Spectral Learning

Sepandar D. Kamvar
SCCM
Stanford University
Stanford, CA 94305-9040
sdkamvar@cs.stanford.edu

Dan Klein
Computer Science Dept.
Stanford University
Stanford, CA 94305-9040
klein@cs.stanford.edu

Christopher D. Manning
Computer Science Dept.
Stanford University
Stanford, CA 94305-9040
manning@cs.stanford.edu

Abstract

We present a simple, easily implemented spectral learning algorithm which applies equally whether we have no supervisory information, pairwise link constraints, or labeled examples. In the unsupervised case, it performs consistently with other spectral clustering algorithms. In the supervised case, our approach achieves high accuracy on the categorization of thousands of documents given only a few dozen labeled training documents for the 20 Newsgroups data set. Furthermore, its classification accuracy increases with the addition of unlabeled documents, demonstrating effective use of unlabeled data. By using normalized affinity matrices which are both symmetric and stochastic, we also obtain both a probabilistic interpretation of our method and certain guarantees of performance.

1 Introduction

Spectral algorithms use information contained in the eigenvectors of a data affinity (i.e., item-item similarity) matrix to detect structure. Such an approach has proven effective on many tasks, including information retrieval [Deerwester *et al.*, 1990], web search [Page *et al.*, 1998; Kleinberg, 1998], image segmentation [Meila and Shi, 2000], word class detection [Brew and Schulte im Walde, 2002] and data clustering [Ng *et al.*, 2002]. But while spectral algorithms have been very useful in unsupervised learning (clustering), little work has been done in developing spectral algorithms for supervised learning (classification).

In this work, we consider the adaptation of spectral clustering methods to classification. We first present a method for combining item similarities with supervision information to produce a Markov transition process between data items. We call this Markov process the "interested reader" model by appeal to the special case of text clustering/classification. Our algorithm incorporates supervisory information whenever it is available, either in the form of pairwise constraints or labeled data (or both). Empirically, our algorithm achieves high accuracy when supplied either small amounts of labeled data (Section 4) or small numbers of pairwise constraints (Section 5).

2 The "Interested Reader" Model

We propose a Markov chain model similar in spirit to the "random surfer" model of [Page *et al.*, 1998].[1] This description is motivated in the context of text categorization, but the model depends only on notions of pairwise data similarity and is completely general. In the model, there is a collection of documents, each of which has some (possibly unknown) topic. A reader begins with some document of interest and continues to read successive documents. When she chooses the next document to read, she tries to read another document on the same topic, and hence will prefer other documents which are similar to her current document. Some mapping between similarities and transition probabilities must be chosen; we describe a specific choice in Section 3.

These transition probabilities define a Markov chain among the documents in the collection. If there exist distinct topic areas in the document set (or, generally, if there are clusters in the data), this Markov chain will be composed of subsets that have high intra-set transition probabilities, and low inter-set transition probabilities. We will refer to these subsets as *cliques*. Each of the cliques corresponds to a topic in the text clustering problem.

Of course, the natural clusters in the data need not be perfectly compatible with document labels, and we have said nothing about the use of supervision information. In Section 4, we use supervision to override the similarity-based transition probilities. For example, we will disallow transition between two documents which are known to be differently-labeled, regardless of their pairwise similarity.

3 Spectral Clustering Algorithms

In this section, we discuss the process of turning an *affinity matrix* A of pairwise document similarities into a normalized Markov transition process N. The eigenvectors of N are then used to detect blocks or or near-blocks in N, which will correspond to clusters of the data.

[1] Note that there is an important difference between the way these two models are used; the random surfer model is used for the first left eigenvector of the transition matrix, which indicates the relative amount of time the process spends at each data item. On the other hand, we are interested in right eigenvectors of our transition matrix, which more straightforwardly relate to (near-)block structure in the transition matrix.

Form spectral representation:

1. Given data B, form the affinity matrix $A \in \mathbb{R}^{n \times n} = f(B)$.
2. Define D to be the diagonal matrix with $D_{ii} = \sum_j A_{ij}$.
3. Normalize: $N = (A + d_{max} I - D)/d_{max}$.
4. Find $x_1, \ldots, x_k$, the k largest eigenvectors of N and form the matrix $X = [x_1, \ldots, x_k] \in \mathbb{R}^{n \times k-1}$.
5. Normalize the rows of X to be unit length.

