7

Link-based Classification

Lise Getoor

Summary. A key challenge for machine learning is the problem of mining richly structured data sets, where the objects are linked in some way due to either an explicit or implicit relationship that exists between the objects. Links among the objects demonstrate certain patterns, which can be helpful for many machine learning tasks and are usually hard to capture with traditional statistical models. Recently there has been a surge of interest in this area, fuelled 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 chapter we propose a framework for modeling link distributions, a link-based model that supports discriminative models describing both the link distributions and the attributes of linked objects. We use a structured logistic regression model, capturing both content and links. We systematically evaluate several variants of our link-based model on a range of data sets including both web and citation collections. In all cases, the use of the link distribution improves classification performance.

7.1 Introduction

Traditional data mining tasks such as association rule mining, market basket analysis and cluster analysis commonly attempt to find patterns in a data set characterized by a collection of independent instances of a single relation. This is consistent with the classical statistical inference problem of trying to identify a model given a random sample from a common underlying distribution.

A key challenge for machine learning is to tackle the problem of mining more richly structured data sets, for example multi-relational data sets in which there are record linkages. In this case, the instances in the data set are linked in some way, either by an explicit link, such as a URL, or a constructed link, such as join between tables stored in a database. Naively applying traditional statistical inference procedures, which assume that instances are independent, can lead to inappropriate conclusions [15]. Care must be taken that potential correlations due to links are handled appropriately. Clearly, this is

information that should be exploited to improve the predictive accuracy of the learned models.

Link mining is a newly emerging research area that is at the intersection of the work in link analysis [10, 16], hypertext and web mining [3], relational learning and inductive logic programming [9] and graph mining [5]. Link mining is potentially useful in a wide range of application areas including bioinformatics, bibliographic citations, financial analysis, national security, and the Internet. Link mining includes tasks such as predicting the strength of links, predicting the existence of links, and clustering objects based on similar link patterns.

The link mining task that we focus on in this chapter is *link-based classification*. Link-based classification is the problem of labeling, or classifying, objects in a graph, based in part on properties of the objects, and based in part on the properties of neighboring objects. Examples of link-based classification include web-page classification based both on content of the web page and also on the categories of linked web pages, and document classification based both on the content of a document and also the properties of cited, citing and co-cited documents.

Three elements fundamental to link-based classification are:

- **link-based feature construction** – how do we represent and make use of properties of the neighborhood of an object to help with prediction?
- **collective classification** – the classifications of linked objects are usually correlated, in other words the classification of an object depends on the classification of neighboring objects. This means we cannot optimize each classification independently, rather we must find a globally optimal classification.
- **use of labeled and unlabeled data** – The use of labeled and unlabeled data is especially important to link-based classification. A principled approach to collective classification easily supports the use of labeled and unlabeled data.

In this chapter we examine each of these elements and propose a statistical framework for modeling link distributions and study its properties in detail. Rather than an ad hoc collection of methods, the proposed framework extends classical statistical approaches to more complex and richly structured domains than commonly studied.

The framework we propose stems from our earlier work on link uncertainty in probabilistic relational models [12]. However in this work, we *do not* construct explicit models for link existence. Instead we model link distributions, which describe the neighborhood of links around an object, and can capture the correlations among links. With these link distributions, we propose algorithms for link-based classification. 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 as well as a structured logistic regression model that combines one logistic regression model built over content with a separate logistic regression model built over links.

Having learned a model, the next challenge is classification using the learned model. A learned link-based model specifies a distribution over link and content attributes and, unlike traditional statistical models, these attributes may be correlated. Intuitively, for linked objects, updating the category of one object can influence our inference about the categories of its linked neighbors. This requires a more complex classification algorithm. Iterative classification and inference algorithms have been proposed for hypertext categorization [4, 28] and for relational learning [17, 25, 31, 32]. Here, we also use an iterative classification algorithm. One novel aspect is that unlike approaches that make assumptions about the influence of the neighbor's categories (such as that linked objects have similar categories), we explicitly learn how the link distribution affects the category. We also examine a range of ordering strategies for the inference and evaluate their impact on overall classification accuracy.

