Event Classification and Relationship Labeling in Affiliation Networks

Abstract

Many domains are best described as an affiliation network in which there are entities such as actors, events and organizations linked together in a variety of relationships. Relational classification in these domains requires the collective classification of both entities and relationships. In this paper, we investigate the use of relational Markov networks (RMN) for relational classification in affiliation networks. We study two tasks, event classification and relationship labeling, and discuss general issues in constructing an appropriate RMN from an affiliation network. We evaluate our methods on a novel dataset describing terrorist affiliation networks which includes data about actors (terrorists), events (terrorist attacks) and organizations (terrorist organizations). Our results highlight several important issues concerning the effectiveness of relational classification and our experiments show that the relational structure significantly helps relationship labeling.

1. Introduction

Traditional machine learning techniques mainly concentrate on identically and independently distributed samples. However, most real-world datasets are relational in nature and the correlations due to the link structure provide an important source of information. Recent research has focused on making use of the relational structure to improve the quality of prediction. Some of these works include Relational Markov networks (RMN) (Taskar et al., 2002), Conditional Random fields (CRFs) (Lafferty et al., 2001), Relational Dependency networks (RDNs) (Neville & Jensen, 2004), Lu and Getoor (2003) etc. We refer the interested reader to Getoor and Diehl (2005) for a survey of recent research in this area.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

In this paper, we study the application of relational learning techniques to affiliation networks. Affiliation networks describe the associations of actors and events and are a common form of two-mode social networks. The particular dataset that we study is a three-mode network that describes people, events and organizations.

We investigate two tasks in these networks: **event classification** and **relationship labeling**. In event classification, we are trying to predict the label on a node. In our case, we predict event labels. In relationship labeling, we are trying to label a relationship between two actors. Here we predict the labels on actor-actor relationships. One of the non-trivial aspects in this domain is how the three-mode network can be transformed to support each task. Also, while there has been a significant amount of work on relational classification, there has been much less work done on link labeling.

The affiliation network which we study comes from the Profile in Terror (http://profilesinterror.mindswap.org/) (PIT) project. The PIT knowledge base captures terrorism intelligence extracted from various sources like news media reports. The knowledge base describes a network of terrorists, terrorist attacks and terrorist organizations. One of the challenges in this domain is that the intelligence is scarce in nature and information available can be partial and incomplete.

In this paper, we apply relational classifiers to perform two tasks: terrorist attack classification and terrorist relationship labeling. For comparison, we provide as a baseline, the results returned by a content-only maximum entropy classifier (flat model). We hypothesize that by considering the relational structure, the accuracy of classification and labeling using a RMN will be higher than the flat model which uses features on data entities. Interestingly, our results for this domain show that the relational structure does not help significantly with event classification accuracy; on the other hand, it is of significant importance for relationship labeling. We discuss both the obvious and less obvious reasons for this difference in importance.

The remainder of the paper is organized as the follows. Section 2 describes the PIT knowledge base. Section 3 describes relational Markov networks, the relational classifier we used in our experiments. Section 5 describes the event classification problem and Section 6 describes the relationship labeling problem. In Section 7 we discuss our findings and conclude with future work in Section 8.

2. PIT Knowledge Base

PIT is a comprehensive semantic knowledge base capturing relevant information designed to help analysts analyze past information, share common knowledge and respond, and hopefully prevent, terrorism threats. It is not merely a data portal. The schema of PIT is described in OWL and the knowledge base is stored in a RDF datastore. The information is stored in a machine accessible way to facilitate information exchange and reasoning. It also provides a platform for studying advanced and new methodologies for predictive modeling, terrorist (social) network analysis, and visualization of terrorists activities and relationships.