For clustering:

6. Treating each row of X as a point in $\mathbb{R}^k$, cluster into k clusters using k-means or any other sensible clustering algorithm.
7. Assign the original point x_i to cluster j if and only if row i of X was assigned to cluster j.

For classification:

6. Represent each data point i by the row X_i of X.
7. Classify these rows as points in $\mathbb{R}^k$ using any reasonable classifier, trained on the labeled points.
8. Assign the data point i the class c that X_i was assigned.

Figure 1: Spectral Learning Algorithm.

Algorithm	**Normalization**	$\boldsymbol{\nu(A, D)}$
MNCUT	Divisive	$N = D^{-1}A$
NJW	Symmetric Divisive	$N = D^{-1/2}AD^{-1/2}$
LSA	None	$N = A$
SL	Normalized Additive	$N = (A + d_{max}I - D)/d_{max}$

Table 1: Normalizations used by spectral methods.

3.1 Calculating the Transition Matrix

In order to fully specify the data-to-data Markov transition matrix, we must map document similarities to transition probabilities. Let A be the affinity matrix over documents whose elements A_{ij} are the similarities between documents i and j. When we are given documents i as points x_i and a distance function $d(x_i, x_j)$, a common definition is $A_{ij} = e^{-d(x_i,x_j)/2\sigma^2}$, where σ is a free scale parameter. In LSA [Deerwester *et al.*, 1990], we are given a row-normalized term-document matrix B, and A is defined to be $B^T B$ (the cosine similarity matrix [Salton, 1989]).

We may map document similarities to transition probabilities in several of ways. We can define $N = D^{-1}A$ [Meilă and Shi, 2001], where D is the diagonal matrix whose elements $D_{ii} = \sum_j D_{ij}$. This corresponds to transitioning with probability proportional to relative similarity values. Alternatively, we can define $N = (A + d_{max}I - D)/d_{max}$ [Fiedler, 1975; Chung, 1997], where d_{max} is the maximum rowsum of A. Here, transition probabilities are sensitive to the absolute similarity values. For example, if a given document is similar to very few others, the interested reader may keep reading that document repeatedly, rather than move on to another document. While either of these normalizations are plausible, we chose the latter, since it had slight empirical performance benefits for our data.

In [Meilă and Shi, 2001], it is shown that a probability transition matrix N for a Markov chain with k strong cliques will have k piecewise constant eigenvectors, and they suggest clustering by finding approximately equal segments in the top k eigenvectors. Our algorithm uses this general method as well, but some of the details differ; our algorithm is shown in Figure 1. This algorithm is most similar to the algorithm presented in [Ng *et al.*, 2002], which we call NJW after its authors. In fact, the only difference is the type of normalization used. There are two differences between our algorithm and MNCUT from [Meilă and Shi, 2001]; the normalization of A is again different, and, additionally, MNCUT does not row normalize X (step 5). Table 1 describes the different types of normalizations and mentions some algorithms that use them.

	Spectral Learning	**k-means**
3 NEWS	0.84	0.20
20 NEWS	0.36	0.07
LYMPHOMA	0.50	0.10
SOYBEAN	0.41	0.34

Table 2: A comparison of Spectral Learning and k-means.

It should be noted that for data sets where there are distant outliers, additive normalization can lead to very poor performance. This is because, with additive normalization, the outliers become their own clique. Therefore, the clusters will represent outliers rather than true clusters. In a dataset where there are distant outliers, divisive normalization is likely to lead to better performance.

3.2 Parameter Selection

The importance of parameter selection is often overlooked in the presentation of standard spectral clustering methods. With different values of σ, the results of spectral clustering can be vastly different. In [Ng *et al.*, 2002], the parameter σ is chosen based on that value of σ that gives the least distorted clusters.

In our text experiments, the data B was a term-document matrix, and the similarity function f gave the pairwise cosine similarities, with an entry A_{ij} set to zero if neither i was one of the top k nearest-neighbors of j nor the reverse. Thresholding the affinity matrix in this manner is very useful, as spectral methods empirically work much better when there are zeros in the affinity matrix for pairs of items that are not in the same class. For our experiments, we chose $k = 20$; however, one may learn the optimal k in the same manner that [Ng *et al.*, 2002] learn the optimal scale factor σ.

