2013 IEEE 13th International Conference on Data Mining

Maximizing Expected Model Change for Active Learning in Regression

Wenbin Cai Ya Zhang* Jun Zhou
Shanghai Key Laboratory of Multimedia Processing and Transmissions
Shanghai Jiao Tong University, Shanghai, China
E-mail: {cai-wenbin, ya_zhang, zhoujun}@sjtu.edu.cn

***Abstract*—Active learning is well-motivated in many supervised learning tasks where unlabeled data may be abundant but labeled examples are expensive to obtain. The goal of active learning is to maximize the performance of a learning model using as few labeled training data as possible, thereby minimizing the cost of data annotation. So far, there is still very limited work on active learning for regression. In this paper, we propose a new active learning framework for regression called Expected Model Change Maximization (EMCM), which aims to choose the examples that lead to the largest change to the current model. The model change is measured as the difference between the current model parameters and the updated parameters after training with the enlarged training set. Inspired by the Stochastic Gradient Descent (SGD) update rule, the change is estimated as the gradient of the loss with respect to a candidate example for active learning. Under this framework, we derive novel active learning algorithms for both linear regression and nonlinear regression to select the most informative examples. Extensive experimental results on the benchmark data sets from UCI machine learning repository have demonstrated that the proposed algorithms are highly effective in choosing the most informative examples and robust to various types of data distributions.**

***Keywords*-Active learning, Linear Regression, Nonlinear regression, Expected Model Change Maximization**

I. Introduction

Data collection and annotation is a fundamental problem in data mining and machine learning. The widely used method for data collection is called passive learning, where training examples are randomly selected from the underlying distribution and manually annotated by human editors. However, due to the expensive cost associated with the above data collection process, it is always the case that there are not enough data examples to train a high quality model. To reduce the cost of data annotation, active learning aims to choose the most informative examples that maximize the accuracy of the model trained if they are labeled and added to the training set. A typical active learning process can be briefly described as follows: 1) Generate a base model from a small initial training set. 2) Select examples with a sampling function from a large set of unlabeled data and label them. 3) Add the newly labeled examples to the training set and update the model. This sampling process is repeated until a certain performance expectation is met or a certain labeling budget is used up. Active learning is well-motivated in many supervised learning tasks where unlabeled data may be abundant but labeled data points are expensive to obtain.

Active learning has been extensively studied for classification problems. One of the most widely used approaches is uncertainty sampling which aims to choose the examples whose labels the current classifier is most uncertain about [1], [3], [8]. This active learning strategy is usually straightforward to implement for probabilistic models. Take binary classification for example, uncertainty sampling is to choose the examples whose posterior probabilities are nearest 0.5 [1]. For multi-class classification problems, margin-based uncertainty sampling is a popular method which aims to select the examples that have the smallest margin between the first and second most probable class labels [3]. For non-probabilistic learning models such as Support Vector Machines (SVMs), this strategy selects the examples which are closest to the separating hyperplane [8]. Compared to active learning for classification, it is non-trivial to identify the uncertain examples in regression tasks. First, the output for regression is a continuous value rather than a set of class posterior probabilities, making the margin-based sampling strategy unsuitable. Second, there is no notion of distance in regression tasks, and hence the distance-based sampling method is not applicable.

So far, there is still very limited work on active learning for regression [7], [12], [13], [14], [10], [17]. Therefore, a general active learning framework for regression is of great need. This paper considers the capacity of examples to change the current model, and attempts to tackle the problem of active learning for regression by maximizing the model change. Previous work with similar idea have been studied for classification [11] and ranking [15] tasks. In [11], Settles et al. introduce a theoretical active learning strategy, named Expected Gradient Length (EGL), which aims to query the examples that would maximally change the current model. The model change is estimated as the length of the updated gradient of the objective function with respect to the model parameters, which is obtained by the accumulated training set. In [15], the change is measured as the difference between the current model and the additional model trained with the selected data examples.

We propose a new active learning framework for regres-

*Corresponding author: Ya Zhang, E-mail: ya_zhang@sjtu.edu.cn.

1550-4786/13 $31.00 © 2013 IEEE
DOI 10.1109/ICDM.2013.104

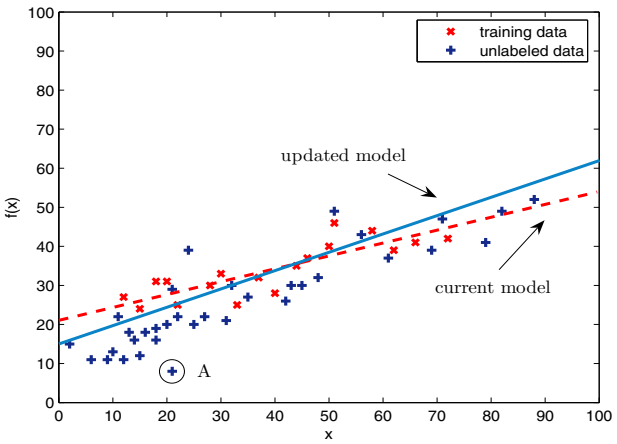


Figure 1: An illustrative example for EMCM algorithm in linear regression. The red cross points denote the training data set, and the blue plus points represent the unlabeled data set. The dotted red line is the current linear regression model. After choosing the example A leading to the greatest change in the current model, the model is updated with the accumulated data (denoted as the solid blue line).

sion called Expected Model Change Maximization (EMCM), which measures the change as the difference between the current model parameters and the new parameters trained with the enlarged training set. Inspired by the Stochastic Gradient Descent (SGD) update rule, where the model parameters are updated repeatedly using the gradient of the loss with respect to each single training example, we use the gradient of the error with respect to a candidate example to estimate the model change. Under this framework, we first derive a novel active sampling algorithm for linear regression to choose the examples that would maximally change the current model, where the change is straightforward to calculate according to the gradient. For nonlinear regression, we choose the Gradient Boosting Decision Tree (GBDT), a well-known nonlinear regression model, as the base learner in this study. Compared to linear regression, GBDT has a more complicated model form, and it is difficult to measure the model change using the gradient directly. To solve this problem, we generate super features from trees using feature mapping and represent each unlabeled example with super features, so that the GBDT model could be approximated as a linear regression model. We then propose an active learning algorithm to choose the examples which lead to the largest change in the model with respect to the super features. Extensive experimental results on benchmark data sets from UCI machine learning repository have demonstrated that the proposed active learning algorithms are highly effective in selecting the most informative examples and robust to various types of data distributions.