7.2 Background

There has been a growing interest in learning from structured data. By structured data, we simply mean data best described by a graph where the nodes in the graph are objects and the edges/hyper-edges in the graph are links or relations between objects. Tasks include hypertext classification, segmentation, information extraction, searching and information retrieval, discovery of authorities and link discovery. Domains include the world-wide web, bibliographic citations, criminology, bio-informatics to name just a few. Learning tasks range from predictive tasks, such as classification, to descriptive tasks, such as the discovery of frequently occurring sub-patterns.

Here, we describe some of the most closely related work to ours, however because of the surge of interest in recent years, and the wide range of venues where research is reported (including the International World Wide Web Conference (WWW), the Conference on Neural Information Processing (NIPS), the International Conference on Machine Learning (ICML), the International ACM conference on Information Retrieval (SIGIR), the International Conference of Management of Data (SIGMOD) and the International Conference on Very Large Databases (VLDB)), our list is sure to be incomplete.

Probably the most famous example of exploiting link structure is the use of links to improve information retrieval results. Both the well-known page rank [29] and hubs and authority scores [19] are based on the link-structure of the web. These algorithms work using in-links and out-links of the web pages to evaluated the importance or relevance of a web-page. Other work, such Dean and Henzinger [8] propose an algorithm based on co-citation to find

related web pages. Our work is not directly related to this class of link-based algorithms.

One line of work more closely related to link-based classification is the work on hypertext and web page classification. This work has its roots in the information retrieval community. A hypertext collection has a rich structure beyond that of a collection of text documents. In addition to words, hypertext has both incoming and outgoing links. Traditional bag-of-words models discard this rich structure of hypertext and do not make full use of the link structure of hypertext.

Beyond making use of links, another important aspect of link-based classification is the use of unlabeled data. In 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 [1, 27, 34]. 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 [26].

Blum and Mitchell [2] propose 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. 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 [24], 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.* [27] 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 [13].

Joachims *et al.* [18] 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 [12, 25, 31], 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 data set [6]. This data set 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 data sets 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 skewed training and test sets.

Chakrabarti *et al.* [4] proposed 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.* [28] 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.*, 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.* [30] study the use of inductive logic programming (ILP) to combine text and link features for classification. In contrast to Chakrabarti *et al.* and Oh *et al.*, where class labels are used as features, they incorporate the unique document IDs of the 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.* [33] compare several well-known categorization learning algorithms: naive Bayes [22], kNN [7], and FOIL on three data sets. They find that adding words from linked neighbors is sometimes helpful for categorization and sometimes

harmful. They define five hypertext regularities for hypertext categorization. Their experiments indicate that application of this knowledge to classifier design is crucial for real-world categorization. However, the issue of discovering the regularity is still an open issue.

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 [4] 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 examine a number of different types of links and methods for representing the link neighborhood of an object. 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 labeled 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.

7.3 Link-based Models

Here we propose a general notion of a link-based model that supports rich probabilistic models based on the distribution of links and based on attributes of linked objects.

7.3.1 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. $L_{i \to j}$ 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 supports 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 the following relations:

- $In(X_i)$ – the set of incoming neighbors of object X_i, $\{X_j \mid L_{j \to i} \in \mathcal{L}\}$.
- $Out(X_i)$ – the set of outgoing neighbors of object X_i, $\{X_j \mid L_{i \to j} \in \mathcal{L}\}$.

- $Co\text{-}In(X_i)$ – The set of objects co-cited with object X_i, $\{X_j \mid X_j \neq X_i$ and there is a third object X_k that links to both X_i and $X_j\}$. We can think of these as the co-citation in-links (Co-In), because there is an object X_k with in-links to both X_i and X_j.
- $Co\text{-}Out(X_i)$ – The set of objects co-cited by object X_i, $\{X_j \mid X_j \neq X_i$ and there is a third object X_k to which both X_i and X_j link$\}$. We can think of these as the co-citation out-links (Co-Out), because both X_i and X_j have out links to some object X_k.

