Hindawi
Journal of Electrical and Computer Engineering
Volume 2023, Article ID 9787854, 1 page
https://doi.org/10.1155/2023/9787854

Retraction

Retracted: Active Learning Query Strategies for Linear Regression Based on Efficient Global Optimization

Journal of Electrical and Computer Engineering

Received 19 December 2023; Accepted 19 December 2023; Published 20 December 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] T. Zong, N. Li, and Z. Zhang, "Active Learning Query Strategies for Linear Regression Based on Efficient Global Optimization," *Journal of Electrical and Computer Engineering*, vol. 2022, Article ID 2891463, 16 pages, 2022.

Hindawi
Journal of Electrical and Computer Engineering
Volume 2022, Article ID 2891463, 16 pages
https://doi.org/10.1155/2022/2891463

Research Article

Active Learning Query Strategies for Linear Regression Based on Efficient Global Optimization

Tianxin Zong, Na Li, and Zhigang Zhang

School of Mathematics and Physics, University of Science and Technology Beijing, Beijing, China

Correspondence should be addressed to Tianxin Zong; s20190831@xs.ustb.edu.cn

Received 16 April 2022; Revised 4 May 2022; Accepted 11 May 2022; Published 9 June 2022

Academic Editor: Xuefeng Shao

Active learning, a subfield of machine learning, can train a good model by selecting a minimum number of labeled samples. In many machine learning scenarios, needed information (such as the best value in unlabeled datasets) is acquired by prediction. When there is too little data in the training model, the prediction accuracy would obviously affect the accuracy of the results. To establish a high-performance regression model for a small dataset while accelerating the search for the best sample, a new active learning query strategy, EGO-ALR, that combines efficient global optimization (EGO) and active learning for regression (ALR) was proposed. It was found that the performance of EGO-ALR was significantly better than the original ALR query strategies in terms of the root mean square error (RMSE), correlation coefficient (CC), and opportunity cost (Oppo Cost). Specifically, EGO-ALR increased the Oppo Cost by an average of 25.27% when the RMSE and CC values were not more than 1.07% different from the original ALR. This study validated the efficiency and robustness of EGO-ALR approaches using 19 datasets from various domains and three distinct linear regression models (Ridge regression, Lasso, and Elastic network).

1. Introduction

Regression refers to estimating the value of a dependent variable (output) from one or more independent variables (characteristics). In a practical regression problem, some labeled samples (the independent variables and dependent variables are known) need to be trained by an appropriate approach to establish an accurate regression model. In general, the quality of performance of the trained model is proportional to the number of labeled samples. Data annotation is usually the biggest bottleneck in machine learning. Searching, managing, and labeling large amounts of data are often time-consuming and expensive to train a good model [1]. For example, in emotion estimation from speech signals, it is easy to record several speech utterances, but multiple assessors are needed to evaluate the emotion primitives [2]. In the problem of new alloy design, one can freely adjust the material composition within the range, but the synthesis and characterization of new materials should simultaneously consider the synthesis difficulty and cost of materials [3]. Similarly, in the research of video recommendation systems, users can simply upload videos, but few people manually annotate the metadata in detail; thus, costly annotation by experts is required, which will lead to a severe lack of text views and to a lack of training data for recommender systems [4].

To enable use of applications with missing labeled samples, investigators propose ALR [5]. ALR can sequentially select some of the most beneficial samples for labeling, so that the trained model gives the most accurate predictions for the remaining unlabeled samples. ALR is iterative: first, one builds an initial model from a small number of labeled training samples, and then by some selection strategies, the most valuable samples among the unlabeled samples are labeled and added to the training set for the next round of modeling. This process will iterate until stopping conditions are met, such as the maximum number of iterations, the maximum number of labeled samples, and the cross-validation accuracy of the model.

According to different query scenarios, investigators divide ALR into population-based ALR, flow-based ALR, and pool-based ALR [6]. The pool-based situation is considered in this study, in which a pool of unlabeled samples is given, and the goal is to select some training samples from the pool to improve a linear regression model.

When a machine learning regression model is established, people usually use this model to predict samples with unknown labels and then obtain the needed information from the prediction results. In material design problems, machine learning is frequently used to extrapolate to a vast unexplored search space to search for the best performing material [7], but the accuracy of predictions is closely related to the performance of the regression models. To guide the experiment to the ideal material quickly, Balachandran et al. combined active learning with experimental design and proposed an adaptive iterative design strategy [3] to accelerate the material discovery process. This strategy is used in the scenario of multiple material designs [8–11].

The adaptive strategy first defines a utility equation as the key to selecting the subsequent experimental sample. Then, the strategy predicts the most beneficial sample for experimental verification, and finally the strategy feeds the verified data to the machine learning model to improve the accuracy of predicting the best value. In this way, the samples with the best target performance are screened out with the least number of experiments. The utility functions commonly use the EGO algorithm [12] and the knowledge gradient algorithm [13].

ALR can only solve the problem using as few samples as possible to improve the ability of the machine learning model. However, training a high-precision regression model is not its ultimate goal in practical application. People prefer to predict the samples in an unknown space to get the best beyond the existing one. Although EGO can meet the needs of objective optimization, this strategy often sacrifices the precision of the model's prediction while performing rapid optimization [3]. So, there is currently no query strategy that perfectly balances the prediction performance and optimization performance under small sample conditions.

It is easy to find that the principle of adaptive strategy is similar to ALR. Both strategies select the most beneficial samples from some approaches. If the utility function of global optimization is integrated with ALR, can we accelerate the progress to finding the best samples while building a model with high predictive performance? To address this question, this paper studied a class of ALR approaches based on optimization algorithms. The principal contributions are the following:

(i) A brand-new AL query strategy, combining the EGO algorithm with the ALR approach, was proposed to optimally balance the needs of "exploitation" (aims at improving the predictive model) and "exploration" (aims at finding the best sample).

(ii) The EGO-ALR inherits the usage of EGO: it can freely change the optimization direction, and its performance is stable regardless of finding the minimum or maximum value.

(iii) Extensive experiments were carried out on three common linear regression models and 19 datasets from different application domains, demonstrating the effectiveness and robustness of EGO-ALR. It also shows that this query strategy can even outperform the original ALR in both prediction and optimization performance in some cases.

The remainder of this paper is organized as follows. Section 2 introduces the EGO algorithm and some existing pool-based ALR approaches. Section 3 elaborates on the combined framework of the proposed approach. Section 4 conducts extensive experiments on 19 datasets, elucidating the experimental results and superiority performance of our approach. Finally, the conclusion and future work are given in Section 5.

2. Related Work

2.1. Existing Pool-Based ALR Query Strategies. The existing pool-based ALR approaches can be classified into two scenarios: supervised and unsupervised. Most existing ALR approaches are supervised; these approaches need some ground-truth labels to guide the sample selection. Unsupervised ALR does not require any label information when selecting samples. Next, several commonly used supervised and unsupervised ALR approaches are introduced below.

2.1.1. Supervised ALR Query Strategies. Query by committee (QBC) [14] is widely used in different fields [15–19]. Assuming that N is the number of samples in the dataset, QBC first randomly selects and labels K0 samples, then establishes a committee of l learners from the existing training set (usually by bootstrapping), and predicts the samples in the unlabeled pool. QBC will select the samples from the pool on which the committee disagrees the most to label, that is:

$$\sigma_n^2 = \frac{1}{l}\sum_{i=1}^{l}\left(y_n^i - \mu_n\right)^2, \quad n = K_0 + 1, \ldots, N, \tag{1}$$

where $\mu_n = (1/l)\sum_{i=1}^{l} y_n^i$ and y_n^l is the prediction result of the ith model built by bootstrap for the nth unlabeled sample x_n. QBC selects the sample with the largest σ_n^2 to label.

Expected model change (EMCM) [20] is an ALR approach for regression and classification [21–23]; EMCM has a variety of algorithms [24–26]. Expected model change for linear regression is considered in this report. Expected model change first randomly selects and labels K_0 samples to train a linear regression model. The prediction result of the model for the nth unlabeled sample is set as $\hat{y}_n$. Then, EMCM uses bootstrap to build l linear regression models. In each sequential iteration, the labeled samples are those that change the linear regression parameters the most, that is:

$$g\left(x_n\right) = \frac{1}{l}\sum_{i=1}^{l}\left\|\left(y_n^i - \hat{y}_n\right)x_n\right\|, n = K_0 + 1, \ldots, N. \tag{2}$$

EMCM labels the sample with the largest $g\ (x_n)$.

2.1.2. Unsupervised ALR Query Strategies. Yu and Kim proposed an unsupervised ALR approach based on greedy sampling [27], which is also applied to image signal processing [28]. Greedy sampling initially labels at least one sample, but Yu and Kim do not define the first sample. Therefore, the study used an improved method, GSx, proposed by Wu et al. [29]. The idea of GSx is to take the sample closest to the centroid in the pool as the first labeled sample and then select a sample in a greedy way such that it is farthest from all the selected samples at each sequential iteration:

$$d_{nm} = \left\| x_n - x_m \right\|, \quad m = 1, \ldots, K_0, \; n = K_0 + 1, \ldots, N, \tag{3}$$

where x_m is the labeled sample. GSx first calculates the distance between x_n and x_m of all unlabeled samples, then it computes the minimum distance from x_n to x_m:

$$d_n = \min_m d_{mn}, \quad n = K_0 + 1, \ldots, N, \tag{4}$$

and selects the sample with the largest d_n.

