Active Learning Based on Locally Linear Reconstruction

Lijun Zhang, *Student Member*, *IEEE*, Chun Chen, *Member*, *IEEE*, Jiajun Bu, *Member*, *IEEE*, Deng Cai, *Member*, *IEEE*, Xiaofei He, *Senior Member*, *IEEE*, and Thomas S. Huang, *Life Fellow*, *IEEE*

Abstract—We consider the active learning problem, which aims to select the most representative points. Out of many existing active learning techniques, optimum experimental design (OED) has received considerable attention recently. The typical OED criteria minimize the variance of the parameter estimates or predicted value. However, these methods see only global euclidean structure, while the local manifold structure is ignored. For example, I-optimal design selects those data points such that other data points can be best approximated by linear combinations of all the selected points. In this paper, we propose a novel active learning algorithm which takes into account the local structure of the data space. That is, each data point should be approximated by the linear combination of only its neighbors. Given the local reconstruction coefficients for every data point and the coordinates of the selected points, a transductive learning algorithm called Locally Linear Reconstruction (LLR) is proposed to reconstruct every other point. The most representative points are thus defined as those whose coordinates can be used to best reconstruct the whole data set. The sequential and convex optimization schemes are also introduced to solve the optimization problem. The experimental results have demonstrated the effectiveness of our proposed method.

Index Terms—Active learning, experimental design, local structure, reconstruction.

1 Introduction

In many real-word applications, there are huge volumes of unlabeled data, but the labels are usually difficult to get and expensive. Semi-supervised learning [1], [2], [3] addresses this problem by exploring additional information contained in the unlabeled data. Active learning reduces the labeling cost in a complementary way by querying the labels of the most informative points. Thus, instead of being a passive recipient of data to be processed, the active learner has the ability to control what data are added to its training set [4]. In this way, we expect that the active learner can achieve high accuracy using as few labeled points as possible [5].

The main challenge in active learning is how to evaluate the informativeness of the unlabeled points. One of the most widely used principles is *uncertainty sampling*. That is, the active learner queries those points whose predicted labels are most uncertain using the current trained model. This principle has been applied to logistic regression [6], support vector machines [7], nearest neighbor classifiers [8], [9], etc. Other popular active learning principles include *query by committee* [10], [11], *estimated error reduction* [12], [13], and *variance reduction* [4], [14].

The principle of *variance reduction* is derived from *Optimum Experimental Design* (OED) [14]. 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 OED is concerned with the design of experiments that are expected to minimize variances of a parameterized model [14], [15], [16], [17]. There are two types of selection criteria for OED. One type is to choose data points to minimize the variance of the model's parameters, which results in D, A, and E-optimal Design. The other is to minimize the variance of the prediction value, which results in I and G-optimal Design.

Recently, Yu et al. have proposed Transductive Experimental Design (TED) [16], which has yielded impressive results. TED is fundamentally based on I-optimal design but evaluates the average predictive variance over one test set that is given beforehand. It has been shown that finding those points which minimize the average predictive variance of the estimated function is equivalent to finding those points such that other points can be best approximated by linear combinations of the selected points. TED is a global method in the sense that each data point is linearly reconstructed by using *all* of the selected data points, no matter how far away the selected data points are from the point to be reconstructed.

In reality, the high-dimensional data may not be uniformly distributed in the whole ambient space. Instead, recent studies [18], [19], [20], [21] have shown that naturally occurring data may reside on a lower dimensional submanifold which is embedded in the high-dimensional ambient space. However, previous approaches such as TED fail to take into account this manifold structure. Given

- *L. Zhang, C. Chen, and J. Bu are with the Zhejiang Provincial Key Laboratory of Service Robot, College of Computer Science, Cao Guangbiao Building, Yuquan Campus, Zhejiang University, Hangzhou 310027, China. E-mail: {zljzju, chenc, bjj}@zju.edu.cn.*
- *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 310027, China. E-mail: {dengcai, xiaofeihe}@cad.zju.edu.cn.*
- *T.S. Huang is with the Beckman Institute for Advanced Sciences and Technology, University of Illinois at Urbana Champaign, 405 North Mathews Ave., Urbana, IL 61801. E-mail: huang@ifp.uiuc.edu.*

Manuscript received 29 Jan. 2010; revised 26 Aug. 2010; accepted 25 Nov. 2010; published online 28 Jan. 2011.
Recommended for acceptance by J. Winn.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2010-01-0069.
Digital Object Identifier no. 10.1109/TPAMI.2011.20.

a data point, it is more reasonable to reconstruct it by using only its nearest neighbors [18].

In this paper, we propose a novel active learning algorithm which selects the most representative points with respect to the intrinsic geometrical structure of the data. Inspired by Locally Linear Embedding (LLE) [18], we assume that each data point and its neighbors lie on or close to a locally linear patch of the manifold. Then, the manifold structure is characterized by the linear coefficients that reconstruct each data point from its neighbors. A transductive learning algorithm called Locally Linear Reconstruction (LLR) is proposed to reconstruct the whole data set by using the given local reconstruction coefficients for every data point and the coordinates of the selected points. The most representative points are therefore defined as those whose coordinates can be used to best reconstruct the whole data set. A sequential optimization scheme and a convex relaxation are proposed to solve the optimization problem.

The outline of the paper is as follows: In Section 2, we review the related work in experimental design. Our proposed active learning algorithm (LLR_{Active}) is introduced in Section 3. In Section 4, we propose two computational schemes to solve the optimization problem. Experiments are presented in Section 5. Finally, we provide some concluding remarks and suggestions for future work in Section 6.

Notation. Capital letters (e.g., M) are used to denote matrices. For a given matrix M, we denote its ith column by M_{*i} and its ith row by M_{i*}. Script capital letters (e.g., $\mathcal{X}$) are used to denote ordinary sets. Blackboard bold capital letters (e.g., $\mathbb{R}$) are used to denote number sets. Small letters (e.g., λ) are used to denote scalars. Bold small letters (e.g., $\boldsymbol{\lambda}$) are used to denote vectors. We use $\mathbf{x}_i$ to denote both the ith point and its coordinate (a column vector).

2 Related Work

As described, the work most related to our proposed approach is optimum experimental design. In this section, we will briefly describe the generic active learning problem and then provide a review of the conventional experimental design criteria and the recently proposed Transductive Experimental Design algorithm.

2.1 The Active Learning Problem

The generic problem of active learning is the following. Given a set of points $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ in $\mathbb{R}^d$, find a subset $\mathcal{Z} = \{\mathbf{x}_{s_1}, \ldots, \mathbf{x}_{s_k}\} \subseteq \mathcal{X}$ which contains the most informative points. That is, if the points $\mathbf{x}_{s_i}$ $(i = 1, \ldots, k)$ are labeled and used as training points, we can predict the labels of the unlabeled points most precisely. Active learning is usually referred to as experimental design in statistics. Since our approach is motivated by recent progress in experimental design [14], [16], [17], we begin with a brief description of it.

2.2 Optimum Experimental Design

We consider a linear regression model

$$y = \mathbf{w}^T\mathbf{x} + \epsilon, \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^d$ is the parameter vector, y is the real-valued output, and ϵ is the measurement noise with zero mean and constant variance σ^2. Optimum experimental design attempts to select the most informative experiments (or data points) to learn a prediction function $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ so that the expected prediction error can be minimized. Given a set of measured data points $(\mathbf{x}_{s_1}, y_1), \ldots, (\mathbf{x}_{s_k}, y_k)$, the most popular estimation method is *least squares*, in which we minimize the residual sum of squares (RSS):

$$RSS(\mathbf{w}) = \sum_{i=1}^{k}(y_i - f(\mathbf{x}_{s_i}))^2. \tag{2}$$

Let $Z = [\mathbf{x}_{s_1}, \ldots, \mathbf{x}_{s_k}]^T$ and $\mathbf{y} = [y_1, \ldots, y_k]^T$. The optimal solution is

$$\widehat{\mathbf{w}} = (Z^TZ)^{-1}Z^T\mathbf{y}. \tag{3}$$

It can be proved [22] that $\widehat{\mathbf{w}}$ is an unbiased estimation of $\mathbf{w}$ and its covariance can be expressed as

$$\mathrm{Cov}(\widehat{\mathbf{w}}) = \sigma^2(Z^TZ)^{-1}. \tag{4}$$

The criteria of OED [14] can be classified into two categories. The first category is to select the points $\mathbf{x}_{s_i}$ in order to minimize the size of the parameter covariance matrix [23]. The typical methods in this category include D, A, and E-optimal design. D-optimal design minimizes the determinant of $\mathrm{Cov}(\widehat{\mathbf{w}})$, and thus minimizes the volume of the confidence region. A-optimal design minimizes the trace of $\mathrm{Cov}(\widehat{\mathbf{w}})$, and thus minimizes the dimensions of the enclosing box around the confidence region. E-optimal design minimizes the largest eigenvalue of $\mathrm{Cov}(\widehat{\mathbf{w}})$, and thus minimizes the size of the major axis of the confidence region.

The other category of experimental design criteria is to select the points $\mathbf{x}_{s_i}$ in order to minimize the variance of predicted value over some region of interest $\mathcal{O}$ [24], [25]. Given a test point $\mathbf{v} \in \mathcal{O}$, the predicted value is $\widehat{\mathbf{w}}^T\mathbf{v}$ with variance $\mathbf{v}^T\mathrm{Cov}(\widehat{\mathbf{w}})\mathbf{v}$. The two most common criteria in this category are I and G-optimal design. I-optimal design minimizes the average predictive variance $\int_{\mathbf{v}\in\mathcal{O}} \mathbf{v}^T\mathrm{Cov}(\widehat{\mathbf{w}})\mathbf{v}\, d\mu(\mathbf{v})$, where μ is a probability distribution on $\mathcal{O}$. G-optimal design minimizes the maximum predictive variance, i.e., $\max_{\mathbf{v}\in\mathcal{O}}\{\mathbf{v}^T\mathrm{Cov}(\widehat{\mathbf{w}})\mathbf{v}\}$.

