G-Optimal Design with Laplacian Regularization

Chun Chen[1], Zhengguang Chen[1], Jiajun Bu[1], Can Wang[1*], Lijun Zhang[1], Cheng Zhang[2]
[1] College of Computer Science, Zhejiang University, Hangzhou, China
{Chenc, cerror, bjj, wcan, zljzju}@zju.edu.cn
[*]Corresponding Author
[2] China Disabled Persons' Federation Information Center, Beijing, China
zhangcheng@cdpf.org.cn

Abstract

In many real world applications, labeled data are usually expensive to get, while there may be a large amount of unlabeled data. To reduce the labeling cost, active learning attempts to discover the most informative data points for labeling. Recently, Optimal Experimental Design (OED) techniques have attracted an increasing amount of attention. OED is concerned with the design of experiments that minimizes variances of a parameterized model. Typical design criteria include D-, A-, and E-optimality. However, all these criteria are based on an ordinary linear regression model which aims to minimize the empirical error whereas the geometrical structure of the data space is not well respected. In this paper, we propose a novel optimal experimental design approach for active learning, called Laplacian G-Optimal Design (LapGOD), which considers both discriminating and geometrical structures. By using Laplacian Regularized Least Squares which incorporates manifold regularization into linear regression, our proposed algorithm selects those data points that minimizes the maximum variance of the predicted values on the data manifold. We also extend our algorithm to nonlinear case by using kernel trick. The experimental results on various image databases have shown that our proposed LapGOD active learning algorithm can significantly enhance the classification accuracy if the selected data points are used as training data.

Introduction

In many information processing tasks, labels are often expensive to obtain while a vast amount of unlabeled data are easily available. To address this problem, active learning techniques attempt to select the most informative data points for labeling (Cohn, Ghahramani, and Jordan 1996). Active learning is also referred to as experimental design in statistics, whose intent is to select the most informative samples to learn a predictive function with minimum variance and expected prediction error.

In experimental design, the sample $\mathbf{x}$ is referred to as experiment and its label y is referred to as measurement. The Optimum Experimental Design (OED) (Atkinson, Donev, and Tobias 2007) is concerned with the design of experiments that minimizes variances of a parameterized model by maximizing confidence in a given model, minimizing parameter variances for system identification, or minimizing the model's output variance. There are many classical OED approaches, including A-, D-, E-, and G-Optimality. All of these approaches are based on least squares regression model and only take into account the labeled data when evaluating the learned model.

Another category of active learning methods is based on Support Vector Machines (Tong and Koller 2002). SVM active learning selects those points closest to the decision boundary. Therefore, SVM active learning is label dependent and can only be applied when there is an initial classifier. Unlike SVM active learning, OED techniques are label independent and hence is more applicable. Another drawback of SVM active learning is the small training size problem. When the number of training examples is small, SVM may deliver poor classification accuracy. This poor classification can significantly impair the identification of the informative examples and consequently reduce the learning performance.

Motivated by recent progresses on semi-supervised learning (Belkin, Niyogi, and Sindhwani 2006; Chapelle, Schölkopf, and Zien 2006) and optimal experimental design (Yu, Bi, and Tresp 2006), we propose in this paper a novel active learning algorithm called Laplacian G-optimal Design (LapGOD). Different from the traditional G-optimal design which effectively sees only the Euclidean structure of data space, we employ the manifold assumption that if two data points are sufficiently close to each other, their labels tend to be the same. Based on Laplacian Regularized Least Squares (LapRLS), our approach aims to minimize the maximum variance of the predicted values. Moreover, LapGOD may be conducted either in the original space or in the reproducing kernel Hilbert space (RKHS) into which data points are mapped. This gives rise to kernel LapGOD. There are also other works of OED based active learning by using graph Laplacian. (He et al. 2007) used graph Laplacian in the formulation of A-optimal design. (He, Ji, and Bao 2009b; He 2010) used a similar scenario for Laplacian regularized D-optimal design. Comparing to these works, our proposed approach uses a different optimality criteria. Moreover, the approaches presented in (He et al. 2007; He, Ji, and Bao

2009b; He 2010; Zhang et al. 2009) find an approximate solution which minimize an upper bound of the objective function, while our approach directly minimizes the objective function.

Related Work

The generic problem of active learning can be formalized as follows. Given a set of data points $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\}$, where each $\mathbf{x}_i$ is a d-dimensional vector, active learning aims to find a subset $\mathcal{V} \subset \mathcal{X}$ containing the most informative data points. That is, the subset $\mathcal{V}$ can improve the classifier the most if it is labeled and used as training data.

