A Random Walks Method for Text Classification

YUNPENG XU[1], XING YI[2], CHANGSHUI ZHANG[1]

[1]State Key Laboratory of Intelligent Technologies and Systems,
Tsinghua University, 100084 Beijing, P.R. China

[2]Center for Intelligent Information Retrieval, Department of Computer Science,
University of Massachusetts, Amherst, MA01003, USA

E-MAIL:xuyp03@mails.tsinghua.edu.cn, zcs@mail.tsinghua.edu.cn, yixing@cs.umass.edu

Abstract

Practical text classification system should be able to utilize information from both expensive labelled documents and large volumes of cheap unlabelled documents. It should also easily deal with newly input samples. In this paper, we propose a random walks method for text classification, in which the classification problem is formulated as solving the absorption probabilities of Markov random walks on a weighted graph. Then the Laplacian operator for asymmetric graphs is derived and utilized for asymmetric transition matrix. We also develop an induction algorithm for the newly input documents based on the random walks method. Meanwhile, to make full use of text information, a difference measure for text data based on language model and KL-divergence is proposed, as well as a new smoothing technique for it. Finally an algorithm for elimination of ambiguous states is proposed to address the problem of noisy data. Experiments on two well-known data sets: $WebKB$ and $20Newsgroup$ demonstrate the effectivity of the proposed random walks method.

1 Introduction

Text classification, an important research area in data mining and knowledge discovery, has broad applications such as web information mining, junk mails filtering, personal news recommendation system, document databases organization, etc. [10] [11] are referred to as two comprehensive surveys on text classification problems. Considering the fact that labelling documents is both costly and time-consuming, we will follow the *Semi-Supervised Learning* approach where there is a small number of labelled documents and large volumes of unlabelled documents. Over the past few years, many semi-supervised algorithms have been proposed with application to different problems. A comprehensive survey up to 2001 can be found in [12], or a more recent one in [20].

In a practical text classification system, many things need to be carefully considered such as how to design efficient classification algorithms, how to extract as much information from the text as possible, how to efficiently deal with new coming documents, etc. In this paper, we propose a random walks method for text classification, which consists of two aspects: random walks model and graph representation for text. Random walks model formulates the classification as a problem of solving the absorption probability of random walks on a weighted graph. More specifically, labeled documents are viewed as the absorbing states while unlabeled documents as the non-absorbing states. Then the predicted labels of unlabeled documents are determined by the probabilities that they are absorbed by different absorbing states. In this method, the Laplacian operator for asymmetric graphs is derived so that asymmetric transition matrix can be used for learning. Then an induction algorithm is developed for dealing with new documents based on this method. In the usual graph representation of text data, each document is represented as a vertex on the graph, in which the most important information is the distance between two vertices. Then in this paper a new distance measure for the text data based on language model [8] [13] and KL-divergence is defined in order to better capture the difference between documents. In the end we discuss the smoothing technique and also the elimination algorithm of ambiguous states to address the problem of learning with noisy data.

The paper is organized as follows. Section 2 introduces the random walks model. Section 3 describes the graph representation for text. Experimental results are shown in Section 4. Section 5 concludes and discusses our future work.

2 Random Walks Model

Firstly, some terminologies and notations are given. A **graph** is a pair $G =< V, E >$ which consists of vertices $v \in V$ and edges $e \in E \subseteq V \times V$. A **finite states graph** is a graph with finite vertices. A graph is **weighted**, if each edge is associated with a value $w_{ij} \in R$, which does not necessarily satisfy $w_{ij} = w_{ji}$. The **degree** d_i of

vertex v_i is $d_i = \sum_j w_{ij}$, i.e., the summation of weights of all edges that are connected to v_i. The **Laplacian Matrix** of the graph is defined to be $L = D - W$ [7], where W is the weight matrix, D is a diagonal matrix, whose i^{th} diagonal element is d_i.