PIT contains all relevant knowledge about facilities, activities, terror attacks, terrorists and organizations. Facilities include places where terrorist are trained and offices which are used as activities bases. Activities include different forms of events like raising funds, recruiting members and smuggling weapons. Terrorists plot and execute different types of attacks: arson, bombing, kidnapping, NBCR (Nuclear, Biology, Chemistry and Radiology) attack, and weapon attack. Terrorists function by participating and forming organizations including terrorist organizations and criminal gangs. The core of a terrorism knowledge domain are terrorists, terrorist attacks and terrorist organizations. These are the actors, events and organizations respectively in the PIT network. Each PIT object is described by a set of features. For example, each attack has a short "label" which is a one sentence summary and "description" consisting of a paragraph with a detailed description. Each attack also has features describing when the attack happened, the location of the attack, the number of injured people and the number of people killed.

Different entities in knowledge base are related to each other in complex ways to form a network. Terrorist attacks are linked to the organizations that planned the attacks. Terrorists are linked to each other if they contact each other, use the same facility, are members of the same family, or belong to the same terrorist organization. Terrorists are considered to have been in contact with each other if they attend the same meeting, they communicate (via a phone call, an email or text message) with each other, or they transfer money to each other. Terrorists are considered to use the same facility if they go to the same mosque, attend the same training camp or work in the same office. Understanding the various relations can help analysts uncover the underlying network of terrorists as well as terrorist organizations and identify ongoing activities and plans.

3. Relational Markov Networks

Undirected graphical models or *Markov networks* (Cowell et al., 1999) have been shown to be an effective way to represent diverse classification problems and correlations due to the link structure. Due to the flexibility they offer, all our experiments with the PIT dataset were performed using *Relational Markov networks* (RMNs) (Taskar et al., 2002), an extension of Markov networks to relational domains. Here we review the RMN framework.

Let $\mathbf{V}$ be a set of discrete random variables, and let $\mathbf{v}$ be an assignment of values to the random variables. A Markov network is described by a graph $G = (\mathbf{V}, E)$ and a set of parameters Ψ. Let $C(G)$ denote a set of (not necessarily maximal) cliques in G. For each $c \in C(G)$, let V_c denote the nodes in the clique. Each clique c has a clique potential $\psi_c(V_c)$ which is a non-negative function on the joint domain of V_c and let $\Psi = \{\psi_c(V_c)\}_{c \in C(G)}$. For classification problems we are often interested in conditional models. Let $\mathbf{X}$ be the set of observed random variables we condition on and let $\mathbf{x}$ denote the observed values of $\mathbf{X}$. Let X_c denote the observed random variables in clique $c \in C(G)$ and let x_c denote the observed values of X_c. Let $\mathbf{Y}$ be the set of target random variables to which we want to assign labels and let $\mathbf{y}$ denote an assignment to $\mathbf{Y}$. Let Y_c denote the set of target random variables in clique $c \in C(G)$ and let y_c denote an assignment to it. A *conditional Markov network* or *conditional random field* is a Markov network (G, Ψ) which defines the distribution $P(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in C(G)} \psi_c(x_c, y_c)$ where $Z(\mathbf{x}) = \sum_{\mathbf{y}'} \prod_c \psi_c(x_c, y'_c)$.

Conditional Markov networks, as presented above, are not suited for event classification tasks since they involve clique specific potentials $\psi_c(V_c)$. RMNs (Taskar et al., 2002) are an extension of the Markov network framework to relational domains where we define the clique potentials in log-space using a small set of feature functions $\log \psi_c(y_c, x_c) = \sum_i w_i f_i(x_c, y_c)$ where f_i is the i^{th} feature function (usually a simple indicator function) and w_i is a parameter which needs to be estimated.

Parameter estimation for RMNs can be performed using gradient-based optimization methods from fully labeled training data (Taskar et al., 2002). Taskar et al. also show that to estimate the gradient one needs to perform inference over the training data. In relational domains, the underlying Markov network is usually large and densely connected making exact inference infeasible. Thus Taskar et al. propose the use of approximate inference methods like loopy belief propagation (Yedidia et al., 2000).

