Information Sciences 474 (2019) 90–105

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Active learning for regression using greedy sampling

Dongrui Wu[a,*], Chin-Teng Lin[b], Jian Huang[a,*]

[a] Key Laboratory of the Ministry of Education for Image Processing and Intelligent Control, School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China
[b] Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia

ARTICLE INFO

Article history:
Received 26 June 2018
Revised 22 September 2018
Accepted 26 September 2018
Available online 26 September 2018

Keywords:
Active learning
Regression
Greedy sampling
Driver drowsiness estimation

ABSTRACT

Regression problems are pervasive in real-world applications. Generally a substantial amount of labeled samples are needed to build a regression model with good generalization ability. However, many times it is relatively easy to collect a large number of unlabeled samples, but time-consuming or expensive to label them. Active learning for regression (ALR) is a methodology to reduce the number of labeled samples, by selecting the most beneficial ones to label, instead of random selection. This paper proposes two new ALR approaches based on greedy sampling (GS). The first approach (GSy) selects new samples to increase the diversity in the output space, and the second (iGS) selects new samples to increase the diversity in both input and output spaces. Extensive experiments on 10 UCI and CMU StatLib datasets from various domains, and on 15 subjects on EEG-based driver drowsiness estimation, verified their effectiveness and robustness.

1. Introduction

Regression, which estimates the value of a dependent variable (output) from one or more independent variables (predictors, features, inputs), is a common problem in machine learning. To build an accurate regression model, one needs to have some labeled training samples, whose dependent and independent variable values are both known. Generally the more the labeled training samples are, the better the regression performance is. However, in real-world many times it is relatively easy to obtain the values of the independent variables, but time-consuming or expensive to label them. For example, in speech emotion estimation [30,31] in the 3-dimensional space of valance, arousal and dominance [15], it is easy to record a large number of utterances, but time-consuming to evaluate their emotions [2,12]. Another example is driver drowsiness estimation from physiological signals such as the electroencephalogram (EEG) [26–28]. It is relatively easy to collect a large number of EEG trials, but challenging to obtain their groundtruth drowsiness.

To overcome the small sample problem in regression, at least four different directions have been investigated:

1. *Regularization* [13,34], which introduces additional information to improve the generalization performance. For example, ridge regression (RR) [14] and LASSO [20] penalize large regression coefficients.
2. *Transfer learning* [16], which uses data or information from related domains/tasks to improve the regression performance. For example, a large number of labeled EEG data from other subjects could be used to improve the drowsiness estimation performance for a new subject, who has only a few EEG trials [24,28].

* Corresponding authors.
E-mail addresses: drwu@hust.edu.cn (D. Wu), Chin-Teng.Lin@uts.edu.au (C.-T. Lin), huang_jan@mail.hust.edu.cn (J. Huang).

https://doi.org/10.1016/j.ins.2018.09.060
0020-0255/

3. *Semi-supervised learning* [6], which simultaneously makes use of unlabeled data for training, as they also contain very useful information. For example, co-training [33] builds two diverse regressors, and the most confident predictions of each regressor on the unlabeled data are then used to iteratively construct additional labeled training data for the other regressor.
4. *Active learning* [18], which selects the most beneficial unlabeled samples to label, instead of random selection. For example, batch-mode active learning has been employed for EEG-based driver drowsiness estimation [26].

This paper focuses on the fourth direction, i.e., active learning for regression (ALR). Particularly, we consider sequential pool-based ALR [19], in which a pool of unlabeled samples is given, and the goal is to sequentially choose some to label, so that a regression model trained from them can give the most accurate estimates for the remaining unlabeled samples. Compared with the large literature on active learning for classification, there are only a few approaches for sequential pool-based ALR [3,5,32]. The main contributions of this paper to ALR are:

1. We propose two new ALR approaches, inspired by the greedy sampling (GS) approach in [32]. They are easy to understand and implement.
2. Extensive experiments on 10 UCI and CMU StatLib datasets from various domains, and on 15 subjects on EEG-based driver drowsiness estimation, verify the effectiveness and robustness of our proposed approaches.

The remainder of this paper is organized as follows: Section 2 introduces the original GS ALR approach, and proposes two new ALR algorithms. Section 3 describes the 10 UCI and CMU StatLib datasets for evaluating the effectiveness of different ALR approaches, and the corresponding experimental results. Section 4 describes the offline EEG-based driver drowsiness estimation experiment for evaluating the effectiveness of different ALR approaches, and the corresponding experimental results. Finally, Section 5 draws conclusion and points out some future research directions.

2. Greedy sampling ALR approaches

In this section we introduce an existing GS ALR approach in the literature, and propose two new ALR approaches.

2.1. Greedy sampling on the inputs (GSx)

Yu and Kim [32] proposed four passive sampling approaches for regression. Different from most ALR approaches, which generally require updating the regression model in each iteration and computing the predictions for the unlabeled samples, passive sampling selects the sample based entirely on its location in the feature space. Thus, it is independent of the regression model, and has low computational cost.

Among the fours passive sampling approaches in [32], GS achieved the best overall performance. However, the original GS approach did not explain how the first sample was selected. This subsection introduces GSx, which is essentially the same as GS, except that it also includes a strategy to select the first sample for labeling.

Assume the pool consists of N samples $\{\mathbf{x}_n\}_{n=1}^N$, initially none of which is labeled. Our goal is to select K of them to label, and then construct an accurate regression model from them to estimate the outputs for the remaining $N-K$ samples. GSx selects the first sample as the one closest to the centroid of all N samples (i.e., the one with the shortest distance to the remaining $N-1$ samples), and the remaining $K-1$ samples incrementally. The idea is to make the first selection most representative.

Without loss of generality, assume the first k samples have already been selected. For each of the remaining $N-k$ unlabeled samples $\{\mathbf{x}_n\}_{n=k+1}^N$, GSx computes first its distance to each of the k labeled samples:

$$d_{nm}^{\mathbf{x}} = ||\mathbf{x}_n - \mathbf{x}_m||, \quad m = 1, \ldots, k; n = k+1, \ldots, N \tag{1}$$

then $d_n^{\mathbf{x}}$, the shortest distance from $\mathbf{x}_n$ to all k labeled samples:

$$d_n^{\mathbf{x}} = \min_m d_{nm}^{\mathbf{x}}, \quad n = k+1, \ldots, N \tag{2}$$

and finally selects the sample with the maximum $d_n^{\mathbf{x}}$ to label.

In summary, GSx selects the first sample as the one closest to the centroid of the pool, and in each subsequent iteration a new sample located farthest away from all previously selected samples in the input space to achieve the diversity among the selected samples. Its pseudo-code is given in Algorithm 1 .

2.2. Greedy sampling on the output (GSy)

GSx achieves diversity in the input space. Our proposed GSy aims to achieve diversity in the output space.

Like GSx, in GSy initially the pool consists of N unlabeled samples and zero labeled sample. To evaluate the diversity in the output space, we need to know the outputs (labels) of all samples, either true or estimated. In other words, GSy cannot be applied before K_0 labeled samples are obtained, where K_0 is the minimum number of labeled samples required to build a regression model. In this paper we set K_0 as the number of features in the input space, and use GSx to select the first K_0 samples to label.

Algorithm 1: The GSx ALR approach, slightly modified from GS in [32] on the initialization.

Input: N unlabeled samples, $\{\mathbf{x}_n\}_{n=1}^N$;
K, the maximum number of labels to query.
Output: The regression model $f(\mathbf{x})$.
// Initialize the first selection
Set $Z = \{\mathbf{x}_n\}_{n=1}^N$, and $S = \emptyset$;
Identify $\mathbf{x}'$, the sample closest to the centroid of Z;
Move $\mathbf{x}'$ from Z to S;
Re-index the sample in S as $\mathbf{x}_1$, and the samples in Z as $\{\mathbf{x}_n\}_{n=2}^N$;
// Select $K-1$ more samples incrementally
for $k = 1, \ldots, K-1$ **do**
  for $n = k+1, \ldots, N$ **do**
    Compute $d_n^{\mathbf{x}}$ in (2);
  end
  Identify the $\mathbf{x}'$ that has the largest $d_n^{\mathbf{x}}$;
  Move $\mathbf{x}'$ from Z to S;
  Re-index the samples in S as $\{\mathbf{x}_m\}_{m=1}^{k+1}$, and the samples in Z as $\{\mathbf{x}_n\}_{n=k+2}^N$;
end
Query to label all K samples in S;
Construct the regression model $f(\mathbf{x})$ from S.