2.3 Transductive Experimental Design

Recently, Yu et al. [16] proposed the TED approach, which can be seen as the discrete version of I-optimal design. TED considers the Regularized Least Squares formulation (ridge regression) as follows:

$$\widehat{\mathbf{w}}_{ridge} = \underset{\mathbf{w}}{\mathrm{argmin}} \sum_{i=1}^{k}(y_i - f(\mathbf{x}_{s_i}))^2 + \gamma\|\mathbf{w}\|^2, \tag{5}$$

where $\gamma \geq 0$ is the regularization parameter. It is easy to check that the optimal solution has the following expression:

$$\widehat{\mathbf{w}}_{ridge} = (Z^TZ + \gamma I)^{-1}Z^T\mathbf{y}, \tag{6}$$

where I is the identity matrix. The covariance matrix of $\widehat{\mathbf{w}}_{ridge}$ is

$$\begin{aligned}
&\mathrm{Cov}(\widehat{\mathbf{w}}_{ridge}) \\
&\quad = (Z^TZ + \gamma I)^{-1}Z^T\mathrm{Cov}(\mathbf{y})Z(Z^TZ + \gamma I)^{-1} \\
&\quad = \sigma^2(Z^TZ + \gamma I)^{-1}Z^TZ(Z^TZ + \gamma I)^{-1} \\
&\quad = \sigma^2(Z^TZ + \gamma I)^{-1}(Z^TZ + \gamma I - \gamma I)(Z^TZ + \gamma I)^{-1} \\
&\quad = \sigma^2(Z^TZ + \gamma I)^{-1} - \gamma\sigma^2(Z^TZ + \gamma I)^{-2}.
\end{aligned} \tag{7}$$

Since the regularization parameter γ is usually set to be very small, we have

$$\mathrm{Cov}(\widehat{\mathbf{w}}_{ridge}) \approx \sigma^2 (Z^T Z + \gamma I)^{-1}. \tag{8}$$

Similarly to I-optimal design, TED selects those points which can minimize the average predictive variance over one pregiven test set. For simplicity, we assume the test set is just $\mathcal{X}$. Let $X = [\mathbf{x}_1, \ldots, \mathbf{x}_m]^T$. The average predictive variance is

$$\begin{aligned} &\frac{1}{m}\sum_{i=1}^{m} \mathbf{x}_i^T \mathrm{Cov}(\widehat{\mathbf{w}}_{ridge})\mathbf{x}_i \\ &\quad\approx \frac{\sigma^2}{m}\sum_{i=1}^{m} \mathbf{x}_i^T (Z^T Z + \gamma I)^{-1}\mathbf{x}_i \\ &\quad= \frac{\sigma^2}{m}\mathrm{Tr}(X(Z^T Z + \gamma I)^{-1} X^T). \end{aligned} \tag{9}$$

Thus, TED is formulated as the following optimization problem:

$$\min \mathrm{Tr}\Big(X\big(Z^T Z + \gamma I\big)^{-1} X^T\Big) \tag{10}$$

with variable $Z = [\mathbf{x}_{s_1}, \ldots, \mathbf{x}_{s_k}]^T$. After some mathematical derivation, the above problem can be formulated as

$$\min \sum_{i=1}^{m} \|\mathbf{x}_i - Z^T \boldsymbol{\alpha}_i\|^2 + \gamma \|\boldsymbol{\alpha}_i\|^2, \tag{11}$$

where the variables are $Z = [\mathbf{x}_{s_1}, \ldots, \mathbf{x}_{s_k}]^T$ and $\boldsymbol{\alpha}_i \in \mathbb{R}^k$, $i = 1, \ldots, m$ [16]. The first term in the objective function shows that the data points selected by TED are the most representative ones. That is, the selected points can be used to reconstruct the whole data set most precisely. The second term indicates that TED penalizes the norm of the reconstruction coefficients. So, it tends to select points with large norm. Notice that TED is closely related to the problem of Column-Based Matrix Decomposition [26].

3 Active Learning Based on Locally Linear Reconstruction

In this section, we introduce a novel active learning algorithm based on the principle of locally linear reconstruction.

3.1 Locally Linear Reconstruction

Recent studies [18], [19], [20], [21], [27] have shown that naturally occurring data may reside on a lower dimensional submanifold which is embedded in the high-dimensional ambient space. However, previous experimental design approaches only take into account the global euclidean structure of the data space, whereas the local manifold structure is not well respected.

Inspired by LLE [18], we assume that the data lie on a low-dimensional manifold which can be approximated linearly in a local area of the high-dimensional space. Therefore, we require that a data point can only be linearly reconstructed from its *neighbors*. The optimal reconstruction coefficients can be obtained by solving the following problem [18]:

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} \|\mathbf{x}_i - \sum_{j=1}^{m} W_{ij}\mathbf{x}_j\|^2 \\ \text{s.t.} \quad & \sum_{j=1}^{m} W_{ij} = 1,\ i = 1, \ldots, m \\ & W_{ij} = 0 \ \text{ if } \ \mathbf{x}_j \notin N_p(\mathbf{x}_i), \end{aligned} \tag{12}$$

where the variable is the matrix $W \in \mathbb{R}^{m \times m}$. Here, W_{ij} summarizes the contribution of the jth data point to the ith reconstruction, and $N_p(\mathbf{x}_i)$ is the neighborhood of $\mathbf{x}_i$ defined by its p nearest neighbors.

To measure the representativeness of the selected data points, we need to design a data reconstruction mechanism by using the reconstruction coefficients. Given a set of selected data points $\{\mathbf{x}_{s_1}, \ldots, \mathbf{x}_{s_k}\} \subseteq \mathcal{X}$, we propose a transductive learning algorithm, called LLR, to reconstruct the data points. Let $\{\mathbf{q}_1, \ldots, \mathbf{q}_m\}$ denote the reconstructed points. Their coordinates are determined by minimizing the following cost function:

$$\epsilon(\mathbf{q}_1, \ldots, \mathbf{q}_m) = \sum_{i=1}^{k} \|\mathbf{q}_{s_i} - \mathbf{x}_{s_i}\|^2 + \mu \sum_{i=1}^{m} \|\mathbf{q}_i - \sum_{j=1}^{m} W_{ij}\mathbf{q}_j\|^2, \tag{13}$$

where μ is a suitable constant. The role of the first term of the right-hand side in the cost function is to fix the coordinates of the selected data points. The second term requires the reconstructed points to share the same local geometrical structure with the original points.

Let $X = [\mathbf{x}_1, \ldots, \mathbf{x}_m]^T$, $Q = [\mathbf{q}_1, \ldots, \mathbf{q}_m]^T$, and Λ be an $m \times m$ diagonal matrix whose diagonal entry Λ_{ii} is 1 if $i \in \{s_1, \ldots, s_k\}$ and 0 otherwise. Then, the above cost function (13) can be rewritten in the following matrix form:

$$\epsilon(Q) = \mathrm{Tr}((Q - X)^T \Lambda (Q - X)) + \mu \mathrm{Tr}(Q^T M Q), \tag{14}$$

where $M = (I - W)^T (I - W)$. Requiring that the gradient of $\epsilon(Q)$ vanish gives the following equation:

$$\Lambda(Q - X) + \mu M Q = 0. \tag{15}$$

Finally, the reconstructed points are given by

$$Q = (\mu M + \Lambda)^{-1} \Lambda X. \tag{16}$$

The LLR algorithm presented here shares many common properties with LLE [18]. For example, we use the same objective function (12) to find the reconstruction coefficients. However, the goals of LLE and LLR are different. LLE uses the reconstruction coefficients to obtain lower dimensional representations of the original data points. Suppose $\mathbf{y}_i$ is the $l(\ll d)$-dimensional embedding of $\mathbf{x}_i$, $i = 1, \ldots, m$. LLE solves the following optimization problem to obtain $\mathbf{y}_i$s:

$$\epsilon(\mathbf{y}) = \sum_{i=1}^{m} \Big(\mathbf{y}_i - \sum_{j=1}^{m} W_{ij}\mathbf{y}_j\Big)^2. \tag{17}$$

For our LLR algorithm, the goal is to reconstruct the data set. Therefore, the reconstructed data point $\mathbf{q}_i$ has the same dimension as the original data point $\mathbf{x}_i$. Moreover, for the selected data points $\mathbf{x}_{s_i}$, $i = 1, \ldots, k$, their coordinates are given. Therefore, their reconstructions (i.e., $\mathbf{q}_{s_i}$) should

be as close to their original coordinates (i.e., $\mathbf{x}_{s_i}$) as possible. Our ultimate goal is to select the most representative data points, so that the reconstruction error can be minimized.

There are also some works in semi-supervised learning which have a similar principle of LLR, such as [2], [28], [29]. However, all of these approaches aim to predict the labels for the unlabeled points by using both labeled and unlabeled points. In LLR, there is no label prediction task. The task of LLR is to reconstruct the data set, given some selected points and the reconstruction coefficients.

3.2 Selecting the Most Representative Points

Given the original data points $\mathbf{x}_1, \ldots, \mathbf{x}_m$, and the reconstructed data points $\mathbf{q}_1, \ldots, \mathbf{q}_m$, the reconstruction error can be measured as follows:

$$\begin{aligned} & e(\mathbf{x}_{s_1}, \ldots, \mathbf{x}_{s_k}) \\ &= \|X - Q\|_F^2 \\ &= \|X - (\mu M + \Lambda)^{-1}\Lambda X\|_F^2 \\ &= \|X - (\mu M + \Lambda)^{-1}(\Lambda + \mu M - \mu M)X\|_F^2 \\ &= \|(\mu M + \Lambda)^{-1}\mu M X\|_F^2, \end{aligned} \tag{18}$$

where $\|\cdot\|_F^2$ is the matrix Frobenius norm, which is defined as $\|A\|_F^2 = \mathrm{Tr}(AA^T) = \mathrm{Tr}(A^TA)$. Clearly, the reconstruction error is only dependent on the selected data points $\mathbf{x}_{s_1}, \ldots, \mathbf{x}_{s_k}$.

