Collective Inference for Network Data with Copula Latent Markov Networks

Rongjing Xiang
Departments of Computer Science
Purdue University
rxiang@cs.purdue.edu

Jennifer Neville
Departments of Computer Science and Statistics
Purdue University
neville@cs.purdue.edu

ABSTRACT

The popularity of online social networks and social media has increased the amount of *linked* data available in Web domains. Relational and Gaussian Markov networks have both been applied successfully for classification in these relational settings. However, since Gaussian Markov networks model joint distributions over *continuous* label space, it is difficult to use them to reason about uncertainty in discrete labels. On the other hand, relational Markov networks model probability distributions over *discrete* label space, but since they condition on the graph structure, the marginal probability for an instance will vary based on the structure of the subnetwork observed around the instance. This implies that the marginals will not be *identical* across instances and can sometimes result in poor prediction performance. In this work, we propose a novel latent relational model based on *copulas* which allows use to make predictions in a discrete label space while ensuring identical marginals and at the same time incorporating some desirable properties of modeling relational dependencies in a continuous space. While copulas have recently been used for descriptive modeling, they have not been used for collective classification in large scale network data and the associated conditional inference problem has not been considered before. We develop an approximate inference algorithm, and demonstrate empirically that our proposed Copula Latent Markov Network models based on approximate inference outperform a number of competing relational classification models over a range of real-world relational classification tasks.

1. INTRODUCTION

With the emergence of online social networks and social media, there are many web domains which consist of not only a set of data instances, but also the observed relationships (e.g., hyperlinks) that naturally encode statistical dependencies among the data instances. Collective classification methods (see e.g. [26]) aim to exploit such dependencies among the instances in order to improve predictive performance compared to conventional learning methods that assume independent and identically distributed (IID) instances. For example, in online social networks, it may be of interest to predict a user's missing profile information, such as their political view or gender. Collective classification models predict the missing profile labels in the network *jointly* based on the observed labels on other related nodes in the network, as well as other observed attributes.

Existing collective classification approaches can be grouped into two main categories. The first category of methods attempt to summarize the network information as "relational" features and then combine them with each instance's local features while learning a conventional classifier, e.g. [12, 28]. While this feature-construction type of approach can be used with any arbitrary IID learning algorithm, recent research has shown that it is often more accurate to learn, and reason with, the dependencies among the class labels of linked examples. This has motivated research in a second category of methods, which use Markov networks, or other probabilistic graphical models, to represent the *joint* distribution of attributes in relational data, e.g. [29, 25, 22]. These joint models have been shown to perform well when applied for *relational* inference, and thus have been popular in the statistical relational learning community.

As a general class of models, *relational Markov networks*[1] provide a principled framework to represent a joint distribution over the class labels $\mathbf{Y}$ in a network, conditioned on the graph structure G and the observed attributes $\mathbf{X}$: $p(\mathbf{Y}|\mathbf{X}, G)$. Although several methods have been developed to learn relational Markov networks from relational data, and empirical performance when the models are applied for prediction is often good, the models have several limitations. First, since the model is conditioned on the relational structure, the prediction for an instance i will vary based on the observed structure of G. Although, the relational structure is indeed the very thing that relational models are trying to exploit, the marginal probability for i: $p(y_i|\mathbf{x}_i) = \sum_{\mathbf{y}\setminus y_i} p(\mathbf{y}|\mathbf{x}_i)$, will vary when it is inferred with the same probabilistic model p but using different observed subgraphs G from the underlying domain (which often comprises a single, infinite network). This can lead to poor behavior in the model, and is a reason why in some cases relational models perform worse than IID classifiers that just model $p(y_i|\mathbf{x}_i)$. Moreover, since it models the full joint distribution, the relational Markov network representation does

[1]Relational Markov networks are discriminative probabilistic models that can be viewed as conditional random fields applied to heterogeneous graphs.

WSDM'13, February 4–8, 2013, Rome, Italy.

not allow us to explicitly specify the form of the marginal probability distribution (e.g., $p(y_i|\mathbf{x}_i) = f(y_i, \mathbf{x}_i, \boldsymbol{\theta})$), which makes it difficult to impose any constraint of equality on the marginals.

Second, [34] showed that the relative performance of relational Markov networks with parameters estimated by different methods, i.e., maximum likelihood (MLE) and maximum pseudolikelihood (MPLE), is not consistent as the amount of labeled instances varies in the test network. This is because when the model family is misspecified (which is typically the case for real network data), discrete Markov network models estimated with MLE-type approaches are often afflicted by *under propagation* error when the test network is abundantly labeled, while those estimated with MPLE-type approaches are often afflicted by *over propagation* error when the test network is sparsely labeled.

In contrast to relational Markov networks which model probability distributions over the discrete label space $\mathbf{y}$, *Gaussian Markov networks* model joint distributions over the *continuous* label space [37, 16]. Due to their harmonic property, Gaussian Markov networks usually exhibit better performance when the model is largely misspecified and in practice algorithms based on Gaussian Markov networks have been shown to perform well in relational domains. Many simple *collective classification* algorithms can be interpreted as instantiations of a Gaussian Markov network that is combined with a particular inference procedure for prediction—for example, iterative weighted voting [13] and personalized page rank [36, 30, 10]. Although these Gaussian Markov network models often provide a solid baseline that is hard to outperform, the models also have several limitations. First, since they model continuous attributes, they do not provide probability distributions over the discrete label space. Although thresholding approaches are typically used to transform the continuous predictions into discrete labels, the continuous representation makes it hard to (1) argue about uncertainty in the discrete labels or justify the thresholding of predictions, and (2) to formally analyze the dependence structure of the network data from the model. Moreover, the relational attributes are not typically incorporated into the model, i.e., $p(\mathbf{y}|G)$ is used instead of $p(\mathbf{y}|\mathbf{X}, G)$. While there is some work that attempts to move beyond this to learn kernel functions between linked instances [38], to date, there is not a systematic framework for learning Gaussian Markov networks with attributes in relational settings.

To address these issues and combine the strengths of Gaussian Markov networks' continuous representation with the strengths of relational Markov network methods for learning the dependencies among relational attributes, in this work we propose a novel latent relational model based on copulas [21]. The copula approach allows us to make the intuitively reasonable assumption of identical marginals an explicit constraint in the model. This provides a mechanism to remove the "independent" but not the "identical" from the conventional IID assumption when we model dependencies in relational datasets. While such a model is conceptually desirable for network data, to the best of our knowledge, copulas have not been applied to predictive tasks in real world network datasets.

Copulas were first explored in statistics and have attracted attentions from the machine learning community recently [6, 11, 32, 24] due to their flexibility in modeling the dependence among random variables. However, to date, the primary focus on copulas has been for descriptive modeling such as density estimation, clustering and feature selection, instead of directly applying the copula ideas into discriminative or predictive modeling. On the other hand, the fact that copulas enable the separation of dependence from marginals facilitates new approaches to jointly model networked instances while at the same time incorporating the general advantage of IID models that can use the attributes on instances. Due to the distributional transformations involved in copulas, conditional inference—which is the central task in our setting of collective classification in network data—is not straightforward and has not previously been studied. The contributions of our work is thus two-fold:

- First, we develop a copula latent Markov network model for network data. The model differs from existing copula based models in that it reasons about the dependence among the *latent* label tendency, instead of directly among the labels. Moreover, the proposed model elegantly extends a general class of IID classifiers based on the reformulation of IID probabilistic models from a latent variable perspective.