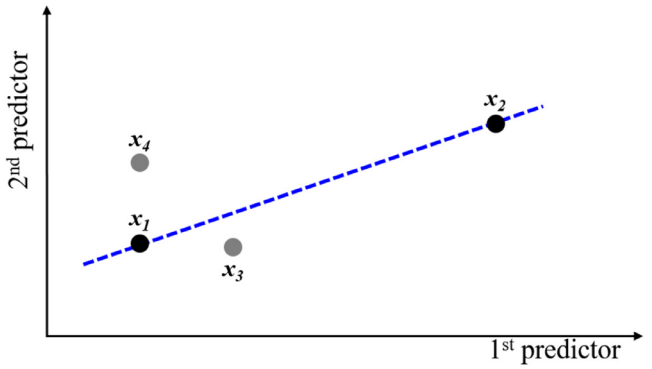

Fig. 1. Illustration of GSy. GSy will select $\mathbf{x}_3$ to label if the 1st predictor is more sensitive (important) than the 2nd, and $\mathbf{x}_4$ otherwise.

Assume the first k ($k \geq K_0$) samples have already been labeled with outputs $\{y_m\}_{m=1}^k$, and a regression model $f(\mathbf{x})$ has been constructed. For each of the remaining $N-k$ unlabeled samples $\{\mathbf{x}_n\}_{n=k+1}^N$, GSy computes first its distance to each of the k outputs:

$$d_{nm}^y = |f(\mathbf{x}_n) - y_m|, \quad m = 1, \ldots, k; n = k+1, \ldots, N \tag{3}$$

and d_n^y, the shortest distance from $f(\mathbf{x}_n)$ to $\{y_m\}_{m=1}^k$:

$$d_n^y = \min_m d_{nm}^y, \quad n = k+1, \ldots, N \tag{4}$$

and then selects the sample with the maximum d_n^y to label.

In summary, GSy selects the first a few samples using GSx to build an initial regression model, and then in each subsequent iteration a new sample located farthest away from all previously selected samples in the output space to achieve diversity among the selected samples. Its pseudo-code is given in Algorithm 2 . Note that GSy is no longer a passive sampling approach, because it needs to update $f(\mathbf{x})$ in each iteration.

The rationale for GSy can be illustrated by the following simple example shown in Fig. 1. Assume the input space has only two dimensions, and we have only four samples:

$$\mathbf{x}_1 = (x_{11}, x_{12})^T \tag{5}$$

$$\mathbf{x}_2 = (x_{21}, x_{22})^T \tag{6}$$

$$\mathbf{x}_3 = (x_{31}, x_{32})^T = (x_{11} + \delta, x_{12})^T \tag{7}$$

Algorithm 2: The GSy ALR approach.

Input: N unlabeled samples, $\{\mathbf{x}_n\}_{n=1}^N$;
K, the maximum number of labels to query.
Output: The regression model $f(\mathbf{x})$.
// Initialize the first selection
Set $Z = \{\mathbf{x}_n\}_{n=1}^N$, and $S = \emptyset$;
Identify $\mathbf{x}'$, the sample closest to the centroid of Z;
Move $\mathbf{x}'$ from Z to S;
Re-index the sample in S as $\mathbf{x}_1$, and the samples in Z as $\{\mathbf{x}_n\}_{n=2}^N$;
// Select $K_0 - 1$ more samples incrementally using GSx
Identify K_0, the minimum number of labeled samples required to construct $f(\mathbf{x})$;
for $k = 1, \ldots, K_0 - 1$ **do**
  for $n = k, \ldots, N$ **do**
    Compute $d_n^{\mathbf{x}}$ in (2);
  end
  Identify the $\mathbf{x}'$ that has the largest $d_n^{\mathbf{x}}$;
  Move $\mathbf{x}'$ from Z to S;
  Re-index the samples in S as $\{\mathbf{x}_m\}_{m=1}^{k+1}$, and the samples in Z as $\{\mathbf{x}_n\}_{n=k+2}^N$;
end
Query to label the K_0 samples in S;
Construct the regression model $f(\mathbf{x})$ from S;
// Select $K - K_0$ more samples incrementally
for $k = K_0, \ldots, K - 1$ **do**
  for $n = k, \ldots, N$ **do**
    Compute d_n^{y} in (4);
  end
  Identify the $\mathbf{x}'$ that has the largest d_n^{y};
  Move $\mathbf{x}'$ from Z to S;
  Query to label $\mathbf{x}'$ in S;
  Re-index the samples in S as $\{\mathbf{x}_m\}_{m=1}^{k+1}$, and the samples in Z as $\{\mathbf{x}_n\}_{n=k+2}^N$;
  Update the regression model $f(\mathbf{x})$ using S.
end

$$\mathbf{x}_4 = (x_{41}, x_{42})^T = (x_{11}, x_{12} + \delta)^T \tag{8}$$

where the first two have labels y_1 and y_2, respectively, the last two are unlabeled, and δ is a small number. A regression function $f(\mathbf{x})$ is built from $(\mathbf{x}_1, y_1)$ and $(\mathbf{x}_2, y_2)$. We want to select $\mathbf{x}_3$ or $\mathbf{x}_4$ to label so that the estimation error of $f(\mathbf{x})$ on the four samples can be maximally reduced.

For simplicity, assume both $f(\mathbf{x}_3)$ and $f(\mathbf{x}_4)$ are closer to y_1 than to y_2, so we only need to consider their distance to y_1 in GSy. The sensitivity of $f(\mathbf{x})$ to the first predictor, evaluated around $\mathbf{x}_1$, can be approximated as

$$s_1 = \frac{|f(\mathbf{x}_3) - y_1|}{|x_{31} - x_{11}|} = \frac{|f(\mathbf{x}_3) - y_1|}{|\delta|} \tag{9}$$

Similarly, the sensitivity of $f(\mathbf{x})$ to the second predictor, evaluated also around $\mathbf{x}_1$, can be approximated as

$$s_2 = \frac{|f(\mathbf{x}_4) - y_1|}{|x_{42} - x_{12}|} = \frac{|f(\mathbf{x}_4) - y_1|}{|\delta|} \tag{10}$$

When $|s_1| > |s_2|$, which means $f(\mathbf{x})$ is more sensitive to the first predictor than to the second, we should select a sample that can help refine the first regression coefficient for labeling. Generally the more diverse the values of the first predictor are, the more accurate its regression coefficient can be determined. So, in this case we should select $\mathbf{x}_3$ for labeling. Note that $|s_1| > |s_2|$ implies $|f(\mathbf{x}_3) - y_1| > |f(\mathbf{x}_4) - y_1|$. According to the procedure of GSy, indeed $\mathbf{x}_3$ will be selected. Similarly, when $|s_1| < |s_2|$, GSy will correctly select $\mathbf{x}_4$ for labeling.

2.3. Improved greedy sampling (iGS) on both inputs and output

GSx considers only the diversity in the input (feature) space, by computing the minimum distance between an unlabeled sample and all existing labeled samples, using all features. However, maybe not all features are useful; even if all features are useful, they may have different importance. GSx does not take feature selection/weighting into consideration.

Algorithm 3: The iGS ALR approach.