Representation-diversity (RD) proposed by Wu is also unsupervised and can be used for linear regression [30], and RD derived some excellent unsupervised ALRs such as IRD [31] and iRDM [1]. It performs k-means clustering ($k = K_0$) and selects the first K_0 samples closest to the centroid from each cluster. When selecting the $K_0 + 1$th sample, RD performs k-means clustering ($k = K_0 + 1$) on all samples in the pool, identifies the largest cluster that does not contain the labeled samples, and selects the one closest to the centroid as the $K_0 + 1$th sample. The basic RD algorithm can also be combined with other supervised ALR approaches for better performance [30]. For example, RD-EMCM combines RD and EMCM.

The ALR approach improves only the prediction performance as the criterion for selecting samples. Due to multiple limitations such as too few labeled samples, the complexity of the dataset distribution, and the defects of the algorithm itself, it is difficult for even the ALR to achieve the required prediction accuracy in only a few iterations. The results obtained by such a model have many deviations and cannot be used as a reference.

2.2. Efficient Global Optimization. EGO [12] is an algorithm with many related extensions to different types of research [32, 33]. We first introduce the expected improvement [34] before introducing EGO. Let $f^* = \min\ (y_1, \ldots, y_m)$ be the current best value in the training set. Before labeling x_n, its value y_n is uncertain. The uncertainty at y_n is modeled as the realization of a normally distributed random variable Y with mean and standard deviation determined by bootstrap. If the normal density function with the mean and standard deviation is plotted at x_n, y_n has a certain probability to be better than f^*. Expected improvement (EI) weighs all possible improvements by the associated density value at the point. Formally, the improvement at x_n is $I = \max\ (f{*}\text{-}Y, 0)$. Because Y is a random variable, this expression is also a random variable. Simply take the expected value to obtain the EI:

$$E[I(x_n)] = E[\max(\mu * - Y, 0)]. \tag{5}$$

To compute this expectation, the notations μ_n and σ_n are introduced to denote the expectation and standard deviation at x_n. Y is normal (μ_n, σ_n^2). By performing some integrals by parts on the right side of equation (5), one can expand as

$$E[I(x_n)] = (\mu * - \mu_n)\Phi\left(\frac{\mu * - \mu_n}{\sigma_n}\right) + \sigma_n \phi\left(\frac{\mu * - \mu_n}{\sigma_n}\right). \tag{6}$$

In the above equation, $\phi\ (\cdot)$ and $\Phi\ (\cdot)$ are the standard normal density and distribution function. EGO will select the sample with the largest $E[I(x_n)]$.

The EGO considers both the predicted (uncertain) and optimized (best) value, but the prediction error of the EGO-constructed model is still high [3]. The advantage of EGO is that in the case of large regression error, the approach can also be more effective than random selection or direct selection of the best value of the prediction. Because of this characteristic, EGO is used mostly in data-driven material design applications, which require selecting samples with better performance by a small number of iterations.

3. ALR Query Strategies Integrated with EGO

ALR is a subfield of machine learning that can train a good model by selecting a minimum number of labeled samples, and EGO is a global optimization algorithm that can quickly find the best sample. This study proposes a query strategy that integrates EGO with ALR, called EGO-ALR. By combining the advantages of the EGO and the ALR during sample selection, the method can accelerate the optimization process while maintaining prediction quality, which effectively reduces the influence of model performance on the outcome.

The complete framework of the active learning method using EGO-ALR query strategy is shown in Figure 1. First, resample the training set by bootstrap for l times. Second, train the resampled set into a regression model through machine learning. Then, predict all samples in the pool and use the EGO-ALR query strategy to calculate the information of each unlabeled sample. Finally, select and mark the most informative sample and add it to the training set.

Due to the different principles of supervised and unsupervised ALR query strategies, this paper needs to discuss the different approaches of EGO combined with the two kinds of ALR query strategies and point out some special changes.

3.1. Combination of Supervised ALR and EGO. The pseudocode of the supervised ALR combined with EGO is given in Algorithm 1. Pool U firstly consists of N unlabeled samples and 0 labeled samples. Set K_0 as the number of samples in the initial training set. Because it is combined with the supervised ALR approach, all samples in the initial training set will be randomly selected and labeled. Assuming that the first K samples ($K \geq K_0$) have been labeled, for the remaining N-K unlabeled samples, EGO-ALR first computes separately the "information" in both EGO and supervised ALR.

The "information" is defined as a measure of how valuable a sample is to be labeled, for example, σ_n^2 used in QBC, $g\ (x_n)$ used in EMCM, and $E[I(x_n)]$ used in EGO. Note that the "information" between each approach may have

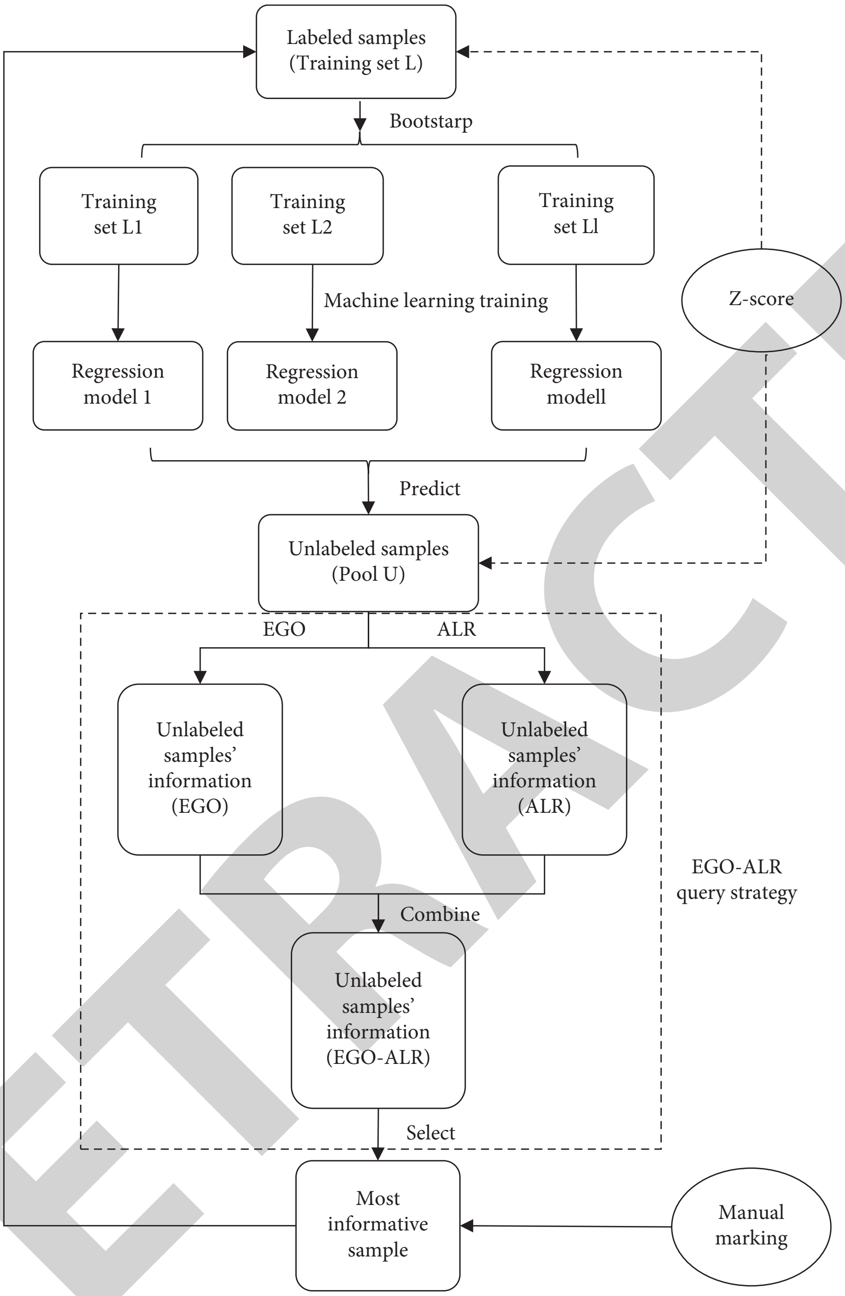


Figure 1: Framework of the active learning approach using EGO-ALR query strategy.

significantly different dimensions, and a larger scale may dominate the other "information." Thus, EGO-ALR normalizes the "information" by min-max normalization and then adds it after weighting by the parameter c to reduce the sensitivity of the formula to scale. Here is an example of the "information" after the combination of EGO and QBC:

$$T_n = c \cdot \sigma_n^{2\,*} + E[I(x)]^*, \tag{7}$$

where c is the adjustable weight and $*$ represents the normalized value. For labeling, EGO-ALR selects the sample with the largest T_n.

Because the value of parameter c is the most effective way to balance the prediction performance of ALR and the optimization performance of EGO, the value has a crucial influence on the sampling results. Generally, the larger the c value, the closer the sample selection to ALR, and model prediction performance increases while optimization performance (find the best value) decreases; the smaller the c value, the closer the sample selection to EGO, and model prediction performance decreases while optimization performance increases. The effect of different parameters c on the results is explained in Section 4.7.