- Second, we design a message passing based scheme to solve the conditional inference problem in the copula model. This approximate inference approach is scalable for collective classification tasks in networks of tens of thousands of nodes, and is furthermore amenable to parallel implementations for even larger scale applications.

The rest of the paper is organized as follows. In Section 2, we begin with a description of a latent variable model reformulation of a general class of IID classifiers to motivate our proposed approach. Building on this formulation, in Section 3 we derive a copula latent Markov network (CLMN) model which unifies the homogeneous marginal classifiers and the dependence among the latent effects into a joint probabilistic model over the network data. In section 4, we design a message passing inference procedure for the collective classification task using CLMNs. In section 5 the CLMN model is compared with several alternative collective classification methods, and we show empirically that our copula-based approach outperforms several state-of-art relational classifiers over a range of real-world relational classification scenarios. Section 6 reviews related work and Section 7 concludes the paper.

2. LATENT VARIABLE VIEW OF PROBABILISTIC CLASSIFIERS

Our proposed model aims to extend an *IID* probabilistic classifier into a *collective* relational classifier, while still retaining all but the *independence* property of the original classifier. We first outline a latent variable view of probabilistic classifiers, to motivate and describe the approach. While a latent variable interpretation is well known for the probit regression model, analogous interpretations apply to a more general class of probabilistic classifiers.

Consider a binary classification task: $\mathbf{x} \mapsto y$, where $\mathbf{x} \in \mathcal{R}^d, y \in \{-1, 1\}$. Most probabilistic classifiers exploit the use of continuous, monotonically increasing functions $F(\eta) : \mathcal{R} \mapsto [0, 1]$ to map the observed attributes $\mathbf{x}$ to a class label y. Typically η is further modeled as a function of the input

attributes $\mathbf{x}$ and a parameter vector $\mathbf{w}$: $\eta := \eta(\mathbf{x}; \mathbf{w})$, resulting in a binary classification model: $p(y=1) = F(\eta(\mathbf{x}; \mathbf{w}))$. Any generalized linear model with binary response fits into this class of models. For example, when $\eta(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T\mathbf{x}$, $F(\cdot) = \frac{1}{1+e^{-\cdot}}$, we obtain the logistic regression model.

When the function F satisfies a further condition:

$$\lim_{a \leftarrow -\infty} F(a) = 0, \lim_{a \rightarrow +\infty} F(a) = 1$$

there always exists a continuous random variable $z \in \mathcal{R}$ for which F is the CDF. Let $f := F'$, i.e., the corresponding PDF of the distribution F. If we assume that the distribution is symmetric around zero (i.e., the CDF satisfies $F(a) = 1 - F(-a)$), we can then relate the class label y to the *sign* of a latent variable z sampled from f with a location shift $\eta(\mathbf{x}; \mathbf{w})$. More specifically, the probabilistic classification model $p(y_i|\mathbf{x}_i)$ can be thought of as modeling the following generative process:

$$\begin{aligned} z_i &\sim p(z_i = z|\mathbf{x}_i = \mathbf{x}) = f(z - \eta(\mathbf{x}_i; \mathbf{w})), \\ y_i &= sign(z_i) \end{aligned} \tag{1}$$

The equivalence is clear to see when the latent variable z is integrated out: $p(y = 1) = \int_0^{+\infty} f(z - \eta(\mathbf{x}; \mathbf{w})) = 1 - F(-\eta(\mathbf{x}; \mathbf{w})) = F(\eta(\mathbf{x}, \mathbf{w}))$.

The corresponding graphical model for this interpretation of probabilistic classifiers is shown in Figure 1(a). As a concrete example, consider the case of logistic regression where $F(\cdot) = \frac{1}{1+e^{-(\cdot)}}$, $p(z) = f(z - \eta(\mathbf{x}, \mathbf{w})) = \frac{e^{-(z-\eta(\mathbf{x},\mathbf{w}))}}{(1+e^{-(z-\eta(\mathbf{x},\mathbf{w}))})^2}$. In this case, $p(z)$ is the PDF of a logistic distribution with mean $\eta(\mathbf{x}, \mathbf{w})$ and scale 1. As another example, in the case of the probit classifier, z is a normal random variable with mean $\eta(\mathbf{x}, \mathbf{w})$ and variance 1. With some abuse of notation, hereinafter, we will use f_i to denote $f(z_i - \eta(\mathbf{x}_i; \mathbf{w}))$, the PDF of z_i, and F_i to denote its CDF.

Here the latent variable z_i for each data instance i can be interpreted as the latent tendency for instance i to have a positive label y_i. Besides capturing discriminative information from the input attributes $\mathbf{x}_i$, z_i can also capture other effects that are not present in $\mathbf{x}_i$, but which also contribute to determining y_i. When no information other than the attributes $\mathbf{x}$ is available, these additional effects can only be modeled as *random* effects (as in the model above). However, in relational domains when links among data instances are observed, the correlation among the class labels of linked instances is an example of such an additional effect, and can thus be interpreted as dependencies among the latent variables $\mathbf{z}$. In short, this latent variable view provides a way to decouple the random effects from the deterministic effects of $\mathbf{x}$ on $\mathbf{y}$, so that we can focus on modeling the correlations among the random effects.

We will use this view to combine the two aspects of relational data—the across-instance *relational* dependence structure and the *instance* level dependence of y_i on $\mathbf{x}_i$—into a coherent probabilistic model, while respecting and making use of the typical assumption of homogeneous (i.e., identically distributed) data instances. Copulas provide an ideal solution for this, since they separate marginal distributions from the joint distribution. Using a copula approach, we can learn a single (identical) marginal model $p(z|\mathbf{x})$ tied across all data instances, while also incorporating the relational dependence in the joint distribution over $\mathbf{z}$. Furthermore, the latent variable representation allows us to incorporate the strengths of Gaussian Markov networks through the relational dependence among the continuous $\mathbf{z}$, while at the same time modeling the uncertainty in the discrete label space $\mathbf{y}$. Our modeling idea differs from previous copula based models [29, 25, 22] in that it is a discriminative model and can be applied directly to classification tasks. As a result, the associated conditional inference task has not been considered before and we develop approximate inference algorithms to make it tractable for large networks. We discuss our proposed model in detail next.

3. COPULA LATENT MARKOV NETWORKS

Assuming that we would like to use some joint probability model Φ of our choice to model the dependence structure among $z_1, z_2, \ldots, z_n$ in a network of n nodes, but each marginal z_i has to be from a specified distribution $F_i(z_i)$. Then based on Sklar's theorem, we can use Φ to model the dependencies among a set of auxiliary variables $t_1, t_2, \ldots, t_n$ and use a copula function to facilitate the transformation from Φ to a joint distribution $F(z_1, z_2, \ldots, z_n)$ with the desired marginals.

An n-copula is an n-dimensional distribution function with n univariate margins distributed as $U(0, 1)$. The following theorem is fundamental in copula modeling:

THEOREM 1 (SKLAR'S THEOREM.). *Let Φ be an n-dimensional distribution function with marginal distribution functions Φ_1, Φ_2, ..., Φ_n. Then there exists an n-copula C such that for all $(t_1, t_2, \ldots, t_n)$:*

$$\Phi(t_1, t_2, \ldots, t_n) = C(\Phi_1(t_1), \Phi_2(t_2), \ldots, \Phi_n(t_n)).$$

If $\Phi_1, \Phi_2, \ldots, \Phi_n$ are all continuous then C is unique.