Optimal Experimental Design (OED) has recently received considerable attention due to its theoretical foundation and practical effectiveness (Flaherty, Jordan, and Arkin 2005). Consider a linear regression model

$$y = \boldsymbol{\theta}^T \mathbf{x} + \epsilon,$$

where $\boldsymbol{\theta}$ is the weight vector and ϵ is an unknown error with zero mean and variance σ^2. OED attempts to select the most informative experiments (or data points) to learn a prediction function

$$f(\mathbf{x}) = \boldsymbol{\theta}^T \mathbf{x}, \tag{1}$$

so that the expected prediction error can be minimized (Atkinson, Donev, and Tobias 2007).

Let $\mathcal{V} = \{\mathbf{v}_1, \cdots, \mathbf{v}_k\} \subset \mathcal{X}$ denote the selected subset and $\mathbf{y} = (y_1, \cdots, y_k)^T$ denote the label vector. Define $V = (\mathbf{v}_1, \cdots, \mathbf{v}_k)$. The maximum likelihood estimate for the weight vector is given by

$$\widehat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta}} \sum_{i=1}^{k} (\boldsymbol{\theta}^T \mathbf{v}_i - y_i)^2 = (VV^T)^{-1} V\mathbf{y} \tag{2}$$

The covariance matrix of $\widehat{\boldsymbol{\theta}}$ can be computed as

$$Cov(\widehat{\boldsymbol{\theta}}) = \sigma^2 (VV^T)^{-1} \tag{3}$$

The most informative data points are thus defined as those minimizing $Cov(\widehat{\boldsymbol{\theta}})$.

The most common scalar measures of $Cov(\widehat{\boldsymbol{\theta}})$ include A-, D-, and E-optimality criteria. D-optimality minimizes the determinant of $Cov(\widehat{\boldsymbol{\theta}})$, and thus minimizes the volume of the confidence region. In A-optimality the trace of $Cov(\widehat{\boldsymbol{\theta}})$, i.e. the total variance of the parameter estimates, is minimized. E-optimality minimizes the maximum eigenvalue of $Cov(\widehat{\boldsymbol{\theta}})$, and thus minimizes the size of the major axis of the confidence region.

The major drawback of the above optimality criteria is that they do not directly characterize the quality of predictions on test data. G-optimality overcomes this limitation by minimizing the maximum variance of the predicted values. For each data point $\mathbf{x} \in \mathcal{X}$, the variance of its prediction value is $\mathbf{x}^T Cov(\widehat{\boldsymbol{\theta}})\mathbf{x}$. Thus, G-optimal design selects the most informative data points by solving the following optimization problem:

$$\min_{V} \max_{\mathbf{x} \in \mathcal{X}} \mathbf{x}^T (VV^T)^{-1} \mathbf{x} \tag{4}$$

Laplacian G-optimal Design

In this section, we introduce our Laplacian G-Optimal Design algorithm for active learning. LapGOD is fundamentally based on Laplacian Regularized Least Squares (LapRLS) (Belkin, Niyogi, and Sindhwani 2006).

Estimated Parameter of Laplacian Regularized Least Squares

Laplacian Regularized Least Squares (LapRLS) (Belkin, Niyogi, and Sindhwani 2006) is a semi-supervised learning approach which tries to discover both discriminant and geometrical structures of the data space.

LapRLS first constructs an affinity graph $\mathcal{G}$ with weight matrix S. Each node of $\mathcal{G}$ corresponds to a data point and S_{ij} is a similarity measure between $\mathbf{x}_i$ and $\mathbf{x}_j$. Then LapRLS minimizes the following regularized risk (Belkin, Niyogi, and Sindhwani 2006):

$$\min_f \frac{\alpha}{m} \|f\|_{\mathcal{G}}^2 + \beta \|f\|^2 + \frac{1}{k} \sum_{i=1}^{k} l(\mathbf{v}_i, y_i, f), \tag{5}$$

where $\alpha \geq 0$ and $\beta \geq 0$ are the regularization parameters and l is the loss function. The graph regularizer $\|f\|_{\mathcal{G}}^2$ is defined as

$$\|f\|_{\mathcal{G}}^2 = \mathbf{f}^T \nabla_{\mathcal{G}} \mathbf{f} \tag{6}$$

where $\mathbf{f} = \big(f(\mathbf{x}_1), \cdots, f(\mathbf{x}_m)\big)^T$. $\nabla_{\mathcal{G}} \in R^{m \times m}$ is a function of $\mathcal{G}$ which determines the specific form of regularization imposed. Belkin et al. suggest using graph laplacian $L = D - S$, where D is a diagonal matrix and $D_{ii} = \sum_j S_{ij}$. There are many choices of the weight matrix S. A simple definition is as follows:

$$S_{ij} = \begin{cases} 1, & \mathbf{x}_i \in \mathcal{N}_k(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i), \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where $\mathcal{N}_k(\mathbf{x})$ denotes the set of k nearest neighbors of $\mathbf{x}$.