3.3 Empirical Results

We compared the spectral learning algorithm in Figure 1 to k-means on 4 data sets:

- 20 NEWSGROUPS a collection of approximately 1000 postings from each of 20 usenet newsgroups.[2]
- 3 NEWSGROUPS 3 of the 20 newsgroups: sci.crypt, talk.politics.mideast, and soc.religion.christian.
- LYMPHOMA gene expression profiles of 96 normal and malignant lymphocyte samples. There are 2 classes: Diffuse Large B-Cell Lymphoma (42 samples), and Non-DLCBL (54 samples) [Alizadeh, 2000].

[2] From http://www.ai.mit.edu/~jrennie/20Newsgroups/; a total of 18828 documents. Documents were stripped of headers, stopwords, and converted to lowercase. All numbers were discarded. All words that occur in more than 150 or less than 2 documents were removed.

- SOYBEAN is the SOYBEAN-LARGE data set from the UCI repository. 15 classes.[3]

The results are shown in Table 2. The numbers reported are adjusted Rand Index values [Hubert and Arabie, 1985] for the clusters output by the algorithms. The Rand Index is frequently used for evaluating clusters, and is based on whether pairs are placed in the same or different clusters in two partitionings. The Adjusted Rand Index ranges from -1 to 1, and its key property is that the expected value for a random clustering is 0. The result that spectral methods generally perform better than k-means is consistent with the results in [Ng *et al.*, 2002; Brew and Schulte im Walde, 2002]. In some cases, the poor performance of k-means reflects its inability to cope with noise dimensions (especially in the case of the text data) and highly non-spherical clusters (in the case of the composite negative cluster for LYMPHOMA).[4] However, spectral learning outperforms k-means on the SOYBEAN dataset as well, which is a low-dimensional, multi-class data set.

4 Spectral Classification

In the previous section, we described *clustering* a data set by creating a Markov chain based on the similarities of the data items with one another, and analyzing the dominant eigenvectors of the resulting Markov matrix. In this section, we show how to *classify* a data set by making two changes. First, we modify the Markov chain itself by using class labels, when known, to override the underlying similarities. Second, we use a classification algorithm in the spectral space rather than a clustering algorithm.

4.1 Modifying the "Interested Reader" Model

The model described in Section 2 can be modified to incorporate labeled data in the following simple manner. If the interested reader happens to be at a labeled document, the probability that she will choose another labeled document of the same category is high, while the probability that she will choose a labeled document of a different category is low (or zero). Transition probabilities to unlabeled documents are still proportional to their similarity to the current source document, whether the current document is labeled or not.

We wish to create a Markov matrix that reflects this modified model. We propose doing this in the following manner, using the normalization introduced in Section 3. For most similarity functions, the maximum pairwise similarity value is 1, and the minimum similarity is 0. Therefore, we would like to say that two points in the same class are maximally similar, and two points in different classes are minimally similar:

1. Define the affinity matrix A as in the previous algorithms.
2. First, for each pair of points (i, j) that are in the same class, assign the values $A_{ij} = A_{ji} = 1$.
3. Likewise, for each pair of points (i, j) that are in different classes, assign the values $A_{ij} = A_{ji} = 0$.
4. Normalize $N = \frac{1}{d_{max}}(A + d_{max} I - D)$.

This gives us a symmetric Markov matrix describing the "interested reader" process which uses supervisory information when present, and data similarities otherwise. A strength of this model lies in the fact that it incorporates unlabeled data, whereas the majority of classification models deal strictly with the labeled data. A benefit of additive normalization is that, after the affinities are adjusted, same-labeled pairs will always have a higher (or equal) mutual transition probability than unlabeled pairs. This will not necessarily be the case with other normalization schemes.

4.2 A Spectral Classification Algorithm

Again, if natural classes occur in the data, the Markov chain described above should have cliques. Furthermore, the cliques will become stronger as the number of labeled documents increases. Given this model, we wish to categorize documents by assigning them to the appropriate clique in the Markov chain. The spectral clustering methods given in Section 3 can be adapted to do classification by replacing the final few steps (clustering in spectral space) with the steps shown in Figure 1 (which classify in spectral space).

The key differences between the spectral classifier and the clustering algorithm are (a) that our transition matrix A incorporates labeling information, and (b) we use a classifier in the spectral space rather than a clustering method. What is novel here is that this algorithm is able to classify documents by the similarity of their transition probabilities to known subsets of B. Because the model incorporates both labeled and unlabeled data, it should improve not only with the addition of labeled data, but also with the addition of unlabeled data. We observe this empirically in Section 4.3.

