IPSJ Transactions on Mathematical Modeling and its Applications, vol.2, no.3, pp.87-96, 2009.

Output Divergence Criterion for Active Learning in Collaborative Settings

Neil Rubens, Ryota Tomioka, and Masashi Sugiyama

Department of Computer Science, Tokyo Institute of Technology
2-12-1, O-okayama, Meguro-ku, Tokyo, 152-8552, Japan

Abstract

In this paper, we address the task of active learning for linear regression models in collaborative settings. The goal of active learning is to select training points that would allow accurate prediction of test output values. We propose a new active learning criterion that is aimed at *directly* improving the accuracy of the output value estimation by analyzing the effect of the new training points on the estimates of the output values. The advantages of the proposed method are highlighted in collaborative settings – where most of the data points are missing, and the number of training data points is much smaller than the number of the parameters of the model.

1 Introduction

The goal of supervised learning is to learn a function that allows to accurately predict the output for previously unseen inputs. A function is learned from the training data, consisting of inputs and outputs from the unknown target function. A popular phrase in computer science "Garbage in, Garbage Out" summarizes well the importance of the training data in the learning process.

The training data (needed by the supervised learning) may not be available in advance and therefore may need to be acquired. A common way of acquiring the training data is by specifying the inputs (or select inputs in cases where obtaining inputs at the arbitrary location is not possible) and obtaining the corresponding outputs from the target function. However, obtaining output values often incurs a cost (in terms of money, effort, time, availability, etc.). For example, asking a user to evaluate a movie (in order to learn user's preferences) is costly in terms of the user's effort and time. The degree to which a training point allows us to approximate the function varies. For example, rating a popular item may not be useful for approximating the user's preferences since most users assign a positive rating to a popular item. On the other hand, rating an item which is representative of many other items and whose rating is uncertain may allow us to better approximate the user's preferences. The goal of active learning (AL) is to avoid acquiring the training points that are not useful. To achieve this, active learning criterion selects input points (for which the output values will be obtained) as to maximize the accuracy of the learned function. What makes the AL task challenging is that we have to predict the improvement in the accuracy of the learned function with regard to the input point before its output value is obtained, since once the output value is obtained it incurs a cost.

In traditional settings, we learn an approximation of the target function based on the training data from the target function. The number of training points is often assumed to be large enough in order to obtain an accurate approximation of the target function 9).

In some cases, there are not enough training points to obtain an accurate function approximation. In addition, obtaining training points from the target function could be costly. In *collaborative settings*, however, the training points from other functions (collaborative data) could be available at no cost 6), 1). For example, while rating a movie may require a substantial effort from a user, the same movie could have already been rated by many other users. The goal of active learning in collaborative settings is to utilize the training data of other functions, in order to improve the accuracy of the target function of interest.

There have been a number of works addressing the task of active learning in collaborative settings. Many of the works concentrate on the domain of recommender systems, due to the availability of datasets and the widespread use of collaborative methods (often referred to as collaborative filtering). Nakamura et al. 11) considered the collaborative filtering problem as an image restoration problem; where a user's preferences are restored from the items for which the user has expressed preferences. Selecting the item to be rated is then formulated as finding a restriction operator (an operator that restricts the components to those having observed values) that minimizes the expected squared error. Boutiler et al. 2) selected the item to rate based on the expected value of information with respect to the explicit probabilistic model; where the value of information is expressed as the expected improvement in the decision quality after the item is rated. In 7),4), the items to be rated are selected so that they would allow the system to assign the user to a certain stereotype.

Linear regression based methods are often used in collaborative filtering 1). However, there are few works on active learning in collaborative settings for linear regression based methods 15),1). Standard linear regression-based active learning approaches 3),10),5) can be applied to collaborative setting but are not necessarily well suited for it. These approaches tend to improve the accuracy of output estimates by selecting training points that may allow to improve the function's parameters. However, in collaborative setting, improved parameters may not necessarily translate into improved accuracy. A famous statistician Vladimir Vapnik noted that: "When solving a problem of interest, do not solve a more general problem as an intermediate step. Try to get the answer that you really need but not a more general one." The ultimate goal of active learning is to improve the accuracy of output estimates (and not necessary the accuracy of the function's parameters). Motivated by this, we propose a new active learning method that is aimed

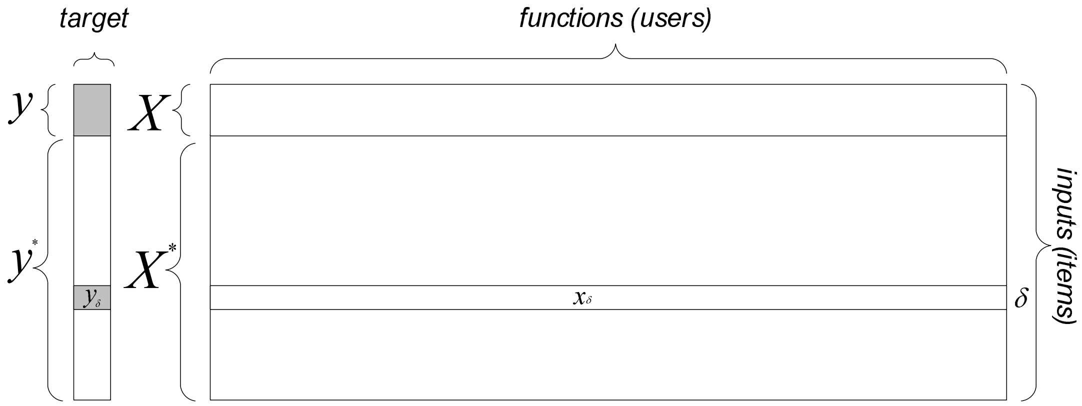

Figure 1: Problem representation. The task is to select an input δ for which the output value of the target function y_δ will be provided, as to better approximate the output values $\mathbf{y}^*$.

at *directly* improving the accuracy of the output value estimates, by analyzing the effect of the new training points on the estimates of the output values.