4. Constructing the Markov network

Due to the multi-relational nature of the dataset we have a wide variety of ways to construct the Markov network.

4.1. Markov network for Event Classification

In event classification, the object labels depend on related object labels. But objects can be related in a variety of ways, and it is not always clear which relationship or relationships to focus on.

For event classification, specifically terrorist attack classification, we considered two different ways to relate events:

- *loc RMN*: we connect each pair of attacks that occurred in the same geographical location.
- *loc+org RMN*: we connect each pair of attacks that occurred in the same geographical location and were organized by the same organization.

4.2. Markov network for Relationship Labeling

In relationship labeling, the labels of the relations between entities are correlated with each other. A convenient way to capture this correlation is to represent it as a Markov network where each node refers to an actor and each edge is a relation connecting two actors. Unfortunately, this *actor graph* is not conducive to the problem of relationship labeling where we need to have random variables representing the labels on the relations. We apply a simple transformation to the actor graph to construct a new graph, the *link graph*, where *each node represents a relation* and *each edge connects two relations having a common actor*. More specifically, let t_i, t_j and t_k represent actors in the actor graph and let r_{ij} and r_{jk} denote the relations connecting t_i, t_j and t_j, t_k respectively. In the link graph, we introduce an edge connecting every pair of such relations r_{ij} and r_{jk} since they have an actor in common (viz. t_j). We report experiments with RMNs that included cliques for each edge in such link graphs (*dyad RMNs*). In Figure 1 we show a small subset of the terrorist graph from the PIT knowledge base (Figure 1 (a)) and the corresponding link graph (Figure 1 (b)). Notice that the link graph contains dense clusters of nodes and greater number of edges.

The main intuition behind connecting nodes representing relations is to exploit the correlation amongst labels on relations connecting the same actors. Our experiments indicate that relations involving the same actors often have the same labels. Taskar et al. (2004) used a similar approach to classify hyperlinks connecting university webpages.

One of the problems we faced while experimenting with dyadic link graphs generated with the above described approach is that the generated graph is often too dense for RMNs to handle. In particular, approximate inference techniques like loopy belief propagation (Yedidia et al., 2000) are known to provide poor approximations when there are a number of short, closed loops (Yedidia et al., 2005) (a direct consequence of high link density) in the underlying Markov network. Due to the poor quality of inference, the parameter estimation for RMNs often did not converge to desirable values.

In an effort to reduce the edge density in the link graphs and to determine whether other types of correlations exist in the dataset we experimented with what Taskar et al. (2004) refers to as *transitivity* patterns. Let t_i, t_j and t_k represent three terrorists in the terrorist graph and let r_{ij}, r_{jk} and r_{ki} represent the corresponding relationships then we introduce a clique amongst r_{ij}, r_{jk} and r_{ki} in the link graph. This type of a clique generation procedure captures correlations where transitivity holds, e.g., suppose t_i belongs to the same family as t_j and t_i shares a familial bond with t_k then t_j and t_k also belong to the same family. To keep the edge density of the generated link graphs in check we did not consider the dyadic introduced previously while experimenting with such *triad* cliques (triad RMNs).

5. Event Classification

We begin by describing the PIT data which is used for event classification and then describe our results.

5.1. Terrorist Attack Dataset

In the PIT knowledge base, there are a total 1,293 terror attacks each classified into one of six classes: **arson** (2.4%), **bombing** (43.5%), **kidnapping** (13.8%), **NBCR** (0.6%), **other attack** (1%) and **weapon attack** (38.5%). We note that bombing, weapon attack

