Manifold Adaptive Experimental Design for Text Categorization

Deng Cai, *Member, IEEE* and Xiaofei He, *Senior Member, IEEE*

Abstract—In many information processing tasks, labels are usually expensive and the unlabeled data points are abundant. To reduce the cost on collecting labels, it is crucial to predict which unlabeled examples are the most informative, *i.e.*, improve the classifier the most if they were labeled. Many active learning techniques have been proposed for text categorization, such as SVM$_{Active}$ and Transductive Experimental Design. However, most of previous approaches try to discover the discriminant structure of the data space, whereas the geometrical structure is not well respected. In this paper, we propose a novel active learning algorithm which is performed in the data manifold adaptive kernel space. The manifold structure is incorporated into the kernel space by using graph Laplacian. This way, the manifold adaptive kernel space reflects the underlying geometry of the data. By minimizing the expected error with respect to the optimal classifier, we can select the most representative and discriminative data points for labeling. Experimental results on text categorization have demonstrated the effectiveness of our proposed approach.

Index Terms—Text categorization, active learning, experimental design, manifold learning, kernel method

1 Introduction

Text classification has been a fundamental problem in many information processing tasks [1], [14], [17], [22], [32], [44]. In order to train a classifier that can automatically distributes documents into different semantic categories, one usually needs to collect a large set of labeled examples. In order to reduce the efforts in collecting labels, many researchers studied to use *active learning* [37] for text categorization. The key problem in active learning is determining which unlabeled examples would be the most informative, i.e., improve the classifier the most if they were labeled and used as training examples.

There has been a long tradition of research on active learning in machine learning community [10], [12], [16]. One popular group of algorithms select the most uncertain data given previously trained models. One representative algorithm in this group is SVM$_{Active}$. Based on the observation that the closer to the SVM boundary a data point is, the less reliable its classification is, Tong et al. proposed SVM$_{Active}$ which selects those unlabeled data points closest to the boundary to solicit user's labeling so as to achieve maximal refinement on the hyperplane between the two classes [43]. Another group of algorithms choose the most informative points that optimize some expected measures [12]. Many algorithms in statistics belong to this category. In statistics, the problem of selecting samples to label is typically referred to as *experimental design*[2]. Classical optimal experimental design approaches include A-optimal design, D-optimal design, and E-optimal design. Recently, Yu et al. has proposed Transductive Experimental Design (TED) with either sequential [45] or convex [46] optimization which has yielded impressive results on text categorization. TED is fundamentally based on optimal experimental design but evaluates the expected prediction error on both labeled and unlabeled examples.

Standard learning systems operate on input data after they have been transformed into feature vectors living in a m-dimensional space. In such a space, standard learning tasks like classification, clustering, data selection (active learning) can be performed. The resulting hypothesis will then be applied to test points in the same vector space, in order to make predictions. Recently, various researchers (see [3], [33], [41]) have considered the case when the data is drawn from sampling a probability distribution that has support on or near to a *submanifold* of the ambient space. Here, a d-dimensional submanifold of a Euclidean space $\mathbb{R}^m$ is a subset $\mathcal{M}^d \subset \mathbb{R}^m$ which locally looks like a flat d-dimensional Euclidean space [26]. In order to detect the underlying manifold structure, many *manifold learning* algorithms have been proposed, such as Locally Linear Embedding (LLE) [33], ISOMAP [41], and Laplacian Eigenmap [3]. One of the key ideas in manifold learning approaches is the so called locally invariant idea [18], *i.e.*, the nearby points are likely to have the similar embedding/labels.

All the early manifold learning techniques mainly focus on dimensionality reduction. Recently, the manifold idea (or, locally invariant idea) has been successfully applied to clustering [30], semi-supervised learning [4], [25], [40], [47], topic modeling [9] and matrix factoriza-

- *D. Cai and X. He are with the State Key Lab of CAD&CG, College of Computer Science, Zhejiang University, 388 Yu Hang Tang Rd., Hangzhou, Zhejiang, China 310058. E-mail: {dengcai,xiaofeihe}@cad.zju.edu.cn.*

Manuscript received 26 Aug. 2009; revised 08 May. 2010; accepted 17 Sep. 2010

tion [6]. Particularly, the manifold idea achieved great successes on various text analysis tasks [7], [8], [20], [24]. For example, both Tansductive SVM [24] and spectral graph transducers [25] (two of the popular semi-supervised learning algorithms for text analysis) used the locally invariant idea. All these approaches demonstrated that learning performance can be significantly enhanced if the geometrical structure is exploited and the local invariance is considered. It is very natural that this idea should also be considered in active learning. However, most of the existing active learning algorithms fail to take into account the intrinsic manifold structure.

In this paper, we propose a novel manifold adaptive active learning algorithm for text categorization. By using a data-dependent norm on reproducing kernel Hilbert space (RKHS) proposed by Vikas et al. [40], we can warp the structure of the RKHS to reflect the underlying geometry of the data. The conventional optimal experimental design can then be performed in the manifold adaptive kernel space. We discuss how to kernelize the convex transductive experimental design which gives rise to nonlinear manifold adaptive data selection for text categorization.

The rest of the paper is organized as follows: in Section 2, we provide a brief review of the related work. Our manifold adaptive active learning algorithm for text categorization is introduced in Section 3. The experimental results are presented in Section 4. Finally, we provide the concluding remarks and suggestions for future work in Section 5.

2 Background

The generic problem of active learning is the following. Given a set of points $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ in $\mathbb{R}^m$, find a subset $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k\} \subset \mathcal{X}$ which contains the most informative points. In other words, the points $\mathbf{z}_i (i = 1, \cdots, k)$ can improve the classifier the most if they are labeled and used as training points.

There has been extensive research in machine learning on this subject. Some popular directions include selecting the most uncertain data given previously trained models [42] and selecting the most representative points by exploiting the cluster structure of the data[13]. One representative algorithm which selects the most uncertain data is SVM$_{Active}$[42], [43]. This method selects the points that can reduce the size of the version space as much as possible. Since it is difficult to measure the version space, the authors provide three approximations. One of them which selects the points closest to the current decision boundary is called SimpleMargin. This method was also proposed by Schohn and Cohn [35] and has been very popular. Some other methods include query-by-committee [39], density-weighted methods [29], [38], and explicit error-reduction techniques [34], [48]. Please refer [37] for a comprehensive treatment of active learning approaches.

In statistics, the problem of selecting samples to label is typically referred to as *experimental design*. The sample $\mathbf{x}$ is referred to as *experiment*, and its label y is referred to as *measurement*. The study of *optimal experimental design* (OED) [2] is concerned with the design of experiments that are expected to minimize variances of a parameterized model. Since the approach described in this paper will be based on OED, we give some detailed descriptions on optimal experimental design as follows.

2.1 Optimal Experimental Design

We consider learning a linear function $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ from *observation* $y = \mathbf{w}^T\mathbf{x}+\epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is observation error. Suppose we have a set of labeled example points $(\mathbf{z}_1, y_1), \cdots, (\mathbf{z}_k, y_k)$, where y_i is the label of $\mathbf{z}_i$. Thus, the maximum likelihood estimate of $\mathbf{w}$ is obtained by

$$\widehat{\mathbf{w}} = \arg\min_{\mathbf{w}} \left\{ J(\mathbf{w}) = \sum_{i=1}^{k} \left(\mathbf{w}^T\mathbf{z}_i - y_i\right)^2 \right\} \qquad (1)$$

By Gauss-Markov theorem, we know that $\mathbf{e} = \hat{\mathbf{w}} - \mathbf{w}$ has zero mean and a covariance matrix given by $\sigma^2 C_{\mathbf{w}}$, where $C_{\mathbf{w}}$ is the inverted Hessian of $J(\mathbf{w})$

$$C_{\mathbf{w}} = \left(\frac{\partial^2 J_{sse}}{\partial \mathbf{w}^2}\right)^{-1} = \left(\sum_{i=1}^{k} \mathbf{z}_i\mathbf{z}_i^T\right)^{-1} = \left(ZZ^T\right)^{-1}$$

where $Z = (\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k)$. Then OED formulates the optimization problem as minimization of some measurement of estimation error derived from $C_{\mathbf{w}}$. Three most common measures are trace of $C_{\mathbf{w}}$ (leads to A-optimal design), determinant of $C_{\mathbf{w}}$ (leads to D-optimal design) and maximum eigenvalue of $C_{\mathbf{w}}$ (leads to E-optimal design). Some recent work on optimal experimental design can be found in [15], [21], [45].

3 Manifold Adaptive Experimental Design

In order to incorporate the manifold structure into the learning process, a natural way is to perform learning tasks in manifold adaptive kernel space. In this section, we will describe our manifold adaptive experimental design approach which is fundamentally based on transductive experimental design and manifold adaptive kernel. We begin with a description of transductive experimental design.

3.1 Transductive Experimental Design

Let $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ be the set of all the data points and $\mathcal{Z} = \{\mathbf{z}_1, \cdots, \mathbf{z}_k\} \subset \mathcal{X}$ be the set of selected points for labeling.