2 Problem Formulation

2.1 Linear Regression in Collaborative Settings

Let us formulate the task of function approximation for collaborative settings in a linear regression form (see Figure 1). A matrix entry corresponds to the output value of a function for an input i.e. $x_{i,j} = f_j(\mathbf{x}_i)$ (for recommender systems it corresponds to a rating of an item i by a user j). We want to approximate the output values $\mathbf{y}$ of the target function through the linear combination of the output values of other functions (corresponding to the column vectors of matrix $\mathbf{X}$) weighted by the parameters $\boldsymbol{\beta}$:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{1}$$

$\mathbf{X} \in \mathbb{R}^{t \times p}$, where t is the number of inputs and p is the number of functions; $\mathbf{y} \in \mathbb{R}^t$, parameters $\boldsymbol{\beta} \in \mathbb{R}^p$, and $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_t)$ normally distributed i.i.d. noise with mean zero and unknown variance σ^2. We can obtain the least squares estimator $\widehat{\boldsymbol{\beta}}$ of the parameter values as:

$$\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-\mathbf{1}} \mathbf{X}^\top \mathbf{y}, \tag{2}$$

where $^\top$ denotes the transpose. The target output values are then approximated in terms of output values of other functions.

Collaborative settings could be considered a special case of traditional settings with the following characteristics:

- matrix is sparse (most of the matrix entries are missing)

- the number of inputs and the number functions/features (dimensions of the matrix) are large
- the number of training points is much smaller than the dimensions of the matrix

Since in the collaborative settings the matrix $\mathbf{X}$ is sparse, the matrix $\mathbf{X}^\top\mathbf{X}$ is singular and is not invertible. To cope with this, we add a regularization constant $\alpha\mathbf{I}$ to $\mathbf{X}^\top\mathbf{X}$ (where the value of α is positive and is small e.g. $\alpha = 0.1$). This ensures that $\mathbf{X}^\top\mathbf{X} + \alpha\mathbf{I}$ is full rank (invertible), and improves numerical stability. Parameters $\widehat{\boldsymbol{\beta}}$ could now be expressed as:

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X} + \alpha\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}. \tag{3}$$

We can approximate the output values $\mathbf{y}^*$ of the test inputs by estimates $\widehat{\mathbf{y}}$ as:

$$\widehat{\mathbf{y}} = \mathbf{X}^*\widehat{\boldsymbol{\beta}}, \tag{4}$$

where $\mathbf{X}^*$ are the output values of the functions for the test inputs. We measure how well $\widehat{\mathbf{y}}$ approximates $\mathbf{y}^*$ by the generalization error G:

$$G(\widehat{\mathbf{y}}) = \left\|\widehat{\mathbf{y}} - \mathbf{y}^*\right\|^2. \tag{5}$$

2.2 Active Learning Task

We consider the following task. We are allowed to sequentially select for which inputs the output values (of the target function) are obtained. We want to select an input δ, so that obtaining and adding its output value y_δ to the existing output values $\mathbf{y}$ minimizes the generalization error G:

$$argmin_\delta G. \tag{6}$$

3 Related Work

An information matrix is typically used for identifying inputs, obtaining output values for which, allows us to reduce the generalization error. The inverse of the information matrix $\mathbf{A}$:

$$\mathbf{A}^{-1} = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}, \tag{7}$$

allows us to estimate the error of the approximated parameters $\widehat{\boldsymbol{\beta}}$ (obtained by Eq.(2) and used for estimating the output values by Eq.(4)). The active learning task could then be formulated as the minimization of the parameter's estimation error based on a particular optimality criterion of the information matrix.

3.1 A-optimal

The A-optimal design 3) seeks to minimize the trace of the inverse of the information matrix i.e.:

$$\min tr \mathbf{A}^{-1}. \tag{8}$$

This criterion results in minimizing the average variance of the estimates of the regression parameters $\widehat{\boldsymbol{\beta}}$.

3.2 D-optimal

The D-optimal design 10) seeks to maximize the determinant of the information matrix i.e.:

$$\max |\mathbf{A}| . \tag{9}$$

This criterion results in maximizing the differential Shannon information content of the parameter estimates.

3.3 E-optimal

The E-optimal design 5) seeks to minimize the 2-norm of the inverse of the information matrix i.e.:

$$\min \left\| \mathbf{A}^{-1} \right\|_2 . \tag{10}$$

This criterion results in maximizing the minimum eigenvalue of the information matrix.

3.4 Transductive Experimental Design

The transductive experimental design 15) seeks to maximize the trace of the information matrix with respect to test points i.e.:

$$\max tr \left(\mathbf{X}^{*} \mathbf{X}^{\top} \mathbf{A}^{-1} \mathbf{X} \mathbf{X}^{*\top} \right) . \tag{11}$$

This criterion results in finding representative training points that span a linear space for retaining most of the information of the test points.

4 Proposed Method

4.1 Motivations

The methods described in Section 3 tend to indirectly improve the estimates of output values $\widehat{\mathbf{y}}$ by improving the estimates of parameters $\widehat{\boldsymbol{\beta}}$. However, in collaborative settings, this may not necessarily be efficient for the reasons outlined bellow:

- The ultimate goal is to obtain good estimates $\widehat{\mathbf{y}}$ of the output values $\mathbf{y}$, and not necessarily good estimates $\widehat{\boldsymbol{\beta}}$ of the parameters $\boldsymbol{\beta}$.

- Traditional optimal design methods tend to assume that the bias is sufficiently small, and concentrate on minimizing the variance. However, in the current settings, the value of bias is not necessarily small, since the number of training points is much smaller than the number of parameters. So reducing the variance and ignoring the bias may not necessarily be an effective way of minimizing the generalization error.
- In the current settings, the size of $\mathbf{y}$ is smaller than the size of $\boldsymbol{\beta}$, so optimizing estimates of $\mathbf{y}$ (instead of estimates of $\boldsymbol{\beta}$) may be more computationally efficient.

4.2 Method

The generalization error measures how well the estimated output values approximate the true output values. Traditionally, active learning methods attempt to improve the generalization error indirectly by identifying training points that will improve the estimates of the parameters (used in turn for obtaining the estimated output values). However, in collaborative settings, improving the parameter estimates may not necessarily be an effective way of reducing the generalization error (as discussed in Section 4.1). We note that in the calculation of the generalization error, the true output values are not affected by the addition of the new training point, while the estimates of the output values do change. Therefore, we propose to estimate the effect of a new training point on the value of the generalization error in terms of changes in the estimates of the output values.