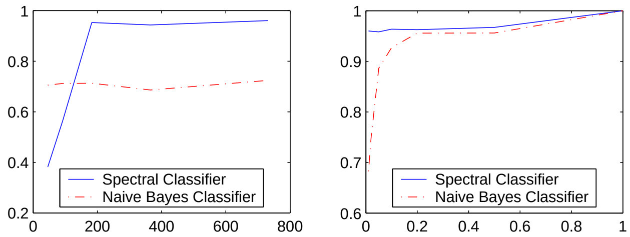
4.3 Empirical Results

Cliques

It was suggested in Section 4.2 that the Markov chain described above will have cliques, that is, subsets of nodes in the graph that are internally fast-mixing, but are not mutually fast-mixing. Figure 4 shows the thresholded sparsity pattern for the affinity matrices for the 3 Newgroups data set, as labeled data is added. The left matrix is the affinity matrix for 1% labeled data. Even the underlying similarities show block-like behavior, if weakly. To the extent that the unlabeled data gives a block-like affinity matrix, clusters naturally exist in the data; this is the basis for spectral clustering. The subsequent matrices have increasing fractions of data labeled. The effect of adding labeled data is to sharpen and coerce the natural clusters into the desired classes. As more labels are added, the blocks become clearer, the cliques become stronger, and, in the limit of 100% labeled data, the interested reader will never accidently jump from a document of one topic to one of another.

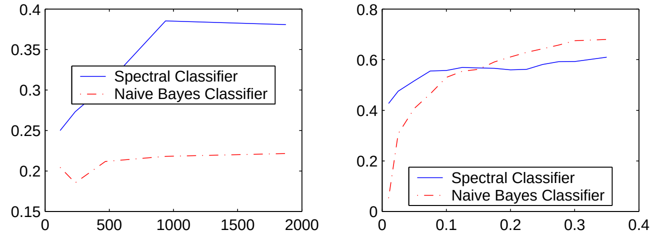
[3] There are has 562 instances, 35 features, and 15 different classes. It is nominal; Hamming distance was used.

[4] For some of these sets, the k-means numbers are low. This is partially illusory, due to the zeroed expectation of the adjusted Rand index, and partially a real consequence of the sparse high-dimensionality of the text data. Better k-means results on text typically require some kind of aggressive dimensionality reduction, (usually LSA, another spectral method) or careful feature selection (or both).



(a) Adding Unlabeled Data (b) Adding Labeled Data

Figure 2: Categorization accuracy on the 3 Newsgroups task as the number of (a) unlabeled and (b) labeled points increases. In (a), 12 labeled documents and the given number of unlabeled docments were used as a training set. In (b), the training set is all of 3 Newsgroups, with the given fraction labeled. In both cases, the test set for a given run consisted of all documents in 3 Newsgroups whose labels were not known during training for that run.



(a) Adding Unlabeled Data (b) Adding Labeled Data

Figure 3: Categorization accuracy on the 20 Newsgroups task as the (a) amount of unlabeled data and (b) fraction of labeled data increases.

Accuracies

To validate the utility of spectral classification, we performed the following experiments on the 20 Newsgroups data set.

We built two batch classifiers. The first was a standard multinomial Naive Bayes (NB) classifier with standard add-one smoothing. The second was a spectral classifier as described above, which used a single-nearest neighbor classifier to perform the final classification in the spectral space. The affinity matrix A was the thresholded cosine similarity between documents.[5] We note here that the thresholding is important, since it weakens the effect of outliers. Furthermore, it saves space and computation time, since the resulting affinity matrix is sparse.

We split the data into a labeled training set and an unlabeled test set. For classification, the spectral classifier processed both the training and test set, but was evaluated on the test set only.

Figure 2(a) shows the effect of using a sample of 12 documents from the 3 Newsgroups data as a labeled training set, with an increasing number of unlabeled documents as a test set. The accuracy of the NB classifier is, of course, constant up to sampling variation, since it discards unlabeled data. The spectral classifier is more accurate than the NB classifier when given sufficiently many additional unlabeled documents to incorporate.

