Robust Collective Classification with Contextual Dependency Network Models*

Yonghong Tian[1], Tiejun Huang[1,2], and Wen Gao[1,2]

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China
[2] Digital Media Institute, Peking University, Beijing 100080, China
{yhtian, tjhuang, wgao}@jdl.ac.cn

Abstract. In order to exploit the dependencies in relational data to improve predictions, relational classification models often need to make simultaneous statistical judgments about the class labels for a set of related objects. Robustness has always been an important concern for such collective classification models since many real-world relational data such as Web pages are often accompanied with much noisy information. In this paper, we propose a contextual dependency network (CDN) model for classifying linked objects in the presence of noisy and irrelevant links. The CDN model makes use of a dependency function to characterize the contextual dependencies among linked objects so that it can effectively reduce the effect of irrelevant links on the classification. We show how to use the Gibbs inference framework over the CDN model for collective classification of multiple linked objects. The experiments show that the CDN model demonstrates relatively high robustness on datasets containing irrelevant links.

1 Introduction

Many real-world datasets are characterized by the presence of complex relational structure: Web, bibliographic data, social networks, epidemiological records, etc. In such relational data, entities are related to each other via different types of relations (e.g., hyperlinks, citations, friendships). For classification of relational data, the relational structure can be exploited to achieve better predictions. Often, relational classification models need to make simultaneous statistical judgments about the class labels for a set of related objects, rather than classifying them separately. Clearly, such collective classification models are capable of significantly improving probabilistic inference in relational data [1].

Recently, some combinative relational classification (CRC) algorithms (e.g., [4][5] [6][16][17]) have been proposed for classification of link data by integrating relational feature generation into traditional machine learning algorithms. Due to the implementation simplicity, CRC algorithms are often used as the baselines for classification of link data. For example, this paper uses neighborhood iterative classification (NIC)[4] and linkage logistic regression (LLR)[6] as baseline link-based models. Several researchers also proposed statistical relational models (SRMs) to characterize the correlation between link data, e.g., probabilistic relational models (PRMs)[7], probabilistic entity-relationship models (PERs)[15], relational Markov networks (RMNs)[2], Markov logic

* This work is supported by China-America Digital Academic Library project (grant No. CADAL2004002).

X. Li, O.R. Zaiane, and Z. Li (Eds.): ADMA 2006, LNAI 4093, pp. 173–180, 2006.
© Springer-Verlag Berlin Heidelberg 2006

networks (MLNs)[10] and relational dependency networks (RDNs)[8][9]. These models allow the specification of a probability model for types of objects, and also allow attributes of an object to depend probabilistically on attributes of other related objects. Thereinto, RDNs offer several advantages over the other models, including the interpretable representation that facilitates knowledge discovery in relational data, the ability to represent cyclic dependencies, and the simple but efficient methods for learning model structure and parameters [9].

However, the link structure in real world is more complex. Links can be from an object to another object of the same topic, or they can point at objects of different topics. The latter are sometimes referred to as "noise" when they do not provide useful and predictive information for categorization. To perform robust reasons in such "noisy" data sets, this paper proposes a contextual dependency network (CDN) model. Similar to RDNs, the CDN model also uses dependency networks (DNs) [11] for modeling relational data. On top of the DN framework, we introduce additional parameters called dependency functions to directly capture the strengths of dependencies among linked objects. In this way, CDNs can effectively reduce the effect of the irrelevant links on the classification. Moreover, we also show how to use the Gibbs inference framework over the learned CDN model for collective classification of multiple linked objects. Experiments were performed on Cora and WebKB to compare the classification performance of CDNs with RDNs and two baseline link-based models. The experimental results indicate that the CDNs can scale well in the presence of noise.

We present the formulation, learning and inference of CDNs in Section 2. Experiments and results are presented in Section 3. We conclude the paper in Section 4.

