Blockmodeling and Text Classification

Christopher Hundt

May 19, 2008

1 Introduction

In this paper I discuss the application of stochastic blockmodeling to the domain of legal opinions, and specifically to the end of classifying those legal opinions into topics.

I begin by defining the basic social network problem and the stochastic blockmodel in Section 2. In Section 3 I introduce the problem of labeling legal opinions and discuss the particular features of this domain. Section 4 describes the supervised methods that I will use. Then Section 5 shows how blockmodeling can be used with the citation information from legal opinions and presents evidence that it captures their natural relational structure. In Section 6 I combine blockmodeling with supervised methods to improve the resulting classifiers and give experimental results. Finally, Section 7 compares this method to other ways of using citation structure.

2 Stochastic blockmodeling

Stochastic blockmodeling is a way to model the interactions between nodes in a social network. The idea was first developed by Harrison White and his colleagues in the 1970s [LW71, WBB76] as a group of techniques for identifying groups of entities, or nodes, in a social network which are "structurally equivalent." Structural equivalence meant that nodes within the same group tended to relate to each other and to other nodes in a similar fashion. The groups were called "blocks" because, when the nodes were rearranged so as to put nodes in the same group together, the adjacency matrix (the matrix in which the i, j entry describes the relationship between node i and node j) had sub-matrices (blocks) of similar values (as in Table 3 of [NS01]) which corresponded to the groups.

Further work developed the idea of a more statistically motivated "stochastic equivalence," under which definition two nodes were equivalent if one had the same probability as the other of having a relationship with any third node [FW81, HL81, HLL83]. In both structural and stochastic equivalence, the basic idea is that if two nodes are in the same group or block then you can't distinguish them just by looking at their relationships with other nodes.

The early approaches tended to look for block structure *a priori*, frequently using "attribute" information about the nodes, and then see how well it fit the relational data. An alternative approach is *a posteriori* blockmodeling, in which a statistical model is used to infer the block structure from the observed relational information. This was studied by Wasserman and Anderson in [WA87], and expanded upon to create models of varying generality and assumptions [NS97, NS01, KGT04].

In the remainder of this section I lay out the formal specification of the social network problem and describe the approach I will use.

2.1 Problem

The observed information is simply the relationships between different nodes. There are n nodes and for every $i < j \leq n$ the observation y_{ij} describes completely the nature of the relationship between node i and node j. Define $\mathcal{B}$ to be the set of possible different values of y_{ij}.

Many models of social networks have been developed in last 50 years. Some, such as the p_1 model [HL81] and the p^* family of models [RPW99], are based on statistics of the graph and provide parameterized distributions for generating a social network. Others assume that the nodes are embedded in some latent space and that their relative positions in that space influence their interactions [HRH02]. For stochastic blockmodels, we assume only that each node i has a "block" or "group" specified by the integer x_i and that the distribution of y_{ij} depends only on (x_i, x_j) and the other parameters of the model (in particular, each entry of the adjacency matrix is conditionally independent from the others given the group assignment). Generally, an arbitrary conditional distribution of relationships is allowed, which provides greater flexibility than other models, at the cost of requiring greater effort for learning.

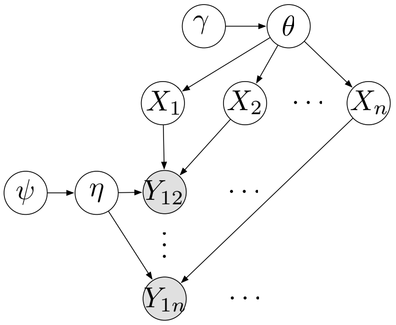


Figure 1: The stochastic blockmodel from [NS01]. See the text for explanation.