Thus, the most representative points are naturally defined as those which minimize the reconstruction error (18). That is, given their coordinates, we can reconstruct the whole data set most precisely by using the LLR algorithm. Suppose we are going to select k points, the active learning problem is, thus, formally defined below:

Definition 1. *Active Learning based on LLR:*

$$\begin{aligned} \min \quad & \|(\mu M + \Lambda)^{-1}\mu M X\|_F^2 \\ \text{s.t.} \quad & \Lambda \text{ is diagonal,} \\ & \Lambda_{ii} \in \{0, 1\},\ i = 1, \ldots, m \\ & \sum_{i=1}^{m} \Lambda_{ii} = k, \end{aligned} \tag{19}$$

where the variable is the diagonal matrix $\Lambda \in \mathbb{R}^{m \times m}$.

Given the optimal solution $\widehat{\Lambda}$ of (19), we select those data points whose corresponding entries in the diagonal matrix Λ are 1. After we obtain the labels of the selected points, we can use any supervised or semi-supervised algorithms [1], [2], [3], [22], [30] to predict the labels of the other points.

4 Optimization Scheme

The optimization problem of LLR_{Active} (19) is difficult due to its combinatorial nature. In this section, we develop two optimization schemes to solve (19). The first one is a sequential greedy approach, and the second one is a convex relaxation. The solution of sequential approach is suboptimal, but its sequential property makes it much more efficient than convex optimization and it thus can be applied to large-scale data sets. Moreover, our experimental results show that there is only a slight difference between sequential and convex optimization performance. On the other hand, the convex relaxation approach can guarantee to find the globally optimal solution of the relaxed problem, but it is computationally expensive.

4.1 The Sequential Approach

Suppose a set of n points $\mathcal{Z}_n = \{\mathbf{x}_{s_1}, \ldots, \mathbf{x}_{s_n}\} \subseteq \mathcal{X}$ have been selected as the n most representative ones. Let Λ_n denote the corresponding $m \times m$ diagonal matrix whose diagonal entry $(\Lambda_n)_{ii}$ is 1 if $\mathbf{x}_i \in \mathcal{Z}_n$ and 0 otherwise. Let Γ_i be an $m \times m$ matrix whose (i,i)th entry is 1 and all the other entries are 0. The $(n+1)$th point $\mathbf{x}_{s_{n+1}}$ can be found by solving the following problem:

$$s_{n+1} = \underset{i \notin \{s_1, \ldots, s_n\}}{\mathrm{argmin}} \|(\mu M + \Lambda_n + \Gamma_i)^{-1}\mu M X\|_F^2. \tag{20}$$

As can be seen, the most expensive calculation in (20) is the matrix inverse $(\mu M + \Lambda_n + \Gamma_i)^{-1}$. Since the matrix M is sparse, the sparse Cholesky factorization [31] can be applied to accelerate the calculation of $(\mu M + \Lambda_n + \Gamma_i)^{-1}\mu M X$. But the sequential solver based on the sparse Cholesky factorization still needs to perform $m - n$ factorizations in order to solve (20), and thus doesn't scale well.

A much faster method is to use the Sherman-Morrison-Woodbury formula [32] to avoid directly inverting a matrix. Given an invertible matrix A, two column vectors $\mathbf{u}$ and $\mathbf{v}$, the Sherman-Morrison-Woodbury formula states:

$$(A + \mathbf{u}\mathbf{v}^T)^{-1} = A^{-1} - \frac{A^{-1}\mathbf{u}\mathbf{v}^T A^{-1}}{1 + \mathbf{v}^T A^{-1}\mathbf{u}}. \tag{21}$$

Denote the ith unit vector as $\mathbf{e}_i$. It is easy to check that $\Gamma_i = \mathbf{e}_i\mathbf{e}_i^T$. Define

$$H = (\mu M + \Lambda_n)^{-1}.$$

Let H_{*i} denotes the ith column of H, and H_{i*} denotes the ith row of H. Following (21), we get

$$(\mu M + \Lambda_n + \Gamma_i)^{-1} = H - \frac{H_{*i}H_{i*}}{1 + H_{ii}}. \tag{22}$$

With (22), the objective function of (20) can be rewritten as

$$\begin{aligned} & \|(\mu M + \Lambda_n + \Gamma_i)^{-1}\mu M X\|_F^2 \\ &= \mu^2 \mathrm{Tr}(HMXX^TMH) - \frac{2\mu^2 H_{i*}MXX^TMHH_{*i}}{1 + H_{ii}} \\ &\quad + \frac{\mu^2 H_{i*}H_{*i}H_{i*}MXX^TMH_{*i}}{(1 + H_{ii})^2}. \end{aligned} \tag{23}$$

For brevity, the derivations of (22) and (23) are given in Appendices A and B, respectively, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2011.20.

Denote $A = MXX^TM$. Notice that $\mathrm{Tr}(HAH)$ is a constant when selecting the $(n+1)$th data point. Therefore, the optimization problem (20) becomes

$$s_{n+1} = \underset{i \notin \{s_1, \ldots, s_n\}}{\mathrm{argmin}} \frac{1}{1 + H_{ii}} \left(\frac{H_{i*}H_{*i}H_{i*}AH_{*i}}{1 + H_{ii}} - 2H_{i*}AHH_{*i} \right). \tag{24}$$

TABLE 1
The Sequential Approach for LLR_{Active}

Input: The candidate data set $(\{\mathbf{x}_1, \ldots, \mathbf{x}_m\})$, the number of nearest neighbor (p), the number of points to be selected (k) and the regularization parameter (μ).
Output: The indexes of the k most representative points, $\mathcal{I} = \{s_1, \ldots, s_k\}$.

1: Initialize W by solving problem (12)
2: $M \leftarrow (I - W)^T(I - W)$
3: $X \leftarrow [\mathbf{x}_1, \ldots, \mathbf{x}_m]^T$
4: $\mathcal{I} \leftarrow \emptyset$
5: $H \leftarrow (\mu M)^{-1}$
6: $A \leftarrow MXX^TM$
7: **for** $n = 1$ to k **do**
8: **for** $i = 1$ to m **do**
9: **if** $i \notin \mathcal{I}$ **then**
10: $z(i) = \frac{1}{1+H_{ii}} H_{i*} \left(A\left(\frac{\|H_{i*}\|^2}{2(1+H_{ii})} I - H\right)\right) H_{*i}$
11: **end if**
12: **end for**
13: $s_n = \mathrm{argmin}_{i \notin \mathcal{I}}\, z(i)$
14: $\mathcal{I} \leftarrow \mathcal{I} \cup s_n$
15: $H \leftarrow H - \frac{H_{*s_n} H_{s_n *}}{1+H_{s_n s_n}}$
16: **end for**
17: **return** $\mathcal{I}$

Since $H_{i*}H_{*i} = \|H_{i*}\|^2$, the optimization problem (24) can be further simplified as

$$s_{n+1} = \underset{i \notin \{s_1, \ldots, s_n\}}{\mathrm{argmin}} \frac{1}{1+H_{ii}} H_{i*} \left(A\left(\frac{\|H_{i*}\|^2}{2(1+H_{ii})} I - H\right)\right) H_{*i}. \tag{25}$$

After we have selected the $(n+1)$th point $\mathbf{x}_{s_{n+1}}$, the H matrix can be updated as

$$H \leftarrow (\mu M + \Lambda_{n+1})^{-1} = (\mu M + \Lambda_n + \Gamma_i)^{-1}.$$

The matrix inverse can be computed according to (22). This process is repeated until we have selected k points. In the beginning, there are no data points selected. Therefore, we set $H = (\mu M)^{-1}$. Since M is singular, a small ridge term is added to it. The sequential approach is summarized in Table 1.

4.2 The Convex Relaxation

In this section, we discuss how to perform convex relaxation to solve the optimization problem (19).

First, we rewrite the objective function of (19) as follows:

$$\begin{aligned} &\|(\mu M + \Lambda)^{-1}\mu MX\|_F^2 \\ &\quad = \mu^2 \mathrm{Tr}(X^T M(\mu M + \Lambda)^{-1}(\mu M + \Lambda)^{-1} MX) \\ &\quad = \mu^2 \mathrm{Tr}(X^T M(\mu^2 M^2 + \mu M\Lambda + \mu\Lambda M + \Lambda)^{-1} MX), \end{aligned} \tag{26}$$

where, in line 3, we use the property $\Lambda^2 = \Lambda$.

Since Λ is diagonal, we introduce a vector $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_m]^T$ such that $\Lambda = \mathrm{diag}(\boldsymbol{\lambda})$. Here, the value of λ_i indicates whether or not the data point $\mathbf{x}_i$ is selected. Define an affine function

$$h(\boldsymbol{\lambda}) = \mu^2 M^2 + \sum_{i=1}^{m} \lambda_i \left(\mu M_{*i}\mathbf{e}_i^T + \mu \mathbf{e}_i M_{i*} + \mathbf{e}_i \mathbf{e}_i^T\right).$$

Thus, the original optimization problem (19) is equivalent to

$$\begin{aligned} \min \quad & \mathrm{Tr}(X^T M h(\boldsymbol{\lambda})^{-1} MX) \\ \text{s.t.} \quad & \boldsymbol{\lambda} \in \{0,1\}^m,\ \mathbf{1}^T\boldsymbol{\lambda} = k, \end{aligned} \tag{27}$$

where the variable is $\boldsymbol{\lambda} \in \mathbb{R}^m$ and $\mathbf{1}$ is a column vector of all ones. Notice that the variable vector $\boldsymbol{\lambda}$ is sparse and has only k nonzero entries.

In order to solve the above optimization problem efficiently, we relax the integer constraints on λ_is and allow λ_is to take real nonnegative values. Then, the value of λ_i indicates how significantly $\mathbf{x}_i$ contributes to the minimization in problem (27). The sparseness of $\boldsymbol{\lambda}$ can be controlled by minimizing the ℓ_1-norm of $\boldsymbol{\lambda}$ $(\|\boldsymbol{\lambda}\|_1)$, which has conventionally been applied to lasso regression [22], [33].