7.3.2 Object Features

The attributes of an object provide a basic description of the object. Traditional classification algorithms are based on object attributes. In a linked-based approach, it may also make sense to use attributes of *linked* objects. Furthermore, if the links themselves have attributes, these may also be used.[1] However, in this paper, we simply use object attributes, and 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 and the object features we use are word occurrences.

7.3.3 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 both in and out 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

[1] Essentially this is a propositionalization [11, 20] of the aspects of the neighborhood of an object in the graph. This is a technique that has been proposed in the inductive logic programming community and is applicable here.

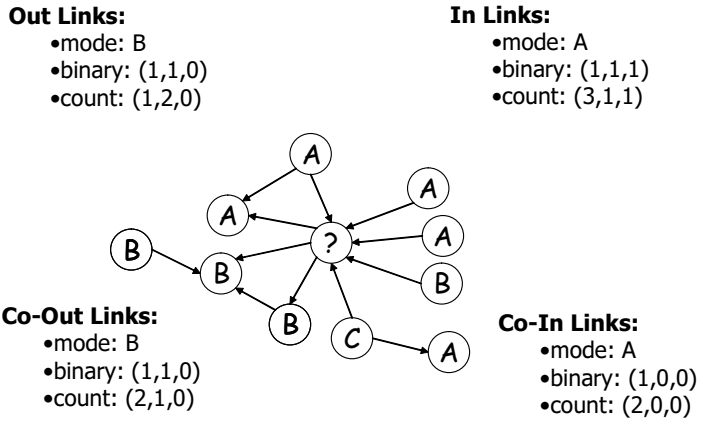


Fig. 7.1. Assuming there are three possible categories for objects, A, B and C, the figure shows examples of the mode, binary and count link features constructed for the object labeled with ?.

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 7.1 shows examples of the three types of link features computed for an object for each category of links (In links, Out links, Co-In links and Co-Out links).

7.4 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 [35]:

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

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

$$\hat{C}(X) = \text{argmax}_{c \in C} \frac{P(c \mid OA(X)) \prod_{t \in \{In, Out, Co\text{-}In, Co\text{-}Out\}} P(c \mid LD_t(X))}{P(c)}$$

where OA(X) are the object features and $LD_t(X)$ are the link features for each of the different types of links t and we make the (probably incorrect) assumption that they are independent. $P(c \mid OA(X))$ and $P(c \mid LD_t(X))$ are defined as

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

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

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

7.5 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

$$P(c(X) : X \in D^u \mid D) = \prod_{X \in D^u} P(c(X) \mid OA(X), LD_{In}(X), LD_{Out}(X), LD_{Co\text{-}In}(X), LD_{Co\text{-}Out}(X))$$

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

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

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 between 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.

In the iterative step there are many possible orderings for objects. One approach is based simply on the number of links; Oh *et al.* [28] report no significant improvement using this method. Neville and Jensen [25] propose an iterative classification algorithm where the ordering is based on the inference posterior probability of the categories. They report an improvement in classification accuracy. We explore several alternate orderings based on the estimated link statistics. We propose a range of link-based adaptive strategies which we call *Link Diversity*. Link diversity measures the number of different categories to which an object is linked. The idea is that, in some domains at least, we may be more confident of categorizations of objects with low link – diversity in essence, the object's neighbors are all in agreement. So we may wish to make these assignments first, and then move on to the rest of the pages. In our experiments, we evaluate the effectiveness of different ordering schemes based on link diversity.

7.6 Results

We evaluated our link-based classification algorithm on two variants of the Cora data set [23], a data set that we constructed from CiteSeer entries [14] and WebKB [6].

The first Cora data set, CoraI, 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 data set. After stemming and removing stop words and rare words, the dictionary contains 1400 words.