2.2 Model specification

The model I chose, introduced in [NS01], is an *a posteriori* blockmodel that assumes a fixed number K of possible groups and a multinomial assignment of groups to nodes in which each node's group assignment is conditionally independent of the other nodes' group assignments given the the multinomial parameter θ. The probability of an assignment in which there are n_k nodes of each group k is thus

$$p(n_1, n_2, \ldots, n_K) = \frac{n!}{n_1! n_2! \cdots n_K!} \theta_1^{n_1} \theta_2^{n_2} \cdots \theta_K^{n_K}.$$

A Dirichlet prior γ is placed on θ:

$$p(\theta|\gamma) \propto \theta_1^{\gamma-1} \theta_2^{\gamma-1} \cdots \theta_K^{\gamma-1};\ \sum_{k=1}^{K} \theta_k = 1$$

Given the complete group assignment x, each relationship y_{ij} is conditionally independent of all the others and is governed by parameters $\{\eta_{hk}\}_{1 \le h \le k \le K}$. Each η_{hk} is a $|\mathcal{B}|$-dimensional parameter for a multinomial distribution over $\mathcal{B}$:

$$\eta_{hk}^{b} = P(Y_{ij} = b | X_i = h, X_j = k).$$

These vectors η_{hk} are given a common Dirichlet prior with parameter ψ:

$$p(\eta|\psi) \propto \prod_{1 \le h \le k \le K} \left(\left(\eta_{hk}^{1}\right)^{\psi-1} \left(\eta_{hk}^{2}\right)^{\psi-1} \cdots \left(\eta_{hk}^{|\mathcal{B}|}\right)^{\psi-1} \right);\ \forall h, k\ \sum_{b=1}^{|\mathcal{B}|} \eta_{hk}^{b} = 1$$

3 Labeling Legal Opinions

Topical classification of legal opinions is an important area of text classification. Legal opinions are written accounts of reasoning behind the opinions of judges in making a legal decision. Judges frequently refer to other judges' opinions as guidance in resolving issues and, furthermore, in common-law jurisdictions like the United States, the decision of a higher court constitutes binding precedent that lower courts in its jurisdiction are expected to follow.

For this reason, legal opinions are compiled into large collections for reference by those parties interested in finding the "case law" related to a given issue. The great quantity of legal opinions written means that the collections must have some structure. One type of structure is classification into various topics.

Legal opinions are in some ways different from other types of texts. As discussed in [Tho01], legal opinions may discuss many issues, some of which could not reasonably be called a focus of the opinion. Furthermore, legal opinions contain many specialized "terms of art" which tend not to appear in other types of texts. This, however, makes the task somewhat easier, as it lends a certain uniformity of phrasing to the opinions.

3.1 Features: Bag of Words

As the main concern of this project is to test machine-learning techniques, I did not dedicate a great amount of time to creating a textual feature structure for the opinions. I chose a simple "bag-of-words" representation, in which each opinion is represented as a vector consisting of all the unique words and bigrams (pairs of consecutive words) appearing in the opinion and the number of times each appears. Word order is ignored; hence the phrase "bag of words." To reduce overfitting, I ignored all words that do not appear in at least four opinions. To aid in generalization I converted all words to their "stems" using a variation of Porter's stemming algorithm [Por97]. Furthermore, for each individual tagging task I ranked the features according to their "information gain": the entropy of the tag label minus the conditional entropy of the tag label given the feature value. All features ranked below 150 by this heuristic were discarded, as they were generally uninformative and tended to do more harm (through overfitting) than good when included.

3.2 Citation Structure

Another favorable feature of legal opinions is uniformity of citation format. The majority of jurisdictions follow the style laid out in *The Bluebook: A Uniform System of Citation*, and those who do not often follow some close variant. Furthermore, most case law is published in periodic "reporters," and the location in the relevant reporter is cited when an opinion wishes to cite a previously written opinion. For example, United States Supreme Court decisions are published in the U.S. Reporter, which is cited as "x U.S. y," where x is the volume of the reporter containing the opinion of interest, and y is the number of the page where it starts.

This extremely simple format and great uniformity makes it quite easy to tell what other authorities are cited by a given opinion. This structure can then be used as additional information about the opinions: the authorities cited may help indicate what the opinions are about. Investigating how blockmodeling can help use this information is the main purpose of this project.