Figure 2(b) shows the effect of supplying increasingly large fractions of the 3 Newsgroups data set as labeled training instances, and using the remainder of the data set as test instances. The spectral classifier outperforms Naive Bayes, more substantially so when there is little labeled data. Figures 3(a) and (b) show the same graphs for the 20 Newsgroups data set. Again, spectral classification performs well, especially when less than 10% of the data is labeled. It should be noticed that, for this data set, Naive Bayes outperforms the spectral classifier in the strongly supervised case (>15% labeled data). The strength of Spectral Learning lies in incorporating unlabeled data, and, for the strongly supervised case, this is of less value.

Spectral Space

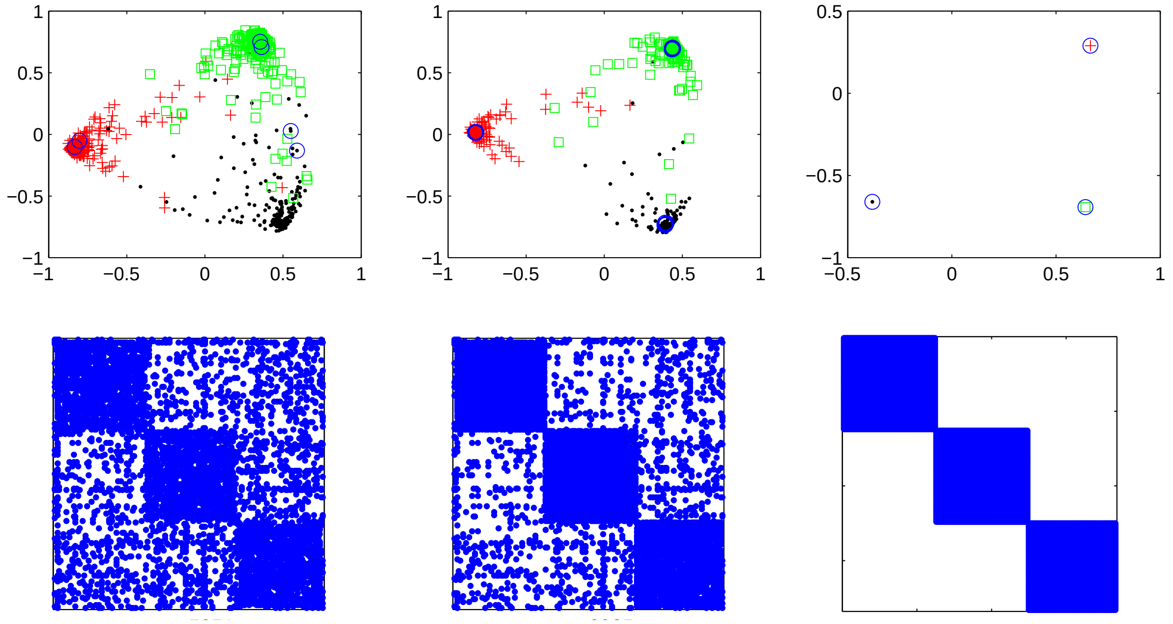
To further investigate the behavior of the spectral classifier, we performed the following experiment. We took the 3 Newsgroups data and labeled various fractions of each of the 3 classes. We then plotted each document's position in the resulting 3-dimensional spectral space (the space of the rows of the matrix X as defined by our spectral learning algorithm). Figure 4 shows dimensions 2 and 3 of these plots. With no labels, the data does not tightly cluster. As we add labels (circled points), two things happen. First, the labeled points move close to same-labeled points, away from different-labeled points, and generally towards the outside, since they are "hubs" of the Markov process. Second, they pull the unlabeled points out radially along with them. This is effective in that it seems to pull the classes apart, even though the classes were not naturally very strong clusters in the unlabeled data.

4.4 Related Work

In [Yu and Shi, 2001], a spectral grouping method, which they call "Grouping with Bias", is presented that allows for top-level bias, as in labeled data. They formulate the problem as a constrained optimization problem, where the optimal partition is sought, subject to the constraint that the normalized cut values of any two nodes that are preassigned to the same class should be the same.