3.2. Combination of Unsupervised ALR and EGO.

The combined query strategy of unsupervised ALR is similar to

```
Input: $x_n$, a pool of $N$ unlabeled samples; $K$, maximum number of samples to label;
$c$, weighting parameters.
Output: regression model $f(x)$
Randomly select and label $K_0$ samples;
Construct the initial regression model $f(x)$ with $K_0$ samples;
for $m = K_0 + 1, \ldots, K$ do
    Build $L$ regression models using bootstrap from the training set
    for $n = m, \ldots, K$ do
        EGO-QBC: compute $\sigma_n^2$ in (1) and $E[I(x_n)]$ in (6);
        min-max normalization of $\sigma_n^2$ and $E[I(x_n)]$, marked as $\sigma_n^{2*}$ and $E[I(x)]^*$
        Compute $T_n = c \cdot \sigma_n^{2*} + E[I(x)]^*$
        EGO-EMCM: compute $g(x_n)$ in (2) and $E[I(x_n)]$ in (6);
        min-max normalization of $g(x_n)$ and $E[I(x_n)]$, marked as $g(x_n)^*$ and $E[I(x)]^*$
        Compute $T_n = c \cdot g(x_n)^* + E[I(x)]^*$
    end
    Label the sample with the largest $T_n$ and add it to the training set.
end
Update the regression model $f(x)$ with the labeled $K$ samples.
```

Algorithm 1: The proposed supervised EGO-ALR approach.

```
Input: $x_n$, a pool of $N$ unlabeled samples; $K$, maximum number of samples to label;
$c$, weighting parameters.
Output: regression model $f(x)$
Select and label the initial $K_0$ samples with the GSx (or RD) algorithm;
Construct the initial regression model $f(x)$ with $K_0$ samples;
for $m = K_0 + 1, \ldots, K$ do
    Build $L$ regression models using bootstrap from the training set
    for $n = m, \ldots, K$ do
        EGO-GSx: compute $d_n$ in (4) and $E[I(x_n)]$ in (6);
        min-max normalization of $d_n$ and $E[I(x_n)]$, marked as $d_n^*$ and $E[I(x)]^*$
        Compute $T_n = c \cdot d_n^* + E[I(x)]^*$
        RD-EGO: perform $k$-means ($k = n$) clustering on all samples in the pool;
        Identify the largest cluster that does not contain labeled samples
        Compute $E[I(x_n)]$ in (6) for the samples in the cluster
    end
    Label the sample with the largest $T_n$ (or $E[I(x_n)]$) and add it to the training set.
end
Update the regression model $f(x)$ with the labeled $K$ samples.
```

Algorithm 2: The proposed unsupervised EGO-ALR approach.

supervised ALR, with the difference in the initial training samples. "Unsupervised" means the selection of samples is independent of the label information. When the pools are all unlabeled samples, unsupervised ALR can still select samples for labeling by corresponding strategies. Algorithm 2 gives the pseudocode. The combination of GSx and EGO uses the method described in Section 3.1. The combination of RD and EGO adopts the method described by Wu [30], that is, it selects RD to initialize and uses EGO to select samples in the largest cluster lacking labeled samples.

In summary, EGO-ALR randomly (or using unsupervised ALR) selects the first K_0 samples to build an initial regression model, and, in each subsequent iteration, EGO-ALR chooses the sample with the largest T_n (or $E[I(x_n)]$) to achieve the combination and balance of "exploitation" and "exploration."

4. Results

This section conducted experiments on 19 datasets and three linear regression models to establish the performance of the proposed EGO-ALR. The experimental device was a personal computer, and the programming language was MATLAB R2018b.

4.1. Data Sources. A total of 19 datasets were used in the experiment. Sixteen datasets were from the UCI Machine Learning Library and three were from the CMU StatLib

Table 1: Summary of the 19 regression datasets.

Dataset	Source	No. of samples	No. of raw features	No. of final features
Concrete-CS	UCI[1]	103	7	7
Concrete-Flow	UCI[1]	103	7	7
Concrete-Slump	UCI[1]	103	7	7
Yatch	UCI[2]	308	6	6
AutoMPG	UCI[3]	392	7	9
RealEstate	UCI[4]	414	6	6
NO2	StatLib[5]	500	7	7
PM10	StatLib[6]	500	7	7
Housing	UCI[7]	506	13	13
CPS	StatLib[8]	534	10	19
Energy-Cooling	UCI[9]	768	7	7
Energy-Heating	UCI[9]	768	7	7
Concrete	UCI[10]	1030	8	8
Airfoil	UCI[11]	1503	5	5
Wine-red	UCI[12]	1599	11	11
Wine-white	UCI[12]	4898	11	11
HEA	Journal[13]	165	9	9
Direct	Journal[14]	534	12	12
Indirect	Journal[14]	1836	15	15

[1] https://archive.ics.uci.edu/ml/datasets/Concrete+Slump+Test. [2] https://archive.ics.uci.edu/ml/datasets/Auto+MPG. [3] https://archive.ics.uci.edu/ml/datasets/Real+estate+valuation+data+set. [4] https://archive.ics.uci.edu/ml/datasets/Real+estate+valuation+data+set. [5] http://lib.stat.cmu.edu/datasets/NO2.dat. [6] http://lib.stat.cmu.edu/datasets/PM10.dat. [7] https://archive.ics.uci.edu/ml/machine-learning-databases/housing/. [8] http://lib.stat.cmu.edu/datasets/CPS_85_Wages. [9] https://archive.ics.uci.edu/ml/datasets/energy+efficiency. [10] https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength. [11] https://archive.ics.uci.edu/ml/datasets/airfoil+self-noise. [12] https://archive.ics.uci.edu/ml/datasets/wine+quality. [13] https://www.sciencedirect.com/science/article/abs/pii/S1359645419301430. [14] https://pubs.acs.org/doi/abs/10.1021/acsami.1c15021.

Datasets Archive. These sources have been used in many ALR studies [1, 20, 23, 27, 29–31]. Table 1 summarizes the datasets. Before the experiment, all datasets were removed of samples with missing features, special characters, and garbled characters. Two datasets, AutoMPG and CPS, contained some categorical features, which needed to be converted into numerical features by one-hot coding before the experiment (this conversion increased the number of features).

In addition, there are three datasets from the field of materials collected from the literature: HEA [10], Direct [11], and Indirect [11]. Note that the features of the HEA dataset were calculated from the data and feature formula provided by the literature. Before the experiment, each dimension of the feature space was normalized by Z-score, so that the mean of the feature dimension was zero and the standard deviation was one.

4.2. Comparison Algorithm. The study compared the performance of 10 different approaches as follows:

(1) Base line, BL, which randomly selects all samples for labeling.

(2) EGO, which is introduced in Section 2.

(3) EMCM: supervised ALR, which is introduced in Section 2.1.

(4) EGO-EMCM ($c = 2$): the combination of EGO and EMCM, which is introduced in Section 3.1.

(5) QBC: supervised ALR, which is introduced in Section 2.1.

(6) EGO-QBC ($c = 2$): the combination of EGO and QBC, which is introduced in Section 3.1.

(7) GSx: unsupervised ALR, which is introduced in Section 2.2.

(8) EGO-GSx ($c = 2$): the combination of EGO and GSx, which is introduced in Section 3.2.

(9) RD: unsupervised ALR, which is introduced in Section 2.2.

(10) RD-EGO: the combination of EGO and RD, which is introduced in Section 3.2.

4.3. Evaluation Process. For each dataset, first randomly select 50% of the total samples as the training pool U and the remaining 50% as the test set T: U (50%) +T (50%). Because the benefits of the ALR method are reflected in modeling with a small number of samples, each approach selected $K\in [5, 50]$ sample from the training set U. The entire process was repeated 100 times to eliminate the effect of randomness on the results.

After each iteration of each approach, RMSE and CC were computed as measures of prediction performance. To measure the ability to find the best value of different approaches, Oppo Cost was also used in the evaluation. Oppo Cost was defined as the modulus difference between the current best and the overall best [3]. Powell and Ryzhov [35] also used Oppo Cost to compare the performance of knowledge gradient and EGO algorithms. To horizontally compare different approaches on different datasets, the Oppo Cost in this paper specifically refers to the normalized Oppo Cost.

The formulas of the three evaluation indicators are as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}, \tag{8}$$

$$\text{CC} = \frac{n\sum_{i=1}^{n} y_i \hat{y}_i - \sum_{i=1}^{n} y_i \sum_{i=1}^{n} \hat{y}_i}{\sqrt{n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2} \cdot \sqrt{n\sum_{i=1}^{n} \hat{y}_i^2 - \left(\sum_{i=1}^{n} \hat{y}_i\right)^2}}, \tag{9}$$

$$\text{Oppo Cost} = \frac{\left|\mu^{**} - \mu^{*}\right|}{y_{\max} - y_{\min}}, \tag{10}$$

where y_i is the actual value of the test set sample, $\hat{y}_i$ is the predicted value of the test set sample, and n is the number of samples in the test set; $i = 1, 2, \ldots, n$. μ^* is the best-so-far and

μ^{**} is the overall best value in the training pool. $y_{\max}$ and $y_{\min}$ are the maximum and minimum values in the training pool. Note that RMSE and CC evaluate the ability of the regression model to predict the samples in the test set, whereas Oppo Cost evaluates the ability to find the samples with the best performance during the iteration. Thus, (10) is calculated for the training pool, not the test set.

Note that CC was not directly optimized in the objective function of these approaches [1, 29, 30]. Generally, a regression model with a CC close to 1 should have a smaller RMSE, but there is no guarantee (see the experimental results below for details). Thus, the CC can be viewed only as a secondary measure of prediction performance for reference.