2.1 Random Walks for Semi-supervised learning

Random walks is a stochastic model on a weighted finite states graph, which exploits the structure of data in a probabilistic way. Here, we consider its application in semi-supervised classification problems. Given such a graph, suppose there are l labeled vertices $V^L = \{v_1, v_2, ..., v_l\}$ with labels f^L, and u unlabeled vertices $V^U = \{v_{l+1}, v_{l+2}, ..., v_{l+u}\}$, the task is to predict the labels f^U of V^U. f^L and f^U are from the same label set $G = \{g_1, g_2, ..., g_K\}$. The random walks approach to this problem is direct and natural: it regards all the vertices on the graph as different states of a Markov chain. The one-step transition probability p_{ij} from v_i to v_j is defined by

$$p_{ij} = \frac{w_{ij}}{d_i}, \tag{2.1}$$

or written in the matrix form

$$P = D^{-1}W. \tag{2.2}$$

The label f_i of the vertex $v_i \in V^U$ is then determined by the probabilities that it reaches different labeled vertices, i.e., if the summation of the probabilities that it reaches the vertices with label g_k is the largest, it is assigned label g_k.

In this paper, we consider a Markov chain with absorbing states, which assigns different roles for labeled and unlabeled data respectively. More specifically, the labeled samples are viewed as the absorbing states while the unlabeled samples as the non-absorbing states, which means the former will only transit to themselves with probability 1, whereas the latter are free to transit to any other states. Therefore once the unlabeled samples reach the absorbing states, they will be absorbed and will not move on. Then the label f_i of the vertex $v_i \in V^U$ is determined by the probabilities that it is absorbed by different labeled vertices in equilibrium.

Let P be the one-step transition matrix of the chain. For convenience, we partition the matrix according to the label status of the vertices as

$$P = \begin{pmatrix} I & O \\ P_{UL} & P_{UU} \end{pmatrix}, \tag{2.3}$$

where I is an $l \times l$ identical matrix, O is an $l \times u$ matrix with all 0 elements, P_{UL} is the transition probability that non-absorbing states transit to absorbing states, P_{UU} is the transition probability for non-absorbing states. It is known [6] that the absorbing probabilities that non-absorbing states are absorbed by absorbing states in equilibrium is given by

$$\begin{aligned} P_r(V^L|V^U) &= (I - P_{UU}^{-1})P_{UL} \\ &= L_{UU}^{-1}W_{UL} \end{aligned} \tag{2.4}$$

where L_{UU} and W_{UL} are corresponding blocks of L and W. Therefore, the probabilities of the unlabeled samples being absorbed by the labeled sample set A_k with label g_k is

$$P_r(A^k|V^U) = L_{UU}^{-1}W_{UL}f_k, \tag{2.5}$$

where f_k is an $l \times 1$ vector which satisfies $f_{k,i} = \begin{cases} 1 & f_i^L = k \\ 0 & f_i^L \neq k \end{cases}$. Then the labels of the unlabeled samples are given by

$$f^U = \arg\max_k \{P_r(A^k|V^U)\}. \tag{2.6}$$

2.2 Review of Related Work

The idea of applying random walks model to semi-supervised learning is not new. In [14], Szummer adopted a Markov random walks model without absorbing states. The model assumes that for each data there exists a label distribution, the parameters of which are then estimated by EM algorithm. However, there are three major differences between [14] and the model used here. First, [14] has no absorbing states. Second, [14] considers the situation after a number of transitions, therefore crucially depends on the choice of the time parameter. Instead we consider the chain in an equilibrium state. Third, the values of the labels on the labeled data in [14] are not fixed, but here the labeled data are regarded as absorbing states and therefore have fixed values. [19] proposed a regularization framework based on Gaussian field and Harmonic functions, which is closer to our approach. They got the same label function as formula 2.5 but with different interpretation, which is not surprising since the estimation of data labels using random walks can also be regarded as a *Dirichlet Problem* to find a harmonic functions having the boundary values[6]. However, our random walks interpretation has a better formulation of the problem, and is easier to be generalized to multi-class problems and out-of-sample problems. In [6], Doyle also introduced the interesting connection between random walks and electric networks.

2.3 Laplacian Operator for Asymmetric Graphs

In a real problem, the transition matrix P of random walks does not necessarily satisfy symmetry, i.e., $p_{ij} = p_{ji}$ might be violated. Correspondingly, the graph weights do not necessarily satisfy $w_{ij} = w_{ji}$. Graphs with asymmetric weights are termed *Asymmetric Graphs*. However, for learning problem asymmetry in formula 2.5 is a trouble, since L_{UU} defined in symmetric way might not be semi-definite, which would probably have negative eigenvalues and therefore produce negative absorbing probabilities. To address this problem, we will show in the following that asymmetric graphs still have Laplacian matrices in symmetric form. Our proof is given in the framework of graph analysis[18].