Input: N unlabeled samples, $\{\mathbf{x}_n\}_{n=1}^N$;
K, the maximum number of labels to query.
Output: The regression model $f(\mathbf{x})$.
// Initialize the first selection
Set $Z = \{\mathbf{x}_n\}_{n=1}^N$, and $S = \emptyset$;
Identify $\mathbf{x}'$, the sample closest to the centroid of Z;
Move $\mathbf{x}'$ from Z to S;
Re-index the sample in S as $\mathbf{x}_1$, and the samples in Z as $\{\mathbf{x}_n\}_{n=2}^N$;
// Select $K_0 - 1$ more samples incrementally using GSx
Identify K_0, the minimum number of labeled samples required to construct $f(\mathbf{x})$;
for $k = 1, \ldots, K_0 - 1$ **do**
  for $n = k, \ldots, N$ **do**
    Identify the $\mathbf{x}'$ that has the largest $d_n^{\mathbf{x}}$;
  end
  Move $\mathbf{x}'$ from Z to S;
  Re-index the samples in S as $\{\mathbf{x}_m\}_{m=1}^{k+1}$, and the samples in Z as $\{\mathbf{x}_n\}_{n=k+2}^N$;
end
Query to label the K_0 samples in S;
Construct the regression model $f(\mathbf{x})$ from S;
// Select $K - K_0$ more samples incrementally
for $k = K_0, \ldots, K - 1$ **do**
  for $n = k, \ldots, N$ **do**
    Compute $d_n^{\mathbf{xy}}$ in (11);
  end
  Identify the $\mathbf{x}'$ that has the largest $d_n^{\mathbf{xy}}$;
  Move $\mathbf{x}'$ from Z to S;
  Query to label $\mathbf{x}'$ in S;
  Re-index the samples in S as $\{\mathbf{x}_m\}_{m=1}^{k+1}$, and the samples in Z as $\{\mathbf{x}_n\}_{n=k+2}^N$;
  Update the regression model $f(\mathbf{x})$ using S.
end

GSy considers only the diversity in the output (label) space, by computing the minimum distance between the estimated output for a sample and all existing outputs. Our example in the previous subsection shows that GSy tries to select a new sample that can significantly increase the diversity of the most sensitive predictor. Thus, it implicitly considers feature selection/weighting. However, the predictor sensitivities are evaluated using $f(\mathbf{x})$ constructed from a very small number of labeled samples, so they may not be very accurate. In other words, feature selection/weighting in GSy may not be reliable.

In this subsection we propose an improved greedy sampling (iGS) approach, which combines GSx and GSy, to ensure that we take feature selection/weighting into consideration, but can also avoid catastrophic failure if feature selection/weighting is misleading.

Like GSx and GSy, initially the pool consists of N unlabeled samples and zero labeled sample. In iGS we again set K_0 to be the number of features in the input space, and use GSx to select the first K_0 samples to label. Assume the first k samples have already been labeled with labels $\{y_n\}_{n=1}^k$. For each of the remaining $N-k$ unlabeled samples $\{\mathbf{x}_n\}_{n=k+1}^N$, iGS computes first $d_{nm}^{\mathbf{x}}$ in (1) and d_{nm}^{y} in (3), and $d_n^{\mathbf{xy}}$:

$$d_n^{\mathbf{xy}} = \min_m d_{nm}^{\mathbf{x}} d_{nm}^{y}, \quad n = k+1, \ldots, N \tag{11}$$

and then selects the sample with the maximum $d_n^{\mathbf{xy}}$ to label. Note that we use $d_{nm}^{\mathbf{x}} d_{nm}^{y}$ instead of $d_{nm}^{\mathbf{x}} + d_{nm}^{y}$ or $(d_{nm}^{\mathbf{x}})^2 + (d_{nm}^{y})^2$ because $d_{nm}^{\mathbf{x}}$ and d_{nm}^{y} may have significantly different scales, and in the latter two formulas a term with a larger scale may dominate the other, whereas $d_{nm}^{\mathbf{x}} d_{nm}^{y}$ is not sensitive to the scales.

In summary, iGS selects the first a few samples using GSx to build an initial regression model, and then in each subsequent iteration a new sample located farthest away from all previously selected samples in both input and output spaces to achieve balanced diversity among the selected samples. Its pseudo-code is given in Algorithm 3 . Note that iGS is no longer a passive sampling approach, because it needs to update $f(\mathbf{x})$ in each iteration.

3. Experiments on UCI and CMU statlib datasets

Extensive experiments on 10 UCI and CMU StatLib datasets are performed in this section to demonstrate the performances of GSy and iGS.

Table 1
Summary of the 10 UCI and CMU StatLib datasets.

Dataset	Source	No. of samples	No. of raw features	No. of numerical features	No. of categorical features	No. of total features
Concrete-Slump[a]	UCI	103	7	7	0	7
Yacht[b]	UCI	308	6	6	0	6
autoMPG[c]	UCI	392	7	6	1	9
NO2[d]	StatLib	500	7	7	0	7
PM10[d]	StatLib	500	7	7	0	7
Housing[e]	UCI	506	13	13	0	13
CPS[f]	StatLib	534	11	8	3	19
Concrete[g]	UCI	1030	8	8	0	8
Wine-red[h]	UCI	1599	11	11	0	11
Wine-white[h]	UCI	4898	11	11	0	11

[a] https://archive.ics.uci.edu/ml/datasets/Concrete+Slump+Test.
[b] https://archive.ics.uci.edu/ml/datasets/Yacht+Hydrodynamics.
[c] https://archive.ics.uci.edu/ml/datasets/auto+mpg.
[d] http://lib.stat.cmu.edu/datasets/.
[e] https://archive.ics.uci.edu/ml/machine-learning-databases/housing/.
[f] http://lib.stat.cmu.edu/datasets/CPS_85_Wages.
[g] https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength.
[h] https://archive.ics.uci.edu/ml/datasets/Wine+Quality.

3.1. Datasets

We used 10 datasets from the UCI Machine Learning Repository[1] and the CMU StatLib Datasets Archive[2] that have been used in previous ALR experiments [4,5,23,32]. Their summary is given in Table 1. We used one-hot coding to convert categorical features into numerical features. Then, we normalized each dimension of the feature space to have mean zero and standard deviation one.

3.2. Algorithms

We compared the performances of six different sample selection algorithms:

1. Baseline (BL), which randomly selects all K samples.
2. Query-by-Committee (QBC) [17]. It first bootstraps the k labeled samples into P copies, each containing k samples but with duplicates, and builds a regression model from each copy, i.e., the committee consists of P regression models. Then, for each of the $N-k$ unlabeled samples, it computes the variance of the P individual predictions, and selects the one with the maximum variance to label.
3. Expected model change maximization (EMCM) [5]. It first uses all k labeled samples to build a linear regression model. Then, it also uses bootstrap to construct P linear regression models. For each of the $N-k$ unlabeled samples, it computes the expected model change when that sample is labeled and added to the training dataset. EMCM selects the sample with the maximum expected model change to label.
4. GSx, which has been introduced in Section 2.1.
5. GSy, which has been introduced in Section 2.2.
6. iGS, which has been introduced in Section 2.3.

All six algorithms built an RR model from the labeled samples, which minimizes the following regularized loss function:

$$l(\lambda, \boldsymbol{\beta}) = \sum_{m=1}^{K} (y_m - \boldsymbol{\beta}^T \mathbf{x}_m)^2 + \lambda |\boldsymbol{\beta}|^2 \tag{12}$$

where $\boldsymbol{\beta}$ contains the regression coefficients, and $\lambda = 0.01$ was used in our study. We used RR instead of ordinary least squares linear regression because the number of labeled samples is very small, so RR, with regularization on the regression coefficients, generally results in better generalization performance than the ordinary linear regression.

[1] http://archive.ics.uci.edu/ml/index.php.
[2] http://lib.stat.cmu.edu/datasets/.

3.3. Evaluation process

The evaluation process was similar to those used in our previous research on pool-based ALR [23,26]. For each dataset, we first randomly selected 80% of the total samples as the pool[3], initialized the first K_0 labeled samples (K_0 is the dimensionality of the input space) either randomly (for BL, QBC and EMCM) or by GSx (for GSx, GSy and iGS), identified one sample to label in each iteration by different algorithms, and built an RR model. The maximum number of samples to be labeled, K, was 20% of the dataset size. For datasets too small or too large, we constrained $K \in [20, 60]$.

To obtain statistically meaningful results, we ran this evaluation process 100 times for each dataset and each algorithm, each time with a randomly chosen 80% population pool.

3.4. Performance measures

After each iteration of each algorithm, we computed the root mean squared error (RMSE) and correlation coefficient (CC) as the performance measures.

Because different algorithms selected different samples to label, the remaining unlabeled samples in the pool were different for each algorithm, so we cannot compare their performances based on the remaining unlabeled samples. Because in pool-based ALR the goal is to build a regression model to label all samples in the pool as accurately as possible, we computed the RMSE and CC using all samples in the pool, where the labels for the K selected samples were their true labels, and the labels for the remaining $N-K$ unlabeled samples were the estimates from the regression model.

