Extracting Community Structure Features for Hypertext Classification

Dell Zhang
SCSIS
Birkbeck, University of London
London WC1E 7HX, UK
dell.z@ieee.org

Robert Mao
Microsoft Corp.
EPDC5/2352, South County Business Park
Leopardstown, Dublin, Ireland
robmao@microsoft.com

Abstract

Standard text classification techniques assume that all documents are independent and identically distributed (i.i.d.). However, hypertext documents such as web pages are interconnected with links. How to take advantage of such links as extra evidences to enhance automatic classification of hypertext documents is a non-trivial problem. We think a collection of interconnected hypertext documents can be considered as a complex network, and the underlying community structure of such a document network contains valuable clues about the right classification of documents. This paper introduces a new technique, Modularity Eigenmap, that can effectively extract community structure features from the document network which is induced from document link information only or constructed by combining both document content and document link information. A number of experiments on real-world benchmark datasets show that the proposed approach leads to excellent classification performance in comparison with the state-of-the-art methods.

1 Introduction

Automatic text classification or categorization is a fundamental problem in information retrieval and organization [9]. Standard text classification techniques assume that all documents are independent and identically distributed (i.i.d.). However, hypertext documents such as web pages are interconnected with links. Although we can easily employ standard text classification techniques for the problem of hypertext classification, we anticipate that a better classification performance could be achieved by exploiting additional document link information.

It is usually not promising to use the raw links directly as features. Due to the sparsity of document network, two documents in the same class may appear to be dissimilar on surface when they are not directly connected, but such a pair of documents are likely to be connected indirectly by some paths with a number of intermediate documents. In other words, it is important to exploit not only local link information but also global structure information for effective hypertext classification.

We think a collection of interconnected hypertext documents can be considered as a *complex network* [7], and the underlying *community structure* [8] of such a document network contains valuable clues about the right classification of documents. A number of graph-based classification methods have emerged in recent years (see Section 2). However, they are not really suitable for complex networks because they do not take the degree distribution of network into consideration. This paper introduces a new technique, Modularity Eigenmap, that can effectively extract community structure features from the document network for enhanced hypertext classification.

2 Related Work

Some hypertext classification methods rely solely on document link information. Previous research studies have shown that using raw links directly as features does not work well in practice [2, 12]. Inspired by the success of PageRank and HITS in Web search and mining, people have tried to apply link analysis techniques for hypertext classification. Gyöngyi et al. propose a technique similar to Personalised PageRank for large-scale web page classification [5]. Zhou et al. propose a technique in the semi-supervised learning framework, Directed Graph Regularization (DGR), that combines the 'hub' and 'authority' information of web pages for their classification [11].

Some hypertext classification methods make use of both document content and document link information. The simplest method is to expand each document's feature vector by incorporating the features of its neighbours (directly linked documents), but this method seems to have difficulties with parameter tuning and often does not provide a robust solution [2]. Chakrabarti et al. propose to address the hypertext

classification problem in the Markov Random Fields framework by using the iterative Relaxation Labelling algorithm [2]. Cohn and Hofmann propose to address the hypertext classification problem by constructing a latent space of both document content and document link information [4] where probabilistic LSI (PLSI) is used for content analysis and probabilistic HITS (PHITS) is used for link analysis. Recently Zhu et al. propose to use the technique Matrix Factorization (MF) to find latent factors from both document content matrix and document link matrix for the task of hypertext classification, and they also extend it to Supervised Matrix Factorization (SupMF) by taking document label information into account as well [12].

3 Approach

We consider a collection of interconnected hypertext documents as a *complex network* [7]. As in most real-world complex networks, nodes (documents) in such a document network tend to divide into communities, with a high density of edges within communities and a low density of edges between them. Therefore the underlying *community structure* [8] of such a document network must contain valuable clues about the right classification of documents. In this paper, we attempt to extract community structure features from document network by finding a low-dimensional representation for the community structure of documents.