THEOREM 2.1. *The Laplacian matrix for asymmetric graph is given by $L = D - M$, where $M_{ij} = \sqrt{W_{ij}W_{ji}}$.*

Proof. We start the proof from the definitions of some basic operators on a discrete graph. Consider two arbitrary vertices $v_i, v_j \in V$, and the edge $e_{ij} \in E$ between them. Let $\mathcal{H}(V)$ be the Hilbert space of real-value function endowed with the usual inner product $< \varphi, \phi >= \sum_v \varphi(v)\phi(v)$, where φ and ϕ are two arbitrary functions in $\mathcal{H}(V)$. Similarly, we define $\mathcal{H}(E)$ and let ψ be an arbitrary function in it.

In graph analysis, the *graph gradient* is defined to be

$$\nabla\varphi(v_i, v_j) = \sqrt{w_{ij}}\varphi(v_i) - \sqrt{w_{ji}}\varphi(v_j) \tag{2.7}$$

for both symmetric and asymmetric graphs. It is clear that $\nabla\varphi(v_i, v_j) = -\nabla\varphi(v_j, v_i)$ always holds. The *graph divergence* $(div\psi)(v_i)$ measures the net outflow of function at vertex v_i and is defined by the following adjoint operator

$$< \nabla\varphi, \psi >=< \varphi, div\psi > . \tag{2.8}$$

Plugging the expression for graph gradient into formula 2.8 and expand the latter, we have

$$(div\psi)(v_i) = \sum\nolimits_j \sqrt{w_{ij}}(\psi_{ij} - \psi_{ji}), \tag{2.9}$$

where the summation is over all v_j that are connected to v_i. By virtue of the definition of Laplacian operator

$$\Delta\varphi = div(\nabla\varphi) \tag{2.10}$$

and substitute the expressions for graph divergence and graph gradient, it follows that

$$(\Delta\varphi)(v_i) = (d_i\varphi(v_i) - \sum\nolimits_j \sqrt{w_{ij}w_{ji}}\varphi(v_j)). \tag{2.11}$$

Directly from formula 2.11, we obtain the Laplacian matrix for asymmetric graph

$$L = D - M, \tag{2.12}$$

where $M_{ij} = \sqrt{W_{ij}W_{ji}}$. □

Formula 2.12 demonstrates that the Laplacian matrix of an asymmetric graph still has an explicit definition. On the one hand, M is the geometrical average of the weight matrix W and its transposition. Therefore, it is symmetric and has no computational problem in solving formula 2.5. On the other hand, the degree matrix D is computed from the asymmetric W instead of M, therefore still possesses the original asymmetric information.

2.4 Random Walks Induction

In practise, classifiers are not expected to be retrained each time when new documents are input. Instead, they should be able to easily deal with the new input data. This capacity of the classifier is said to be *Inductive Learning* or *Induction*.

Some works have investigated the problem of Induction. In [4], Chapelle proposed to approximate a new test data with observed data, which is not quite direct. In [16], Yu approximated the label function with the linear combination of a set of basis functions which is able to handle new test points. But basis functions should be carefully chosen to guarantee a good result. In [1], expansion to new points depends on the Representer Theorem of RKHS. This approach is complicated and good cost functions and parameters should be carefully selected. Here we propose an induction algorithm based on the Markov random walks method, which is natural, simple and effective.

The algorithm is based on a simple assumption: the newly input data will not distort the transition relations among the observed data. In fact, it also holds implicitly in[1] [4] [16]. This means that the observed data will not transit to these new data. Therefore, we have the following augmented transition matrix

$$P = \begin{pmatrix} I & O & o \\ P_{UL} & P_{UU} & o \\ P_{NL} & P_{NU} & o \end{pmatrix}, \tag{2.13}$$

where P_{NL} and P_{NU} are the transition probabilities that the new data transit to the labeled data V^L and unlabeled data V^U, respectively. For a new data v_N and an observed data $v_i \in V^L \cup V^U$, the transition probability from v_N to v_i is determined by their edge weight, i.e.,

$$p_{Ni} = \frac{w_{Ni}}{\sum\limits_{j \in V^U \cup V^L} w_{Nj}}. \tag{2.14}$$

Then we have the probability that v_N is absorbed by the absorbing states A^k with label g_k

$$p_r(A^k|v_N) = \sum_{j \in V^U} p_{Nj} p_r(A^k|v_j) + \sum_{j \in A^k} p_{Nj}, \tag{2.15}$$

