeTriad: Decoding Triadic Closure

In this section, we start with some basic ideas to solve the problem of decoding triadic closure and then present the proposed model. Based on the definition in previous section, in a straightforward way, the problem can be cast as a multiple classification problem. In particular, each closed triad is viewed as an instance, and features are defined for each instance. Then we can use labeled data to train a classification model and apply the trained model for classifying unlabeled

[1] http://arnetminer.org/decodetriad.

triads. However, such a method cannot capture correlations among different triads. Moreover, it is unclear how to incorporate the topological structure information into the model to help decode the triadic closure process.

In this paper, we propose a new probabilistic factor graph model, referred to as DeTriad, to address these technical challenges. Compared to the traditional classification model, the proposed DeTriad model has three advantages. First, the model is very general and flexible. It is very easy to incorporate any kinds of features into the model to solve the task. Second, the model is able to model correlations among closed triads. In addition, the model can easily support semi-supervised learning. This is important, as in a social network, it is very likely that we have only partial labeled information (some labeled information for part of the network). The technical challenge is how to combine the labeled and the related unlabeled structure information to design a more accurate decoding model.

Modeling. For a given network $G = (V, E)$, we first extract all closed triads as instances in our problem, and for each instance $\Delta_i = (v_a, v_b, v_c)$, we define a feature vector $\boldsymbol{x}_i$, and associate a variable $y_i \in \mathcal{Y}$ to it to represent its decoding result, where $\mathcal{Y}$ is the decoding space. As mentioned before, for an undirected network, there are six possible values for the variable y_i. Additionally, in a network, we may have some labeled triads (we know their decoding results) and also many other triads without labels. For those triads without labels, we denote $y =?$, which is actually what we need to infer. We use $\boldsymbol{X}$ to denote all the features defined for all instances (closed triads) and use Y to denote the set of variables $\{y\}$. Given this, we construct a factor graph model as follows.

A factor graph presents a general way to describe the (undirected) graphical model, with more emphasis on the factorization of the distribution [Kschischang *et al.*, 2001]. In a factor graph, variables ($\boldsymbol{x}$ and y) are denoted as vertices in the graphical model, and the graphical structure of the factor graph represents the correlations between variables. According to the graphical structure, the probability distribution over the graph can be factorized as a collection of functions defined on the *cliques* of the graph. A clique c is a fully connected subset of the variables Y_c in the graph. In our problem, we consider the correlation between two related decoding variables y_i and y_j as a clique.

Now, we explain how we use the factor graph model to model the triadic closure decoding problem. Given the input G, $\boldsymbol{X}$ and the corresponding Y, we can define the joint distribution over the decoding results Y given G and $\boldsymbol{X}$ as

$$P(Y|\boldsymbol{X}, G) = \prod_{\Delta_i} f(y_i|\boldsymbol{x}_i) \prod_{i \sim j} h(y_i, y_j), \tag{1}$$

where $i \sim j$ represents triads Δ_i and Δ_j are correlated with each other, for example (A, B, C) and (A, C, D) in Figure 2. The joint probability has two types of factor functions.

- **Local factor:** $f(y_i|\boldsymbol{x}_i)$ represents the posterior probability of the decoding result, given all features defined for triad Δ_i;
- **Correlation factor:** $h(y_i, y_j)$ represents the correlation between the decoding results of Δ_i and Δ_j.

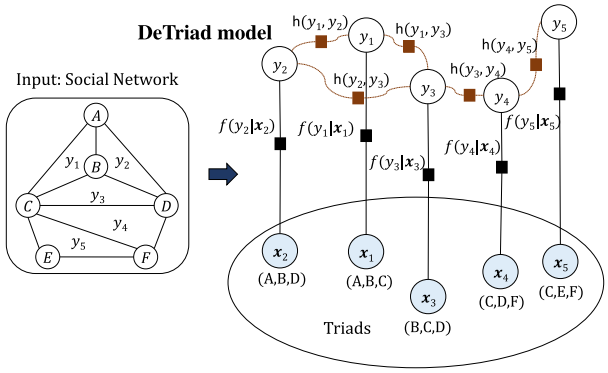

Figure 3: Example of the DeTriad model. Notation y_i indicates the decoding results of Δ_i; $\boldsymbol{x}_i$ denotes local feature vector of Δ_i; $f(y_i|\boldsymbol{x}_i)$ and $h(y_i, y_j)$ respectively represent local factor and correlation factor in the proposed model.