For each approach, three regularized linear regression models were used for training:

(i) Ridge regression, Ridge [36]: regularization coefficient $\lambda = 0.1$.

(ii) Lasso regression, Lasso [37]: regularization coefficient $\lambda = 0.1$.

(iii) Elastic network, Enet [38]: regularization coefficient $\lambda = 0.1$, penalty item mixing parameter $\alpha = 0.5$.

The regularized regression was chosen over ordinary linear regression because the number of labeled samples was too small. Thus, the model, which regularized the coefficients, usually achieved better performance compared with the ordinary linear regression model.

4.4. Experimental Result on Ridge. Figure 2 shows the average RMSE, CC, and Oppo Cost of different optimization directions for 10 methods with 19 datasets when using Ridge as the regression model.

(1) The performance of all ten approaches improved as the value of K increased (smaller RMSE, Oppo Cost, and larger CC), which was intuitive. However, because of the small number of samples in the early stage, there were some fluctuations in the result. For example, when $K \in [5, 10]$, the RMSE and CC results on 15 of the 19 datasets showed unexpected changes (larger RMSE and smaller CC when K increases). Of course, this problem was lessened after K continued to increase.

(2) Intuitively, the prediction performance of the ten approaches was better than BL in most datasets, which suggested that the samples selected by the strategy could indeed improve the performance of the regression model.

(3) Most algorithms with better optimization performance are related to EGO. EGO-ALR approaches in different optimization directions all had smaller Oppo Costs compared with ALR approaches. EGO achieved the smallest Oppo Costs on 15 of the 19 datasets (the remaining four datasets had the second smallest Oppo Cost). The above shows that the EGO and the approaches combined with EGO were the best sample selection approaches for optimization, no matter finding the maximum or the minimum.

This study additionally computed the area under the curve (AUC) of the mean RMSEs, CCs, and Oppo Costs for the Ridge regression model, denoted as AUC-RMSE, AUC-CC, and AUC-OPPO, respectively, to compare the prediction and optimization performance more concretely between the approaches (Figure 3). Because the AUCs from different datasets varied greatly, the AUC results were normalized to the AUC of BL; thus, the AUC of BL was always 1.

We made the following observations:

(1) On average, GSx had the largest AUC-CC (1.0857) and the smallest AUC-RMSE (0.8384) for most datasets. The prediction performance of QBC (AUC-CC = 1.0648, AUC-RMSE = 0.8831) was slightly better than EMCM (AUC-CC = 1.0588, AUC-RMSE = 0.8897), and both were better than RD (AUC-CC = 1.0224, AUC-RMSE = 0.9314). The performance of EGO (AUC-CC = 1.0297, AUC-RMSE = 0.9691) was only better than BL.

(2) For most datasets, EGO-EMCM, EGO-QBE, and EGO-GSx had similar prediction performance relative to their original ALR. Specifically, the maximum absolute value of AUC-CC between the three combined approaches and their original ALR was 0.176, and the average was 0.007. The maximum absolute value of the AUC-RMSE difference was 0.111, and the average was 0.009. The results of RD-EGO were peculiar; its average AUC-CC was 1.0271 and greater than RD in 13 of the 19 datasets. Meanwhile, the average AUC-RMSE of RD-EGO was 0.9143 and smaller than RD in 13 datasets of the 19 datasets. The prediction performance of RD-EGO was better than the RD on average, which was consistent with the description of the performance of RD combined with other approaches reported by Wu [30].

(3) The optimization results of RD (average 1.1638) were the worst for 17 of the 19 datasets and the second lowest on the other two datasets. The optimization performance of QBC (average 0.6542) was also slightly better than EMCM (average 0.6625). GSx (average 0.5803) was the best among the four ALR approaches, whereas EGO (average 0.43957), as a global optimization algorithm, had better performance compared with all approaches.

(4) EGO-EMCM, EGO-QBC, and RD-EGO had significantly smaller AUC-OPPO than their original ALR in all datasets. EGO-GSx had smaller AUC-OPPO (average 0.4829) than GSx for 17 of 19 datasets. Of the remaining two datasets, the AUC-OPPO difference between EGO-GSx and GSx was at most 0.067. Generally, the optimization performance of the four EGO-ALRs was always better than all ALRs. EGO-GSx, as the combination approach of EGO and GSx, had better optimization performance than the other three EGO-ALRs. The optimization performance of the remaining three approaches, ranked from the best to the worst, was RD-EGO, EGO-QBC, and EGO-EMCM.

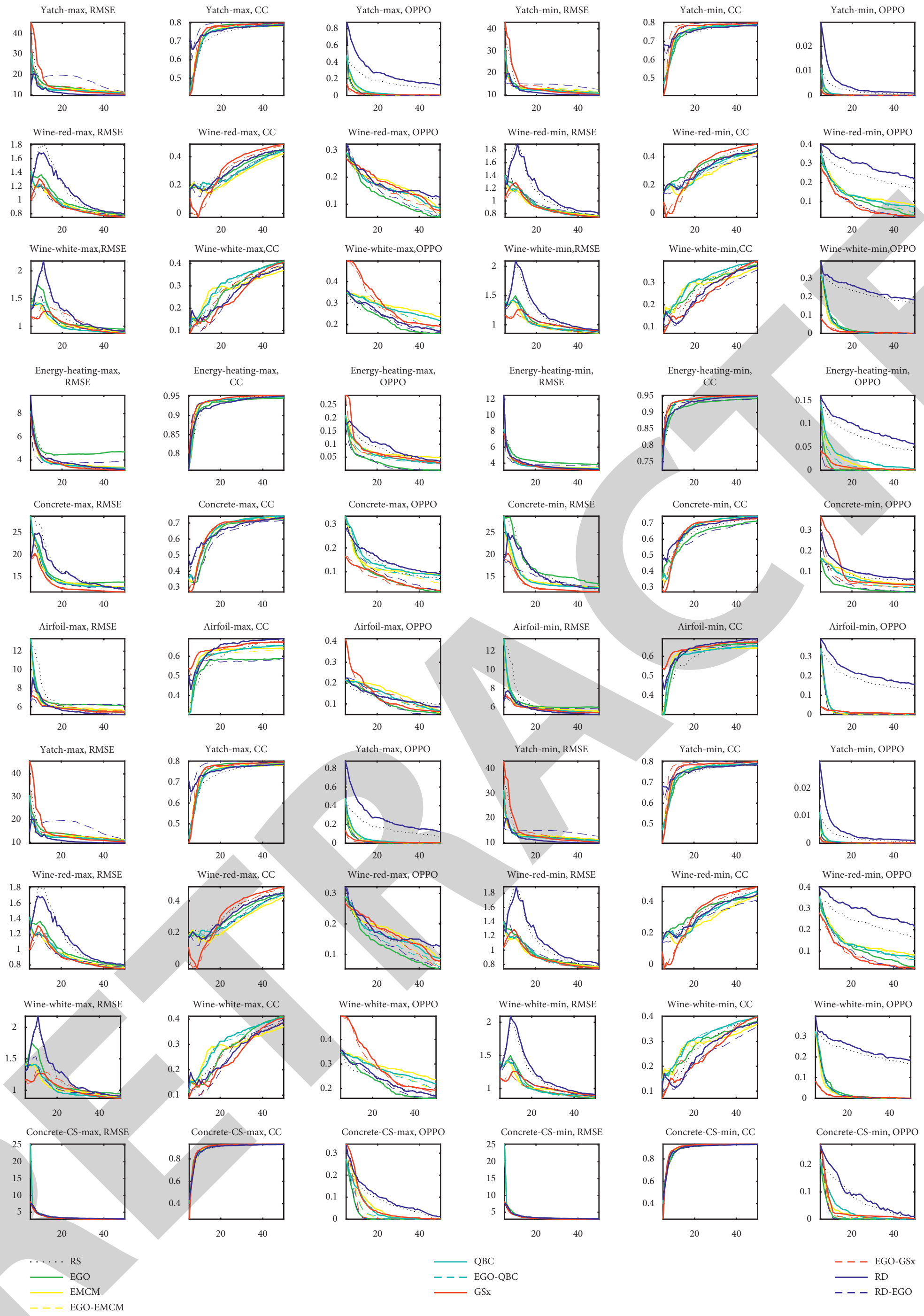


Figure 2: Continued.