Define $\Phi_i^{(-1)}(u) = \inf\{x : \Phi(x) \geq u\}$ as the quasi-inverse of Φ_i. Then as a result of Theorem 1, for all $(u_1, \ldots, u_n) \in [0, 1]^n$:

$$C(u_1, \ldots, u_n) = \Phi(\Phi_1^{(-1)}(u_1), \ldots, \Phi_n^{(-1)}(u_n))$$

While the copula formulation provides ways to study the joint CDF Φ (and later F), we will primarily work with the PDF ϕ (and f) for the benefit of applying well established dependence models and inference algorithms in machine learning. In the following corollary, we make precise what we mean by "a joint probability density over $z_1, z_2, \ldots, z_n$ with the required marginals f_i and the desired dependence structure ϕ."

COROLLARY 1. *Let $f_1, f_2, \ldots, f_n$ denote the marginal PDFs of continuous variables $z_1, z_2, \ldots, z_n \in \mathcal{R}$, and let $F_1, F_2, \ldots, F_n$ denote the corresponding marginal CDFs. Let $\phi(t_1, t_2, \ldots, t_n)$ be an arbitrary, continuous joint density function over n random variables $t_1, \ldots, t_n \in \mathcal{R}$, with the respective joint CDF denoted by Φ. Then:*

1. *There exist a unique copula function C over Uniform(0,1) variables and a unique joint distribution function F such that the marginal distributions of F are exactly F_i, and the joint distribution satisfies: $F(z_1, \ldots, z_n) = C(u_1, \ldots, u_n) = \Phi(\Phi_1^{(-1)}(u_1), \ldots, \Phi_n^{(-1)}(u_n))$, where $u_i = F_i(z_i)$.*

2. *Let $c := \frac{\partial^n C}{\partial u_1 \ldots \partial u_n}$, i.e., the* copula density. *Then the*

joint density function f of F is given as follows:

$$f(z_1, z_2, \ldots, z_n) = c(u_1, \ldots, u_n) \prod_{i=1}^{n} f_i(z_i) \qquad (2)$$
$$= \frac{\phi(t_1, t_2, \ldots, t_n)}{\prod_{i=1}^{n} \phi_i(t_i)} \prod_{i=1}^{n} f_i(z_i)$$

where $u_i = F_i(z_i)$ and $t_i = \Phi_i^{(-1)}(u_i)$.

PROOF. By Sklar's theorem, since Φ is continuous, there exists a unique C such that, for any instantiation $(t_1, t_2, \ldots, t_n)$, $\Phi(t_1, t_2, \ldots, t_n) = C(\Phi_1(t_1), \Phi_2(t_2), \ldots, \Phi_n(t_n))$. On the other hand, since z_i are continuous variables, the inverse CDF function $F_i^{(-1)}$ is uniquely defined. Therefore, we define a joint distribution function F over $\mathbf{z}$ point wise. For any $\mathbf{z} \in \mathcal{R}^n$, let $u_i = F_i(z_i)$, and let $F(z_1, \ldots, z_n) = C(u_1, \ldots, u_n)$. Then $F(z_1, \ldots, z_n) = C(u_1, \ldots, u_n) = \Phi(\Phi^{(-1)}(u_1), \ldots, \Phi^{(-1)}(u_n))$, i.e., C is the unique copula distribution function and F is the unique CDF that satisfy the required condition and we have shown part 1.

Now consider the PDF $f(z_1, z_2, \ldots, z_n) = \frac{\partial^n F}{\partial z_1 \partial z_2, \ldots, \partial z_n}$. By the chain rule,

$$f = \frac{\partial^n F}{\partial z_1 \partial z_2, \ldots, \partial z_n} = \frac{\partial^n C}{\partial u_1 \ldots \partial u_n} \prod_{i=1}^{n} \frac{\partial F_i}{\partial z_i} = c \prod_{i=1}^{n} f_i$$

Moreover,

$$\frac{\partial^n C}{\partial u_1 \ldots \partial u_n} \prod_{i=1}^{n} f_i = \frac{\partial^n \Phi}{\partial t_1 \ldots \partial t_n} \prod_{i=1}^{n} \frac{\partial \Phi_i^{-1}}{\partial u_i} \prod_{i=1}^{n} f_i$$
$$= \frac{\partial^n \Phi}{\partial t_1 \ldots \partial t_n} \prod_{i=1}^{n} \frac{1}{\phi_i(t_i)} \prod_{i=1}^{n} f_i = \frac{\phi(t_1, \ldots, t_n)}{\prod_{i=1}^{n} \phi_i(t_i)} \prod_{i=1}^{n} f_i$$

Therefore, $f(z_1, z_2, \ldots, z_n) = c(u_1, \ldots, u_n) \prod_{i=1}^{n} f_i(z_i) = \frac{\phi(t_1, t_2, \ldots, t_n)}{\prod_{i=1}^{n} \phi_i(t_i)} \prod_{i=1}^{n} f_i(z_i)$. Thus we obtain Equation (2). □

Now, we will use Equation (2), and combine it with Equation (1), to model the joint probability over the labels in a relational network. We can interpret the model with the following generative process. First, a joint set of auxiliary variables $t_1, t_2, \ldots, t_n$ is sampled from ϕ. Then the marginal CDFs Φ_i are used to transform each t_i to a uniform $[0, 1]$ variable u_i, and each quasi-inverse $F_i^{(-1)}$ is used to obtain z_i from u_i, i.e., $z_i = F_i^{(-1)}(u_i) = F_i^{(-1)}(\Phi_i(t_i))$. Finally the $y_1, y_2, \ldots, y_n$ can be computed directly from $z_1, z_2, \ldots, z_n$. This generative process is summarized as follows.

- Generate $(t_1, t_2, \ldots, t_n) \sim \phi$.
- Apply the transformation: $u_i = \Phi_i(t_i)$ so that $\mathbf{u}$ follows the copula distribution C corresponding to Φ.
- Apply the transformation: $z_i = F_i^{(-1)}(u_i)$.
- Label: $y_i = sign(z_i)$.

Combining Equation (2) with (1), we obtain the joint likelihood over $y_1, y_2, \ldots, y_n$ and $z_1, z_2, \ldots, z_n$:

$$p(y_1, \ldots, y_n, z_1, \ldots, z_n) \qquad (3)$$
$$= \frac{\phi(t_1, t_2, \ldots, t_n)}{\prod_{i=1}^{n} \phi_i(t_i)} \prod_{i=1}^{n} \left(f_i(z_i) I_{y_i = sign(z_i)} \right)$$

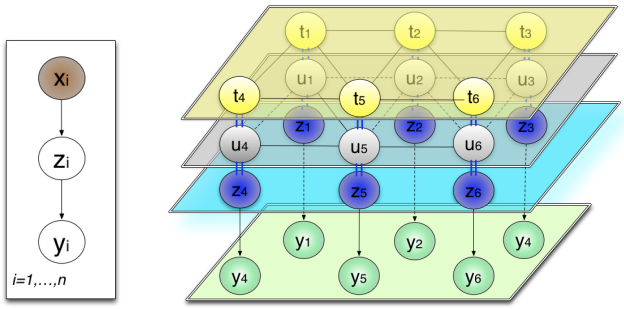


Figure 1: The two facets of a CLMN. (a) The graphical model representation of independent probabilistic classification, which is also the marginal facet of a CLMN. (b) The dependence facet of a CLMN.

where $t_i = \Phi^{(-1)}(u_i)$, $u_i = F_i(z_i)$, and I is the indicator function. i.e., $\delta = 1$ if $y_i = sign(z_i)$, otherwise $\delta = 0$. The modeling task is then divided into two subproblems: the dependence modeling ϕ over the auxiliary variables $t_1, t_2, \ldots, t_n$, where t_i is simply the distributionally transformed version of z_i, and the marginal modeling $f_i(z_i)$. In addition, we see from Corollary 1 that switching from the discrete label domain to the continuous latent variable has resulted in the additional advantage of the uniqueness property of the copula construction. We will choose F_i and f_i to be a model from the family described in Section 2, i.e., they represent the discriminative probabilistic model of the binary label and the latent continuous variable respectively, conditioned on the attributes $\mathbf{x}_i$ and parameterized by $\mathbf{w}$. In our experiments, we use a logistic model. Thus, the marginal of z_i follows a logistic distribution—which resembles the normal distribution but with a heavier tail.