4 Supervised Learning

The first task is to establish a base supervised-learning approach for using the bag-of-words features to choose tags. Since an opinion may have more than one label, my approach was to treat each label as creating a separate binary classification problem. Although the labels are not independent, I did not take advantage of any dependence that they might have; I trained an independent binary classifier for each tag. Below I briefly describe the supervised-learning models that I use.

4.1 Decision Trees

Decision trees may make branching decisions, each of which is a threshold on a given word or bigram count. It has potentially unlimited classification power, and attempts to limit overfitting by restricting the tree generation in some way. I generate a C4.5 tree by the J48 algorithm. [Qui93]

Decision trees serve as a useful base classifier for the purposes of this project because they perform well in choosing tags based on the bag-of-words features only and they have significant flexibility in how to use whatever citation-based features I may create.

4.2 Support Vector Machines

Support Vector Machine, or SVM, algorithms create a linear decision boundary of maximum margin in the feature space. (For an introduction to SVMs see [Bur98].) I used the Sequential Minimal Optimization (SMO) implementation of the SVM ([Pla98]) with a Gaussian kernel $\kappa(x, y) = \exp(-\|x - y\|^2/(2\sigma^2))$.

I use Support Vector Machines with the "bag-of-citations" method of using citation data (described in Section 7.1).

5 Using citation data

Although the opinions in the corpus I used are too close in time to cite each other, it is common for two or more opinions to cite the same other authority, such as a statute or an earlier opinion. One can thus create a "social network" from the opinions in which the relationship between two opinions is based on co-occurrence of citations between those opinions. It may then be possible to exploit this network structure in learning.

5.1 Stochastic blockmodeling on citation structure

It is not obvious how the relationship between two nodes should influence their likelihood of receiving the same tag: many pairs of opinions which do not have any tag in common will still have citations in common, and many pairs which do have tags in common will not happen to cite the same other authority. Since blockmodeling does not make any assumptions about how class membership affects relationship probabilities (only that it does affect relationship probabilities), it may be appropriate for this task.

For the purposes of applying the model from [NS01], I chose to define the relationship between each pair of opinions as having type 0, 1, 2, or 3, with the types defined as follows:

- Type 3: The two opinions have an identical citation (cite the same prior case, same section in a statute, etc.).
- Type 1 or Type 2: There is a pair of citations which refer to a similar source, but are not identical. For example, they might both cite from the same title in U.S. Code but not the same section. Type 2 indicates greater similarity than Type 1.
- Type 0: One opinion has no citation that is even similar to any citation in the other.

	Immigration	Discrimination	Sentencing
A	16	0	0
B	2	20	3
C	2	0	17

Table 1: Group assignments of labeled examples after convergence of Gibbs sampling with the blockmodel.

5.2 "Reasonableness" of blockmodeling

One important question when evaluating a new model is whether it seems to capture the inherent structure of the data it is used on. To that end, I took 20 positively labeled opinions from each of the Sentencing, Immigration, and Discrimination tags, set up the network structure described in the previous section, and ran Gibbs sampling to determine the posterior distribution of group assignments under the assumption of three groups ($K = 3$). Gibbs sampling is an iterative method of modeling the posterior by repeated sampling—see [CG92] for a detailed introduction. I followed the general method presented in [NS01], including linearly decreasing γ from $10n$ to $100K$ and increasing ψ from $1/n$ to 1, and monitoring statistics of the distribution for convergence. At the time of convergence each opinion was assigned to a single group with probability greater than 0.999. Table 1 summarizes the group assignment within each group, and reveals that blockmodeling captured the topical structure quite well: group A was entirely opinions tagged Immigration, group B had only five opinions (out of 25) that were not tagged Discrimination, and group C had only two opinions (out of 19) not tagged Sentencing. This is somewhat remarkable, considering that no tag information is directly encoded in the relationship structure.

Furthermore, Table 2 demonstrates that this grouping results in nontrivial block structure: different groups relate to each other in significantly different ways. For example, all three groups A, B, and C have different probabilities of relationship type 1 (a somewhat close citation match) with other opinions in the same group. And two opinions from group B (the one associated with Discrimination) are much more likely to have no similar citations than a pair from either of the other groups. The blockmodel is able to exploit this kind of information.