where $p_r(A^k|v_j)$ is the probability that $v_j \in V^U$ is absorbed by A^k. We can further write formula 2.15 in the following matrix form

$$p_r(A^k|v_N) = (P_{NU} L_{UU}^{-1} W_{UL} + P_{NL}) f_k. \tag{2.16}$$

where, L_{UU}^{-1} can be computed and saved by the transductive learning introduced in section 2.1, therefore no extra inversion is needed. Similarly, the labels of the new data are given as

$$f^U = \arg\max_k \{P_r(A^k|v_N)\}. \tag{2.17}$$

Our induction algorithm is designed based on the intuitive interpretation of the random walks model in the framework of random walks. The algorithm utilizes both local information via one-step transition to labeled points and global information via transition to unlabeled points. Consequently, it is not a simple approximation of nearest neighborhoods. In terms of computational complexity, this induction algorithm requires no more inversion of matrix. The only matrix operations we need to implement are the multiplication and addition. Therefore, the algorithm is fast and has significant advantage on computation.

In figure 1, we give a toy example on two moon data to test the random walks model and the induction algorithm. The observed data points are plotted out using small dots. For each class, we randomly label one data, which is marked with small triangle. The classification results are described by using different colors. Classification boundary is determined using the above induction algorithm on data that are randomly sampled from the data space. It can be seen that the proposed algorithm can successfully classify both the observed data and the newly input data.

3 Graph Representation for Text

When adopting the random walks model, data should be represented as vertices on the graph. The most important information concerning this graph representation is the distance between data. Therefore, in order to guarantee a satisfying classification result, we need first precisely measure the distances between different documents. In this section, we propose the Text Divergence for measuring text difference based on language model and KL-Divergence. In experiments, a relevant Uniform Laplace Smoothing technique is utilized and an Ambiguous State Elimination Algorithm is proposed to remove noisy data.

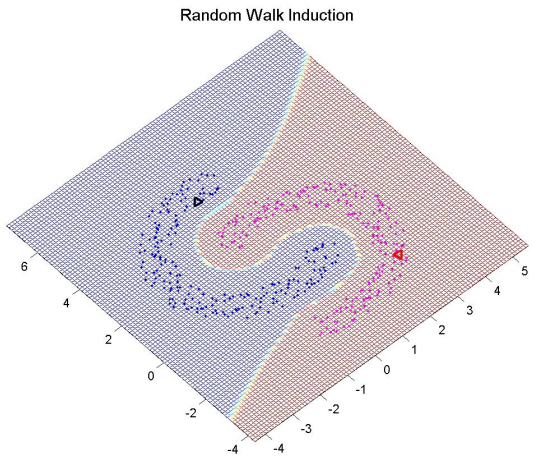


Figure 1: A toy example for the random walks model and its induction algorithm.

3.1 Text Divergence

Let T_i and T_j be two arbitrary documents. When measuring the difference between these two documents, we do not just regard them as two different sets of words. Instead, we think their words are generated by two different word distributions. In fact, the statistical Language Model, which is a probabilistic mechanism for generating text, views a document in the same way, i.e., document is generated by a certain "source", which corresponds to the word distribution here. It is natural that we would adopt the KL-Divergence as a measure of difference between the two documents since it can capture well the difference of distributions between these two documents. Therefore we define the Text Divergence (TD) to be

$$\begin{aligned} TD(T_i, T_j) &= KL(T_i, T_j) \\ &= H(T_i||T_j) - H(T_i||T_i) \end{aligned} \tag{3.18}$$

where $H(T_i||T_j)$ is the cross entropy between two distributions, i.e.,

$$H(T_i||T_j) = -\sum_w P_r(w|T_i) \log(P_r(w|T_j)), \tag{3.19}$$

where summation is over all words in the vocabulary, including both occurred ones and un-occurred ones. We will discuss the smoothing technique for the un-occurred words in the next section.

Strictly speaking, TD is not a *distance* measure, since it is not necessarily symmetric. This is why we use *difference* as above. However, we can still use it given the method introduced in section 2.3.