Given a complex network (graph) G with n nodes (documents) and m edges, we describe it using its $n \times n$ adjacency matrix $\mathbf{A}$ with elements A_{ij} representing the number (or weight) of edges between node i and node j. Since real-world complex networks (including document networks) are usually sparse, most elements in $\mathbf{A}$ should be 0. In this paper we focus on undirected networks, so $\mathbf{A}$ is symmetric. The degree of a node i, i.e., the number of edges connected to a node i, is given by $k_i = \sum_j A_{ij}$. Define $\mathbf{D} = \mathrm{diag}(k_1, \ldots, k_n)$.

Consider the division of a network into c non-overlapping communities, and define an $n \times c$ indicator matrix $\mathbf{F} = (\mathbf{f}_1|\mathbf{f}_2|\ldots|\mathbf{f}_c)$ with one column as the binary indicator vector for each community, such that $F_{ij} = 1$ if node i belongs to community j and $F_{ij} = 0$ otherwise. It is easy to see that the columns of $\mathbf{F}$ are mutually orthogonal, that each row of $\mathbf{F}$ sums to 1, and that $\mathrm{Tr}(\mathbf{F}^T\mathbf{F}) = n$. Since our purpose is to extract community structure features, we can discard the constraint that the elements of $\mathbf{F}$ is binary but allow them to take any real value. Thus a choice of c communities would be equivalent to choosing $d = c - 1$ independent, mutually-orthogonal columns $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_d$ (because only $c - 1$ of the columns are independent with the last being fixed by the condition that each row of $\mathbf{F}$ sums to 1). The value of the i-th element in the j-th leading eigenvector $\mathbf{f}_j$ indicates the "strength" with which node i belongs to the j-th community, sort of a fuzzy membership function. Furthermore, let g_i denote the group to which vertex $\mathbf{x}_i$ belongs. The function $\delta(g_i, g_j) = 1$ if $g_i = g_j$ and $\delta(g_i, g_j) = 0$ otherwise. It can be shown that $\delta(g_i, g_j) = \sum_{k=1}^{c} F_{ik}F_{jk}$.

3.1 Laplacian Eigenmap

A problem closely related to community discovery is *graph partitioning* that has been studied extensively in computer science for a variety of applications. One of the most widely used graph partitioning methods is spectral partitioning that attempts to minimise the *cut-size*, i.e., the number of edges running between different groups of nodes: $S = \frac{1}{2}\sum_{i,j} A_{ij}(1 - \delta(g_i, g_j))$. We can rewrite the cut-size for this division of network in matrix form as $S = \frac{1}{2}\mathrm{Tr}(\mathbf{F}^T\mathbf{L}\mathbf{F})$, where $\mathbf{L}$ is the *Laplacian matrix* [3] defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$. It is known from linear algebra that S would be minimised by choosing the $d = c - 1$ eigenvectors of $\mathbf{L}$ corresponding to the d smallest eigenvalues.

All eigenvalues of $\mathbf{L}$ are non-negative, and the graph G has z connected components if and only if $\mathbf{L}$ has z zero eigenvalues with corresponding eigenvectors being piecewise constant on the connected components [3]. Without loss of generality, suppose that the d smallest *positive* eigenvalues of $\mathbf{L}$ in increasing order are $0 < \lambda_1 \leq \lambda_2 \ldots \leq \lambda_d$ and then the corresponding bottom eigenvectors $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_d$ can be used as the d extracted features. We call this feature extraction method Laplacia Eigenmap (LapEig). It is essentially embedding the network structure in a low-dimensional space [1].

The *normalised Laplacian* [3] that has many attractive theoretical properties $\tilde{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$ can also be used in the above formulas. Laplacian Eigenmap using $\tilde{\mathbf{L}}$ instead of $\mathbf{L}$ provides better classification performance in our experiments, so the reported experimental results of Laplacian Eigenmap (in Section 4) are all based on $\tilde{\mathbf{L}}$.

The idea of Laplacian Eigenmap has been proposed before by Belkin et al. [1] from the perspective of dimensionality reduction or manifold learning as well as semi-supervised learning, but to our knowledge it has not been applied to hypertext classification yet.

Despite its success in the partitioning of simple graphs (such as k-nearest-neighbours graphs), Laplacian based spectral partitioning is poor in detecting natural communities in real-world complex networks. The fundamental problem of using this technique for community discovery is that cut sizes are not really the right thing to optimise because they don't accurately reflect our intuitive concept of network communities [8]. A good division of a network into communities is not merely one in which the number of edges running across communities is small. Rather, it is one in which the number of edges across communities is

smaller than expected. It has been reported that Laplacian based spectral graph partitioning often fails to find the right division of a complex network [8]. Consequently the effectiveness of Laplacian Eigenmap for extracting community structure features would be limited.