First, let us reformulate the goal of minimizing the generalization error in terms of the changes in its value that adding a training point causes. Let us denote the generalization error when the number of training points is equal to t by G_t. Let us denote the input of the next training point by δ; and the generalization error after the output value y_δ is obtained by G_{t+1}. Let us express G_{t+1} as:

$$G_{t+1} = G_t - (G_t - G_{t+1}).$$

The value of G_t is fixed in advance (since we are considering a sequential scenario). The value of G_{t+1} depends on the choice of δ. In order for G_{t+1} to be minimized the difference between generalization errors G_t and G_{t+1} needs to be maximized i.e.:

$$min_\delta G_{t+1} = G_t - max_\delta (G_t - G_{t+1}).$$

So the original task of minimizing the generalization error could be reformulated as maximizing the difference between the generalization errors G_t and G_{t+1} i.e.:

$$argmin_\delta G_{t+1} = argmax_\delta (G_t - G_{t+1}). \tag{12}$$

Let us denote $\widehat{\mathbf{y}}_t$ as the estimates of output values when the number of training samples is equal to t; and $\widehat{\mathbf{y}}_{t+1}$ as the estimates of output values after the value of y_δ was obtained and added to the existing ratings $\mathbf{y}$. Let us rewrite the difference between generalization

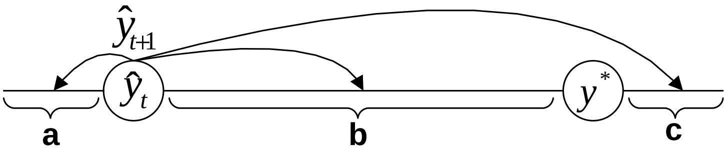

Figure 2: Location of the estimate of the output value $\widehat{y}$ after the training point δ is added to the training set (making the number of training points equal to $t+1$).

errors G_t and G_{t+1} (also referred to as $\triangle G$) in terms of a difference between $\widehat{\mathbf{y}}_t$ and $\widehat{\mathbf{y}}_{t+1}$:

$$\begin{aligned}\triangle G &= G_t - G_{t+1}\\ &= \|\widehat{\mathbf{y}}_t - \mathbf{y}^*\|^2 - \|\widehat{\mathbf{y}}_{t+1} - \mathbf{y}^*\|^2\\ &= \|\widehat{\mathbf{y}}_t\|^2 - 2\left\langle \widehat{\mathbf{y}}_t, \mathbf{y}^*\right\rangle + \|\mathbf{y}^*\|^2 - \|\widehat{\mathbf{y}}_{t+1}\|^2 + 2\left\langle \widehat{\mathbf{y}}_{t+1}, \mathbf{y}^*\right\rangle - \|\mathbf{y}^*\|^2\\ &= \|\widehat{\mathbf{y}}_t\|^2 - 2\left\langle \widehat{\mathbf{y}}_t - \widehat{\mathbf{y}}_{t+1}, \mathbf{y}^*\right\rangle - \|\widehat{\mathbf{y}}_{t+1}\|^2 .\end{aligned}$$

Defining $\boldsymbol{\epsilon} = \mathbf{y}^* - \widehat{\mathbf{y}}_{t+1}$, we have

$$\begin{aligned}&\|\widehat{\mathbf{y}}_t\|^2 - 2\left\langle \widehat{\mathbf{y}}_t - \widehat{\mathbf{y}}_{t+1}, \mathbf{y}^*\right\rangle - \|\widehat{\mathbf{y}}_{t+1}\|^2\\ &\quad= \|\widehat{\mathbf{y}}_t\|^2 - 2\left\langle \widehat{\mathbf{y}}_t, \widehat{\mathbf{y}}_{t+1}\right\rangle + \|\widehat{\mathbf{y}}_{t+1}\|^2 + 2\left\langle \widehat{\mathbf{y}}_{t+1} - \widehat{\mathbf{y}}_t, \boldsymbol{\epsilon}\right\rangle\\ &\quad= \|\widehat{\mathbf{y}}_t - \widehat{\mathbf{y}}_{t+1}\|^2 + 2\left\langle \widehat{\mathbf{y}}_{t+1} - \widehat{\mathbf{y}}_t, \boldsymbol{\epsilon}\right\rangle . \end{aligned} \tag{13}$$

Note that this decomposition is different from the standard bias-variance decomposition. Let us denote the first term of the above Eq.(13) by T_1:

$$T_1 = \|\widehat{\mathbf{y}}_t - \widehat{\mathbf{y}}_{t+1}\|^2 ,$$

and the second term by T_2:

$$T_2 = 2\left\langle \widehat{\mathbf{y}}_{t+1} - \widehat{\mathbf{y}}_t, \boldsymbol{\epsilon}\right\rangle .$$

The value of $\triangle G$ could not be calculated directly since the true output values $\mathbf{y}^*$ are not accessible. Estimating the value of term T_2 relies on the estimate of the values in $\mathbf{y}^*$, since $\boldsymbol{\epsilon} = \mathbf{y}^* - \widehat{\mathbf{y}}_{t+1}$ and $\mathbf{y}^*$ is not accessible. In the current settings, the number of training samples is small, so the estimate of $\mathbf{y}^*$ is likely to be unreliable. However, estimating the value of term T_1 requires only the estimate of a single value y^*_δ, so the estimate of T_1 is less likely to be error-prone than the estimate of T_2.

