Link-based Classification using Labeled and Unlabeled Data

Qing Lu QINGLU@CS.UMD.EDU
Dept. of Computer Science, University of Maryland, College Park, MD 20742

Lise Getoor GETOOR@CS.UMD.EDU
Dept. of Computer Science/UMIACS, University of Maryland, College Park, MD 20742

Abstract

There has been a surge of interest in learning using a mix of labeled and unlabeled data. General approaches include semi-supervised learning and tranductive inference. In this paper we look at some of the unique ways in which unlabeled data can improve performance when doing *link-based* classification, the classification of objects making use of both object descriptions and the links between objects.

1. Introduction

The problem of mining richly structured datasets, where the objects are linked in some way, is a new challenge for machine learning. In many cases this data can be described by a graph; links or edges among the objects may demonstrate certain patterns, which may be helpful for many machine learning tasks and are usually hard to capture using traditional statistical models. Objects may be labeled or unlabeled, and classification should exploit the unlabeled data and the link structure between both labeled and unlabeled objects.

Link mining is a newly emerging research area that is at the intersection of the work in link analysis (Jensen & Goldberg, 1998; Feldman, 2002), hypertext and web mining (Chakrabarti, 2002), relational learning and inductive logic programming (Dzeroski & Lavrac, 2001) and graph mining (Cook & Holder, 2000). Link mining is potentially useful in a wide range of application areas including bioinformatics, bibliographic citations, financial analysis, national security, and the Internet.

Recently there has been a great increase of interest in this area, fueled largely by interest in web and hypertext mining, but also by interest in mining social networks, bibliographic citation data, epidemiological data and other domains best described using a linked or graph structure. In this setting, unlabeled data provides information not only about the distribution of the objects, but also about the distribution of links.

Here we describe a framework for modeling link distributions, a link-based model introduced in Lu and Getoor (2003) that supports discriminatively trained models describing both links and the attributes of linked objects. In order to capture the joint distributions of the links, we use a logistic regression model for both the content and the links. A key challenge is structuring the model appropriately; simply throwing both links and content attributes into a 'flat' logistic regression model does not perform well.

In this paper, we examine the different ways in which unlabeled data can be used to improve classification performance in relational domains:

- Just as in the case of classical machine learning framework, in which there are no links among the data, unlabeled data can help us learn the distribution over object descriptions.
- Links among the unlabeled data (or test set) can provide information that can help with classification.
- Links between the labeled training data and unlabeled (test) data induce dependencies that should not be ignored.

The idea that each of these aspects are important is not new, nor is the idea that the use of an appropriate expectation-maximization algorithm can provide a unified framework for combining all these pieces of information. Our contribution is an empirical study of the effect of each of these sources of information in a novel probabilistic model, a logistic regression model based on both object features and properties of the link neighborhoods.

Proceedings of the ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data, Washington DC, 2003.

2. Related Work

For supervised learning, it is expensive and labor-intensive to construct a large, labeled set of examples. However in many domains it is relatively inexpensive to collect unlabeled examples. Recently several algorithms have been developed to learn a model from both labeled and unlabeled examples (Nigam et al., 2000; Zhang & Oles, 2000; Blum & Chawla, 2001). Successful applications in a number of areas, especially text classification, have been reported. Interestingly, a number of results show that while careful use of unlabeled data is helpful, it is not always the case that more unlabeled data improves performance (Nigam, 2001).

Blum and Mitchell (1998) proposes a co-training algorithm to make use of unlabeled data to boost the performance of a learning algorithm. They assume that the data can be described by two separate feature sets which are not completely correlated, and each of which is predictive enough for a weak predictor respectively. The co-training procedure works to augment the labeled sample with data from unlabeled data using these two weak predictors. Their experiments show positive results on the use of unlabeled examples to improve the performance of the learned model. In (Mitchell, 1999), the author states that many natural learning problems fit the problem class where the features describing the examples are redundantly sufficient for classifying the examples. In this case, the unlabeled data can significantly improve learning accuracy. There are many problems falling into this category: web page classification; semantic classification of noun phrases, learning to select word sense and object recognition in multimedia data.