The second Cora data set, CoraII,[2] contains 30,000 papers, each categorized into one of ten possible topics: information retrieval, databases, artificial intelligence, encryption and compression, operating systems, networking, hardware and architecture, data structure algorithms and theory, programming and human–computer interaction. We consider only the 3352 documents that are cited or cite other papers. There are 8594 citations in the data set.

[2] www.cs.umass.edu/~mccallum/code-data.html

After stemming and removing stop words and rare words, the dictionary contains 3174 words.

The CiteSeer data set has 3312 papers from six categories: Agents, Artificial Intelligence, Database, Human Computer Interaction, Machine Learning and Information Retrieval. There are 7522 citations in the data set. After stemming and removing stop words and rare words, the dictionary for CiteSeer contains 3703 words.

The WebKB data set contains web pages from four computer science departments, categorized into topics such as faculty, student, project, course and a catch-all category, other. In our experiments we discard pages in the "other" category, which generates a data set with 700 pages. After stemming and removing stop words, the dictionary contains 2338 words. For WebKB, we train on three schools, plus 2/3 of the fourth school, and test on the last 1/3.

On Cora and CiteSeer, for each experiment, 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 for a validation set used to find the appropriate regularization parameter λ. Common values of λ were 10^{-4} or 10^{-5}. On WebKB, we learned models for a variety of λ; here we show the best result.

In our experiments, we compared a baseline classifier (Content) with our link-based classifiers (Mode, Binary, Count). We compared the classifiers:

- **Content**: Uses only object attributes.
- **Mode**: Combines a logistic regression classifier over the object attributes with separate logistic regression classifiers over the mode of the In Links, Out Links, Co-In Links, and Co-Out Links.
- **Binary**: Combines a logistic regression classifier over the object attributes with a separate logistic regression classifier over the binary link statistics for all of the links.
- **Count-Link**: Combines a logistic regression classifier over the object attributes with a separate logistic regression classifier over the counts link statistics for all of the links.

7.6.1 Link Model Comparison

Table 7.1 shows details of our results using four different metrics (accuracy, precision, recall and F1 measure)[3] on the four data sets. Figure 7.2 shows a summary of the results for the F1 measure.

[3] A true positive is a document that is correctly labeled. Let TP be the number of true positives, FP be the number of false positive, TN be the number of true negatives, FP be the number of false negatives. Accuracy is the percentage of correctly labeled documents, $\frac{TP+TN}{TP+FP+TN+FN}$. Precision, recall and the F1 measure are macro-averaged over each of the categories. Precision is the percentage of documents that are predicted to be of a category, that actually are of that category $\frac{TP}{TP+FP}$. Recall is the percentage of documents that are predicted to be of a category, out of all the documents of the category $\frac{TP}{TP+FN}$. The F1 measure is $\frac{2PR}{R+P}$.

Table 7.1. Results with **Content**, **Mode**, **Binary** and **Count** models on CoraI, CoraII, CiteSeer and WebKB. Statistically significant results (at or above 90% confidence level) for each row are shown in bold.

CoraI				
	Content	Mode	Binary	Count
avg accuracy	68.14	82.35	77.53	**83.14**
avg precision	67.47	81.01	77.35	81.74
avg recall	63.08	80.08	76.34	81.20
avg F1 measure	64.17	80.0	75.69	**81.14**

CoraII				
	Content	Mode	Binary	Count
avg accuracy	67.55	83.03	81.46	**83.66**
avg precision	65.87	78.62	74.54	80.62
avg recall	47.51	75.27	75.69	76.15
avg F1 measure	52.11	76.52	74.62	**77.77**

CiteSeer				
	Content	Mode	Binary	Count
avg accuracy	60.59	71.01	69.83	**71.52**
avg precision	55.48	64.61	62.6	65.22
avg recall	55.33	60.09	60.3	61.22
avg F1 measure	53.08	60.68	60.28	**61.87**