Let y_n be the true label for $\mathbf{x}_n$, and $f(\mathbf{x}_n)$ the prediction from the RR model. Without loss of generality, assume the first k samples are selected by an algorithm and hence their true labels are known. Then,

$$RMSE = \left[\frac{1}{N}\sum_{n=1}^{N}(y_n - y'_n)^2\right]^{1/2} \tag{13}$$

$$CC = \frac{\sum_{n=1}^{N}(y_n - \bar{y})(y'_n - \bar{y}')}{\sqrt{\sum_{n=1}^{N}(y_n - \bar{y})^2}\sqrt{\sum_{n=1}^{N}(y'_n - \bar{y}')^2}} \tag{14}$$

where

$$y'_n = \begin{cases} y_n, & n = 1, \ldots, k \\ f(\mathbf{x}_n), & n = k+1, \ldots, N \end{cases} \tag{15}$$

$$\bar{y} = \frac{1}{N}\sum_{n=1}^{N} y_n; \quad \bar{y}' = \frac{1}{N}\sum_{n=1}^{N} y'_n \tag{16}$$

Note that although both RMSE and CC were used as our performance measures, we should consider the RMSE as the *primary* one, because it was directly optimized in the objective function of the regression model [see (12)]. Generally as the RMSE decreases, the CC should increase, but not always. In other words, we expect that an ALR approach that gives a small RMSE should also have a large CC, but this is not always true. So, the CC can only be viewed as a *secondary* performance measure.

3.5. Experimental results

The RMSEs and CCs for the six algorithms on the 10 datasets, averaged over 100 runs, are shown in Fig. 2. Generally as K increased, all six algorithms achieved better performance (smaller RMSE and larger CC), which is intuitive, because more labeled training samples generally result in a more reliable RR model. iGS achieved the smallest RMSE and largest CC on most datasets.

To see the forest for the trees, we also define an aggregated performance measure called the area under the curve (AUC) for the average RMSE and the average CC on each of the 10 datasets in Fig. 2. The AUCs for the RMSEs are shown in Fig. 3(a), where for each dataset, we used the AUC of BL to normalize the AUCs of the other five algorithms, so the AUC of BL was always 1. For the RMSE, a smaller AUC indicates a better performance. Similarly, we also show the normalized AUCs of the CCs in Fig. 3(b), where a larger AUC indicates a better performance. Fig. 3 shows that:

1. Each of QBC, EMCM, GSx, GSy and iGS achieved smaller RMSEs than BL on at least 9 of the 10 datasets, and larger CCs than BL on at least 8 of the 10 datasets, suggesting that these five ALR approaches were all effective.
2. On average the rank of the performances of the six algorithms, from the best to the worst, was iGS $>$ GSy $>$ GSx $>$ EMCM $\approx$ QBC $>$ BL. iGS combines the advantages of GSx and GSy, and outperformed both of them.

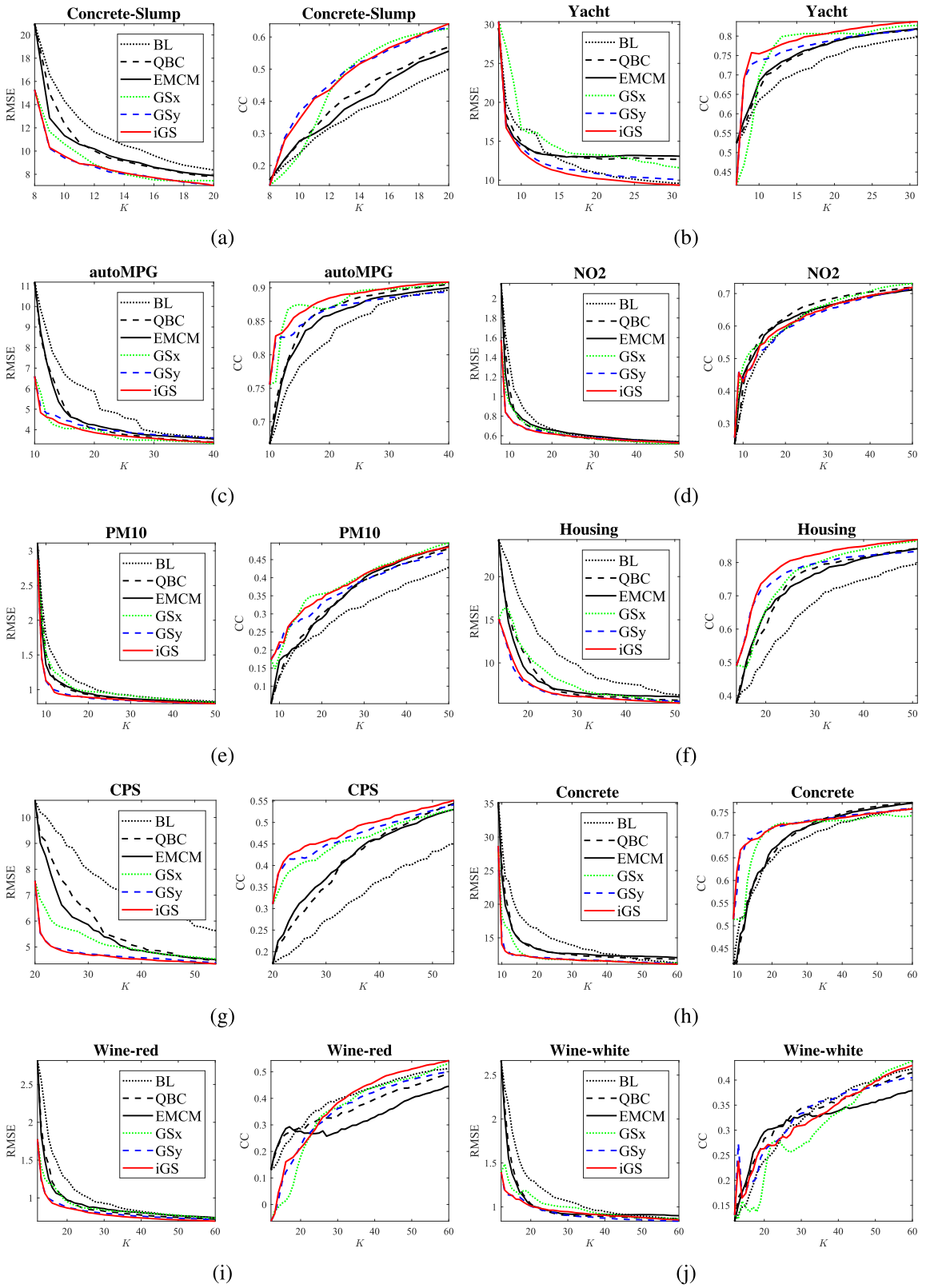

Fig. 2. Performances of the six algorithms on the 10 datasets, averaged over 100 runs. (a) Concrete-Slump; (b) Yacht; (c) autoMPG; (d) NO2; (e) PM10; (f) Housing; (g) CPS; (h) Concrete; (i) Wine-red; (j) Wine-white. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