Nigam et al. (2000) introduce an EM algorithm for learning a Naive Bayes classifier from labeled and unlabeled examples. The algorithm first trains a classifier based on labeled documents and then probabilistically classifies the unlabeled documents. Then both labeled and unlabeled documents participate in the learning procedure. This process repeats until it converges. The ideas of using co-training and EM algorithms for learning from labeled and unlabeled data are fully investigated in (Ghani, 2001)

Joachims (1999) proposes a transductive support vector machine (TSVM) for text classification. A TSVM takes into account a particular test set and tries to optimize the classification accuracy for that particular test set. This also is an important means of using labeled and unlabeled examples for learning.

In other recent work on link mining (Neville & Jensen, 2000; Getoor et al., 2002; Taskar et al., 2002), models are learned from fully labeled training examples and evaluated on a disjoint test set. In some cases, the separation occurs naturally, for example in the WebKB dataset (Craven et al., 1998). This dataset describes the web pages at four different universities, and one can naturally split the data into a collection of training schools and a test school, and there are no links from the test school web pages to the training school pages. But in other cases, the datasets are either manipulated to extract disconnected components, or the links between the training and test sets are simply ignored. One major disadvantage of this approach is that it discards links between labeled and unlabeled data which may be very helpful for making predictions or may artificially create a skewed training and test set.

Chakrabarti et al. (1998) propose an iterative relaxation labeling algorithm to classify a patent database and a small web collection. They examine using text, neighboring text and neighbor class labels for classification in a rather realistic setting wherein some portion of the neighbor class labels are known. In the start of their iteration, a bootstrap mechanism is introduced to classify unlabeled documents. After that, classes from labeled and unlabeled documents participate in the relaxation labeling iteration. They showed that naively incorporating words from neighboring pages reduces performance, while incorporating category information, such has hierarchical category prefixes, improves performance.

Oh et al. (2000) also suggest an incremental categorization method, where the classified documents can take part in the categorization of other documents in the neighborhood. In contrast to the approach used in Chakrabarti et al. (1998), they do not introduce a bootstrap stage to classify all unlabeled documents. Instead they incrementally classify documents and take into account the classes of unlabeled documents as they become available in the categorization process. They report similar results on a collection of encyclopedia articles: merely incorporating words from neighboring documents was not helpful, while making use of the predicted class of neighboring documents was helpful.

Popescul et al. (2002) study the use of inductive logic programming(ILP) to combine text and link features for classification. In contrast to Chakrabarti et al. (1998) and Oh et al. (2000) where class labels are used as features, they incorporate the unique document IDs of neighborhood as features. Their results also demonstrate that the combination of text and link features often improves performance.

These results indicate that simply assuming that link documents are on the same topic, and incorporating the features of linked neighbors is not generally effective. One approach is to identify certain types of hypertext regularities such as encyclopedic regularity (linked objects typically have the same class) and co-citation regularity (linked objects do not share the same class, but objects that are cited by the same object tend to have the same class). Yang et al. (2002) gives an in-depth investigation of the validity of these regularities across several datasets and using a range of classifiers. They found that the usefulness of the regularities varied, depending on both the dataset and the classifier being use.

Here, we propose a probabilistic method that can learn a variety of different regularities among the categories of linked objects using labeled and unlabeled examples. Our method differs from the previous work in several ways. First, instead of assuming a Naive Bayes model (Chakrabarti et al., 1998) for the class labels in the neighborhood, we adopt a logistic regression model to capture the conditional probability of the class labels given the object attributes and link descriptions. In this way our method is able to learn a variety of different regularities and is not limited to a self-reinforcing encyclopedic regularity. We propose an algorithm to make predictions using both labeled and unlabeled data. Our approach makes use of the description of unlabeled data and all of the links between unlabeled and label data in an iterative algorithm for finding the collective labeling which maximizes the posterior probability for the class labels of all of the unlabeled data given the observed labeled data and links.