3.2 Modularity Eigenmap

One proven-effective approach to community discovery is maximising the quality function known as *modularity* [8] over the possible divisions of a network: $Q = \frac{1}{2m}\sum_{ij}[A_{ij} - P_{ij}]\delta(g_i, g_j)$, where $P_{ij} = (k_i k_j)/(2m)$. In fact, P_{ij} is the *expected* number of edges between node i and node j in the 'null model' — a *random* graph with the same degree distribution as the given network. Optimising modularity reflects our intuition that the number of edges within communities should be higher than expected by chance. Only if the number of within-community edges ($\frac{1}{2}\sum_{ij} A_{ij}\delta(g_i, g_j)$) is significantly higher than it would be expected purely by chance ($\frac{1}{2}\sum_{ij} P_{ij}\delta(g_i, g_j)$) can we justifiably claim to have found significant community structure. Maximising modularity has been shown to produce excellent community discovery results in standardised tests. We can rewrite the modularity for this division of network in matrix form as $Q = \frac{1}{2m}\mathrm{Tr}(\mathbf{F}^T\mathbf{M}\mathbf{F})$, where $\mathbf{M}$ is the *modularity matrix*[1] [8] defined as $\mathbf{M} = \mathbf{A} - \mathbf{P}$. It is known from linear algebra that Q would be maximised by choosing the $d = c - 1$ leading eigenvectors of $\mathbf{M}$ corresponding to the d largest positive eigenvalues.

According to the above analysis, we propose the Modularity Eigenmap (ModEig) algorithm to extract community structure features from the network, as shown in Figure 1.

Only eigenvectors corresponding to positive eigenvalues can give positive contributions to the modularity, so Modularity Eigenmap can extract at most p community structure features if the modularity matrix has p positive eigenvalues. In practice, a small number $d \ll p$ of community structure features would be enough to achieve a good hypertext classification performance (see Section 4).

3.3 Combining Content and Link

The proposed Modularity Eigenmap technique can be used straightforwardly to extract community structure features from the document network induced by the links among documents. However, it should be beneficial to exploit both document content information and document link information for hypertext classification. One simple method to achieve this goal is to create hybrid feature vectors by concatenating content-based document feature vectors and corresponding link-based document feature vectors (such as those consist of community structure features extracted by Modularity Eigenmap). In this paper we take an alternative approach that considers document content and document link information all in the framework of Modularity Eigenmap: we construct a hybrid document network which is actually the sum of two document networks, one derived from document content similarity and the other induced by the links among documents, and then use Modularity Eigenmap to extract community structure features from the hybrid document network. When deriving the content-based document network, we simply connect each pair of documents by an edge that is weighted by their content similarity. We have found through experiments that such a graph combination method is more stable and robust than the feature combination method, but a detailed discussion on this issue is omitted as it is not the focus of this paper. Although it is possible to assign different weights to content-based document network and link-based document network respectively when combining them into the hybrid document network, we just use equal weights in our experiments (see Section 4). To distinguish the usage of Modularity Eigenmap on the pure link-based document network and that on the combined document network, we denote the former ModEig (link) and the latter ModEig (content+link).

[1] We have extended the original definition of modularity matrix [8] to weighted networks here.

Input: The $n \times n$ adjacency matrix $\mathbf{A}$ and the desired dimension of feature space d.

Output: The d-dimensional feature vectors representing the n nodes.

- Construct the modularity matrix $\mathbf{M} = \mathbf{A} - \mathbf{P}$;
- Solve the eigenvalue problem $\mathbf{M}\mathbf{f} = \lambda\mathbf{f}$;
- Choose the d leading eigenvectors $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_d$ corresponding to the d largest positive eigenvalues $\lambda_1 \geq \lambda_2 \ldots \geq \lambda_d > 0$;
- Return the i-th row of the matrix $(\mathbf{f}_1|\mathbf{f}_2|\ldots|\mathbf{f}_d)$ as the community structure feature vector for the node i.