Following the convention in the field of optimization, we use $\succeq$ to denote componentwise inequality between two vectors with the same dimension. For example, $\boldsymbol{\alpha} \succeq \boldsymbol{\beta}$ means that $\alpha_i \geq \beta_i$, for all i. Because all the elements of $\boldsymbol{\lambda}$ are nonnegative, $\|\boldsymbol{\lambda}\|_1$ is equal to $\mathbf{1}^T\boldsymbol{\lambda}$. Finally, the optimization problem becomes

$$\begin{aligned} \min \quad & \mathrm{Tr}(X^T M h(\boldsymbol{\lambda})^{-1} MX) + \alpha \mathbf{1}^T\boldsymbol{\lambda} \\ \text{s.t.} \quad & \boldsymbol{\lambda} \succeq \mathbf{0}, \end{aligned} \tag{28}$$

where the variable is $\boldsymbol{\lambda} \in \mathbb{R}^m$ and $\mathbf{0}$ is the column vectors of all zeros. It can be shown that the problem (28) is a convex optimization problem with variable $\boldsymbol{\lambda}$ [33].

The objective function of problem (28) is continuously differentiable twice, so it can be solved directly by standard optimization techniques [33]. In particular, we show that it can be cast as a Semi-Definite Programming (SDP) problem, which can be solved using a standard SDP package. By introducing an auxiliary variable $P \in \mathbb{R}^{d \times d}$, the problem (28) can be equivalently rewritten as

$$\begin{aligned} \min \quad & \mathrm{Tr}(P) + \alpha \mathbf{1}^T\boldsymbol{\lambda} \\ \text{s.t.} \quad & P \succeq_{\mathbb{S}_d^+} X^T M h(\boldsymbol{\lambda})^{-1} MX \\ & \boldsymbol{\lambda} \succeq \mathbf{0} \end{aligned} \tag{29}$$

with variables $P \in \mathbb{R}^{d \times d}$ and $\boldsymbol{\lambda} \in \mathbb{R}^m$. Here, $\mathbb{S}_d^+$ denotes the set of symmetric positive semi-definite $d \times d$ matrices, which is called positive semi-definite cone in the field of optimization. The associated generalized inequality $\succeq_{\mathbb{S}_d^+}$ is the usual matrix inequality: $A \succeq_{\mathbb{S}_d^+} B$ means $A - B$ is a positive semi-definite $d \times d$ matrix [33].

The problem (29) can be cast as an SDP by using the Schur complement theorem [33]. Given a symmetric matrix X partitioned as

$$X = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}.$$

If A is invertible, the matrix $S = C - B^T A^{-1} B$ is called the Schur complement of A in X. The Schur complement theorem states that, if A is positive definite, then X is positive semi-definite if and only if S is positive semi-definite. According to this theorem, problem (29) is equivalent to the following SDP problem:

TABLE 2
The Convex Relaxation for LLR_{Active}

Input: The candidate data set ($\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$), the number of nearest neighbor (p), the number of points to be selected (k), the regularization parameters (μ and α).
Output: The indexes of the k most representative points, $\mathcal{I} = \{s_1, \ldots, s_k\}$.

1: Initialize W by solving problem (12)
2: $M \leftarrow (I - W)^T(I - W)$
3: $X \leftarrow [\mathbf{x}_1, \ldots, \mathbf{x}_m]^T$
4: $\mathcal{I} \leftarrow \emptyset$
5: $h(\boldsymbol{\lambda}) \leftarrow \mu^2 M^2 + \sum_{i=1}^{m} \lambda_i (\mu M_{*i} \mathbf{e}_i^T + \mu \mathbf{e}_i M_{i*} + \mathbf{e}_i \mathbf{e}_i^T)$
6: Solve the following SDP problem

$$\begin{aligned} \min \quad & \mathrm{Tr}(P) + \alpha \mathbf{1}^T \boldsymbol{\lambda} \\ \text{s.t.} \quad & \begin{bmatrix} h(\boldsymbol{\lambda}) & MX \\ X^T M & P \end{bmatrix} \succeq_{\mathbb{S}^+_{d+m}} 0 \\ & \boldsymbol{\lambda} \succeq \mathbf{0} \end{aligned}$$

7: **for** $n = 1$ to k **do**
8: $\quad s_n = \mathrm{argmax}_{i \notin \mathcal{I}} \lambda_i$
9: $\quad \mathcal{I} \leftarrow \mathcal{I} \cup s_n$
10: **end for**
11: **return** $\mathcal{I}$

$$\begin{aligned} \min \quad & \mathrm{Tr}(P) + \alpha \mathbf{1}^T \boldsymbol{\lambda} \\ \text{s.t.} \quad & \begin{bmatrix} h(\boldsymbol{\lambda}) & MX \\ X^T M & P \end{bmatrix} \succeq_{\mathbb{S}^+_{d+m}} 0 \\ & \boldsymbol{\lambda} \succeq \mathbf{0} \end{aligned} \tag{30}$$

with variables $P \in \mathbb{R}^{d \times d}$ and $\boldsymbol{\lambda} \in \mathbb{R}^m$. As explained previously, $A \succeq_{\mathbb{S}^+_{d+m}} 0$ means that A is a positive semidefinite $(d+m) \times (d+m)$ matrix. Once the optimal solution $\boldsymbol{\lambda}^*$ is obtained, the most representative points are defined as those with the largest λ_i^*s. We summarize the method of convex relaxation in Table 2.

Note that in problem (30), the number of variables is $d^2 + m$. Thus, the computational cost of the convex relaxation is high and very sensitive to the dimensionality of the data. In real-world applications where high-dimensional data are common, the sequential approach discussed in Section 4.1 is preferred.

5 Experimental Results

To demonstrate the effectiveness of our proposed algorithm, we evaluate and compare five active learning methods:

- **Random Sampling** method, which randomly selects points from the data set.
- **A-Optimal Design** (AOD) [14] as described in Section 2.2. Similarly to the sequential approach of LLR_{Active}, we implement a sequential algorithm to solve it.
- **Transductive Experimental Design**[1] [16] as described in Section 2.3.
- **Support Vector Machine active learning** (SVM_{Active}) [7] which selects the points closest to the current decision boundary of the SVM classifier as the most informative ones. We implement SVM_{Active} based on the LIBSVM package [34].
- **Active Learning Based on Locally Linear Reconstruction** proposed in this paper. The convex relaxation of LLR_{Active} is solved by the CVX package [35], [36].

1. A sequential solver can be downloaded from http://www.dbs.ifi.lmu.de/~yu_k/.

5.1 Toy Examples

In this section, we apply the active learning algorithms on two synthetic data sets to give some intuition about how each algorithm works. The data sets are

- Two-circle data set (Fig. 1): There are 32 points on the big circle and 16 points on the small circle.
- Two-moon data set (Fig. 2): There are 100 points for each moon.

We apply AOD, TED, sequential LLR_{Active}, and convex LLR_{Active} to select the most informative points on the two data sets. Here, SVM_{Active} cannot be applied due to the lack of labeled points. The results are shown in Figs. 1 and 2. The points selected by each active learning algorithm are marked as solid dots. The numbers beside the selected points denote the orders in which they were selected. Compared with AOD and TED, the points selected by our LLR_{Active} algorithm can better represent the original data set. Particularly for the two-circle data set, both AOD and TED select data points from the large circle. Therefore, even though these selected points are labeled, we are still unable to perform classification since all of the labeled points are from the same class. Moreover, we see that the difference between AOD and TED is small.

As can be seen from these two examples, the points selected by LLR_{Active} can indeed reflect the manifold structure of the data set. On the other hand, both AOD and TED tend to select those points with large norms [16]. The points with large norms might be hard to predict [16]. However, they may not be able to best represent the whole data set, especially when there is manifold structure present. Therefore, even though these points are labeled, they may not be able to improve the classification performance the most.

5.2 Real-World Data Sets

In this section, we carry out classification experiments on three real-world data sets to compare different active learning algorithms quantitatively.

5.2.1 Experimental Setting

In our experiments, we use the points selected by each active learning algorithm as the training data to train a classifier, and the unselected points are used as the testing data. The classification accuracy of the associated classifier is used to measure the performance of each active learning algorithm.

To handle multiclass classification problem, we adopt the *one-versus-all* (OVA) scheme. If the training data contain c classes, OVA trains c binary classifiers and each binary classifier separates one class (positive) from all the other classes (negative). To classify each unselected point, these c classifiers are applied to the test point, and its class label is determined according to the largest output value from the classifiers. Two classifiers, Support Vector Machine (SVM) [30] and Laplacian Regularized Least Squares (LapRLS) [3],

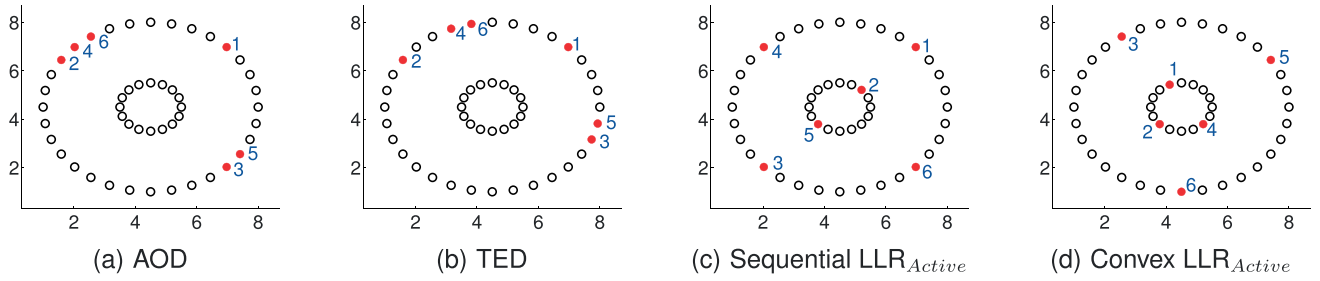


Fig. 1. Data selection by active learning algorithms. The numbers beside the selected points indicate the orders in which they were selected. Clearly, the points selected by Sequential LLR_{Active} and Convex LLR_{Active} can better represent the original data set.