3. Link-based models

In this section, we review the link-based models described in Lu and Getoor (2003). We define a general notion of a link-based model that can be used for object classification based on the distribution of links and based on attributes of linked objects.

3.1. Definitions

The generic link-based data we consider is essentially a directed graph, in which the nodes are objects and edges are links between objects.

$\mathcal{O}$ - The collection of objects, $\mathcal{O} = \{X_1, \ldots, X_N\}$ where X_i is an object, or node in the graph. $\mathcal{O}$ is the set of nodes in the graph.

$\mathcal{L}$ - The collections of links between objects. $L_{i\rightarrow j}$ is a link between object X_i and object X_j. $\mathcal{L}$ is the set of edges in the graph.

$\mathcal{G}(\mathcal{O}, \mathcal{L})$ - The directed graph defined over $\mathcal{O}$ by $\mathcal{L}$.

Our model will support classification of objects based both on features of the object *and* on properties of its links. The object classifications are a finite set of categories $\{c_1, \ldots, c_k\}$ where $c(X)$ is the category c of object X. We will consider the neighbors of an object X_i via incoming, outgoing and co-citation links:

$I(X_i)$ - the set of incoming neighbors of object X_i, $\{X_j \mid L_{j\rightarrow i} \in \mathcal{L}\}$.

$O(X_i)$ - the set of outgoing neighbors of object X_i, $\{X_j \mid L_{i\rightarrow j} \in \mathcal{L}\}$.

$Co(X_i)$ - The set of objects co-cited with object X, $\{X_j \mid X_j \neq X_i$ and there is a third object X_k that links to both X_i and $X_k\}$.

3.2. Object features

The attributes of an object provide a basic description of the object. Traditional classification algorithms are based on object attributes. We use the notation $OA(X)$ for the attributes of object X. As an example, in the scientific literature domain, the object features might consist of a variety of text information such as title, abstract, authorship and content. In the domains we examined, the objects are text documents the object features we use are word occurrences.

3.3. Link features

To capture the link patterns, we introduce the notion of link features as a way of capturing the salient characteristics of the objects' links. We examine a variety of simple mechanisms for doing this. All are based on statistics computed from the linked objects rather than the *identity* of the linked objects. Describing only the limited collection of statistics computed from the links can be significantly more compact than storing the link incidence matrix. In addition, these models can accommodate the introduction of new objects, and thus are applicable in a wider range of situations.

We examine several ways of constructing link features. All are constructed from the collection of the categories of the linked objects. We use $LD(X)$ to denote the link description.

The simplest statistic to compute is a single feature, the mode, from each set of linked objects from the in-links, out-links and co-citation links. We call this the **mode-link** model.

We can use the frequency of the categories of the linked objects; we refer to this as the **count-link** model. In this case, while we have lost the information about the individual entity to which the object is connected, we maintain the frequencies of the different categories.

A middle ground between these two is a simple binary feature vector; for each category, if a link to an object of that category occurs at least once, the corresponding feature is 1; the feature is 0 if there are no links to this category. In this case, we use the term **binary-link** model. Figure 1 shows examples of the three types of link features computed for an object for each category of links (in-links, out-links and co-citation links).

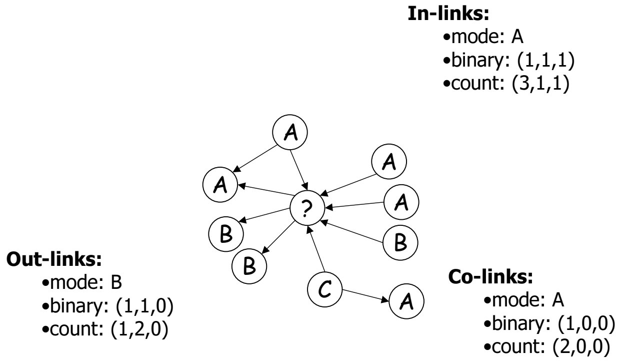

Figure 1. Assuming there are three possible categories for objects, A, B and C, the figure shows examples of the mode, binary and count link features constructed for the object labeled with ?.