The key idea of Transductive Experimental Design (TED) is to minimize the average expected square predictive error of the learned function f. For any $\mathbf{x}$, let $\hat{y} = \hat{\mathbf{w}}^T\mathbf{x}$ be its predicted observation. The expected square prediction error can be written as follows:

$$\begin{aligned} & E(y-\hat{y})^2 \\ = \; & E\big(\epsilon+\mathbf{w}^T\mathbf{x}-\hat{\mathbf{w}}^T\mathbf{x}\big)^2 \\ = \; & \sigma^2+\mathbf{x}^T[E(\mathbf{w}-\hat{\mathbf{w}})(\mathbf{w}-\hat{\mathbf{w}})^T]\mathbf{x} \\ = \; & \sigma^2+\sigma^2\mathbf{x}^T(ZZ^T)^{-1}\mathbf{x} \end{aligned}$$

Interestingly, the expected square prediction error of $\mathbf{x}$ does not depend on the labels, but only the training points Z. The average expected square predictive error over the complete data set $\mathcal{X}$ is

$$\frac{1}{n}\sum_{i=1}^{n} E(y_i-\hat{\mathbf{w}}^T\mathbf{x}_i)^2=\sigma^2+\sigma^2\mathrm{Tr}\big(X^T(ZZ^T)^{-1}X\big) \quad (2)$$

In order to minimize the average expected square predictive error, one should find a subset $\mathcal{Z}$ which minimizes Eq. (2). However, it can be verified that this optimization problem is NP-hard [45] and therefore infeasible to find a global optimum. After some mathematical derivations, the minimization of average expected square predicative error can be formulated as an equivalent optimization problem as follows:

$$\min_{\boldsymbol{\alpha}_i\in\mathbb{R}^k, Z=(\mathbf{z}_1,\cdots,\mathbf{z}_k)} \sum_{i=1}^{n}\|\mathbf{x}_i-Z^T\boldsymbol{\alpha}_i\|^2+\mu\|\boldsymbol{\alpha}_i\|^2 \quad (3)$$

Yu et al. [45] proposes a sequential greedy algorithm that selects $\mathbf{z}_i's$ one at time. However, the obtained result is suboptimal.

Recently, a convex relaxation of (2) was introduced in [46]. By introducing auxiliary variables $\boldsymbol{\beta}=(\beta_1,\cdots,\beta_m)$ to control the inclusion of examples into the training set, the optimization problem can be rewritten as follows:

$$\begin{aligned} \min_{\boldsymbol{\beta},\boldsymbol{\alpha}_i\in\mathbb{R}^n} \quad & \sum_{i=1}^{n}\Big(\|\mathbf{x}_i-X^T\boldsymbol{\alpha}_i\|^2+\sum_{j=1}^{n}\frac{\alpha_{i,j}^2}{\beta_j}\Big)+\gamma\|\beta\|_1 \quad (4) \\ s.t. \quad & \beta_i\geq 0, i=1,\cdots,n, \end{aligned}$$

where $\boldsymbol{\alpha}_i=(\alpha_{i,1},\cdots,\alpha_{i,n})^T$ and $\|\cdot\|_1$ denotes the ℓ_1 norm. As suggested by Lasso regression [19], the minimization of the ℓ_1 norm $\|\boldsymbol{\beta}\|_1$ leads to a sparse $\boldsymbol{\beta}$. That is, some entries of $\boldsymbol{\beta}$ will be zero. It is easy to check that, if $\beta_j=0$, then all $\alpha_{1,j},\cdots,\alpha_{n,j}$ have to be zero, otherwise the objective function goes to infinity. Thus, the j-th example will not be selected. It can be shown that the optimization problem (4) is convex, and therefore global optimum can be obtained. For the details, please see [46].

Convex TED has shown its promising results on text categorization. However, it fails to take into account the intrinsic manifold structure which has been shown very useful for improving the learning performance by many previous studies [4], [28].

3.2 Manifold Adaptive Kernel

In order to incorporate the manifold structure into the active learning process, a natural way is to perform active learning tasks in manifold adaptive kernel space. In the following we discuss how to incorporate the manifold structure into the reproducing kernel Hilbert space (RKHS) which leads to manifold adaptive kernel space.

Kernel trick is usually applied in the hope of discovering the nonlinear structure in the data by mapping the original nonlinear observations into a higher-dimensional linear space [36]. The most commonly used kernels include Gaussian kernel and polynomial kernel. However, the nonlinear structure captured by the data-independent kernels may not be consistent with the intrinsic manifold structure, such as geodesic distance, curvature, and homology [4], [31].

In this work, we adopt the manifold adaptive kernel proposed by Vikas et al. [40]. Let $\mathcal{V}$ be a linear space with a positive semi-definite inner product (quadratic form) and let $S:\mathcal{H}\rightarrow\mathcal{V}$ be a bounded linear operator. We define $\tilde{\mathcal{H}}$ to be the space of functions from $\mathcal{H}$ with the modified inner product [40]

$$\langle f,g\rangle_{\tilde{\mathcal{H}}}=\langle f,g\rangle_{\mathcal{H}}+\langle Sf,Sg\rangle_{\mathcal{V}}.$$

Vikas et al. have shown that $\tilde{\mathcal{H}}$ is still a RKHS [40].

Given the examples $\mathbf{x}_1,\cdots,\mathbf{x}_m$, let $S:\mathcal{H}\rightarrow\mathbb{R}^m$ be the evaluation map

$$S(f)=\Big(f(\mathbf{x}_1),\cdots,f(\mathbf{x}_m)\Big)^T.$$

Denote $\mathbf{f}=\big(f(\mathbf{x}_1),\cdots,f(\mathbf{x}_m)\big)^T$ and $\mathbf{g}=\big(g(\mathbf{x}_1),\cdots,g(\mathbf{x}_m)\big)^T$. Notice that $\mathbf{f},\mathbf{g}\in\mathcal{V}$, thus we have

$$\langle Sf,Sg\rangle_{\mathcal{V}}=\langle\mathbf{f},\mathbf{g}\rangle=\mathbf{f}^TM\mathbf{g}$$

where M is a positive semi-definite matrix. We define

$$\mathbf{k}_{\mathbf{x}}=\Big(\mathcal{K}(\mathbf{x},\mathbf{x}_1),\cdots,\mathcal{K}(\mathbf{x},\mathbf{x}_m)\Big).$$

It can be shown that the reproducing kernel in $\tilde{\mathcal{H}}$ is [40]:

$$\tilde{\mathcal{K}}(\mathbf{x},\mathbf{z})=\mathcal{H}(\mathbf{x},\mathbf{z})-\lambda\mathbf{k}_{\mathbf{x}}^T(I+MK)^{-1}M\mathbf{k}_{\mathbf{z}}, \quad (5)$$

where I is an identity matrix, K is the kernel matrix in $\mathcal{H}$, and $\lambda\geq 0$ is a constant controlling the smoothness of the functions. The key issue now is the choice of M, so that the deformation of the kernel induced by the data-dependent norm, is motivated with respect to the intrinsic geometry of the data.

In order to model the manifold structure, we construct a nearest neighbor graph G. For each data point $\mathbf{x}_i$, we find its k nearest neighbors denoted by $\mathcal{N}(\mathbf{x}_i)$ and put an edge between $\mathbf{x}_i$ and its neighbors. There are many choices for the weight matrix on the graph. A simple one is as follows:

$$W_{ij}=\begin{cases} 1, & \text{if } \mathbf{x}_i\in\mathcal{N}(\mathbf{x}_j) \text{ or } \mathbf{x}_j\in\mathcal{N}(\mathbf{x}_i); \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

The graph Laplacian [11] is defined as $L=D-W$ where D is a diagonal degree matrix given by $D_{ii}=\sum_j W_{ij}$.

The graph Laplacian provides the following smoothness penalty on the graph:

$$\mathbf{f}^T L\mathbf{f} = \sum_{i=1}^{n} \Big(f(\mathbf{x}_i) - f(\mathbf{x}_j)\Big)^2 W_{ij}$$

By setting $M = L$, we eventually get the following manifold adaptive kernel:

$$\mathcal{K}_{\mathcal{M}}(\mathbf{x},\mathbf{z}) = \mathcal{K}(\mathbf{x},\mathbf{z}) - \lambda \mathbf{k}_{\mathbf{x}}^T (I + LK)^{-1} L\mathbf{k}_{\mathbf{z}}. \tag{7}$$

It is important to note that all the existing popular kernels (*e.g.*, Gaussian kernel, polynomial kernel and linear kernel) can be transformed to manifold adaptive kernels. For text analysis, previous studies [23], [44] have shown that linear models are enough due to the large number of features for text data. Actually, a stronger conclusion can be found at [5] which shows that a linear mapping (function) can unfold any manifold structure in the data as long as the data points are linear independent (it is usually true for the text data because the number of features of the text data is usually larger than the number of samples). Thus, we simply use the linear kernel (transformed to manifold adaptive kernel) in our text categorization experiments.

3.3 Convex TED in Reproducing Kernel Hilbert Space

In the following we discuss how to perform convex TED in the manifold adaptive kernel space.

For given examples $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}^m$ with a positive definite mercer kernel $\mathcal{K} : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$, there exists a unique RKHS $\mathcal{H}$ of real valued functions on $\mathbb{R}^m$. Let $\mathcal{K}_t(s)$ be the function of s obtained by fixing t and letting $\mathcal{K}_t(s) \doteq \mathcal{K}(s,t)$. $\mathcal{H}$ consists of all finite linear combinations of the form $\sum_{i=1}^{l} a_i \mathcal{K}_{t_i}$ with $t_i \in \mathbb{R}^m$ and limits of such functions as the t_i becomes dense in $\mathbb{R}^m$. We have $\langle \mathcal{K}_s, \mathcal{K}_t \rangle_{\mathcal{H}} = \mathcal{K}(s,t)$ [36].