Once TD is computed, it follows the weight between

two documents on the graph, which is

$$w_{ij} = \exp\{-\frac{TD(T_i, T_j)^2}{2\sigma^2}\}, \tag{3.20}$$

where σ is estimated using an exhaustive search on the training set in our approach. Of course, more sophisticated learning methods can be adopted if necessary. Furthermore, weight matrix W should better be sparse so as to depress noise and to alleviate computational burden. We therefore regulate that w_{ij} is non-zero only for $T_j \in N_b\{T_i\}$, where $N_b\{T_i\}$ denotes the set of nearest neighbors of T_i.

3.2 Uniform Laplace Smoothing

To compute TD between two documents, we need to estimate the probability of each word in the documents, which is defined to be

$$P_r(w_k|T_i) = \frac{\#(w_k)_{T_i}}{\#(w)_{T_i}}, \tag{3.21}$$

where $\#(w_k)_{T_i}$ is the total number of times that word w_k occurs in T_i, $\#(w)_{T_i}$ is the total number of times that all words occur in T_i. However, the so-called "Zero-Frequency Problem" will happen if some words do not occur in the document. Therefore, smoothing techniques are needed. The most frequently adopted techniques include Laplace Smoothing, Lindstone Smoothing, Good-Turing Estimate and some interpolation methods. We refer [5] as a comprehensive empirical study on smoothing.

Here, we consider the Laplace Smoothing, which adds a small constant ε to each count and then normalize. ε is usually 1 or smaller, and has the same value for all documents. Then the word probability is given by

$$P_r(w_k|T_i) = \frac{\#(w_k)_{T_i} + \varepsilon}{\#(w)_{T_i} + \#word \times \varepsilon}, \tag{3.22}$$

where $\#word$ is the number of different words, i.e., the dimension of text features. However, this simple technique has a rather poor performance in the application here because the additive constant would probably distort the original word distributions. What is more, the distortion would dramatically vary over all documents, which severely affects the estimation of TD.

We expect the adopted smoothing technique can successfully solve the zero frequency problem without much disturbance to the word distribution. Therefore, we propose the following Uniform Laplace Smoothing(ULS). The key idea is very simple: each un-occurred word in all documents should be smoothed to the same small probability δ. δ should be small enough such that the disturbance to word distribution is negligible. Here we set it to be one-tenth of the smallest occurred word's probability of all documents. The laplace additive constant ε is then determined by δ, which is

$$\varepsilon = \frac{\#(w)_{T_i} \times \delta}{1 - \#word \times \delta}. \tag{3.23}$$

For a given document, ε is fixed. But different documents could have different ε, i.e., documents with larger word size could have larger ε, and vise versa. This smoothing technique can properly allow for the variance of different documents.

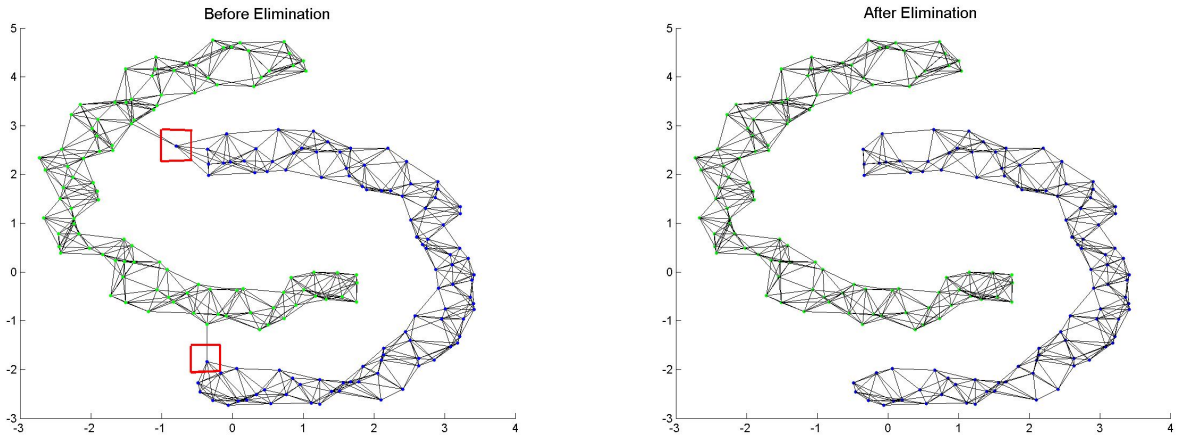
3.3 Ambiguous States Elimination

Data from a real problem are usually noisy. Some samples may have uncertain labels or even multiple labels. For example, in the news data, it is not always easy to differentiate documents from the politics topic and those from the economic topic. Some news may contain information concerning both topics since the two are frequently related. If we want to classify news from these two topics, such ambiguous samples in the training set would probably lead to a poor classification result. We illustrate the problem with figure 2 *a*. Each data point in the graph selects its seven nearest neighbors as its next possible transition states. However, the noise data, which is enclosed by red squares, cannot find their neighbors correctly. Such data are termed *Ambiguous States*. These states are troublesome in random walks: they might mistakenly transit to the samples with wrong labels and consequently cause errors in classification.