Figure 1 presents an illustrative example to explain the EMCM algorithm for linear regression. The red cross points denote the training data set, and the blue plus points represent the unlabeled data set (denoted as pool). As shown in the figure, we observe that the proposed EMCM algorithm chooses the example A for annotation to maximally change the current model. The regression model then is updated with the enlarged training set and achieves a higher accuracy.

The main contributions of this paper are summarized as follows.

- We propose a new active learning framework for regression , called Expected Model Change Maximization (EMCM), which chooses the examples that result in the greatest change to the current model.
- Under this framework, We derive a novel active learning algorithm for linear regression.
- We approximate the Gradient Boosting Decision Tree (GBDT), a well-known nonlinear regression model, as a linear regression model using feature mapping, and derive an active learning algorithm.

The rest of this paper is organized as follows: Section II briefly reviews the related work. Section III introduces the general framework of Expected Model Change Maximization (EMCM). The proposed active learning for regression algorithms, including linear and nonlinear regression, are presented in Section IV. Section V presents the experiments and interprets the results. Finally, we conclude the paper and propose for future directions in Section VI.

II. Related Work

In this section, We briefly review the related work. A comprehensive active learning survey can be found in [5].

A. Active Learning for Classification

So far, a number of strategies for active learning have been proposed for classification problems. One common strategy is called uncertainty sampling [1], [3], [8], which aims to query the examples about which the current model is least certain how to label. This strategy is usually straightforward for probabilistic models using entropy to measure the uncertainty [3]. For the non-probabilistic models such as Support Vector Machines (SVMs), this strategy selects the example which is close to the separating boundary [8].

Query by committee (QBC) [2] is another typical active learning strategy. The QBC strategy generates a committee of models and selects the unlabeled example about which the committee members disagree the most. A popular function to quantify the disagreement is vote entropy. To efficiently generate the committee, ensemble learning algorithms, such as Bagging and Boosting, have been employed [4].

Another decision-theoretic active learning strategy aims to minimize the generalization error of the model trained. Roy et al. [6] proposed an optimal active sampling method to choose the examples that lead to the lowest generalization error on the test set once labeled. The weakness is that the

computational cost of this method is extremely high. Instead of choosing examples yielding the smallest generalization error, Nguyen et al. [18] suggested to select the examples having the largest contribution to the current error.

B. Active Learning for Regression

Compare to active learning for classification, there is still very limited work on active learning for regression.

Castro et al. [12] analyzed active learning in the regression setting with a certain noise rate. Sugiyama [13] proposed a theoretical active learning approach based on generalization error minimization under the assumption that the distribution of test input points is known and the training examples are allowed to locate at any desired position, and denoted it as population-based active learning. Sugiyama et al. [14] then extend the population-based active learning to the pool-based active learning where the examples are chosen from a given pool set. The base learner used is an additive regression model, and the parameters are learned by importance-weighted least-squares minimization. Cohn et al. [7] proposed a statistically optimal active learning approach, which aims to choose the examples minimizing the output variance to reduce the generalization error. They applied the approach to various types of models including locally weighted regression. Freund et al. [2] argued that the QBC framework could be applied when the outputs were not binary or even discrete. This is related to the variance-based QBC for regression [17]. Yu et al. [10] proposed passive sampling heuristics for regression based on the geometric characteristics in feature space.

III. The Framework of Expected Model Change Maximization

In this section, we first introduce the general framework of Expected Model Change Maximization (EMCM). Then we provide a short interpretation to better motivate the proposed EMCM framework.

A. The EMCM Framework

In supervised learning problems, the objective is to learn a model f that minimizes the generalization error on the unseen data:

$$\epsilon = \int_{\mathcal{X}\times\mathcal{Y}} \ell[f(x), y(x)]dP(x, y), \tag{1}$$

where $y(x)$ is the true label of x, and $f(x)$ is the predicted label. $\ell[f(x), y(x)]$ is a given loss function. Because the joint distribution $P(x, y)$ is usually not known, formula (1) can not be directly solved. In practice, we are given a training set $\mathcal{D} = \{(x_i, y_i), x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}_{i=1}^n$ drawn i.i.d. from $P(x, y)$, i.e. $\mathcal{D} \sim P(x, y)$, and construct the model that minimizes the empirical error:

$$\hat{\epsilon}_{\mathcal{D}} = \sum_{i=1}^{n} \ell[f(x_i), y_i]. \tag{2}$$

Suppose the model is parameterized by θ. To find θ minimizing the empirical error, a widely used search approach is called Stochastic Gradient Descent (SGD), where the parameters θ are repeatedly updated according to the negative gradient decent of the loss $\ell(\theta)$ with respect to each training example (x_i, y_i):

$$\theta := \theta - \alpha \frac{\partial \ell_{x_i}(\theta)}{\partial \theta} \qquad i = 1, 2, ..., n, \tag{3}$$

where α is called the learning rate.