Let us investigate if T_1 alone is a good predictor of $\triangle G$. Let us consider three possible cases of the location of $\widehat{y}_{t+1}$ (an element of $\widehat{\mathbf{y}}_{t+1}$) in relation to the corresponding elements $\widehat{y}_t$ and y^*, as illustrated in Figure 2. In case (b), adding a training point improves the estimate of the true output value. In this case, maximizing T_1 also maximizes $\triangle G$. In case (a), adding a training point deteriorates the estimate of the true output value. In case (c), adding a training point causes the estimate to overshoot the true output value. In both cases (a) and (c) maximizing T_1 does not maximize $\triangle G$. In Figure 3, we show the distribution of the location of $\widehat{y}_{t+1}$ relative to $\widehat{y}_t$ and y^* (plotted from the data from

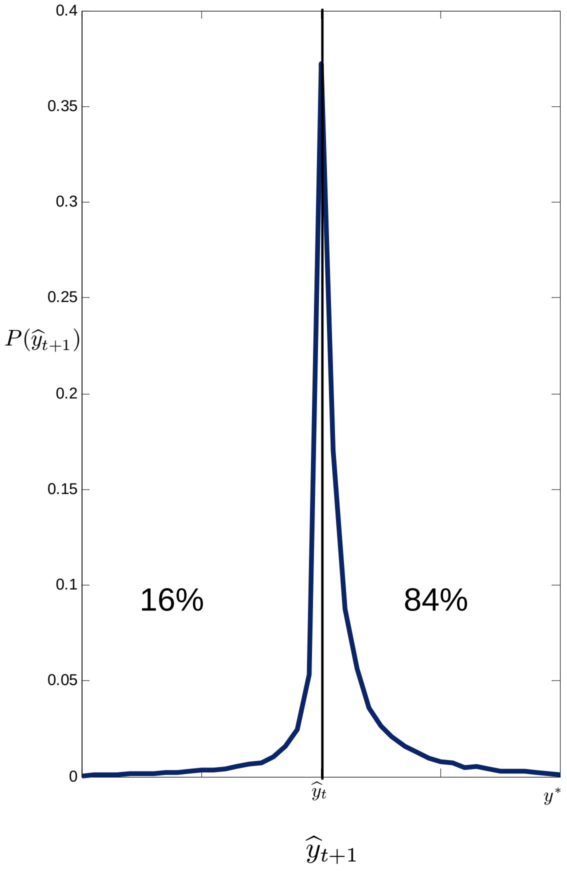

Figure 3: Distribution of $\widehat{y}_{t+1}$ in relation to y^* and $\widehat{y}_{t+1}$ (Section 5.3).

the numerical experiment described in Section 5.3). Case (b) is much more frequent than cases (a) and (c). Even when cases (a) and (c) do occur, the probability of the output estimate significantly deteriorating is low. Since T_1 is less prone to error and is more likely to be applicable, we use it as an estimator of $\triangle G$ and define the active learning criterion J as:

$$J(\delta) = \|\widehat{\mathbf{y}}_t - \widehat{\mathbf{y}}_{t+1}\|^2 . \tag{14}$$

The training point is then selected as:

$$argmax_\delta J(\delta). \tag{15}$$

4.3 Criterion Formulation in Linear Regression Settings

Let us formulate the proposed criterion for the linear regression settings (Section 2.1) as:

$$J(\delta) = \|\widehat{\mathbf{y}}_t - \widehat{\mathbf{y}}_{t+1}\|^2 = \left\|\mathbf{X}^* \left(\widehat{\boldsymbol{\beta}}_t - \widehat{\boldsymbol{\beta}}_{t+1}\right)\right\|^2 , \tag{16}$$

where the least-squares estimators $\widehat{\boldsymbol{\beta}}_t$, $\widehat{\boldsymbol{\beta}}_{t+1}$ of the parameter values are obtained by using Eq.(3). Let

$$\mathbf{A} = \mathbf{X}^\top \mathbf{X} + \alpha \mathbf{I}, \tag{17}$$

where $\alpha\mathbf{I}$ is a regularization parameter (where $0 < \alpha \ll 1$), this ensures that matrix $\mathbf{A}$ is invertible. We can rewrite the parameter estimate $\widehat{\boldsymbol{\beta}}_t$ as:

$$\widehat{\boldsymbol{\beta}}_t = \mathbf{A}^{-1}\mathbf{X}^\top \mathbf{y}. \tag{18}$$

The parameters $\widehat{\boldsymbol{\beta}}_{t+1}$, after the output value for the input δ was added could be expressed as:

$$\begin{aligned}\widehat{\boldsymbol{\beta}}_{t+1} &= (\mathbf{A} + \mathbf{x}_\delta \mathbf{x}_\delta^\top)^{-1}(\mathbf{X}^\top \mathbf{y} + \mathbf{x}_\delta y_\delta) \\ &= (\mathbf{A} + \mathbf{x}_\delta \mathbf{x}_\delta^\top)^{-1}\mathbf{X}^\top \mathbf{y} + (\mathbf{A} + \mathbf{x}_\delta \mathbf{x}_\delta^\top)^{-1}\mathbf{x}_\delta y_\delta. \end{aligned} \tag{19}$$

We can rewrite the first term of Eq.(19) by using the Woodbury formula 14) as:

$$(\mathbf{A} + \mathbf{x}_\delta \mathbf{x}_\delta^\top)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{x}_\delta \mathbf{x}_\delta^\top \mathbf{A}^{-1}}{1 + \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta}.$$

Then expand the terms of Eq.(19) as:

$$(\mathbf{A} + \mathbf{x}_\delta \mathbf{x}_\delta^\top)^{-1}\mathbf{X}^\top \mathbf{y} = \mathbf{A}^{-1}\mathbf{X}^\top \mathbf{y} - \frac{\mathbf{A}^{-1}\mathbf{x}_\delta \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{X}^\top \mathbf{y}}{1 + \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta}, \tag{20}$$

$$(\mathbf{A} + \mathbf{x}_\delta \mathbf{x}_\delta^\top)^{-1}\mathbf{x}_\delta y_\delta = \mathbf{A}^{-1}\mathbf{x}_\delta y_\delta - \frac{\mathbf{A}^{-1}\mathbf{x}_\delta \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta y_\delta}{1 + \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta}. \tag{21}$$