Let $\phi : \mathbb{R}^m \to \mathcal{H}$ be a feature map from the input space $\mathbb{R}^m$ to $\mathcal{H}$, and $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$. Let $\phi(X)$ denote the data matrix in RKHS, that is, $\phi(X) = \big(\phi(\mathbf{x}_1), \cdots, \phi(\mathbf{x}_n)\big)$. Similarly, we define $\phi(Z) = \big(\phi(\mathbf{z}_1), \cdots, \phi(\mathbf{z}_n)\big)$. The convex TED optimization problem in RKHS can be written as follows:

$$\begin{aligned} \min_{\boldsymbol{\beta}, \boldsymbol{\alpha}_i \in \mathbb{R}^n} & \sum_{i=1}^{n} \Big(\|\phi(\mathbf{x}_i) - \phi(X)\boldsymbol{\alpha}_i\|^2 + \sum_{j=1}^{n} \frac{\alpha_{i,j}^2}{\beta_j} \Big) + \gamma \|\boldsymbol{\beta}\|_1 \\ \text{s.t.} \quad & \beta_j \ge 0, j = 1, \cdots, n. \end{aligned} \tag{8}$$

Let $diag(\boldsymbol{\beta})$ be a diagonal matrix whose entries are $\beta_1, \cdots, \beta_n$. Thus,

$$\sum_{j=1}^{n} \frac{\alpha_{i,j}^2}{\beta_j} = \boldsymbol{\alpha}_i^T diag(\boldsymbol{\beta})^{-1} \boldsymbol{\alpha}_i.$$

By some simple algebraic steps, we get

$$\begin{aligned} & \sum_{i=1}^{n} \Big(\|\phi(\mathbf{x}_i) - \phi(X)\boldsymbol{\alpha}_i\|^2 + \sum_{j=1}^{n} \frac{\alpha_{i,j}^2}{\beta_j} \Big) \\ = & \sum_{i=1}^{n} \Big(\big(\phi(\mathbf{x}_i) - \phi(X)\boldsymbol{\alpha}_i\big)^T \big(\phi(\mathbf{x}_i) - \phi(X)\boldsymbol{\alpha}_i\big) + \\ & \boldsymbol{\alpha}_i^T diag(\boldsymbol{\beta})^{-1} \boldsymbol{\alpha}_i \Big) \\ = & \sum_{i=1}^{n} \Big(\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_i) - 2\boldsymbol{\alpha}_i^T \phi(X)^T \phi(\mathbf{x}_i) + \\ & \boldsymbol{\alpha}_i^T \phi(X)^T \phi(X) \boldsymbol{\alpha}_i + \boldsymbol{\alpha}_i^T diag(\boldsymbol{\beta})^{-1} \boldsymbol{\alpha}_i \Big) \end{aligned}$$

Now, taking the derivative of the objective function (8) with respect to $\boldsymbol{\alpha}_i$ and requiring it to be zero, we get:

$$-2\phi(X)^T \phi(\mathbf{x}_i) + 2\phi(X)^T \phi(X)\boldsymbol{\alpha}_i + 2diag(\boldsymbol{\beta})^{-1} \boldsymbol{\alpha}_i = 0$$

Finally, we get:

$$\boldsymbol{\alpha}_i = \Big(diag(\boldsymbol{\beta})^{-1} + \phi(X)^T \phi(X) \Big)^{-1} \phi(X)^T \phi(\mathbf{x}_i). \tag{9}$$

We define a $n \times n$ kernel matrix K such that $K_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$. Let $\mathbf{u}_i$ be the *i-th* column (or row, since K is symmetric) vector of K:

$$\mathbf{u}_i = \big(\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_1), \cdots, \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_n)\big)^T = \phi(X)^T \phi(\mathbf{x}_i).$$

By noticing that $\phi(X)^T \phi(X) = K$, Eq. (9) can be rewritten as follows:

$$\boldsymbol{\alpha}_i = \Big(diag(\boldsymbol{\beta})^{-1} + K \Big)^{-1} \mathbf{u}_i \tag{10}$$

Once $\boldsymbol{\alpha}_i$'s are obtained, we can fix $\boldsymbol{\alpha}_i$'s and find the minimum solution for β_j. Again, we take the derivative of the objective function (8) with respect to β_j and require the derivative to be zero. By noticing that β_j is non-negative, we have

$$\sum_{i=1}^{n} \big(-\frac{\alpha_{i,j}^2}{\beta_j^2} \big) + \gamma = 0 \tag{11}$$

Finally, we get:

$$\beta_j = \sqrt{\frac{\sum_{i=1}^{n} \alpha_{i,j}^2}{\gamma}} \tag{12}$$

So $\alpha_{i,j}$ and β_j can be iteratively computed. Since the objective function is convex, the globally optimal solution is guaranteed to be obtained.

3.4 The Manifold Adaptive Experimental Design Algorithm

We summarize our manifold adaptive experimental design (MAED) algorithm as follows (also in Table 1):

1) **Construct the manifold adaptive kernel**. Construct a k nearest neighbor G with weight matrix defined in (6). Calculate the graph Laplacian $L = D - W$. Let $\mathcal{K}$ be any data independent kernel (e.g. Gaussian or linear kernel) associated with the kernel matrix K.

TABLE 1
The algorithm of Manifold Adaptive Experimental Design (MAED)

```
Input:   Data set with $n$ unlabeled samples, the number of selected samples $k$,
         the number of nearest neighbors $p$, the manifold adaptive regularization parameter $\lambda$,
         the sparse regularization parameter $\gamma$, the data independent kernel type (e.g., Gaussian, linear)
Output:  $k$ selected samples.
1: Construct a nearest neighbor graph with weight matrix $W$ as in Eq. (6) and,
   calculate the graph Laplacian $L = D - W$.
2: Compute the kernel matrix $K$ with the input kernel type.
3: Compute the manifold adaptive kernel matrix $K_{\mathcal{M}}$ as in Eq. (13)
4: Initialize $\alpha_{i,j} = 1$, and iteratively compute $\beta_j$ and $\boldsymbol{\alpha}_i$ as in Eq. (14) and Eq. (15), until convergence
5: Rank the data points according to $\beta_j (j = 1, \cdots, n)$ in descending order, and return the top $k$ data points.
```

That is, $K_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$. Let $\mathbf{k}_i$ be the *i-th* column vector of K. Calculate the manifold adaptive kernel matrix $K_{\mathcal{M}}$ as follows:

$$\begin{aligned} K_{\mathcal{M},ij} =& \mathcal{K}_{\mathcal{M}}(\mathbf{x}_i, \mathbf{x}_j) \\ =& K_{ij} - \lambda \mathbf{k}_i^T (I + LK)^{-1} L \mathbf{k}_j. \end{aligned} \tag{13}$$

2) **Solve manifold adaptive active learning optimization problem**. Let $\mathbf{u}_i$ be the *i-th* column (or row, since $K_{\mathcal{M}}$ is symmetric) vector of $K_{\mathcal{M}}$. Initialize $\alpha_{i,j} = 1$, and iteratively compute

$$\beta_j = \sqrt{\frac{\sum_{i=1}^n \alpha_{i,j}^2}{\gamma}}, j = 1, \cdots, n. \tag{14}$$

$$\boldsymbol{\alpha}_i = \left(diag(\boldsymbol{\beta})^{-1} + K_{\mathcal{M}} \right)^{-1} \mathbf{u}_i, i = 1, \cdots, n. \tag{15}$$

until convergence.

3) **Data selection**. Rank the data points according to $\beta_j (j = 1, \cdots, n)$ in descending order, and select the top k data points.

Once we select the most informative data points, any classification algorithm can be applied to do pattern classification.

Constructing the p nearest neighbor graph in the first step of MAED needs $O(pn^2)$. Computing the data independent kernel matrix K in the second step needs $O(n^2)$. Computing the manifold adaptive kernel matrix in the third step needs $O(n^3)$ and the fourth step needs $O(tn^3)$ where t is the iteration times. In our experiments, the MAED algorithm converges very fast and t is usually less than 20. The overall computational cost of MAED is $O(n^3)$, which is the same as the original convex TED algorithm in the kernel space.

4 Experiments

In this section, we evaluate the performance of our proposed algorithm and compare it with the state-of-the-art active learning algorithms for text categorization.

4.1 Simple Toy Example

Our MAED algorithm is fundamentally based on TED. The difference between them is whether the geometric structure of the data is considered. To get a intuitive idea of how the two algorithms perform differently, we give a simple toy example in Figure 1. The data set contains two circles. Eight points are selected by TED and MAED. Both algorithms use the Gaussian kernel. As can be seen, all the points selected by TED are from the small circle, while MAED selects five points from the big circle and three from the small circle. Clearly, the points selected by our MAED algorithm can better represent the original data set.

4.2 Data and Experimental Settings

Our empirical study on text categorization was conducted based on three real-world text corpora.

- The first data set is 20Newsgroups corpus[1], which contains 18,744 documents with 61,188 distinct words. This data set has 20 categories, each with around 1000 documents.
- The second data set is a subset of the Reuters-21578 text data set[2]. This subset has 2,919 documents, including categories ‘acq’, ‘crude’, ‘trade’, and ‘money’, each with 2,025, 321, 298, and 245 documents respectively. In this data set we have 10,499 distinct words.
- The third data set is a subset of the RCV1-v2 corpus [27]. RCV1 contains the information of topics, regions and industries for each document and a hierarchical structure for topics and industries. A set of 9,625 documents with 29,992 distinct words is chosen for our experiments, including categories ‘C15’, ‘ECAT’, ‘GCAT’, and ‘MCAT’, each with 2,022, 2,064, 2,901, and 2,638 documents respectively.

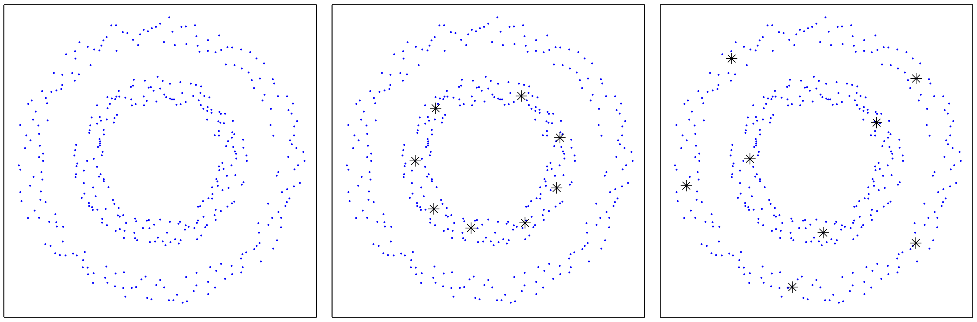
The standard TF-IDF weighting scheme is used to generate the feature vector for each document:

$$tf\text{-}idf = (1 + \log tf) \times \log \frac{N}{df}$$

where N is the number of documents in the corpus and df is the number of documents containing a particular word.