Here we consider the update rule in active learning cases. If a candidate example x^+ is added to the training set with a given label y^+, the empirical risk on the enlarged training set $\mathcal{D}^+ = \mathcal{D} \cup (x^+, y^+)$ then becomes:

$$\hat{\epsilon}_{\mathcal{D}^+} = \sum_{i=1}^{n} \ell[f(x_i), y_i] + \underbrace{\ell_{x^+}[f(x^+), y^+]}_{:=\ell_{x^+}(\theta)}. \tag{4}$$

Thus, the model obtained by minimizing the empirical risk is changed due to the inclusion of the new example (x^+, y^+). We measure the model change $\mathcal{C}(x^+)$ as the parameter change using the gradient of the loss at the example x^+:

$$\mathcal{C}(x^+) = \alpha \frac{\partial \ell_{x^+}(\theta)}{\partial \theta}. \tag{5}$$

The goal of the sampling strategy is to choose the example x^* that could maximally change the current model, and the selection function can be formulated as:

$$x^* = \underset{x\in\text{pool}}{\operatorname{argmax}} ||\mathcal{C}(x)||. \tag{6}$$

In practice, we do not know the true label of the data point x^+ in advance. Therefore, we are not able to estimate the model change in Eq.(6) directly. Instead, we use the expected change over all possible labels $y^+ \in \{y_1, y_2, ..., y_K\}$ to approximate the true change. Suppose the learning rate α for each candidate example is identical, the EMCM for active learning strategy is represented as:

$$x^* = \underset{x\in\text{pool}}{\operatorname{argmax}} \sum_{k=1}^{K} P(y_k|x)||\frac{\partial \ell_x(\theta)}{\partial \theta}||, \tag{7}$$

where $P(y_k|x)$ is the conditional probability of label y_k given data example x estimated by the current model.

B. Interpretation

In this section, we empirically show the link between the model change and the generalization error reduction to better motivate the proposed EMCM framework. We aim to answer the following question: does the sampling strategy lead to a better generalization performance?

The answer is described as follows: First, the generalization error could be changed if and only if the model is changed. In other words, the example that can not update the current model is useless for active learning. Second, a big model change may not always result in a good generalization

performance because it may choose an outlier. However, in the active learning task, we repeatedly choose the unlabeled examples from a pool set. If the model is changed due to the outlier, this sampling strategy will certainly choose a good example that can maximize the change again in the next data selection round, so that the negative effect of the outlier will be relieved. Because the number of outliers is usually very small in practice, we expect that the proposed sampling strategy will lead to a good generalization ability with more data sampled.

IV. EMCM for Active Learning in Regression

In this section, we first briefly introduce the linear and nonlinear regression models used as the base learners in this study. Then, the details of the active learning algorithms are provided respectively.

A. Regression Models

1) Linear Regression: A linear regression model assumes that the regression function is linear with regards to the example features, and the model form can be formulated as:

$$f(x;\theta) = \sum_{i=0}^{p} \theta_i x_{(i)} = \theta^{\mathrm{T}} x, \tag{8}$$

where $x_{(0)} = 1$ is the intercept term, and $x_{(i)}, i = 1, 2, ..., p$ are the features of example x. The linear model is parameterized by the weight vector θ.

2) Nonlinear Regression: In this study, we choose Gradient Boost Decision Tree (GBDT) as the learner for nonlinear regression. GBDT can be expressed as an additive model:

$$f(x;\{\lambda,\Theta\}_1^M) = \sum_{m=1}^{M} \lambda_m h_m(x;\Theta_m), \tag{9}$$

where $\{\lambda,\Theta\}_1^M$ parameterize the model. Each base tree $h_m(x;\Theta_m)$ is a J-terminal node regression tree:

$$h(x;\{\gamma,R\}_1^J) = \sum_{j=1}^{J} \gamma_j \mathbf{1}(x \in R_j), \tag{10}$$

where $\{\gamma\}_1^J$ are coefficients, $\{R\}_1^J$ are the regions partitioned by the decision tree, and $\mathbf{1}(\cdot)$ is the indicator function of the region association. More details about GBDT can be found in [9].

B. EMCM for Active Learning in Linear Regression

The objective of regression is to learn a function $\hat{f} : \mathcal{X} \to \mathcal{Y}$ that minimizes a given loss. Given the training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n, x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$, in the case of squared-error loss, the empirical risk can be represented as:

$$\hat{\epsilon}_{\mathcal{D}} = \frac{1}{2}\sum_{i=1}^{n}(f(x_i) - y_i)^2. \tag{11}$$

Assume a candidate example x^+ with the label y^+ is added to the training set. The empirical risk on the expanded training set $\mathcal{D}^+ = \mathcal{D} \cup (x^+, y^+)$ then becomes:

$$\hat{\epsilon}_{\mathcal{D}^+} = \frac{1}{2}\sum_{i=1}^{n}(f(x_i) - y_i)^2 + \underbrace{\frac{1}{2}(f(x^+) - y^+)^2}_{:=\ell_{x^+}(\theta)}. \tag{12}$$

The derivative of the squared-error loss $\ell_{x^+}(\theta)$ with respect to the parameters θ at x^+ is formulated as:

$$\begin{aligned}\frac{\partial \ell_{x^+}(\theta)}{\partial \theta} &= (f(x^+) - y^+)\frac{\partial f(x^+)}{\partial \theta} \\ &= (f(x^+) - y^+)\frac{\partial \theta^{\mathrm{T}} x^+}{\partial \theta} \\ &= (f(x^+) - y^+)x^+ \end{aligned} \tag{13}$$

Since the true label y^+ is actually unknown before querying in practice, we utilize bootstrap to construct an ensemble $\mathcal{B}(K) = \{f_1, f_2, ..., f_K\}$ to estimate the prediction distribution $\{y_1, y_2, ..., y_K\}$, and use the expected model change to approximate the true model change. The connection between bootstrap and prediction distribution has been investigated in previous work [16]. Therefore, the sampling function for linear regression can be formulated as:

$$x^* = \underset{x \in \text{pool}}{\operatorname{argmax}} \frac{1}{K}\sum_{k=1}^{K} ||(f(x) - y_k(x))x||. \tag{14}$$

The pseudo-code for EMCM for linear regression is shown in Algorithm 1.