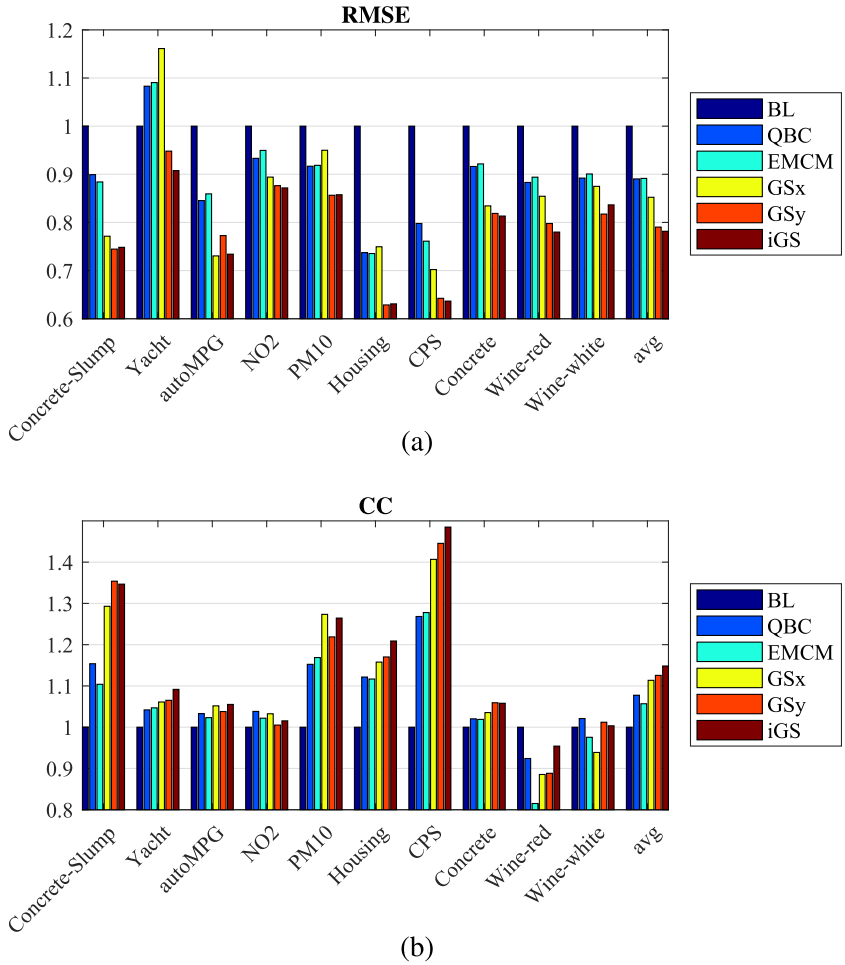

Fig. 3. Normalized AUCs of the six algorithms on the 10 datasets. (a) RMSE; (b) CC.

Table 2
Normalized RMSEs and CCs of the six approaches on the 10 datasets.

	Dataset	BL	QBC	EMCM	GSx	GSy	iGS
RMSE	Concrete-Slump	1.00	0.90	0.88	0.77	0.74	0.75
	Yacht	1.00	1.08	1.09	1.16	0.95	0.91
	autoMPG	1.00	0.85	0.86	0.73	0.77	0.73
	NO2	1.00	0.93	0.95	0.89	0.88	0.87
	PM10	1.00	0.92	0.92	0.95	0.86	0.86
	Housing	1.00	0.74	0.74	0.75	0.63	0.63
	CPS	1.00	0.80	0.76	0.70	0.64	0.64
	Concrete	1.00	0.92	0.92	0.83	0.82	0.81
	Wine-red	1.00	0.88	0.89	0.85	0.80	0.78
	Wine-white	1.00	0.89	0.90	0.87	0.82	0.84
	Average	**1.00**	**0.89**	**0.89**	**0.85**	**0.79**	**0.78**
CC	Concrete-Slump	1.00	1.15	1.10	1.29	1.35	1.35
	Yacht	1.00	1.04	1.05	1.06	1.07	1.09
	autoMPG	1.00	1.03	1.02	1.05	1.04	1.06
	NO2	1.00	1.04	1.02	1.03	1.01	1.02
	PM10	1.00	1.15	1.17	1.27	1.22	1.26
	Housing	1.00	1.12	1.12	1.16	1.17	1.21
	CPS	1.00	1.27	1.28	1.41	1.45	1.48
	Concrete	1.00	1.02	1.02	1.04	1.06	1.06
	Wine-red	1.00	0.92	0.82	0.89	0.89	0.95
	Wine-white	1.00	1.02	0.98	0.94	1.01	1.00
	Average	**1.00**	**1.08**	**1.06**	**1.11**	**1.13**	**1.15**

Table 3
p-values of non-parametric multiple comparisons on the AUCs of RMSEs and CCs on the 10 UCI and CMU StatLib datasets.

		BL	QBC	EMCM	GSx	GSy
RMSE	QBC	**.0000**				
	EMCM	**.0000**	.1663			
	GSx	**.0000**	**.0000**	**.0000**		
	GSy	**.0000**	**.0000**	**.0000**	**.0000**	
	iGS	**.0000**	**.0000**	**.0000**	**.0000**	.1045
CC	QBC	**.0000**				
	EMCM	**.0000**	**.0120**			
	GSx	**.0000**	**.0050**	**.0000**		
	GSy	**.0000**	**.0000**	**.0000**	.0896	
	iGS	**.0000**	**.0000**	**.0000**	**.0001**	**.0076**

These observations were also confirmed by Table 2, which shows the detailed ranks of the six approaches on the 10 datasets, according to the AUCs.

3.6. Statistical analysis

To determine if the differences between different algorithms were statistically significant, we also performed non-parametric multiple comparison tests on the AUCs using Dunn's procedure [10,11], with a *p*-value correction using the False Discovery Rate method [1]. The *p*-values for the AUCs of RMSEs and CCs are shown in Table 3, where the statistically significant ones are marked in bold. Table 3 shows that:

1. All five ALR approaches had statistically significantly better RMSEs and CCs than BL, suggesting again that they were effective.
2. Among the three existing ALR approaches, GSx had statistically significantly better RMSE and CC than both QBC and EMCM.
3. GSy had statistically significantly better RMSE and CC than QBC and EMCM, and also statistically significantly better RMSE than GSx.
4. iGS had statistically significantly better RMSE than all other approaches except GSy, and also statistically significantly better CC than all other approaches.

These observations verified the effectiveness of the two proposed approaches, particularly iGS.

4. Experiments on EEG-Based driver drowsiness estimation

Experiments on offline EEG-based driver drowsiness estimation are performed in this section to further demonstrate the performances of GSy and iGS.

4.1. Experiment setup

The experiment and data used in [26] were again used in this paper. Sixteen healthy subjects with normal/corrected-to-normal vision participated in a sustained-attention driving experiment [7,8], consisting of a real vehicle mounted on a motion platform with 6 degrees of freedom immersed in a 360° virtual-reality scene, simulating monotonous driving at 100 km/h on a straight and empty highway. Each experiment was conducted for about 60–90 min in the afternoon when the circadian rhythm of sleepiness reached its peak. Random lane-departure disturbances were applied every 5–10 s, and participants needed to steer the vehicle to compensate for them as quickly as possible. The response time was recorded and later converted to a drowsiness index. Participants' scalp EEG signals were also recorded using a 500Hz 32-channel Neuroscan system (30-channel EEGs plus 2-channel earlobes).

The Institutional Review Board of the Taipei Veterans General Hospital approved the experimental protocol.

4.2. Preprocessing and feature extraction

The preprocessing and feature extraction procedures were almost identical to those in our recent research [26].

The 16 subjects had different lengths of experiment, because the disturbances were presented randomly every 5–10 s. To ensure a fair comparison, we used only the first 3600 s data for each subject. Data from one subject was not recorded correctly, so we used only 15 subjects.

[3] For a fixed pool, GSx gives a deterministic selection sequence because it does not involve randomness. So, we need to vary the pool in order to study its statistical property.

We defined a function [24] to map the response time τ to a drowsiness index $y \in [0, 1]$:

$$y = \max\left\{0, \frac{1 - e^{-(\tau-\tau_0)}}{1 + e^{-(\tau-\tau_0)}}\right\} \tag{17}$$

$\tau_0 = 1$ was used in this paper. The drowsiness indices were then smoothed using a 90-second square moving-average window to reduce variations.

EEGLAB [9] was used for EEG signal preprocessing. After 1–50 Hz band-pass filtering, the EEG data were downsampled from 500 Hz to 250 Hz and re-referenced to averaged earlobes.

Our goal was to predict the drowsiness index for each subject every 10 s (called a sampling point in this paper). All 30 EEG channels were used in feature extraction. We epoched 30-second EEG signals right before each sampling point, and computed the average power spectral density (PSD) in the theta band (4–7.5 Hz) for each channel using Welch's method [21]. Next, we converted the 30 theta band powers to dBs. To remove noise or bad channel readings, we removed channels whose maximum dBs were larger than 20. We then normalized the dBs of each remaining channel to mean zero and standard deviation one, and extracted 10 leading principal components. The projections of the dBs onto these principal components were then normalized to [0, 1] and used as our features.

4.3. Experimental results

The six algorithms introduced in Section 3.2 were again compared in this experiment. The evaluation process was the same as that in Section 3.3.

The RMSEs and CCs for the six algorithms on the first 10 subjects, averaged over 100 runs, are shown in Fig. 4. The average performances across the 15 subjects are shown in Fig. 5. Generally as K increased, all six algorithms achieved better performance (smaller RMSE and larger CC), which is intuitive, because more labeled training samples generally result in a more reliable RR model. `iGS` achieved the smallest RMSE and largest CC for most subjects.