2 Contextual Dependency Network Model

2.1 Relational Data

In general, link data can be viewed as an instantiation of a relational schema $\mathcal{S}$ where entities are interconnected. A schema specifies a set of object types $\mathbf{T}$. Each object type $t \in \mathbf{T}$ is associated with a set of attributes. Moreover, a link dataset can be represented as a directed (or undirected) graph $\mathcal{G}_\mathrm{D}=(\mathcal{O}_\mathrm{D}, \mathcal{L}_\mathrm{D})$, where $o_i \in \mathcal{O}_\mathrm{D}$ the node denotes an object (e.g., authors, papers) and the edge $o_i \rightarrow o_j \in \mathcal{L}_\mathrm{D}$ denotes a link from o_i to o_j (e.g., author-of, citation). Clearly, attributes of an object can depend probabilistically on its other attributes, and on attributes of other linked objects in $\mathcal{G}_\mathrm{D}$.

The link regularities in many real-world data are very complex. That is, many real-world link data such as Web pages may well exhibit the "partial co-referencing" regularity, i.e., objects with the same class tend to link to objects that are semantically related to each others, but also link to a wide variety of other objects without semantic reason [3]. Clearly, links are less informative in this case, but sometimes also provide additional information for the classification of the objects in question. Instead of eliminating these links outright, the approach that we take in CDN is to weigh these links differently through dependency functions that can be learn from the training data set.

2.2 Model Formulation

Dependency networks (DNs) [11] are probabilistic graphical models that are similar to Bayesian Networks (BNs). They differ in that the graphical structures of DNs are not required to be acyclic. A DN $\mathcal{D}=(\mathcal{G}, \mathbf{P})$ encodes the conditional independence constraints that each variable is independent of all other variables in $\mathbf{X}$ given its parents, where the direct graph $\mathcal{G}$ encodes the dependency structure and $\mathbf{P}$ is a set of conditional probability distributions (CPDs) satisfying $p(X_i|\mathbf{Pa}_i) = p(X_i|\mathbf{X}\backslash X_i)$ for each $X_i \in \mathbf{X}$ ($\mathbf{Pa}_i$ denotes the parents of X_i). An advantage of DNs is that for both structure learning and parameter estimation, the CPD for each variable can be learned independently using any standard classification or regression algorithm [11].

By simply extending DNs to a relational setting, RDNs [8][9] use a bidirected model graph $\mathcal{G}_{\mathrm{M}}$ with a set of CPDs $\mathbf{P}$). Each node in $\mathcal{G}_{\mathrm{M}}$ corresponds to an attribute A_i^t and is associated with a CPD $p(a_i^t|\mathbf{Pa}(a_i^t))$. The RDN learning algorithm is much like the DN learning algorithm, except it uses relational probability trees (RPTs) to learn CPDs [9]. However, the link structure is not a part of the probabilistic model of RDNs, thus we cannot predict links and more importantly use the links to improve prediction about other attributes in the model [7].

Instead of specifying a single CPD for the class label of an object with type given both other attributes of that object and attributes of other related objects (as in RDNs), CDNs define two CPDs: one for capturing *intrinsic dependency* and the other for capturing *relational dependency*. More formally,

$$P(C_i|\mathbf{Pa}(C_i)) = \alpha_t P(C_i|\mathbf{Pa}^{(\mathrm{L})}(C_i)) + (1-\alpha_t)P(C_i|\mathbf{Pa}^{(\mathrm{N})}(C_i)), \tag{1}$$

where $\mathbf{Pa}^{(\mathrm{L})}(C_i)$ denotes the "local" parents of C_i (i.e., attributes in $\mathbf{Pa}^{(\mathrm{L})}(C_i)$ are associated with object o_i), $\mathbf{Pa}^{(\mathrm{N})}(C_i)$ denotes the "networked" parents of (C_i) (i.e., attributes in $\mathbf{Pa}^{(\mathrm{N})}(C_i)$ are associated with objects in $\mathcal{G}_{\mathrm{D}}$ that are related to o_i). For convenience, we refer to $\mathbf{Pa}^{(\mathrm{L})}(C_i)$ as *intrinsic* CPDs, $\mathbf{Pa}^{(\mathrm{N})}(C_i)$ as *relational* CPDs, and accordingly $\mathbf{Pa}(C_i)$ as *full* CPDs or directly CPDs for short. Parameter α_t is a scalar with $0 \le \alpha_t \le 1$ to capture the strength of the intrinsic dependency for objects of each type $t \in \mathbf{T}$. Moreover, CDNs introduce some parameters, called *dependency functions*, to directly capture the different strengths of contextual dependencies among linked objects such that the relational CPD $\mathbf{Pa}^{(\mathrm{N})}(C_i)$ is expressed as