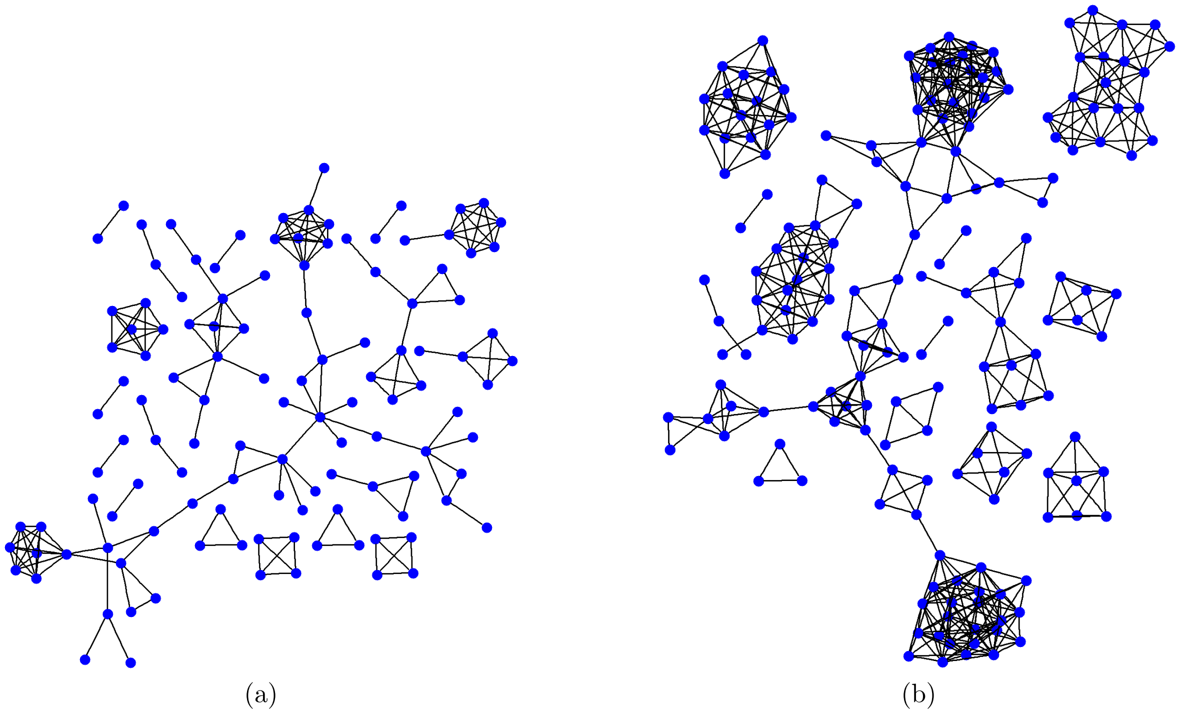


Figure 1. (a) A terrorist graph from the PIT dataset consisting of 181 edges and (b) the corresponding link graph consisting of 580 edges

and kidnapping are the three major attack types and they dominate the dataset. NBCR (Nuclear, Biology, Chemistry and Radiology) and other attack only have less than 1% data in the dataset. We begin exploring the dataset by considering the problem of collectively assigning each attack instance its correct class label.

5.2. Experiments

We split the dataset into three sets each containing around 430 instances to be labeled and performed three-fold cross validation. Each set was created using stratified sampling so all sets contain the same distribution of class labels.

As part of the evidence, we include with each terror attack instance various types of information such as year of attack, keywords from a description written by a human etc. As a baseline, we compare the various RMNs against a content-only maximum entropy classifier (flat model).

For each classifier, we assume a "shrinkage" prior and compute the MAP estimate of the parameters. More precisely, we assumed that different parameters are a priori independent and define $p(w_i) = \lambda w_i^2$. We tried a range of regularization constants for each classifier and found that $\lambda = 10$ returned the best results. Taskar et al. (2002) report using a regularization constant of the same magnitude $\lambda \approx 5.5$.

	Flat	RMN loc	RMN loc+org
Avg. Accuracy	87.06	86.93	87.1

Table 1. Average classification accuracy of terrorist attacks

As Table 1 shows, the RMNs and the flat model return almost identical performance and there is not much to choose between them. The reason for this turns out to be the high quality of evidence we considered. Each terror attack is accompanied by a description that was written by a human while entering the terror attack into the knowledge base. This description frequently contains some highly predictive words, e.g., "explosion" or "detonated" in the case of a bombing etc.