WebKB				
	Content	Mode	Binary	Count
avg accuracy	87.45	88.52	78.91	87.93
avg precision	78.67	77.27	70.48	77.71
avg recall	72.82	73.43	71.32	73.33
avg F1 measure	71.77	73.03	66.41	72.83

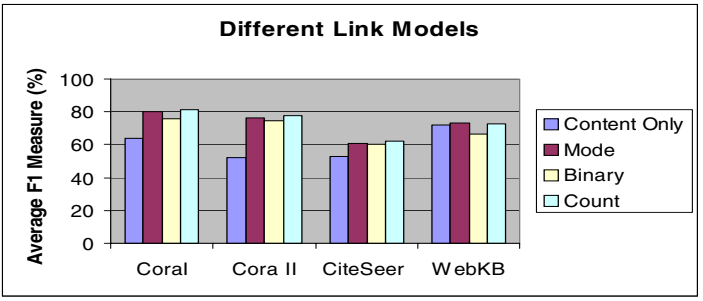


Fig. 7.2. Average F1 measure for different models (**Content**, **Mode**, **Binary** and **Count**) on four data sets (CoraI, CoraII, CiteSeer and WebKB).

In this set of experiments, all of the links (In Links, Out Links, Co-In Links, Co-Out Links) are used and we use a fixed ordering for the iterative classification algorithm.

For all four data sets, the link-based models outperform the content only models. For three of the four data sets, the difference is statistically significant at the 99% significance level. For three of the four data sets, **count** outperforms **mode** at the 90% significance level or higher, for both accuracy and F1 measure. Both **mode** and **count** outperform **binary**; the difference is most dramatic for CoraI and WebKB.

Clearly, the **mode**, **binary** and **count** link-based models are using information from the description of the link neighborhood of an object to improve classification performance. **Mode** and **count** seem to make the best use of the information; one explanation is that while **binary** contains more information in terms of which categories of links exist, it loses the information about which link category is most frequent. In many domains one might think that **mode** should be enough information, particulary bibliographic domains. So it is somewhat surprising that the **count** model is the best for our three citation data sets.

Our results on WebKB were less reliable. Small changes to the ways that we structured the classifiers resulted in different outcomes. Overall, we felt there were problems because the link distributions were quite different among the different schools. Also, after removing the other pages, the data set is rather small.

7.6.2 Effect of Link Types

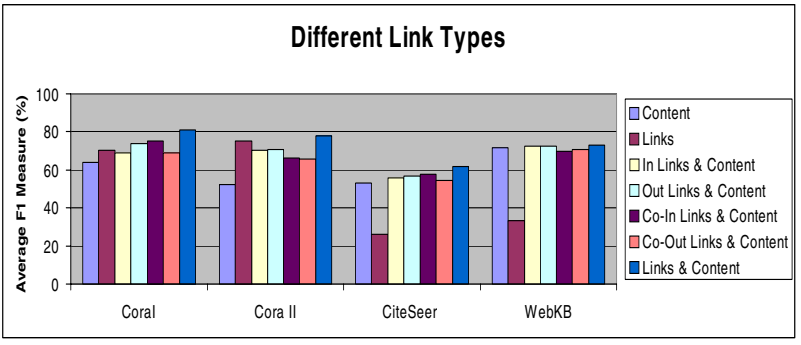


Fig. 7.3. Average F1 measure for **Count** on four data sets (CoraI, CoraII, CiteSeer and WebKB) for varying content and links (Content, Links, In Links & Content, Out Links & Content, Co-In Links & Content, Co-Out links & Content and Links & Content).

Next we examined the individual effect of the different categories of links: **In Links**, **Out Links**, **Co-In Links** and **Co-Out links**. Using the **count** model, we included in the comparision **Content**, with a model which used

all the links, but no content (**Links**),[4] and **Link & Content** (which gave us the best results in the previous section). Figure 7.3 shows the average F1 accuracy for the four of the data sets using different link types.