The experimental settings in this work are basically the same as those in [46]. We conduct one-against-all classification for each category and treat each problem as binary classification. We use the standard precision,

[1] http://people.csail.mit.edu/jrennie/20Newsgroups/
[2] http://www.daviddlewis.com/resources/testcollections/reuters21578/



(a) Data (b) TED (c) MAED

Fig. 1. Data selection by active learning algorithms TED and MAED. The selected data points are marked as $*$. Clearly, the points selected by our MAED algorithm can better represent the original data set.

TABLE 2
Text categorization results on 20Newsgroups. (on the unlabeled data)

	Macro-F_1 (%)				Micro-F_1 (%)			
k	Random	Simple Margin	convex TED	MAED	Random	Simple Margin	convex TED	MAED
20	13.0±0.9	10.0±0.6	27.8±2.0	**34.4±2.4**	17.9±0.9	15.3±0.6	31.6±2.0	**36.6±2.3**
40	20.6±0.6	25.8±0.9	36.6±1.9	**41.9±1.9**	25.1±0.7	29.3±0.8	38.8±1.9	**42.9±2.0**
60	26.8±0.6	28.3±0.8	39.8±1.7	**42.7±2.3**	30.9±0.6	30.5±0.8	42.1±1.6	**44.1±2.3**
80	32.0±0.6	34.7±0.7	43.4±1.4	**46.9±2.5**	35.7±0.6	37.0±0.7	45.6±1.2	**48.0±2.4**
100	36.3±0.9	37.9±0.9	46.4±1.0	**48.0±2.0**	39.7±0.8	41.2±0.7	48.8±1.2	**50.2±2.1**
120	40.3±1.0	47.4±0.9	48.7±1.0	**51.5±1.6**	43.5±0.9	49.7±0.8	51.1±1.3	**53.6±1.6**
140	43.6±0.8	49.9±0.8	51.1±1.2	**52.8±1.6**	46.5±0.7	51.8±0.6	53.3±1.1	**54.9±1.4**
160	46.5±0.8	53.4±0.9	53.1±0.9	**55.4±1.7**	49.2±0.7	56.0±0.9	55.3±0.9	**57.3±1.5**
180	49.1±0.6	55.2±0.6	55.0±1.1	**56.9±1.5**	51.6±0.5	57.8±0.5	57.0±1.1	**58.7±1.5**
200	51.2±0.5	57.6±0.5	57.2±1.1	**59.0±1.4**	53.6±0.5	59.6±0.5	58.9±0.9	**60.6±1.5**

TABLE 3
Text categorization results on 20Newsgroups. (on the test data)

	Macro-F_1 (%)				Micro-F_1 (%)			
k	Random	Simple Margin	convex TED	MAED	Random	Simple Margin	convex TED	MAED
20	13.0±0.9	14.6±0.5	27.7±1.9	**34.1±2.5**	17.9±0.9	21.7±0.6	31.5±2.0	**36.3±2.4**
40	20.6±0.6	26.1±1.0	36.4±2.1	**41.5±1.8**	25.1±0.8	29.6±0.8	38.7±2.0	**42.6±2.1**
60	26.8±0.6	28.6±0.8	39.5±1.7	**42.4±2.3**	30.9±0.6	30.8±0.8	41.8±1.7	**43.8±2.3**
80	32.0±0.6	34.6±0.7	43.0±1.4	**46.3±2.5**	35.7±0.6	37.0±0.7	45.2±1.2	**47.5±2.5**
100	36.3±0.9	37.8±1.0	46.0±1.2	**47.5±2.1**	39.7±0.8	41.1±0.7	48.4±1.3	**49.7±2.1**
120	40.4±1.0	46.8±0.9	48.1±1.1	**50.8±1.6**	43.5±0.9	49.2±0.8	50.5±1.3	**52.9±1.6**
140	43.6±0.8	49.4±0.8	50.5±1.4	**52.1±1.6**	46.5±0.8	51.3±0.6	52.8±1.2	**54.2±1.5**
160	46.5±0.8	52.8±0.7	52.5±0.9	**54.8±1.6**	49.2±0.7	55.4±0.9	54.8±1.0	**56.8±1.5**
180	49.1±0.6	54.7±0.6	54.5±1.1	**56.3±1.3**	51.6±0.5	56.8±0.5	56.5±1.2	**58.2±1.5**
200	51.1±0.5	56.4±0.6	56.7±1.3	**58.4±1.5**	53.6±0.5	58.9±0.5	58.5±1.2	**60.1±1.5**

recall and F_1 measure [44]. Precision is the ratio of correct assignments by the classifier divided by the total number of the classifier's assignments. Recall is defined to be the ratio of correct assignments by the classifier divided by the total number of correct assignments. The F_1 measure combines recall (r) and precision (p) with an equal weight in the following form:

$$F_1(r,p) = \frac{2rp}{r+p}$$

These scores can be computed for the binary decisions on each individual category first and then be averaged over categories. Or, they can be computed globally over all the $n \times m$ binary decisions where m is the number of total test documents, and n is the number of categories in consideration. The former way is called *macro-averaging* and the latter way is called *micro-averaging*. It is understood that the micro-averaged scores tend to be dominated by the classifier's performance on common categories, and the macro-averaged scores are more influenced by the performance on rare categories. Providing both kinds of scores is more informative than providing either alone [44]. Another popular metric in our situation is AUC score, i.e., *area under the Receiver Operating Characteristic (ROC) curve*, which is used in [46]. We also report the AUC score in our experiment.

In each run of the experiments, an active learning method is applied to select a given number k of training examples, $k = \{5, 10, \cdots, 50\}$ on Reuters and RCV1 and $k = \{20, 40, \cdots, 200\}$ on 20Newsgroups, then a classifier is trained on these examples with their labels. The trained classifier is then used to predict the class labels of the remaining examples, and both Macro-F1 and Micro-F1 scores are computed based on the results.

TABLE 4
Test AUC score on 20Newsgroups.

	AUC score (%)			
k	Random	Simple Margin	convex TED	MAED
20	38.5±1.3	36.2±1.1	49.5±3.0	**54.5±2.5**
40	53.4±1.3	57.6±1.0	66.4±2.2	**70.6±2.3**
60	59.7±0.8	61.1±0.9	71.0±1.4	**73.6±2.0**
80	63.4±0.5	65.7±0.5	73.5±0.3	**76.5±1.1**
100	65.8±0.3	67.3±0.4	75.1±0.5	**76.6±0.9**
120	67.8±0.3	75.2±0.3	76.6±0.5	**79.5±0.7**
140	69.5±0.2	76.7±0.3	78.0±0.6	**79.9±0.7**
160	71.0±0.2	79.5±0.2	79.1±0.5	**82.0±0.7**
180	72.5±0.2	80.4±0.2	80.2±0.5	**82.7±0.7**
200	73.7±0.2	81.7±0.2	81.2±0.4	**83.5±0.7**

TABLE 7
Test AUC score on Reuters-21578.

	AUC score (%)			
k	Random	Simple Margin	convex TED	MAED
5	49.1±3.3	44.8±9.8	73.5±8.4	**84.1±9.7**
10	69.0±3.9	69.2±6.7	93.8±0.8	**96.8±0.3**
15	79.2±3.5	72.1±4.5	95.9±0.7	**97.4±0.4**
20	86.0±2.6	87.8±4.3	96.9±0.6	**97.9±0.5**
25	90.7±1.5	92.5±1.1	97.4±0.3	**98.2±0.2**
30	93.3±1.4	95.7±1.5	98.0±0.4	**98.5±0.3**
35	94.4±1.6	97.0±1.1	98.3±0.3	**98.5±0.3**
40	95.3±1.3	97.8±0.9	98.3±0.2	**98.5±0.2**
45	96.6±0.4	98.4±0.5	98.4±0.2	**98.6±0.2**
50	97.0±0.5	98.6±0.4	98.4±0.2	**98.8±0.2**

In order to randomize the experiments, in each run of experiments we restrict the training examples to be selected from a random candidate set of 50% of the total data. Strictly speaking, since the candidate set is available for all the active learning algorithms, the remaining 50% of the total data can be regarded as the test data. Thus, we reported the classification results on both unlabeled set (all the unlabeled data) and test set (the remaining 50% of the total data). For each combination of active learning method and a number k, we compute the mean and standard deviation based on 10 randomized experiments. The following four active learning methods are evaluated and compared:

- **Random Sampling** method uniformly selects examples as training data. We use this method as the baseline for active learning.
- **Simple Margin** method is proposed in [43]. This method selects the example closest to the current decision boundary of the classifier, which is a usual SVM using the hinge loss.
- **Convex TED** method is proposed in [46].
- **Manifold Adaptive Experimental Design** (MAED) method, as described in Section 3.4, is a new method proposed in this paper.

We note that all the methods use least-squares SVM (LSSVM) as the base classification method, except the Simple Margin method that uses hinge-loss SVM. In all the experiments we fix the parameter as $\lambda = 0.1$.

4.3 Text Categorization Results

In this subsection, we discuss the performance of the four different algorithms on text categorization. Before experimental comparison, it would be important to note that the algorithms Random Sampling, Convex TED and MAED are all *classifier-independent*, while the algorithm Simple Margin is *classifier-dependent*. For the former three algorithms, the data selection is performed globally. In other words, the selected data points will be used for all the binary classification tasks. However, for Simple Margin, since the active learning (data selection) process is dependent on the decision boundary, for each binary classification task we have to select k data points for labeling. In our experiments, four categories are used, thus Simple Margin may select maximally $4k$ data points, if there is no overlap. Moreover, since Simple Margin is classifier-dependent, it needs at least one example for each category to train the initial classifier. In our experiments, we randomly select one example from each category to train an initial SVM classifier for Simple Margin.

4.3.1 20 Newsgroups

We apply the above mentioned four algorithms to text categorization on 20Newsgroups. Given training size k, the average classification performance measured by Macro-F1 and Micro-F1 is reported in Table 2 (on all the unlabeled data) and Table 3 (on test data). As can be seen, our MAED algorithm outperforms the other three algorithms in all the cases. Random sampling performs the worst in most of the cases. As we have mentioned, Simple Margin uses much more labels than other algorithms. Even so, Convex TED outperforms Simple Margin in most of the cases.