Next we consider the problem of modeling the dependence over the auxiliary variables $t_1, t_2, \ldots, t_n$. The dependence structure over continuous variables can be conveniently modeled using a Gaussian Markov network (GMN) with node potentials and edge potentials as follows:

$$\phi(\mathbf{t}) = \frac{1}{Z} \exp\left(-\frac{1}{2} \left(\sum_{i,j:(i,j) \in E} (t_i - t_j)^2 \right) - \frac{\epsilon}{2} \sum_{i=1}^{n} t_i^2 \right)$$
$$= \frac{1}{Z} \exp\left(-\frac{1}{2} \mathbf{t}^T (\epsilon \mathbf{I} + \Delta) \mathbf{t} \right) \qquad (4)$$

where the normalization factor $Z = (2\pi)^{n/2} (|\epsilon \mathbf{I} + \Delta|)^{-1/2}$ and Δ is the graph Laplacian:

$$\Delta = \text{diag}(\text{Degree}_1, \text{Degree}_2, \ldots, \text{Degree}_n) - \mathbf{A}$$

where $\mathbf{A}$ is the adjacency matrix of the undirected data network or, in the case when the data network is a directed graph with adjacency matrix $\tilde{\mathbf{A}}$, let $\mathbf{A} = \frac{1}{2}(\tilde{\mathbf{A}} + \tilde{\mathbf{A}}^T)$, so that $\mathbf{A}$ is symmetric. Thus, ϕ defines a multivariable Gaussian distribution over $t_1, t_2, \ldots, t_n$ with precision matrix $\epsilon \mathbf{I} + \Delta$. The term $\epsilon \mathbf{I}$ is commonly adopted in practical implementation of GMNs for numerical stability. $\epsilon > 0$ ensures that the precision matrix is of full rank, so the Gaussian distribution is not intrinsic.

Summary: While the generic CLMN model is represented by Equation 2 and is described by the sampling process above Equation 3, the model specification in our implementation can be summarized as follows:

- *Dependence structure*—Joint PDF ϕ over latent variables $\mathbf{t}$:

$$\phi(\mathbf{t}) = \frac{1}{Z}\exp\left(-\frac{1}{2}\left(\sum_{i,j:(i,j)\in E}(t_i - t_j)^2\right) - \frac{\epsilon}{2}\sum_{i=1}^{n} t_i^2\right)$$

- *Marginal structure*—each PDF f is an independent classifier of latent variable z:

$$f_i(z_i|\mathbf{w}, \mathbf{x}_i) = \frac{e^{-(z_i - \mathbf{w}^T\mathbf{x}_i)}}{(1 + e^{-(z_i - \mathbf{w}^T\mathbf{x}_i)})^2}$$

- *Glue between joint dependence and marginal models*—marginal CDF transform:

$$\Phi_i(t_i) = u_i = F_i(z_i)$$

- *Classification*—discrete label is determined from latent variable sign:

$$y_i = sign(z_i)$$

Discussion of GMN dependence model.
Our choice of GMNs as the dependence model is inspired by its empirical success on relational collective classification tasks. The GMN provides a probabilistic modeling view of two popular collective classification models: iterative weighted voting with relaxed labels (WV) [13] and personalized page rank (PPR) [36, 10]. Based on the Neuman series expansion of the graph Laplacian (see e.g., the walk-sum analysis of GMNs by [16]), it can be shown that these approaches are related to GMNs and latent GMNs respectively. First, given a set of labeled instances L and their true labels $\mathbf{y}_L^*$, the MAP solution $p(y_i|\mathbf{y}_L^*)$ of the GMN model in Equation 4 results in label scores (for each unlabeled node i) that equal to those of the WV algorithm at convergence. Second, the PPR algorithm converges to the MAP of $p(y_i|\mathbf{y}_L^*)$ for a latent Gaussian Markov network (LGMN) that uses the normalized Laplacian $\tilde{\Delta}$ (instead of the Laplacian Δ in Equation 4). Each observation y_i is a noisy version of a latent variable, just like in CLMNs, but with a Gaussian noise. While the GMN reformulation of the WV method is well known to the relational learning community, it appears that the probabilistic modeling extension of PPR has not been explored before. Although the above LGMN reformulation of PPR considers noise in the observations, it does not make full use of such latent variable modeling because the actual observations y_i^* should binary instead of real-valued. Therefore, to improve it, one can simply modify the observation part $f(y_i|z_i)$ using a logistic or a probit function—indeed, this is the common approach taken when a Gaussian model is applied in classification tasks (e.g., in Gaussian process classifiers), and in our experiments we include a LGMN for comparison. Unlike PPRs, this LGMN model is no longer analytically solvable, and inference techniques similar to those for the CLMN model would need to be employed. On the other hand, they do not share the CLMN advantages that allow for easy incorporation of node attributes and learning the model using general independent classifiers.

4. USING CLMNS FOR COLLECTIVE CLASSIFICATION

We explore the utility of CLMNs in collective classification tasks, where a network of linked data instances $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$ is observed. A subset of the data instances, $j \in L$, are labeled and the goal is to infer labels for the remainder of the instances, $i \in U$.

4.1 Parameter learning

Before applying CLMNs for collective classification, we need to learn the parameters $\mathbf{w}$ in the marginal model f. Thanks to the marginal preserving property of copulas, parameter estimation for CLMNs can be simple and efficient.

A commonly explored learning approach for copula-based models consists of two stages (see e.g. [6, 4]). First, the marginal distribution is estimated as if we were learning from IID data instances. Second, fixing the estimated marginals, the dependence model is estimated. In the proposed CLMN model, since there is no parameter in the dependence model, only the first stage is involved, i.e., we only need to train the marginal classifier $p(y|\mathbf{x})$ using the observed class labels $y_i \in \{+1, -1\}$ and the attributes $\mathbf{x}_i$ of labeled instances (while ignoring the unlabeled instances) using the typical learning algorithm for the chosen classifier. In this work, we use an $\ell 2$ regularized logistic regression model with parameters $\mathbf{w}$, which we train by gradient-based optimization methods.

We note that this learning approach is also supported by statistical learning theory for dependent data (see e.g. [35]): Under reasonable mixing conditions—i.e., instances that are remote in the network are weakly dependent of each other—the learned marginal classifiers would converge to the optimal one within the model family as the number of labeled instances in the network increases. On the other hand, consider an alternative learning approach that maximizes the joint log-likelihood (conditioned on labeled instances) over the whole observed network. In application domains of our interest, typically only a single network (e.g., a subnetwork of the whole domain) is used for training and inference. There has been little theoretical analysis as to whether, or under which conditions, a joint approach to learning will converge in this setting—as the size of the single training/test network grows. Moreover, empirical results have suggested that such a joint approach is not robust when the model family is misspecified [5].