The difference between the parameter estimates could then be expressed as:

$$\begin{aligned}\widehat{\boldsymbol{\beta}}_{t+1} - \widehat{\boldsymbol{\beta}}_t &= \mathbf{A}^{-1}\mathbf{x}_\delta y_\delta - \frac{\mathbf{A}^{-1}\mathbf{x}_\delta \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta y_\delta}{1 + \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta} - \frac{\mathbf{A}^{-1}\mathbf{x}_\delta \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{X}^\top \mathbf{y}}{1 + \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta} \\ &= \mathbf{A}^{-1}\mathbf{x}_\delta y_\delta - \frac{1 + \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta - \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta}{1 + \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta} - \frac{\mathbf{A}^{-1}\mathbf{x}_\delta \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{X}^\top \mathbf{y}}{1 + \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta} \\ &= \frac{\mathbf{A}^{-1}\mathbf{x}_\delta (y_\delta - \mathbf{x}_\delta^\top \widehat{\boldsymbol{\beta}}_t)}{1 + \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta}.\end{aligned}$$

The difference between the output values could now be expressed as:

$$\widehat{\mathbf{y}}_{t+1} - \widehat{\mathbf{y}}_t = \mathbf{X}^* \frac{\mathbf{A}^{-1}\mathbf{x}_\delta (y_\delta - \mathbf{x}_\delta^\top \widehat{\boldsymbol{\beta}}_t)}{1 + \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta}.$$

The proposed criterion is then formulated as:

$$\begin{aligned} J(\delta) &= \|\widehat{\mathbf{y}}_{t+1} - \widehat{\mathbf{y}}_t\|^2 \\ &= \left(\frac{y_\delta - \mathbf{x}_\delta^\top \widehat{\boldsymbol{\beta}}_t}{1 + \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta} \right)^2 \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{X}^{*\top}\mathbf{X}^* \mathbf{A}^{-1}\mathbf{x}_\delta. \end{aligned} \tag{22}$$

4.4 Interpretation

Let us look at a possible interpretation of the proposed criterion (Eq.(22)) and its relation to existing active learning criterions. Let us rewrite the criterion as:

$$J(\delta) = (y_\delta - \mathbf{x}_\delta^\top \widehat{\boldsymbol{\beta}}_t)^2 \frac{\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{X}^{*\top}\mathbf{X}^{*}\mathbf{A}^{-1}\mathbf{x}_\delta}{(1+\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta)^2} = J_R \frac{J_S}{J_P}, \tag{23}$$

where

$$J_R = (y_\delta - \mathbf{x}_\delta^\top \widehat{\boldsymbol{\beta}}_t)^2, \tag{24}$$

$$J_S = \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{X}^{*\top}\mathbf{X}^{*}\mathbf{A}^{-1}\mathbf{x}_\delta, \tag{25}$$

$$J_P = (1+\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta)^2. \tag{26}$$

The term $J_R = (y_\delta - \mathbf{x}_\delta^\top \widehat{\boldsymbol{\beta}}_t)^2$ represents the residual value, i.e. the squared error between the actual output value y_δ and its estimate $\mathbf{x}_\delta^\top \widehat{\boldsymbol{\beta}}_t$. The $\mathbf{x}_\delta$ with larger residual value is then favored by the term J_R. Taking the residual value into account corresponds to the residual-based active learning methods12).

The next term J_S may be rewritten further as:

$$J_S = \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{X}^{*\top}\mathbf{X}^{*}\mathbf{A}^{-1}\mathbf{x}_\delta = \sum_{\mathbf{x}_t \in \boldsymbol{X}^*} \left(\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_t\right)^2, \tag{27}$$

where $\boldsymbol{X}^*$ denotes a set of row vectors of the matrix $\mathbf{X}^*$. By noticing that $\mathbf{x}_\delta \in \boldsymbol{X}^*$, we can further rewrite the above term as:

$$J_S = (\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta)^2 + \sum_{\mathbf{x}_t \in \boldsymbol{X}^* \setminus \mathbf{x}_\delta} \left(\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_t\right)^2. \tag{28}$$

The part $\frac{J_S}{J_P}$ of the proposed criterion could now be rewritten as:

$$\begin{aligned}\frac{J_S}{J_P} =& \frac{(\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta)^2 + \sum_{\mathbf{x}_t \in \boldsymbol{X}^* \setminus \mathbf{x}_\delta} \left(\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_t\right)^2}{(1+\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta)^2} \\ =& \frac{(\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta)^2}{(1+\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta)^2} + \frac{\sum_{\mathbf{x}_t \in \boldsymbol{X}^* \setminus \mathbf{x}_\delta} \left(\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_t\right)^2}{(1+\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta)^2} \\ =& J_O + J_T,\end{aligned} \tag{29}$$

where

$$J_O = \frac{(\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta)^2}{(1+\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta)^2}, \tag{30}$$

$$J_T = \frac{\sum_{\mathbf{x}_t \in \boldsymbol{X}^* \setminus \mathbf{x}_\delta} \left(\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_t\right)^2}{(1+\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta)^2}. \tag{31}$$

In order to interpret the meaning of the terms J_O and J_T, let us eigen-decompose the matrix $\mathbf{X}^\top\mathbf{X}$ into its eigenvalues and eigenvectors as:

$$\mathbf{X}^\top\mathbf{X} = \sum_{i=1}^{p} \lambda_i \boldsymbol{\varphi}_i \boldsymbol{\varphi}_i^\top, \tag{32}$$

where λ_i are eigenvalues such that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m > \lambda_{m+1} = \ldots = \lambda_d = 0$ and associated eigenvectors $\boldsymbol{\varphi}_i$, and m is the rank of the matrix $\mathbf{X}^\top\mathbf{X}$. We can also rewrite the matrix $\mathbf{A}$ as:

$$\begin{aligned}\mathbf{A} &= \mathbf{X}^\top\mathbf{X} + \alpha\mathbf{I} \\ &= \sum_{i=1}^{p} \lambda_i \boldsymbol{\varphi}_i \boldsymbol{\varphi}_i^\top + \alpha\mathbf{I}.\end{aligned} \tag{33}$$

Let us examine the conditions under which the value of J_O increases. Let $\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta = a$, then we can rewrite J_O as:

$$J_O = \frac{a^2}{(a+1)^2} \tag{34}$$

The value of J_O is non-negative and is monotone increasing with respect to a. So let us examine under which conditions the value of a is large. We can rewrite a as:

$$\begin{aligned}a &= \sum_{i=1}^{p} \left(\mathbf{x}_\delta^\top \boldsymbol{\varphi}_i\right)^2 \frac{1}{\lambda_i + \alpha} \\ &= \sum_{i=1}^{m} \left(\mathbf{x}_\delta^\top \boldsymbol{\varphi}_i\right)^2 \frac{1}{\lambda_i + \alpha} + \frac{1}{\alpha} \sum_{i=m+1}^{p} \left(\mathbf{x}_\delta^\top \boldsymbol{\varphi}_i\right)^2.\end{aligned} \tag{35}$$