Algorithm 1 EMCM for active learning in linear regression

Input: the small labeled data set $\mathcal{D}$=$\{(x_i, y_i)\}_{i=1}^n$, the unlabeled pool set, the linear regression model $f(x;\theta)$ trained with the labeled data set.

```
Construct an ensemble with bootstrap examples:
  B(K) = {f_1, f_2, ..., f_K}
for each x in pool set do
    for k ← 1 to K do
        y_k(x) ← f_k(x)
        Calculate the derivative using Eq.(13):
        ∇_θ ℓ_k(θ) ← (f(x) − y_k(x))x
    end for
    Estimate the true model change by expectation calculation over K possible labels using Eq.(14).
end for
```

Output: Select the x^* having the largest expected change.

C. EMCM for Active Learning in Nonlinear Regression

GBDT is characterized by parameters $\{\lambda, \{\gamma, R\}_1^J\}_1^M$, where $\{\gamma, R\}_1^J$ are the parameters of each base function, a J-terminal node regression tree. As tree models are not smooth and the parameters are not derivable, we can not directly

Algorithm 2 EMCM for active learning in GBDT regression

Input: the small labeled data set $\mathcal{D}$=$\{(x_i, y_i)\}_{i=1}^n$, the unlabeled pool set, the GBDT regression model $f(x;\lambda)$ trained with the labeled data set.

```
Construct an ensemble with bootstrap examples: B(K) = {f_1, f_2, ..., f_K}
for each x in pool set do
    Generate super features from trees: φ(x) = [h_1(x), h_2(x), ..., h_M(x)]^T
    for k ← 1 to K do
        y_k(x) ← f_k(x)
        Calculate the derivative using Eq.(17): ∇_θ ℓ_k(θ) ← (f(x) − y_k(x))φ(x)
    end for
    Estimate the true model change by expectation calculation over K possible labels using Eq.(18).
end for
```

Output: Select the x^* having the largest expected change.

estimate the model change using the gradient. To solve this problem, We assume that adding a single example to the training set does not change the structure of the tree. For a regression tree, the predictive rule is: $x \in R_j \Rightarrow h(x) = \gamma_j$. Because the number of examples chosen in each sampling round is relatively small compared to the size of current training data, in most cases the region where x is located is not changed. Thus, the assumption is reasonable. Under this assumption, we employ the concept of super features based on individual trees and map each unlabeled example to the super features:

$$\phi(x) = [h_1(x), h_2(x), ..., h_M(x)]^{\mathrm{T}}. \tag{15}$$

With the super features, the GBDT model can be approximated as a linear regression model:

$$f(x; \{\lambda\}_1^M) = \sum_{m=1}^{M} \lambda_m h_m(x) = \lambda^{\mathrm{T}} \phi(x). \tag{16}$$

Here, we assume that the parameters of each regression tree, i.e. $\{\gamma, R\}_1^J$, remain unchanged and focus on the change in parameters $\{\lambda\}_1^M$ for active learning.

The derivative of the squared-error $\ell_{x^+}(\lambda)$ with respect to the parameters $\lambda = \{\lambda_1, \lambda_2, ..., \lambda_M\}$ at the candidate data point (x^+, y^+) is:

$$\begin{aligned} \frac{\partial \ell_{x^+}(\lambda)}{\partial \lambda} &= (f(x^+) - y^+) \frac{\partial f(x^+)}{\partial \lambda} \\ &= (f(x^+) - y^+) \frac{\partial \lambda^{\mathrm{T}} \phi(x^+)}{\partial \lambda} \\ &= (f(x^+) - y^+) \phi(x^+). \end{aligned} \tag{17}$$

Similar to active learning for linear regression, we create an ensemble $\mathcal{B}(K) = \{f_1, f_2, ..., f_K\}$ on bootstrap examples to estimate the prediction distribution $\{y_1, y_2, ..., y_K\}$, and approximate the true change using its expectation. The final sampling criteria for GBDT can be expressed as:

$$x^* = \underset{x \in \text{pool}}{\operatorname{argmax}} \frac{1}{K} \sum_{k=1}^{K} ||(f(x) - y_k(x))\phi(x)||. \tag{18}$$

The corresponding pseudo-code is given in Algorithm 2.

Table I: The information of the five regression data sets from UCI repository.

Data set		# Examples	# Features
Concrete		1030	8
Housing		506	13
Wine	Redwine	1599	11
	Whitewine	4898	11
Yacht		308	6

Table II: The statistics of the five benchmark regression data sets in the active learning setting.