Figure 1. ModEig algorithm.

3.4 Implementation

The Modularity Eigenmap technique requires computation of the d leading eigenvectors of the modularity matrix corresponding to the largest positive eigenvalues. The Lanczos method demands $\mathrm{O}(n^3)$ time to find all eigenvectors of a matrix, but there are variants of the Lanczos method as well as other completely different methods that

can find just a few leading vectors faster. Since as shown in our experiments (Section 4), a small number of community structure features would be enough to achieve excellent hypertext classification accuracy, therefore the algorithm of Modularity Eigenmap can run in time $O(n^2)$ for a sparse document network [8].

4 Experiments

4.1 Datasets

We conduct our experiments on two collections of real-world datasets, WebKB[2] and Cora[3].

The WebKB datasets consist of about 6,000 web pages from the computer science departments of four universities: *Cornell, Texas, Washington and Wisconsin*. The web pages are classified into categories such as 'course', 'faculty' and 'student'. For each dataset, we use the *co-citation graph* derived from the original directed hyperlinks (citations) as the link-based document network for our algorithms, i.e., two pages (nodes) are connected by an edge if there is a third page linking to or being linked to both of them, and multiple edges are allowed between one pair of nodes. This is because for Web data co-citations are usually more reliable than hyperlinks which has been reported in past research studies [10] and also confirmed by our experiments.

The Cora datasets consist of the abstracts and references of about 34,000 computer science papers. In our experiments, we use only the papers in four research areas, *Data Structure (DS), Hardware and Architecture (HA), Machine Learning (ML) and Programming Language (PL)*, and we discard the papers without reference to other papers in the same area. The papers are classified according to their subfields in the research area. For each dataset, we use the *undirected graph* derived from the original directed references as the link-based document network for our algorithms, i.e., two papers (nodes) are connected by an edge if one of them cites the other or vice versa, and multiple edges are allowed between one pair of nodes.

4.2 Settings

We perform our hypertext classification experiments on each of the above datasets by first extracting community structure features from the document network and then using linear Support Vector Machine (SVM) [6] to classify the documents represented as feature vectors[4]. The implementation of linear SVM used in our experiments is LIBSVM[5].

[2]http://www.cs.cmu.edu/∼webkb/

[3]http://www.cs.umass.edu/∼mccallum/code-data.html

[4]Each document feature vector is normalised to unit length.

[5]http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

We measure classification performance by 5-fold cross-validation accuracy, i.e., the dataset is randomly split into five equal-size folds and the classification experiment is repeated for five times, each time with one fold for test and four other folds for training. For experiments with our proposed Modularity Eigenmap technique, we do not tune the SVM parameters but simply take their default values. For experiments using other methods, the parameters are tuned through cross-validation on the training folds data, as in [12].

We compare our proposed approach with the state-of-the-arts methods for hypertext classification, some using only document link information while some using both document content and document link information. When using document content information, each document is regarded as a bag-of-words and pre-processed by TF×IDF weighting. The methods being compared are briefly described as follows.

- **DGR (link)** refers to Directed Graph Regularization [11] that is only based on document link information (see Section 2).
- **LapEig (link)** refers to SVM using community structure features extracted by Laplacian Eigenmap from pure link-based document network (see Section 3). The reported results are from the experiments using 30 features.
- **ModEig (link)** refers to SVM using community structure features extracted by Modularity Eigenmap from pure link-based document network (see Section 3). The reported results are from the experiments using 30 features unless otherwise noted.
- **PLSI+PHITS (content+link)** refers to probabilistic LSI plus probabilistic HITS [4] that exploits both document content and document link information (see Section 2).
- **MF (content+link)** refers to SVM using features extracted by Matrix Factorization [12] that exploits both document content and document link information (see Section 2). The SVM regularization parameter '-c ' is tuned on the training folds data via cross-validation. The reported results are from the experiments using 50 features.
- **SupMF (content+link+label)** refers to Supervised Matrix Factorization [12] that takes into account document label information in addition to document content and document link information (see Section 2). The SVM regularization parameter '-c ' is tuned on the training folds data via cross-validation. The reported results are from the experiments using 50 features.