4.2 Approximate Inference

While learning is relatively well studied for copula based models in the literature, little attention has been paid to the problem of conditional inference, especially for large scale network data. Collective classification with the CLMN model corresponds exactly to the conditional inference problem, which is the central problem we attempt to address in developing CLMN algorithms. During collective classification, we need to apply the model with the estimated parameters $\hat{\mathbf{w}}$ to predict the label probabilities of the unlabeled data instances $i \in U$. Exact inference of the posterior is intractable.

For approximate inference in graphical models with con-

tinuous variables, there have been two commonly explored strategies: Nonparametric Belief Propagation (NBP) [27] Expectation Propagation (EP) [18]. While the former uses a nonparametric representation for local messages, the latter explores parametric forms to approximate the messages. For our CLMN model, we need to compute the CDFs and inverse CDFs of the approximate marginals of t_i, and the parametric approximation employed by EP makes this computation straightforward. In addition, EP has already been successfully applied to other large scale probabilistic inference problems such as Gaussian process classification (e.g. [8]). Therefore, we propose an approximation inference algorithm based on EP in this work. Since the developed algorithm uses a distributed message passing scheme, it is amenable to parallel implementation and is thus feasible for applications to large scale networks.

The inference task is to estimate $p(\mathbf{y}_i|\mathbf{y}_L, \mathbf{x}, \mathbf{w})$ for $i \in U$ given the observed labels $\mathbf{y}_L$, the attributes of all nodes $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, and the parameters $\mathbf{w}$.

$$p(y_i|\mathbf{y}_L, \mathbf{x}, \mathbf{w}) = \int_{z_i \in \mathcal{R}} p(y_i|z_i) p(z_i|\mathbf{y}_L, \mathbf{x}, \mathbf{w}) \mathrm{d}z_i$$
$$= \int_{\mathbf{t} \in \mathcal{R}^n} p\Big(y_i | z_i = F^{(-1)}(\Phi_i(t_i))\Big)\, p(\mathbf{t}|\mathbf{y}_L, \mathbf{x}, \mathbf{w}) \mathrm{d}\mathbf{t}$$

where the last step follows from the copula transformation. Using Equation (3), the posterior is:

$$p(\mathbf{t}|\mathbf{y}_L, \mathbf{x}, \mathbf{w}) \propto \phi(t_1, t_2, \ldots, t_n) \prod \frac{g_i(z_i)}{\phi_i(t_i)} \qquad (5)$$

where $z_i = F_i^{-1}(\Phi_i(t_i))$ and

$$g_i(z_i) = \begin{cases} f_i(z_i; \mathbf{w}, \mathbf{x}) & \text{if } i \in U \\ f_i(z_i; \mathbf{w}, \mathbf{x}) I_{y_i^* = sign(z_i)} & \text{if } i \in L \end{cases}$$

Exact inference of the posterior is intractable, so we develop an approximation inference algorithm based on Expectation Propagation (*EP*) [18].

Since $p(\mathbf{t}|\mathbf{y}_L)$ is intractable, we first proceed by finding an approximate distribution $q(\mathbf{t})$ for it. Specifically, we make a fully factored approximation $q = \prod_i q_i(t_i)$. Each approximation factor q_i is in the Gaussian form: $q_i(t_i; \lambda_i, h_i) = e^{-\frac{1}{2}\lambda_i t_i^2 + h_i t_i^2}$. The approximation factors are updated based on a message passing scheme. We denote the messages sent from node i to node j by $m_{i\to j}(t_j)$. The messages are also in Gaussian forms: $m_{i\to j}(t_j) = e^{-\frac{1}{2}\tilde{\lambda}_{i\to j} t_j^2 + \tilde{h}_{i\to j} t_j}$. Under the EP scheme, when updating each factor q_i, the KL divergence $KL(p||q)$ is minimized locally—which leads to the matching of first and second order moments in the cases where the approximation factors are Gaussian. We give the detailed message passing scheme in Algorithm 1. While the communications of messages between neighboring nodes (i.e., Steps 2a and 2c of Algorithm 1) are akin to those of loopy belief propagation in a Gaussian Markov network [31], the moment matching during incorporating the factor $\frac{g_i(z_i)}{\phi_i(t_i)}$ (i.e., Step 2b of the algorithm) cannot be done analytically, and numerical integration is needed. The required integral is of the form $\Gamma = \int \gamma(t) \frac{g_i(F_i^{-1}(\Phi_i(t_i)))}{\phi_i(t_i)} \mathcal{N}(t_i|\hat{\mu}_i, \hat{v}_i)$ where $\hat{\mu}_i = \hat{h}_i/\hat{\lambda}_i, \hat{v}_i = 1/\hat{\lambda}_i$, and $\gamma(t)$ is either t or t^2. This integral has a complicated shape, so care must be taken to obtain an accurate estimate. For labeled instances $i \in L$, we apply importance sampling to compute Γ: sample z_i from the truncated distribution F_i, where the left interval $(-\infty, \mathbf{w}^T\mathbf{x}_i)$ is truncated if $y_i^* = +1$, and otherwise the right interval $(\mathbf{w}^T\mathbf{x}_i, +\infty)$ is truncated. In practice we sample the respective truncated uniform variables u, transform it using inverse CDF to obtain samples of t_i, and then weight the samples using $\mathcal{N}(\hat{\mu}_i, \hat{v}_i)/\phi_i(t_i)$, which is of a Gaussian form. For unlabeled instances $i \in U$, we apply a Metropolis Hasting method due to [20]. In each step, we sample s from $\mathcal{N}(\hat{\mu}, \hat{v})$, and then move from the old sample t^{old} with a step size δ: $t^{\text{new}} = \delta s + \sqrt{1-\delta^2} t^{\text{old}}$, t^{new} is then accepted with probability $\min\left(1, \frac{f_i(F_i^{-1}(\Phi_i(t^{\text{new}})))\phi_i(t^{\text{old}})}{f_i(F_i^{-1}(\Phi_i(t^{\text{old}})))\phi_i(t^{\text{new}})}\right)$.

Algorithm 1 Approximate inference of $p(\mathbf{t}|\mathbf{y}_L)$ using message-passing EP

Input: Network structure, observed labels $\mathbf{y}_L^*$, the marginals f_i and ϕ_i for $i = 1, \ldots, n$.
Step 1: Initialize the messages $m_{i\to j}$ by $\tilde{h}_{i\to j} = 0, \tilde{\lambda}_{i\to j} = 0$.
for each iteration **do**
  for each instance $i = 1, 2, \ldots, n$ **do**
    Step 2a: Collecting messages from neighboring nodes (where ∂i denotes i's Markov blanket):

$$\hat{\lambda}_i = \sum_{j \in \partial i} \tilde{\lambda}_{j\to i}, \quad \hat{h}_i = \sum_{j \in \partial i} \tilde{h}_{j\to i}$$

    Step 2b: Compute $\lambda_i^{\text{new}}, h_i^{\text{new}}$ by matching the moments of $q_i(t_i; \lambda_i^{\text{new}}, h_i^{\text{new}})$ with those of $\frac{g_i(z_i)}{\phi_i(t_i)} q_i(t_i; \hat{\lambda}_i, \hat{h}_i)$
    Step 2c: Compute the messages to the neighboring nodes:

$$\tilde{\lambda}_{i\to j} = -\Delta_{ij}^2 / \lambda_{i\setminus j}^{\text{new}}, \quad \tilde{h}_{i\to j} = -\Delta_{ij} h_{i\setminus j}^{\text{new}} / \lambda_{i\setminus j}^{\text{new}}$$

    where $\lambda_{i\setminus j}^{\text{new}} = \lambda_i^{\text{new}} - \tilde{\lambda}_{j\to i}$, and $h_{i\setminus j}^{\text{new}} = h_i^{\text{new}} - \tilde{h}_{j\to i}$.