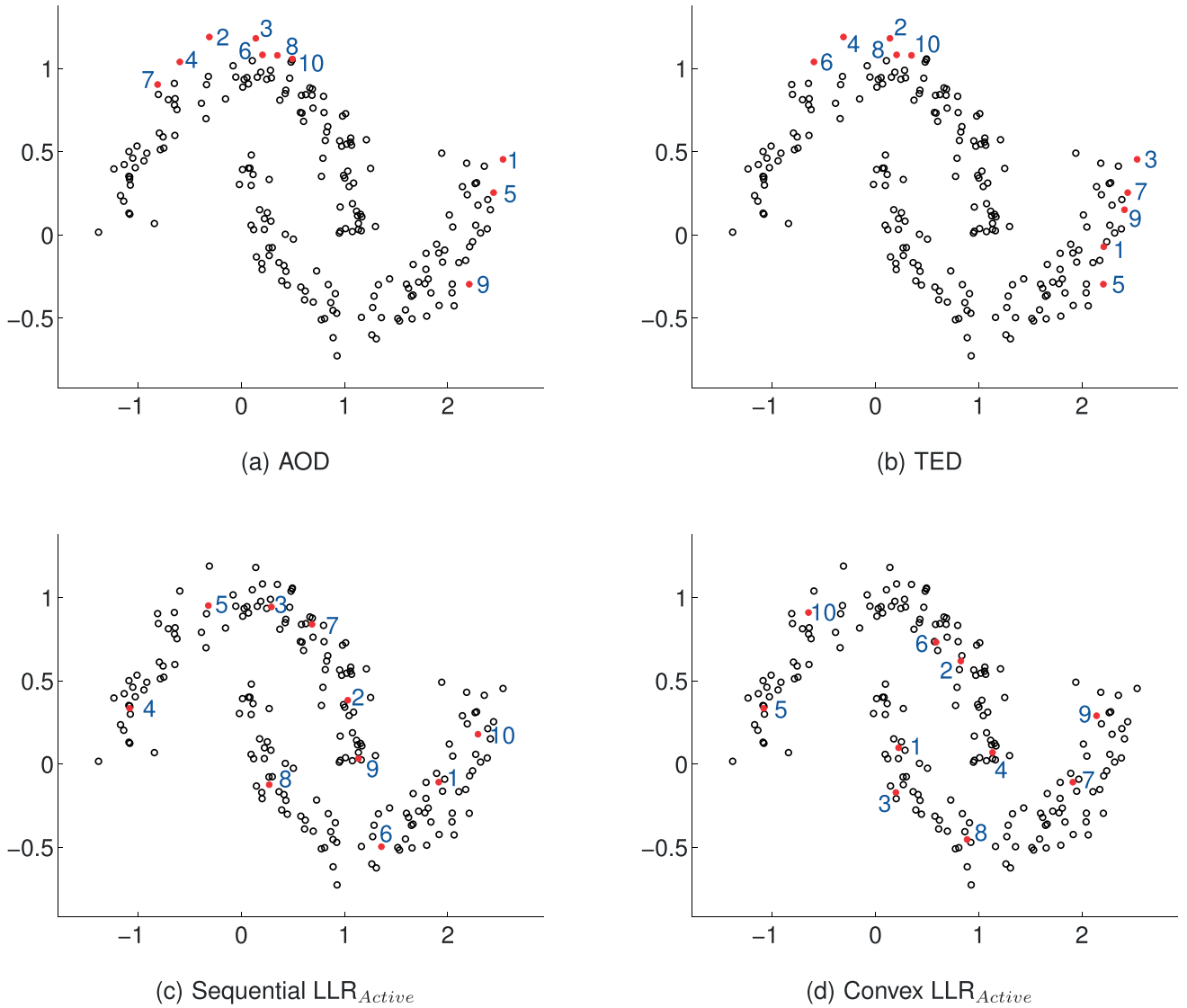
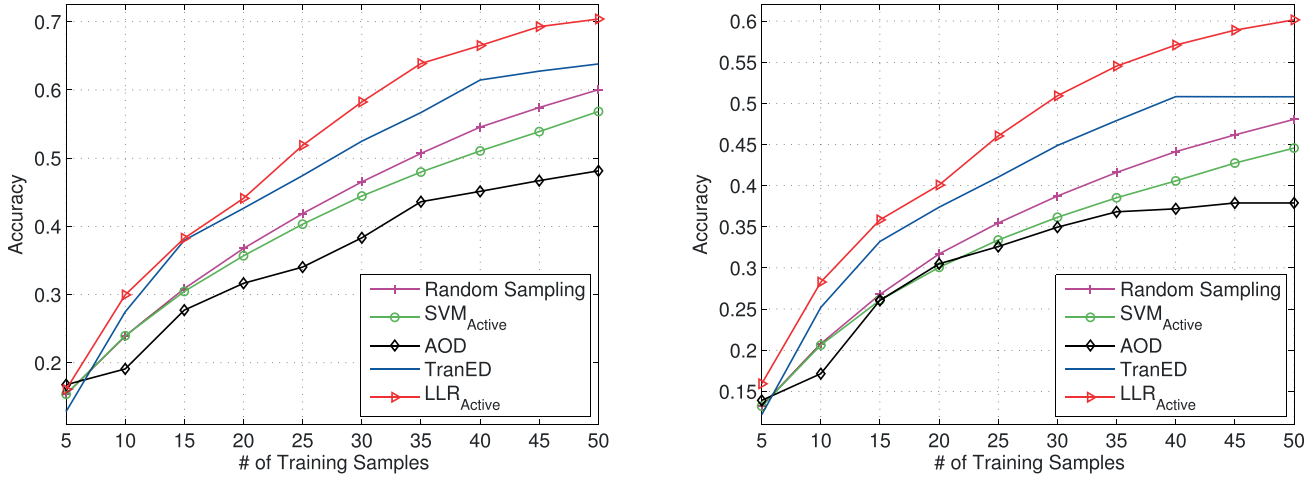


Fig. 2. Data selection by active learning algorithms. The numbers beside the selected points indicate the orders in which they were selected. Clearly, the points selected by Sequential LLR_{Active} and Convex LLR_{Active} can better represent the original data set.

are used. SVM is a supervised learning algorithm, while LapRLS is a semi-supervised learning algorithm.

It would be important to note that the algorithms Random Sampling, AOD, TED, and LLR_{Active} are all *label-independent*. The SVM_{Active} algorithm selects the most informative points based on the current SVM classifier, which is trained on some pregiven labeled data. In our experiments, the initial SVM classifier of SVM_{Active} is trained on the points selected by Random Sampling. To obtain stable results, Random Sampling and SVM_{Active} are repeated 10 times for each experimental test. Since the original SVM_{Active} algorithm is designed for two-class problems, we need to extend it to handle multiclass cases. Suppose there are c classes in the training data. We can train c *one-versus-all* SVM classifiers. For any unselected point, its informativeness is determined by the SVM classifier whose boundary is closest to this point. The informative distance of this point is defined as the distance between this point

Fig. 3. Sample images from the Yale face database.



(a) The average classification accuracy by SVM (b) The average classification accuracy by LapRLS

Fig. 4. Classification results on the Yale face database. The face images selected by each active learning algorithm are used as the training data, and the unselected images are used as the testing data. The evaluations are conducted with 20 randomly generated subsets of the original database. These figures show the average classification accuracy versus the number of the training samples.

TABLE 3
Classification Results on the Yale Face Database (mean ± standard deviation (percent))

k	The classification accuracy by SVM					The classification accuracy by LapRLS				
	Random	SVM_{Active}	AOD	TED	LLR_{Active}	Random	SVM_{Active}	AOD	TED	LLR_{Active}
5	15.4±3.7	15.4±3.7	**16.8±2.1**	12.9±2.0	16.1±2.4	13.2±3.6	13.2±3.6	13.9±2.0	12.2±2.9	**15.9±2.0**
10	23.9±4.2	24.0±4.8	19.1±1.9	27.5±4.3	**30.0±3.1**	20.8±4.9	20.6±5.2	17.1±1.7	25.2±4.9	**28.3±3.7**
15	30.9±4.9	30.5±4.8	27.7±2.5	38.0±2.5	**38.3±3.7**	26.8±5.5	26.1±5.5	26.1±2.1	33.2±2.1	**35.9±3.8**
20	36.8±5.7	35.7±5.3	31.7±3.0	42.7±2.9	**44.1±4.1**	31.7±5.7	30.1±5.6	30.5±2.0	37.4±3.4	**40.1±3.6**
25	41.9±6.1	40.3±5.1	34.0±3.7	47.4±2.9	**51.9±3.5**	35.5±5.7	33.4±5.4	32.6±2.6	41.0±2.7	**46.0±3.9**
30	46.5±5.8	44.4±5.0	38.3±3.8	52.5±3.2	**58.2±4.1**	38.7±5.3	36.2±5.3	35.0±2.6	44.9±3.4	**50.9±3.3**
35	50.7±5.5	48.0±5.5	43.6±2.8	56.7±4.6	**63.9±3.8**	41.6±5.1	38.5±5.1	36.8±2.7	47.9±3.5	**54.5±2.5**
40	54.5±5.3	51.1±5.3	45.1±2.9	61.5±4.7	**66.5±3.3**	44.1±5.1	40.6±5.1	37.2±2.1	50.8±3.6	**57.1±2.3**
45	57.4±5.7	53.9±5.6	46.7±2.9	62.8±4.4	**69.3±3.7**	46.2±5.3	42.8±5.0	37.9±3.0	50.8±3.3	**58.9±3.0**
50	60.0±5.5	56.8±5.8	48.2±3.1	63.8±4.5	**70.4±3.0**	48.1±5.4	44.6±5.3	37.9±3.1	50.8±3.8	**60.1±2.5**
Avg.	41.8±5.2	40.0±5.1	35.1±2.9	46.6±3.6	**50.9±3.5**	34.7±5.2	32.6±5.1	30.5±2.4	39.4±3.4	**44.8±3.0**

and its closest SVM classifier. Then, we select those points with shortest informative distances.