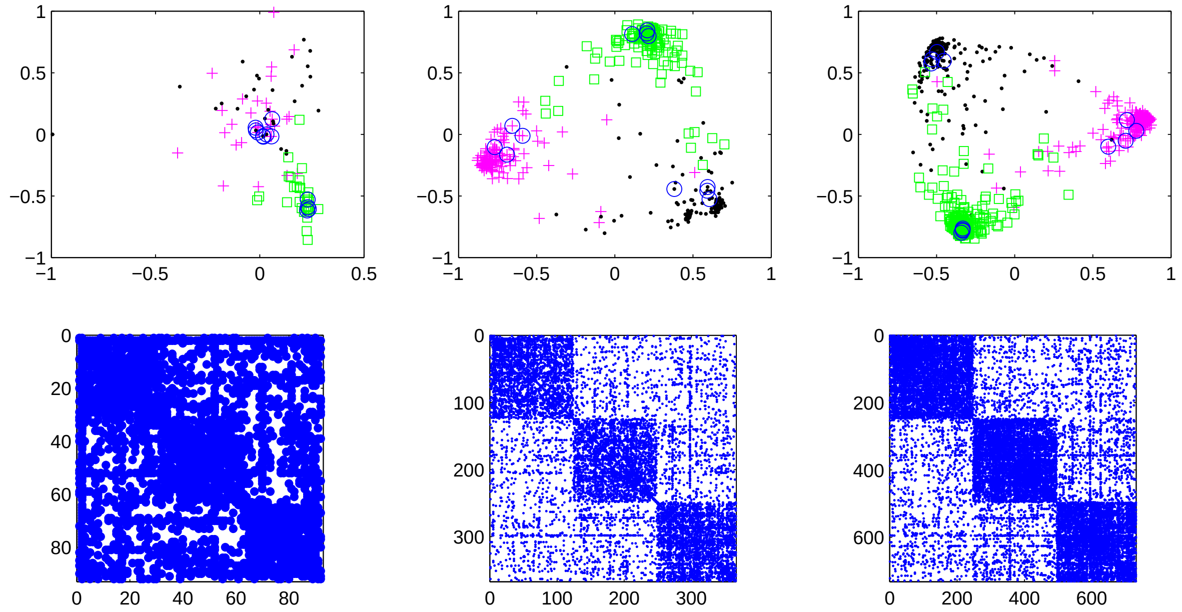
The main drawback with the algorithm in [Yu and Shi, 2001] is that it only constrains same-labeled data points. The algorithm we present here benefits from the zeros in the sparsity pattern introduced by differently-labeled pairs. Furthermore, it should be noted that, in the multi-class case, labeled sets combinatorialy tend to embody more differently-labeled pairs than same-labeled pairs. The other drawback to not using the information given by differently-labeled points is that the trivial partition (all points in one cluster) will satisfy the constraints, even when many points are labeled. In fact, when all the data is labeled, it is likely that the partition found by the Grouping with Bias algorithm will be the trivial partition. Figure 6(a) shows that our Spectral Classifier outperforms the Grouping with Bias algorithm for the 3 Newsgroups data set. In fact, Grouping with Bias started performing slightly worse when a large fraction of the data was labeled.

[5] For each document, we take the most similar 20 documents, and put those similarities in the appropriate row and column. All other entries are 0.



1% Labeled 50% Labeled 100% Labeled

Figure 4: Three classes of the 20 NEWSGROUPS data set in spectral space with increasing mounts of *labeled* data. The classes are sci.crypt (pluses), talk.politics.mideast (dots), and soc.religion.christian (squares). The labeled points are circled. The bottom graphs show the sparsity patterns of the associated affinity matrices.



92 Unlabeled Docs 366 Unlabeled Docs 732 Unlabeled Docs

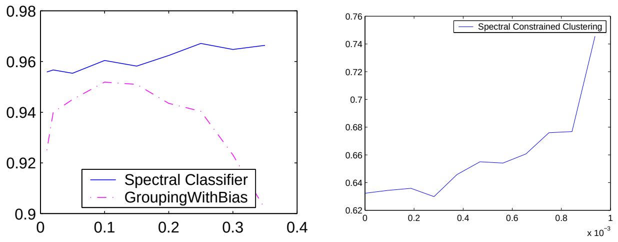
Figure 5: Three classes of the 20 NEWSGROUPS data set in spectral space with increasing amounts of *unlabeled* data. The classes are sci.crypt (pluses), talk.politics.mideast (dots), and soc.religion.christian (squares). There are 12 labeled documents (circled).

5 Constrained Spectral Clustering

Recently, there has been interest in *constrained clustering* [Wagstaff and Cardie, 2000; Klein *et al.*, 2002], which involves clustering with two types of pairwise constraints:

1. *Must-links*: two items are known to be in the same class.
2. *Cannot-links*: two items are in different classes.