Data set	# Ex. in $\mathcal{L}$	# Ex. in $\mathcal{U}$	# Ex. in $\mathcal{T}$
Concrete-1	100 (10%)	730 (70%)	200 (20%)
Concrete-2	200 (20%)	630 (60%)	200 (20%)
Concrete-3	300 (30%)	530 (50%)	200 (20%)
Housing-1	50 (10%)	356 (70%)	100 (20%)
Housing-2	100 (20%)	306 (60%)	100 (20%)
Housing-3	150 (30%)	256 (50%)	100 (20%)
Redwine-1	160 (10%)	1119 (70%)	320 (20%)
Redwine-2	320 (20%)	959 (60%)	320 (20%)
Redwine-3	480 (30%)	799 (50%)	320 (20%)
Whitewine-1	490 (10%)	3428 (70%)	980 (20%)
Whitewine-2	980 (20%)	2938 (60%)	980 (20%)
Whitewine-3	1470 (30%)	2448 (50%)	980 (20%)
Yacht-1	30 (10%)	218 (70%)	60 (20%)
Yacht-2	60 (20%)	188 (60%)	60 (20%)
Yacht-3	90 (30%)	158 (50%)	60 (20%)

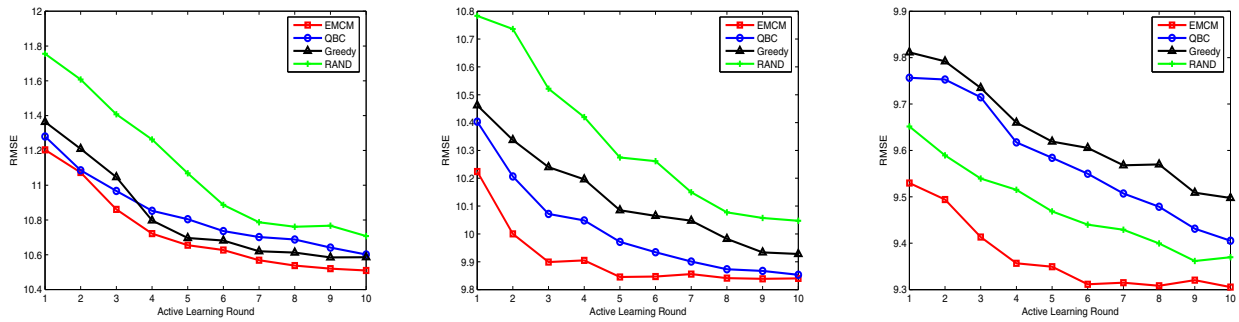
V. Experiments

A. Data Set and Experimental Settings

To validate the performance of the proposed active learning algorithms, we use five benchmark data sets of various sizes from UCI machine learning repository[1]: **Concrete**, **Housing**, **Wine** (The **Wine** dataset consists of two collections: **Redwine** and **Whitewine**), and **Yacht**. These five real-world data sets are released from various types of domains and have been widely used for testing regression algorithms. Table I shows the information of the five regression data sets.

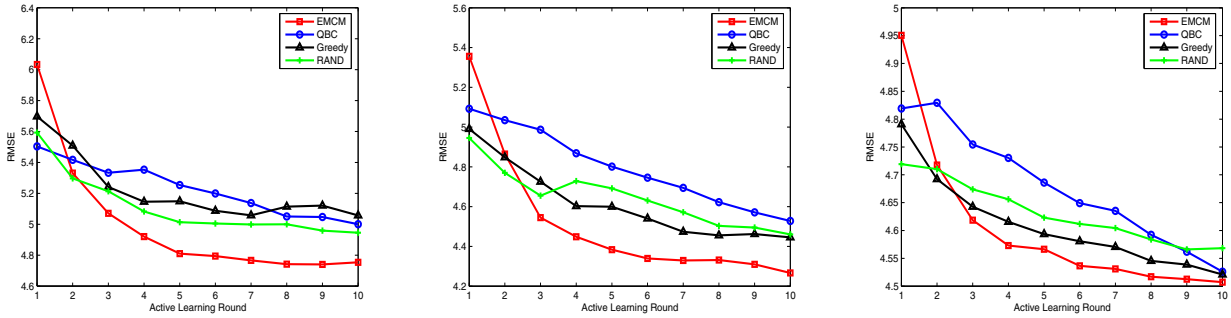
Each data set is randomly split into three disjoint subsets: the base labeled training set (denoted as $\mathcal{L}$), the unlabeled pool set (denoted as $\mathcal{U}$), and the test set (denoted as $\mathcal{T}$). We use the base labeled training set $\mathcal{L}$ as the small labeled data

[1]http://archive.ics.uci.edu/ml/



(a) Concrete-1 (b) Concrete-2 (c) Concrete-3

Figure 2: Comparison results for the linear regression model on the **Concrete** data set in terms of RMSE.



(a) housing-1 (b) housing-2 (c) housing-3

Figure 3: Comparison results for the linear regression model on the **Housing** data set in terms of RMSE.

set to train the initial regression models. The pool set $\mathcal{U}$ is used as a large size unlabeled data set to select the most informative examples, and the test set $\mathcal{T}$ is used to evaluate different active learning algorithms. To test the robustness of the proposed algorithms, we independently construct three active learning scenarios for each data set to simulate different types of data distributions: $\mathcal{L}(10\%)$+$\mathcal{U}(70\%)$+$\mathcal{T}(20\%)$, $\mathcal{L}(20\%)$+$\mathcal{U}(60\%)$+$\mathcal{T}(20\%)$, and $\mathcal{L}(30\%)$+$\mathcal{U}(50\%)$+$\mathcal{T}(20\%)$. Table II provides the statistics of the five benchmark data sets in the active learning setting. We normalize the features with the function below:

$$f^N_{(i,j)} = \frac{f_{(i,j)} - min\{f_{(i,j)}; i \in n\}}{max\{f_{(i,j)}; i \in n\} - min\{f_{(i,j)}; i \in n\}}, \quad (19)$$

where n denotes the number of examples in the data set, and $f_{(i,j)}$ represents the j-th feature from the i-th example.

Two regression models are used as the base learners: the linear regression model and the GBDT model. The size of the ensemble K is empirically set to be 4. In this study, the active learning process iterates 10 rounds. In each round of active selection, 3% of the whole examples are chosen from $\mathcal{U}$ and labeled. These examples are then added to the training set, and the regression models are re-trained and tested on the separate test set $\mathcal{T}$.