There are two parameters in our algorithm, namely, the number of nearest neighbors (p) and the regularization parameter μ in (13). These two parameters are empirically set to 10 and 0.01, respectively. Since Sequential LLR_{Active} and Convex LLR_{Active} perform comparably to each other but Sequential LLR_{Active} is much more computationally efficient, we apply Sequential LLR_{Active} for data selection in the following experiments on face recognition, handwritten digit recognition, and image classification.

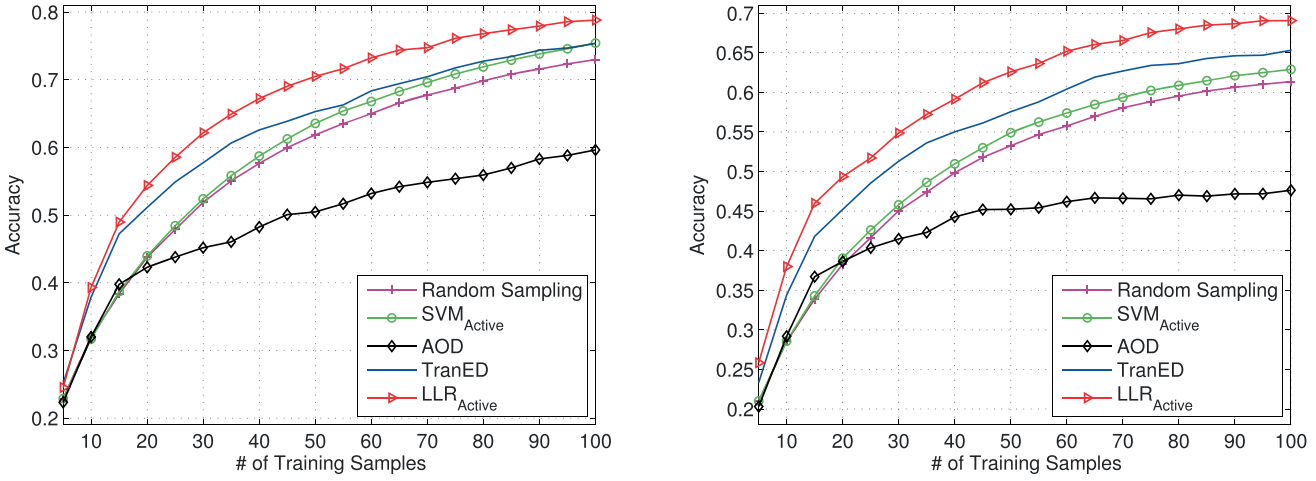
5.2.2 Face Recognition

The Yale face database[2] is used in this experiment. It contains 165 gray-scale images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink. All the face images are manually aligned and cropped. The size of each cropped image is 32×32 pixels, with 256 gray levels per pixel. Thus, each image is represented as a 1,024-dimensional vector. Fig. 3 shows some sample images from the Yale face database.

The evaluations are conducted with 20 randomly generated subsets of the original data set. The average classification accuracy is computed over these 20 tests. For each test, 10 images from each class are randomly chosen to form the data set. Therefore, there are 150 (15×10) images per test, and each active learning algorithm is applied to select a given number $k(= 5, 10, \ldots, 50)$ of training samples. The unselected samples are used as the testing data.

Fig. 4a shows the average classification accuracy versus the number of training (selected) samples by using SVM as the classification algorithm. Fig. 4b shows the average classification accuracy by using LapRLS as the classification algorithm. Each curve in the figures represents an active learning algorithm. As can be seen, our LLR_{Active} algorithm

2. http://cvc.yale.edu/projects/yalefaces/yalefaces.html.



(a) The average classification accuracy by SVM (b) The average classification accuracy by LapRLS

Fig. 5. Classification results on the USPS handwritten digits data set. The digit images selected by each active learning algorithm are used as the training data and the unselected images are used as the testing data. The evaluations are conducted with 20 randomly generated subsets of the original data set. These figures show the average classification accuracy versus the number of the training samples.

TABLE 4
Classification Results on the USPS Handwritten Digits Data Set (mean ± standard deviation (percent))

k	The classification accuracy by SVM					The classification accuracy by LapRLS				
	Random	SVM_{Active}	AOD	TED	LLR_{Active}	Random	SVM_{Active}	AOD	TED	LLR_{Active}
10	32.1±4.2	31.7±4.5	32.0±5.1	37.9±3.3	**39.3±3.7**	28.6±4.0	28.6±4.7	29.2±4.3	34.4±3.5	**38.0±3.2**
20	43.8±4.3	44.0±4.5	42.3±4.9	51.1±4.1	**54.4±3.2**	38.3±4.4	39.1±4.4	38.6±4.7	45.2±3.3	**49.4±3.3**
30	51.9±3.9	52.4±4.2	45.2±5.5	57.7±4.2	**62.2±1.9**	45.0±3.9	45.8±4.1	41.5±4.8	51.3±3.5	**54.9±2.4**
40	57.6±3.7	58.7±4.0	48.2±4.8	62.6±3.9	**67.2±2.3**	49.8±3.7	51.0±3.8	44.3±4.9	55.0±3.3	**59.1±3.2**
50	61.9±3.8	63.5±3.6	50.5±5.1	65.3±3.3	**70.5±2.5**	53.3±3.7	54.9±3.4	45.2±5.3	57.6±3.5	**62.6±2.8**
60	65.0±3.5	66.8±3.5	53.2±5.1	68.4±3.2	**73.2±1.9**	55.8±3.5	57.4±3.2	46.2±5.1	60.4±2.7	**65.2±2.1**
70	67.7±3.2	69.6±3.1	54.8±4.5	70.4±2.6	**74.7±1.9**	58.1±3.3	59.3±2.9	46.6±4.7	62.7±2.8	**66.5±2.0**
80	69.9±3.0	71.9±2.9	55.9±4.6	72.7±2.7	**76.8±1.6**	59.5±3.0	60.9±2.7	47.0±4.0	63.6±2.3	**68.0±2.1**
90	71.6±2.8	73.8±2.6	58.3±4.2	74.4±2.5	**77.9±1.8**	60.6±2.7	62.1±2.5	47.2±3.8	64.6±2.2	**68.7±2.0**
100	73.0±2.5	75.4±2.4	59.6±4.2	75.4±2.1	**78.8±1.3**	61.4±2.7	62.9±2.4	47.6±3.8	65.3±2.2	**69.1±1.9**
Avg.	59.4±3.5	60.8±3.5	50.0±4.8	63.6±3.2	**67.5±2.2**	51.0±3.5	52.2±3.4	43.3±4.5	56.0±2.9	**60.1±2.5**

significantly outperforms the other active learning algorithms in most cases. The performance difference gets larger as the number of training samples increases. The TED algorithm outperforms Random Sampling in most cases. However, both AOD and SVM_{Active} are even worse than Random Sampling. When only five samples are selected, there exist some classes which do not have any labeled samples. Therefore, in this case, all of the algorithms yield low recognition rate. As the number of selected samples increases, the recognition rates of all of the algorithms increase. However, both AOD and TED tend to converge when more than 40 samples are selected for training, whereas the classification algorithms (SVM and LapRLS) can consistently benefit from the samples selected by our LLR_{Active} algorithm.

Table 3 shows the detailed classification accuracies, as well as the standard deviations, for each algorithm. As can be seen, with only 30-35 selected samples, LLR_{Active} performs comparably to or even better than the other algorithms with 50 selected samples. This way, the labeling cost can be significantly reduced by using our algorithm.

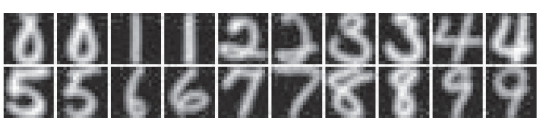


Fig. 6. Sample images from the USPS handwritten digits data set.

5.2.3 Handwritten Digits Recognition

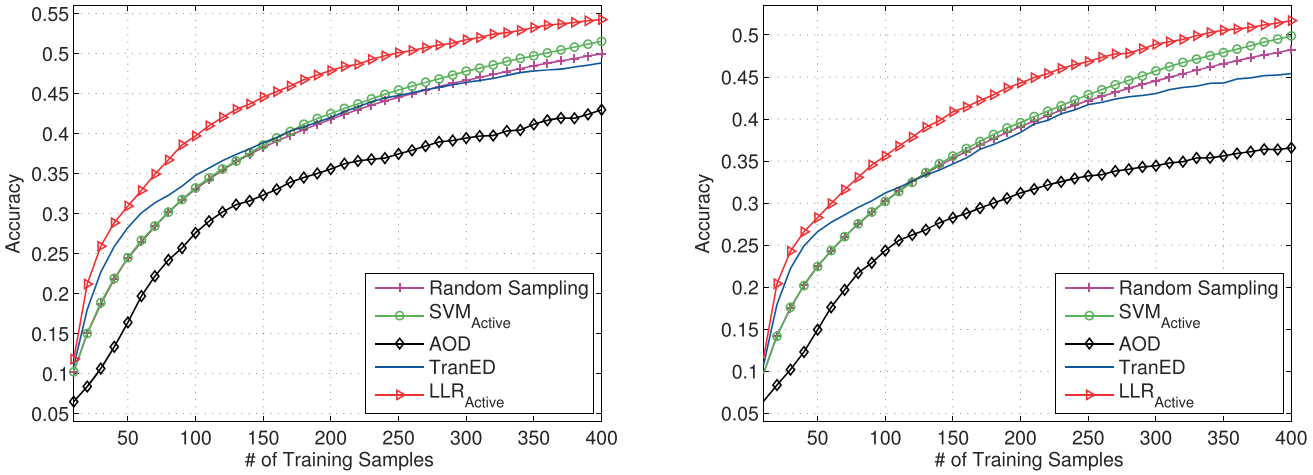
We use the USPS handwritten digits data set[3] in this experiment. This data set contains 8-bit gray-scale images of "0" through "9." The size of each image is 16×16 pixels. Thus, each digit image is represented as a 256-dimensional vector. Fig. 6 shows some sample images from the USPS data set.