To improve the accuracy of random walks classification, we need to eliminate the ambiguous states. The ambiguity of a sample is determined by its local information. Intuitively, samples with low ambiguity should "fit" well with their neighbors. Here, "fit" means that samples should be close to their neighbors and their local shapes should be similar. By contrast, ambiguous data are sparse. They have larger distances to their neighbors and have distinct local shapes, therefore different local information. The local information used in our method is the eigenvalue of local covariance matrices. It is clear that ambiguous states should have larger local eigenvalues than their neighbors'. Therefore, similar to the inadaptability defined in [17], we define the ambiguity of sample v_i to be

$$Amb_i = \sum_j \frac{\lambda_{ij}}{\overline{\lambda_{ij}}}, \tag{3.24}$$

where λ_{ij} is the j^{th} eigenvalue of the local covariance matrix of v_i, $\overline{\lambda_{ij}}$ is the average of λ_{kj} over all v_i's neighbors $v_k \in N_b\{v_i\}$.



a *b*

Figure 2: Illustration of ambiguous states and their elimination.

The computation of local covariance matrix needs the original feature representation. However, in some occasions we only have the graph representation of data. Fortunately, it's not difficult to solve the local eigenvalues without local covariance matrix. In fact, MDS is the method to find the embedding coordinates from the distance matrix [3]. Therefore, when only the local distance matrix H of v_i is available, the local eigenvalues can be obtained through decomposition of the matrix $-N\widetilde{H}N$, where $\widetilde{H}$ is the matrix for squared distance, $N = I - \mathbf{1} \times \mathbf{1}^T/n$, $\mathbf{1}$ is a vector of n ones, n is the number of neighbors.

In figure 3, we give the ambiguities of all samples in the *course* and *faculty* subsets of the *WebKB* data. Then the ambiguous states can be eliminated by setting a proper threshold, or the number of samples to be eliminated. For example, in Figure 3, we get a threshold shown by the green line if 2% samples are to be removed. The elimination of the ambiguous states would clearly improve the neighborhood graph construction, therefore reduce the possibility of false transition among samples, as shown in figure 2 *b*.

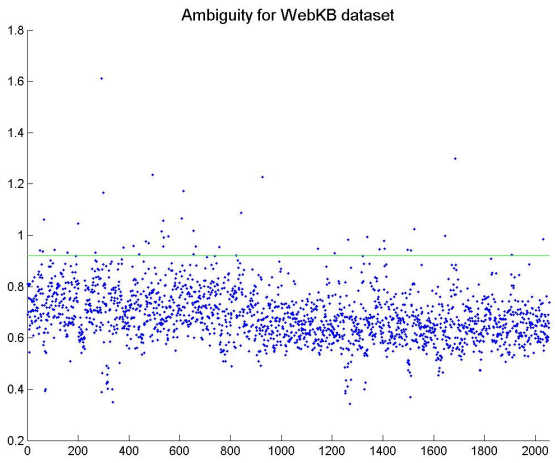


Figure 3: The ambiguities of all samples in the *course* and *faculty* subsets of the *WebKB* data, where the largest six eigenvalues are used in formula 3.24. The horizontal line is the threshold if 2% samples are to be eliminated.