Since α is set to a value close to zero (Eq.(17)), the $\frac{1}{\alpha}\sum_{i=m+1}^{p}\left(\mathbf{x}_\delta^\top \boldsymbol{\varphi}_i\right)^2$ part dominates in the above equation. We may then approximate a as:

$$a \approx \frac{1}{\alpha} \sum_{i=m+1}^{p} \left(\mathbf{x}_\delta^\top \boldsymbol{\varphi}_i\right)^2. \tag{36}$$

The value of $\frac{1}{\alpha}\sum_{i=m+1}^{p}\left(\mathbf{x}_\delta^\top \boldsymbol{\varphi}_i\right)^2$ is large when we choose $\mathbf{x}_\delta$ that belongs to the null space of $\mathbf{X}^\top\mathbf{X}$ spanned by $\{\boldsymbol{\varphi}_i\}_{i=m+1}^{p}$. This is equivalent to $\mathbf{x}_\delta$ being orthogonal to the *training space* (the range of $\mathbf{X}^\top\mathbf{X}$ spanned by $\{\boldsymbol{\varphi}_i\}_{i=1}^{m}$). So the J_O part of the criterion favors $\mathbf{x}_\delta$ that is ‘not close’ to the training space. This would be related to variance-based AL methods 10),3),5),13).

Let us examine the conditions under which the value of J_T increases. Let us try to simplify the formulation of the J_T. By using the fact that the denominator $(1 + \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta)^2 \geq 1$, we may obtain the lower bound of J_T as:

$$J_T \geq \sum_{\mathbf{x}_t \in \boldsymbol{X}^* \backslash \mathbf{x}_\delta} \left(\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_t\right)^2. \tag{37}$$

We can rewrite the $\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_t$ part of the above equation as:

$$\begin{aligned}\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_t &= \sum_{i=1}^{p} \left(\mathbf{x}_\delta^\top \boldsymbol{\varphi}_i\right)\left(\mathbf{x}_t^\top \boldsymbol{\varphi}_i\right)\frac{1}{\lambda_i+\alpha} \\ &= \sum_{i=1}^{m} \left(\mathbf{x}_\delta^\top \boldsymbol{\varphi}_i\right)\left(\mathbf{x}_t^\top \boldsymbol{\varphi}_i\right)\frac{1}{\lambda_i+\alpha} + \frac{1}{\alpha}\sum_{i=m+1}^{p} \left(\mathbf{x}_\delta^\top \boldsymbol{\varphi}_i\right)\left(\mathbf{x}_t^\top \boldsymbol{\varphi}_i\right). \end{aligned} \tag{38}$$

Since α is set to a value close to zero (Eq.(17)), the $\frac{1}{\alpha}\sum_{i=m+1}^{p}\left(\mathbf{x}_\delta^\top \boldsymbol{\varphi}_i\right)\left(\mathbf{x}_t^\top \boldsymbol{\varphi}_i\right)$ part dominates in the above equation. We may then approximate the above equation as:

$$\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_t \approx \frac{1}{\alpha}\sum_{i=m+1}^{p} \left(\mathbf{x}_\delta^\top \boldsymbol{\varphi}_i\right)\left(\mathbf{x}_t^\top \boldsymbol{\varphi}_i\right), \tag{39}$$

and may now approximate the lower bound of the term J_T as:

$$\begin{aligned} J_T &\geq \sum_{\mathbf{x}_t \in \boldsymbol{X}^* \setminus \mathbf{x}_\delta} \left(\mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_t\right)^2 \\ &\approx \sum_{\mathbf{x}_t \in \boldsymbol{X}^* \setminus \mathbf{x}_\delta} \left(\frac{1}{\alpha}\sum_{i=m+1}^{d} \left(\mathbf{x}_\delta^\top \boldsymbol{\varphi}_i\right)\left(\mathbf{x}_t^\top \boldsymbol{\varphi}_i\right)\right)^2. \end{aligned} \tag{40}$$

The term J_T favors $\mathbf{x}_\delta$ whose projection onto the null space of $\mathbf{X}^\top\mathbf{X}$ is 'close' to the projections of the vectors $\boldsymbol{X}^* \setminus \mathbf{x}_\delta$ onto the null space. Taking the test points into account is related to the transductive active learning method 15).

4.5 Implementation Considerations

In this section we address the details that need to be considered for the implementation of the proposed active learning criterion.

4.6 Criterion Estimation

We are not able to calculate the value of the proposed criterion directly since the output value of the sample y_δ is not known. Let us denote by $J(\delta|\, y_\delta = r)$ the value of the criterion $J(\delta)$ when $y_\delta = r$. We may then approximate the value of the criterion as:

$$J(\delta) \approx \sum_r P(y_\delta = r) J(\delta|\, y_\delta = r),$$

Since we assume no prior knowledge of $P(y_\delta = r)$, we approximate it by the non-informative uniform distribution.

4.7 Handling Missing Data

In collaborative settings, the data matrix is assumed to be very sparse, so proper handling of missing values (e.g. items that the user provided no ratings for) is important. We handle missing values in the following way. First, we calculate the mean output value of a function and use it to centralize the output values of the given function. Missing values are then assigned a value of zero, which corresponds to the mean output value of the function.

5 Numerical Experiments

5.1 Experiment Settings

Let us describe the settings that are common to the experiments. We have selected a popular collaborative dataset MovieLens 8) for the numerical experiments. The MovieLens dataset consists of approximately 1 million ratings for $3,900$ movies by $6,040$ users. We randomly select 100 users that have each rated at least 100 items. For each user, we randomly select 50 points (items) as potential training points and use the rest of the points as a test set. All of the users' output values (ratings) are withheld. For each user, training points are selected in a sequential manner by an active learning algorithm. After the training point is selected, its output value is revealed and the point is added to the training set. For the random active learning method, training points are selected following the uniform distribution. For all of the applicable active learning methods, the value of α (of the regularization parameter in Eq.(17)) is set to 0.1.