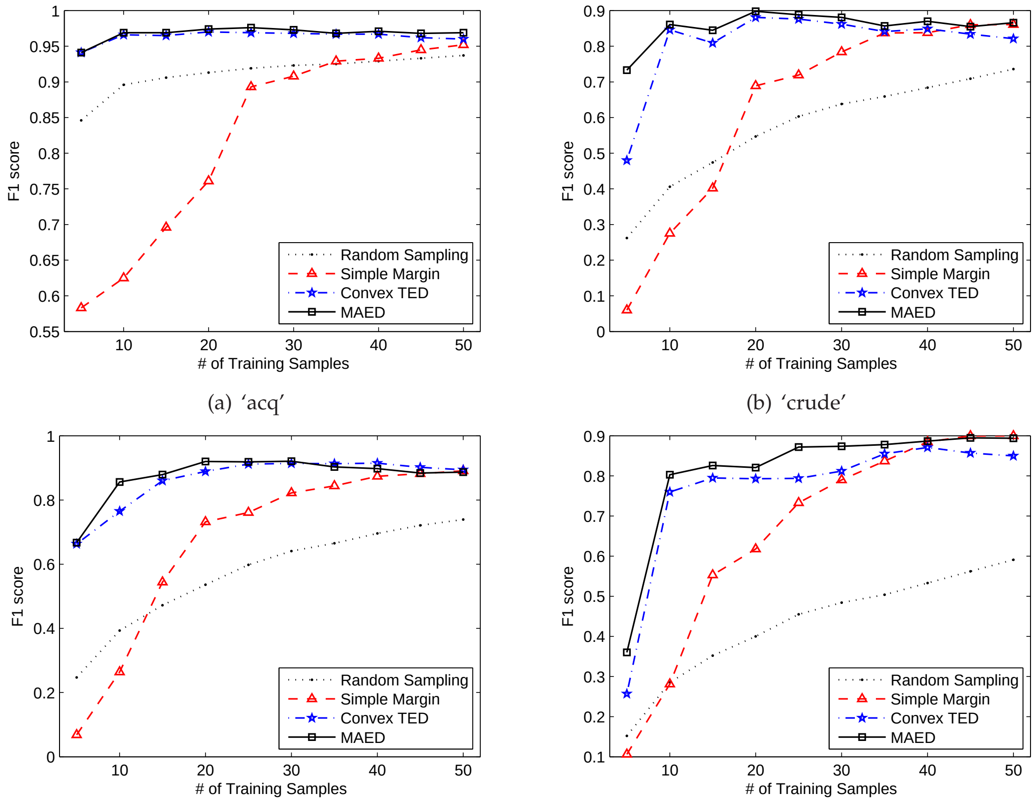
For all the compared algorithms, their classification accuracies increase with more training examples. Although our algorithm performs the best in the entire scope, it is worthwhile to note that it performs especially good when there is limited number of training examples. In practice, when only very small number of examples are selected, it would be possible that for some categories, there is no example selected at all. In this case, all the examples in those categories will be misclassified into other categories. Therefore, when there are only limited labeling resources available, the active learning performance is crucial for the ultimate classification results. As can be seen, when $k = 20$, our algorithm achieves 0.344 Macro-F1 score and 0.366 Micro-F1 score. To achieve comparable accuracy, Convex TED has to label 40 examples, Simple Margin has to label 80 examples for each binary classification task, and Random Sampling has to label 100 examples. For $k = 20$, Simple Margin performs even worse than Random Sampling. This result clearly shows that our algorithm can significantly reduce human labeling task. As more labels are used, the performance difference of the four algorithms gets smaller. Table 4 shows the AUC score of all the compared algorithms. We can get the similar conclusion.

TABLE 5
Text categorization results on Reuters-21578. (on the unlabeled data)

	Macro-F_1 (%)				Micro-F_1 (%)			
k	Random	Simple Margin	convex TED	MAED	Random	Simple Margin	convex TED	MAED
5	37.7±2.3	20.4±9.5	58.5±7.8	**67.5±7.7**	72.0±1.9	53.2±28.0	83.3±3.0	**84.8±1.9**
10	49.5±3.4	36.1±12.3	83.4±2.6	**87.2±1.0**	79.2±1.0	57.9±27.4	91.6±1.2	**93.1±0.5**
15	55.1±3.5	54.9±13.2	85.7±2.7	**88.0±0.8**	81.5±1.1	62.9±19.2	92.6±1.0	**93.6±0.5**
20	59.9±2.9	70.0±8.4	88.3±1.7	**90.3±1.6**	83.1±0.9	75.0±14.9	93.9±0.6	**94.9±0.8**
25	64.4±3.2	77.7±5.3	88.8±1.8	**91.4±1.0**	84.6±1.1	87.2±3.1	94.1±0.9	**95.3±0.5**
30	67.2±2.6	82.6±3.6	88.9±2.6	**91.2±1.2**	85.6±0.9	90.0±3.0	94.1±1.3	**95.1±0.7**
35	68.8±2.3	86.2±2.3	89.4±2.2	**90.1±1.3**	86.1±0.8	91.5±1.5	94.1±1.2	**94.4±0.7**
40	71.1±2.5	88.3±2.1	90.0±1.5	**90.7±0.9**	86.9±0.9	93.0±1.4	94.3±0.8	**94.7±0.6**
45	73.1±2.4	89.7±0.7	88.9±1.6	**90.0±1.2**	87.7±1.0	93.8±0.6	93.6±0.9	**94.3±0.7**
50	75.1±2.5	90.1±1.4	88.1±1.0	**90.4±1.7**	88.4±1.1	94.1±0.5	93.2±0.5	**94.5±1.0**

TABLE 6
Text categorization results on Reuters-21578. (on the test data)

	Macro-F_1 (%)				Micro-F_1 (%)			
k	Random	Simple Margin	convex TED	MAED	Random	Simple Margin	convex TED	MAED
5	37.6±2.3	19.5±10.0	57.6±7.9	**66.6±8.4**	72.2±1.8	52.8±28.3	82.9±3.1	**84.4±2.2**
10	49.4±3.4	35.9±12.7	83.3±3.2	**87.0±1.2**	79.3±1.2	57.9±27.6	91.6±1.4	**93.1±0.6**
15	55.0±3.5	55.0±13.2	85.7±2.7	**88.1±0.9**	81.6±1.3	62.9±19.2	92.7±1.0	**93.6±0.6**
20	59.7±2.9	69.8±8.7	88.2±2.1	**90.2±1.8**	83.2±1.2	75.0±15.1	93.9±0.7	**94.8±1.0**
25	64.2±3.1	77.5±5.4	88.4±1.9	**91.3±0.9**	84.7±1.3	87.1±3.1	94.0±0.9	**95.3±0.5**
30	67.0±2.6	82.7±4.0	89.0±2.8	**91.3±1.9**	85.7±1.2	90.1±3.2	94.2±1.3	**95.2±1.0**
35	68.6±2.2	86.4±2.5	89.7±2.3	**90.3±1.6**	86.2±1.0	91.6±1.6	94.3±1.2	**94.5±0.9**
40	70.8±2.5	88.6±2.3	90.5±1.7	**90.9±1.1**	87.0±1.1	93.2±1.4	94.5±0.9	**94.8±0.7**
45	72.8±2.5	90.1±0.9	89.4±1.7	**90.3±1.5**	87.7±1.2	94.0±0.7	93.9±0.9	**94.5±0.9**
50	74.8±2.7	90.5±1.6	88.7±1.2	**90.6±1.8**	88.4±1.4	94.3±0.6	93.5±0.7	**94.7±1.0**



(a) 'acq'

(b) 'crude'

(c) 'trade'

(d) 'money'

Fig. 2. Classification performance on different categories of Reuters-21578 data set.

4.3.2 Reuters-21578

The average text categorization performance measured by Micro-F1 and Macro-F1 on Reuters data set is reported in Table 5 (on all the unlabeled data) and Table 6 (on test data). Our algorithm MAED consistently outperforms the other three algorithms. Convex TED performs the second best in most of the cases. When the number of training examples is less than 25, Random Sampling performs better than or comparably to Simple Margin. For other cases, it performs worse than Simple Margin. However, it would be important to note that Simple Margin uses more training examples than the other algorithms.

For all the algorithms, the classification accuracies increase with more training examples. Similar to the results on 20Newsgroups, our MAED algorithm performs especially good when the training size is small. Particularly, when $k = 5$, MAED achieves 0.675 Macro-F1 score and 0.848 Micro-F1 score. To achieve comparable results, Random Sampling needs to label more than 25 examples and Simple Margin needs to label more than 20 examples. Convex TED performs comparably to MAED on this data set. Table 7 shows the AUC score of all the compared algorithms. We can get the similar conclusion.

Besides the averaged performance comparison, we also show the classification results on each individual binary classification task in Fig. 2. As can be seen, for all the categories, our MAED algorithm outperforms the other three algorithms. The performance improvement of our algorithm is especially significant when k is small. Convex TED also performs very well, especially on the categories ‘acq’ and ‘trade’. For some categories Simple Margin performs worse than Random Sampling when $k \leq 20$. This is probably because that Simple Margin is classifier-dependent. When the labeled examples is limited, the initially estimated boundary may not be accurate enough.

4.3.3 RCV1

The average text categorization performance measured by Micro-F1 and Macro-F1 on RCV1 data set is reported in Table 8 (on all the unlabeled data) and Table 9 (on test data). We have the similar experimental results as the previous two data sets. Our algorithm MAED consistently outperforms the other three algorithms. Convex TED performs the second best in most of the cases. When the number of training examples is less than 30, Random Sampling performs better than or comparably to Simple Margin. For other cases, it performs worse than Simple Margin.

Similar to the results on 20Newsgroups and Reuters, our MAED algorithm performs especially good when the training size is small. Particularly, when $k = 5$, MAED achieves 0.626 Macro-F1 score and 0.658 Micro-F1 score. To achieve comparable results, Random Sampling needs to label more than 20 examples and Simple Margin needs to label more than 25 examples. Convex TED performs comparably to MAED on this data set. Table 10 shows the AUC score of all the compared algorithms. We can get the similar conclusion.