4. Predictive model for object classification

Clearly we may make use of the object and link features in a variety of models such as Naive Bayes classifiers, SVMs and logistic regression models. For the domains that we have examined, logistic regression models have outperformed Naive Bayes models, so these are the models we have focused on.

For our predictive model, we used a regularized logistic regression model. Given a training set of labeled data (x_i, c_i), where $i = 1, 2, \ldots, n$ and $c_i \in \{-1, +1\}$, to compute the conditional probability $P(c \mid w, x)$ is to find the optimal w for the discriminative function, which is equivalent to the following regularized logistic regression formulation (Zhang & Oles, 2001):

$$\hat{w} = \text{arginf}_w \frac{1}{n} \sum_{i=1}^{n} \ln(1 + exp(-w^T x_i c_i)) + \lambda w^2$$

where we use a zero-mean independent Gaussian prior for the parameter w: $P(w) = exp(\lambda w^2)$.

The simplest model is a flat model, which uses a single logistic regression model over both the object attributes and link features. We found that this model did not perform well, and instead we found that a structured logistic regression model, which uses separate logistic regression models (with different regularization parameters) for the object features and the link features outperformed the flat model. Now the MAP estimation for categorization becomes

$$\hat{C}(X) \quad = \quad \text{argmax}_{c \in C} \frac{P(c \mid OA(X)) P(c \mid LD(X))}{P(c)}$$

where OA(X) are the object features and LD(X) are the link features and we make the (probably incorrect) assumption that they are independent. $P(c \mid OA(X))$ and $P(c \mid LD(X))$ are defined as

$$P(c \mid OA(X)) = \frac{1}{exp(-w_o^T OA(X) c) + 1}$$

$$P(c \mid LD(X)) = \frac{1}{exp(-w_l^T LD(X) c) + 1}$$

where w_o and w_l are the parameters for the regularized logistic regression models for $P(c \mid OA(X))$ and $P(c \mid LD(X))$ respectively.

5. Link-based classification using Labeled and Unlabeled Data

Given data D consisting of labeled data D^l and unlabeled data D^u, we define a posterior probability over D^u as

$$P(c(X) : X \in D^u \mid D) = \prod_{X \in D^u} P(c(X) \mid OA(X), LD(X))$$

We use an EM-like iterative algorithm to make use of both labeled data $D^l = \{(x_i, c(x_i) : i = 1, .., n\}$ and unlabeled data $D^u = \{(x_j^*, c(x_j^*) : j = 1, ..., m\}$ to learn our model. Initally a structured logistic regression model is built using labeled data D^l. First, we categorize data in D^u

$$c(x_j^*) \quad = \quad \text{argmax}_{c \in C} \frac{P(c \mid OA(x_j^*)) P(c \mid LD(x_j^*))}{P(c)}$$

where $j = 1, ..., m$. Next this categorized D^u and labeled data D^l are used to build a new model.

Step 1: (Initialization) Build an initial structured logistic regression classifier using content and link features using only the labeled training data.

Step 2: (Iteration) Loop while the posterior probability over the unlabeled test data increases:

1. Classify unlabeled data using the current model.
2. Recompute the link features of each object. Re-estimate the parameters of the logistic regression models.

In our above iterative algorithm, after we categorize the unlabeled data, the link descriptions for all labeled and unlabeled data will change due to the links among labeled and unlabeled data. The first step is to recompute the link descriptions for all data based on the results from the current estimates and the link graph over labeled and unlabeled data.

6. Results

We evaluated our link-based classification algorithm on the Cora dataset (McCallum et al., 2000) and a dataset that we constructed from CiteSeer entries (Giles et al., 1998). In both domains, document frequency (DF) is used to prune the word dictionary. Words with DF values less than 10 are discarded.