Constrained clustering allows one to do exploratory data analysis when one has some prior knowledge, but not class labels. The classifier presented in section 4 can be easily modi-



(a) (b)

Figure 6: (a) Classification accuracy vs. fraction of pairs constrained for spectral classifier vs. grouping with bias. (b) Rand Index of spectral constrained clustering vs. fraction of pairs constrained (shown up to 0.1% constrained). Both are on 3 NEWSGROUPS data.

fied for this kind of prior knowledge: we have been reducing labeling information to pairwise information during affinity modification all along.

Specifically, the affinity matrix is now defined as follows:

1. Define the affinity matrix A as before.
2. For each pair of must-linked points (i, j) assign the values $A_{ij} = A_{ji} = 1$.
3. For each pair of cannot-linked points (i, j) that are in different classes, assign the values $A_{ij} = A_{ji} = 0$.
4. Normalize $N = \frac{1}{d_{max}}(A + d_{max}I - D)$.

This is equivalent to the *imposing constraints* step in [Klein *et al.*, 2002]. In this step, must-linked points are made more similar than any other pair of points in the data set, and cannot-linked points are made more dissimilar than any pair of points in the data set.

The spectral constrained clustering algorithm proceeds just as the other spectral clustering algorithms presented here; the only difference is that it uses the modified normalized affinity matrix presented in this section.

Figure 6(b) shows a plot of accuracy vs. number of constraints for the 3 NEWSGROUPS data set using spectral constrained clustering. The accuracy is measured by the constrained Rand index [Klein *et al.*, 2002; Wagstaff and Cardie, 2000]. The accuracy increases with number of constraints, showing that the spectral constrained clustering can effectively use constraints to better cluster data.

6 Conclusion

We present here a probabilistic model and an associated spectral learning algorithm that is able to work for the unsupervised, semi-supervised, and fully-supervised learning problems. We show that this algorithm is able to cluster well, and further is able to effectively utilize prior knowledge, either given by pairwise constraints or by labeled examples.

Acknowledgments

This paper is based on research supported in part by the National Science Foundation under Grant No. IIS-0085896, and by the Research Collaboration between NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation and CSLI, Stanford University (research project on Concept Bases for Lexical Acquisition and Intelligently Reasoning with Meaning).

References

[Alizadeh, 2000] A.A. Alizadeh. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–11, 2000.

[Brew and Schulte im Walde, 2002] C. Brew and S. Schulte im Walde. Spectral clustering for german verbs. In *Proceedings of EMNLP-2002*, 2002.

[Chung, 1997] F. Chung. *Spectral Graph Theory*. AMS, Providence, RI, 1997.

[Deerwester *et al.*, 1990] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[Fiedler, 1975] M. Fiedler. A property of eigenvectors of non-negative symmetric matrices and its application to graph theory. *Czeckoslovak Mathematical Journal*, 25:619–672, 1975.

[Hubert and Arabie, 1985] L. J. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[Klein *et al.*, 2002] Dan Klein, Sepandar D. Kamvar, and Christopher D. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *The Nineteenth International Conference on Machine Learning*, 2002.

[Kleinberg, 1998] J. Kleinberg. Authoritive sources in a hyperlinked environment. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[Meila and Shi, 2000] Marina Meila and Jianbo Shi. Learning segmentation by random walks. In *Advances in Neural Information Processing Systems*, pages 873–879, 2000.

[Meilă and Shi, 2001] Marina Meilă and Jianbo Shi. A random walks view of spectral segmentation. In *AI and Statistics (AISTATS)*, 2001.

[Ng *et al.*, 2002] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Proceedings of Advances in Neural Information Processing Systems (NIPS 14)*, 2002.

[Page *et al.*, 1998] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[Salton, 1989] G. Salton. *Automatic Text Processing: the transfomation, analysis, and retrieval of information by computer*. Addison-Wesley, 1989.

[Wagstaff and Cardie, 2000] Kiri Wagstaff and Claire Cardie. Clustering with instance-level constraints. In *The Seventeenth International Conference on Machine Learning*, pages 1103–1110, 2000.

[Yu and Shi, 2001] Stella Yu and Jianbo Shi. Grouping with bias. Technical Report CMU-RI-TR-01-22, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2001.