0	A	B	C
A	0.011	0.960	0.876
B	0.960	0.600	0.887
C	0.876	0.887	0.017

1	A	B	C
A	0.239	0.021	0.071
B	0.021	0.111	0.057
C	0.071	0.057	0.369

Table 2: Cross-group probabilities of relationship types 0 and 1 in the groups learned in the experiment described in Section 5.2.

6 Using blockmodeling to learn tags

Now that the relationship structure of the legal opinions has been defined and blockmodeling on labeled opinions has been shown to create groups that correspond to the labels, the next task is to devise a method for using blockmodeling to aid in labeling new opinions.

6.1 Inference

To take advantage of the blockmodel on unlabeled opinions I used the following simple scheme: choose a subset of nonintersecting tags $t_1, \ldots, t_d$ and assign groups to the training (labeled) set, assigning an opinion to group i if it is labeled with tag t_i and to group 0 if it is not labeled with any of the tags $t_1, \ldots, t_d$. Then choose the parameters as follows:

$$\hat{\theta}_k = \frac{n_k}{n} \qquad \hat{\eta}_{hk} = \frac{m^b_{hk} + 10}{n_h n_k + 40},\ h < k \qquad \hat{\eta}^b_{hh} = \frac{m^b_{hh} + 10}{\binom{n_h}{2} + 40}$$

where n_k is the number of opinions in group k (i.e., the number of opinions with tag t_k if $k \geq 1$ or the number of opinions with none of these tags if $k = 0$) and m^b_{hk} is the number of relationships of type b between an opinion in group h and an opinion in group k. The denominators in the expressions for $\hat{\eta}$ simply represent counts: $n_h n_k$ is the number of pairs of opinions with one in group h and one in group k and $\binom{n_h}{2}$ is the number of pairs of opinions where both are in group h. $\hat{\theta}$ and $\hat{\eta}$ are the maximum a priori (MAP) estimates when $\gamma = 1$ and $\psi = 11$.

Now these parameter estimates $\hat{\theta}$ and $\hat{\eta}$ can be applied to estimate the probability of membership in each group for a new, unlabeled example based on its relationships with the labeled examples: if the labeled opinions are numbered $1, \ldots, l$ (so the labels are $x_1, \ldots, x_l$), and we want to predict the label X_{l+1} of an unlabeled opinion, given its relationships with the labeled data $\{y_{i,l+1}; 1 \le i \le l\}$, we can compute

$$p_k := P(X_{l+1} = k) \propto \hat{\theta}_k \prod_{i=1}^{l} \hat{\eta}_{x_i k}^{y_{i,l+1}}. \tag{1}$$

6.2 Experimental results

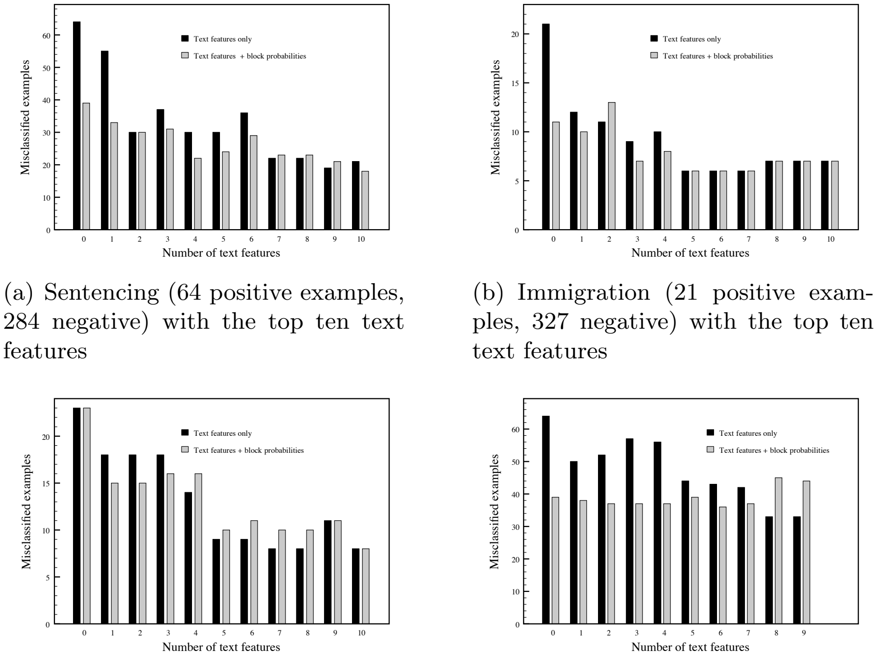
To evaluate this method of using citation structure I tested it on the entire labeled data set with the Immigration, Sentencing, Discrimination, and Intellectual Property tags. For each opinion i of the 348 labeled examples I calculated group membership probabilities $\{p_k^i\}_{k=0}^4$ using the maximum-likelihood parameters derived from the other 347 labeled examples, where p_k^i is the probability (under the ML parameters) that opinion i has the k-th tag, and p_0^i is the probability that opinion i does not have any of those four tags, computed as in (1).