$$P(C_i|\mathbf{Pa}^{(\mathrm{N})}(C_i)) = \sum_{o_{ik} \in \mathbf{Pa}(o_i)} \sigma_{i,ik} P(C_i|\mathbf{Pa}_{ik}^{(\mathrm{N})}(C_i)), \tag{2}$$

where $\mathbf{Pa}_{ik}^{(\mathrm{N})}(C_i)$ is the parent set of C_i in attributes of object $o_{ik} \in \mathbf{Pa}(o_i)$, and $\sigma_{i,ik}$ is the dependency function of o_i on o_{ik}, which is used to measure how much $\mathbf{Pa}_{ik}^{(\mathrm{N})}(C_i)$ affects the distribution of C_i. Here we assume that a function $\sigma_{i,ik}$ is called a *dependency function* of object o_i on object $o_{ik} \in O_{\mathrm{D}}$ if it satisfies: (1) $\sigma_{i,ik} \ge 0$; (2) $\sum_{o_{ik} \in \mathbf{Pa}(o_i)} \sigma_{i,ik} = 1$; (3) The function $\sigma_{i,ik}$ consists of at least two components: the mutual information $I(o_i; o_{ik})$ and the linkage kernel $K(o_i, o_{ik}) = f(\varphi_{i,ik})$. Several oft-used kernel functions (e.g., polynomial, exponential, or sigmoid functions) can be adopted to construct linkage kernels from the link features $f(\varphi_{i,ik})$ between o_i and o_{ik}[14], given a parameter β_t for each type

$t \in \mathbf{T}$ of objects. Here we use mutual information $I(o_i; o_{ik})$ to measure the statistically semantic correlation among o_i and o_{ik}. The higher the $I(o_i; o_{ik})$, the easier it is to estimate one object given the other, or vice versa. Thus we have the following definition:

Definition 1. For the relational schema $\mathcal{S}$, a CDN model $\mathcal{M}=(\mathcal{G}_{\mathrm{M}}, \mathbf{P}, \theta)$ defines:

1. a directed model graph $\mathcal{G}_{\mathrm{M}}$ in which each node corresponds to an attribute of objects with type $t \in \mathbf{T}$ and each edge represents the dependency among attributes,
2. a set of template CPDs $\mathbf{P}=\mathbf{P}^{(\mathrm{L})} \cup \mathbf{P}^{(\mathrm{N})}$ where $\mathbf{P}^{(\mathrm{L})}$ and $\mathbf{P}^{(\mathrm{N})}$ are the intrinsic and relational CPDs respectively, and
3. a parameter set $\theta=\left\{\alpha_t, \beta_t, \pi_t, a_{i,j}^{t,t'}\right\}_{t \in \mathbf{T}}$ that are used to specify dependency functions among linked objects in any link graph that is defined by the schema $\mathcal{S}$, where $\pi_t=\left\{p_i^t = P(c_i^t)\right\}$ are the priors, and $\left\{p(c_i^t|c_j^{t'}) \mid t' \in \mathbf{T}\right\}$ are the transition probabilities.

For a given link graph $\mathcal{G}_{\mathrm{D}}$, a CDN model uses the $\mathcal{G}_{\mathrm{M}}$ and $\mathcal{G}_{\mathrm{D}}$ to instantiate an inference graph $\mathcal{G}_{\mathrm{I}}=(\mathcal{V}_{\mathrm{I}}, \mathcal{E}_{\mathrm{I}})$ during inference so as to represent the probabilistic dependencies among all variables in a test set [9]. Figure 1 shows an example of $\mathcal{G}_{\mathrm{M}}$ and $\mathcal{G}_{\mathrm{I}}$. Given a CDN model $\mathcal{M}$, the full joint distribution over the unknown label variables in $\mathcal{G}_{\mathrm{D}}$ can be approximately expressed as follows:

$$\begin{aligned} P(\mathcal{G}_{\mathrm{D}}|\mathcal{M}) &= \prod_{t \in \mathbf{T}} \prod_{o_i \in \mathbf{I}(t)} P(C_i|\mathbf{Pa}(C_i)) \\ &= \prod_{t \in \mathbf{T}} \prod_{o_i \in \mathbf{I}(t)} \left[\alpha_t P(C_i|\mathbf{Pa}^{(\mathrm{L})}(C_i)) + (1-\alpha_t) P(C_i|\mathbf{Pa}_{ik}^{(\mathrm{N})}(C_i))\right] \\ &= \prod_{t \in \mathbf{T}} \prod_{o_i \in \mathbf{I}(t)} \left[\sum_{o_{ik} \in \{o_i\} \cup \mathbf{Pa}(o_i)} \widetilde{\sigma}_{i,ik} P(C_i|\mathbf{Pa}_{ik}^{*}(C_i))\right], \end{aligned} \tag{3}$$

where

$$\widetilde{\sigma}_{i,ik} = \begin{cases} \alpha_t, & o_{ik} = o_i, \\ (1-\alpha_t)\sigma_{i,ik}, & o_{ik} \in \mathbf{Pa}(o_i), \\ 0, & \text{otherwise.} \end{cases}$$

and

$$\mathbf{Pa}_{ik}^{*}(C_i) = \begin{cases} \mathbf{Pa}_{ik}^{(\mathrm{L})}(C_i), & o_{ik} = o_i, \\ \mathbf{Pa}_{ik}^{(\mathrm{N})}(C_i), & o_{ik} \in \mathbf{Pa}(o_i). \end{cases}$$

CDNs first approximate the full joint distribution for a collection of related objects with a set of CPDs. Then each CPD can be further modeled as a linear combination of an intrinsic CPD and a set of relational CPDs with the weights represented by dependency functions. This would facilitate ease of knowledge acquisition and domain modeling, and provide computational savings in the inference process.

2.3 Learning

Like DNs, both the structure and parameters of CDNs are determined through learning a set of CPDs. For a CDN model, the parameter-estimation task is to learn a set of

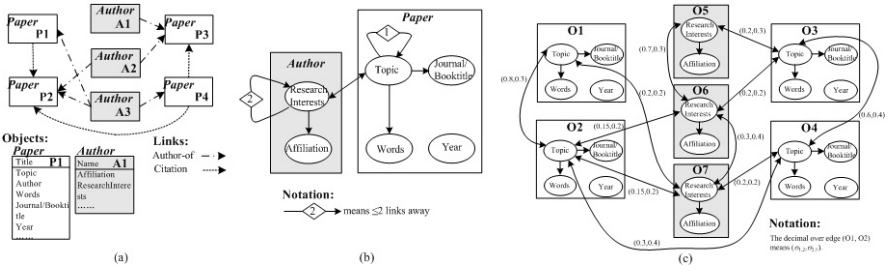

Fig. 1. (a) A link graph, (b) the model graph and (c) the inference graph

model parameters $\left\{\mathbf{P}_t^{(\mathrm{L})}, \mathbf{P}_t^{(\mathrm{N})}, \alpha_t, \beta_t, \pi_t\right\}_{t\in\mathbf{T}}$ from a training set $\mathcal{G}'_{\mathrm{D}}=(\mathcal{O}'_{\mathrm{D}}, \mathcal{L}'_{\mathrm{D}})$, where $\mathbf{P}^{(\mathrm{L})}$ and $\mathbf{P}^{(\mathrm{N})}$ are the intrinsic CPDs and relational CPDs respectively, α_t is the self-reliant factor, β_t is the parameter for linkage kernels, $\pi_t=\left\{p_i^t = P(c_i^t)\right\}$ are the priors. The learned parameters are then applied to a separate testing set $\mathcal{G}_{\mathrm{D}}$. Note that the transition probabilities $a_{i,j}^{t,t'}$ can be obtained by using the intrinsic and relational CPDs.