Finally, the loss function $l(\mathbf{x}_i, y_i, f)$ can be defined as

$$l(\mathbf{v}_i, y_i, f) = (f(\mathbf{v}_i) - y_i)^2. \tag{8}$$

Consider a linear function $f(\mathbf{x}) = \boldsymbol{\theta}^T \mathbf{x}$. Thus, the optimal solution is given by

$$\widehat{\boldsymbol{\theta}}_{LapRLS} = (VV^T + \alpha XLX^T + \beta I)^{-1} V\mathbf{y} \tag{9}$$

Besides semi-supervised learning, graph Laplacian has also been widely used in other learning tasks such as dimensionality reduction (Cai et al. 2006; He et al. 2005; He, Cai, and Min 2005; He, Ji, and Bao 2009a; Min, Lu, and He 2004).

The Objective Function of LapGOD

Similar to conventional optimal experimental design techniques, we first compute the parameter covariance matrix of LapRLS. Define $P = VV^T + \alpha XLX^T + \beta I$. It is easy to see that P is symmetric. The covariance matrix of $\widehat{\boldsymbol{\theta}}_{LapRLS}$ is (He et al. 2007; He 2010):

$$\begin{aligned} &\operatorname{Cov}(\widehat{\boldsymbol{\theta}}_{LapRLS}) \\ =&P^{-1} V \operatorname{Cov}(\mathbf{y}) V^T P^{-1} \\ =&\sigma^2 P^{-1} V V^T P^{-1} \end{aligned} \tag{10}$$

For each data point $\mathbf{x} \in \mathcal{X}$, the variance of its predicted value is $\sigma^2 \mathbf{x}^T P^{-1} V V^T P^{-1} \mathbf{x}$. Thus, the maximum prediction variance is given by

$$\max_{\mathbf{x}_i \in \mathcal{X}} \{\mathbf{x}_i^T P^{-1} V V^T P^{-1} \mathbf{x}_i\} \tag{11}$$

Note that, the maximum prediction variance is dependent on the selected data points, i.e. V. Finally, the most informative data points are defined as those minimizing the maximum prediction variance. The objective function of our LapGOD algorithm is formally stated below:

Definition Laplacian G-optimal Design (LapGOD):

$$\min_{V} \max_{\mathbf{x}_i \in \mathcal{X}} \{\mathbf{x}_i^T P^{-1} V V^T P^{-1} \mathbf{x}_i\} \tag{12}$$

with variable $V = (\mathbf{v}_1, \cdots, \mathbf{v}_k)$ and $\mathbf{v}_i \in \mathcal{X}$, $i = 1, \cdots, k$.

The Algorithm

In this section, we describe a sequential optimization scheme to solve (12).

Suppose a set of $k(\geq 0)$ points $\mathcal{V}_k = \{\mathbf{v}_1, \cdots, \mathbf{v}_k\} \subset \mathcal{X}$ have been selected. Let $V_k = (\mathbf{v}_1, \cdots, \mathbf{v}_k)$ be a $d \times k$ matrix. We define

$$P_k = V_k V_k^T + \alpha X L X^T + \beta I, k \geq 1,$$

and

$$P_0 = \alpha X L X^T + \beta I.$$

The $(k+1)$-th point $\mathbf{v}_{k+1}$ can be selected by solving the following problem:

$$\begin{aligned} \mathbf{v}_{k+1} = \operatorname*{argmin}_{\mathbf{v} \in \mathcal{X} - \mathcal{V}_k} \max_{\mathbf{x}_i \in \mathcal{X}} \{&\mathbf{x}_i^T \left(P_k + \mathbf{v}\mathbf{v}^T\right)^{-1} \\ &\left(V_k V_k^T + \mathbf{v}\mathbf{v}^T\right)\left(P_k + \mathbf{v}\mathbf{v}^T\right)^{-1} \mathbf{x}_i\} \end{aligned} \tag{13}$$

For each data point $\mathbf{v} \in \mathcal{X} - \mathcal{V}_k$, its corresponding maximum prediction variance can be computed as

$$\max_{\mathbf{x}_i \in \mathcal{X}} \{\mathbf{x}_i^T \left(P_k + \mathbf{v}\mathbf{v}^T\right)^{-1} \left(V_k V_k^T + \mathbf{v}\mathbf{v}^T\right)\left(P_k + \mathbf{v}\mathbf{v}^T\right)^{-1} \mathbf{x}_i\} \tag{14}$$