B. Comparison Methods and Evaluation Metric

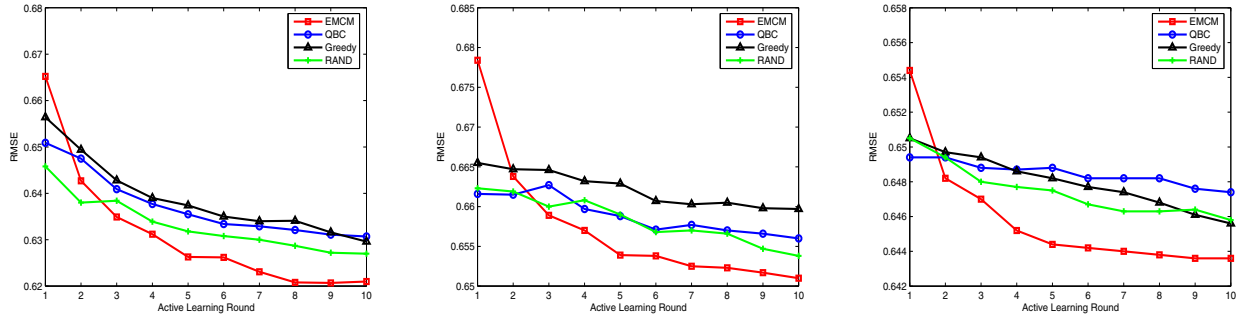
In order to test the effectiveness of the proposed active learning algorithms, our algorithms are compared with the following active learning for regression approaches:

- QBC [2], [17]: The QBC algorithm for regression is to choose the example that has the highest variance among the members' prediction. The committee is constructed on bootstrap examples with Bagging type which refers to Query-by-Bagging in this work.
- Greedy [10]: The Greedy algorithm is to select the new example which has the largest minimum distance from the labeled set in feature space.
- Random: The random selection, which is widely used in practice, represents a baseline (denoted as RAND).

For evaluation, a widely adopted metric for regression, Root Mean Squared Error (RMSE), is used to measure the performance of the regression models on the test set:

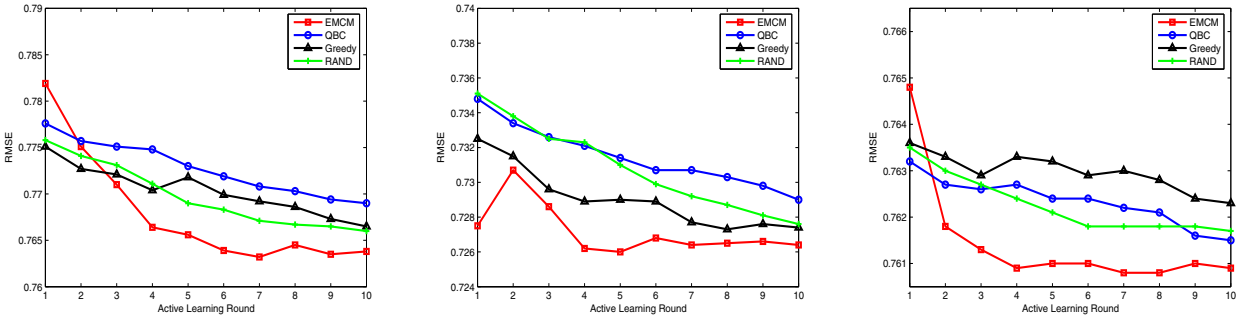
$$\text{RMSE} = \sqrt{\frac{1}{|\text{T}|}\sum_{i=1}^{|\text{T}|}(f(x_i) - y_i)^2}, \quad (20)$$

where $|\text{T}|$ denotes the size of the test set, y_i is the ground truth of the example x_i, and $f(x_i)$ is the prediction. To



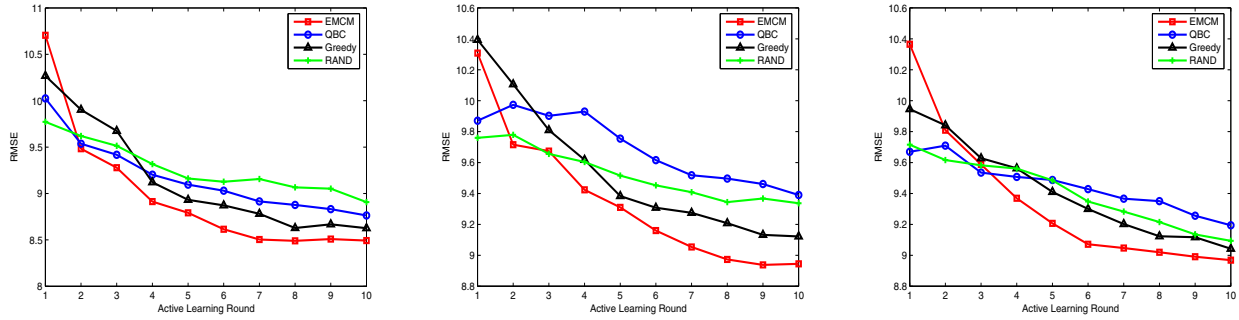
(a) Redwine-1 (b) Redwine-2 (c) Redwine-3

Figure 4: Comparison results for the linear regression model on the **Redwine** data set in terms of RMSE.



(a) whitewine-1 (b) whitewine-2 (c) whitewine-3

Figure 5: Comparison results for the linear regression model on the **Whitewine** data set in terms of RMSE.