The CDN learning algorithm is in principle based on pseudo-likelihood techniques, which avoids the complexities of estimating a full joint distribution. With the assumption that the objects in $\mathcal{O}'_{\mathrm{D}}$ are independent, the prior p_i^t can be estimated by the relative frequencies of objects with the class label c_i^t in $\mathcal{O}'_{\mathrm{D}}$, and the intrinsic CPDs $\mathbf{P}_t^{(\mathrm{L})}$ can be estimated by any probabilistic classification or regression techniques (called *intrinsic models*) such as naïve Bayes (NB) (e.g., [4]), logistic regression (e.g., [6]), or probabilistic support vector machine (SVM) (e.g., [12]). For the relational CPDs $\mathbf{P}^{(\mathrm{N})}$, however, we cannot directly use the standard statistical learning methods since the labels of objects are correlated. By modeling the learning of $\mathbf{P}^{(\mathrm{N})}$ as a dynamic interacting process of multiple Markov chains, here we use the self-mapping transformation algorithm [13] to learn $\mathbf{P}^{(\mathrm{N})}$. That is, the graph $\mathcal{G}'_{\mathrm{D}}$ is partitioned into N' subgraphs, each of which contains an object o_i and its parents $\mathbf{Pa}(o_i)$. For each subgraph $\mathcal{G}'_{\mathrm{D}_i}=(\mathcal{O}'_{\mathrm{D}_i}, \mathcal{L}'_{\mathrm{D}_i})$, the relational CPD parameter can be learned by using the self-mapping transformation [13]. This process is repeated for all subgraphs until convergence. Lastly, for the parameters α_t and β_t, we can set the appropriate values empirically or by the cross-validation method. For example, α_t is set to be 0.7~0.8 for type=paper and 0.4~0.5 for type=author in the citation data.

2.4 Inference

During inference, a CDN model uses the $\mathcal{G}_{\mathrm{M}}$ and $\mathcal{G}_{\mathrm{D}}$ to instantiate an inference graph $\mathcal{G}_{\mathrm{I}}$. This process includes two operations: (1) Each object-attribute pair gets a separate, local copy of the appropriate CPD (including an intrinsic CPD and a relational CPD). (2) Calculate the dependency functions using the parameter set θ of the CDN model.

In general, the CDN inference graph can be fairly complex. Clearly, exact inference over this complex network is impractical, so we must resort to approximate inference. As in DNs and RDNs, we also use ordered Gibbs sampling for approximate inference over CDN inference graphs. First, a bootstrap step is used to assign an initial label for each unlabeled object using only the intrinsic models. That is, $p(C_i|\mathcal{M})$ can be initial-

ized as $p(C_i|\mathbf{Pa}^{(\mathrm{L})}(C_i))$ and an initial CDN inference graph $\mathcal{G}_{\mathrm{I}}^{(0)}$ can be constructed over the link graph $\mathcal{G}_{\mathrm{D}}$. Gibbs inference then proceeds iteratively to estimate the joint posterior distribution over the class variables of all unlabeled objects. For each variable, the *influence propagation* step is performed to return a refined posterior probability $p(C_i|\mathcal{M})$ given both other attributes of that object (i.e., $\mathbf{Pa}^{(\mathrm{L})}(C_i)$) and attributes of other related objects (i.e., $\mathbf{Pa}^{(\mathrm{N})}(C_i)$). This process is repeated for each unknown variable in the graph $\mathcal{G}_{\mathrm{I}}$. After a sufficient number of iterations, the values will be drawn from a stationary distribution [11]. This paper adopts a mixed iteration convergence criteria for Gibbs inference, including the convergence of the log-likelihood over all unobserved label variables, the consistency of the maximum a posterior (MAP) estimates among two consecutive iterations, and a predefined iteration upper bound.

3 Experiments

In this paper, we used two real-world datasets, i.e., Cora and WebKB, each of which can be viewed as a link graph. The whole Cora dataset consists of about 37,000 papers. In common with many other works (e.g., [6]), we use the subset (denote by Cora_0) with 4331 papers of Machine Learning and 11,873 citations. Moreover, several extended datasets, denoted by Cora_δ, are constructed by adding into Cora_0 different amounts of miscellaneous links that point from Cora_0 to papers with other topics. On the other hand, the WebKB dataset contains approximately 4100 pages from four computer science departments, with a five-valued type attribute (i.e., faculty, student, project, course and other), and 10,400 links between pages. Similarly, the base subset of pages with the four labels is denoted by WebKB_0. We construct several extended sets WebKB_δ by appending some links that point to other pages. With different δ values, the Cora_δ and WebKB_δ datasets may exhibit different link regularities. For simplicity, we set $\{0, 0.05, 0.10, 0.15, 0.18\}$ to values in for the two datasets.