$\mathbf{v}_{k+1}$ is obtained as the one minimizing the maximum prediction variance. The most expensive computation in (13) is the matrix inverse $(P_k + \mathbf{v}\mathbf{v}^T)^{-1}$. By using Sherman-Morrison-Woodbury formula (Golub and Loan 1996), $(P_k + \mathbf{v}\mathbf{v}^T)^{-1}$ can be rewritten as follows:

$$(P_k + \mathbf{v}\mathbf{v}^T)^{-1} = P_k^{-1} - \frac{P_k^{-1}\mathbf{v}\mathbf{v}^T P_k^{-1}}{1 + \mathbf{v}^T P_k^{-1} \mathbf{v}}. \tag{15}$$

Therefore, we only need to compute the inverse of P_0. The inverse of $P_{k+1} (= P_k + \mathbf{v}_{k+1}\mathbf{v}_{k+1}^T)$ can be computed according to (15) once $\mathbf{v}_k$ is obtained. The algorithmic procedure is summarized in Algorithm 1.

Nonlinear Generalization of LapGOD

Most of traditional optimal experimental design techniques are based on linear regression model. However, in many real world applications, the data may not be linearly separable. In this section, we discuss how to generalize our LapGOD algorithm to nonlinear case by performing experimental design in Reproducing Kernel Hilbert Space (RKHS).

Algorithm 1 Sequential optimization approach for LapGOD

Input: A set of data points $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$, the number of points to be selected (n), the regularization parameters (α, β).

Output: The n most informative points: $\mathcal{V} = \{\mathbf{v}_1, \cdots, \mathbf{v}_n\} \subset \mathcal{X}$.

Construct a nearest neighbor graph with weight matrix W (see Eq. 7) and compute the Laplacian matrix L.
$X \leftarrow (\mathbf{x}_1, \cdots, \mathbf{x}_m)^T$.
$P_0 \leftarrow \alpha X L X^T + \beta I$, compute P_0^{-1}.
$\mathcal{V} \leftarrow \emptyset$.
$V_0 \leftarrow 0$.
for $k = 0$ to $n-1$ **do**
  for $i = 1$ to m **do**
    if $\mathbf{x}_i \in \mathcal{X} - \mathcal{V}$ **then**
      $A_i \leftarrow P_k^{-1} - \frac{P_k^{-1}\mathbf{x}_i\mathbf{x}_i^T P_k^{-1}}{1+\mathbf{x}_i^T P_k^{-1}\mathbf{x}_i}$
      $g(\mathbf{x}_i) = \max_{\mathbf{x}_j \in \mathcal{X}} \{\mathbf{x}_j^T A_i \left(V_k V_k^T + \mathbf{x}_i \mathbf{x}_i^T\right) A_i \mathbf{x}_j\}$
    end if
  end for
  $\mathbf{v}_{k+1} = \operatorname{argmin}_{\mathbf{x} \in \mathcal{X} - \mathcal{V}} g(\mathbf{x})$.
  $\mathcal{V} \leftarrow \mathcal{V} \cup \mathbf{v}_{k+1}$.
  $V_{k+1} \leftarrow (\mathbf{v}_1, \cdots, \mathbf{v}_{k+1})$
  $P_{k+1}^{-1} \leftarrow P_k^{-1} - \frac{P_k^{-1}\mathbf{v}_{k+1}\mathbf{v}_{k+1}^T P_k^{-1}}{1+\mathbf{v}_{k+1}^T P_k^{-1}\mathbf{v}_{k+1}}$
end for
return $\mathcal{V}$

Performing LapRLS in RKHS

Let $\mathcal{H}$ denote a RKHS associated with a kernel function $\mathcal{K}(\cdot,\cdot)$. We can rewrite the regularized risk Eq. 5 in the RKHS:

$$\min_{f \in \mathcal{H}} \frac{\alpha}{m} \|f\|_{\mathcal{G}}^2 + \beta \|f\|_{\mathcal{H}}^2 + \frac{1}{k} \sum_{i=1}^{k} l(\mathbf{v}_i, y_i, f) \tag{16}$$

By Representer Theorem(Schölkopf and Smola 2002), the optimal solution $\widehat{f}$ is an expansion of kernel functions over both the labeled and unlabeled data:

$$\widehat{f}(\mathbf{x}) = \sum_{i=1}^{m} \lambda_i \mathcal{K}(\mathbf{x}, \mathbf{x}_i) \tag{17}$$