    Step 2d: Update q_i: $\lambda_i := \lambda_i^{\text{new}}, h_i := h_i^{\text{new}}$.
  end for
end for
Step 3: Return the mean μ_i and variance v_i of the pseudo-marginals q_i: $\mu_i = h_i/\lambda_i, v_i = 1/\lambda_i$.

After obtaining the posterior pseudo marginals q_i from Algorithm 1, we can use the variables z_i transformed from t_i for prediction. The full collective classification procedure using CLMNs is shown in Algorithm 2. In practice, we observe that Algorithm 1 converges to a good solution within a constant number of iterations. Since the complexity of each EP iteration is $O(|E|)$, the complexity of Algorithm 1 is $O(|E|)$. To find the marginals of the latent Markov network in Step 1 of Algorithm 2, we can compute the full covariance matrix, i.e., the inverse of the Laplacian (with ϵ added to the diagonal elements). Since the networks in our applications are usually sparse, Step 1 typically scales as $O(n^2)$, although it may be accelerated to $O(n)$ by using further approximations. For example, since only the marginal variance of each node in the latent Gaussian Markov network needs to be computed, the wavelet based approximation developed by [15] may be applied for this purpose.

Algorithm 2 Collective classification for unlabeled data instances $i \in U$.

Input: Network structure, attributes $\mathbf{x}_1, \ldots, \mathbf{x}_n$, observed labels $\mathbf{y}_L^*$, and the marginal model f_i.
Step 1: Compute graph Laplacian Δ and the marginals ϕ_i in the dependence model.
Step 2: Apply Algorithm 1 to find the pseudo marginals $q_i(t_i)$ for $i = 1, \ldots, n$.
for each unlabeled instance $i \in U$ **do**
  Numerically evaluate the posterior mean $\hat{y}_i$ by sampling t_i from q_i and computing z_i by marginal CDF transform.
end for

5. EXPERIMENTS

In this section, we evaluate our proposed CLMN model on several real world network datasets from different domains, and demonstrate its effectiveness by comparing it with state-of-art relational classifiers.

5.1 Comparison methods

We consider representative approaches from the following major related methods:

Marginal only classifiers (LR). The baseline logistic regression classifier with $\ell 2$ regularization, which is the same as the marginal model for CLMNs in our experiments, is included to show the effectiveness of using attribute information alone.

Simple iterative collective classifiers (GMN). We include the widely-used, simple collective classifier—iterative weighted voting [13] where we apply the equivalent Gaussian Markov network [37] formulation to directly compute the MAP $\hat{\mathbf{y}}_U = \text{argmax}_{\mathbf{y}_U} p(\mathbf{y}_U|\mathbf{y}_L^*)$, which gives the relaxed label predictions of iterative weighted voting.

Personalized Page Rank collective classifiers (PPR). As mentioned in Section 3, another class of random-walk based classifiers, Personalized Page Rank, have also been widely applied to network data [36, 30, 10].

Latent Gaussian models (LGMN). In contrast to the relaxed labeling approach taken in GMNs, here we use a natural probabilistic model over the binary label space, while using the GMN to model the dependence structure over latent variables z_i. Here z_i determines the labels y_i through a noisy link function. This LGMN model falls into the general category of latent Gaussian models. In our experiments, we use the probit link function and apply EP for approximate inference. We note that we also experimented with the logistic link function for the LGMN model but observed that it performed worse than the probit link function in this setting, possibly due to the fact that the logistic-Gaussian factor cannot be analytically integrated and approximation degrades performance.

Relational feature based classifiers (SocDim). Instead of modeling the joint dependency of the labels, methods in this category extracts network-based features and combine them with other features to be applied in a conventional classifier. We include the recently proposed latent social dimension [28] approach from this category for comparison. The SocDim method uses top eigenvectors of the modularity matrix [23] of the graph as relational features. In our experiments, the top 200 eigenvectors are extracted, and we apply these features with a logistic regression classifier.

Relational Markov Networks (RMN). These methods attempt to model the statistical dependency among the labels, just like the CLMN model does. We include the widely applied relational Markov network [29] for comparison. We learn RMN parameters using a recently developed component likelihood based approach [33], which has been shown to enjoy desirable theoretical properties. We apply the model for collective classification using Gibbs sampling of 100,000 iterations.

Proposed CLMN model. In the implementation of the CLMN model, we specify ϵ in the dependence model to be 1, and the step size δ in the Metropolis Hasting algorithm to be 0.7. We run EP for 40 iterations during inference. Furthermore, to ensure proper convergence of the EP algorithm, we employ the common strategy of partial updates on the messages based on a dampening factor. More specifically, at each iteration k, we apply a step size $\alpha = e^{-0.125k}$, and the messages are updated as $q_i = (q_i^{\text{new}})^{\alpha}(q_i^{\text{old}})^{1-\alpha}$. Correspondingly, the practical updates applied in Step 2d of Algorithm 1 are: $\lambda_i := \alpha\lambda_i^{\text{new}} + (1-\alpha)\lambda_i$, $h_i := \alpha h_i^{\text{new}} + (1-\alpha)h_i$.

In all experiments, we sample part of data network to be the labeled set (i.e., $L \subset V_G$), and then learn the parameters of the LR, SocDim, RMN and CLMN models based on the labeled instances in L (no parameter estimation is involved in the other models). We then test all the models for prediction on the unlabeled instances (i.e., $U = V_G - L$). All results are averaged over 5 trials with different labelings.

5.2 Datasets

We apply the methods to four real-world network datasets drawn from E-Commerce, social network, social media, relational, and biological domains.

Facebook.

This dataset includes user profile attributes as well as friendship links among users. Our sample network consists of 6,694 users, which comprise a set of students from a university's Facebook network. The relational graph is constructed using the friendship links. We consider the classification tasks based on political views ("Conservative" or not). We use the users' gender, relationship status and religious view from their public profile as attributes.

IMDB.

This dataset is drawn from the Internet Movie Database (www.imdb.com). We used a sample of 1,068 movies released in the U.S. between 1996 and 2001. The binary classification task is to predict movie opening weekend returns (> \$2 million or not). We construct the relational graph by linking two movies together if they share the same producer. Two movie genre attributes (comedy, action) are used.

Amazon.

We use the subset of the Amazon co-purchasing network [9] which comprise of DVDs. After filtering DVDs with very few ratings, we obtain a network of 13,522 nodes. The links are extracted based on the "Customers Who Bought This Item Also Bought" field in the raw data. We include the average rating and 24 top level genres as the attributes on

each node, and predict whether the DVD is a best seller (sales rank threshold: 20,000).

Gene.
This dataset is collected from KDD Cup 2001 (www.cs.wisc.edu/dpage/kddcup2001/). It is a relational dataset containing information about the yeast genome at the the protein level and consists of 1,243 nodes. We predict protein localization ("nucleus" or not) from the interaction structure and 41 binary attributes associated with genes/proteins. These attributes belong to four different types: Essential (4), Phenotype (11), Chromosome (16) and Function (13). We use this dataset mainly to demonstrate the superior ability of the CLMN model to combine attributes and network structures. While most proprietary data from social or web domains naturally contain rich attribute information about the nodes and they are usually incorporated when developing any real application for such domains, the publicly-available datasets that contain both heterogeneous network topologies and informative attributes are sparse. Therefore, we include the gene dataset to demonstrate the broader applicability of the CLMN model.