(a) yacht-1 (b) yacht-2 (c) yacht-3

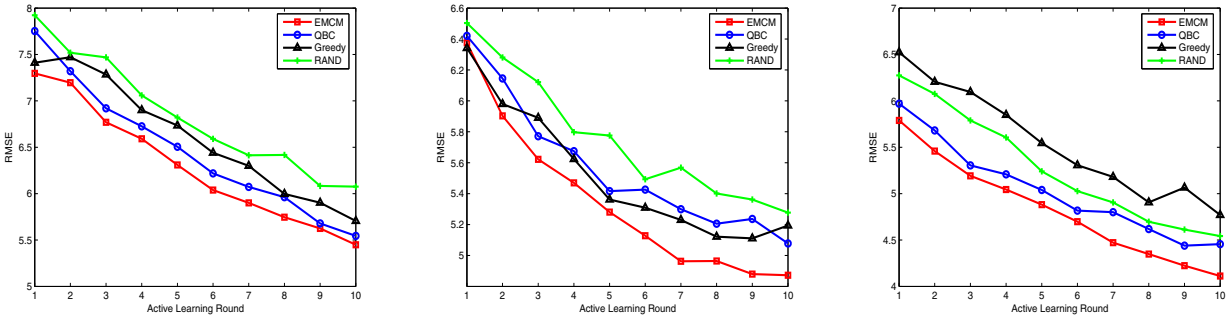
Figure 6: Comparison results for the linear regression model on the **Yacht** data set in terms of RMSE.

avoid random fluctuation, each experiment is repeated 10 times and the averaged RMSE scores are reported.

C. Comparison Results and Discussion

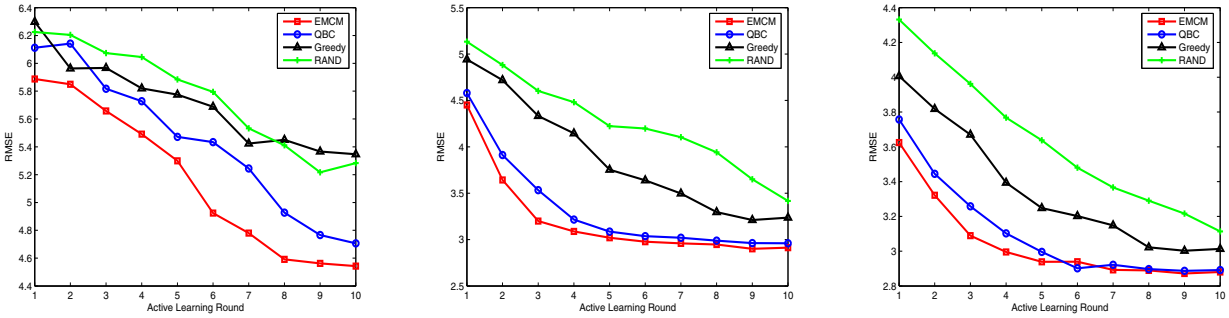
In this subsection, we first present and discuss the experimental results for linear regression. Then, the comparison results for nonlinear regression with GBDT are provided.

1) Active Learning for Linear Regression: The results of the four sampling algorithms on the **Concrete** data set are shown in Figure 2. The X-axis represents the number of iterations for the active learning process, and the Y-axis denotes the value of RMSE. For all four active learning algorithms, the RMSE decreases when the number of training examples increases, which agrees with the intuition that model quality is positively correlated with the size of training sets. The proposed EMCM method is observed to perform the best among the four methods. A possible explanation is that the EMCM algorithm estimates the model change as the gradient of the squared-error loss, which is directly related to the objective function RMSE used to evaluate the models.



(a) Concrete-1 (b) Concrete-2 (c) Concrete-3

Figure 7: Comparison results for the GBDT model on the **Concrete** data set in terms of RMSE.



(a) Housing-1 (b) Housing-2 (c) Housing-3

Figure 8: Comparison results for the GBDT model on the **Housing** data set in terms of RMSE.

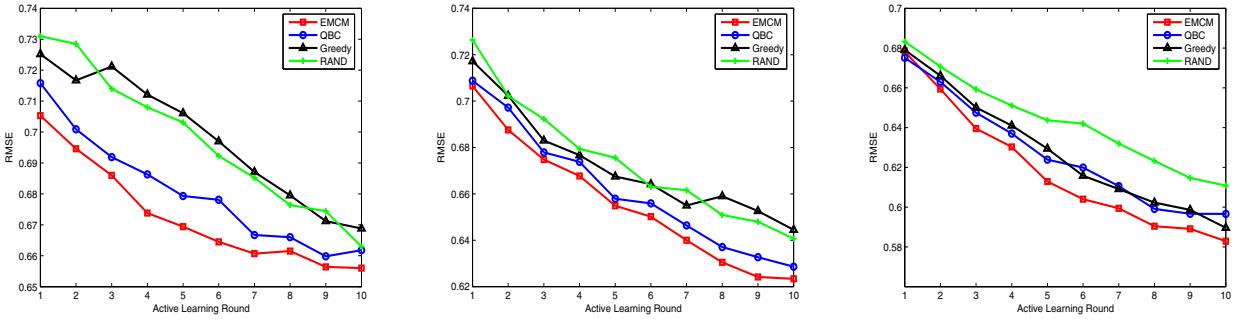
Hence, the examples chosen by the EMCM method are more likely to contribute positively to improve the regression model. Moreover, for all of the three types of data distributions (denoted as **Concrete-1**, **Concrete-2**, and **Concrete-3**), the EMCM approach consistently outperforms the other three sampling algorithms (i.e. QBC, Greedy and RAND), indicating that the EMCM algorithm is robust to various types of data distributions.

Figure 3 shows the comparison on the **Housing** data set. The proposed EMCM algorithm performs slightly worse than the other three sampling algorithms at the very beginning of the active learning process. The performance of EMCM algorithm increases quickly as the active learning proceeds and after three iterations, EMCM starts to outperform the other three methods. A possible explanation for the phenomena may be as follows. At the initial phase of active selection, the EMCM algorithm may choose some outliers in maximizing the change to the current model, which lead to a decrease in the performance. As discussed in Section III-B, to maximize the change in the latter round of active learning, the EMCM algorithm will then select the good examples so that the negative effect of the outliers is offset. Because the number of outliers is usually very small, the EMCM algorithm could perform well with more data examples sampled. Since we are mainly concerned with the final quality of the model trained in practice, the EMCM algorithm is still very promising in real-world applications.