In future, we aim to exclude this type of human written evidence and use other type of automatically gathered evidence to find out if machine learning classifiers can label effectively.

6. Relationship Labeling

At the core of terrorist activity is a network of personal connections that allows the terrorist organization to function. Consequently, looking at who knows whom and how they are related to each other is central to understanding the extent of terrorist activities. Intelligence information can show that two terrorists are related in some ways but in what exact way is often unknown. Therefore it is important to understanding the nature of the relation structure amongst the terrorists from the known data and be able to label all the unknown relations.

6.1. Terrorist Relation Dataset

In the PIT knowledge base, there are a total 435 terrorists hand-classified into one of four types: *terrorist* (70%), *terrorist leader* (11%), *politician* (6%) and *people* (a grab bag comprising of individuals who are not assigned to any particular type, 13%). The terrorists are connected by a total 917 binary relations. Each relation is hand-tagged with one or more labels describing the nature of the relation:

- **accomplice** (53.1%): An accomplice relation means two people are members of the same terrorist organization.
- **family** (14.8%): A family relation means two people are in the same family (e.g. father-son, husband-wife, uncle-nephew, cousin-cousin).
- **contact** (19.6%): A contact relation means two people have contacted each other (e.g. attend the same meeting, email each other, call each other via phone).
- **congregate** (12.4%): A congregate relation means two people use the same facility (e.g. go to the same training camp).

Our aim is to assign each relation its correct set of labels. Since a relation can have more than one label, this problem is an instance of *relational multi-label classification*.

6.2. Experiments

We split the terrorist relation dataset into two sets and performed two-fold cross validation. Each set was created using stratified sampling so both sets contain the same distribution of class labels.

As part of the evidence, we included with each relation various types of information belonging to the terrorists involved in the relation like nationality, words from their biography, the label on the terrorist etc. We compared the various RMNs against a baseline content-only maximum entropy classifier (flat).

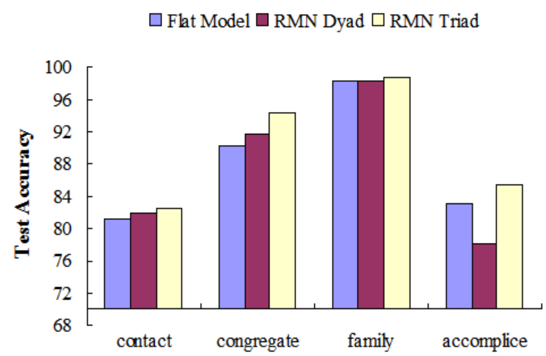


Figure 2. The average classification accuracy for binary terrorist relationship labeling.

Just as before, for each classifier, we assumed a "shrinkage" prior and compute the MAP estimate of the parameters using a regularization constant of 10.

6.2.1. Multi-label classification results

We first report the results of the *relational multi-label classification* experiments. A simple way to perform multi-label classification is to learn numerous binary one-against-the-rest classifiers. Thus we learn four different types of classifiers one for each of accomplice, family, contact and congregate. The results are shown in Figure 2.

Figure 2 shows that the triad RMN always does better than the flat model. As we remarked earlier, the dyad RMN sometimes (in the case of accomplice) fails to improve upon the results of the flat model due to the excessive link density. Another reason for the lower than expected accuracy for the dyad RMN could be due to the loss of information of reducing all out-of-class relations to the same label. For example, when classifying for contact, all relations labeled congregate, family or accomplice will be reduced to non-contact thus losing information in the form of correlations.

As part of our future work we aim to utilize methods such as Ghamrawi and McCallum (2005) to perform collective multi-label classification.