Let $\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_m)^T$. Let K_{XV} be a $m \times k$ matrix ($K_{XV,ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{v}_j)$), K_{VX} be a $k \times m$ matrix ($K_{VX,ij} = \mathcal{K}(\mathbf{v}_i, \mathbf{x}_j)$, and K_{XX} be a $m \times m$ matrix ($K_{XX,ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$). Thus, the optimal $\boldsymbol{\lambda}$ minimizing Eq. (16) is given by (Belkin, Niyogi, and Sindhwani 2006; He et al. 2007; He 2010):

$$\boldsymbol{\lambda} = (K_{XV} K_{VX} + \alpha K_{XX} L K_{XX} + \beta K_{XX})^{-1} K_{XV} \mathbf{y} \tag{18}$$

The Algorithm

Define $H = K_{XV} K_{VX} + \alpha K_{XX} L K_{XX} + \beta K_{XX}$. The covariance of $\boldsymbol{\lambda}$ can be computed as follows:

$$\begin{aligned} &\operatorname{Cov}(\boldsymbol{\lambda}) \\ =&H^{-1} K_{XV} \operatorname{Cov}(\mathbf{y}) K_{VX} H^{-1} \\ =&\sigma^2 H^{-1} K_{XV} K_{VX} H^{-1} \end{aligned} \tag{19}$$

Let $\mathbf{u}_i$ be the i-th column vector of K_{XX}, and $\mathcal{U}$ be the set of $\{\mathbf{u}_1, \cdots, \mathbf{u}_m\}$. Clearly, $\mathbf{u}_i = (\mathcal{K}(\mathbf{x}_i, \mathbf{x}_1), \cdots, \mathcal{K}(\mathbf{x}_i, \mathbf{x}_m))^T$ and $\mathbf{u}_i$ corresponds to $\mathbf{x}_i \in \mathcal{X}$. Similar to selecting $\mathbf{x}_i$ in the original space, here we select $\mathbf{u}_i$ in the RKHS. Let $\mathcal{W}_k = \{\mathbf{w}_1, \cdots, \mathbf{w}_k\} \subset \mathcal{U}$ denote the set of selected points and $\mathbf{w}_i$ corresponds to $\mathbf{v}_i \in \mathcal{X}$. Let $W_k = (\mathbf{w}_1, \cdots, \mathbf{w}_k)$ be a $m \times k$ matrix. Clearly, $W_k = K_{XV}$. Thus, the nonlinear LapGOD algorithm is formally defined below:

Definition Kernel Laplacian G-optimal Design:

$$\min_{W} \max_{\mathbf{u}_i \in \mathcal{U}} \{\mathbf{u}_i^T H^{-1} W W^T H^{-1} \mathbf{u}_i\} \tag{20}$$

with variable $W = (\mathbf{w}_1, \cdots, \mathbf{w}_k)$ and $\mathbf{w}_i \in \mathcal{U}$, $i = 1, \cdots, k$.

In the following, we describe a sequential optimization scheme to solve (20). Suppose $k(\geq 0)$ points have been selected. We define

$$\begin{aligned} H_k &= K_{XV_k} K_{V_k X} + \alpha K_{XX} L K_{XX} + \beta K_{XX} \\ &= W_k W_k^T + \alpha K_{XX} L K_{XX} + \beta K_{XX}, \end{aligned} \tag{21}$$

and

$$H_0 = \alpha K_{XX} L K_{XX} + \beta K_{XX}.$$

Then the (k+1)-th point is selected by solving the following optimization problem:

$$\begin{aligned} \mathbf{w}_{k+1} = \operatorname*{argmin}_{\mathbf{u} \in \mathcal{U} - \mathcal{W}_k} \max_{\mathbf{u}_i \in \mathcal{U}} \{ & \mathbf{u}_i^T (H_k + \mathbf{u}\mathbf{u}^T)^{-1} \\ & (W_k W_k^T + \mathbf{u}\mathbf{u}^T)(H_k + \mathbf{u}\mathbf{u}^T)^{-1} \mathbf{u}_i \} \end{aligned} \tag{22}$$

Similar to linear LapGOD, the inverse of $H_k + \mathbf{u}\mathbf{u}^T$ can be computed as follows:

$$(H_k + \mathbf{u}\mathbf{u}^T)^{-1} = H_k^{-1} - \frac{H_k^{-1} \mathbf{u}\mathbf{u}^T H_j^{-1}}{1 + \mathbf{u}^T H_k^{-1} \mathbf{u}} \tag{23}$$