The Cora dataset contains 4187 machine learning papers, each categorized into one of seven possible topics. We consider only the 3181 papers that are cited or cite other papers. There are 6185 citations in the dataset. After stemming and removing stop words and rare words, the dictionary contains 1400 words.

The CiteSeer dataset has approximately 3600 papers from six categories: Agents, Artificial Intelligence, Database, Human Computer Interaction, Machine Learning and Information Retrieval. There are 7522 citations in the dataset. After stemming and removing stop words and rare words, the dictionary for CiteSeer contains 3000 words.

6.1. Prediction with links between training and test sets

We began by investigating the issue of exploiting the links between test and training data for predictions.

In other work Neville and Jensen (2000), Getoor et al. (2002) and Taskar et al. (2002) using link distributions for categorization, the experimental data set are split into training set and test set, and any links across training and test sets are ignored.

In reality, in domains such as web and scientific literature, document collections are changing dynamically. There are new papers published, and new web sites created. New objects and edges are being added to the existing graph. A more realistic evaluation, such as that done in Chakrabarti et al. (1998), exploits the links between test and training. In an effort to understand this phenomena more fully, we examined the effect of ignoring links between training and test sets. Here we compared a method which discards all link information across training set and test set, which is denoted as ”Test Links Only”, with a more realistic method which keeps all the links between test and training sets. which is denoted as ”Complete Links”. With ”Test Links Only”, in our iterative classification process, the link descriptions of test data are constructed based only on the link graph over test data, while with ”Complete Links” link descriptions of test data are formulated over the link graph using both training and test data. For each experiment on Cora and CiteSeer, each domain is split into three data sets with equal size and a three-fold cross validation is done. We take one split as a test set, and the remaining two splits are used to train our model: one for training and the other is used as a validation set for setting the regularization parameter for the logistic regression models.

In our experiments, we compared a baseline model (Content-Only) with our link-based models (Mode-Link, Binary-Link, Count-Link). We compared the models:

- **Content-Only**: Uses only object attributes.
- **Mode-Link**: Combines a logistic regression model over the object attributes with a separate logistic regression model over the mode of the in-links, out-links and co-citations.
- **Binary-Link**: Combines a logistic regression model over the object attributes with a separate logistic regression model over the binary link statistics for the in-links, out-links and co-citations.
- **Count-Link**: Combines a logistic regression model over the object attributes with a separate logistic regression model over the counts link statistics for the in-links, out-links and co-citations.

Table 1 and Table 2 show the summary of our experimental results on both Cora and CiteSeer domains. These results demonstrate that the complete link structure is informative and can be used to improve overall performance. We did a paired t-test on F1 measure. For Binary-Link and Count-Link models, using ”Complete links” performs better than using ”Test Links Only” with significance level above 95% in both Cora and CiteSeer. Our models (Binary-Link and Count-Link) outperform both the base model (Content-Only) and the simplest link-based model (Mode-Link). For more details on our link-based model, we refer the reader to Lu and Getoor (2003).

	Test Links Only				Complete Links		
	Content-Only	Mode-Link	Binary-Link	Count-Link	Mode-Link	Binary-Link	Count-Link
Accuracy	0.678	0.708	0.707	0.709	0.717	0.756	**0.758**
Precision	0.649	0.673	0.673	0.675	0.717	0.761	**0.759**
Recall	0.631	0.688	0.687	0.69	0.679	0.721	**0.725**
F1 Measure	0.646	0.68	0.68	0.682	0.697	0.74	**0.741**

Table 1. Results using ”Test Links Only” and ”Complete Links” on Cora.

