UCRL-TR-225637

LAWRENCE LIVERMORE NATIONAL LABORATORY

A Survey of Probabilistic Models for Relational Data

P. S. Koutsourelakis

October 26, 2006

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

A Survey of Probabilistic Models for Relational Data

Predictive Knowledge Systems Initiative

P.S. Koutsourelakis

1 Introduction

Traditional data mining methodologies have focused on "flat" data i.e. a collection of identically structured entities, assumed to be independent and identically distributed. However, many real-world datasets are innately relational in that they consist of multi-modal entities and multi-relational links (where each entity- or link-type is characterized by a different set of attributes). Link structure is an important characteristic of a dataset and should not be ignored in modelling efforts, especially when statistical dependencies exist between related entities. These dependencies can in fact significantly improve the accuracy of inference and prediction results, if the relational structure is appropriately leveraged (Figure 1).

The need for models that can incorporate relational structure has been accentuated by new technological developments which allow us to easily track, store, and make accessible large amounts of data. Recently, there has been a surge of interest in statistical models for dealing with richly interconnected, heterogeneous data, fuelled largely by information mining of web/hypertext data, social networks, bibliographic citation data, epidemiological data and communication networks.

Graphical models have a natural formalism for representing complex relational data and for predicting the underlying evolving system in a dynamic framework.

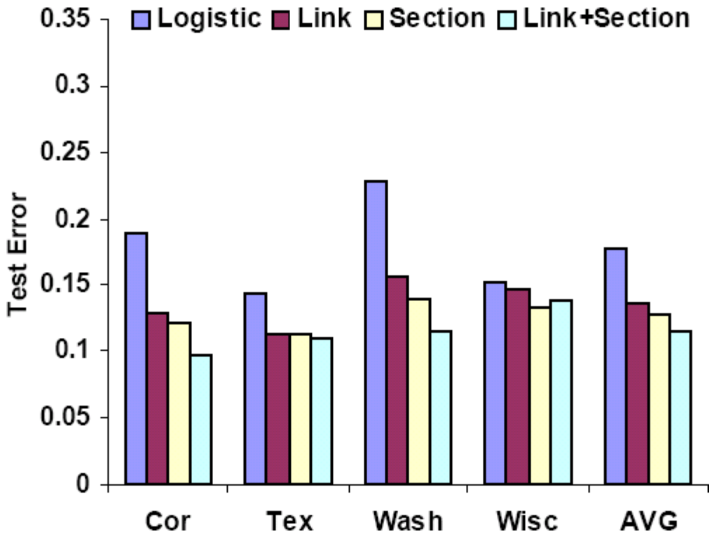


Figure 1: Comparison of flat (i.e. not taking into account relational structure) versus collective classification on WebKB database: *Logistic* is a flat logistic regression model and *Link, Section* and *Link+Section* are three different relational models. Figure taken from [38]

The present survey provides an overview of probabilistic methods and techniques that have been developed over the last few years for dealing with relational data. Particular emphasis is paid to approaches pertinent to the research areas of pattern recognition, group discovery, entity/node classification, and anomaly detection. We start with supervised learning tasks, where two basic modelling approaches are discussed – i.e. discriminative and generative. Several discriminative techniques are reviewed and performance results are presented. Generative methods are discussed in a separate survey. A special section is devoted to latent variable models due to their unique characteristics and usefulness in static and dynamic frameworks and in both supervised and unsupervised learning processes.

Section 4 contains a brief discussion of unsupervised learning techniques with an emphasis on computational efficiency and large networks. Finally, section 5 discusses performance metrics with an emphasis on classification problems.

2 Supervised Learning of Graphical Models

In broad terms, the methods of supervised learning in graphical models can be partitioned into generative and discriminative classes. Provided with sufficient training data, the discriminative approach is expected to yield superior accuracy as compared to its generative counterpart since no modelling power is expended on the marginal distribution of input features. This is especially true in classification and regression in relational structures, which do not exhaust potential inference problems. Conversely, if the probabilistic model of the relational data is accurate, the generative approach can perform better with less data. In general it is less prone to overfitting and allows one to more easily specify meaningful priors on the model parameters.

In the following, we review several discriminative models for addressing problems of probabilistic inference in general graph structures and communication networks. To illustrate the fundamental difference between these two types of models, we follow the approach of Minka [22]. Let $\boldsymbol{y}$ denote the attributes of the entities that we wish to predict and $\boldsymbol{x}$ represent the observed input variables. In a generative setting, one defines a joint model $p_g(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$ which depends on a set of parameters $\boldsymbol{\theta}$. These parameters are selected so that the model p_g provides a good representation of the data. The aforementioned model could also be written as:

$$p_g(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) = p_g(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\theta}) p_g(\boldsymbol{x}; \boldsymbol{\theta}) \tag{1}$$

Hence the maximum likelihood estimation of $\boldsymbol{\theta}$ given data $\{\boldsymbol{x_i}, \boldsymbol{y_i}\}$ would require optimizing:

$$L_g(\boldsymbol{\theta}) = \sum_i (\log p_g(\boldsymbol{y_i} \mid \boldsymbol{x_i}; \boldsymbol{\theta}) + \log p_g(\boldsymbol{x_i}; \boldsymbol{\theta})) \tag{2}$$

On the other hand, in a discriminative setting one needs only to define a conditional model $p_d(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\theta})$ where the parameters θ are now used to define the conditional distribution and are independent of $\boldsymbol{x}$. This can be combined with an arbitrary prior of $\boldsymbol{x}$, i.e. $p_d(\boldsymbol{x}; \boldsymbol{\theta'})$, which depends on a new set of parameters $\boldsymbol{\theta'}$ that are not necessarily the same as $\boldsymbol{\theta}$. Hence the joint pdf of the discriminative model can be written as:

$$p_d(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}, \boldsymbol{\theta'}) = p_d(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\theta}) p_d(\boldsymbol{x}; \boldsymbol{\theta'}) \tag{3}$$

and the log-likelihood for $(\boldsymbol{\theta}, \boldsymbol{\theta'})$:

$$L_d(\boldsymbol{\theta}, \boldsymbol{\theta'}) = \sum_i (\log p_d(\boldsymbol{y_i} \mid \boldsymbol{x_i}; \boldsymbol{\theta}) + \log p_d(\boldsymbol{x_i}; \boldsymbol{\theta'})) \tag{4}$$

As it can be readily seen by comparing Equations (2) and (4), the latter exhibits more flexibility in interpreting the data because it does not require that $\boldsymbol{\theta} = \boldsymbol{\theta'}$. In particular, in cases where we are interested only in predictions for $\boldsymbol{y}$, that is only in the conditional $p_d(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\theta})$, the second term in Equation (4) becomes irrelevant. If however a generative model was used for the latter problem, inadvertently the accuracy in capturing $p_g(\boldsymbol{y} \mid \boldsymbol{x})$ would be (at least) partially compromised. This happens because the parameters $\boldsymbol{\theta}$ are determined so they provide a good interpretation of $p_g(\boldsymbol{x})$ in addition to $p_g(\boldsymbol{y} \mid \boldsymbol{x})$ (Equation (2)).