5.2 Effect of Active Learning

In this experiment, we investigate the effect of active learning (training point selection) on the generalization error. We use a random active learning algorithm to sequentially select 20 training points for each user. At each step, we record the change in the generalization error $\triangle G$. Results of the experiment are presented in Figure 4. In line with the expectations, the effect of training point selection on generalization error is large when the number of training points is small. As the number of training points increases, the effect of training point selection decreases, and flattens out after the number of training points is larger than 10. In order to better distinguish the effect of active learning on generalization error, we limit the size of the training set to 10 for the rest of the experiments.

5.3 Validity of Assumptions

In this experiment, we investigate whether the assumptions that the proposed algorithm relies upon are satisfied. As discussed in Section 4.2, the proposed criterion relies on the value of the error (of the output estimate) not increasing, and the output estimate not overshooting the true value. We use the experiment settings described in Section 5.1 and

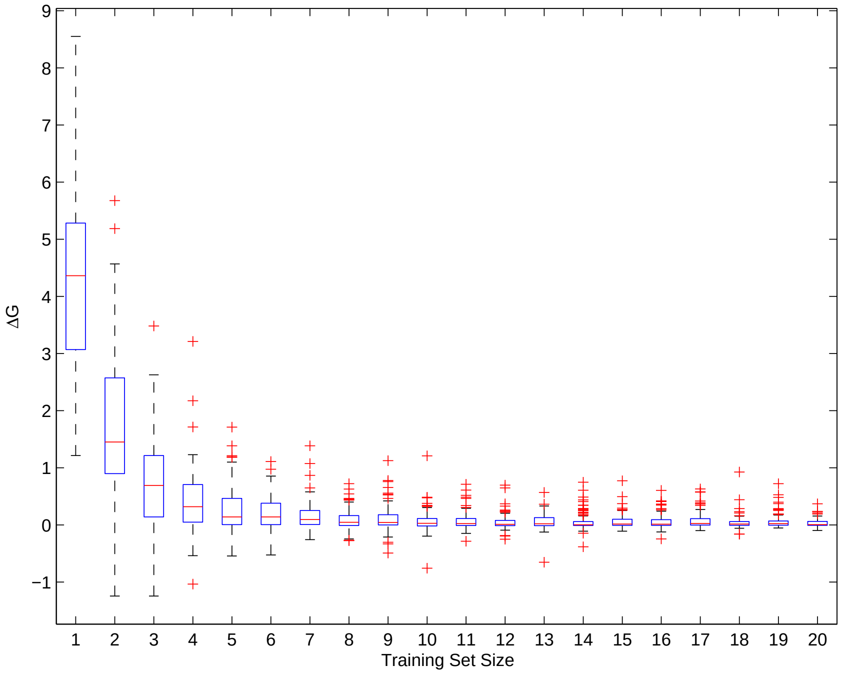

Figure 4: Effect of the training point selection (active learning) on the generalization error with respect to the training set size (Section 5.2).

the random active learning method. For each run, we record the values of $\widehat{y}_t$, $\widehat{y}_{t+1}$ and $y^*$and then plot the distribution of the $\widehat{y}_{t+1}$ normalized by $\widehat{y}_t - y^*$. From results shown in Figure 3, we can see that deterioration of the estimate and overshooting of the true value occurs with the low probability and the value of the resulting error is likely to be small. Therefore, due to only a mild violation of the assumptions, the proposed criterion is still likely to be accurate.

5.4 Criterion Accuracy Evaluation

In this experiment, we evaluate how accurately the proposed criterion estimates the change in the generalization error (caused by the addition of the new training point). We use the experiment settings described in Section 5.1 and the random active learning method. For each run, we record the actual values of the proposed criterion T_1, and the value that it estimates $\triangle G$. Results are presented in Figure 5. The term T_1 models $\triangle G$ well, except in a relatively rare situations where $\triangle G < 0$. However, for the active learning task, we are interested in the training points that improve the model i.e. $\triangle G > 0$. Therefore for the task of active learning, the proposed criterion could be considered a good predictor of $\triangle G$.

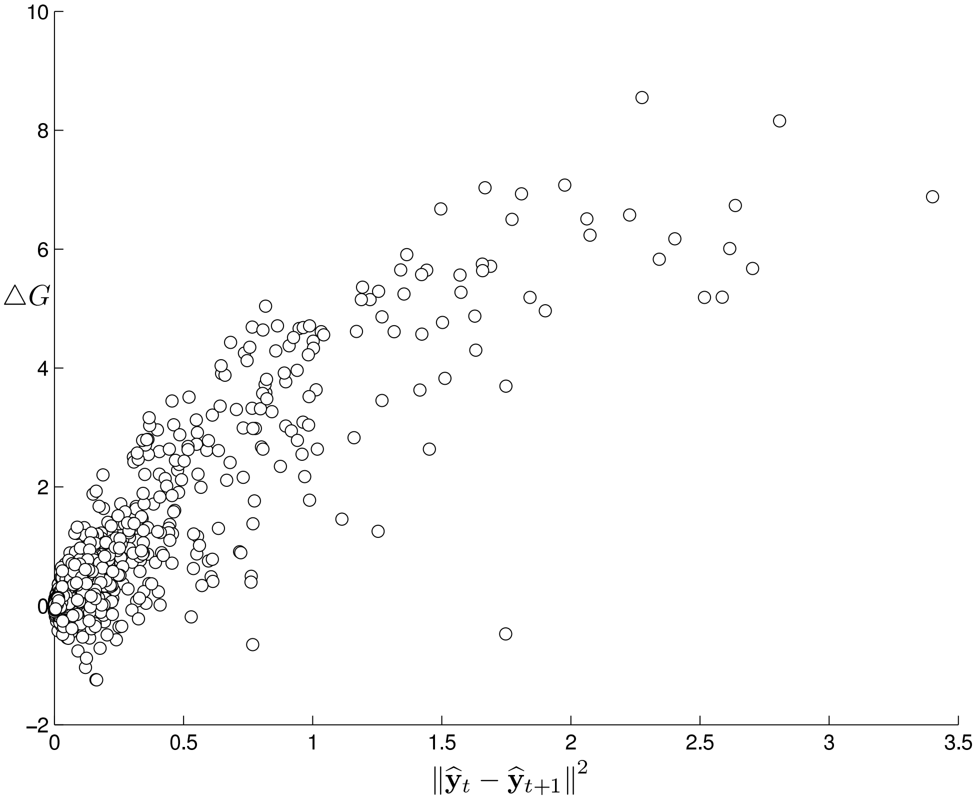

Figure 5: Relation between the value of $T_1 = \|\widehat{\mathbf{y}}_t - \widehat{\mathbf{y}}_{t+1}\|$ and the value that it tries to approximate $\triangle G$ (Section 5.4).