The AUCs for the RMSEs are shown in Fig. 6(a). Again, for each subject, we used the AUC of BL to normalize the AUCs of the other five algorithms, so the AUC of BL was always 1. As before, a smaller AUC for RMSE indicates a better performance, and a larger AUC for CC indicates a better performance. Fig. 6 shows that:

1. Each of the five ALR approaches achieved smaller RMSEs than BL on all 15 subjects, and larger CCs than BL on at least 14 subjects, suggesting that they were all effective.
2. On average the rank of the performances of the six algorithms, from the best to the worst, was `iGS` $>$ `GSy` $\approx$ `GSx` $>$ `EMCM` $>$ `QBC` $>$ BL. `iGS` combines the advantages of `GSx` and `GSy`, and outperformed both of them.

These observations were also confirmed by Table 4, which shows the detailed ranks of the six approaches on the 15 subjects, according to the AUCs.

4.4. Statistical analysis

To determine if the differences between different algorithms were statistically significant, we also performed non-parametric multiple comparison tests on the AUCs using the procedure described in Section 3.6. The p-values for the AUCs of RMSEs and CCs are shown in Table 5, where the statistically significant ones are marked in bold. Table 5 shows that:

1. All five ALR approaches had statistically significantly better RMSEs and CCs than BL, suggesting that they were effective.
2. Among the three existing ALR approaches, `GSx` had statistically significantly better RMSEs and CCs than `QBC` and `EMCM`.
3. `GSy` had statistically significantly better RMSE than `QBC`, `EMCM` and `EMCM`, and statistically significantly better CC than `QBC` and `EMCM`.
4. `iGS` had statistically significantly better RMSE and CC than all other approaches.

All these observations were generally consistent with our observations on the 10 UCI and CMU StatLib datasets, demonstrating the effectiveness and robustness of our proposed approaches. Particularly, on average `iGS` achieved the best performance among the six.

4.5. Discussions

ALR is usually used for the scenario that the number of labeled samples is very small. When the number of labeled samples is small, e.g., much smaller than the number of features, the baseline regression model (e.g., ridge regression in this paper) will have very high variance. A practical solution to this problem is to reduce the number of features to match the number of labeled samples, so that the baseline regression model is more stable. A commonly employed dimensionality reduction method is PCA, which has been used in this section: we used 10 PCA features instead of 30 PSD features in driver drowsiness estimation.

However, the performances of our proposed `GSy` and `iGS` are also superior to existing ALR approaches when the feature dimensionality is high. To demonstrate this, we performed another experiment using the 30 PSD features, and increased

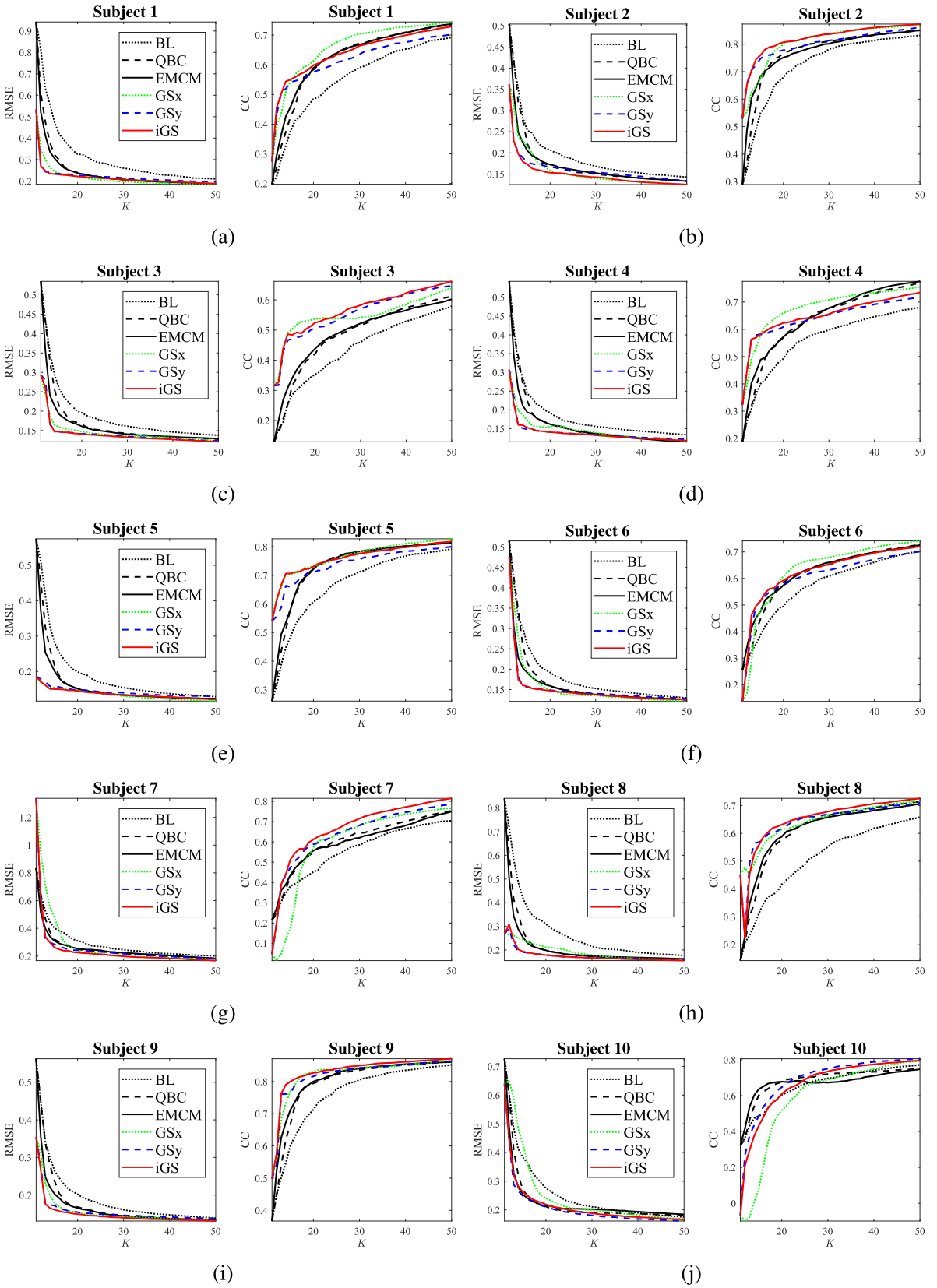

Fig. 4. Performances of the six algorithms on the first 10 subjects, averaged over 100 runs.

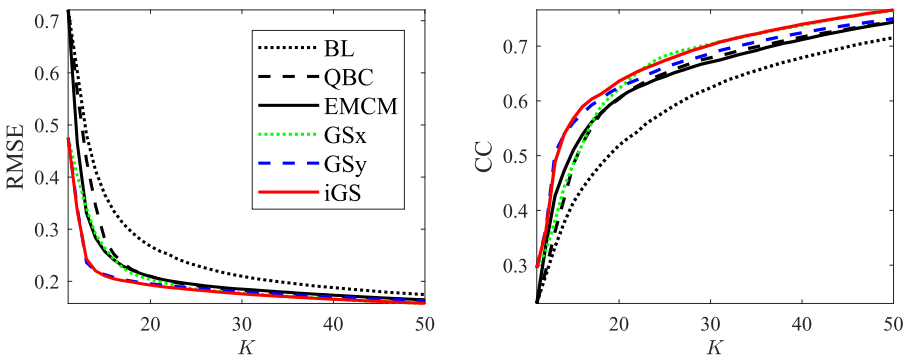

Fig. 5. Average performances of the six algorithms across the 15 subjects.