The two factors can be instantiated in different ways, reflecting our prior knowledge for different applications. In this paper, we give a general definition for the two types of factors. For the local factor, we define it as

$$f(y_i|\boldsymbol{x}_i) = \frac{1}{Z_1} \exp\{\sum_{k=1}^{d} \alpha_k f_k(x_{ik}, y_i)\}, \tag{2}$$

where Z_1 is a normalization factor; d is the number of features defined for triad Δ_i; $f_k(x_{ik}, y_i)$ is the k^{th} feature we defined for Δ_i, and α_k is its corresponding weight, to be estimated by the learning algorithm.

Modeling Correlations. Straightforwardly, the correlation factor can be modeled in a Markov random field; thus by the fundamental Hammersley-Clifford theorem [Hammersley and Clifford, 1971], we can define the correlation factor as:

$$h(y_i, y_j) = \frac{1}{Z_2} \exp\{\sum_{k} \mu_k h_k(y_i, y_j)\}, \tag{3}$$

where Z_2 again is a normalization factor, $h_k(y_i, y_j)$ represents the k^{th} correlation feature function we defined between two triads Δ_i and Δ_j; and μ_k is the weight of the k^{th} correlation feature function.

However, as shown in Figure 2, the correlations among multiple triads are complicated. The challenge is how to capture the complicated correlations in the proposed model. We present two methods based on the triadic closure process: *Synchronous* method and *Asynchronous* method.

Synchronous method: Given two closed triads $\Delta_1 = (A, B, C)$ and $\Delta_2 = (D, B, C)$, Δ_2 is more likely to be closed by edge e_{BC} if Δ_1 is closed by the edge. Similarly, if e_{BC} is the k^{th} $(k = 1, 2, 3)$ formed edge in Δ_1, it is also more likely to be the k^{th} formed edge in Δ_2. Thus, for two closed triads Δ_i and Δ_j that share the edge e, we define the following factor function between y_i and y_j:

$$h_(y_i, y_j) = \frac{1}{Z_3} \exp\{\sum_{k} \mu_k \cdot I_k(y_i, y_j)\}, \tag{4}$$

where $k \in \{1,2,3\}$, μ_k is the model parameter of the correlation factor, Z_3 is the normalization term, and $I_k(y_i, y_j)$ is defined as an indicator function that indicates whether the shared edge e is the k^{th} formed edge in y_i and y_j.

Asynchronous method: we extend this approach to the case where the shared edge is not the k^{th} formed edge in both decoding results. We still take $\Delta_1 = (A, B, C)$ and $\Delta_2 = (D, B, C)$ for an example. If Δ_1 is closed by edge e_{BC}, the likelihood that e_{BC} is the second formed edge in Δ_2 is intuitively higher than the likelihood that it is the first formed edge in Δ_2. Similarly, if e_{BC} is the k_1^{th} formed edge in Δ_1, it could correlate with the case where e_{BC} is the k_2^{th} formed edge in Δ_2. We use a factor function to model the correlation. Specifically, for two closed triads Δ_i and Δ_j that share an edge e, we define the following factor function between y_i and y_j:

$$h(y_i, y_j) = \frac{1}{Z_4} \exp\{\sum_{k_i,k_j} \mu_{k_i,k_j} \cdot I_{k_i,k_j}(y_i, y_j)\}, \tag{5}$$

where $k_i, k_j \in \{1,2,3\}$, μ_{k_i,k_j} is the model parameter for this type of correlation factor, Z_4 is the normalization term, and $I_{k_i,k_j}(y_i, y_j)$ is defined as an indicator functions that indicates whether the shared edge e is the k_i^{th} formed edge in y_i and the k_j^{th} formed edge in y_j.