	Test Links Only				Complete Links		
	Content-Only	Mode-Link	Binary-Link	Count-Link	Mode-Link	Binary-Link	Count-Link
Accuracy	0.612	0.636	0.635	0.639	0.661	0.666	**0.678**
Precision	0.554	0.572	0.569	0.573	0.594	0.603	**0.601**
Recall	0.558	0.579	0.576	0.581	0.595	0.603	**0.609**
F1 Measure	0.558	0.575	0.573	0.577	0.594	0.603	**0.605**

Table 2. Results using ”Test Links Only” and ”Complete Links” on CiteSeer.

6.2. Link-based classification using labeled and unlabeled data

In previous section we experimented with labeled and unlabeled data for predictions. Next we explored the learning with labeled and unlabeled data using the iterative algorithm proposed in Section 5. To better understand the effects of unlabeled data, we compared the performance of our algorithm with varying amounts of labeled and unlabeled data.

For each domain (Cora or CiteSeer), we randomly choose 20% of the data as test data. We compared the performance of the algorithms when different percentages (20%, 40%, 60%, 80%) of the remaining data is labeled. We compared the accuracy when only the labeled data is used for training (labeled only) with the case where both labeled and the remaining unlabeled data is used for training (labeled and unlabeled). We compared the models:

- **Content-Only**: Uses only object attributes.
- **Labeled-Only**: the binary-link model is learned on labeled data only. The only unlabeled data used is the test set.
- **Labeled and Unlabeled**: the binary-link model is learned on both labeled and all of the unlabeled data.

Figure 2 shows the results averaged over 5 different runs. The algorithm which makes use of all of the unlabeled data gives better performance than the model which uses only the labeled data. Interestingly for the Cora dataset, more unlabeled data is not always better. The average improvement in F1 measure is 6% on Cora and 4.9% on CiteSeer. We did a paired t-test. For both datasets, the algorithm which uses both labeled and unlabeled data outperforms the algorithm which uses labeled-only data; even with 80% of the data labeled and only 20% of the data unlabeled, the improvement in error on the test set using unlabeled data is statistically significant at the 95% confidence level for both Cora and Citeseer.

7. Conclusions

In link-based classification, unlabeled data provides useful information in three important ways: first, it gives us additional information about the distribution of object attribute values; second, links among unlabeled data in the test set provide useful information about classification and third, links between labeled (training) data and unlabeled (test) data also provide useful information that should not be ignored. When the classification problem is properly modeled, and we don’t distort the data by removing links between the test and training and inference is used for collective classification, we are able to make use of all of the information that unlabeled data provides.

Acknowledgements

This study was supported in part by the Advanced Research and Development Activity (ARDA) under Award Number NMA401-02-1-2018. The views, opinions, and findings contained in this report are those of the author(s) and should not be construed as an official Department of Defense position, policy, or decision unless so designated by other official documentation.

References

Blum, A., & Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. *Proc. 18th International Conf. on Machine Learning* (pp. 19–26). Morgan Kaufmann, San Francisco, CA.