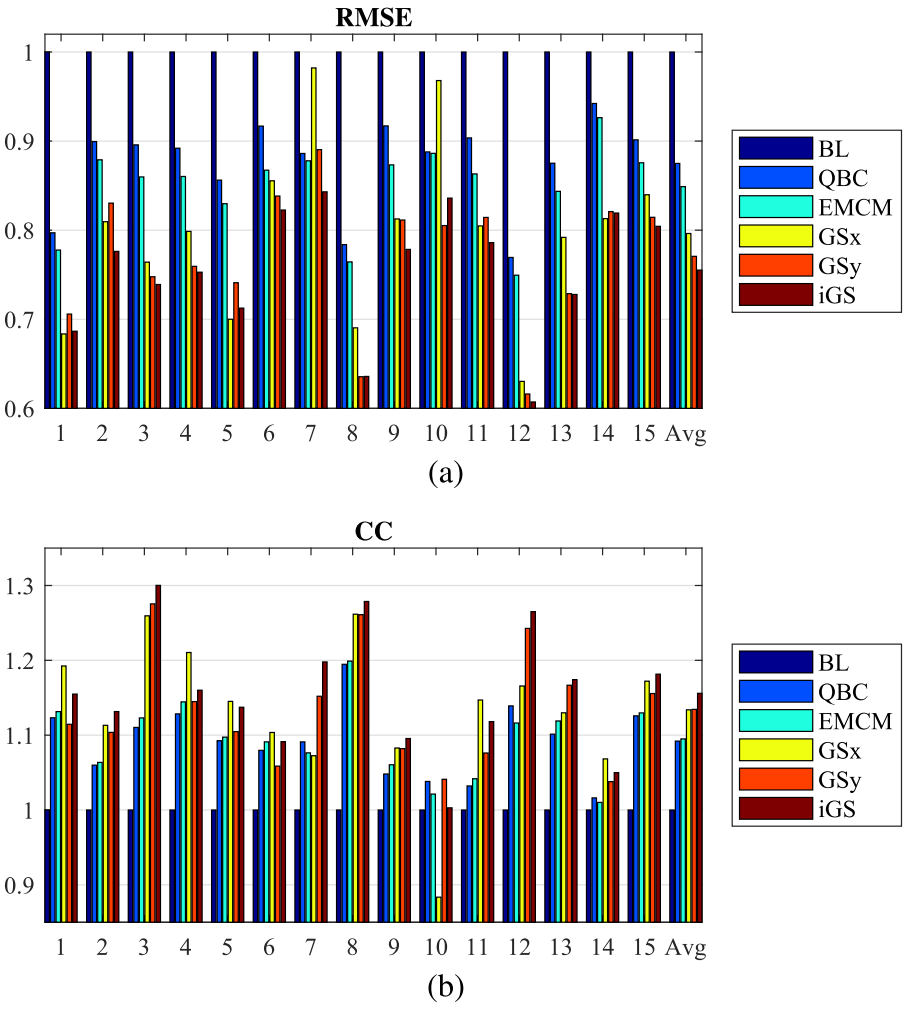

Fig. 6. Normalized AUCs of the six algorithms on the 15 subjects. (a) RMSE; (b) CC.

the number of labeled samples from 30 to 60. To save space, we only show the average performances of the six algorithms across the 15 subjects in Fig. 7. Clearly, our proposed `GSy` and `iGS` still ranked in the top two.

Recall that in (11) we use $d^{\mathbf{x}}_{nm}d^{y}_{nm}$ instead of $d^{\mathbf{x}}_{nm}+d^{y}_{nm}$ because $d^{\mathbf{x}}_{nm}$ and d^{y}_{nm} may have significantly different scales. To validate that $d^{\mathbf{x}}_{nm}d^{y}_{nm}$ may be better than $d^{\mathbf{x}}_{nm}+d^{y}_{nm}$, we performed another experiment, in which (18) was used to replace (11) in `iGS`, and the resulting algorithms is denoted as `iGS-sum`:

$$d^{\mathbf{x}\mathbf{y}}_{n}=\min_{m}\left(d^{\mathbf{x}}_{nm}+d^{y}_{nm}\right),\quad n=k+1,\ldots,N \tag{18}$$

Table 4
Normalized RMSEs and CCs of the six approaches on the 15 subjects.

	Subject	BL	QBC	EMCM	GSx	GSy	iGS
RMSE	1	1.00	0.80	0.78	0.68	0.71	0.69
	2	1.00	0.90	0.88	0.81	0.83	0.78
	3	1.00	0.90	0.86	0.76	0.75	0.74
	4	1.00	0.89	0.86	0.80	0.76	0.75
	5	1.00	0.86	0.83	0.70	0.74	0.71
	6	1.00	0.92	0.87	0.86	0.84	0.82
	7	1.00	0.89	0.88	0.98	0.89	0.84
	8	1.00	0.78	0.76	0.69	0.64	0.64
	9	1.00	0.92	0.87	0.81	0.81	0.78
	10	1.00	0.89	0.89	0.97	0.81	0.84
	11	1.00	0.90	0.86	0.80	0.81	0.79
	12	1.00	0.77	0.75	0.63	0.62	0.61
	13	1.00	0.88	0.84	0.79	0.73	0.73
	14	1.00	0.94	0.93	0.81	0.82	0.82
	15	1.00	0.90	0.88	0.84	0.81	0.80
	Average	**1.00**	**0.87**	**0.85**	**0.80**	**0.77**	**0.76**
CC	1	1.00	1.12	1.13	1.19	1.11	1.15
	2	1.00	1.06	1.06	1.11	1.10	1.13
	3	1.00	1.11	1.12	1.26	1.28	1.30
	4	1.00	1.13	1.14	1.21	1.14	1.16
	5	1.00	1.09	1.10	1.15	1.10	1.14
	6	1.00	1.08	1.09	1.10	1.06	1.09
	7	1.00	1.09	1.08	1.07	1.15	1.20
	8	1.00	1.19	1.20	1.26	1.26	1.28
	9	1.00	1.05	1.06	1.08	1.08	1.10
	10	1.00	1.04	1.02	0.88	1.04	1.00
	11	1.00	1.03	1.04	1.15	1.08	1.12
	12	1.00	1.14	1.12	1.17	1.24	1.27
	13	1.00	1.10	1.12	1.13	1.17	1.17
	14	1.00	1.02	1.01	1.07	1.04	1.05
	15	1.00	1.13	1.13	1.17	1.16	1.18
	Average	**1.00**	**1.09**	**1.09**	**1.13**	**1.13**	**1.16**

Table 5
p-values of non-parametric multiple comparisons on the AUCs of RMSEs and CCs on EEG-based driver drowsiness estimation.

		BL	QBC	EMCM	GSx	GSy
RMSE	QBC	**.0000**				
	EMCM	**.0000**	**.0000**			
	GSx	**.0000**	**.0000**	**.0000**		
	GSy	**.0000**	**.0000**	**.0000**	**.0002**	
	iGS	**.0000**	**.0000**	**.0000**	**.0000**	**.0009**
CC	QBC	**.0000**				
	EMCM	**.0000**	.2425			
	GSx	**.0000**	**.0000**	**.0000**		
	GSy	**.0000**	**.0080**	**.0000**	.1056	
	iGS	**.0000**	**.0000**	**.0000**	**.0024**	**.0000**

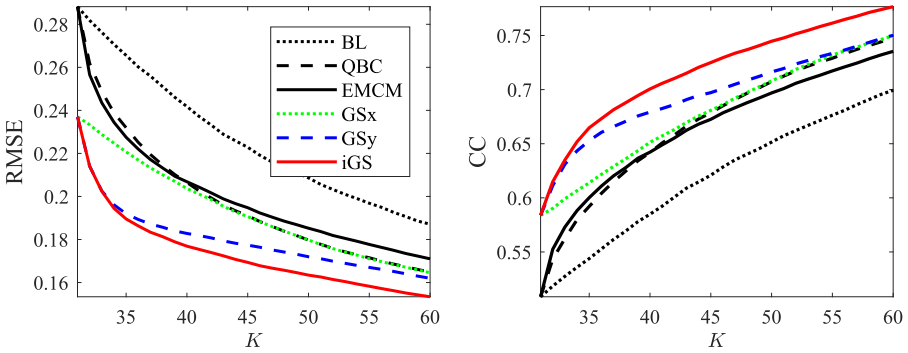

Fig. 7. Average performances of the six algorithms across the 15 subjects, when the 30 PSD features are used.

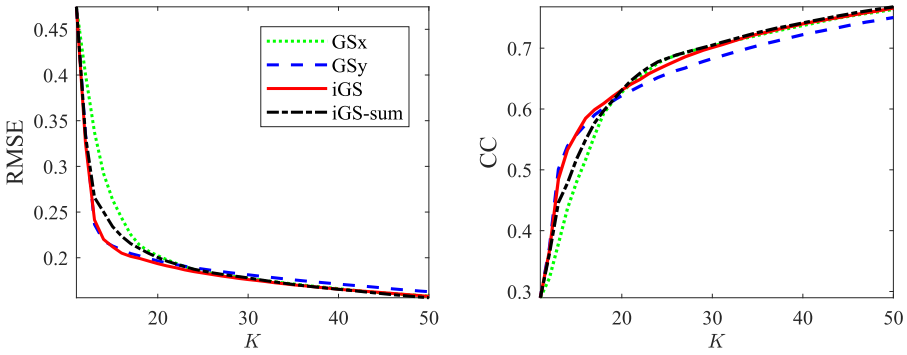

Fig. 8. Average performances of `GSx`, `GSy`, `iGS` and `iGS-sum` across the 15 subjects, when the 10 PCA features are used.