2.1 Discriminative Models

In this section, we review four basic approaches to discriminative modelling of relational data — namely Conditional Random Fields, Relational Markov Networks, Markov Logic Networks, and Structural Logistic Regression. Even though these techniques emerged at roughly the same time, their motivations and applications are quite different.

If $\boldsymbol{x}$ denotes the input variables that are observed and $\boldsymbol{y}$ the output variables that we wish to infer (in a classification or regression setting), then discriminative models are represented by an undirected graph (not necessarily acyclic). Consider for example a citation network consisting of papers that we wish to label based on their general topics (e.g. in a mathematical database those topics can be number theory, topology, analysis, logic, etc). These labels represent the output variables $\boldsymbol{y}$ and the attributes of each paper (e.g. title words, authors' names, keywords) the input variables $\boldsymbol{x}$ (Figure

2). In this setting, an associated Markov network factors the conditional distribution $p(\boldsymbol{y} \mid \boldsymbol{x})$ as follows:

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{1}{Z} \prod_{A} \Psi_A(\boldsymbol{y_A}, \boldsymbol{x_A}) \tag{5}$$

where the factors Ψ_A (clique potentials) are non-negative functions of the nodes in each clique, Z is the normalization constant, A is an index over all cliques and $\boldsymbol{y_A}, \boldsymbol{x_A}$ the variables associated with clique A (Figure 2). An alternative graphical representation of this structure is provided by a factor graph, a bipartite graph in which a variable node v (belonging to $\boldsymbol{x}$ or $\boldsymbol{y}$) is connected to a factor node Ψ_A if v is an argument in Ψ_A. Most often, particularly for computational implementation, it is assumed that each factor is parameterized by an exponential form. Thus Equation (5) can be written as:

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{1}{Z} \prod_{\Psi_A} \exp \left\{ \sum_{k=1}^{K(A)} \lambda_{Ak} f_{Ak}(\boldsymbol{y_A}, \boldsymbol{x_A})) \right\} \tag{6}$$

The feature functions or sufficient statistics f can be binary (as in text modelling applications) or real-valued (as in computer vision models). Roughly speaking they specify the cliques and potentials between attributes of related entities. Consider for example a dataset in which the entities are web-pages and the relations are hyperlinks from one web-page to another. If $\boldsymbol{y}$ represents the labels of those web-pages and we assume that entities with the same labels tend to be linked, then we can capture this by introducing for each link a clique between the labels of the source and its target pages. The potential of the clique will then have higher values for identical label assignments to the linked pages. Similar formulations can be adopted for other relational schemas.

In contrast to generative models, the feature functions can introduce long-range dependencies and cycles in graphical representation. This allows for added modelling flexibility in capturing complex relational structures. Logistic regression, a well-studied statistical model for classification, can be viewed as the simplest example of a discrim-

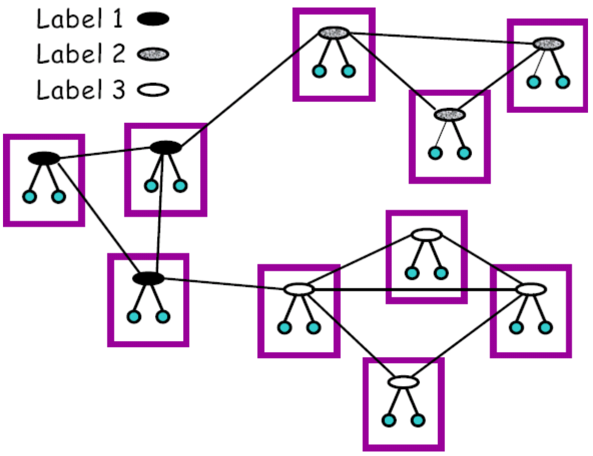


Figure 2: An unrolled Markov net over linked documents. The ellipses indicate labels/topics (output variables $\boldsymbol{y}$) and the circles the attributes of each document (input variables $\boldsymbol{x}$). For each link, a clique is introduced between the labels of the source and the target page. Note that documents with the same label tend to be linked to each other. This can be captured by having higher values of the potential on each clique for assignments that give common labels to the linked pages. Figure taken from [38].

inative Markov model.

For several practical applications (which are characterized by statistical homogeneity along the graph), the weights λ can be tied, i.e. we can partition the factors of the graph into a number of clique templates C_i whose parameters are the same. In this case, the conditional distribution can be rewritten as:

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{1}{Z} \prod_{C_i} \prod_{\Psi_c \in C_i} \exp \left\{ \sum_{k=1}^{K(i)} \lambda_{ik} f_{ik}(\boldsymbol{y_c}, \boldsymbol{x_c})) \right\} \tag{7}$$

Conditional Random Fields (CRFs) represent one of the first attempts to introduce discriminate models in relational settings [18]. The motivation was to address problems related to segmenting and collectively labelling sequence data (e.g. text) with higher accuracy compared to existing alternatives such as the generative Hidden Markov Models (HMMs) and the discriminative maximum entropy Markov models (MEMMs). The original framework was further developed and generalized in a series of papers that ensued [20, 44, 36, 5, 6, 34, 25, 35] with applications in various types of relational structures such as entity recognition and classification in text, RNA structural alignment and protein structure prediction, labelling and segmentation of images, object recognition in computer vision. In these papers, various versions of CRFs have appeared with different levels of complexity and clique sizes. Linear chain CRFs are perhaps the simplest version of CRFs for sequence modelling and can be considered the discriminative counterpart of HMMs. If the state space is not particularly large, inference can be facilitated by employing variants of the dynamic-programming algorithms for HMMs. Learning of the weights λ is based on finding the mode of the posterior when Gaussian priors are used. The optimization component is usually carried out using gradient ascent, conjugate gradients, the Broyden-Fletcher-Goldfarb-Shanno optimization algorithm (BFGS) or a limited memory BFGS (referred to as L-BFGS). For general CRFs, the choice for an inference method depends on the amount of training data available for $\boldsymbol{y}$. In problems with incomplete training data maximization of the posterior is performed using gradient ascent or Expectation-Maximization (EM) [35]. For inference with complete training data, approximate methods such as pseudo-likelihood, variational approaches, or loopy belief propagation have been recommended. An additional advantage of CRFs is that the descriptive ability of the possible feature functions can be quantitatively assessed. In [20], a greedy optimization algorithm is presented that performs automatic feature induction, i.e. it selects those feature functions f that significantly increase the conditional likelihood if added to the model. This allows for more compact descriptions and near-optimal use of computational resources in learning the model.

