社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－MPS－72 (17)
2008－BIO－15 (17)
2008／12／17

Output Divergence Criterion for Active Learning in Collaborative Settings

Neil Rubens, Ryota Tomioka, Masashi Sugiayama

Department of Computer Science, Tokyo Institute of Technology

***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.**

I. Introduction

In standard supervised learning settings, we try to learn (approximate) a target function from the data consisting of inputs and the corresponding outputs from the target function. Recently, collaborative settings are becoming more common[1]. In standard settings, it is assumed that all of the data comes from the same function; whereas in *collaborative settings* there is not enough data from the target function to obtain a reliable approximation. However, in collaborative settings, the data from many other functions are available. Utilizing the data from other functions allows us to obtain a better approximation of the target function and/or to reduce the cost of data acquisition. For example, in recommender systems domain, it is common for a user to rate only a small portion of the items.

For the task of *active learning*, it is assumed that in order to better approximate the function, we can select inputs for which the output values will be obtained. For example, in order to better learn a user's preferences in the domain of recommender systems, we can ask the user to express preferences for the selected item. However, 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. Therefore, choosing samples carefully may allow us to obtain a better approximation of the true function.

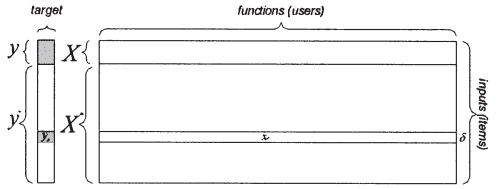

Figure 1. A matrix entry corresponds to the output value of a function for an input (for recommender systems it corresponds to a rating of an item by a user). The matrix is sparse (most of the entries are missing). The task is to select an input δ for which the output value of the target function y_δ will be provided, so as to better approximate the output values $\mathbf{y}^*$.

II. Problem Formulation

A. Linear Regression in Collaborative Settings

Let us formulate the task of function approximation for collaborative settings in a linear regression form. As illustrated in Figure 1, 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 $\boldsymbol{\beta}$: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \mathbf{X} \in \mathbb{R}^{t \times p}$, 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: $\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}$, where $^\top$ denotes the transpose. However, in the collaborative settings the matrix $\mathbf{X}$ is often sparse, so 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: $\widehat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X} + \alpha\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$. We can approximate the output values $\mathbf{y}^*$ of the test inputs by estimates $\widehat{\mathbf{y}}$ as: $\widehat{\mathbf{y}} = \mathbf{X}^*\widehat{\boldsymbol{\beta}}$, 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 $G(\widehat{\mathbf{y}}) = \|\widehat{\mathbf{y}} - \mathbf{y}^*\|^2$.

B. 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: $argmin_\delta G$.

III. 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 $\mathbf{A}^{-1} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1}$,allows us to estimate the error of the approximated parameters $\widehat{\boldsymbol{\beta}}$. 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: $\min tr\mathbf{A}^{-1}$ for the the A-optimal design [2], $\max |\mathbf{A}|$ for the D-optimal design [5], $\min \left\|\mathbf{A}^{-1}\right\|_2$ for the E-optimal design [3], $\max tr\left(\mathbf{X}^*\mathbf{X}^\top \mathbf{A}^{-1}\mathbf{X}\mathbf{X}^{*\top}\right)$ for the transductive experimental design [6].

IV. Proposed Method

A. Motivations

The methods described in Section III 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.

B. Method

The generalization error measures how well the estimated output values approximate the true output values. 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.

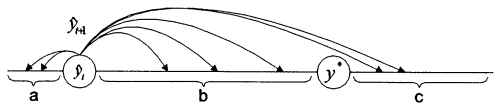

Figure 2. Location of the estimate of the output value $\widehat{y}$ after the training point δ is added to the training set (making the number of training points equal to $t+1$).