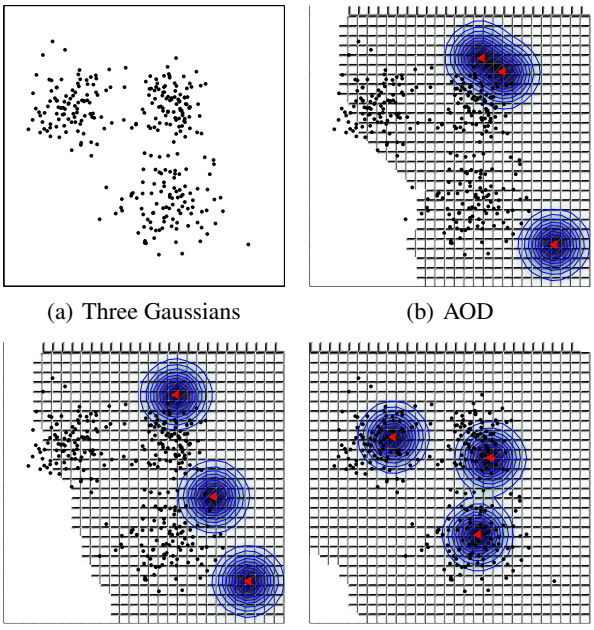
As can be seen, the computation of nonlinear LapGOD is essentially the same as that of linear LapGOD. The only difference is that the data points $\mathbf{x}_i$'s are replaced by $\mathbf{u}_i$'s.

Experimental Results

Several experiments were carried out to demonstrate the effectiveness of our algorithm. We compare our LapGOD algorithm with random sampling, SVM active learning (SVM$_{active}$) (Tong and Koller 2002), and two optimal experimental design methods, that are, A-Optimal Design (AOD) and G-Optimal Designs (GOD) (Atkinson, Donev, and Tobias 2007).

Toy Example

A simple synthetic example is given in Fig.1. The data set is generated from a mixture of three Gaussians. If the selected point is close to the centers of the Gaussians, then the low predictive variance area (the shadow area in the figure) covers a space with most data samples present. We apply AOD, GOD, and LapGOD to select three points. Note that, SVM$_{active}$ can not be applied in this case due to the lack of labeled points. As can be seen, both AOD and GOD tend to selected points with large norm. The three points selected by our LapGOD algorithm are very close to the centers of the three Gaussians, and hence can best represent the original data set.



(a) Three Gaussians (b) AOD (c) GOD (d) LapGOD

Figure 1: Data selection by active learning algorithms. The red points are the selected points. Clearly, the points selected by LapGOD can best represent the three Gaussians.

Experimental Settings

Two face data sets were used in our experiments. The first one is the Yale face database [1] and the second one is the AT&T face database [2]. The size of each face image in all the experiments is 32×32 pixels, with 256 gray levels per pixel. Thus, each face image can be represented as a point in 1024-dimensional space.

We apply different active learning algorithms to select the face images for training. Then linear regression or SVM classifier is trained to perform face recognition. Since face recognition is essentially a multi-class classification problem, we adopt the *one-vs-all* strategy. We compare the following active learning approaches for face recognition:

- A-Optimal Design + linear regression (**AOD**)
- G-Optimal Design + linear regression (**GOD**)
- SVM$_{active}$ + linear regression (**SVM$_{active}$ + Reg**)
- **SVM$_{active}$ + SVM**
- Laplacian G-Optimal Design + linear regression (**LapGOD**)

Note that, SVM$_{active}$ can not be applied at the beginning when there is no labeled images available. Therefore, for SVM$_{active}$, we first randomly select 20 training images and then apply SVM$_{active}$ to select more training images. In our

[1] http://cvc.yale.edu/projects/yalefaces/yalefaces.html

[2] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

Table 1: Recognition error rates on Yale Database (mean±std-dev).

k	Recognition error rate (%)				
	SVM_{Active}+Reg	SVM_{Active}+SVM	AOD	GOD	LapGOD
20	64.4±4.9	64.2±6.0	67.7±3.3	**61.7±5.7**	63.0±4.3
30	54.7±6.6	57.0±5.0	57.8±4.4	52.7±6.3	**46.5±2.6**
40	46.1±7.2	47.2±6.0	50.6±4.1	48.0±5.6	**39.8±3.7**
50	38.2±7.3	43.4±7.6	45.5±4.0	42.3±4.1	**30.6±3.4**
60	35.2±6.0	37.2±6.4	42.0±2.5	38.4±3.8	**27.2±3.5**
70	30.2±5.4	33.9±6.6	38.1±3.4	32.4±6.2	**24.4±3.3**
80	27.6±7.0	29.9±7.0	31.1±4.0	30.5±5.6	**22.6±5.2**
Avg.	42.3±6.3	44.7±6.4	47.5±3.6	43.7±5.3	**36.3±3.7**