Finally, by integrating Eqs. 1-5, we obtain the following joint probability

$$\begin{aligned} P(Y|\boldsymbol{X}, G) = \frac{1}{Z} \exp\{&\sum_{\Delta_i} \sum_{k=1}^{d} \alpha_k f_k(x_{ik}, y_i) \\ &+ \sum_{i\sim j} \sum_{k} \mu_k h_k(y_i, y_j)\}, \end{aligned} \tag{6}$$

where Z is the normalization factor (also called the partition function).

Figure 3 shows an example of our proposed model. The left part is the input network, where we have six users and five closed triads—e.g., (A, B, C) and (A, B, D). In the decoding task, we associate a variable y_i with each closed triad. All features defined over Δ_i are denoted as $\boldsymbol{x}_i$. We have five local factors $f(y_i|\boldsymbol{x}_i), i = 1, 2, 3, 4, 5$. In addition, we also consider correlations among the triads. For example, the decoding result of (A, B, C) may have a strong correlation with the decoding result of (A, B, D). Given this, we build a correlation factor $h(.)$ between y_1 and y_2. The joint distribution over the whole set of random variables can be factorized as the product of all factors, i.e., $\prod_{i=1}^{5} f(y_i|\boldsymbol{x}_i) \cdot h(y_1, y_2)h(y_1, y_3)h(y_2, y_3)h(y_3, y_4)h(y_4, y_5)$.

Learning the DeTriad model is to estimate a parameter configuration $\boldsymbol{\theta} = (\{\alpha\}, \{\mu\})$ from the training data, which maximizes the log-likelihood objective function $\mathcal{O}(\boldsymbol{\theta}) = \log P(Y^L|\boldsymbol{X}, G)$.

Learning. One challenge to solving the objective function is that the input data is partially labeled. To calculate the normalization factor Z, one needs to sum up the likelihood of possible states for all nodes in the factor graph, including unlabeled nodes. To deal with this, we use the labeled data to infer the unknown labels. Here Y^L denotes the set of known labels, and $Y|Y^L$ denotes a full labeling configuration Y consistent with the known labels. Therefore, we have:

Input: network G, features $\boldsymbol{X}$, learning rate η
Output: estimated parameters $\boldsymbol{\theta}$

Initialize $\boldsymbol{\theta} \leftarrow \mathbf{0}$;
repeat
  Perform LBP to calculate $P(y_i|Y^L, \boldsymbol{X}, G)$, $P(y_i, y_j|Y^L, \boldsymbol{X}, G)$;
  Perform LBP to calculate $P(y_i|\boldsymbol{X}, G)$, $P(y_i, y_j|\boldsymbol{X}, G)$;
  Calculate the gradient ∇_{μ_k} of μ_k according to Eq. 8 (for α_k with a similar formula):

$$\begin{aligned} \nabla_{\mu_k} =& \mathbf{E}_{P_{\mu_k}(y_i,y_j|Y^L,\boldsymbol{X},G)}[h_k(y_i, y_j)] \\ &- \mathbf{E}_{P_{\mu_k}(y_i,y_j|\boldsymbol{X},G)}[h_k(y_i, y_j)] \end{aligned}$$

  Update parameter $\boldsymbol{\theta}$ with the learning rate η:

$$\boldsymbol{\theta}_{\text{new}} = \boldsymbol{\theta}_{\text{old}} + \eta \cdot \boldsymbol{\nabla}_{\boldsymbol{\theta}}$$

until *Convergence*;

Algorithm 1: Learning algorithm for the DeTriad model.