Relational Markov Networks (RMNs), which first appeared in [38], are a type of general CRF in which the graphical structure and parameter tying are determined by an SQL-like syntax. They share the same underlying principles and modelling assumptions with CRFs and relevant discriminative models. As such, the graphical structure of RMNs is based on the relational structure of the domain and can easily

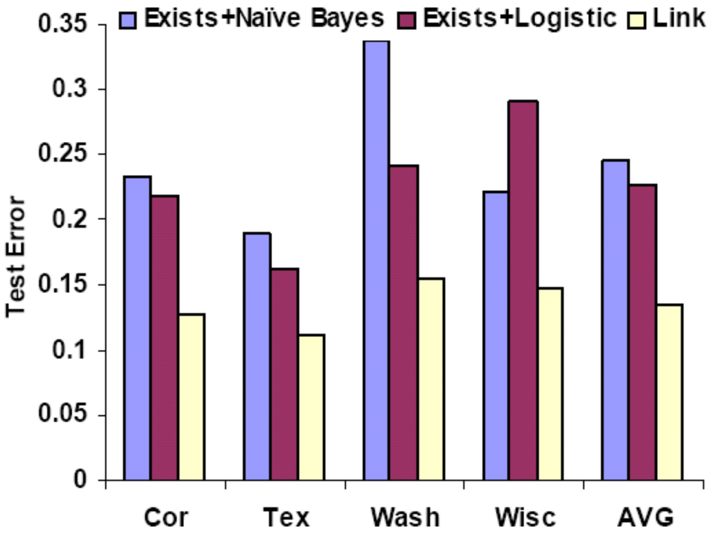


Figure 3: Comparison of discriminative and generative models for WebKB database. *Exists+Naive Bayes* is completely generative. *Exists+Logistic* is generative in the links but locally discriminative in the page labels given to the local features (words, meta-words). *Link* is completely discriminative. Figure taken from [38].

model complex patterns over related entities. Original applications involved collective classification of linked web-pages with approximately 1400 nodes. RMNs achieved a labelling error of about 10% in contrast to 20% by a simple logistic regression scheme and by generative models (Figure 3). The problem of link prediction over 5 possible link types, in the same database has also been considered where RMNs were found to perform better than existing techniques. Similar success was also observed in link prediction for social networks [41, 39].

Maximum margin Markov networks (MMMNs) represent a combination of RMNs with Support Vector Machines [40, 37]. As a result they carry desirable features from both formulations such as the use of kernels (which can efficiently deal with high-dimensional feature spaces) and the ability to capture correlations in structured data. An efficient algorithm has been proposed for learning MMMNs based on a compact quadratic program formulation. Experiments in several problems such as handwritten character recognition and collective hypertext classification demonstrate very signifi-

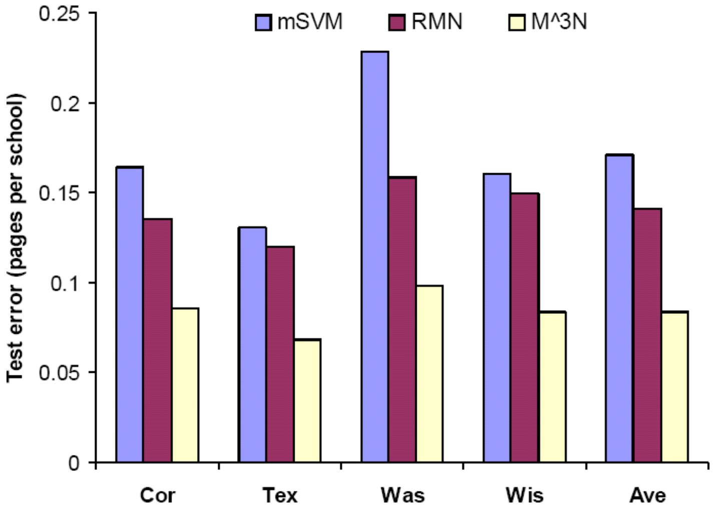
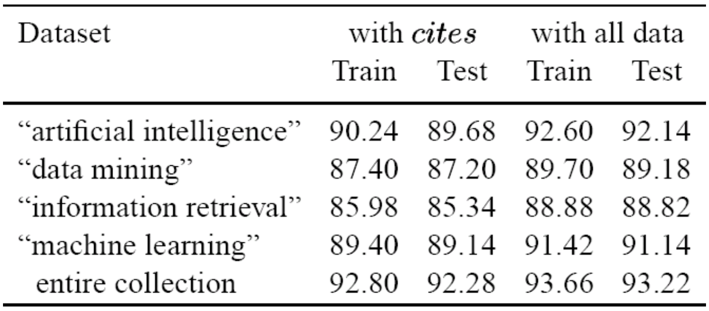


Figure 4: Comparison of various methods on WebKB database. *mSVM* corresponds to multi-class SVM, *RMN* to Relational Markov Networks and *M3N* to MMMN. Figure taken from [40].

cant performance gains, in the order of 30% to 40% in relative accuracy, over alternative approaches such as logistic regression, standard RMNs and SVMs (Figure 4).

Markov logic networks [28, 31] are discriminative models where the feature functions in Equation (6) take on the form of first-order logic clauses. They have been successfully applied to collective classification problems and comparative results have also been produced for assessing different techniques of approximate inference. In [16], a novel procedure for the selection of feature functions was presented that combines ideas from inductive logic programming (ILP) and feature induction in Markov networks. The algorithm performs a beam or shortest-first search over the space of clauses, guided by a weighted pseudo-likelihood measure.