TABLE 10
Test AUC score RCV1.

	AUC score (%)			
k	Random	Simple Margin	convex TED	MAED
5	56.2±3.1	39.7±2.1	73.9±1.7	**78.7±1.3**
10	73.4±1.6	58.1±1.7	84.9±0.5	**87.5±0.6**
15	80.0±0.9	73.0±2.3	86.2±0.9	**89.3±0.9**
20	83.4±0.6	77.5±4.2	87.4±1.4	**91.2±1.4**
25	85.5±0.5	83.1±3.0	89.1±1.0	**92.2±0.7**
30	87.0±0.5	87.9±1.6	91.0±0.9	**92.7±0.7**
35	88.3±0.4	89.7±1.4	92.5±0.9	**93.3±0.8**
40	89.4±0.4	91.8±1.1	92.6±0.8	**93.5±1.3**
45	90.3±0.4	92.6±1.0	93.4±0.8	**93.6±1.1**
50	90.9±0.4	92.9±0.8	93.7±0.4	**94.2±0.8**

Fig. 3 plots the text categorization performance vs. the number of training examples on each binary classification task. Again, MAED consistently outperforms the other three algorithms on all the four categories.

We have so far compared the four algorithms on 20Newsgroups, Reuters-21578 and RCV1 corpora. Clearly, our MAED algorithm yields relatively more impressive results on 20Newsgroups. Since the 20Newsgroups data set is more difficult than Reuters-21578 and RCV1, it seems that our algorithm is more suitable for the difficult data sets.

4.4 Parameter Selection

An essential parameter in our MAED model is the regularization parameter λ in manifold adaptive kernel construction. MAED boils down to the original TED when $\lambda = 0$. In our previous experiments, we simply set $\lambda = 0.1$. Figure 4 shows how the average performance of MAED varies with the λ.

As we can see, the performance of MAED is very stable with respect to the parameter λ. MAED achieves consistently good performance with the λ varying from 0.001 to 0.1 on all the three data sets.

4.5 Experiments on Incremental Active Learning

Our previous experiment mainly examines the performances of different active learning algorithms on their “batch mode”, *i.e.*, there is no labeled points at the beginning and the active learning algorithms are required to select relatively small number of samples. In reality, another more realistic setting could be that a certain number of n training points are already available (based on human expertise), and are then complemented by additional k (smaller than n) points based on active learning. This setting can be called “incremental mode”. In this subsection, we will examine the performances of different active learning algorithms in the incremental mode.

The 20Newsgroups data set is used in this experiment. We use the “bydate” split (which has around 60% training data and 40% testing data) provided on the home page of 20Newsgroups[3]. The experimental setting is as follows:

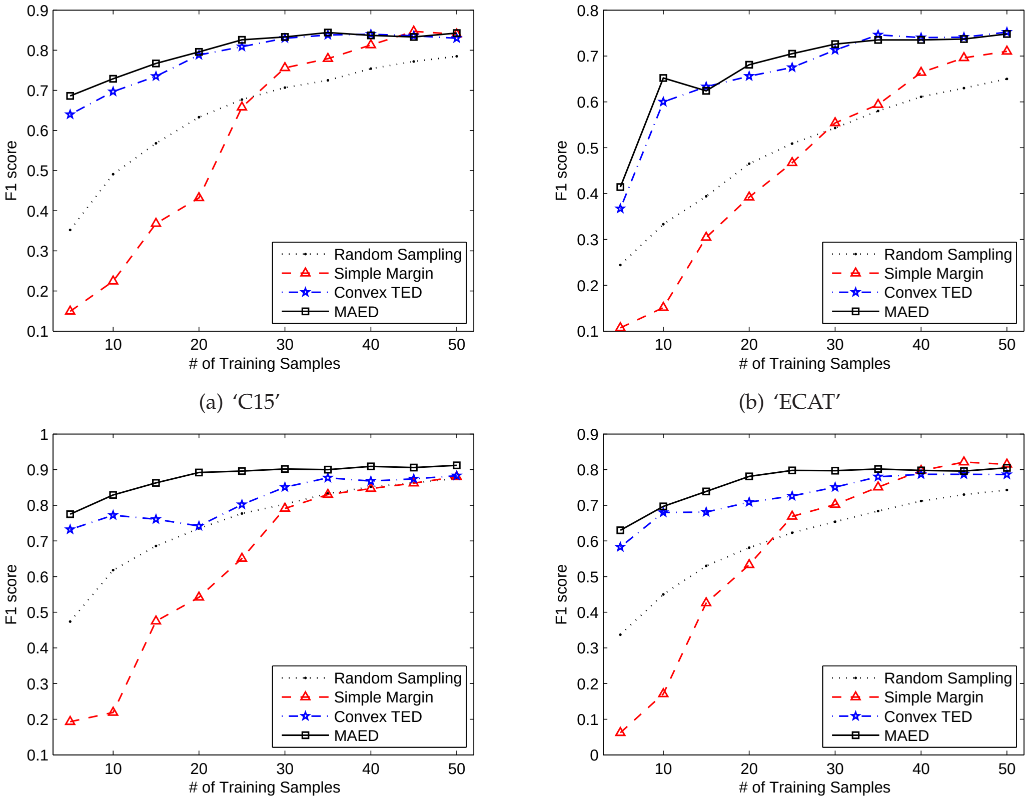
[3] http://people.csail.mit.edu/jrennie/20Newsgroups/

TABLE 8
Text categorization results on RCV1. (on the unlabeled data)

	Macro-F_1 (%)				Micro-F_1 (%)			
k	Random	Simple Margin	convex TED	MAED	Random	Simple Margin	convex TED	MAED
5	35.2±2.4	12.8±2.6	58.1±1.8	**62.6±1.9**	43.2±2.2	26.6±4.6	61.0±2.2	**65.8±2.3**
10	47.3±2.1	19.1±5.5	68.7±2.6	**72.7±1.9**	52.7±2.1	30.3±5.0	69.6±3.2	**73.4±2.2**
15	54.5±1.6	39.3±7.0	70.2±3.4	**74.8±1.6**	58.7±1.4	45.4±7.0	70.3±3.6	**76.3±1.9**
20	60.3±1.8	47.5±14.1	72.4±4.4	**78.8±2.3**	63.5±1.6	51.2±12.5	72.0±4.6	**80.0±1.9**
25	64.7±1.4	61.1±8.9	75.3±3.7	**80.6±1.9**	67.3±1.2	62.1±9.0	75.0±4.3	**81.6±1.5**
30	67.7±1.3	70.1±4.7	78.6±3.4	**81.4±1.7**	70.0±1.1	71.8±4.7	78.6±3.8	**82.3±1.5**
35	70.5±1.3	73.9±4.9	81.0±2.4	**82.0±1.7**	72.6±1.2	75.4±4.3	81.1±2.6	**82.7±1.7**
40	73.2±1.2	78.0±3.9	80.9±2.5	**82.0±3.1**	74.9±1.0	79.5±3.2	81.0±2.7	**82.7±2.9**
45	74.9±1.0	80.6±3.3	81.0±2.6	**82.2±2.6**	76.4±0.9	80.7±3.1	82.2±2.8	**82.6±2.5**
50	76.3±1.2	81.2±2.9	81.3±1.0	**82.7±1.9**	77.7±1.0	81.2±2.4	82.5±1.2	**83.4±1.7**

TABLE 9
Text categorization results on RCV1. (on the test data)

	Macro-F_1 (%)				Micro-F_1 (%)			
k	Random	Simple Margin	convex TED	MAED	Random	Simple Margin	convex TED	MAED
5	35.2±2.4	13.2±2.6	59.1±1.7	**62.5±1.9**	43.2±2.2	26.1±4.5	59.7±2.1	**65.6±2.2**
10	47.3±2.2	19.3±5.2	69.9±2.6	**72.7±1.9**	52.7±2.2	30.3±5.1	69.7±3.2	**73.4±2.1**
15	54.5±1.7	39.4±7.0	70.4±3.4	**74.8±1.6**	58.7±1.4	45.3±7.0	70.2±3.7	**76.2±1.9**
20	60.4±1.8	47.4±13.1	72.3±4.5	**78.8±2.4**	63.5±1.6	51.2±12.3	72.0±4.6	**80.0±2.1**
25	64.7±1.3	60.5±8.9	75.1±3.6	**80.8±1.9**	67.3±1.1	62.2±8.9	74.8±4.1	**81.8±1.5**
30	67.7±1.3	70.0±4.7	78.5±3.4	**81.5±1.7**	70.0±1.1	71.7±4.7	78.5±3.8	**82.4±1.5**
35	70.6±1.4	73.8±4.9	81.1±2.3	**82.2±1.7**	72.6±1.2	75.4±4.2	81.2±2.5	**82.9±1.8**
40	73.3±1.3	77.8±3.8	80.9±2.5	**82.0±3.1**	74.9±1.0	79.6±3.3	81.0±2.7	**82.8±2.9**
45	74.9±1.0	80.6±3.3	81.1±2.8	**82.0±2.6**	76.4±0.8	80.8±3.0	82.3±3.0	**82.7±2.5**
50	76.3±1.1	81.2±2.8	81.4±1.1	**82.8±2.1**	77.7±0.9	81.3±2.3	82.6±1.3	**83.5±1.9**



(a) ‘C15’

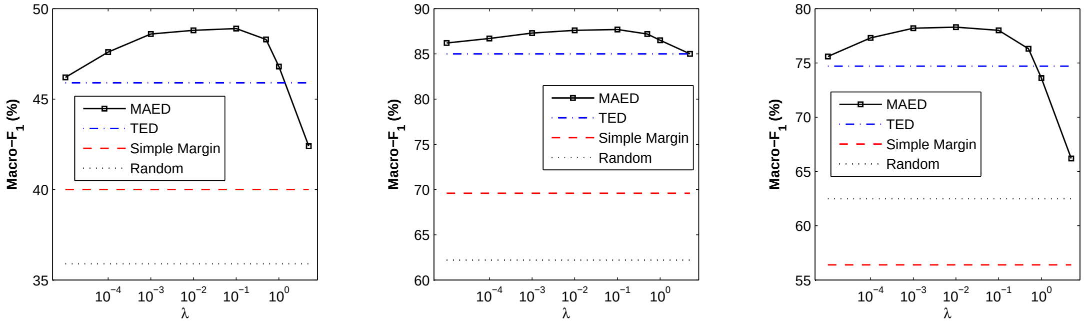
(b) ‘ECAT’

(c) ‘GCAT’

(d) ‘MCAT’

Fig. 3. Classification performance on different categories of RCV1 data set.