The average performances of `GSx`, `GSy`, `iGS` and `iGS-sum` across the 15 subjects are shown in Fig. 8. When K was small, the performance of `iGS-sum` was worse than `GSy`. However, `iGS` outperformed `GSx`, `GSy` and `iGS-sum`, especially when K was small, suggesting that it is better to use product than summation in (11).

5. Conclusion and future research

Usually a substantial amount of labeled training samples are needed to build an accurate regression model with good generalization ability. However, many times in real-world applications we can collect a large number of unlabeled samples, but labeling them is time-consuming or expensive. ALR is a methodology to select the most beneficial unlabeled samples to label, so that a better regression model can be built from a small number of labeled samples. This paper has proposed two new ALR approaches, inspired by a GS approach in the literature. Extensive experiments on 10 UCI and CMU StatLib datasets, and on 15 subjects on EEG-based driver drowsiness estimation, verified their effectiveness and robustness. Particularly, our proposed `iGS`, which considers diversity in both input and output spaces, outperformed several existing ALR approaches.

Our future research will extend `GSx`, `GSy` and `iGS` from regression to classification. Additionally, as described in the Introduction, regularization, transfer learning, and active learning can all be used to cope with regression problems that do not have enough labeled training data. In this paper we have used regularization and active learning together. In the future we will also study how to integrate transfer learning and active learning for regression problems. Our previous research has integrated transfer learning and active learning for classification problems and achieved promising performance [22,25,29].

Acknowledgment

This research was supported by the National Natural Science Foundation of China (61873321).

References

[1] Y. Benjamini, Y. Hochberg, Controlling the false discovery rate: a practical and powerful approach to multiple testing, J. R. Stat. Soc. Ser. B (Methodol.) 57 (1995) 289–300.
[2] M.M. Bradley, P.J. Lang, The International Affective Digitized Sounds (2nd Edition; IADS-2): Affective Ratings of Sounds and Instruction Manual, Tech. rep., b-3, University of Florida, Gainesville, FL, 2007.
[3] R. Burbidge, J.J. Rowland, R.D. King, Active learning for regression based on query by committee, Lect. Notes Comput. Sci. 4881 (2007) 209–218.
[4] W. Cai, M. Zhang, Y. Zhang, Batch mode active learning for regression with expected model change, IEEE Trans. Neural Netw. Learn.Syst. 28 (7) (2017) 1668–1681.
[5] W. Cai, Y. Zhang, J. Zhou, Maximizing expected model change for active learning in regression, in: Proc. IEEE 13th Int'l. Conf. on Data Mining, Dallas, TX, 2013.
[6] O. Chapelle, B. Scholkopf, A. Zien (Eds.), Semi-Supervised Learning, The MIT Press, 2006.
[7] C.H. Chuang, L.W. Ko, T.P. Jung, C.T. Lin, Kinesthesia in a sustained-attention driving task, Neuroimage 91 (2014) 187–202.
[8] S.W. Chuang, L.W. Ko, Y.P. Lin, R.S. Huang, T.P. Jung, C.T. Lin, Co-modulatory spectral changes in independent brain processes are correlated with task performance, Neuroimage 62 (2012) 1469–1477.
[9] A. Delorme, S. Makeig, EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis, J. Neurosci. Methods 134 (2004) 9–21.
[10] O. Dunn, Multiple comparisons among means, J. Am. Stat. Assoc. 56 (1961) 62–64.
[11] O. Dunn, Multiple comparisons using rank sums, Technometrics 6 (1964) 214–252.
[12] M. Grimm, K. Kroschel, S.S. Narayanan, The Vera Am Mittag German audio-visual emotional speech database, in: Proc. Int'l Conf. on Multimedia & Expo (ICME), Hannover, German, 2008, pp. 865–868.
[13] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer, 2009.
[14] A.E. Hoerl, R.W. Kennard, Ridge regression: biased estimation for nonorthogonal problems, Technometrics 12 (1) (1970) 55–67.
[15] A. Mehrabian, Basic Dimensions for a General Psychological Theory: Implications for Personality, Social, Environmental, and Developmental Studies, Oelgeschlager, Gunn & Hain, 1980.
[16] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.
[17] T. RayChaudhuri, L. Hamey, Minimisation of data collection by active learning, in: Proc. IEEE Int'l. Conf. on Neural Networks, Perth, Australia, 3, 1995, pp. 1338–1341.
[18] B. Settles, Active Learning Literature Survey, University of Wisconsin–Madison, 2009 Computer sciences technical report, 1648.

[19] M. Sugiyama, S. Nakajima, Pool-based active learning in approximate linear regression, Mach. Learn. 75 (3) (2009) 249–274.
[20] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. 58 (1) (1996) 267–288.
[21] P. Welch, The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms, IEEE Trans. Audio Electroacoust. 15 (1967) 70–73.
[22] D. Wu, Active semi-supervised transfer learning (ASTL) for offline BCI calibration, in: Proc. IEEE Int'l. Conf. on Systems, Man and Cybernetics, Banff, Canada, 2017.
[23] D. Wu, Pool-based sequential active learning for regression, IEEE Trans. Neural Netw. Learn.Syst. (2018). In press.
[24] D. Wu, C.H. Chuang, C.T. Lin, Online driver's drowsiness estimation using domain adaptation with model fusion, in: Proc. Int'l Conf. on Affective Computing and Intelligent Interaction, Xi'an, China, 2015, pp. 904–910.
[25] D. Wu, B.J. Lance, T.D. Parsons, Collaborative filtering for brain-computer interaction using transfer learning and active class selection, PLoS ONE (2013).
[26] D. Wu, V.J. Lawhern, S. Gordon, B.J. Lance, C.T. Lin, Offline EEG-Based Driver Drowsiness Estimation Using Enhanced Batch-mode Active Learning (EBMAL) for Regression, in: Proc. IEEE Int'l Conf. on Systems, Man and Cybernetics, Budapest, Hungary, 2016, pp. 730–736.
[27] D. Wu, V.J. Lawhern, S. Gordon, B.J. Lance, C.T. Lin, Spectral Meta-learner for Regression (SMLR) Model Aggregation: Towards Calibrationless Brain–computer Interface (BCI), in: Proc. IEEE Int'l Conf. on Systems, Man and Cybernetics, Budapest, Hungary, 2016, pp. 743–749.
[28] D. Wu, V.J. Lawhern, S. Gordon, B.J. Lance, C.T. Lin, Driver drowsiness estimation from EEG signals using online weighted adaptation regularization for regression (owARR), IEEE Trans. Fuzzy Syst 25 (6) (2017) 1522–1535.
[29] D. Wu, V.J. Lawhern, W.D. Hairston, B.J. Lance, Switching EEG headsets made easy: reducing offline calibration effort using active weighted adaptation regularization, IEEE Trans. Neural Syst. Rehabil.Eng. 24 (11) (2016) 1125–1137.
[30] D. Wu, T.D. Parsons, E. Mower, S.S. Narayanan, Speech emotion estimation in 3D space, in: Proc. IEEE Int'l Conf. on Multimedia & Expo (ICME), Singapore, 2010, pp. 737–742.
[31] D. Wu, T.D. Parsons, S.S. Narayanan, Acoustic feature analysis in speech emotion primitives estimation, in: Proc. InterSpeech, Makuhari, Japan, 2010.
[32] H. Yu, S. Kim, Passive sampling for regression, in: IEEE Int'l. Conf. on Data Mining, Sydney, Australia, 2010, ISSN 1550-4786, pp. 1151–1156.
[33] Z.H. Zhou, M. Li, Semi-supervised regression with co-training, in: Proc. 19th Int'l Joint Conf. on Artificial Intelligence, 2005. Edinburgh, Scotland, pp. 908–913
[34] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc. 67 (2) (2005) 301–320.