5.5 Comparison with existing Active Learning algorithms

In this experiment, we evaluate how the proposed method compares with existing methods (Section 3), a random active learning method, and an optimal method (that selects the best possible training sample that results in the largest possible reduction of the generalization error). In order to calculate the value of the proposed criterion, the estimate of the output value y_δ is required. We assume no prior knowledge, and estimate the expected value of the criterion by assuming a uniform distribution over the output values of the input δ (Section 4.5). Results are presented in Figure 6. Existing methods appear to perform poorly under collaborative settings, since most of the existing methods (with the exception of transductive active learning at the later stages) are outperformed by the random method. Existing algorithms do not specifically consider collaborative settings, and this appears to be detrimental to their performance. The proposed algorithm has the best performance (at the statistical significance level of 95%); hence it appears to be a promising active learning method for collaborative settings.

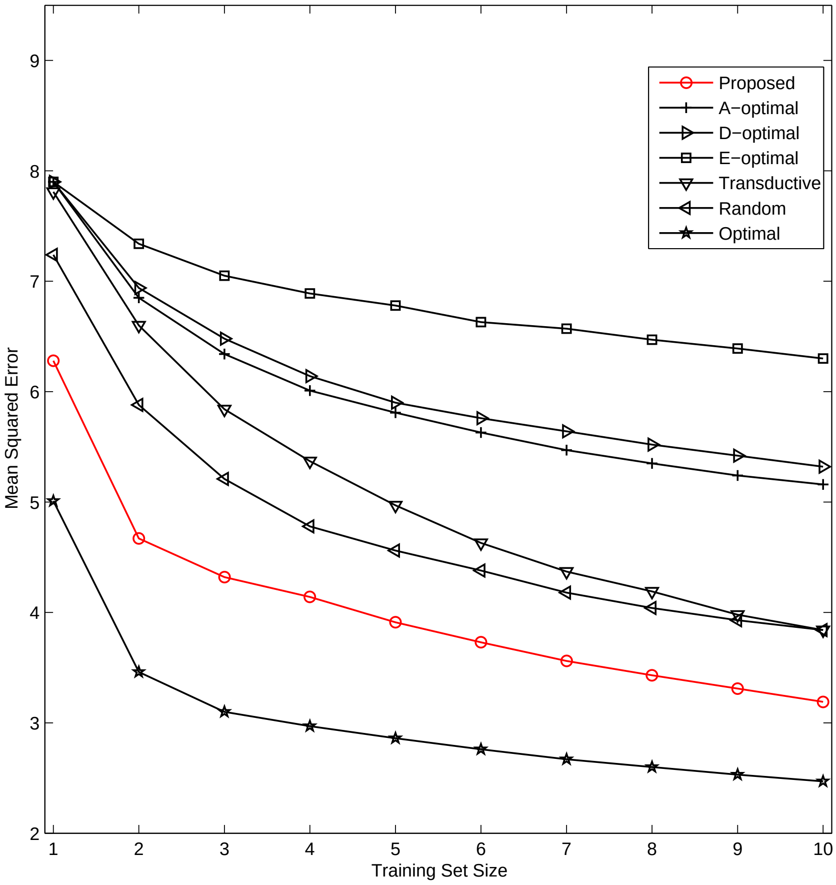

Figure 6: Evaluation of active learning criterions (Section 5.5).

6 Conclusion

We proposed an active learning criterion that aims to directly improve the estimates of the output values. To accomplish this, we select new training points to be added to the training set, that cause a large change in the estimates of output values. As a result, the proposed criterion favors training points that are not represented by the existing training set and are representative of many other points (Section 4.4). In experiments, the proposed approach performs favorably under collaborative settings in comparison with traditional approaches.

Acknowledgements

The authors acknowledge financial support from MEXT (17700142, 18300057, and 053155).

References

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[2] C. Boutilier, R. Zemel, and B. Marlin. Active collaborative filtering. In *Proceedings of the Nineteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 98–106, 2003.

[3] N. Chan. A-optimality for regression designs. Technical report, Stanford University, Department of Statistics, 1981.

[4] S. Dasgupta, W. Lee, and P. Long. A theoretical analysis of query selection for collaborative filtering, 2003.

[5] H. Dette and W. J. Studden. Geometry of e-optimality. *Annals of Statistics*, 21(1):416–43, 1993.

[6] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[7] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

[8] GroupLens, University of Minnesota. Movielens data set. http://movielens.umn.edu.

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

[10] R. C. S. John and N. R. Draper. D-optimality for regression designs: A review. *Technometrics*, 17(1):15–23, Feb. 1975.

[11] A. Nakamura, M. Kudo, and A. Tanaka. Collaborative filtering using restoration operators. In *Proceedings of Principles and Practice of Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pages 339–349. Springer, 2003.

[12] D. Romano and M. Kinnaert. An experiment-based methodology for robust design of optimal residual generators. In *IEEE Conference on Decision and Control*, 2005.

[13] M. Sugiyama and H. Ogawa. Incremental active learning for optimal generalization. *Neural Computation*, 12(12):2909–2940, 2000.

[14] M. A. Woodbury. Inverting modified matrices. Technical report, Statistical Research Group, Princeton University, 1950.

[15] K. Yu, J. Bi, and V. Tresp. Active learning via transductive experimental design. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 1081–1088, New York, NY, USA, 2006. ACM.