1) We randomly select part of the data points from the training set (1%∼90% of the training set) to form the labeled set. A linear SVM classifier is trained on the labeled set and its performance on the test set is



(a) 20Newsgroups (b) Reuters (c) RCV1

Fig. 4. The performance of MAED vs. parameter λ. The MAED is very stable with respect to the parameter λ. It achieves consistent good performance with the λ varying from 0.001 to 0.1.

TABLE 11
Test Macro-F_1 score on 20Newsgroup (%)

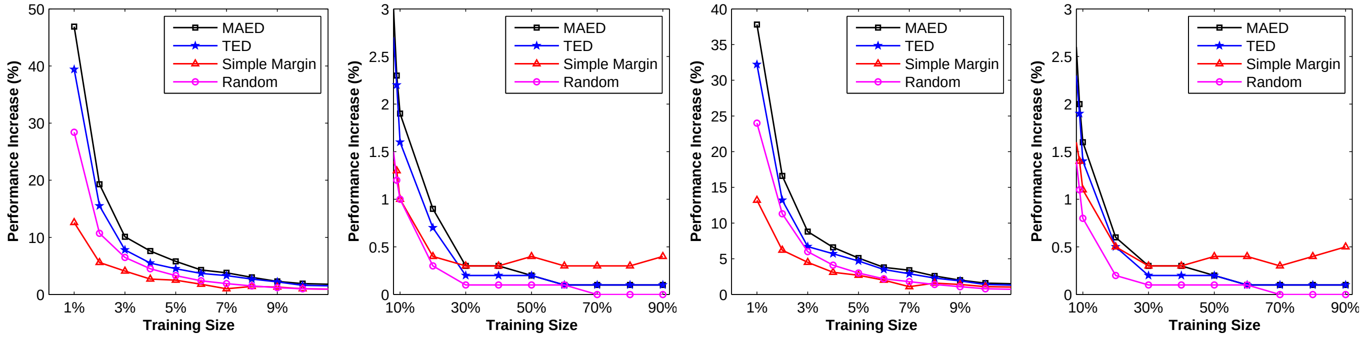
Size of the label set		Baseline	Random		Simple Margin		convex TED		MAED	
113	1%	36.0±3.8	46.2±2.0	28.4%	40.5±2.4	12.6%	50.2±2.2	39.4%	**52.9±1.9**	**46.9%**
225	2%	50.0±3.8	55.4±2.2	10.7%	52.8±2.8	5.6%	57.8±1.8	15.5%	**59.7±2.0**	**19.3%**
338	3%	57.5±2.2	61.3±1.5	6.5%	59.9±2.4	4.1%	62.0±1.4	7.8%	**63.3±1.4**	**10.1%**
451	4%	61.4±2.1	64.2±1.6	4.5%	63.0±2.3	2.7%	64.8±1.2	5.5%	**66.1±1.1**	**7.6%**
563	5%	64.0±2.0	66.1±1.4	3.3%	65.6±1.9	2.5%	66.9±1.1	4.5%	**67.7±1.1**	**5.8%**
676	6%	66.3±1.7	67.9±1.2	2.4%	67.5±1.3	1.8%	68.7±1.1	3.7%	**69.1±1.0**	**4.3%**
789	7%	67.7±1.2	69.0±0.9	1.9%	68.4±1.1	1.0%	69.9±0.9	3.3%	**70.3±0.7**	**3.8%**
902	8%	69.3±0.8	70.4±0.6	1.5%	70.3±0.6	1.4%	71.2±0.6	2.7%	**71.4±0.7**	**3.0%**
1014	9%	70.4±0.7	71.3±0.5	1.2%	71.4±0.6	1.3%	72.0±0.6	2.2%	**72.0±0.6**	**2.3%**
1127	10%	71.4±0.5	72.1±0.5	1.0%	72.1±0.5	1.0%	72.5±0.5	1.6%	**72.7±0.5**	**1.9%**
2254	20%	76.5±0.5	76.7±0.4	0.3%	76.8±0.4	0.4%	77.0±0.3	0.7%	**77.2±0.3**	**0.9%**
3381	30%	78.3±0.3	78.4±0.3	0.1%	**78.6±0.2**	**0.3%**	78.5±0.3	0.2%	**78.6±0.3**	**0.3%**
4508	40%	79.4±0.3	79.5±0.3	0.1%	**79.6±0.3**	**0.3%**	79.6±0.3	0.2%	**79.6±0.2**	**0.3%**
5635	50%	80.2±0.3	80.2±0.2	0.1%	**80.5±0.2**	**0.4%**	80.3±0.3	0.2%	80.3±0.3	0.2%
6761	60%	80.8±0.2	80.9±0.2	0.1%	**81.1±0.3**	**0.3%**	80.9±0.2	0.1%	80.9±0.2	0.1%
7888	70%	81.3±0.1	81.3±0.2	0.0%	**81.5±0.3**	**0.3%**	81.3±0.1	0.1%	81.3±0.1	0.1%
9015	80%	81.6±0.2	81.7±0.2	0.0%	**81.9±0.2**	**0.3%**	81.8±0.2	0.1%	81.8±0.2	0.1%
10142	90%	81.8±0.1	81.8±0.1	0.0%	**82.1±0.1**	**0.4%**	81.9±0.1	0.1%	81.9±0.1	0.1%
11269	100%	82.1								

TABLE 12
Test Micro-F_1 score on 20Newsgroup (%)

Size of the label set		Baseline	Random		Simple Margin		convex TED		MAED	
113	1%	39.6±3.5	49.1±1.8	24.0%	44.8±1.9	13.2%	52.4±2.1	32.2%	**54.6±1.9**	**37.8%**
225	2%	52.6±3.8	58.5±2.2	11.3%	55.8±2.7	6.2%	59.5±1.7	13.2%	**61.3±2.0**	**16.6%**
338	3%	59.6±2.2	63.1±1.5	6.0%	62.2±2.4	4.5%	63.5±1.4	6.7%	**64.8±1.4**	**8.8%**
451	4%	63.4±2.0	66.0±1.5	4.1%	65.4±2.0	3.1%	67.0±1.1	5.7%	**67.6±1.0**	**6.6%**
563	5%	65.9±1.8	67.8±1.3	3.0%	67.6±1.7	2.7%	68.9±1.1	4.7%	**69.2±1.1**	**5.1%**
676	6%	68.1±1.5	69.6±1.0	2.2%	69.4±1.1	2.0%	70.5±1.0	3.5%	**70.6±0.9**	**3.8%**
789	7%	69.4±1.1	70.6±0.8	1.8%	70.2±1.2	1.1%	71.5±0.9	2.9%	**71.8±0.7**	**3.4%**
902	8%	70.9±0.7	71.9±0.6	1.4%	72.0±0.5	1.6%	72.5±0.6	2.3%	**72.8±0.6**	**2.6%**
1014	9%	72.0±0.6	72.8±0.5	1.1%	73.0±0.6	1.4%	73.4±0.6	1.9%	**73.5±0.5**	**2.0%**
1127	10%	73.0±0.5	73.6±0.5	0.8%	73.7±0.4	1.1%	74.0±0.5	1.4%	**74.2±0.5**	**1.6%**
2254	20%	77.6±0.4	77.8±0.3	0.2%	78.0±0.4	0.5%	78.0±0.3	0.5%	**78.1±0.3**	**0.6%**
3381	30%	79.4±0.3	79.5±0.3	0.1%	**79.6±0.2**	**0.3%**	79.5±0.3	0.2%	**79.6±0.3**	**0.3%**
4508	40%	80.4±0.3	80.4±0.3	0.1%	**80.6±0.3**	**0.3%**	80.5±0.3	0.2%	**80.6±0.2**	**0.3%**
5635	50%	81.0±0.3	81.1±0.3	0.1%	**81.4±0.2**	**0.4%**	81.2±0.3	0.2%	81.2±0.3	0.2%
6761	60%	81.7±0.2	81.8±0.2	0.1%	**82.0±0.3**	**0.4%**	81.8±0.2	0.1%	81.8±0.2	0.1%
7888	70%	82.1±0.1	82.1±0.1	0.0%	**82.4±0.3**	**0.3%**	82.2±0.1	0.1%	82.2±0.1	0.1%
9015	80%	82.4±0.2	82.5±0.1	0.0%	**82.8±0.2**	**0.4%**	82.5±0.2	0.1%	82.6±0.2	0.1%
10142	90%	82.6±0.1	82.6±0.1	0.0%	**83.0±0.1**	**0.5%**	82.7±0.1	0.1%	82.7±0.1	0.1%
11269	100%	82.9								

reported as *Baseline*.

2) Each active learning algorithm is asked to select $k = 100$ data points from the training set in addition to the existing labeled set. The linear SVM classifier is then trained on the new labeled set (the original labeled set plus 100 new labeled points) and its performance on the test set is recorded as the performance of the active learning algorithm.



(a) MacroF_1

(b) MicroF_1

Fig. 5. Classification performance of different methods vs. the size of the initial label set on 20Newsgroups data set.

3) The above two steps are repeated 10 times and we report the averages and the standard deviations.