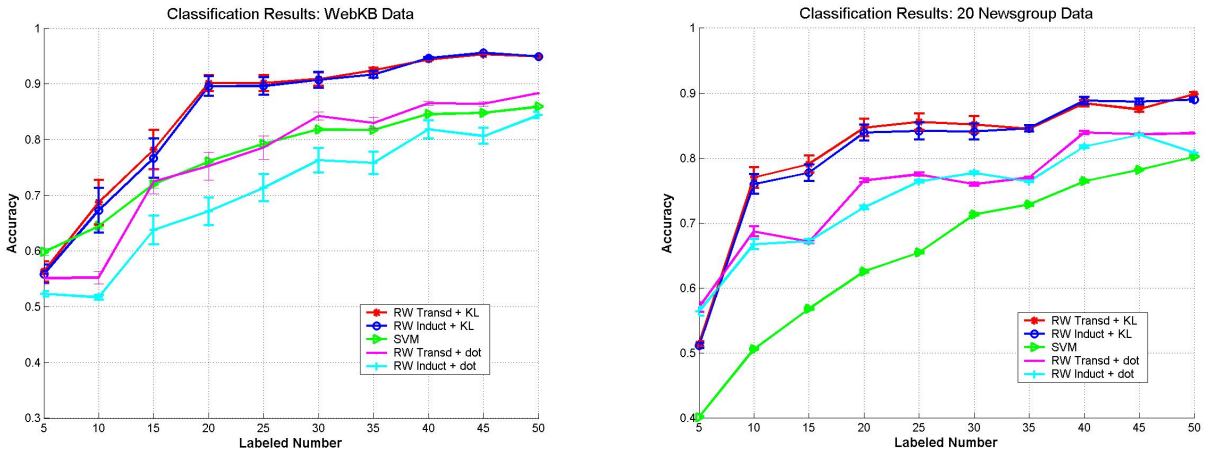
4 Experiments

In this section, we validate the proposed framework with experiments on two well known real data sets: $WebKB$ data and $20Newsgroup$ data.

4.1 WebKB Data[1]

In *WebKB* data, there are overall 8,282 web pages from 4 colleges, which have been manually classified into

[1]This data is available at http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/webkb-data.gtar.gz



a *b*

Figure 4: Experimental results on $WebKB$ and $20Newsgroup$ data. "RW Transd + KL" denotes the result given by random walks transductive learning using the text divergence measure, "RW Induct + KL" denotes the result given by random walks inductive learning using the text divergence measure, "RW Transd + dot" denotes the result given by random walks transductive learning using the dot distance measure, "RW Induct + dot" denotes the result given by random walks inductive learning using the dot distance measure.

7 categories. Here, we consider its two subsets: *course* and *faculty*, which have 2054 pages in total. To extract and select document features, we implement the following operations. First we remove 500 common words from the standard stop-list. We then adopt the *"Document Frequency + χ^2 statistics Max Cut"* [9] method, which results in 5000 demensional document feature vectors. We randomly divide the dataset into three subsets: labeled set, unlabeled set and "unseen" set (to test the induction algorithm). The unlabeled set has a fixed size of 500 documents. In experiment, we vary the size of labeled set and put the left documents into the "unseen" set. Each document chooses 15 nearest neighbors as its next possible transition states. When eliminating ambiguous states, 2% unlabeled documents are removed.

When different number of documents are labeled, we obtain the following classification results, as shown in figure 4 *a*, where each point is the average over 30 random runs. The red line corresponds to the results given by random walks transductive learning on the unlabeled data, while the blue line corresponds to the results given by random walks inductive learning on the "unseen" data. It shows that the transductive learning performs a little better than the inductive learning, but the difference is not obvious. For comparison, we consider SVM, which is known for its superior performance on high dimensional and small labeled problems. Here, RBF kernel is adopted with its parameter being estimated using an exhaustive search on the training set. The classification results are shown using the green line. It can be seen that SVM has a better performance when there are too few labeled samples. But as the number of labeled data increases, random walks will surpass SVM very soon. Besides, we also provide classification results of random walks transductive learning and inductive learning using the common symmetric dot distance measure, which are shown using the magenta and cyan lines respectively. This proves the superb effectiveness of our text divergence measure and its smoothing technique.

4.2 20Newsgroup Data[2]

In 20Newsgroup data, there are overall 20,000 news pieces, which have been manually classified into 20 categories, i.e., around 1000 documents per class. Here, we consider its 4 subsets: autos, motorcycles, baseball, hockey, which have 4000 documents in total. We adopted the same feature extraction and selection method as the above experiment, as well as the same experimental settings. After 30 independent runs, we get the above experimental results, as shown in figure 4 *b*. Clearly, similar conclusions can be drawn except that SVM never exceeds random walks even when the number of labeled data is small.