Finally, Structural Logistic Regression (SLR) is a discriminative model that essentially extends logistic regression in relational settings [26, 27]. In comparison to the aforementioned formulations, it is perhaps the most similar to RMNs in the sense that the feature functions are constructed from SQL queries over the input data. The model uses the Bayesian Information Criterion (BIC) in order to sequentially augment the number of feature functions used. It has been successfully applied to the problem of



Dataset	with *cites*		with all data	
	Train	Test	Train	Test
“artificial intelligence”	90.24	89.68	92.60	92.14
“data mining”	87.40	87.20	89.70	89.18
“information retrieval”	85.98	85.34	88.88	88.82
“machine learning”	89.40	89.14	91.42	91.14
entire collection	92.80	92.28	93.66	93.22

Figure 5: Training and test accuracy (%) of the models learned using only one relation type (*cites*) and all relation types (*cites, author, published_in*). Performance reported for four types of articles in the database (i.e. “artificial intelligence,” “data mining,” “information retrieval,” and “machine learning”) and for the entire collection. Figure taken from [26].

unobserved link prediction in the Citeseer citation database (Figure 5).

2.2 Generative Models

A detailed discussion is contained in ”Survey of Bayesian Models for Modelling of Stochastic Temporal Processes” by Brenda Ng.

3 Latent Variable Models

In this section, we will consider probabilistic models that are defined in terms of some latent or hidden variables. Even though they could have been discussed in the previous sections along with discriminative and generative models, we devote a special section due to their unique characteristics and in order to emphasize their usefulness in various tasks relevant to the PKS project. Latent variables are hidden variables that relate nodes in a graph by grouping. A variety of such models have appeared in the literature in static and dynamic frameworks and in a supervised or an unsupervised learning processes. These models can be used to perform tasks such as link prediction, discovery of groups/clusters with similar characteristics, etc.

Consider a transaction network for which we have some link data $Y_{i,j}$. For simplicity, we assume that $Y_{i,j}$ is symmetric ($Y_{i,j} = Y_{j,i}$) and binary so that there is no link between i and j if $Y_{i,j} = 0$ and a link exists if $Y_{i,j} = 1$. This formulation can be readily extended to cases where $\boldsymbol{Y}$ takes on categorical or even real values to account for the type or volume of transactions between nodes i and j and to problems where the matrix $\boldsymbol{Y}$ is non-symmetric which indicates that the relational structure has a directional character. Consider also a number of covariates pertinent to these nodes, i.e. $\boldsymbol{X_{i,j}} \in \mathbb{R}^k$, which can include attributes of each of the nodes or of the links between them. The goal is to construct a model for predicting $Y_{i,j}$ given the covariate data. The basic assumption is that each node is associated with a latent variable Z_i which completely determines its link properties. Hence the $Y_{i,j}$ are conditionally independent given $\boldsymbol{Z}$ and therefore:

$$P(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\theta}) = \prod_{i,j} P(Y_{i,j} \mid \boldsymbol{X_{i,j}}, Z_i, Z_j, \theta) \tag{8}$$

where θ is a set of parameters to be optimized during parameter learning.

Nowicki and Snijders [24] introduced the first such approach in social networks, where Z_i indicates a membership to an unobserved class, cluster, or group and the probability of a link between two nodes depends only on the respective groups to which the nodes belong. The authors assumed a fixed number of clusters and the membership assignments were drawn from a multinomial distribution. Apart from its significance in link prediction, this model can be useful for group discovery based on relational data.

The same basic idea was recently explored in the terms of Infinite Relational Model (IRM) [15] for the purpose of unsupervised partitioning of various node types into clusters on the basis that a good set of partitions allows relationships between entities to be predicted by their cluster assignment. The authors formulated a framework in which such a task can be performed in the presence of multiple relationship types linking various entity types. Since the number of clusters is initially unknown, the authors adopt a Dirichlet Process prior i.e. a prior that allows for countably infinite clusters. In addition, they use a Beta prior $Beta(\beta, \beta)$ for representing the probability of relation between nodes that belong to any pair of classes. Inference and learning in nonparametric Bayesian models such as the generative IRM is an area that has attracted a lot of attention in recent years and MCMC methods generally provide the optimal solution [23, 14].

A distinctive feature of the IRM is its ability to automatically handle arbitrary collections of relations each of which might take on any number of arguments. Moreover, its Bayesian structure allows increasingly complex representations to be learned as more data become available. IRMs have been applied to several applications such as clustering synthetic data, clustering objects and features, learning ontologies and analyzing social networks. In these cases, the model exhibited high accuracy even when multiple types of nodes and relationships were present. Unfortunately, no information was provided regarding the associated computational effort.

It should also be noted that this formulation was adapted in [19] as a nonparametric prior over Bayes nets. The resulting model represents a nice compromise between learnability and expressivity of relational knowledge. As shown in several problems, the representation of data is superior to that in which a uniform prior is used, both in terms of the amount of data needed to learn the correct structure and in terms of the accuracy between the posterior distribution and the ground truth. Nevertheless, the test problems involve only 10 to 40 variables where inferences can be performed quickly. It is questionable how this modelling technique scales computationally to larger problems.

For a pre-defined number of groups, an extension of the aforementioned framework is the Group-Topic (GT) model presented in [43]. It is a generative model that incorporates information about attributes of the relations between various objects instead of just the existence of the relation itself. This is achieved by conditioning the group membership on a latent variable associated with attributes of the relation. Consider for example an email database in which messages indicate links between people in the network. In the GT model, group formation does not exclusively depend on the existence of the message itself but also on its attributes i.e. the words it contains. It is assumed that each message belongs to a topic (where the total number of topics is fixed) and group assignment depends on the topic. For example one grouping of the nodes might arise when the topic of emails is work-related and another when the topic is related to social-activities. In this way, group discovery is guided by emerging topics and topic discovery is guided by emerging groups. Both modalities are adjusted so that the likelihood of data is increased. Inference in this model is performed by Gibbs sampling which is facilitated by the use of conjugate priors that allow for efficient computation of the posterior distribution. The authors present applications on sixteen years of bills put before the US Senate (in this case, a link is defined if two senators gave the same vote for a bill) and 43 years of similar data from the United Nations Assembly. In both cases, the model is able to identify pertinent topics and groups of senators or nations that voted similarly for each topic.