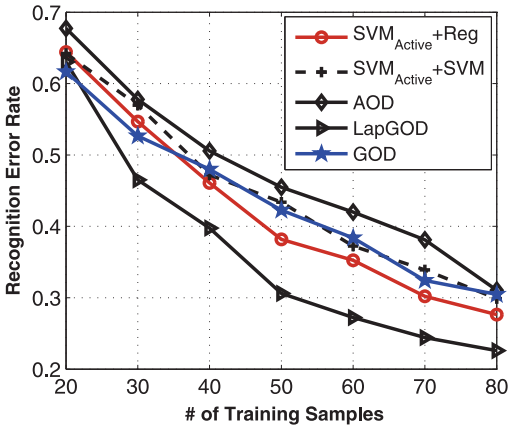


Figure 2: Recognition error rate vs. number of training (selected) samples on Yale face database. As can be seen, our LapGOD algorithm significantly outperforms the other four algorithms.

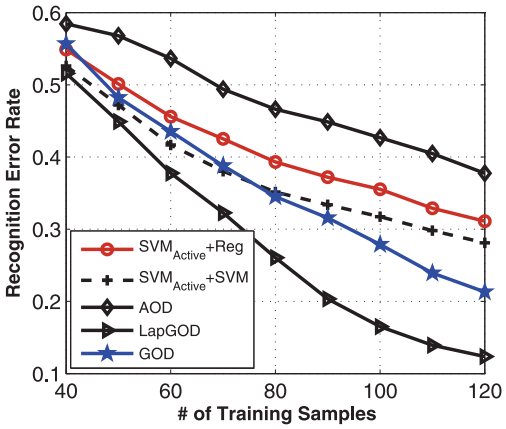


Figure 3: Recognition error rate vs. number of training (selected) samples on AT&T face database. As can be seen, our LapGOD algorithm significantly outperforms the other four algorithms.

algorithm, the parameter k (number of nearest neighbors) is set to be 2, and the regularization parameters α, β are set to be 0.01, 0.001, respectively.

Face Recognition on Yale Database

The Yale database consists of 165 gray scale images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised, and wink.

For each individual, we randomly select ten images. Thus, we get 150 images in total. We repeat this process ten times and perform face recognition for each test. The average recognition error rate over the ten tests is recorded. The active learning algorithms are applied to select different numbers of images for training and all the unlabeled images are used for testing. Figure 2 shows the recognition error rate as a function of the number of training (selected) images. As can be seen, our LapGOD algorithm outperforms the other algorithms in most cases. For all the algorithms, the recognition error rates decrease with more training examples. SVM_{active}+Regression performs the second best. SVM_{active}+SVM and GOD performs comparably to each other. AOD performs the worst.

Table 1 shows the detailed recognition error rates, together with standard deviations, for different algorithms. When 80 images are selected for training, the recognition error rates obtained by SVM_{active}+Reg, SVM_{active}+SVM, AOD, GOD, and LapGOD are 27.6%, 29.9%, 31.1%, 30.5%, and 22.6%, respectively. Comparing to the second best algorithm, i.e. SVM_{active}+Reg, our LapGOD algorithm achieves 18.1% relative error reduction. Also, we can see that AOD and LapGOD achieves the smallest standard deviation, whereas SVM active learning has the largest standard deviation. That is to say the predictive function is more stable by using our method.

Face Recognition on AT&T Database

The AT&T face database consists of a total of 400 face images, of a total of 40 subjects (10 samples per subject). The images were taken at different times, varying the lighting, facial expressions (open or closed eyes, smiling or non-smiling) and facial details (glasses or no glasses). The images were taken with a tolerance for some tilting and rotation of the face up to 20 degrees.

We randomly select 8 images for each individual. Thus, we get 320 images in total. The active learning algorithms are performed on this data set to select different numbers

Table 2: Recognition error rates on AT&T database (mean±std-dev).

k	Recognition error rate (%)				
	SVM_{Active}+Reg	SVM_{Active}+SVM	AOD	GOD	LapGOD
40	54.9±3.4	52.6±3.7	58.4±1.9	55.7±4.2	**51.6±3.0**
50	50.1±3.0	47.2±4.5	56.8±2.1	48.2±6.3	**44.9±4.5**
60	45.6±2.4	41.7±3.5	53.6±2.0	43.5±6.4	**37.8±2.1**
70	42.5±3.6	38.0±4.1	49.4±2.2	38.8±4.3	**32.3±2.3**
80	39.3±4.2	35.2±4.9	46.6±2.7	34.5±3.5	**26.1±3.1**
90	37.2±4.3	33.4±5.1	44.9±2.6	31.6±3.9	**20.4±3.4**
100	35.5±4.0	31.7±4.5	42.7±2.5	27.9±4.2	**16.5±2.4**
110	32.9±4.5	29.8±5.0	40.5±2.1	23.9±3.8	**13.9±2.7**
120	31.1±5.1	28.1±5.4	37.7±2.6	21.3±3.6	**12.4±1.8**
Avg.	41.0±3.8	37.5±4.5	47.9±2.3	36.2±4.5	**28.4±2.8**