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. 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.: $argmin_\delta G_{t+1} = argmax_\delta (G_t - G_{t+1})$. 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 errors G_t and G_{t+1} (also referred to as $\triangle G$) in terms of a difference between $\widehat{\mathbf{y}}_t$ and $\widehat{\mathbf{y}}_{t+1}$: $\triangle G = \left\|\widehat{\mathbf{y}}_t\right\|^2 - 2\left\langle \widehat{\mathbf{y}}_t - \widehat{\mathbf{y}}_{t+1}, \mathbf{y}^*\right\rangle - \left\|\widehat{\mathbf{y}}_{t+1}\right\|^2$. Defining $\boldsymbol{\epsilon} = \mathbf{y}^* - \widehat{\mathbf{y}}_{t+1}$, we have:

$$\triangle G = \left\|\widehat{\mathbf{y}}_t - \widehat{\mathbf{y}}_{t+1}\right\|^2 + 2\left\langle \widehat{\mathbf{y}}_{t+1} - \widehat{\mathbf{y}}_t, \boldsymbol{\epsilon}\right\rangle . \quad (1)$$

Note that this decomposition is different from the standard bias-variance decomposition. Let us denote the first term of the above Eq. (1) by $T_1 = \left\|\widehat{\mathbf{y}}_t - \widehat{\mathbf{y}}_{t+1}\right\|^2$, and the second term by $T_2 = 2\left\langle \widehat{\mathbf{y}}_{t+1} - \widehat{\mathbf{y}}_t, \boldsymbol{\epsilon}\right\rangle$.

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

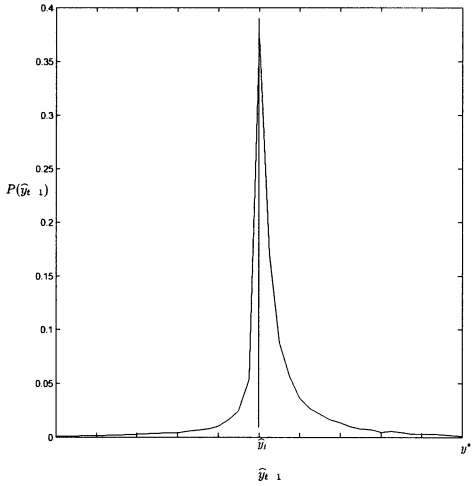

Figure 3. Distribution of $\widehat{y}_{t+1}$ in relation to y^* and $\widehat{y}_{t+1}$ (Section V-C).

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 the numerical experiment described in Section V-C). 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:

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

The training point is then selected as: $argmax_\delta J(\delta)$.

C. Criterion Formulation in Linear Regression Settings

Let us formulate the proposed criterion for the linear regression settings (Section II-A) as: $J(\delta) = \left\| \mathbf{X}^* \left(\widehat{\boldsymbol{\beta}}_t - \widehat{\boldsymbol{\beta}}_{t+1} \right) \right\|^2$. We can rewrite the parameter estimate as: $\widehat{\boldsymbol{\beta}}_t = \mathbf{A}^{-1}\mathbf{X}^\top \mathbf{y}$. The parameters $\widehat{\boldsymbol{\beta}}_{t+1}$, after the output value for the input δ was added could be expressed as: $\widehat{\boldsymbol{\beta}}_{t+1} = (\mathbf{A} + \mathbf{x}_\delta \mathbf{x}_\delta^\top)^{-1}\mathbf{X}^\top \mathbf{y} + (\mathbf{A} + \mathbf{x}_\delta \mathbf{x}_\delta^\top)^{-1}\mathbf{x}_\delta y_\delta$. By using the Woodbury formula, the difference between the parameter estimates could then be expressed as: $\widehat{\boldsymbol{\beta}}_{t+1} - \widehat{\boldsymbol{\beta}}_t = \frac{\mathbf{A}^{-1}\mathbf{x}_\delta (y_\delta - \mathbf{x}_\delta^\top \widehat{\boldsymbol{\beta}}_t)}{1 + \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta}$. The 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:

$$J(\delta) = \left(\frac{y_\delta - \mathbf{x}_\delta^\top \widehat{\boldsymbol{\beta}}_t}{1 + \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{x}_\delta} \right)^2 \mathbf{x}_\delta^\top \mathbf{A}^{-1}\mathbf{X}^{*\top} \mathbf{X}^* \mathbf{A}^{-1}\mathbf{x}_\delta. \tag{3}$$

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.