The GT model is essentially an extension of the model discussed in [17] which incorporates attributes of an entity rather than attributes of relations between entities. In several ways, the GT model is identical to the RART (Role-Author-Recipient-Topic) model presented in [21] which in turn represents an extension of the ART model that appeared in the same paper. The fundamental difference is that ART does not explicitly capture the groups formed. In particular, the generative procedure adopted therein assumes that each word is generated by selecting a recipient x (from the pool of recipients of a message) and a topic is drawn from a multinomial that depends on the author and the recipient x. Words are drawn from a multinomial depending on the topic. The total number of topics and words in the vocabulary is assumed fixed. The ART model was successfully applied in the Enron dataset where it was able to uncover relevant pairs of author-recipient for each topic. Furthermore, the results obtained were combined with the Jensen-Shannon divergence in order to find similarities in the roles of people (in the network) based on the premise that nodes with similar distributions over their communication partners should be considered role-equivalent. Their results compare favorably with standard social-network block structure techniques.

It should be noted that latent cluster models in generative formulations have also been extended to dynamic settings and particularly to applications related to topic discovery and evolution in a corpus of documents [2, 42, 45]. These formulations however only incorporate attributes of the nodes (i.e. in the case of documents, the words contained in a document) and their co-occurrence frequencies. That is, no relational information is exploited in these models or their static counterparts [3, 1].

We return to the problem of link prediction and the formulation of Equation (8). As mentioned earlier latent variables force the links $Y_{i,j}$ to be conditionally independent and represent unobserved random effects in the network structure and behavior. In [13], Hoff et al developed a model that was inspired by social networks, where Z_i denotes the coordinates of each node in an unobserved, so-called "social space". The probability of a link between two nodes depends exclusively on their distance $d(Z_i, Z_j)$ in the social space. These models are able to represent standard network behavior such as clustering and transitivity, and their estimation is fairly straightforward, at least for fairly small networks. Most commonly, the social space is assumed to be $\mathbb{R}^2$ (higher dimensions are also possible but computational effort will increase accordingly) and the standard Euclidean norm is taken as a measure of the distance. In addition, the proposed method provides a visual and interpretable model-based spatial representation of the network structure and relations. Learning of parameters can be done in a maximum Likelihood setting (in fact the likelihood is concave in terms of the relative distances) or a general Bayesian framework. Applications in several social networks (with less than 100 nodes) have been successful in predicting missing links and uncovering social proximity. This model was revisited in [30, 11] and extended to a dynamic setting in [29]. Therein, a first-order Markov Gaussian model was adopted for representing the evolution of each node's coordinates in the social space and several approximations in the log-likelihood were used to alleviate the computational burden. This allowed successful application to networks with up to $11,000$ nodes and over 6 time steps. It was found by the authors that the complexity of algorithm is $O(n \log n)$ where n is the number of nodes.

In the most recent version of the aforementioned model, Hoff [12] assumes a matrix form of the latent variables $Z_{i,j}$ for each pair of nodes i and j and a decomposition of the form:

$$\boldsymbol{Z} = \boldsymbol{M} + \boldsymbol{E} \tag{9}$$

where $\boldsymbol{M}$ represents systematic patterns and $\boldsymbol{E}$ the noise. In order to reduce dimensionality of the unknown parameters, a reduced-rank decomposition of $\boldsymbol{M}$ is used instead:

$$\boldsymbol{M} = \boldsymbol{U}\ \boldsymbol{D}\ \boldsymbol{V} \tag{10}$$

where $\boldsymbol{U}$ and $\boldsymbol{V^T}$ are orthogonal $n \times K$ matrices (where $K << n$) and D a diagonal $K \times K$ matrix. In a Bayesian framework, appropriate priors on the matrices and remaining parameters are introduced in order to fit the model. Applications have been considered in a network with $n = 130$ nodes. The link structure examined was defined by whether country i initiated a conflict with country j. Several covariates such as populations, polity scores, geographic distance were considered. Despite its increased expressivity, the model appears to be computationally expensive especially for large networks, unless a good representation for the random effects matrix $\boldsymbol{Z}$ (Equation (9)) can be found in advance. Such procedures are discussed in the next section.

4 Efficient Unsupervised Learning

Several data mining applications on large graphs and communication networks have recently appeared in the literature with particular emphasis on fast and space efficient computational procedures that are able to deal with hundreds of thousands of nodes in a dynamic environment. We will discuss in more detail matrix decomposition techniques for graph structure and anomaly detection.

Consider a large graph represented as a sparse adjacency matrix $\boldsymbol{A}$ with binary (indicating the presence of absence of a link) or real-valued entries (indicating the volume of an exchange/transaction). The typical way of summarizing and approximating such matrices is through transformations such as SVD (Singular Value Decomposition) or PCA (Principal Component Analysis), which are not space efficient and do not take advantage of the sparsity of $\boldsymbol{A}$. For that purpose, Drineas et al. [7] developed the CUR decomposition that adopts a representation of the form:

$$\boldsymbol{A} \approx \boldsymbol{CUR} \tag{11}$$

where $\boldsymbol{C} \in \mathbb{R}^{m \times c}, \boldsymbol{U} \in \mathbb{R}^{c \times r}$ and $\boldsymbol{R} \in \mathbb{R}^{r \times n}$ $(c, r << m, n)$. An improved version of CUR is the Compact Matrix Decomposition (CMD) presented in [33] which adopts a similar representation as in Equation (11) but requires much less space and computation time. The columns of the matrix $\boldsymbol{C}$ are constructed by sampling the columns of $\boldsymbol{A}$ with weights proportional to their Euclidean norms. The central matrix $\boldsymbol{U}$ is dense but of fairly small dimension, at least compared to the original system. It is shown that this low-rank approximation can capture a significant portion of the activity and identify salient communication patterns associated with rows and columns of $\boldsymbol{C}$ and $\boldsymbol{R}$ matrices. The Frobenius norm can be used to quantify the approximation error, which can be rapidly calculated by partially sampling the entries. This error measure can be readily used for anomaly detection by identifying those columns (or rows) for which the error norm between the original $\boldsymbol{A}$ and its approximation exceeds a predefined threshold. Applications in static citation networks with approximately $500,000$ nodes have shown that this method is successful in achieving high approximation accuracy with reduced memory usage and CPU time. The CMD procedure has also been adapted to time-transient problems where a sudden change in approximation accuracy suggests structural changes of communication patterns.