5.3 Experimental results

In the first set of experiments, we test the performance of these methods on each of the above datasets, while varying the percentage of labeled instances in the network, and sampling the labeled nodes randomly. The experimental results are shown in Figures 2(a)-2(d). Our proposed CLMN model consistently outperforms the other models across all tasks, in some cases by a large margin, demonstrating its superior ability for relational classification. The SocDim approach is a close runner-up in many cases, demonstrating the usefulness of sophisticated relational feature construction methods that effectively flattens network structural information. The GMN model, albeit simple, achieves reasonable accuracy when there is enough correlation in the data, confirming the findings by previous work [14]. The performance of RMNs is rather unstable across different settings, especially when the labeled proportion is small, possibly due to the large learning variance associated with RMNs. Unlike the other models that use both network and attribute information (CLMNs and GMNs), RMNs sometimes perform even worse than the baseline independent classifier LR.

The effect of clustered labeled nodes.
In some applications, labeling is performed by exploring local clusters of the underlying single network. This is not consistent with our previous experiment where instances are labeled randomly throughout the network. To understand the effects of clustered labeling on model performance, we consider the following labeling procedure: given a *dispersion parameter* p, at each step, with probability p we pick an unlabeled node at random to generate a new cluster; with probability $1-p$ the current cluster is expanded greedily by adding an unlabeled node which maximizes the linkage to nodes that are already in the current cluster. Clearly, when $p = 1$, this procedure is equivalent to random sampling as in the previous experiment; when $p = 0$, if the underlying network consists of only one connected component, this procedure would result in one highly connected cluster. The experimental results on the Facebook data with labeled proportion of 0.4 is plotted in Figure 2(e). Other labeled proportions exhibit similar trends. Except for SocDim, all relational classifiers tend to perform relatively poorly when the labeled nodes are highly clustered ($p \leq 0.2$). While the performance of CLMN, LGMN and GMN consistently improves as the labeled nodes are further spread out (p increases), the RMN model demonstrates a different trend—it performs rather poorly when the labeled nodes are highly isolated (e.g., $p = 1$). This is because the sampled graph needs to match the structure of the underlying network in order for RMNs to learn the collective classification model well. In particular, marginal distributions in RMNs estimated from the labeled subnetwork differ more significantly from that of the underlying whole network when p is large.

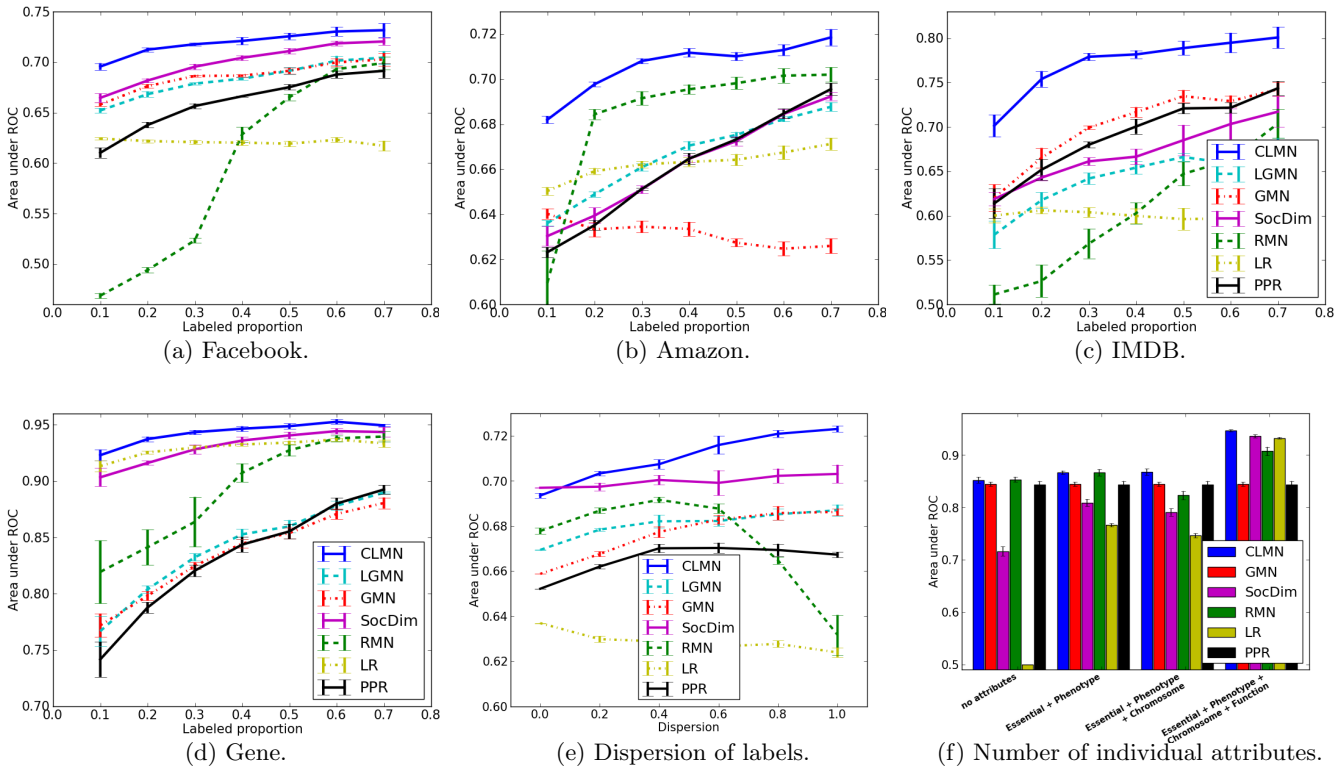
The effect of attributes.
As stated before, the primary goal of our CLMN model is to better combine individual and relational information in the data. Thus, we evaluate the ability of CLMNs to utilize attribute information more carefully in comparison with SocDim and RMNs, by varying the number of attributes used in the model. The results are plotted in Figure 2(f). We include the LR method for comparison as well since it directly shows the usefulness of the different groups of attributes. The GMN model, for which the performance remains unchanged, is also included to show the proportion of the gains due to the relational information. We found that when no attributes or only a few informative attributes are used, the performance of CLMNs and RMNs is essentially the same, while the SocDim approach cannot exploit the relational information adequately. As the amount of informative features increases, the performance of SocDim catches up, while the RMN approach tends to suffer. Overall, this demonstrates the superior ability of CLMNs to utilize attributes in relational settings through the assumption of identical marginal classifiers.

6. RELATED WORK

A large body of research work has focused on latent variable models for relational data [3, 1, 17]. However, unlike CLMNs, which are conditioned on network structures, these are generative models and attempt to model network structures. They are typically applied to link prediction or collaborative filtering tasks. The latent variables in these approaches typically represent the latent block structure of the network data, instead of the latent tendency of activating a certain label as in the case of CLMNs.

Latent Gaussian models such as latent GMNs [38] and relational Gaussian processes [2] have been applied to graph-based node classification. However, they are usually applied in IID scenarios for semi-supervised prediction, where the attributes are used to construct the graph. In our relational tasks, there are natural links among the data instances (e.g., articulated freindships). While it is possible to further incorporate attributes in latent GMNs, the associated hyperparameter learning is usually difficult. Our copula approach provides a solution to the problem of integrating network and attribute information in latent Gaussian models.