6.2.2. Single-label multi-class classification results

As part of our efforts to perform some experiments on multi-class data we obtained a single label dataset by throwing out all the relations with multiple labels. This reduced our dataset from 917 to 884 relations. Thus we obtained a single label multi-class dataset.

Figure 3 shows the results on this dataset for the

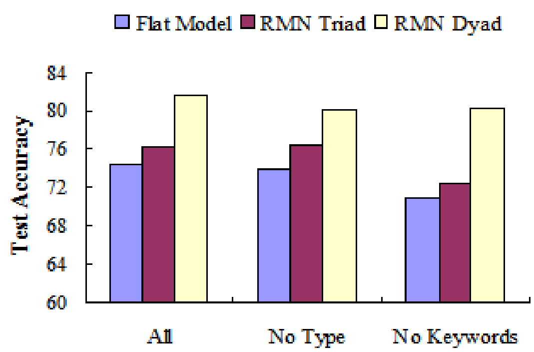


Figure 3. Average classification accuracy of terrorist relation labeling.

three classifiers flat, dyad RMN and triad RMN on the multi-class classification problem. Notice that in Figure 3 the dyad RMN performs consistently better than the triad model and the triad RMN consistently improves upon the results of the flat model showing that both dyad and triad cliques can be useful for relationship labeling. One of the reasons for the dyad RMN doing better than the triad RMN could be the fact that there are a lot more links in the dyad RMN thus allowing the inference procedure to exploit more correlations that exist in the link structure.

In Figure 3, we also report the results of experiments without certain features. In particular, Taskar et al. (2004) report that relation classification (in their case hyperlinks) may be improved if one includes as part of the evidence the labels on the entities (in their case the webpages) themselves. In Figure 3, the "No Type" results were obtained by not including in the set of features comprising of the class labels on the terrorists (leader, terrorist etc.). We confirm that including the labels on the entities as evidence aids the relationship classification. The set of results labeled "No Keywords" were obtained by not utilizing the biographies of the terrorists as evidence which happens to be a substantial part of the feature set and the results show that relational methods can do well even when there is a dearth of evidence.

7. Discussion

We were confronted with a number of issues during the process of pre-processing and experimenting with the PIT dataset. Here we discuss each issue in turn with the hope of identifying important avenues for future work.

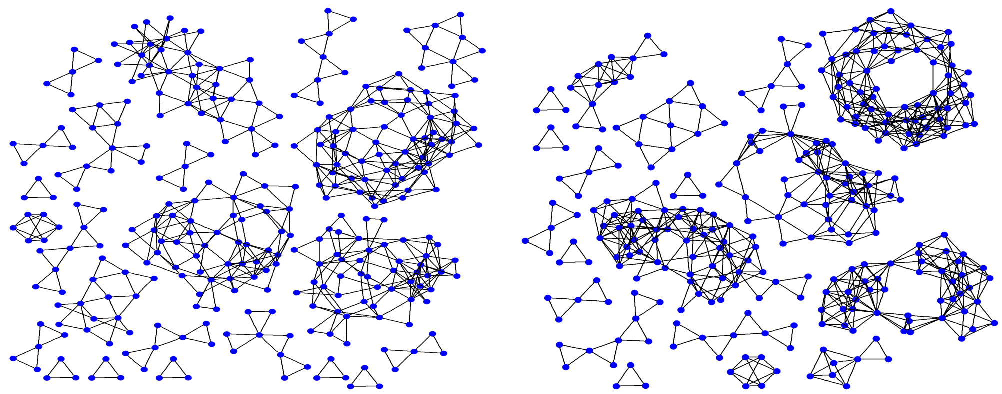
One of the issues that usually comes up during the pre-processing of relational datasets is how to construct training and test datasets? It is not always the case that the dataset itself provides subsets that are natural splits such as the university splits in WebKB (Craven et al., 1998) where each split forms a disjoint graph. One common approach used to create training and test splits for identically and independently distributed samples is to create randomly sampled stratified subsets of the data so that each subset contains the same distribution of class labels. This approach fails on two counts in the case of relational data:

- Random sampling may cause linked entities to fall into different subsets. The links that go from one subset to another are usually ignored during parameter estimation if both the subsets are not used for learning and this means that we are ignoring some information and not using the data fully.