$$\begin{aligned} \mathcal{O}(\boldsymbol{\theta}) &= \log P(Y^L|\boldsymbol{X}, G) = \log \sum_{Y|Y^L} P(Y|\boldsymbol{X}, G) \\ &= \log \sum_{Y|Y^L} \{\sum_{\Delta_i} \sum_{k=1}^{d} \alpha_k f_k(x_{ik}, y_i) + \sum_{i\sim j} \sum_{k} \mu_k h_k(y_i, y_j)\} \\ &- \log \sum_{Y} \{\sum_{\Delta_i} \sum_{k=1}^{d} \alpha_k f_k(x_{ik}, y_i) + \sum_{i\sim j} \sum_{k} \mu_k h_k(y_i, y_j)\} \end{aligned} \tag{7}$$

To solve this objective function, we adopt a gradient descent method (or a Newton-Raphson method). We use μ as the example to explain how we learn the parameters. Specifically, we first write the gradient of each μ_k with regard to the objective function Eq. 7: (A similar gradient can be derived for parameter α_k.)

$$\begin{aligned} \frac{\partial \mathcal{O}(\boldsymbol{\theta})}{\partial \mu_k} =& \mathbf{E}_{P_{\mu_k}(y_i,y_j|Y^L,\boldsymbol{X},G)}[h_k(y_i, y_j)] \\ &- \mathbf{E}_{P_{\mu_k}(y_i,y_j|\boldsymbol{X},G)}[h_k(y_i, y_j)] \end{aligned} \tag{8}$$

Another technical challenge is that the graphical structure in the constructed factor graph model can be arbitrary and may contain cycles, which makes it intractable to directly calculate the marginal distribution $P_{\mu_k}(y_i, y_j|\boldsymbol{X}, G)$ and $P_{\mu_k}(y_i, y_j|Y^L, \boldsymbol{X}, G)$. A number of approximate algorithms can be considered, such as Loopy Belief Propagation (LBP) [Murphy *et al.*, 1999] and Mean-field [Xing *et al.*, 2002]. We chose Loopy Belief Propagation due to its ease of implementation and effectiveness. Specifically, we approximate the marginal distribution $P_{\mu_k}(y_i, y_j|\boldsymbol{X}, G)$ and $P_{\mu_k}(y_i, y_j|Y^L, \boldsymbol{X}, G)$ using LBP. With the marginal probabilities, the gradient can be obtained by summing over all correlation factors. It is worth noting that we need to perform the

LBP process twice in each iteration, once for estimating the marginal distribution of unknown variables $y_i =?$ and again for estimating the marginal distribution over all variables y_i. Finally, with the obtained gradient, we update each parameter with a learning rate η. The learning algorithm is summarized in Algorithm 1.

Decoding. With the estimated parameters $\boldsymbol{\theta}$, we can calculate the marginal probabilities of unknown variables $\{y_i =?\}$. However, the problem of obtaining the exact solution of the marginal probabilities remains intractable. Again, we utilize Loopy Belief Propagation to approximate the solution, i.e., to calculate the marginal distribution of unknown variable $P(y_i|Y^L, \boldsymbol{X}, G)$ and finally obtain $y_i^\star$ for each closed triad $\triangle_i$ by $y_i^\star = \arg\max_{y_i} P(y_i|Y^L, \boldsymbol{X}, G)$.

4 Experiments

4.1 Experimental Setup

Dataset. We perform our experiments in the coauthor network dataset. The dataset is from ArnetMiner [Tang *et al.*, 2008]. Specifically, we collected 1,910,979 publications from ArnetMiner from 1995 to 2014.[2] Then we constructed a coauthor network that contains 1,145,632 authors and 4,322,998 coauthor relationships. For each coauthor relationship, the formation time is set as the earliest year that the two authors started to coauthor a paper. We enumerate all triads in the network and obtain 5,913,455 closed triads. In some closed triads, it might be that a triad has two coauthor relationships formed in the same year; in which case we cannot determine the partial order of the creation time of the three links. Therefore, for evaluation, we only keep those triads that have clear partial orders, and are finally left with 631,465 closed triads with 200,891 authors in the network for our experiments.