On this data set, we also generate 20 tests by randomly selecting 100 images from each class. So, there are 1,000 (10×100) images for each test. We then apply each active learning algorithm to select k $(= 5, 10, \ldots, 100)$ training samples. The average classification accuracy is shown in Fig. 5. Again, our LLR_{Active} algorithm outperforms the other algorithms in all of the cases. TED performs the second best. SVM_{Active} and Random Sampling perform comparably to each other. AOD performs the worst, especially when there are more training samples. Table 4 shows the detailed

3. http://www.cs.toronto.edu/~roweis/data.html.

(a) Aquarelle (b) Fox (c) Beach

Fig. 7. Sample images from categories (a) *Aquarelle*, (b) *Fox*, and (c) *Beach*.



(a) The average classification accuracy by SVM (b) The average classification accuracy by LapRLS

Fig. 8. Classification results on the Corel image data set. The images selected by each active learning algorithm are used as the training data, and the unselected images are used as the testing data. The evaluations are conducted with 20 randomly generated subsets of the original data set. These figures show the average classification accuracy versus the number of the training samples.

TABLE 5
Classification Results on the Corel Image Data Set (mean ± standard deviation (percent))

k	The classification accuracy by SVM					The classification accuracy by LapRLS				
	Random	SVM_{Active}	AOD	TED	LLR_{Active}	Random	SVM_{Active}	AOD	TED	LLR_{Active}
40	21.8±2.4	21.9±2.4	13.4±1.2	25.9±2.3	**28.9±1.5**	20.3±2.3	20.2±2.4	12.3±1.6	24.9±1.8	**26.6±1.5**
80	30.2±2.2	30.2±2.4	24.2±2.0	32.3±1.6	**36.7±1.6**	27.5±2.2	27.5±2.2	21.7±2.0	29.5±1.4	**33.1±1.4**
120	35.5±2.0	35.6±2.1	30.2±2.3	36.6±1.7	**42.0±1.5**	32.6±2.1	32.5±2.0	26.2±1.9	32.6±1.5	**37.8±1.7**
160	39.0±1.9	39.5±1.9	33.0±1.9	39.5±1.8	**45.2±1.3**	36.1±2.0	36.5±2.0	28.8±1.3	35.4±1.5	**41.4±1.1**
200	41.8±1.7	42.5±1.8	35.6±1.9	42.0±1.4	**47.9±0.9**	39.1±1.8	39.6±1.8	31.2±1.7	38.4±1.3	**44.3±0.8**
240	44.1±1.6	44.9±1.5	37.0±2.2	44.5±1.4	**49.7±0.8**	41.7±1.6	42.3±1.6	32.9±1.7	41.1±1.4	**46.5±1.2**
280	45.9±1.4	46.8±1.4	39.0±1.9	45.8±1.4	**51.1±0.9**	43.7±1.5	44.6±1.6	34.0±1.5	42.6±1.7	**47.8±1.0**
320	47.4±1.3	48.6±1.3	39.8±1.8	46.9±1.3	**52.4±0.7**	45.4±1.4	46.7±1.5	34.9±1.6	43.8±1.3	**49.5±1.0**
360	48.8±1.3	50.1±1.3	41.7±1.5	48.0±1.3	**53.6±0.9**	46.9±1.6	48.3±1.4	35.9±1.7	44.8±1.2	**50.7±1.4**
400	50.0±1.3	51.5±1.3	43.0±1.6	48.8±1.0	**54.3±0.8**	48.3±1.5	49.9±1.4	36.6±2.0	45.4±1.3	**51.7±1.1**
Avg.	40.4±1.7	41.2±1.7	33.7±1.8	41.0±1.5	**46.2±1.1**	38.1±1.8	38.8±1.8	29.5±1.7	37.9±1.5	**42.9±1.2**

classification accuracies, as well as the standard deviations, for each algorithm. As can be seen, the classification accuracy obtained by using only 70 samples selected by our LLR_{Active} algorithm is comparable to or better than those by using 100 samples selected by other algorithms.

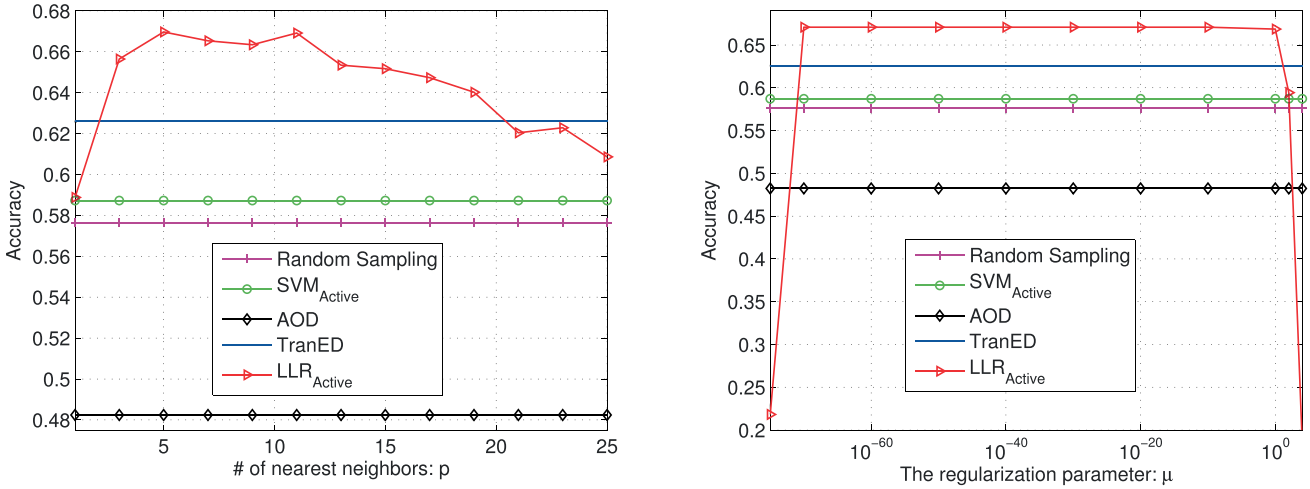
5.2.4 Image Classification

The image data set we used in this experiment consists of 4,000 images of 40 semantic categories, from the Corel image data set. We combine 64-dimensional color histogram and 64-dimensional Color Texture Moment (CTM) to represent the images. The color histogram is calculated using $4 \times 4 \times 4$ bins in HSV space. The Color Texture Moment is proposed by Yu et al. [37]. This way, each image is represented as a 128-dimensional vector. Fig. 7 shows some sample images from the "Aquarelle," "Fox," and "Beach" categories of the Corel data set.

As before, we generate 20 tests by randomly selecting 90 images from each class. Thus, there are 3,600 (40×900) images for each test. We apply each active learning algorithm to select k $(= 10, 20, \ldots, 400)$ training samples. The classification results are shown in Fig. 8. As can be seen, our LLR_{Active} algorithm performs the best. TED outperforms SVM_{Active} and Random Sampling when there are less than 100 training samples. As the number of training samples increases, TED performs slightly worse than SVM_{Active} and Random Sampling. SVM_{Active} and Random Sampling perform comparably to each other. AOD performs the worst for all the cases. Table 5 shows the detailed classification accuracies for each algorithm. Our LLR_{Active} algorithm yields the highest classification accuracy.

5.2.5 Parameter Selection

Our LLR_{Active} has two essential parameters: the number of nearest neighbors p and the regularization parameter μ. These two parameters are empirically set to 10 and 0.01 in the previous experiments. In this section, we examine the



(a) The average classification accuracy versus p (b) The average classification accuracy versus μ

Fig. 9. The impacts of the two parameters p and μ on the performance of LLR_{Active}. For comparison, we let each active learning algorithm select 40 samples as the training data and use SVM as the classifier. The evaluations are conducted with 20 randomly generated subsets from the USPS handwritten digits data set, and we report the average classification accuracy. In (a), the value of μ is fixed at 0.01, and in (b), the value of p is fixed at 10.

impacts of the two parameters on the performance of LLR_{Active}.

The impacts of the two parameters on the three real-world data sets are very similar. For brevity, we just show the results on the USPS handwritten digits data set. For comparison, we let each active learning algorithm select 40 samples as the training data, and use SVM as the classifier. As before, the evaluations are conducted with 20 randomly generated subsets, where each one contains 1,000 samples. Figs. 9a and 9b show how the performance of LLR_{Active} varies with the parameters p and μ, respectively. In Fig. 9a, the value of μ is fixed at 0.01, and in Fig. 9b, the value of p is fixed at 10. As can be seen, LLR_{Active} can achieve significantly better performance over a large range of p and μ. Thus, the parameter selection is not a crucial problem in our algorithm.

6 Conclusions and Future Work

We have introduced a novel active learning algorithm, called LLR_{Active}, to select the most representative points. Given the local geometrical structure of the data space and the coordinates of the selected points, a transductive learning algorithm called LLR is proposed to reconstruct every other point's coordinate. The most representative points are naturally defined as those whose coordinates can be used to best reconstruct the data set. Comparing to previous active learning approaches such as A-Optimal Design, Transductive Experimental Design, and SVM_{Active}, our proposed approach explicitly considers the local manifold structure. Therefore, the selected points by using our approach can improve the classifier the most if they are used as training samples. Experimental results on two synthetic examples and three real-world applications (face recognition, handwritten digits recognition, and image classification) show the effectiveness of our approach.

Central to our algorithm is the locally linear reconstruction scheme. In this work, we adopt the idea from LLE [18] to find the reconstruction coefficients. Thus, each data point can be represented as a linear combination of its neighbors. The advantage of this strategy is that it can well respect the local manifold structure. However, the disadvantage is that the k nearest neighbor search is computationally expensive. The approximate nearest neighbors methods [38], [39] can be used to alleviate the situation. Another possible solution is to adopt the idea of sparse representation for the data [40], [41]. Specifically, each data point is represented by a linear combination of a subset of the data set, and this subset does not necessarily contain the neighboring points. Thus, it would be interesting to explore the sparse reconstruction in the context of active learning.

Moreover, the proposed active learning approach is label independent. Therefore, another possible extension of our work is to consider the use of the labeled samples. Using the label information, we can train an classifier and identify the samples that are most hard to predict. Thus, we may combine our method which selects the most representative samples with the method that selects the most uncertain samples [7]. However, how to find the best trade-off between these two methods is still an open problem. We will investigate this in our future work.