I then added these group membership probabilities as features and tested whether they helped in classification. I did not use all the group membership probabilities in every tagging task, as that tended to cause overfitting. Instead, when creating the input for the classifier for the k-th tag, I added the computed probabilities p_k and p_0, i.e., the probability of having tag t_k and the probability of having none of the four tags.

In general, the word and bigram occurrences alone are very informative and adding citation information to the entire feature set helped little if at all. However, some insight into the interaction of the citation data with the textual information can be gained by showing how citation data helps (or does not help) classification when only some subset of the textual features is considered. To that end, I ran the following experiment: select the top ten word or bigram features by information gain, and use the J48 tree learner to learn on various subsets of those ten features with and without group probabilities from the blockmodel parameter estimation.

The cross-validation errors from these experiments are shown in Figure 2. For the Sentencing and Immigration tags (2(a) and 2(b)) it is clear that the blockmodel features help significantly by themselves or with only a single word or bigram feature. But in the presence of more text features, the blockmodel information appears to be redundant.

For the Discrimination tag (Figure 2(c)) it appears that blockmodel information does not help much, but the data does show an interesting feature: by itself the blockmodel information is useless; the tree simply classifies all examples as negative. But in the presence of a single text feature the blockmodel reduces error significantly. That is, the number correct for one



(a) Sentencing (64 positive examples, 284 negative) with the top ten text features

(b) Immigration (21 positive examples, 327 negative) with the top ten text features

(c) Discrimination (23 positive examples, 325 negatiev) with the top ten text features

(d) Sentencing (64 positive examples, 284 negative) with text features 11–19

Figure 2: The effect of adding group probabilities from blockmodeling to the text features. See the text for more details.

text feature with blockmodel is more than the sum of the number for the blockmodel alone and for the text feature alone. This suggests that the blockmodel information combines well with the textual information.

Finally, Figure 2(d) shows the result of combining blockmodel probabilities with the text features ranked 11 through 20 in terms of information gain for Sentencing. Here we see that the blockmodel probabilities help more consistently, indicating that this method may be of particular use in situations where the text features provide less information.

7 Other methods of using citation information

Blockmodeling seems to relate citation information to tag information well, but it is useful to compare it to other methods of doing the same. A few are presented below and their performance is compared to blockmodeling.

7.1 Bag of citations

One way of using citations is to create a list of authorities cited anywhere in the data and give each opinion a binary vector indicating which authorities from the list it cites. I call this the "bag-of-citation" method, as it treats the citation information from each document as being an unordered collection of citations. It has been used in the past as a way to combine citation information with text information. (See, for example, [CH01].)

To avoid overfitting I left out all citations that do not occur in at least four documents. Still, this creates more than one hundred new features, and each feature is rather insignificant by itself, so decision trees did not work well. Thus I utilized SVMs with these features (Section 4.2).