Feature Definition. Besides correlation features, we define three categories of local features: demography features, interaction features, and social effect features.

Demography features: These represent the number of publications and the number of collaborators for each author in a triad.

Interaction features: These represent the number of common publications, the number of common conferences, the number of common citations, and the number of mutual citations for every pair of authors in a triad.

Social effect features: These are defined based on the PageRank score and the structural hole spanner score [Lou and Tang, 2013] in the coauthor network of each author in a triad.

Comparison Methods. To quantitatively evaluate the effectiveness of the proposed model and the methods for comparison, we randomly divide the closed triads into two subsets of even size: training and test. The training set is used to train the model and the test set is used to evaluate the decoding performance. We compare our model with three alternative methods.

[2]http://arnetminer.org/citation

Table 1: Decoding triadic closure performance.

Algorithm	Spearman	Kendall	Accuracy
Rule	0.4604	0.3525	0.3293
SVM	0.3205	0.2286	0.4121
Logistic	0.3379	0.2407	0.4830
DeTriad-A	0.3060	0.2190	0.5550
DeTriad	**0.2716**	**0.1935**	**0.5964**

Rule: We design a rule-based method to determine the decoding results by assuming that for each edge in a triad, the more interactions it has between its two ends, the earlier the edge is formed. We use the number of coauthor papers as the number of interactions on an edge in the dataset.

SVM: Uses the same local features associated with each triad as our model to train a support vector machine classification model and then uses the model to predict the decoding results in the test data. For SVM, we use svm-perf [Joachims, 2006].

Logistic: Similar to the SVM method. The only difference is that it uses logistic regression model as the classification model. We employ liblinear [Fan *et al.*, 2008] here.

DeTriad: The proposed model that uses a synchronous method to define correlation factors.

DeTriad-A: Differs from DeTriad in that it uses an asynchronous method to define correlation factors.

Evaluation Metrics. We evaluate the performance of different methods in terms of accuracy, normalized Spearman's footrule distance [Spearman, 1904], and normalized Kendall's tau distance [Kendall, 1938]. Let Δ be a triad with three edges e_1, e_2, e_3 sequentially formed. Let σ be a decoding result of Δ. It can be viewed as a permutation on $\{e_1, e_2, e_3\}$, and let $\sigma(e_i)$ denote the rank of edge e_i in σ. The normalized Spearman's footrule distance is given by $F(\sigma) = \frac{1}{4}\sum_{i=1}^{3}|i - \sigma(e_i)|$, which measures the total element-wise displacement. The normalized Kendall's tau distance is given by $K(\sigma) = \frac{1}{3}\sum_{(i,j):i>j}[\sigma(e_i) < \sigma(e_j)]$, which measures the total number of pairwise inversions. For both metrics, we report the average score across all test triads.

All algorithms are implemented in C++, and all experiments are performed on an x64 machine with E5-4650 2.70GHz Intel Xeon CPU (with 64 cores) and 1TB RAM. The operating system is Ubuntu 12.04.

4.2 Decoding Performance

Prediction Performance. We list the performance results for the different methods in Table 1. It can be seen that the proposed model DeTriad and DeTriad-A outperform the other comparison methods on all three measures. In terms of accuracy, DeTriad achieves a 23.47% improvement over Logistic, and 44.72% over SVM. In terms of normalized Spearman's footrule distance and normalized Kendall's tau distance, DeTriad also achieves up to 15% improvement over the other methods. The advantages of the DeTriad model are as follows. First, DeTriad leverages the correlation information among closed triads. Second, DeTriad incorporates the partially observed information into the model, and can thus bet-

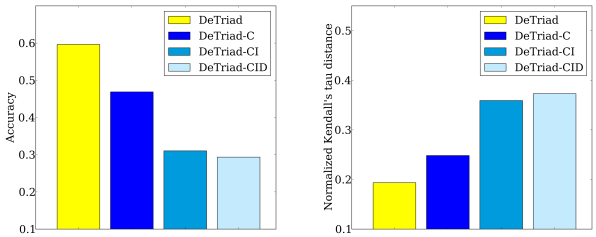

Figure 4: Factor contribution analysis.