The results are shown in Table 11 and 12. For each active learning method, we also compute the relative performance increase comparing to the Baseline. We also plot the results in Figure 5. These result tables and figures clearly show that

- With the additional 100 labeled points (no matter which active learning method is used), the classifier generally becomes better. The 100 points selected by different active learning methods made different amount of contributions in improving the classifier.
- When the size of the initial labeled set is smaller or equal than 2,254 (20% of the training set), MAED selects the 100 most informative data points (achieved best classification performance). When the size of the initial labeled set is larger or equal than 5,635 (50% of the training set), Simple Margin selects the 100 most informative data points. When the size of the initial labeled set is 3,381 (30% of the training set) or 4,508 (40% of the training set), MAED and Simple Margin have the similar performances.
- When the size of the initial labeled set is smaller or equal than 789 (7% of the training set), even the random selection is better than Simple Margin.

As we discussed before, Simple Margin and MAED represent two directions of active learning research. Simple Margin selects the most uncertain data points given the previously trained model and MAED selects the most representative points. The advantages and disadvantages of these two directions can be clearly seen from our experimental results:

- When the size of the initial labeled set is small, the methods which select the most representative points are usually better than the methods which select the most uncertain data points. This is because the initial trained model is not very accurate given a small number of labeled points. On the other hand, by selecting the most representative points, those methods can greatly explore the entire data space.
- When the size of the initial labeled set is large, the methods which select the most uncertain data points can outperform the methods which select the most representative points. With a large amount of labeled points, the initial model can be relatively accurate. Thus, those most uncertain points given by the initial model can provide most amount of new information.
- This suggests a natural way to combine these two active learning directions: One can select the most representative data points if the size of the initial labeled set is small. As the size of the labeled set increases, one can switch to the methods that select the most uncertain data points. In our case, we can use MAED when the size of the labeled points is small and switch to Simple Margin as the size of the labeled points becomes larger. How to decide the switching point is an interesting and important question which remains to be explored in the future.

5 Conclusion and Future Work

We have introduced a novel active learning algorithm for text categorization called Manifold Adaptive Experimental Design (MAED). Unlike most of previous active learning approaches which explore either Euclidean or data-independent nonlinear structure of the data space, our proposed approach explicitly takes into account the intrinsic manifold structure. The local geometry of the data is captured by a nearest neighbor graph. The graph Laplacian is incorporated into the manifold adaptive kernel space in which active learning is then performed. Our proposed algorithm has shown good performance for text categorization on 20Newsgroup, Reuters-21578 and RCV1, especially when only a small number of examples can be labeled.

There are several problems that need to be investigated in the future. First, as the computational complexity of all the kernel based techniques scales with the number of data points, our method may not be applied to large-scale data sets. In this situation, one may apply clustering techniques such as K-means to group the data points into clusters and select some representative points from each clusters. Our method is then applied only to the representative points. Second, in this work the number of queries (k) is pre-given. Another natural scenario is that the acceptable error rate is fixed and the goal is to minimize the number of queries.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grants 60905001 and 90920303, National Basic Research Program of China (973 Program) under Grant 2011CB302206 and Fundamental Research Funds for the Central Universities. Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] R. Angelova and G. Weikum. Graph-based text classification: Learning from your neighbors. In *Proc. the 29th International Conference on Research and Development in Information Retrieval*, Seattle, Washington, 2006. 1

[2] A. C. Atkinson and A. N. Donev. *Optimum Experimental Designs, with SAS*. Oxford University Press, 2007. 1, 2

[3] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, pages 585–591. 2001. 1

[4] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006. 1, 3

[5] D. Cai. *Spectral Regression: A Regression Framework for Efficient Regularized Subspace Learning*. PhD thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, May 2009. 4

[6] D. Cai, X. He, X. Wu, and J. Han. Non-negative matrix factorization on manifold. In *Proc. Int. Conf. on Data Mining (ICDM'08)*, 2008. 2

[7] D. Cai, X. He, W. V. Zhang, and J. Han. Regularized locality preserving indexing via spectral regression. In *Proceedings of the 16th ACM conference on Conference on information and knowledge management (CIKM'07)*, pages 741–750, 2007. 2

[8] D. Cai, Q. Mei, J. Han, and C. Zhai. Modeling hidden topics on document manifold. In *Proceeding of the 17th ACM conference on Information and knowledge management (CIKM'08)*, pages 911–920, 2008. 2

[9] D. Cai, X. Wang, and X. He. Probabilistic dyadic data analysis with local and global consistency. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML'09)*, pages 105–112, 2009. 1

[10] O. Chapelle. Active learning for parzen window classifier. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, 2005. 1

[11] F. R. K. Chung. *Spectral Graph Theory*, volume 92 of *Regional Conference Series in Mathematics*. AMS, 1997. 3

[12] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996. 1

[13] S. Dasgupta and D. Hsu. Hierarchical sampling for active learning. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 208–215, 2008. 2

[14] A. Dayanik, D. D. Lewis, D. Madigan, V. Menkov, and A. Genkin. Constructing informative prior distributions from domain knowledge in text classification. In *Proc. the 29th International Conference on Research and Development in Information Retrieval*, Seattle, Washington, 2006. 1

[15] P. Flaherty, M. I. Jordan, and A. P. Arkin. Robust design of biological experiments. In *Advances in Neural Information Processing Systems 18*, Vancouver, Canada, 2005. 2

[16] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997. 1

[17] B. Gao, G. Feng, T. Qin, Q.-S. Cheng, T.-Y. Liu, and W.-Y. Ma. Hierarchical taxonomy preparation for text categorization using consistent bipartite spectral graph copartitioning. *IEEE Transactions on Knowledge and Data Engineering*, 17(9):1263–1273, September 2005. 1

[18] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 1735–1742, 2006. 1

[19] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag, 2001. 3

[20] X. He, D. Cai, H. Liu, and W.-Y. Ma. Locality preserving indexing for document representation. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'04)*, pages 96–103, 2004. 2

[21] X. He, W. Min, D. Cai, and K. Zhou. Laplacian optimal design for image retrieval. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'07)*, 2007. 2

[22] S. C. Hoi, R. Jin, and M. R. Lyu. Large-scale text categorization by batch mode active learning. In *Proc. the 15th International Conference on World Wide Web*, Edinburgh, Scotland, 2006. 1

[23] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142, 1998. 4

[24] T. Joachims. Transductive inference for text classification using support vector machines. In *International Conference on Machine Learning (ICML)*, pages 200–209, Bled, Slowenien, 1999. 2

[25] T. Joachims. Transductive learning via spectral graph prtitioning. In *International Conference on Machine Learning (ICML)*, pages 290–297, 2003. 1, 2

[26] J. M. Lee. *Introduction to Smooth Manifolds*. Springer-Verlag New York, 2002. 1

[27] D. D. Lewis, Y. Yang, T. G. Rose, G. Dietterich, F. Li, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004. 5

[28] N. Loeff, D. Frsyth, and D. Ramachandran. Manifoldboost: Stagewise function approximation for fully-, semi- and un-supervised learning. In *Proc. International Conference on Machine Learning*, Helsinki, Finland, 2005. 3

[29] A. McCallum and K. Nigam. Employing em in pool-based active learning for text classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 359–367, 1998. 2

[30] A. Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, Cambridge, MA, 2001. 1

[31] P. Niyogi, S. Smale, and S. Weinberger. Finding the homology of submanifolds with high confidence from random samples. Technical report tr-2004-08, Department of Computer Science, University of Chicago, 2004. 3

[32] H. Raghavan and J. Allan. An interative algorithm for asking and incorporating feature feedback into support vector machines. In *Proc. the 30th International Conference on Research and Development in Information Retrieval*, Amsterdam, The Netherlands, 2007. 1

[33] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. 1

[34] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 441–448, 2001. 2

[35] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *The International Conference on Machine Learning*, 2000. 2

[36] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2002. 3, 4

[37] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. 1, 2

[38] B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1069–1078, 2008. 2

[39] H. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the ACMWorkshop on Computational Learning Theory*, pages 287–294, 1992. 2

[40] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *Proc. 2005 Int. Conf. Machine Learning (ICML'05)*, 2005. 1, 2, 3

[41] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. 1

[42] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international*

conference on Multimedia, pages 107–118, 2001. 2

[43] S. Tong and D. Koller. Support vector machine active learning with application to text classification. *Journal of Machine Learning Research*, 2:45–66, 2001. 1, 2, 7

[44] Y. Yang. An evaluation of statistical approaches to text categorization. *Jounal of Information Retrieval*, 1(1/2):67–88, 1999. 1, 4, 6

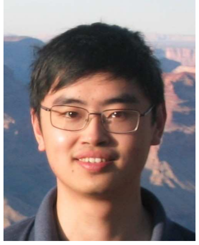
[45] K. Yu, J. Bi, and V. Tresp. Active learning via transductive experimental design. In *Proceedings of the* 23^{rd} *International Conference on Machine Learning*, Pittsburgh, PA, 2006. 1, 2, 3

[46] K. Yu, S. Zhu, W. Xu, and Y. Gong. Non-greedy active learning for text categorization using convex transductive experimental design. In *Proc. 2008 International Conference on Research and Development in Information Retrieval*, Singpore, 2008. 1, 3, 5, 6, 7

[47] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, 2003. 1

[48] X. Zhu, J. Lafferty, and Z. Ghahramani. Combining active learning and semisupervised learning using gaussian fields and harmonic functions. In *Proceedings of the ICML Workshop on the Continuum from Labeled to Unlabeled Data*, pages 58–65, 2003. 2

Deng Cai is an Associate Professor in the State Key Lab of CAD&CG, College of Computer Science at Zhejiang University, China. He received the PhD degree in computer science from University of Illinois at Urbana Champaign in 2009. Before that, he received his Bachelor's degree and a Master's degree from Tsinghua University in 2000 and 2003 respectively, both in automation. His research interests include machine learning, data mining and information retrieval.



Xiaofei He received the BS degree in Computer Science from Zhejiang University, China, in 2000 and the Ph.D. degree in Computer Science from the University of Chicago, in 2005. He is a Professor in the State Key Lab of CAD&CG at Zhejiang University, China. Prior to joining Zhejiang University, he was a Research Scientist at Yahoo! Research Labs, Burbank, CA. His research interests include machine learning, information retrieval, and computer vision.