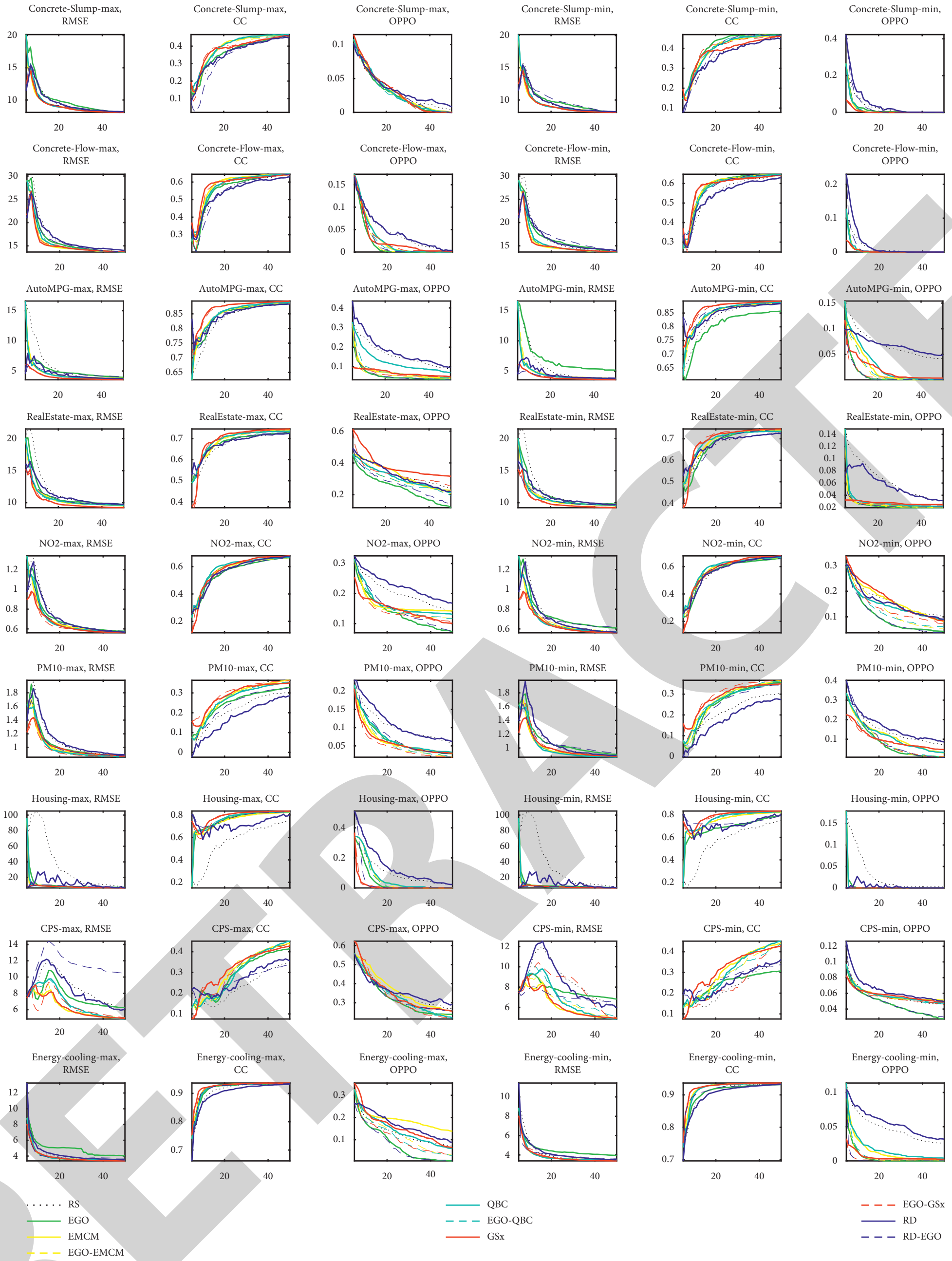


FIGURE 2: Mean RMSEs, CCs, and Oppo Costs of 19 datasets, averaged over 100 runs and different optimization directions. The horizontal axis represents K. Ridge ($\lambda = 0.1$) was used as the regression model.

In summary, the rank of the prediction performance of the 10 approaches could be sorted as follows: GSx ≈ EGO-GSx > QBC ≈ EGO-QBC > EMCM ≈ EGO-EMCM > RD-EGO > RD > EGO > BL. The optimization performance ranking was EGO > EGO-GSx > RD-EGO > GSx > EGO-QBC > EGO-EMCM > QBC > EMCM > BL > RD. The rank confirms that our proposed method, whether combined with supervised ALR or unsupervised ALR, exhibited strong advantages in improving the optimization performance without significant loss of prediction performance.

The measurement standard of the algorithm is not only accuracy but also stability. In the case of similar algorithm performance, the more stable algorithm is usually chosen. Table 2 shows the percent improvement of the AUCs of the mean RMSEs and CCs over BL. Ridge was the regression model.

As seen from Table 2, the improvement of RMSE and CC of all ALR approaches was better than EGO (RMSE = 3.09%, CC = 1.81%). According to the results of RMSE, GSx (mean = 16.17%, std = 15.27%) had the largest and most stable improvement compared with BL, followed by EGO-GSx. CC showed that the improvement of GSx (mean = 7.65%, std = 2.14%) was the largest, the improvement of QBC (mean = 4.08%, std = 3.44%) was the most stable, and

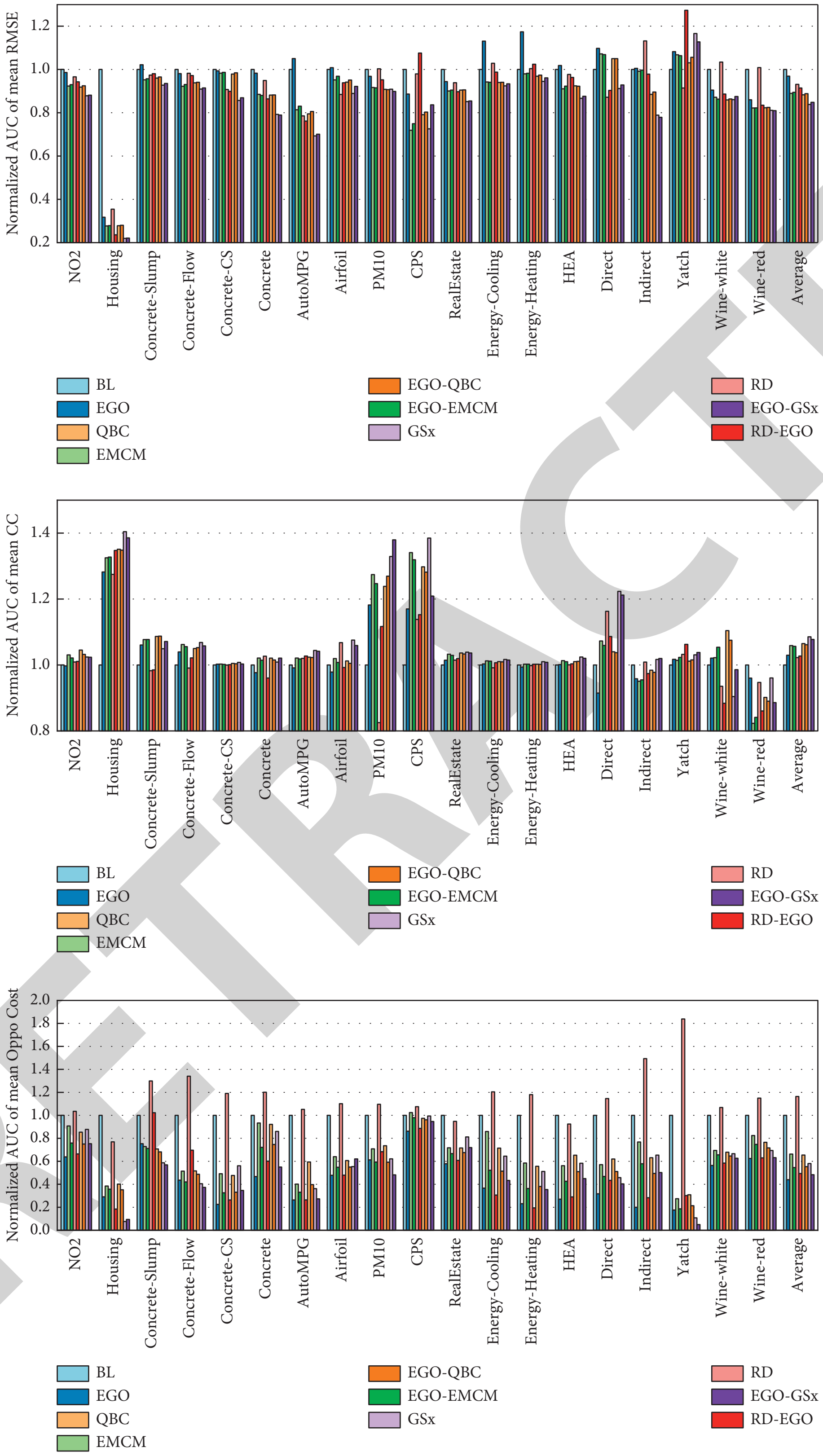


FIGURE 3: Normalized AUCs of the mean RMSEs, CCs, and Oppo Costs for the 19 datasets. Ridge ($\lambda = 0.1$) was used.

Table 2: Percent improvements of the AUCs of the mean RMSEs, CCs, and Oppo Costs over BL (Ridge was the regression model; the best performances are in bold).