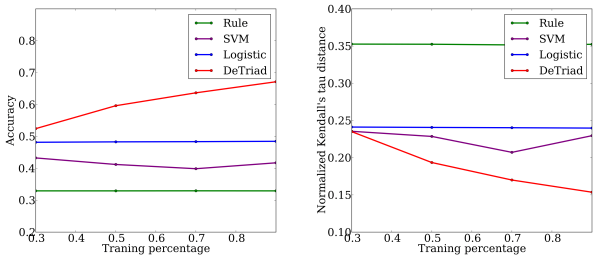

Figure 5: Performance of decoding triadic closure with different percentages of labeled data.

ter fit the data. Moreover, we can see that DeTriad shows better performance than DeTriad-A ($+7.38\%$ in terms of accuracy), which demonstrates that the synchronous method is a better strategy to model the correlations among triadic closures, and the asynchronous method introduces noise to the model.

Factor Contribution Analysis. In this section, we examine the contribution of four different categories of features: demography features (D), interaction features (I), social effect features (S), and correlation features (C). We first rank the individual categories by successively removing each category of features from our model and evaluating the decrease in prediction performance. Here, a larger decrease means greater predictive power. We then remove them one by one according to the order of their prediction power. We denote DeTriad-C as removing correlation features and DeTriad-CI as also removing interaction features. DeTriad-CID indicates the model with the demography features removed. As shown in Figure 4, we observe a significant performance decrease when ignoring each category of features. This indicates that our method works well by combining different categories of features, and each category of features contributes to performance improvement.

Training/Test Ratio. We provide further analysis on the effects of training ratio on decoding performance. Figure 5 shows the prediction results when varying the percentage of labeled closed triads in the dataset. For DeTriad, we see a rising trend as the training set increases. This indicates the positive effect of training data size on decoding triadic closures for our proposed model. The smooth lines of the baseline methods reveal the limited contribution of training data

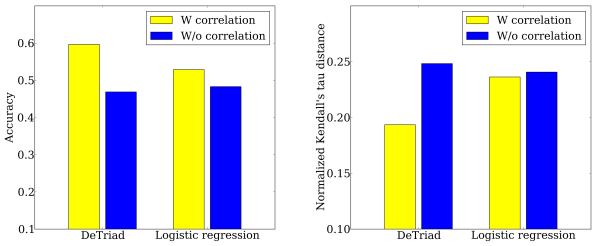

Figure 6: Performance comparison with different correlation features.

size to the decoding task if only using local features. We can see that obvious improvements in decoding accuracy can be obtained by our proposed DeTriad model with different sizes of training datasets.

Effect of Correlation Factors. We further evaluate the effectiveness of using correlation factors to model the correlation information among closed triads. We compare the method of using correlation factors with a feature-based approach. We define a new set of features, where the feature values are calculated as follows: for each edge e in a triad, we calculate the number of labeled triads which also contain edge e, and have e as the k^{th} $(k = 1, 2, 3)$ formed edge in the formation processes. For a given triad Δ, this set of features can capture the correlation between Δ and the labeled triads. We incorporate this new set of features into logistic regression model, and compare it with DeTriad. Figure 6 shows the prediction results of DeTriad and logistic regression with and without the correlation information. We can see that in both evaluation metrics, the improvement of DeTriad by adding correlation factors is significantly higher than that of logistic regression by adding correlation features. This indicates that our method works well by employing correlation factors to model the correlation information among closed triads.

5 Conclusion

In this paper, we study an interesting problem of decoding triadic closure in social networks. We propose a probabilistic factor graph model, DeTriad, which can recover the dynamic information in the triadic closure process. It integrates local features, correlations among closed triads, and partially labeled information into a uniform model. We validate the model on a large collaboration network. Experimental results show that the model effectively solves the task of decoding triadic closure and outperforms several alternative methods by up to 20% in terms of accuracy.