V. Numerical Experiments

A. Experiment Settings

Let us describe the settings that are common to the experiments. We have selected a popular collaborative dataset MovieLens [4] 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. We estimate each user's mean rating and use it to centralize this user's ratings. 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 is set to 0.1.

B. 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, as the number of training points increases, the effect of training point selection decreases.

C. Validity of Assumptions

In this experiment, we investigate whether the assumptions that the proposed algorithm relies upon are satisfied. As discussed in Section IV-B, 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 V-A and the random

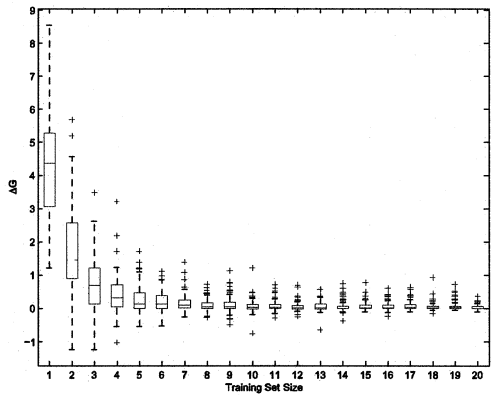

Figure 4. Effect of the training point selection (active learning) on the generalization error with respect to the training set size (Section V-B).

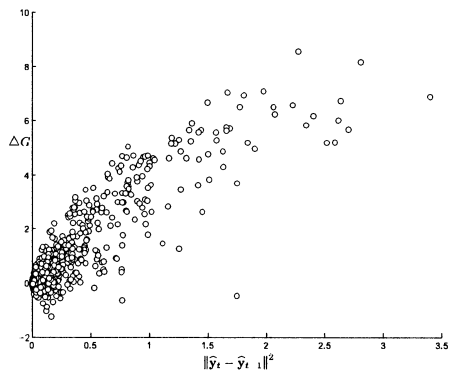

Figure 5. Relation between the value of $T_1 = \|\widehat{y}_t - \widehat{y}_{t+1}\|$ and the value that it tries to approximate $\triangle G$ (Section V-D).

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.

D. Criterion Accuracy Evaluation

In this experiment, we evaluate how accurately the proposed criterion estimates the change in the generalization error. We use the experiment settings described in Section V-A 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 $\triangle G$.

E. Comparison with existing Active Learning algorithms

In this experiment, we evaluate how the proposed method compares with existing methods (Section III), a random active learning method, and an optimal method. Results are presented in Figure 6. The proposed algorithm has the best performance (at the statistical significance level of 95%).

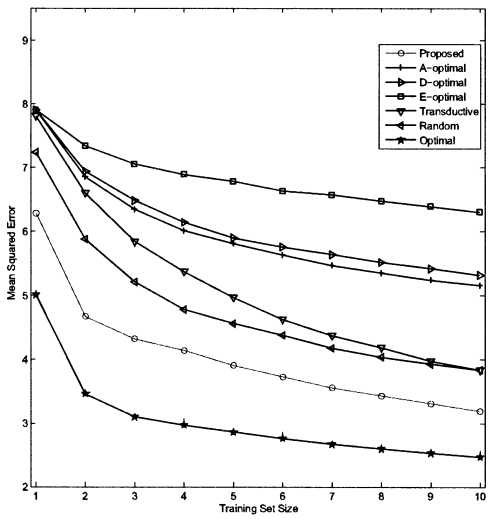

Figure 6. Evaluation of active learning criterions (Section V-E).

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

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

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

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

[6] K. Yu, J. Bi, and V. Tresp. Active learning via transductive experimental design. In *Proceedings of the 23rd Int. Conference on Machine Learning ICML '06*, pages 1081–1088, New York, NY, USA, 2006. ACM.