		EGO	EMCM	EGO-EMCM	QBC	EGO -QBC	GSx	EGO -GSx	RD	RD -EGO
RMSE	Mean	3.09	11.03	10.57	11.69	11.16	**16.17**	15.19	6.86	8.57
	std	0.65	11.12	9.92	9.77	9.97	**15.27**	14.45	4.63	5.81
CC	Mean	1.81	4.08	4.40	5.00	5.01	**7.65**	6.41	4.20	1.96
	std	2.35	**3.44**	3.42	3.55	3.40	2.14	1.58	−1.81	−1.88
Oppo Cost	Mean	**56.04**	33.75	45.48	34.58	44.68	41.97	51.72	−16.38	50.65
	std	**34.73**	16.99	24.37	23.07	26.83	11.68	20.07	−5.19	31.17

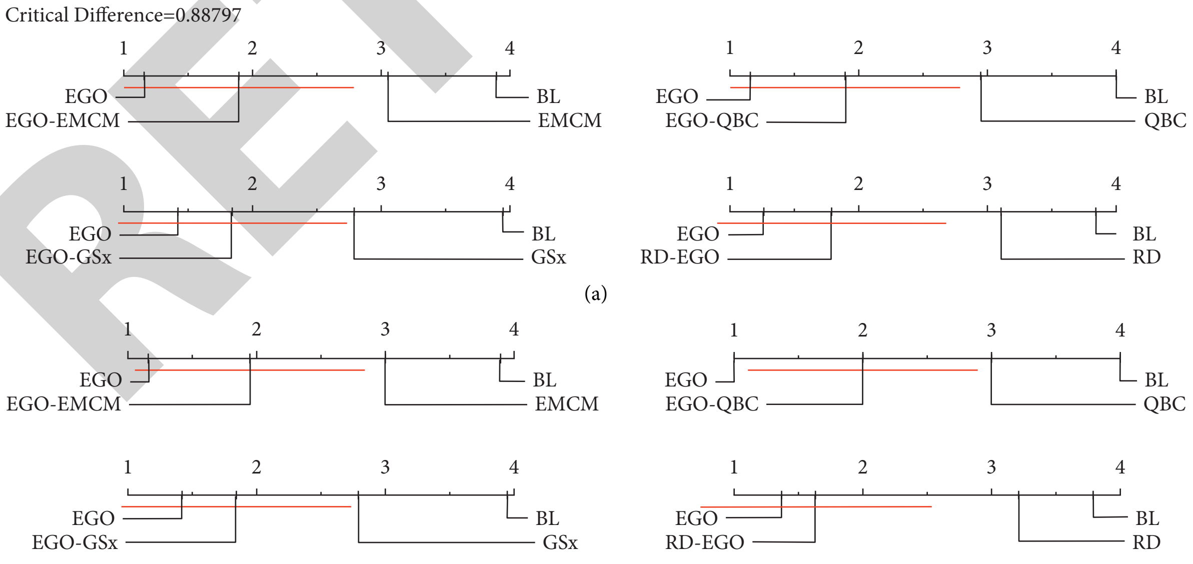
Table 3: Percent improvements of the AUCs of the mean RMSEs, CCs, and Oppo Costs compared with each original ALR approach.

		EGO-EMCM	EGO-QBC	EGO-GSx	RD-EGO
Ridge	RMSE	−0.96	−0.29	−1.07	**1.31**
	CC	−0.16	−0.27	−0.45	**−0.13**
	Oppo Cost	15.37	13.42	14.57	**54.37**
Lasso	RMSE	−0.99	−0.81	−0.82	**3.48**
	CC	**1.20**	−0.36	−0.48	−0.90
	Oppo Cost	22.17	10.92	16.35	**55.45**
Enet	RMSE	−0.70	**−0.68**	−0.75	−0.88
	CC	**0.37**	−0.55	−0.62	−1.99
	Oppo Cost	18.60	11.82	15.82	**54.40**

Table 4: Ff values of Friedman test on the AUCs of the Oppo Costs in four test groups (the critical value was 2.78 ($\alpha = 0.05$)).

	EGO-EMCM	EGO-QBC	EGO-GSx	RD-EGO
Ridge	136.71	217.43	55.51	94.81
Lasso	110.93	—[1]	55.51	94.81
Enet	37.44	57.91	12.77	33.90

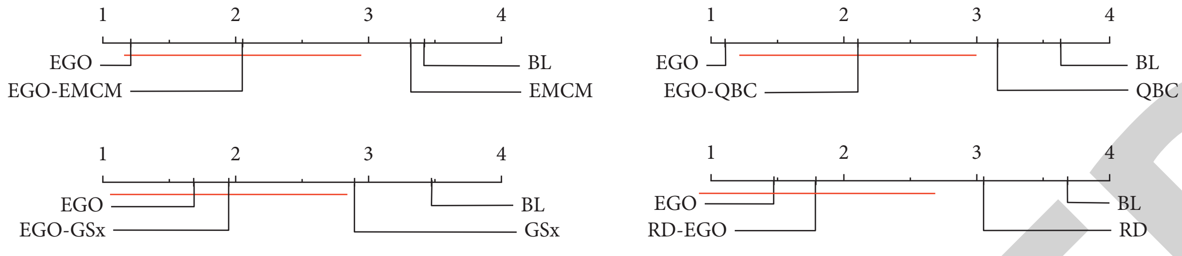
[1] The denominator is 0.



(a)

(b)

Figure 4: Continued.



(c)

Figure 4: Comparison of EGO-ALR against the others with the Bonferroni–Dunn test: (a) Ridge, (b) Lasso, and (c) Enet. All approaches with ranks outside the marked interval are significantly different ($p < 0.1$) from the control.

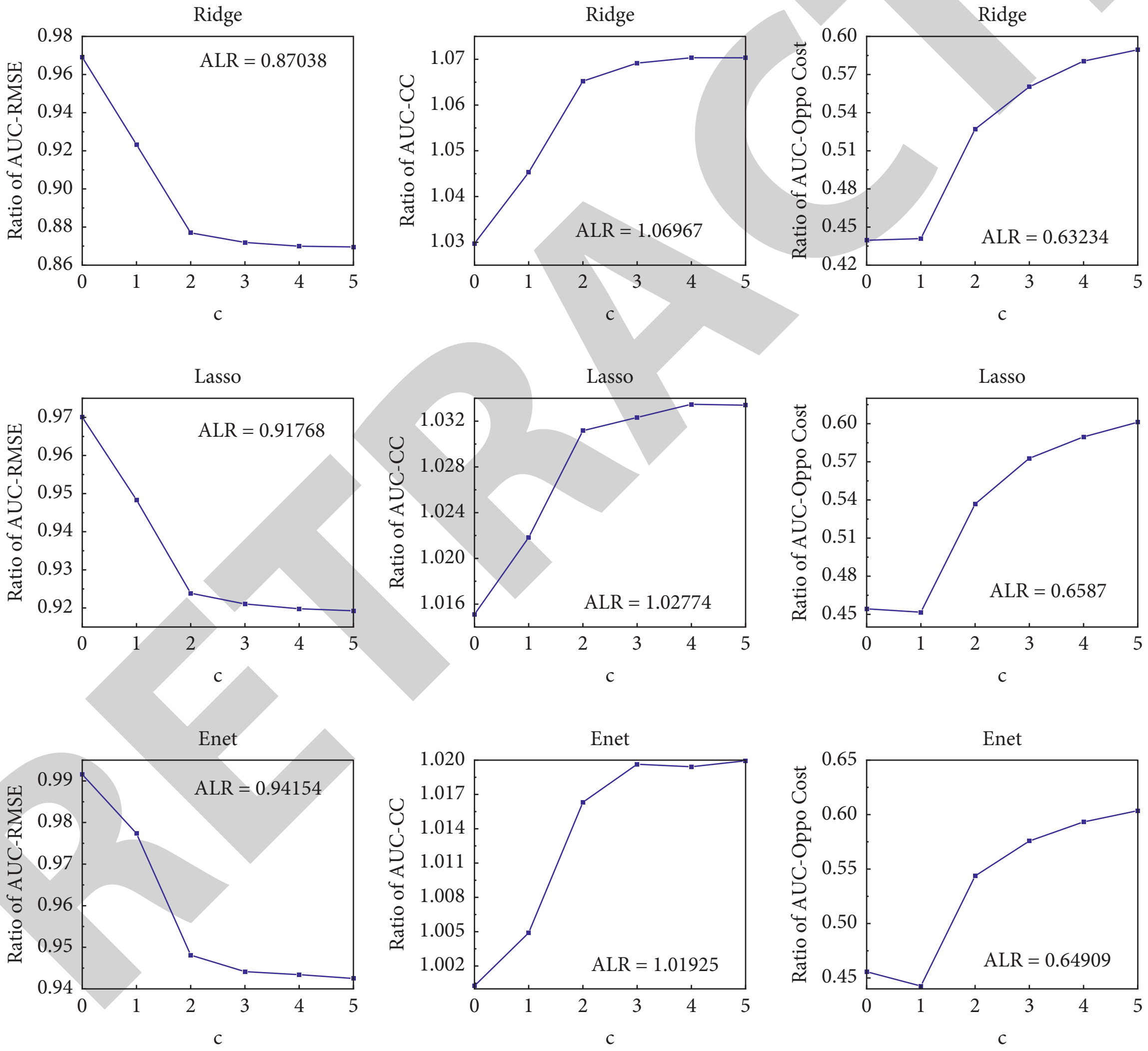


Figure 5: Ratios of AUCs of the mean RMSEs, CCs, and Oppo Costs with different *c*, average of 19 datasets, 100 runs, and different optimization directions.

the improvement of the standard deviation of RD (−1.81%) and RD-EGO (−1.88%) was negative, which indicated that these two approaches were very unstable. EGO-ALR and its original ALRs had a similar improvement in CC and RMSE, and the difference in improvement was less than 1%.

EGO had the largest and most stable improvement in Oppo Cost compared to BL, while EGO-ALRs had a larger and more stable improvement than the four ALR approaches. These results correspond to the ranking of the performance.