Acknowledgements. The work is supported by the National High-tech R&D Program (No. 2014AA015103), National Basic Research Program of China (No. 2014CB340506, No. 2012CB316006), Natural Science Foundation of China (No. 61222212), the Tsinghua University Initiative Scientific Research Program (20121088096), a research fund supported by Huawei Inc. and Beijing key lab of networked multimedia.

References

[Backstrom *et al.*, 2006] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD'06*, pages 44–54, 2006.

[Dong *et al.*, 2014] Yuxiao Dong, Yang Yang, Jie Tang, Yang Yang, and Nitesh V. Chawla. Inferring user demographics and social strategies in mobile social networks. In *KDD'14*, pages 15–24. ACM, 2014.

[Fan *et al.*, 2008] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[Grindrod *et al.*, 2012] Peter Grindrod, Desmond J Higham, and Mark C Parsons. Bistability through triadic closure. *Internet Mathematics*, pages 402–423, 2012.

[Hammersley and Clifford, 1971] J. M. Hammersley and P. Clifford. Markov field on finite graphs and lattices. *Unpublished manuscript*, 1971.

[Huang *et al.*, 2014] Hong Huang, Jie Tang, Sen Wu, Lu Liu, et al. Mining triadic closure patterns in social networks. In *WWW'14*, pages 499–504, 2014.

[Joachims, 2006] Thorsten Joachims. Training linear svms in linear time. In *KDD'06*, pages 217–226. ACM, 2006.

[Kendall, 1938] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, pages 81–93, 1938.

[Kossinets and Watts, 2006] Gueorgi Kossinets and Duncan J Watts. Empirical analysis of an evolving social network. *Science*, 311(5757):88–90, 2006.

[Kschischang *et al.*, 2001] Frank R. Kschischang, Brendan J. Frey, and Hans Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE TOIT*, 47:498–519, 2001.

[Lou and Tang, 2013] Tiancheng Lou and Jie Tang. Mining structural hole spanners through information diffusion in social networks. In *WWW'13*, pages 825–836, 2013.

[Lou *et al.*, 2013] Tiancheng Lou, Jie Tang, John Hopcroft, Zhanpeng Fang, and Xiaowen Ding. Learning to predict reciprocity and triadic closure in social networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(2):5, 2013.

[Milo *et al.*, 2002] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: simple building blocks of complex networks. *Science*, pages 824–827, 2002.

[Murphy *et al.*, 1999] Kevin P Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. In *UAI'99*, pages 467–475, 1999.

[Newman, 2001] Mark EJ Newman. Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2):025102, 2001.

[Opsahl, 2011] Tore Opsahl. Triadic closure in two-mode networks: Redefining the global and local clustering coefficients. *Social Networks*, 2011.

[Romero and Kleinberg, 2010] Daniel Mauricio Romero and Jon M Kleinberg. The directed closure process in hybrid social-information networks, with an analysis of link formation on twitter. In *ICWSM'10*, 2010.

[Sintos and Tsaparas, 2014] Stavros Sintos and Panayiotis Tsaparas. Using strong triadic closure to characterize ties in social networks. In *KDD'14*, pages 1466–1475, 2014.

[Spearman, 1904] Charles Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 15(1):72–101, 1904.

[Tang *et al.*, 2008] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *KDD'08*, pages 990–998, 2008.

[Xing *et al.*, 2002] Eric P Xing, Michael I Jordan, and Stuart Russell. A generalized mean field algorithm for variational inference in exponential families. In *UAI'02*, pages 583–591, 2002.

[Zhang *et al.*, 2015] Jing Zhang, Zhanpeng Fang, Wei Chen, and Jie Tang. Diffusion of "following" links in microblogging networks. *IEEE Transaction on Knowledge and Data Engineering (TKDE)*, 2015.