Clearly using all of the links performs best. Individually, the **Out Links** and **Co-In Links** seem to add the most information, although again, the results for WebKB are less definitive.

More interesting is the difference in results when using *only* **Links** versus **Links & Content**. For CoraI and Citeseer, **Links** only performs reasonably well, while for the other two cases, CoraII and WebKB, it performs horribly. Recall that the content helps give us an initial starting point for the iterative classification algorithm. Our theory is that, for some data sets, especially those with fewer links, getting a good initial starting point is very important. In others, there is enough information in the links to overcome a bad starting point for the iterative classification algorithm. This is an area that requires further investigation.

7.6.3 Prediction with Links Between Training and Test Sets

Next we were interested in investigating the issue of exploiting the links between test and training data for predictions. In other work, Neville and Jensen [25], Getoor *et al.* [12] and Taskar *et al.* [31] used 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 constantly expanding. 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.* [4], exploits the links between test and training.

In an effort to understand this phenomenon 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". The results are shown in Table 7.2. 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. These results demonstrate that the complete link structure is informative and can be used to improve overall performance.

7.6.4 Link-based Classification using Labeled and Unlabeled Data

In the previous section we experimented with making use of labeled data from the training set during testing. Next we explore the more general setting

[4] This model was inspired by results in [21].

Table 7.2. Avg F1 results using "Test Links Only" and "Complete Links" on CoraI, CoraII, CiteSeer and WebKB.

	Test Links Only			Complete Links		
	Mode	Binary	Count	Mode	Binary	Count
CoraI	75.85	71.57	79.16	80.00	75.69	81.14
CoraII	58.70	58.19	61.50	76.52	74.62	77.77
CiteSeer	59.06	60.03	60.74	60.68	60.28	61.87
WebKB	73.02	67.29	71.79	73.03	66.41	72.83

of learning with labeled and unlabeled data using the iterative algorithm proposed in Section 7.5. To better understand the effects of unlabeled data, we compared the performance of our algorithm with varying amounts of labeled and unlabeled data.

For two of the domains, CoraII and 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**).

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

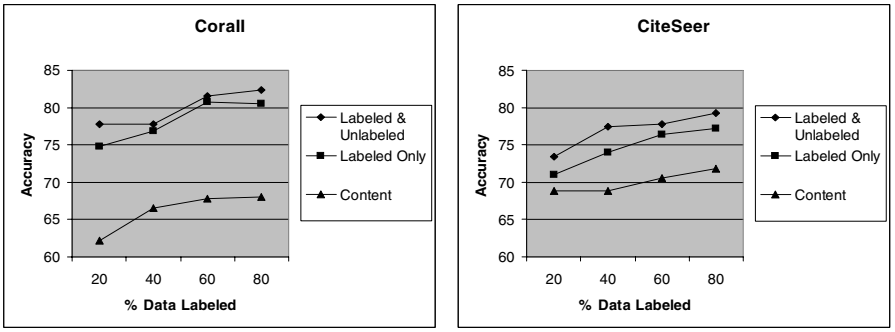
Figure 7.4 shows the results averaged over five different runs. The algorithm which makes use of all of the unlabeled data gives better performance than the model which uses only the labeled data.

For both data sets, 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.

7.6.5 Ordering Strategies

In the last set of experiments, we examined various ICA ordering strategies. Our experiments indicate that final test errors with different ordering strategy have a standard deviation around 0.001. There is no significant difference with various link diversity to order the predictions. We also compared with an ordering based on the posterior probability of the categories as done in Neville and Jensen [25], denoted PP.

While the different iteration schemes converge to about the same accuracy, their convergence rate varies. To understand the effect of the ordering scheme



(a) (b)

Fig. 7.4. (a) Results varying the amount of labeled and unlabeled data used for training on CoraII (b) and on CiteSeer. The results are averages of five runs.