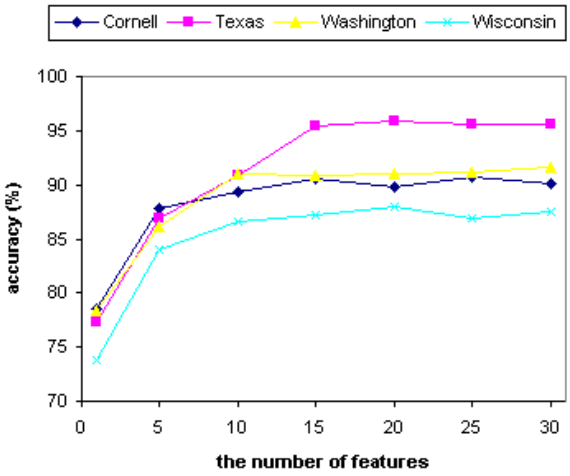
- **ModEig (content+link)** refers to SVM using community structure features extracted by Modularity Eigenmap from hybrid document network fusing content and link together (see Section 3). The reported results are from the experiments using 30 features unless otherwise noted.

4.3 Results

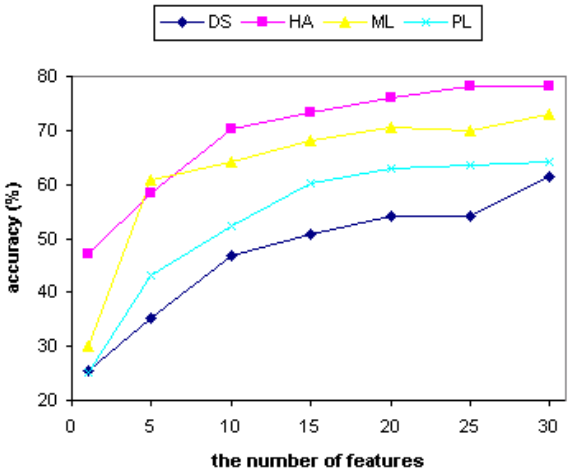
We show the experimental results of ModEig (link) with varying number of features on the WebKB datasets and the Cora datasets in Figure 2(a) and 2(b) respectively. With just a few number ($\leq$ 30) of features, ModEig is able to effectively capture the community structure of document network and lead to a good hypertext classification accuracy. As the number of community structure features increases, the hypertext classification accuracy first increases quickly and then reaches a plateau. In the rest of the paper, only 30 community structure features are used when we talk about the experimental results of ModEig.

We show the experimental results of competing methods on the WebKB datasets in Figure 3(a). When using only document link information, ModEig works significantly better than LapEig on all four datasets, and outperforms DGR on three of the four datasets. When using both document content and document link information, ModEig works significantly better than PLSI+PHITS, and shows similar performance as MF or SupMF. The differences between ModEig results and MF or SupMF results are not significant, but note that for ModEig (1) we use only 30 features instead of 50; (2) we simply combine content-based document network and link-based document network with equal weights; (3) we do not tune the SVM parameters; and (4) we do not make use of document label information, therefore ModEig still has advantages over MF and SupMF.

We show the experimental results of competing methods on the Cora datasets in Figure 3(b). When using only document link information, ModEig works significantly better than LapEig and DGR on all four datasets. When using both document content and document link information, ModEig works significantly better than PLSI+PHITS, MF and SupMF, demonstrating clear advantages of community structure features.



(a) WebKB



(b) Cora

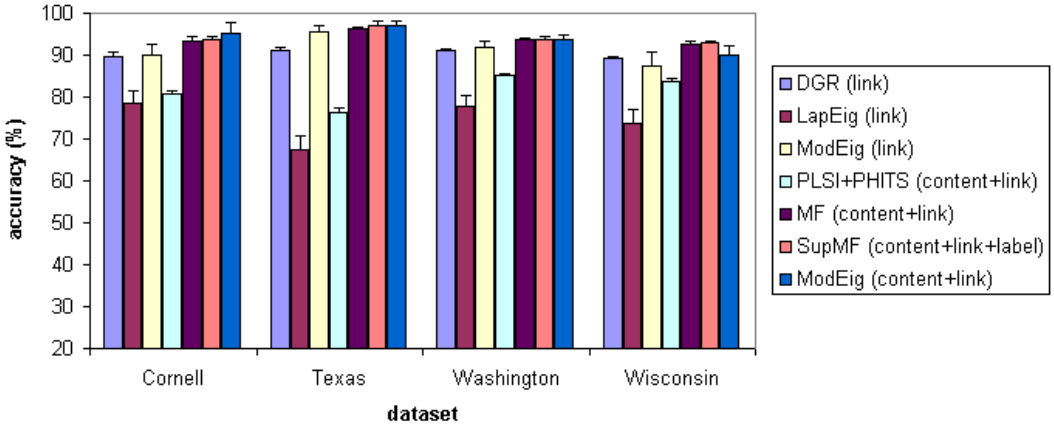
Figure 2. ModEig (link) experimental results.