The same principle has been exploited in [32] in order to detect such patterns in more complex networks, consisting of various node types that require a higher-order tensorial description of their communication structure. The proposed technique is essentially a PCA-type decomposition that is performed over the various modes of the tensor. The authors present ways to perform this process for dynamic data by having time as an additional mode in the tensor (Dynamic Tensor Analysis or DTA for short). They also present a fast approximation to DTA, called the Streaming Tensor Analysis, which performs the updates based on the error's magnitude. Several tests on temporal data of $100,000$ dimensions and several thousand time steps have shown the merits of this approach in anomaly detection and pattern discovery, which is achieved with a relatively small computational burden and memory requirements.

5 Performance Metrics

In cases where the goal is to learn a probability distribution, say $\hat{f}(\boldsymbol{y})$ and the the test data are known to follow a known distribution say $f(\boldsymbol{y})$, then the Kullback-Leibler divergence $D(f||\hat{f})$ can be readily used to evaluate a model's accuracy:

$$D(f||\hat{f}) = -\int f(\boldsymbol{y}) \log \frac{f(\boldsymbol{y})}{\hat{f}(\boldsymbol{y})} d\boldsymbol{y} \tag{12}$$

The latter quantity is always non-negative and becomes zero only when $f \equiv \hat{f}$. Normalized (with respect to the entropy of f) or symmetrized versions of the above expression will also be suitable. In general however, the underlying distribution is not known as the collected data is not generated from an artificial model. In these cases, the performance metrics are problem dependent.

For classification tasks, algorithms are usually evaluated with respect to some test-data (i.e. labelled data) based on which a confusion matrix can be constructed. For the simplest case of binary labelling (i.e. 0 or 1), a confusion matrix contains the number of instances that belong to each of the cases seen in Figure 6.

A measure of accuracy is given by the ratio of correct predictions over the total number of predictions or equivalently by its complement– a.k.a. the error rate. Accuracy can be estimated for various threshold levels by constructing respective confusion

	Predicted 1	Predicted 0
True 1	true positive (hits)	false negative (misses)
True 0	false positive (false alarms)	true negative (correct rejections)

Figure 6: Confusion Matrix

matrices. The optimal threshold is naturally the one that maximizes accuracy. Good values for accuracy depend on the problem at hand and hence accuracy is not a generally applicable metric. Consider for example a test dataset in which 90% and 10% of the data are labelled with 1 and 0 respectively and a classification algorithm that always predicts 1. Then the accuracy value would be 90% but that does not necessarily imply a good labelling scheme. Accuracy however is the only measure from the ones discussed that generalizes to multiple classes.

A more sophisticated metric is the precision-recall curve, initially used in document retrieval applications. This is a x-y diagram where the horizontal axis contains the recall rate i.e. the ratio of true positives (hits) over the total number of true 1 (hits + misses) and the vertical axis depicts the precision rate i.e. the ratio of true positives (hits) over the total number of positives (hits + false alarms). The curve is constructed by calculating the precision-recall pair for various thresholds of the classification scheme. Scalar indicators commonly derived from the curve are called F_β-values. F_1-value is simply the harmonic average of the recall and precision rates i.e. $F = \frac{2\ \mathit{recall}\ \times\ \mathit{precision}}{\mathit{recall}\ +\ \mathit{precision}}$. The *break-even* point is the value for which recall equals precision.

Finally, a metric that is becoming more popular in machine learning problems and has better statistical foundations than most others is the Receiver Operator Characteristic plot or ROC curve (which is closely related to the precision-recall curve). Originally developed in the 1950's as a by-product of research into making sense of radio signals contaminated by noise, ROC curve is also an x-y diagram where the vertical axis contains the recall rate (also called sensitivity) and the horizontal axis the complement of specificity i.e. the ratio of false positives (false alarms) over the total number of True 0 (false alarms + correct rejections). Hence, sensitivity expresses the probability that the model will predict 1 when the true value is 1 and $1-$ specificity expresses the probability that it predicts 1 when in reality it is 0. The best possible prediction method would yield a point in the upper left corner of the ROC space i.e. 100% sensitivity and 100% specificity. A completely random predictor (i.e. one in which the prediction can be represented by the flipping of a coin independently of the values of the predictor variables $\boldsymbol{x}$) would lie on the $x = y$ line. Because most classifiers output a classification metric, e.g., a posterior probability on the two classes, one can generate a ROC by varying the decision threshold on this classification metric and computing the sensitivity and specificity for each decision threshold. For example, a decision threshold of 0.5 means that test samples with posterior probabilities greater or equal to 0.5 will be classified as positive samples. Greater decision thresholds will result in fewer true positives and false positives, while smaller decision thresholds lead to more true positives and false positives. ROC curves always start from $(0, 0)$ and end at $(1, 1)$ (Figure 7). Their most attractive property is that they are insensitive to changes in class distribution. If the proportion of positive to negative instances changes in a test set, the ROC curve will not change. The furthest away from the $x = y$ line, the better the performance of the classification algorithm. This can also be expressed by the Area Under the Curve (AUC value) which measures the average true positive rate of a classifier over the entire range of false positive rates. It is equivalent to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example. In general, classification models with $AUC > 0.9$ are considered excellent [4, 10]. If two ROC curves do not intersect then the method corresponding to the curve above is better.

A recent alternative to the ROC curve is the so-called cost curve which first appeared in [9] and was further developed in [8]. By associating a certain cost to each misclassification entry in the confusion matrix, the expected cost of the classifier can be explicitly represented. Performance (expected cost normalized to be between 0 and 1) is plotted on the $y-$axis. Operating points, meaning combinations of misclassification costs and class distributions, are plotted on the $x-$axis after being normalized to be between 0 and 1 by combining the parameters defining an operating point in the following way:

$$PCF(1) = \frac{p(1)C(0 \mid 1)}{p(1)C(0 \mid 1) + p(0)C(1 \mid 0)} \tag{13}$$

where $C(0|1)$ is the cost of misclassifying a example of class 1 as class 0, $C(1|0)$ is the cost of misclassifying a class 0 example as class 1, $p(1)$ is the probability of a class 1 example, and $p(0) = 1 - p(1)$. The motivation for this PCF definition, and cost curves more generally, originates in the simple situation when misclassification costs are equal. In this case, $PCF(1) = p(1)$ and the y-axis becomes error rate, so the cost curve plots how error rate varies as a function of the prevalence of class 1 examples. The PCF definition generalizes this idea to the case when when misclassification costs are not equal. The PCF formula is intimately tied to the definition of the slope of a line in ROC space, which plays a key role in ROC analysis. The x-axis of cost space is a slope in ROC space normalized to be between 0 and 1. There is a point/line duality between ROC space and cost space, meaning that a point in ROC space is represented by a line in cost space, and a line in ROC space is represented by a point in cost space. A classifier represented by the point (FP, TP) in ROC space is a line in cost space that has $y = FP$ when $x = 0$ and $y = 1 - TP$ when $x = 1$. The set of points defining an ROC curve become a set of lines in cost space. For example, the ROC curve in Figure 7 consists of eight points (including (0,0) and (1,1)). Each point becomes a line in cost space, i.e. the eight dotted lines in Figure 8. Corresponding to the convex hull of the points in ROC space is the lower envelope of the lines in cost space, indicated by the solid line in Figure 8. This expected cost representation, maintains many of the advantages of ROC representation, but is easier to understand. It allows the analyst to immediately see the range of costs and class frequencies where a particular classifier is best and quantify its superiority over other classifiers.

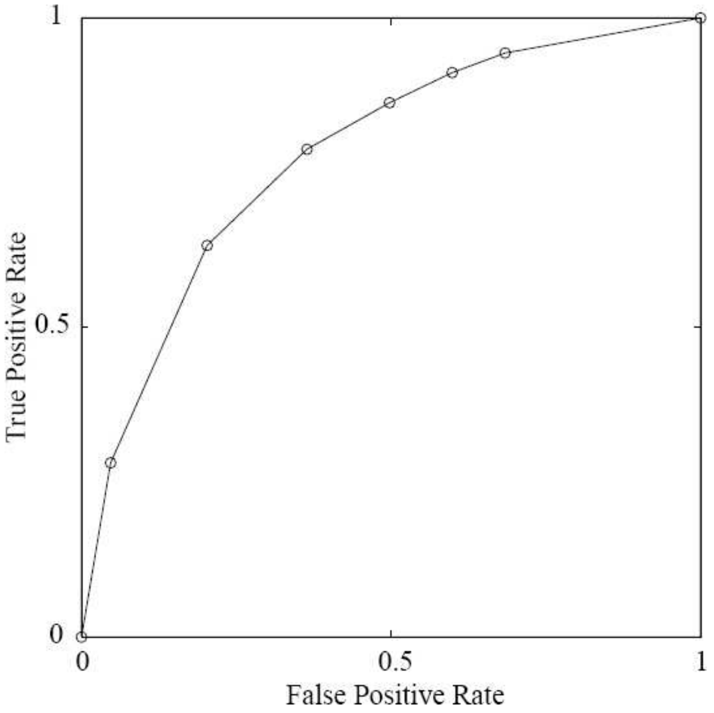


Figure 7: ROC curve: False Positive Rate = 1 - Specificity and True Positive Rate = Sensitivity. Figure taken from [8].

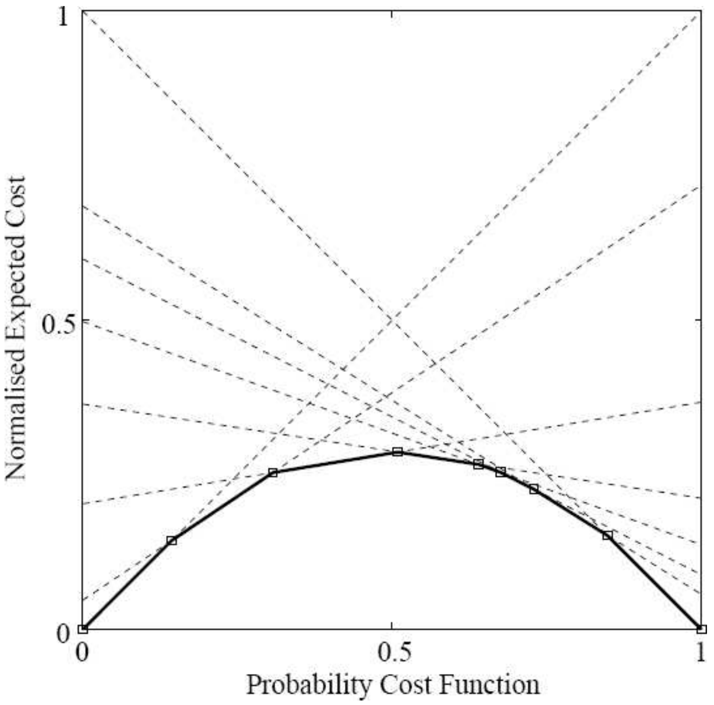


Figure 8: Cost curve corresponding to ROC curve in Figure 7. Each of the 8 points in the ROC curve become lines in the cost space. Figure taken from [8].

6 Conclusions

In this survey, we presented an overview of recently developed methodologies for dealing with relational data with particular emphasis to communication and transaction networks. We discussed two basic graphical models, namely discriminative and generative. The former models are particularly suited to classification or labelling tasks as they have the ability to learn distributions accurately based on a large number of features. They do not however provide information about the structural properties of the system and generally require larger amounts of data for training in comparison to generative models. Special attention was given to latent variable models as they are particularly applicable to capturing group formations and predicting links between nodes in a network. It should be noted that the majority of the literature is devoted to static graphs and extensions to dynamic problems are generally hampered by the increased computational effort. We have also discussed some recently developed methods which are applicable to very large graphs and are able to discover patterns and detect anomalies with relatively small computational requirements. Finally, we presented various performance metrics that have appeared in the literature with emphasis on classification problems.

Acknowledgements

This work was supported by FY06 funding of the Predictive Knowledge Systems Initiative. The valuable comments and suggestions of the following members of this research effort are gratefully acknowledged: Vera Bulaevskaya, Barry Chen, Tina Eliassi-Rad, Imola Fodor, Brian Gallagher, Bill Hanley, Gardar Johannesson, Brenda Ng, and John Nitao.

References

[1] D Blei, T Griffiths, M Jordan, and J Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *NIPS 2003*, 2003.

[2] D Blei and J Lafferty. Dynamic Topic Models. In *23rd ICML*, 2006.

[3] D Blei, A Ng, and M Jordan. Latent dirichlet allocation. In *NIPS 2002*, 2002.