In [14], we have shown that noisy links have high influence on link-based classification. Here our main goal is to demonstrate the robustness of our CDN model in collective classification on noisy datasets. We also use NBs and SVMs as the baseline intrinsic models, and use NICs and LLRs as the baseline link-based models. In addition, we re-construct all the link-based models respectively with NBs and SVMs as their intrinsic models. For convenience, they are denoted by $\mathrm{NIC}_{\mathrm{NB}}$, $\mathrm{NIC}_{\mathrm{SVM}}$, $\mathrm{LLR}_{\mathrm{NB}}$, $\mathrm{LLR}_{\mathrm{SVM}}$, $\mathrm{RDN}_{\mathrm{NB}}$, $\mathrm{RDN}_{\mathrm{SVM}}$, $\mathrm{CDN}_{\mathrm{NB}}$ and $\mathrm{CDN}_{\mathrm{SVM}}$ respectively. The experimental results are shown in figure 2.

On average, $\mathrm{CDN}_{\mathrm{NB}}$ outperformed $\mathrm{NIC}_{\mathrm{NB}}$, $\mathrm{LLR}_{\mathrm{NB}}$ and $\mathrm{RDN}_{\mathrm{NB}}$ respectively by about 6.15%, 6.08% and 5.62% on WebKB, and about 1.46%, 1.73% and 1.13% on Cora; $\mathrm{CDN}_{\mathrm{SVM}}$ outperformed $\mathrm{NIC}_{\mathrm{SVM}}$, $\mathrm{LLR}_{\mathrm{SVM}}$ and $\mathrm{RDN}_{\mathrm{SVM}}$ respectively by about 3.33%, 3.94% and 2.71% on WebKB, and about 2.32%, 2.51% and 1.57% on Cora. More importantly, the relative accuracies of CDNs do not decline along with increasing the parameter δ for the two datasets. In other words, CDNs can effectively exploit the miscellaneous links to improve the classification performance. Comparatively, although RDNs can use the selective relational classification algorithms (e.g., RPTs) to learn a set of CPDs, their performance is also affected by the noisy links in the inference phase. This enlightens us that the selectivity of link features should be directly encoded in the

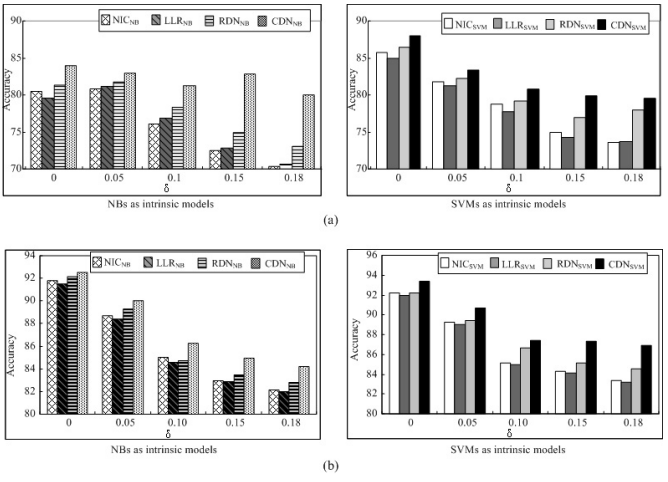

Fig. 2. Comparison of classification accuracies of all link-based models on (a) WebKB and (b) Cora

relational model itself such that the learned model can keep robust in different link data. We also noted one exception that the CDN models performed poorly on WebKB_5. This indicates that the accuracy improvements of CDNs might be not significant when the datasets have only fewer noisy links.

The differences in accuracy between the NIC, LLR, RDN and CDN models may indicate that the improvements are not significant. To investigate this possibility we performed two-tailed, paired t-tests to assess the significance of the results obtained from the four-validation tests. With a few exceptions, CDNs outperform NICs and LLRs at the 90% (averagely 97.7%) significance level on both WebKB and Cora, and outperform RDNs at the 80% (averagely 93.6%) significance level on the two datasets. These results support our conclusions that the classification performance of CDNs is significantly better than NICs, LLRs and RDNs.