5 Conclusions

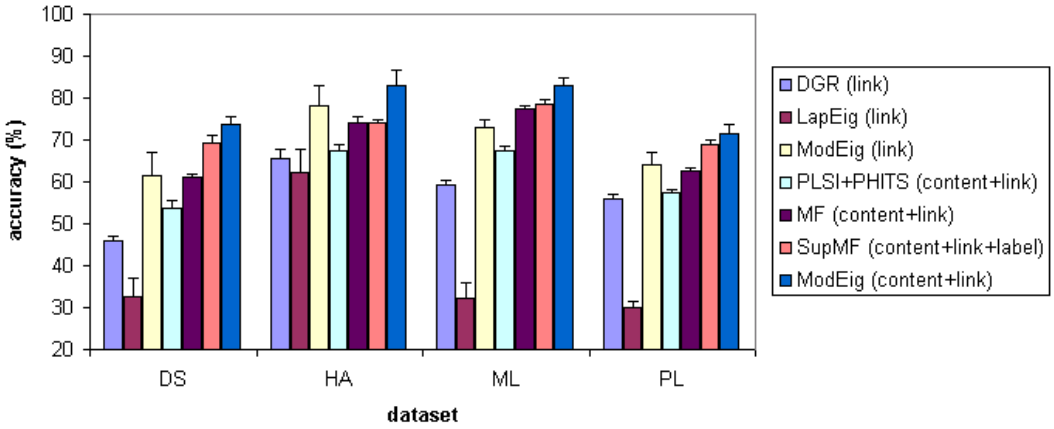
In this paper, we address the problem of hypertext classification from the feature extraction perspective. Specifically we propose a new technique, Modularity Eigenmap, that can effectively extract community structure features from the document network for enhanced hypertext classification. A number of experiments on real-world benchmark datasets show that the proposed approach leads to excellent classification performance in comparison with the state-of-the-art methods. Our feature extraction approach has the advantages of being *algorithm-independent* and *application-independent*: the community structure features extracted with Modularity Eigenmap can be used by any general-purpose machine learning algorithm (such as SVMs and AdaBoost) and many other applications in addition to hypertext classification (such as clustering and learning to rank). Moreover, the technique of Modularity Eigenmap is not necessarily restricted to hypertext documents — it can also be applied to any networked data.

6 Acknowledgements

We would like to thank Dr. Shenghuo Zhu for sharing his pre-processed datasets. Thanks also to the anonymous reviewers for their helpful comments.



(a) WebKB



(b) Cora

Figure 3. Comparison of classification accuracy (mean $\pm$ std %).

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[2] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 307–318, Seattle, WA, 1998.

[3] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[4] D. A. Cohn and T. Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In *Advances in Neural Information Processing Systems (NIPS)*, pages 430–436, Denver, CO, USA, 2000.

[5] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Web content categorization using link information. Technical report, Stanford University, 2007.

[6] T. Joachims. *Learning to Classify Text using Support Vector Machines*. Kluwer, 2002.

[7] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[8] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74:036104, 2006.

[9] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[10] T. Zhang, A. Popescul, and B. Dom. Linear prediction models with graph regularization for web-page categorization. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 821–826, Philadelphia, PA, 2006.

[11] D. Zhou, J. Huang, and B. Scholkopf. Learning from labeled and unlabeled data on a directed graph. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pages 1036–1043, Bonn, Germany, 2005.

[12] S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classification using matrix factorization. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 487–494, Amsterdam, The Netherlands, 2007.