7.2 Label propagation

Label propagation has previously been suggested as a way of doing semi-supervised learning that takes advantage of graph structure [ZG02]. This method calls for creating a matrix f in which f_{ij} is an indicator that opinion i has label j if i is labeled, and f_{ij} is initialized arbitrarily if i is unlabeled (with the constraint that $\sum_j f_{ij} = 1$). Then a matrix P is created where P_{ij} is the edge weight between opinions i and j, normalized so row sums are 1. Then the following two steps are repeated until convergence:

1. Set $f \leftarrow Pf$.

2. For labeled indexes i, reset f_{ij} to label indicators.

	Immigration	Sentencing	Discrimination
A	19	18	10
B	1	1	0
C	0	1	10

Table 3: Group assignments of labeled examples after convergence of EM on the bag-of-citations features.

To use label propagation I chose the edge weight between opinions i and j to be $10^{y_{ij}}$, where y_{ij} is the type of relationship that exists between i and j as defined in Section 5.1. This exponential mapping reflects the fact that the higher link types represent much greater similarity in citations.

My method for testing this algorithm was similar to the method I used to test blockmodeling (Section 6.1). I modified the labels to be either Immigration, Sentencing, Discrimination, Intellectual Property, or None. I then ran the algorithm once for each labeled opinion. During run i I pretended opinion i was unlabeled. After convergence of the run I added the $\{f_{ij}\}_{j=1}^{5}$ as features for opinion i, to be used in the J48 decision-tree learner. As in the case of blockmodeling, when training a classifier for tag t_k I only used f_{ik} and f_{i5} (the entry for tag t_k and the entry for None).

7.3 Comparison of methods

7.3.1 Unsupervised clustering

A preliminary question is whether another method can produce a block structure in an unsupervised way as good as the structure produced by blockmodeling. Label propagation is designed specifically with semi-supervised learning in mind, so it cannot be used in an unsupervised way. But the bag-of-citations features can be used with any number of unsupervised methods. I chose clustering with EM, with a mixture-of-Gaussians model and the number of clusters fixed at 3. I ran the EM algorithm 5000 times on the bag-of-citation features and chose the parameter assignment with the highest likelihood. Repeating this process multiple times yields various local minima, but none of them result in group structures that correspond very closely to the actual tags. The resulting three groups found in one such run are summarized in Table 3. Comparison to Table 1 suggests that blockmodeling does a significantly better job of capturing the natural structure of the data.

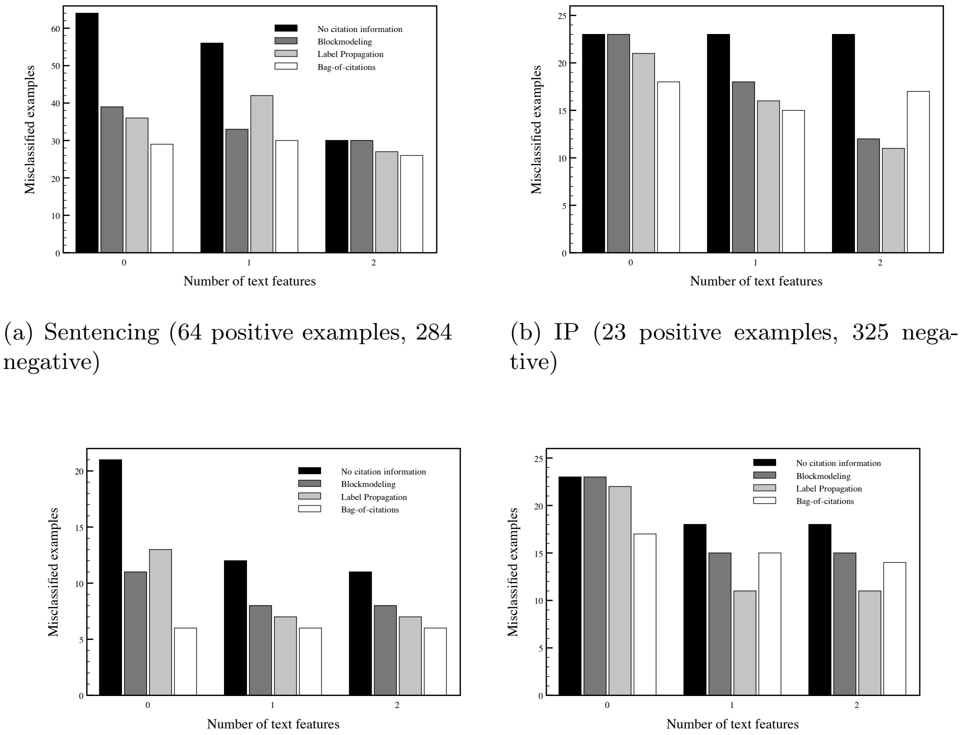
7.3.2 Supervised learning with text features