Currently, we are experimenting with Web image classification tasks to explore more interesting applications of the RDN models. Our basic premise is that Web images which are co-contained in the same pages or contained in co-cited pages are likely to be related to the same topic. We thus can build a robust image classification model by using visual, textual and link information. On a sports Web image collection crawled from Yahoo!, the CDN model obtained about 14% improvements in the average classification accuracy over the SVM classifier that uses visual and textual features.

In summary, the experimental results are generally positive, but in some cases the improvements are not so significant. However, we can safely conclude that the CDN models show relatively high robustness in the link data with a few noisy links.

4 Conclusion

Many link data such as Web pages are often accompanied with a few noisy links. Such noisy links do not provide the predictive information for categorization. To capture

such complex regularities in link data, this paper proposes a robust model for collective classification, i.e., contextual dependency network (CDN) model. Experimental results showed that the CDN model can demonstrate high robustness in the noisy link datasets, and provide good prediction for the attributes of linked objects.

References

1. Jensen,D., Neville, J. and Gallagher, B.: Why collective inference improves relational classification. Proc. 10th ACM Int'l Conf. on Knowledge Discovery and Data Mining (2004) 593–598
2. Taskar, B., Abbeel, P., Koller, D.: Discriminative Probabilistic Models for Relational Classification. Proc. of Uncertainty on Artificial Intelligence, Edmonton, Canada (2002) 485–492
3. Yang, Y., Slattery, S. and Ghani, R.: A Study of Approaches to Hypertext Categorization. J. Intelligent Information system **2/3** (2002) 219–241
4. Chakrabarti, S., Dom, B. and Indyk, P.: Enhanced Hypertext Categorization Using Hyperlinks. Proc. of SIGMOD'98 (1998) 307–318
5. Neville, J., Jensen, D., Friedland, L. and Hay, M.: Learning relational probability trees. Proc. 9th ACM Int'l Conf. on Knowledge Discovery and Data Mining (2003) 625–630
6. Lu Q. and Getoor, L.: Link-based Classification. Proc. 12th Int'l Conf. on Machine Learning (2003) 496–503
7. Friedman, N., Koller, D. and Taskar, B.: Learning Probabilistic Models of Relational Structure. J. Machine Learning Research (2002) 679–707
8. Neville, J. and Jensen, D.: Collective Classification with Relational Dependency Networks. Proc. 2nd Multi-Relational Data Mining Workshop in KDD-2003 (2003)
9. Neville, J. and Jensen, D.: Dependency Networks for Relational Data. Proc. IEEE Int'l Conf. on Data Mining (2004)170–177
10. Richardson, M. and Domingos, P.: Markov Logic Networks. Machine Learning **26(1-2)** (2005) 107–136
11. Heckerman, D., Chickering, D., Meek, C., Rounthwaite, R. and Kadie, C.: Dependency Networks for Inference, Collaborative Filtering, and Data Visualization. J. Machine Learning Re-search 1 (2001) 49–75
12. Sollich, P.: Probabilistic methods for Support Vector Machines. Proc. Advances in Neural Information Processing Systems 12 (2000), MIT Press, 349–355
13. Zhong, S. and Ghosh, J.: A New Formulation of Coupled Hidden Markov Models. Tech. Report, Dept. of Electronic and Computer Engineering, U. of Texas at Austin, USA, (2001)
14. Tian, Y. H., Huang, T. J., Gao, W.: Latent linkage semantic kernels for collective classification of link data. J. Intelligent Information Systems, In Press, (2006)
15. Heckerman, D., Meek, C., Koller, D.: Probabilistic Models for Relational Data, Tech. Report, MSR-TR-2004-30, Microsoft Research (2004)
16. Uwents, W. and Blockeel, H.: Classifying relational data with neural networks. Proc. 15th Int'l Conf. on Inductive Logic Programming, Bonn, Germany (2005) 384–396
17. Neville, J., Jensen, D. and Gallagher, B.: Simple estimators for relational Bayesian classifiers. Proc. 3rd IEEE Int'l Conf. on Data Mining (2003) 609–612