[2]This data is available at http://people.csail.mit.edu/ jrennie/20Newsgroups/

5 Conclusions and Future Work

We propose in this paper a random walk method for text classification. In this method, classification is formulated to the problem of solving the probabilities that unlabeled documents are absorbed by labeled documents. Based on it, we derive the Laplacian operator for asymmetric graphs to solve the computational problem caused by asymmetric transition matrix. We then developed a random walks induction algorithm to expand the method to newly input documents. On the other hand, for better representing text data, the method proposes a new text difference measure based on statistical language model and KL-Divergence, and a relevant smoothing technique. Compared with the common dot distance measure, such a measure can better mine the difference information from the text data. We also propose an ambiguous states elimination algorithm to address the problem of noisy input data. Experiments on $WebKB$ and $20Newsgroup$ data sets show that, the proposed method has a superb performance, which is higher than those of SVM and the random walks using dot distance measure.

However, there are still some problems with the proposed method. First, although the algorithm for ambiguous state elimination can eliminate noisy data, it might also remove some non-noisy data. Our next work is to make this algorithm more robust. Second, we need more experiments to compare the proposed method with other semi-supervised methods, such as Transductive SVM[15], and MinCut[2]. Third, it is interesting to consider active learning based on random walks, which we will investigate later.

Acknowledgements

This work is supported by the Project (60475001) of the National Nature Science Foundation of China. We would like to thank Yungang Zhang, Shijun Wang, Jing Yang, Mikhail Belkin, and Dengyong Zhou for their help with this work. Special thanks to Kai Yu, who provided us his data, which is very helpful to our experiments. We would also thank the anonymous reviewers for their valuable comments.

References

[1] M. Belkin, P. Niyogi, V. Sindhwani, *On Manifold Regularization*, AI & Statistics, 2005.

[2] A. Blum, S. Chawla, *Learning from Labeled and Unlabeled Data using Graph Mincuts*, Proc. 18^{th} International Conf. on Machine Learning, 2001.

[3] I Borg, PJF Groenen, *Modern multidimensional scaling: theory and applications*, Springer, New York, 1997.

[4] O. Chapelle, J. Weston, B. Scholkopf, *Cluster kernels for semi-supervised learning*, Advances in Neural Information Processing Systems 15, 2003.

[5] S. F. Chen, J.T. Goodman, *An empirical study of smoothing techniques for language model*, In Proceedings of the 34th Annual Meeting of the ACL, 1996.

[6] P.G. Doyle, J. Snell *Random Walks and Electric Networks*, Mathematical Assoc. of America, 1984.

[7] R. Merris, *A survey of graph Laplacians*, In *Lin. Multilin. Algebra*, 39 (1995), 19C31.

[8] J.M. Ponte, W.B. Croft, *A language modeling approach to information retieval*, Proceedings of ACM-SIGIR 1998, pages 275-281, 1998.

[9] M. Rogati, Y. Yang, *High-performing feature selection for text classification*, In Proceedings of the eleventh international conference on Information and knowledge management, McLean, Virginia, USA (2002)

[10] F. Sebastiani, *Machine Learning in Automated Text Categorization*, ACM. Computing Surveys, Vol. 34, No. 1, March 2002, pp.1-47.

[11] F. Sebastiani, *Text Categorization*, In *Text Mining and its Applications to Intelligence, CRM and Knowledge Management*, pages 109-129, WIT Press, Southampton, UK, 2005.

[12] M. Seeger, *Learning from Labeled and Unlabeled Data*, Technical Report, University of Edinburgh, 2001.

[13] F. Song, W.B. Croft, *A general language model for information retrieval*, In Proceedings of Eight International Conference on Information and Knowledge Management. (CIKM1999), 1999.

[14] M. Szummer, T. Jaakkola, *Partially labeled classification with Markov random walks*, Advances in Neural Information Processing Systems, 14, 2002.

[15] V. N. Vapnik, *Statistical Learning Theory*, J. Wiley, New York, 1998.

[16] K. Yu, V. Tresp, D. Zhou, *Semi-supervised Induction*, Advances in Neural Information Processing Systems 17, 2005.

[17] Y. Zhang, C. Zhang, S. Wang, *Clustering in Knowledge Embedded Space*, In ECML03, 2003.

[18] D. Zhou, B. Scholkopf, *Transductive Inference with Graphs*, Technical report, (2004).

[19] X. Zhu, Z. Ghahramani, J. Lafferty, *Semi-supervised Learning using Gaussian Fields and Harmonic Functions*, In ICML-2003, 2003.

[20] X. Zhu, *Semi-Supervised Learning Literature Survey*, Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.