It is also useful to compare the performance of blockmodeling with other models in the context of providing extra information for supervised learning. Figure 3 shows results from all three ways of using citation information with no text features, with text feature 10, and with text features 9 and 10. All three methods get good results in the presence of few text features, but there are significant differences in those results between methods.

The bag-of-citations method performs very well, frequently beating the other methods. This may mean that some useful information is sacrificed by transforming the citation features to blockmodel or label-propagation features. On the other hand, this transformation has some advantages. Observe that in none of the four examples does the performance of the SVM with bag-of-citations features improve significantly as more text features are added, but this increase does occur with the other methods. In the case of Intellectual Property (Figure 3(b)) the performance of the other two methods surpasses that of the bag-of-citations method once two text features are present. One explanation for this is that, by sacrificing some information, the features from blockmodeling and label propagation become a little more general; unlike most of the bag-of-citations features they are nonzero for more than a few opinions and refer more to a general category than a specific fact. This means that the J48 learner can plausibly (and does in practice) branch on one of the blockmodel features to determine how it interprets the text features, while this is unlikely for bag-of-citations features—one feature carries too little information to justify a major dependence on it. This is precisely the reason I had to use SMO for the bag-of-citations method.

Label propagation performs slightly better than blockmodeling. As label propagation is based on the simplifying assumption that opinions with the same label are likely to have closer relationships, this suggests that the blockmodel's power to model more general relationship probabilities is not being put to use here.

To test this I repeated the experiment with label propagation and blockmodeling as above, but this time with an artificial network generated according to the probabilities shown in Table 4. Note that the Discrimination-labeled opinions have equal relationship frequency with all other opinions, breaking the simplifying assumption of label propagation. Results are summarized in Figure 4. On the Immigration tag, where the assumption of label propagation holds, both methods perform very well, achieving near perfect classification. On the Discrimination tag, where the assumption is not true, label propagation is useless, while blockmodeling gets very good results.



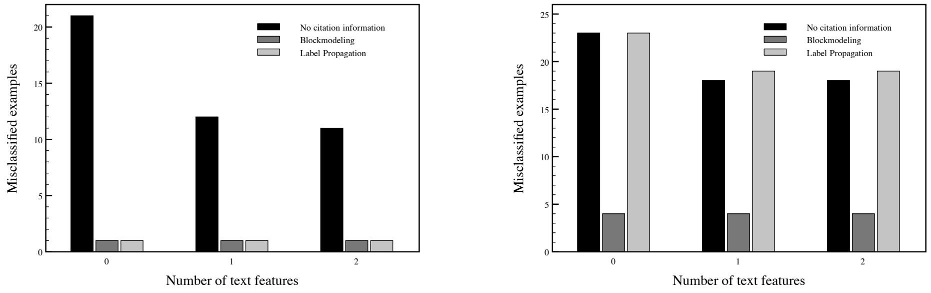
(a) Sentencing (64 positive examples, 284 negative)

(b) IP (23 positive examples, 325 negative)

(c) Immigration (21 positive examples, 327 negative)

(d) Discrimination (23 positive examples, 325 negative)

Figure 3: Experimental results of different methods of using citations



(a) Immigration (21 positive examples, 327 negative)

(b) Discrimination (23 positive examples, 325 negative)

Figure 4: Experimental results on the artificially generated data

0	Immigration	Sentencing	Discrimination	Other/None
Immigration	0.45	0.95	0.75	0.85
Sentencing	0.95	0.55	0.75	0.75
Discrimination	0.75	0.75	0.75	0.75
Other/None	0.85	0.75	0.75	0.65

3	Immigration	Sentencing	Discrimination	Other/None
Immigration	0.55	0.05	0.25	0.15
Sentencing	0.05	0.45	0.25	0.25
Discrimination	0.25	0.25	0.25	0.25
Other/None	0.15	0.25	0.25	0.35

Table 4: Cross-group probabilities of relationship types 0 and 3 in the artificially generated data. There were no relationships of type 1 or 2.