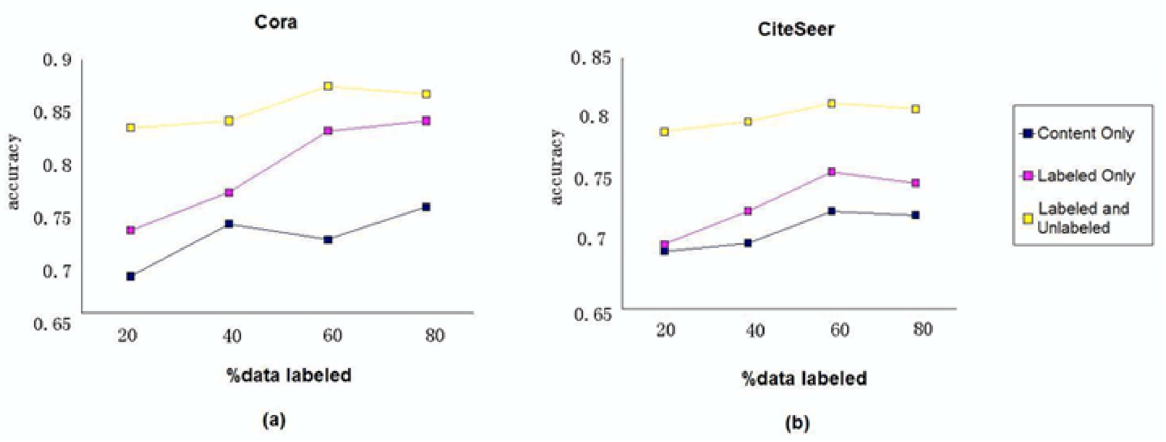

Figure 2. (a) Results varying the amount of labeled and unlabeld data used for training on Cora (b) and on CiteSeer. The results are averages of 5 runs.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*.

Chakrabarti, S. (2002). *Mining the web*. Morgan Kaufman.

Chakrabarti, S., Dom, B., & Indyk, P. (1998). Enhanced hypertext categorization using hyperlinks. *Proc of SIGMOD-98*.

Cook, D., & Holder, L. (2000). Graph-based data mining. *IEEE Intelligent Systems*, *15*, 32–41.

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. *Proc. of AAAI-98*.

Dzeroski, S., & Lavrac, N. (Eds.). (2001). *Relational data mining*. Berlin: Kluwer.

Feldman, R. (2002). Link analysis: Current state of the art. *Tutorial at the KDD-02*.

Getoor, L., Friedman, N., Koller, D., & Taskar, B. (2002). Learning probabilistic models with link uncertainty. *Journal of Machine Learning Research*.

Ghani, R. (2001). Combining labeled and unlabeled data for text classification with a large number of categories. *Proceedings of the IEEE International Conference on Data Mining* (pp. 597–598). San Jose, US: IEEE Computer Society, Los Alamitos, US.

Giles, C. L., Bollacker, K., & Lawrence, S. (1998). CiteSeer: An automatic citation indexing system. *ACM Digital Libraries 98*.

Jensen, D., & Goldberg, H. (1998). *AAAI fall symposium on AI and link analysis*. AAAI Press.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. *Proceedings of ICML-99, 16th International Conference on Machine Learning* (pp. 200–209). Bled, SL: Morgan Kaufmann Publishers, San Francisco, US.

Lu, Q., & Getoor, L. (2003). Link-based classification. *to appear at Proceedings of the Twentieth International Conference on Machine Learning*. Washington DC, US.

McCallum, A., Nigam, K., Rennie, J., & Seymore, K. (2000). Automating the construction of internet portals with machine learning. *Information Retrieval*, *3*, 127–163.

Mitchell, T. (1999). The role of unlabeled data in supervised learning. *Proceedings of the Sixth International Colloquium on Cognitive Science*. San Sebastian, Spain.

Neville, J., & Jensen, D. (2000). Iterative classification in relational data. *Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data*. AAAI Press.

Nigam, K. (2001). *Using unlabeled data to improve text classification*. Doctoral dissertation, Carnegie Mellon University.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. M. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, *39*, 103–134.

Oh, H.-J., Myaeng, S. H., & Lee, M.-H. (2000). A practical hypertext categorization method using links and incrementally available class information. *Proc. of SIGIR-00*.

Popescul, A., Ungar, L., Lawrence, S., & Pennock, D. (2002). Towards structural logistic regression: Combing relational and statistical learning. *KDD Workshop on Multi-Relational Data Mining*.

Taskar, B., Abbeel, P., & Koller, D. (2002). Discriminative probabilistic models for relational data. *Proc. of UAI-02* (pp. 485–492). Edmonton, Canada.

Yang, Y., Slattery, S., & Ghani, R. (2002). A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, *18*, 219–241.

Zhang, T., & Oles, F. J. (2000). A probability analysis on the value of unlabeled data for classification problems. *Proc. 17th International Conf. on Machine Learning* (pp. 1191–1198). Morgan Kaufmann, San Francisco, CA.

Zhang, T., & Oles, F. J. (2001). Text categorization based on regularized linear classification methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5–31.