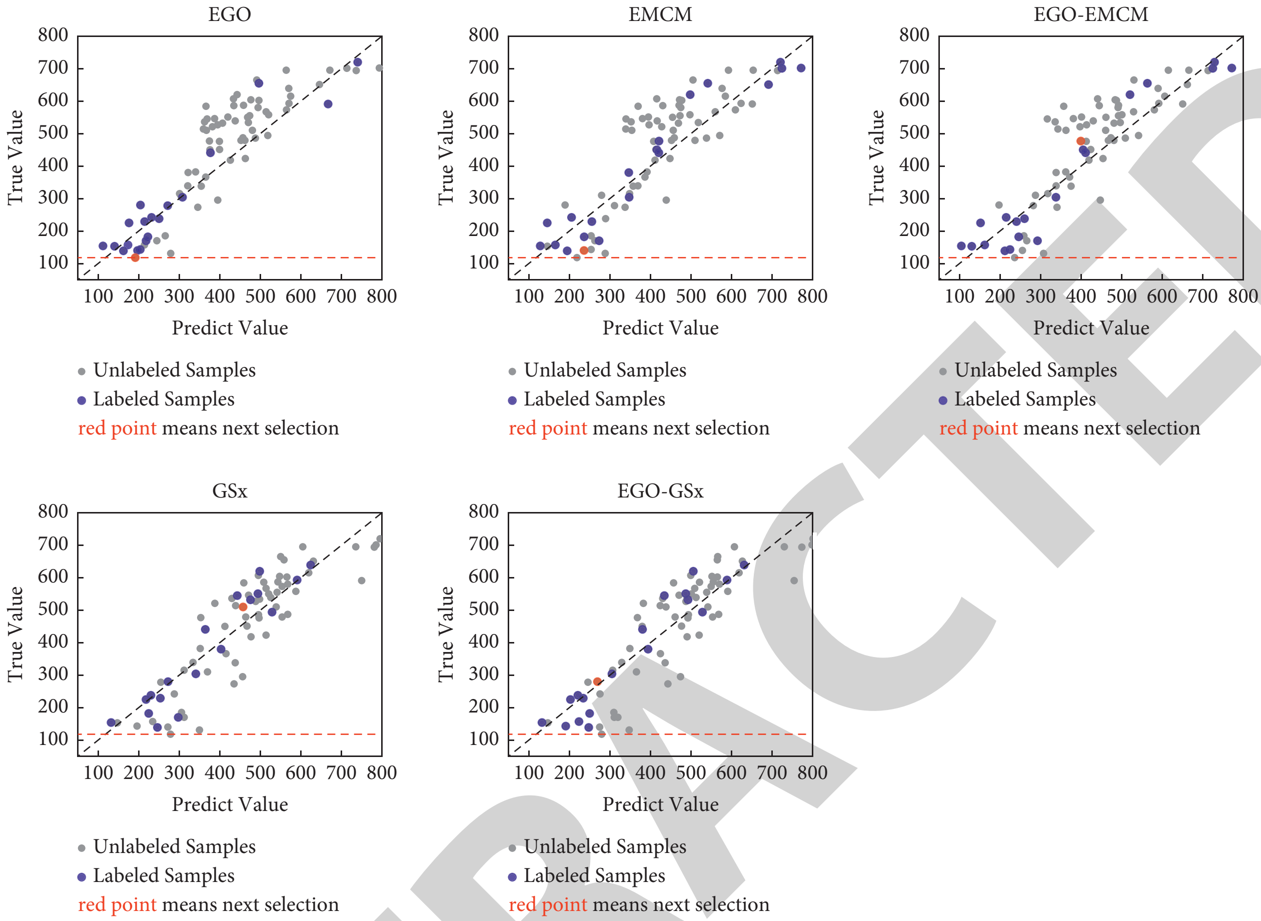


Figure 6: Visualization of sampling results from different approaches during active learning iteration to the 20th round.

4.5. Experimental Results on Lasso and Enet. All the foregoing experiments were repeated with Lasso and Enet models. The conclusions were similar to Section 4.4. For additional results, see Figures S1 and S2 and Tables S1 and S2 in the Supplementary Material. Compared with Ridge, the optimization performance of the 10 approaches was improved significantly with Lasso and Enet, and the standard deviation of RMSE was improved most obviously with Lasso.

To quantify the performance improvement of the four EGO-ALRs compared with their original ALRs, this study computed their percent improvements with the three regression models (Table 3). The lowest promotion percent on RMSE was −1.07%, and the lowest on CC was −1.99%. The lowest on Oppo Cost was 10.92%, and the highest was 55.45%. Regardless of the regression model, the percent improvement of EGO-ALRs in RMSE and CC was not less than −2%, but the percent improvement of Oppo Cost was more than 10%.

RD-EGO had the most significant improvement over RD; the Oppo Cost increased by 55.45% and the RMSE increased by more than 1% on both Ridge and Lasso. The improvement in the Oppo Cost of EGO-EMCM was second only to RD-EGO, and the CC of EGO-EMCM was also positive on the Lasso and Enet models.

4.6. Statistical Analysis. This section established the test groups that compared EGO-ALR with its original ALR, EGO, and BL to see if the differences in Oppo Cost between EGO-ALR and other approaches were statistically significant (EGO-ALR was used as the control approach). There were four EGO-ALRs in our work, so there were four test groups.

First, the Friedman test was performed on 19 datasets, and the calculated statistic F*f* (Table 4) was proposed by Iman and Davenport [39]. All calculated F*f* values were always greater than the critical value F (3,54) = 2.78, which suggested that, regardless of which regression model was used, there were statistical differences among these approaches in each group.

After that, the post hoc test was performed to compare methods. The power of the post hoc test is greater when all methods are compared only to the control method and not between each other [40]. Thus, we used the Bonferroni–Dunn test as post hoc test. At $q = 0.1$, the critical difference (CD) for comparing four approaches on 19 datasets was 0.8880, and the visualization of the post hoc test is shown in Figure 4.

Irrespective of the regression model, the opportunity cost of EGO-ALR was significantly better than that of the original ALR and BL. In addition, there was no significant difference in the average ranking between EGO-ALRs and

EGO except for EGO-QBC on Lasso and Enet models. Thus, it is concluded that the performance improvement of the original ALR and BL by the EGO-ALR was statistically significant. However, the improvement was not significantly different from that of EGO.

4.7. Parameter c Sensitivity. EGO-ALR in Section 3.1 has a parameter c, the weighted value of "information." This section investigated the effect of c on the performance of the EGO-ALR. Figure 5 shows the normalized AUCs of EGO-ALR on Ridge, Lasso, and Enet when the parameter $c \in [0, 5]$. The corresponding results of EGO are also marked on the figure for comparison. Note that EGO-ALR is equivalent to the EGO when $c = 0$.

The performance of EGO-ALR improved as c increased, and performance converged after $c = 3$. In general, the result of $c = 1$ was closer to EGO; too much emphasis on optimization performance leads to a large loss of prediction performance. So, $c = 1$ is not a recommended value. $c > 2$ can maximize the optimization performance without excessive loss of prediction performance. When $c > 2$, the AUC-CC results can even outperform ALR ($c \geq 2$ on the Lasso model, $c \geq 3$ on the Enet model). The larger the c, the stronger the model prediction performance; the smaller the c, the stronger the model optimization performance. This result is in line with the respective sample selection characteristics of EGO and ALR. To find the balance between predicting and optimizing that makes the choice from the EGO-ALR algorithm more meaningful, $c = 2$ was chosen as the parameter of EGO-ALR.

4.8. Visualization of Sample Selection. This section explains the selection behavior of different approaches by the visualization of sample selection, to better visualize the advantages of EGO-ALR.

Taking the results of modeling a typical dataset (HEA) using Ridge regression as an example, the study compared EGO, EMCM, EGO-EMCM, GSx, and EGO-GSx, including supervised ALR and unsupervised ALR. Figure 6 shows the visualization of the sample selection results after iterating to the 20th round (the number of labeled samples is 25). The first 24 samples selected by an approach are marked in blue. The samples selected in the 20th round are marked in red. The black dotted line represents that the predicted value was equal to the actual value, and the red dotted line is the optimization goal of this experiment, that is, the minimum of Y.

Driven by the global optimization strategy, EGO collected samples with low Y and hardly selected samples with high Y, which caused the overall sampling of EGO to be biased. Most of the unlabeled samples are above the black dotted line, which indicates that the predicted value of unlabeled samples was significantly lower than the actual value.

EMCM-selected samples were distributed uniformly in the entire space. However, because the early sampling of EMCM was random and the subsequent sampling was small, the prediction of unlabeled samples in this experiment was also lower than the actual value. This situation would be improved after increasing the number of samples.

GSx-selected samples were more uniform than EMCM and EGO samples, so the prediction results were significantly better. This selection strategy caused it to seldom focus on a cluster for sampling, and it was difficult to find the best value.

The sampling of EGO-EMCM and EGO-GSx followed the original ALR while favoring lower Y clusters. EGO-EMCM and EGO-GSx not only ensured the prediction performance but also had more opportunities to select the best sample. It is further confirmed that the EGO-based ALR approach selects more reasonable samples than the original ALR, which results in better regression performances.