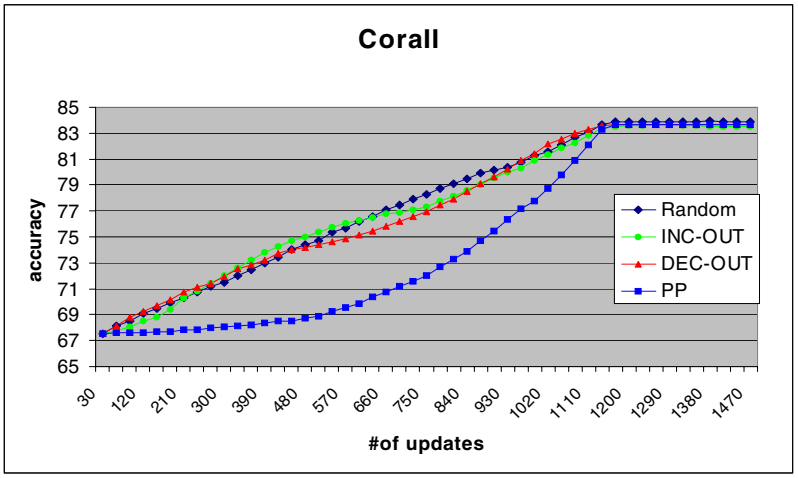


Fig. 7.5. The convergence rates of different iteration methods on the CoraII data set.

at a finer level of detail, Figure 7.5 shows an example of the accuracy of the different iteration schemes for the CoraII data set (to make the graph readable, we show only ordering by increasing diversity of out links (INC-Out) and decreasing diversity of out-links (DEC-Out); the results for in links, co-in links and co-out links are similar). Our experiments indicate that ordering by increasing link diversity converges faster than ordering by decreasing link diversity, and the RAND ordering converges the most quickly at the start.

7.7 Conclusions

Many real-world data sets have rich structures, where the objects are linked in some way. Link mining targets data-mining tasks on this richly-structured data. One major task of link mining is to model and exploit the link distributions among objects. Here we focus on using the link structure to help improve classification accuracy.

In this chapter we have proposed a simple framework for modeling link distributions, based on link statistics. We have seen that for the domains we examined, a combined logistic classifier built over the object attributes and link statistics outperforms a simple content-only classifier. We found the effect of different link types is significant. More surprisingly, the mode of the link statistics is not always enough to capture the dependence. Avoiding the assumption of homogeneity of labels and modeling the distribution of the link categories at a finer grain is useful.

Acknowledgments: I'd like to thank Prithviraj Sen and Qing Lu for their work on the implementation of the link-based classification system. This study was supported by NSF Grant 0308030 and 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 and should not be construed as an official Department of Defense position, policy, or decision unless so designated by other official documentation.

References

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

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

[3] Chakrabarti, S., 2002: *Mining the Web*. Morgan Kaufman.

[4] Chakrabarti, S., B. Dom and P. Indyk, 1998: Enhanced hypertext categorization using hyperlinks. *Proc of SIGMOD-98*.

[5] Cook, D., and L. Holder, 2000: Graph-based data mining. *IEEE Intelligent Systems*, **15**, 32–41.

[6] Craven, M., D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam and S. Slattery, 1998: Learning to extract symbolic knowledge from the world wide web. *Proc. of AAAI-98*.

[7] Dasarathy, B. V., 1991: *Nearest neighbor norms: NN pattern classification techniques*. IEEE Computer Society Press, Los Alamitos, CA.

[8] Dean, J., and M. Henzinger, 1999: Finding related pages in the World Wide Web. *Computer Networks*, **31**, 1467–79.

[9] Dzeroski, S., and N. Lavrac, eds., 2001: *Relational Data Mining*. Kluwer, Berlin.

[10] Feldman, R., 2002: Link analysis: Current state of the art. *Tutorial at the KDD-02*.

[11] Flach, P., and N. Lavrac, 2000: The role of feature construction in inductive rule learning. *Proc. of the ICML2000 workshop on Attribute-Value and Relational Learning: crossing the boundaries*.