[4] B Chen, T Hickling, M Krnjajic, B Hanley, G Clark, J Nitao, D Knapp, L Hiller, and M Mugge. Multi-Layer Perceptrons and Support Vector Machines for Classifying the PAT Data Sets . LLNL - Internal Unclassified Report.

[5] A Culotta, D Kulp, and A McCallum. Gene prediction with conditional random fields. Technical Report Technical Report UM-CS-2005-028, University of Massachusetts, Amherst, 2005.

[6] A Culotta and A McCallum. Joint deduplication of multiple record types in relational data. In *Fourteenth Conference on Information and Knowledge Management (CIKM)*, page 2005.

[7] P Drineas, R Kannan, and M Mahoney. Fast Monte Carlo algorithms for matrices III : Computing compressed matrix decompositions. *SIAM Journal of Computing*, 2005.

[8] C Drummond and RC Holte. What roc curves can't do (and cost curves can). In *Proceedings of the ROC Analysis in Artificial Intelligence, 1st International Workshop. Valencia, Spain*, pages 19–26, August 22, 2004.

[9] Chris Drummond and Robert Holte. Explicitly representing expected cost: An alternative to roc representation. In *Proceedings of Conference in Knowledge Discovery and Data Mining 2000*, 2000.

[10] T Fawcett. ROC graphs: Notes and practical considerations for data mining researchers . Technical report, Tech report HPL-2003-4, HP Laboratories, Palo Alto, CA, USA, 2003.

[11] MS Handcock, Raftery AE, and JM Tantrum. Model-based clustering for social networks. April 27, 2005.

[12] P Hoff. Multiplicative latent factor models for description and prediction in social networks. January 17, 2006.

[13] P. Hoff, AE Raftery, and MS Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090, 2002.

[14] S. Jain and RM Neal. A Split-Merge Markov Chain Monte Carlo Procedure for the Dirichlet Process Mixture Model. *Journal of Computational and Graphical Statistics*, 13:158–182, 2004.

[15] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *AAAI 2006*, 2006.

[16] S Kok and P Domingos. Learning the structure of markov logic networks. In *Proceedings of the Twenty-Second International Conference on Machine Learning*, pages 441–448, 2005.

[17] J Kubica, A Moore, J Schneider, and Y Yang. Stochastic link and group detection. In *AAAI*, 2002.

[18] J Lafferty, A McCallum, and F Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML-2001*, 2001.

[19] VK Mansinghka, C. Kemp, J. B. Tenenbaum, and T. L. Griffiths. Structured priors for structure learning. In *UAI 2006*, 2006.

[20] A McCallum. Efficiently inducing features of conditional random fields. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2003.

[21] A McCallum, A Corrada-Emmanuel, and X Wang. Topic and role discovery in social networks. In *IJCAI 2005*, 2005.

[22] T Minka. Discriminative models, not discriminative training. Technical Report TR-2005-144, Microsoft Research, 2005.

[23] RM Neal. Markov chain sampling methods for Dirichlet process mixture models. Technical Report No. 9815, Dept. of Statistics, University of Toronto, 1998.

[24] K Nowicki and Snijders TA. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96:1077–1087, 2001.

[25] C Pal, C Sutton, and A. McCallum. Sparse forward-backward using minimum divergence beams for fast training of conditional random fields. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2006.

[26] A Popescul and LH Ungar. Structural logistic regression for link analysis. In *Workshop on Multi-Relational Data Mining at KDD*, 2003.

[27] A Popescul, LH Ungar, S Lawrence, and DM Pennock. Statistical relational learning for document mining. In *Proceedings of IEEE International Conference on Data Mining (ICDM)*, 2003.

[28] M Richardson and P Domingos. Markov logic networks. Technical report, Dept. Comp. Sci. and Eng., University of Washington, 2004.

[29] P Sarkar and AW Moore. Dynamic social network analysis using latent space models. In *SIGKDD Explorations: Special Edition on Link Mining*, 2005.

[30] S Shortreed and MS Handcock. Positional estimation within the latent space model for networks. April 9, 2004.

[31] P Singla and P Domingos. Discriminative training of markov logic networks. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*, pages 868–873, 2005.

[32] J Sun, D Tao, and C Faloutsos. Beyond streams and graphs: Dynamic tensor analysis. In *KDD 2006*, Philadelphia, PA, USA, 2006.

[33] J Sun, Y Xie, H Zhang, and C Faloutsos. Compact matrix decomposition for large graphs: Theory and practice. In *32nd VLDB Conference*, Seoul, Korea, 2006.

[34] C Sutton and A. McCallum. Composition of conditional random fields for transfer learning. In *Proceedings of Human Language Technologies / Emprical Methods in Natural Language Processing (HLT/EMNLP)*, 2005.

[35] C Sutton and A McCallum. An introduction to conditional random fields for relational learning. In L Getoor and B Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2006 (to appear).

[36] C Sutton, K Rohanimanesh, and A McCallum. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In *ICML*, 2004.

[37] B Taskar. *Learning Structured Prediction Models: A Large Margin Approach*. PhD thesis, Stanford University, 2004.

[38] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proc. Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, Edmonton, Canada, 2002.

[39] B. Taskar, V. Chatalbashev, and D. Koller. Learning associative markov networks. In *Proceedings of the Twenty-First International Conference on Machine Learning (ICML)*, 2004.

[40] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *Advances in Neural Information Processing Systems (NIPS 2003)*, Vancouver, Canada, 2004. Winner of the Best Student Paper Award.

[41] B. Taskar, M.-F. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *Advances in Neural Information Processing Systems (NIPS 2003)*. Vancouver, Canada, 2004.

[42] X Wang and A McCallum. Topics over time: A non-markov continuous-time model for topical trends. In *ACM SIGKDD-2005*, 2005.

[43] X Wang, N Mohanty, and A 2005 McCallum. Group and topic discovery from relations and text. In *KDD Workshop on Link Discovery: Issues, Approaches and Applications (LinkKDD)*, 2005.

[44] J Weinman, A Hansen, and A McCallum. Sign detection in natural images with conditional random fields. In *IEEE International Workshop on Machine Learning for Signal Processing*, 2004.

[45] X Zhu, Z Ghahramani, and J Lafferty. Time-sensitive Dirichlet Process Mixture Models. Technical Report CMU-CALD-05-104, School of Computer Science, Carnegie Mellon, 2005.