5. Conclusions

This study presents the EGO-ALR query strategy, which combines the ALR and EGO via weighted addition of normalization "information." EGO-ALR combines the benefits of the two original approaches, speeding up the process of optimizing samples while also establishing a high-precision regression model. EGO-ALR circumvents the complexities of sample labeling and the impact of model performance on the accuracy of subsequent results. Furthermore, depending on the demand, EGO-ALR can vary the search direction of the ideal value. The study used multiple ALR approaches and conducted extensive experiments with 19 datasets in different domains. The performance of the EGO-ALR was significantly better than the original ALR as evaluated by RMSE, CC, and opportunity cost. Specifically, EGO-ALR increased the opportunity cost by an average of 25.27% when the RMSE and CC values were not more than 1.07% different from the original ALR. Whether combined with supervised or unsupervised ALR, EGO-ALR had strong adaptability. In addition, the EGO-ALR evaluation results on Ridge are similar to those on Lasso and Enet regression models, demonstrating the stability of this approach in the linear regression model. To make the results of EGO-ALR meaningful, the value range of the parameter $c \geq 2$ is recommended.

As one of the future steps, EGO or more optimization algorithms can be combined with ALR approaches not mentioned in this report. Alternatively, the method of combined query strategy can be extended to a nonlinear regression model or classification problem. The single objective optimization problem in this paper can also be extended to multiobjective optimization problems.

Data Availability

The experimental data used to support the findings of the study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was funded by the University of Science and Technology Beijing (00012259). The authors thank AiMi Academic Services (http://www.aimieditor.com) for the English language editing and review services.

Supplementary Materials

Supplementary materials are the active learning experimental results of Lasso and Enet regression, which contains the normalized AUCs of all models on 19 datasets, and the AUCs percent improvements of each model over BL. These experiments are consistent with the experiments used Ridge regression in Figure 3 and Table 2 in the article. (*Supplementary Materials*)

References

[1] Z. Liu, X. Jiang, H. Luo, W. Fang, J. Liu, and D. Wu, “Pool-based unsupervised active learning for regression using iterative representativeness-diversity maximization (IRDM),” *Pattern Recognition Letters*, vol. 142, pp. 11–19, 2021.

[2] M. Grimm, K. Kroschel, and S. Narayanan, “The Vera Am Mittag German audio-visual emotional speech database,” in *Proceedings of the 2008 IEEE International Conference on Multimedia and Expo*, pp. 865–868, Hannover, German, April 2008.

[3] P. V. Balachandran, D. Xue, J. Theiler, J. Hogden, and T. Lookman, “Adaptive strategies for materials design using uncertainties,” *Scientific Reports*, vol. 6, no. 1, pp. 19660–19669, 2016.

[4] J. J. Cai, J. Tang, Q. G. Chen, Y. Hu, X. Wang, and S. J. Huang, “Multi-view active learning for video recommendation,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 2053–2059, Macao. P.R., China, August 2019.

[5] B. Settles, “Active learning literature survey,” *Science*, vol. 10, pp. 237–304, 1995.

[6] P. Kumar and A. Gupta, “Active learning query strategies for classification, regression, and clustering: a survey,” *Journal of Computer Science and Technology*, vol. 35, no. 4, pp. 913–945, 2020.

[7] T. Lookman, P. V. Balachandran, D. Xue, and R. Yuan, “Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design,” *Npj Computational Materials*, vol. 5, no. 1, pp. 21–17, 2019.

[8] D. Xue, P. V. Balachandran, J. Hogden, J. Theiler, D. Xue, and T. Lookman, “Accelerated search for materials with targeted properties by adaptive design,” *Nature Communications*, vol. 7, no. 1, pp. 11241–11249, 2016.

[9] R. Yuan, Z. Liu, P. V. Balachandran et al., “Accelerated discovery of large electrostrains in BaTiO 3 -based piezoelectrics using active learning,” *Advanced Materials*, vol. 30, no. 7, Article ID 1702884, 2018.

[10] C. Wen, Y. Zhang, C. Wang et al., “Machine learning assisted design of high entropy alloys with desired property,” *Acta Materialia*, vol. 170, pp. 109–117, 2019.

[11] M. Su, R. Grimes, S. Garg, D. Xue, and P. V. Balachandran, “Machine-learning-enabled prediction of adiabatic temperature change in lead-free BaTiO3-based electrocaloric c,” *ACS Applied Materials & Interfaces*, vol. 13, no. 45, pp. 53475–53484, 2021.

[12] D. R. Jones, M. Schonlau, and W. J. Welch, “Efficient global optimization of expensive black-box functions,” *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[13] P. Frazier, W. Powell, and S. Dayanik, “The Knowledge-Gradient Policy for correlated normal beliefs,” *INFORMS Journal on Computing*, vol. 21, no. 4, pp. 599–613, 2009.

[14] T. RayChaudhuri and L. G. Hamey, “Minimisation of data collection by active learning,” in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1338–1341, Perth, WA, Australia, December 1995.

[15] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, “Selective sampling using the query by committee algorithm,” *Machine Learning*, vol. 28, no. 2/3, pp. 133–168, 1997.

[16] S. Buus, S. L. Lauemøller, P. Worning et al., “Sensitive quantitative predictions of peptide-MHC binding by a ’Query by Committee’ artificial neural network approach,” *Tissue Antigens*, vol. 62, no. 5, pp. 378–384, 2003.

[17] R. Burbidge, J. J. Rowland, and R. D. King, “Active learning for regression based on query by committee,” in *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning*, pp. 209–218, Berlin, UK, December 2007.

[18] B. Demir and L. Bruzzone, “A multiple criteria active learning method for support vector regression,” *Pattern Recognition*, vol. 47, no. 7, pp. 2558–2567, 2014.

[19] J. Vandoni, E. Aldea, and S. Le Hégarat-Mascle, “Evidential query-by-committee active learning for pedestrian detection in high-density crowds,” *International Journal of Approximate Reasoning*, vol. 104, pp. 166–184, 2019.

[20] W. Cai, Y. Zhang, and J. Zhou, “Maximizing expected model change for active learning in regression,” in *Proceedings of the IEEE 13th International Conference on Data Mining*, pp. 51–60, Dallas, TX, USA, December 2013.

[21] W. Cai, Y. Zhang, S. Zhou, W. Wang, C. Ding, and X. Gu, “Active learning for support vector machines with maximum model change,” *Machine Learning and Knowledge Discovery in Databases*, vol. 8724, pp. 211–226, 2014.

[22] W. Cai, Y. Zhang, Y. Zhang et al., “Active learning for classification with maximum model change,” *ACM Transactions on Information Systems*, vol. 36, no. 2, pp. 1–28, 2018.

[23] W. Cai, M. Zhang, and Y. Zhang, “Batch mode active learning for regression with expected model change,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 7, pp. 1668–1681, 2017.

[24] A. Freytag, E. Rodner, and J. Denzler, “Selecting influential examples: active learning with expected model output changes,” in *Proceedings of the European Conference on Computer Vision, Computer Vision – ECCV 2014*, pp. 562–577, Springer, Zurich, Switzerland, September 2014.

[25] J. O’Neill, S. Jane Delany, and B. MacNamee, “Model-free and model-based active learning for regression,” in *Advances in Intelligent Systems and Computing*, pp. 375–386, Springer, Cham, 2017.

[26] S. H. Park and S. B. Kim, “Robust expected model change for active learning in regression,” *Applied Intelligence*, vol. 50, no. 2, pp. 296–313, 2020.

[27] H. Yu and S. Kim, “Passive sampling for regression,” in *Proceedings of the IEEE International Conference on Data Mining*, pp. 1151–1156, Sydney, Australia, December 2010.

[28] L. F. O. Chamon and A. Ribeiro, “Greedy sampling of graph signals,” *IEEE Transactions on Signal Processing*, vol. 66, no. 1, pp. 34–47, 2018.

[29] D. R. Wu, C. T. Lin, and J. Huang, “Active learning for regression using greedy sampling,” *Information Sciences*, vol. 474, pp. 90–105, 2019.

[30] D. Wu, “Pool-based sequential active learning for regression,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 5, pp. 1348–1359, 2019.

[31] Z. Liu and D. Wu, “Unsupervised pool-based active learning for linear regression,” 2020, http://export.arxiv.org/pdf/2001.05028.

[32] D. Ginsbourger, R. Le Riche, and L. Carraro, “Kriging is well-suited to parallelize optimization,” in *Computational Intelligence in Expensive Optimization Problems*, pp. 131–162, Springer, Berlin, Heidelberg, GER, 2010.

[33] A. Chaudhuri and R. T. Haftka, “Efficient global optimization with adaptive target setting,” *AIAA Journal*, vol. 52, no. 7, pp. 1573–1578, 2014.

[34] J. Mockus, V. Tiesis, and A. Zilinskas, “The application of Bayesian methods for seeking the extremum,” *Towards Global Optimization*, vol. 2, pp. 117–129, 1978.

[35] W. B. Powell and I. O. Ryzhov, “Optimal learning,” in *Wiley Series in Probability and Statistics*John Wiley & Sons, Hoboken, USA, 2012.

[36] A. E. Hoerl and R. W. Kennard, “Ridge regression: applications to nonorthogonal problems,” *Technometrics*, vol. 12, no. 1, pp. 69–82, 1970.

[37] R. Tibshirani, “Regression shrinkage and selection via the lasso: a retrospective,” *Journal of the Royal Statistical Society: Series B*, vol. 73, no. 3, pp. 273–282, 2011.

[38] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *Journal of the Royal Statistical Society: Series B*, vol. 67, no. 2, pp. 301–320, 2005.

[39] R. L. Iman and J. M. Davenport, “Approximations of the critical region of the fbietkan statistic,” *Communications in Statistics - Theory and Methods*, vol. 9, no. 6, pp. 571–595, 1980.

[40] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.