[12] Getoor, L., N. Friedman, D. Koller and B. Taskar, 2002: Learning probabilistic models with link uncertainty. *Journal of Machine Learning Research*.

[13] 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*, N. Cercone, T. Y. Lin and X. Wu, eds., IEEE Computer Society, San Jose, US, 597–8.

[14] Giles, C., K. Bollacker, and S. Lawrence, 1998: CiteSeer: An automatic citation indexing system. *ACM Digital Libraries 98*.

[15] Jensen, D., 1999: Statistical challenges to inductive inference in linked data. *Seventh International Workshop on Artificial Intelligence and Statistics*.

[16] Jensen, D., and H. Goldberg, 1998: *AAAI Fall Symposium on AI and Link Analysis*. AAAI Press.

[17] Jensen, D, J. Neville. and B. Gallagher, 2004: Why collective inference improves relational classification. *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[18] Joachims, T., 1999: Transductive inference for text classification using support vector machines. *Proceedings of ICML-99, 16th International Conference on Machine Learning*, I. Bratko and S. Dzeroski, eds., Morgan Kaufmann, San Francisco, US, 200–9.

[19] Kleinberg, J., 1999: Authoritative sources in a hyperlinked environment. *Journal of the ACM*, **46**, 604–32.

[20] Kramer, S., N. Lavrac and P. Flach, 2001: Propositionalization approaches to relational data mining. *Relational Data Mining*, S. Dzeroski and N. Lavrac, eds., Kluwer, 262–91.

[21] Macskassy, S., and F. Provost, 2003: A simple relational classifier. *KDD Workshop on Multi-Relational Data Mining*.

[22] McCallum, A., and K. Nigam, 1998: A comparison of event models for naive Bayes text classification. *AAAI-98 Workshop on Learning for Text Categorization*.

[23] McCallum, A., K. Nigam, J. Rennie and K. Seymore, 2000: Automating the construction of Internet portals with machine learning. *Information Retrieval*, **3**, 127–63.

[24] Mitchell, T., 1999: The role of unlabeled data in supervised learning. *Proceedings of the Sixth International Colloquium on Cognitive Science*.

[25] Neville, J., and D. Jensen, 2000: Iterative classification in relational data. *Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, AAAI Press.

[26] Nigam, K., 2001: *Using Unlabeled Data to Improve Text Classification*. Ph.D. thesis, Carnegie Mellon University.

[27] Nigam, K., A. McCallum, S. Thrun, and T. Mitchell, 2000: Text classification from labeled and unlabeled documents using EM. *Machine Learning*, **39**, 103–34.

[28] Oh, H., S. Myaeng, and M. Lee, 2000: A practical hypertext categorization method using links and incrementally available class information. *Proc. of SIGIR-00*.

[29] Page, L., S. Brin, R. Motwani and T. Winograd, 1998: The page rank citation ranking: Bringing order to the web. Technical report, Stanford University.

[30] Popescul, A., L. Ungar, S. Lawrence and D. Pennock, 2002: Towards structural logistic regression: Combining relational and statistical learning. *KDD Workshop on Multi-Relational Data Mining*.

[31] Taskar, B., P. Abbeel and D. Koller, 2002: Discriminative probabilistic models for relational data. *Proc. of UAI-02*, Edmonton, Canada, 485–92.

[32] Taskar, B., E. Segal and D. Koller, 2001: Probabilistic classification and clustering in relational data. *Proc. of IJCAI-01*.

[33] Yang, Y., S. Slattery and R. Ghani, 2002: A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, **18**, 219–41.

[34] Zhang, T., and F. J. Oles, 2000: A probability analysis on the value of unlabeled data for classification problems. *Proc. 17th International Conf. on Machine Learning*, Morgan Kaufmann, San Francisco, CA, 1191–8.

[35] — 2001: Text categorization based on regularized linear classification methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5–31.

Part II

Applications