of training images. The unlabeled images are used for testing. We repeat this process ten times and compute the average recognition error rates. Figure 3 shows the plot of error rate versus the number of training samples. As can be seen, our LapGOD algorithm consistently outperforms all the other algorithms in all cases. Both SVM_{active}+Reg and SVM_{active}+SVM perform worse than GOD. This is probably because the AT&T database has more subjects than the Yale database and hence the estimated decision boundary by SVM may not be accurate enough. Similar to the Yale database, AOD preforms the worst.

Table 2 shows the detailed recognition error rates, together with standard deviations, for different algorithms. When 120 images are selected for training, the error rates obtained by SVM_{active}+Reg, SVM_{active}+SVM, AOD, GOD, and LapGOD are 31.1%, 28.1%, 37.7%, 21.3%, and 12.4%, respectively. Comparing to the second best algorithm, i.e. GOD, our LapGOD algorithm achieves 41.8% relative error reduction.

Conclusion

We have introduced a novel active learning algorithm called Laplacian G-Optimal Design. Our algorithm is motivated from recent progresses on graph based semi-supervised learning and optimal experimental design. Based on Laplacian Regularized Least Squares, we select those points that minimizes the maximum variance of the predicted values on the data manifold. The proposed LapGOD algorithm is different from previous Laplacian based optimal experimental design techniques by using G-optimality criterion. Moreover, unlike (He et al. 2007; He 2010; He, Ji, and Bao 2009b) which find an approximate solution, our approach tries to directly minimize the objective function.

Acknowledgements

This work was supported by China National Key Technology R&D Program (2008BAH26B00 and 2007BAH11B06).

References

Atkinson, A.; Donev, A.; and Tobias, R. 2007. *Optimum Experimental Designs, with SAS*. Oxford University Press, USA.

Belkin, M.; Niyogi, P.; and Sindhwani, V. 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research* 7:2399–2434.

Cai, D.; He, X.; Han, J.; and Zhang, H.-J. 2006. Orthogonal laplacianfaces for face recognition. *IEEE Transactions on Image Processing* 15(11):3608–3614.

Chapelle, O.; Schölkopf, B.; and Zien, A., eds. 2006. *Semi-Supervised Learning*. Cambridge, MA: MIT Press.

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

Flaherty, P.; Jordan, M. I.; and Arkin, A. P. 2005. Robust design of biological experiments. In *NIPS 18*.

Golub, G. H., and Loan, C. F. V. 1996. *Matrix computations*. Johns Hopkins University Press, 3rd edition.

He, X.; Cai, D.; Liu, H.; and Han, J. 2005. Image clustering with tensor representation. In *Proc. ACM International Conference on Multimedia*.

He, X.; Min, W.; Cai, D.; and Zhou, K. 2007. Laplacian Optimal Design for Image Retrieval. In *Proc. 30th ACM SIGIR conference on Research and Development in Information Retrieval*.

He, X.; Cai, D.; and Min, W. 2005. Statistical and computational analysis of locality preserving projection. In *Proceedings of the* 22^{nd} *International Conference on Machine Learning*.

He, X.; Ji, M.; and Bao, H. 2009a. Graph embedding with constraints. In *Proc. 21st International Joint Conference on Artificial Intelligence*.

He, X.; Ji, M.; and Bao, H. 2009b. A unified active and semi-supervised learning framework for image compression. In *IEEE Conference on Computer Vision and Pattern Recognition*.

He, X. 2010. Laplacian regularized D-optimal design for active learning and its application to image retrieval. *IEEE Trans. on Image Processing* 19(1):254–263.

Min, W.; Lu, K.; and He, X. 2004. Locality pursuit embedding. *Pattern Recognition* 37(4):781–788.

Schölkopf, B., and Smola, A. J. 2002. *Learning with Kernels*. MIT Press.

Tong, S., and Koller, D. 2002. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research* 2:45–66.

Yu, K.; Bi, J.; and Tresp, V. 2006. Active learning via transductive experimental design. In *Proceedings of the 23th ICML*.

Zhang, L.; Chen, C.; Chen, W.; Bu, J.; Cai, D.; and He, X. 2009. Convex experimental design using manifold structure for image retrieval. In *Proceedings of the seventeen ACM international conference on Multimedia*, 45–54.