- The intuition behind creating training and test sets is to make sure that they come from the same distribution. Unfortunately, since random stratified sampling does not look at the links, it may be the case that we construct splits containing an unequal number of links. Figure 4 shows two splits that were created from the terrorist relation PIT dataset. Note that Figure 4 (b) is much denser than Figure 4 (a). Clearly, these two splits do not represent the same distribution.

Another important issue that comes up when dealing with relational datasets is the problem of high link density. Common approximate inference techniques, loopy belief propagation (Yedidia et al., 2000) in particular, face problems when run on datasets with high link density, in particular, if the dataset contains numerous densely clustered nodes forming short, closed loops (Yedidia et al., 2005). This usually causes the approximate inference approach to return a poor approximation resulting in poor quality inference.

Our experience with the PIT dataset shows that this dataset is quite different from common relational datasets such as WebKB (Craven et al., 1998) or Cora (McCallum et al., 2000). The PIT dataset contains a larger number of clusters of nodes making it a much more challenging dataset. We hope that such datasets with markedly different properties will help researchers in the field identify new and interesting problems to work on.



(a) (b)

Figure 4. Naively created splits of different densities using stratified sampling. (a) A split comprising of 442 nodes and 225 edges. (b) A split comprising of 442 nodes and 283 edges.

8. Conclusion and Future Work

In this paper we introduce a novel dataset, the PIT knowledge base, and show that affiliation networks can provide a rich source of relational data. We apply relational classifiers to perform two tasks on the PIT knowledge base: event classification and relationship labeling. We compare the performance of the relational classifier against a content-only maximum entropy classifier.

Our experimental results show that a RMN can improve the accuracy and outperform traditional flat algorithms if good cliques can be identified. We need to study further how to form cliques and identify general clique patterns which can be applied across different domains. There is a trade-off between the effectiveness and efficiency of the RMN and we are interested in investigating the optimal density and how to select among different cliques in affiliation networks.

References

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., & Spiegelhalter, D. J. (1999). *Probabilistic networks and expert systems*. Springer, New York.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., & Slattery, S. (1998). Learning to extract symbolic knowledge from the world wide web. *Proceedings of the AAAI*.

Getoor, L., & Diehl, C. (2005). Link mining: A survey. *SIGKDD Explorations vol. 7 issue 2*.

Ghamrawi, N., & McCallum, A. (2005). Collective multi-label classification. *CIKM*.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. 18th International Conf. on Machine Learning*.

Lu, Q., & Getoor, L. (2003). Link-based classification. *Proceeding of 12th International Conference on Machine Learning*.

McCallum, A., Nigam, K., Rennie, J., & Seymore, K. (2000). Automating the construction of internet portals with machine learning. *Information Retrieval*.

Neville, J., & Jensen, D. (2004). Dependency networks for relational data. *Proceedings of the IEEE International Conference on Data Mining*.

Nigam, K., Lafferty, J., & McCallum, A. (1999). Using maximum entropy for text classification. *IJCAI-99 Workshop on Machine Learning for Information Filtering*.

Taskar, B., Abbeel, P., & Koller, D. (2002). Discriminative probabilistic models for relational data. *18th Conference on Uncertainty in Artificial Intelligence*.

Taskar, B., Wong, M.-F., Abbeel, P., & Koller, D. (2004). Link prediction in relational data. *Neural Information Processing Systems*.

Yedidia, J., Freeman, W., & Weiss, Y. (2005). Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory* (pp. 2282–2312).

Yedidia, J., Freeman, W. T., & Weiss, Y. (2000). Generalized belief propagation. *Neural Information Processing Systems* (pp. 689–695).