Acknowledgments

This work was supported by Scholarship Award for Excellent Doctoral Student granted by Ministry of Education, National Key Technology R&D Program of China (2008BAH26B00), Program for New Century Excellent Talents in University (NCET-09-0685), National Basic Research Program of China (973 Program) under Grant 2011CB302206, and Natural Science Foundation of China under Grants 90920303 and 60875044.

References

[1] X. Zhu, "Semi-Supervised Learning Literature Survey," Technical Report 1530, Dept. of Computer Sciences, Univ. of Wisconsin–Madison, 2005.

[2] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Schölkopf, "Learning with Local and Global Consistency," *Advances in Neural Information Processing Systems*, vol. 16, pp. 321-328, MIT Press, 2004.

[3] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples," *J. Machine Learning Research,* vol. 7, pp. 2399-2434, 2006.
[4] D.A. Cohn, Z. Ghahramani, and M.I. Jordan, "Active Learning with Statistical Models," *J. Artificial Intelligence Research,* vol. 4, pp. 129-145, 1996.
[5] B. Settles, "Active Learning Literature Survey," Computer Sciences Technical Report 1648, Univ. of Wisconsin–Madison, 2009.
[6] D.D. Lewis and W.A. Gale, "A Sequential Algorithm for Training Text Classifiers," *Proc. ACM SIGIR,* pp. 3-12, 1994.
[7] S. Tong and D. Koller, "Support Vector Machine Active Learning with Applications to Text Classification," *J. Machine Learning Research,* vol. 2, pp. 45-66, 2002.
[8] M. Lindenbaum, S. Markovitch, and D. Rusakov, "Selective Sampling for Nearest Neighbor Classifiers," *Machine Learning,* vol. 54, no. 2, pp. 125-152, 2004.
[9] A. Fujii, T. Tokunaga, K. Inui, and H. Tanaka, "Selective Sampling for Example-Based Word Sense Disambiguation," *Computational Linguistics,* vol. 24, no. 4, pp. 573-597, 1998.
[10] H.S. Seung, M. Opper, and H. Sompolinsky, "Query by Committee," *Proc. Fifth Ann. Workshop Computational Learning Theory,* pp. 287-294, 1992.
[11] P. Melville and R.J. Mooney, "Diverse Ensembles for Active Learning," *Proc. 21st Int'l Conf. Machine learning,* 2004.
[12] N. Roy and A. McCallum, "Toward Optimal Active Learning through Sampling Estimation of Error Reduction," *Proc. 18th Int'l Conf. Machine Learning,* pp. 441-448, 2001.
[13] X. Zhu, J. Lafferty, and Z. Ghahramani, "Combining Active Learning and Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions," *Proc. Int'l Conf. Machine Learning Workshop Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining,* pp. 58-65, 2003.
[14] A. Atkinson, A. Donev, and R. Tobias, *Optimum Experimental Designs, with SAS.* Oxford Univ. Press, 2007.
[15] X. He, W. Min, D. Cai, and K. Zhou, "Laplacian Optimal Design for Image Retrieval," *Proc. ACM SIGIR,* pp. 119-126, 2007.
[16] K. Yu, J. Bi, and V. Tresp, "Active Learning via Transductive Experimental Design," *Proc. 23rd Int'l Conf. Machine Learning,* pp. 1081-1088, 2006.
[17] L. Zhang, C. Chen, W. Chen, J. Bu, D. Cai, and X. He, "Convex Experimental Design Using Manifold Structure for Image Retrieval," *Proc. 17th ACM Int'l Conf. Multimedia,* pp. 45-53, 2009.
[18] S.T. Roweis and L.K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science,* vol. 290, no. 5500, pp. 2323-2326, Dec. 2000.
[19] J. Tenenbaum, V. de Silva, and J. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science,* vol. 290, no. 5500, pp. 2319-2323, 2000.
[20] M. Belkin and P. Niyogi, "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering," *Advances in Neural Information Processing Systems,* vol. 14, pp. 585-591, MIT Press, 2002.
[21] X. He and P. Niyogi, "Locality Preserving Projections," *Advances in Neural Information Processing Systems,* vol. 16, pp. 153-160, MIT Press, 2004.
[22] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning.* Springer, 2009.
[23] S.P. Asprey and S. Macchietto, "Designing Robust Optimal Dynamic Experiments," *J. Process Control,* vol. 12, no. 4, pp. 545-556, 2002.
[24] R.H. Hardin and N.J.A. Sloane, "A New Approach to the Construction of Optimal Designs," *J. Statistical Planning and Inference,* vol. 37, no. 3, pp. 339-369, 1993.
[25] S.B. Grary and C. Spera, "Optimal Experimental Design for Combinatorial Problems," *Computational Economics,* vol. 9, no. 3, pp. 241-255, Aug. 1996.
[26] X. Li and Y. Pang, "Deterministic Column-Based Matrix Decomposition," *IEEE Trans. Knowledge and Data Eng.,* vol. 22, no. 1, pp. 145-149, Jan. 2010.
[27] C. Chen, L. Zhang, J. Bu, C. Wang, and W. Chen, "Constrained Laplacian Eigenmap for Dimensionality Reduction," *Neurocomputing,* vol. 73, nos. 4-6, pp. 951-958, 2010.
[28] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and Semi-Supervised Learning on Large Graphs," *Proc. 17th Ann. Conf. Computational Learning Theory,* pp. 624-638, 2004.
[29] X. Zhu, Z. Ghahramani, and J.D. Lafferty, "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions," *Proc. 20th Int'l Conf. Machine Learning,* pp. 912-919, 2003.
[30] C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery,* vol. 2, no. 2, pp. 121-167, 1998.
[31] Y. Chen, T.A. Davis, W.W. Hager, and S. Rajamanickam, "Algorithm 887: Cholmod, Supernodal Sparse Cholesky Factorization and Update/Downdate," *ACM Trans. Math. Software,* vol. 35, no. 3, pp. 1-14, 2008.
[32] G.H. Golub and C.F. Van Loan, *Matrix Computations,* third ed. Johns Hopkins Univ. Press, 1996.
[33] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge Univ. Press, 2004.
[34] C.-C. Chang and C.-J. Lin LIBSVM: A Library for Support Vector Machines, http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.
[35] M. Grant and S. Boyd, "CVX: Matlab Software for Disciplined Convex Programming, Version 1.21," http://cvxr.com/cvx, Oct. 2010.
[36] M. Grant and S. Boyd, "Graph Implementations for Nonsmooth Convex Programs," *Recent Advances in Learning and Control,* V. Blondel, S. Boyd, and H. Kimura, eds., vol. 371, pp. 95-110, Springer, 2008.
[37] H. Yu, M. Li, H. Jiang Zhang, and J. Feng, "Color Texture Moments for Content-Based Image Retrieval," *Proc. Int'l Conf. Image Processing,* pp. 24-28, 2002.
[38] M. Muja and D.G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration," *Proc. Int'l Conf. Computer Vision Theory and Application,* pp. 331-340, 2009.
[39] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, and A.Y. Wu, "An Optimal Algorithm for Approximate Nearest Neighbor Searching Fixed Dimensions," *J. ACM,* vol. 45, no. 6, pp. 891-923, 1998.
[40] H. Lee, A. Battle, R. Raina, and A.Y. Ng, "Efficient Sparse Coding Algorithms," *Advances in Neural Information Processing Systems,* vol. 19, pp. 801-808, MIT Press, 2007.
[41] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, and Y. Ma, "Robust Face Recognition via Sparse Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 2, pp. 210-227, Feb. 2009.

Lijun Zhang received the BS degree in computer science from Zhejiang University, China, in 2007. He is currently working toward the PhD degree in computer science at Zhejiang University. His research interests include machine learning, information retrieval, and data mining. He is a student member of the IEEE.

Chun Chen received the BS degree in mathematics from Xiamen University, China, in 1981 and the MS and PhD degrees in computer science from Zhejiang University, China, in 1984 and 1990, respectively. He is a professor in the College of Computer Science, Zhejiang University. His research interests include information retrieval, data mining, computer vision, computer graphics, and embedded technology. He is a member of the IEEE.

Jiajun Bu received the BS and PhD degrees in computer science from Zhejiang University, China, in 1995 and 2000, respectively. He is a professor inthe College of Computer Science, Zhejiang University. His research interests include embedded system, data mining, information retrieval, and mobile databases. He is a member of the IEEE.

Deng Cai received the bachelor's and master's degrees in automation from Tsinghua University, in 2000 and 2003, respectively, and the PhD degree in computer science from the University of Illinois at Urbana Champaign in 2009. He is an associate professor at the State Key Lab of CAD & CG, College of Computer Science, Zhejiang University, China. His research interests include machine learning, data mining, and information retrieval. He is a member of the IEEE.

Xiaofei He received the BS degree in computer science from Zhejiang University, China, in 2000 and the PhD degree in computer science from the University of Chicago in 2005. He is a professor at the State Key Lab of CAD & CG, Zhejiang University, China. Prior to joining Zhejiang University in 2007, he was a research scientist at Yahoo! Research Labs, Burbank, California. His research interests include machine learning, information retrieval, and computer vision. He is a senior member of the IEEE.

Thomas S. Huang received the ScD degree in electrical engineering from the Massachusetts Institute of Technology (MIT). He was on the faculty of MIT and Purdue University. He joined the University of Illinois at Urbana Champaign in 1980 and is currently the William L. Everitt Distinguished Professor of Electrical and Computer Engineering, a research professor at the Coordinated Science Laboratory, a professor at the Center for Advanced Study, and a cochair of the Human Computer Intelligent Interaction major research theme of the Beckman Institute for Advanced Science and Technology. He is a member of the National Academy of Engineering and has received numerous honors and awards, including the IEEE Jack S. Kilby Signal Processing Medal (with A. Netravali) and the King-Sun Fu Prize of the International Association of Pattern Recognition. He has published 21 books and more than 600 technical papers in network theory, digital holography, image and video compression, multimodal human computer interfaces, and multimedia databases. He is a life fellow of the IEEE.