An alternative, widely applied approach to combining network structures and attributes in a unified probabilistic model is based on discrete Markov networks. Variants of this approach include relational Markov networks [29], Markov logic networks [25] and relational dependency networks [22]. These models are based on clique templating, i.e, the same form of



(a) Facebook. (b) Amazon. (c) IMDB.

(d) Gene. (e) Dispersion of labels. (f) Number of individual attributes.

Figure 2: Classification performance on real networks. (a) to (d): AUCs as the proportion of labeled data is varied. (e): Classification results on Facebook data with 40% of the nodes labeled. The dispersion of the labeled nodes is varied from 0 (contiguous subgraph) to 1 (nodes selected at random throughout the network). (f): Classification results on gene data when sets of individual node attributes, of varying size, are used.

clique potentials (e.g., edge potentials) and parameters are tied across the whole network. In contrast to CLMN's "homogeneous marginal" assumption, the probabilistic meaning of the "homogeneous potential" assumption is unclear.

Copulas were developed in statistics and have been widely applied to finance and economics [4, 21]. In machine learning, copula approaches have received much attention lately [6, 11, 32, 24], but they have been used in rather different ways than our proposed model—i.e., the applications revolve around density estimation, clustering and other descriptive modeling tasks rather than prediction. As such, to our knowledge, conditional inference methods have not previously been developed for copula models.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have developed a new discriminative probabilistic model for relational classification in network data. The proposed model is based on: (1) the intuitive assumption that instances in a network are interdependent but nevertheless should be "identically distributed", and (2) a reasoning about the correlations among latent effects that cannot be explicitly explored using independent probabilistic classifiers. We are thus able to extend a generic conventional probabilistic classifiers into a relational classifier. Based on the copula construction, our proposed CLMN model is able to combine the benefits of modeling relational dependencies, with the ability to incorporate attributes, in a general probabilistic classification model. To apply the model for collective classification in large networks, we develop an approximate inference algorithm based on message passing, which has been demonstrated to achieve high classification accuracy on a variety of real network datasets. Note that since both matrix inversion (Step 1 of Algorithm 2) and message passing (Step 2 of Algorithm 2) are immediately parallelizable, a parallel implementation of our collective classification algorithm is straightforward. Therefore, due to its scalability and its robust performance across different settings, the proposed CLMN approach provides an ideal solution for collective classification in online systems such as social network and social media.

There are several future directions that we would like to explore with this work. First, we plan to investigate hyperparameter learning for the dependence model so that we can learn the graph Laplacian instead of fixing it when there are link attributes available. While an obvious starting point in this direction is an EM-EP algorithm, more sophisticated methods (e.g. [19]) may be explored to improve the learning quality. Moreover, *active label acquisition* for the CLMN model is worth studying in the scenarios when labeling costs are high. We may investigate this issue by using the covariance structure of the Gaussian Markov network [7].

Acknowledgement

This research is supported by NSF under grant numbers CCF-0939370 and IIS-1149789. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of the NSF or the U.S. Government.

8. REFERENCES

[1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, June 2008.

[2] W. Chu, V. Sindhwani, Z. Ghahramani, and S. S. Keerthi. Relational learning with gaussian processes. In *NIPS*, pages 289–296, 2006.

[3] P. D. Hoff, A. E. Raftery, M. S. Handcock, and M. S. H. Latent space approaches to social network analysis. *J. Amer. Stat. Assoc.*, 97:1090–1098, 2001.

[4] H. Joe. *Multivariate Models and Dependence Concepts*. Chapman and Hall, London, 1997.

[5] G. Kim, M. Silvapulle, and P. Silvapulle. Comparison of semiparametric and parametric methods for estimating copulas. *Computational Statistics and Data Analysis*, 51:2836■2850, 2007.

[6] S. Kirshner. Learning with tree-averaged densities and distributions. In *NIPS*, Vancouver, Canada, 2007.

[7] A. Krause and C. Guestrin. Nonmyopic active learning of gaussian processes: An exploration-exploitation approach. In *ICML*, pages 449–456, 2007.

[8] M. Kuss, C. E. Rasmussen, and R. Herbrich. Assessing approximate inference for binary gaussian process classification. *JMLR*, 6:1679–1704, 2005.

[9] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1), May 2007.

[10] F. Lin and W. W. Cohen. Semi-supervised classification of network data using very few labels. In *ASONAM '10*, 2010.

[11] H. Liu, J. Lafferty, and L. Wasserman. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *J. Mach. Learn. Res.*, 10:2295–2328, Oct. 2009.

[12] Q. Lu and L. Getoor. Link-based classification. In *ICML*, 2003.

[13] S. Macskassy and F. Provost. A simple relational classifier. In *Proceedings of the 2nd Workshop on Multi-Relational Data Mining, KDD2003*, 2003.

[14] S. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *JMLR*, 8(May):935–983, 2007.

[15] D. Malioutov, J. Johnson, M. J. Choi, and A. Willsky. Low-rank variance approximation in gmrf models: Single and multiscale approaches. *Signal Processing, IEEE Transactions on*, 56(10):4621 –4634, oct. 2008.

[16] D. M. Malioutov, J. K. Johnson, and A. S. Willsky. Walk-sums and belief propagation in gaussian graphical models. *J. Mach. Learn. Res.*, 7:2031–2064, December 2006.

[17] K. T. Miller, T. L. Griffiths, and M. I. Jordan. Nonparametric latent feature models for link prediction. In *NIPS*, 2009.

[18] T. P. Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, UAI '01, pages 362–369, 2001.

[19] I. Murray and R. P. Adams. Slice sampling covariance hyperparameters of latent Gaussian models. In *NIPS*, pages 1723–1731, 2010.

[20] R. M. Neal. Regression and classification using gaussian process priors. *Statistics*, 6:475–501, 1998.

[21] R. B. Nelsen. *An Introduction to Copulas*. Springer, 2006.

[22] J. Neville and D. Jensen. Relational dependency networks. *J. Mach. Learn. Res.*, 8:653–692, 2007.

[23] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3), July 2006.

[24] B. Poczos, Z. Ghahramani, and J. Schneider. Copula-based kernel dependency measures. In *ICML*, 2012.

[25] M. Richardson and P. Domingos. Markov logic networks. *Mach. Learn.*, 62(1-2):107–136, 2006.

[26] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. Collective classification in network data. *Ai Magazine*, 29(3), 2008.

[27] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky. Nonparametric belief propagation. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[28] L. Tang and H. Liu. Relational learning via latent social dimensions. In *KDD*, pages 817–826, 2009.

[29] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *UAI*, 2002.

[30] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. In *ICDM*, pages 613–622, 2006.

[31] Y. Weiss and W. T. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation*, 13:2173–2200, 2001.

[32] A. G. Wilson and Z. Ghahramani. Copula Processes. In *NIPS*, page 11, 2010.

[33] R. Xiang and J. Neville. Relational learning with one network: An asymptotic analysis. In *AISTATS*, 2011.

[34] R. Xiang and J. Neville. Understanding propagation error and its effect on collective classification. In *ICDM*, 2011.

[35] B. Yu. Rates of convergence for empirical processes of stationary mixing sequences. *Ann. Probab.*, 22(1), 1994.

[36] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. SchŽlkopf. Learning with local and global consistency. In *NIPS*, pages 321–328, 2004.

[37] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML '03*, 2003.

[38] X. Zhu, J. Lafferty, and Z. Ghahramani. Semi-supervised learning: From gaussian fields to gaussian processes. Technical report, School of CS, CMU, 2003.