8 Conclusion

These results lend convincing support to the stochastic blockmodel as an effective way to infer the natural structure of a social network—the block structure found in an unsupervised way not only “looks reasonable,” but corresponds closely to the known underlying groups.

Blockmodeling was also shown to be useful as an aid to supervised learning, especially when the nonrelational information available is of poor quality. But other, simpler methods turned out to do just as well or better in these situations. It seems likely that the reason for this is that the complex relational structure that can be captured by blockmodeling does not exist to a great extent in this application. If the simplifying assumption that all nodes are more likely to have close relationships within the same group, inherent to the label propagation method, is true then the flexibility to deal with more complex situations serves as a detriment to blockmodeling in this domain.

References

[Bur98] C. Burges, *A tutorial on support vector machines for pattern recognition*, Data Mining and Knowledge Discovery **2** (1998), no. 2, 121–167.

[CG92] G. Casella and E. George, *Explaining the gibbs sampler*, The American Statistician **46** (1992), no. 3, 167–174.

[CH01] D. Cohn and T. Hofmann, *The missing link - a probabilistic model of document content and hypertext connectivity*, Neural Information Processing Systems 13, 2001.

[FW81] S. Fienberg and S. Wasserman, *Categorical data analysis of single sociometric relations*, Sociological Methodology **12** (1981), 156–192.

[HL81] P. Holland and S. Leinhardt, *An exponential family of probability distributions for directed graphs*, Journal of the American Statistical Association **76** (1981), no. 373, 33–50.

[HLL83] P.W. Holland, K.B. Laskey, and S. Leinhardt, *Stochastic blockmodels: First steps*, Social Networks **5** (1983), 109–137.

[HRH02] P. Ho, A. Raftery, and M. Handcock, *Latent space approaches to social network analysis*, 2002.

[KGT04] C. Kemp, T. Griffiths, and J. Tenenbaum, *Discovering latent classes in relational data*, Tech. Report AI Memo 2004-019, Massachusetts Institute of Technology, 2004.

[LW71] F. Lorrain and H. C. White, *Structural equivalence of individuals in social networks*, Journal of Mathematical Sociology **1** (1971), 49–80.

[NS97] K. Nowicki and T. Snijders, *Estimation and prediction for stochastic blockmodels for graphs with latent block structure*, Journal of Classification **14** (1997), 75–100.

[NS01] ———, *Estimation and prediction for stochastic blockstructures*, Journal of the American Statistical Association **96** (2001), no. 455, 1077.

[Pla98] J. Platt, *Sequential minimal optimization: A fast algorithm for training support vector machines*, 1998.

[Por97] M. F. Porter, *An algorithm for suffix stripping*, 313–316.

[Qui93] J. R. Quinlan, *C4.5: programs for machine learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[RPW99] G. Robins, P. Pattison, and S. Wasserman, *Logit models and logistic regressions for social networks: Iii. valued relations*, Psychometrika **64** (1999), no. 3, 371–394.

[Tho01] P. Thompson, *Automatic categorization of case law*, ICAIL '01: Proceedings of the 8th international conference on Artificial intelligence and law (New York, NY, USA), ACM, 2001, pp. 70–77.

[WA87] S. Wasserman and C. Anderson, *Stochastic a posteriori blockmodels: Construction and assessment*, Social Networks **9** (1987), 1–36.

[WBB76] H. White, S. Boorman, and R. Breiger, *Social structure from multiple networks. i. blockmodels of roles and positions*, The American Journal of Sociology **81** (1976), no. 4, 730–780.

[ZG02] X. Zhu and Z. Ghahramani, *Learning from labeled and unlabeled data with label propagation*, Tech. Report CMU-CALD-02-107, Carnegie Mellon University, 2002.