Figure 4 and Figure 5 illustrate the learning curves of the four sampling methods on the **Redwine** and **Whitewine** data set respectively. As shown in the figures, the EMCM algorithm converges much faster than the other three sampling methods, i.e, the lowest RMSE score is achieved with much less examples added to the training set, demonstrating that the EMCM method is more effective in selecting the most informative examples. Similar results are obtained when comparing the four sampling algorithms on the **Yacht** data set (as shown in Figure 6).

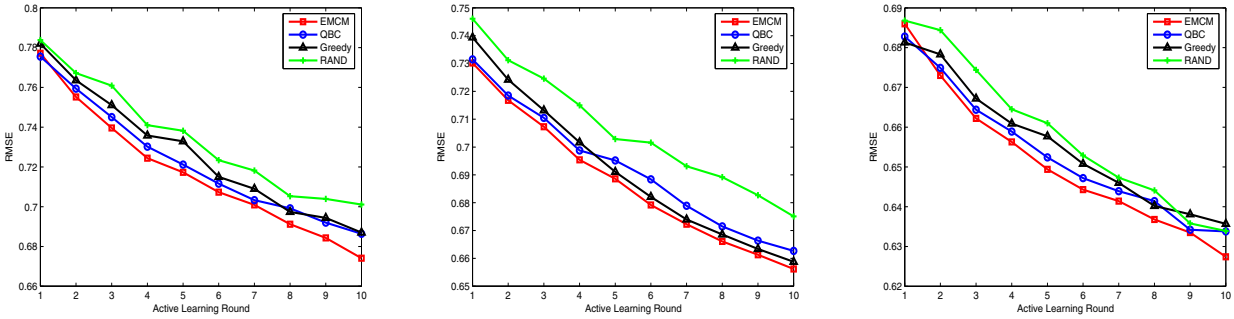
Significance test is performed on the comparisons, and T-test shows that the EMCM algorithm is statistically better ($p<0.05$) than the other three methods in most cases.

2) Active Learning for Nonlinear Regression: Figure 7 compares the experimental results of the four active learning algorithms for GBDT on the **Concrete** data set. As we can see, the EMCM method performs the best. The QBC algorithm performs better than the other two sampling algorithms most of the times, and the Greedy method achieves lower RMSE score than RAND for both the **Concrete**-1 and **Concrete**-2 date sets. For the third data set, **Concrete**-3,



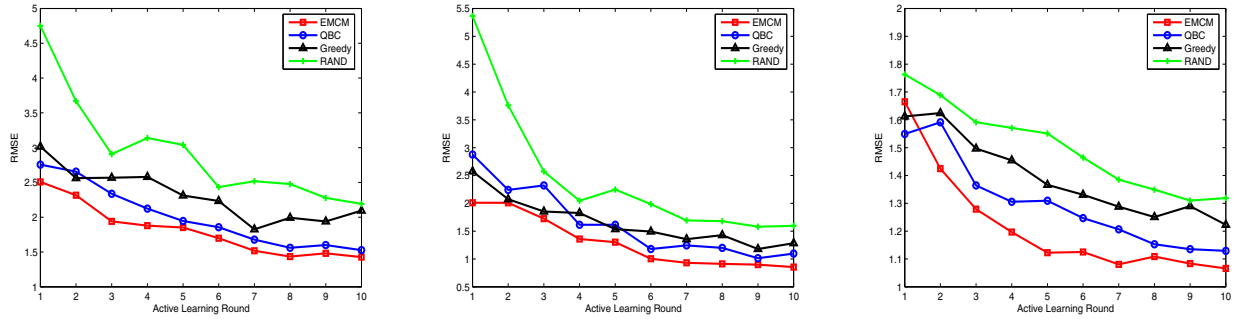
(a) Redwine-1 (b) Redwine-2 (c) Redwine-3

Figure 9: Comparison results for the GBDT model on the **Redwine** data set in terms of RMSE.



(a) Whitewine-1 (b) Whitewine-2 (c) Whitewine-3

Figure 10: Comparison results for the GBDT model on the **Whitewine** data set in terms of RMSE.



(a) Yacht-1 (b) Yacht-2 (c) Yacht-3

Figure 11: Comparison results for the GBDT model on the **Yacht** data set in terms of RMSE.

the Greedy strategy performs worse than the weak RAND baseline. The experimental results have demonstrated that the EMCM approach is effective in sample selection and robust for the nonlinear regression model.

Figure 8 illustrates the comparison results for the GBDT model on the **Housing** data set. We observe that the EMCM algorithm performs significantly better than the other three sampling approaches in **Housing**-1. For the **Housing**-2 and **Housing**-3 cases, the proposed EMCM algorithm slightly outperforms the QBC algorithm, and significantly outperforms the other two approaches. In addition, compared to the results for linear regression where the performance of the model is hurt noticeably at the initial stage of the sampling process (shown in Figure 3), possibly due to the inclusion of outliers, the EMCM algorithm consistently outperforms the other three methods during the entire data selection process. A possible explanation for the results is that the GBDT-based nonlinear regression model is more robust to outliers than the linear regression model.

The experimental results of the four active sampling

algorithms on the **Redwine** and **Whitewine** data sets are shown in Figure 9 and Figure 10, respectively. Among the four sampling methods, the EMCM algorithm consistently performs the best. Figure 11 compares the four methods on the **Yacht** data set, and similar results are observed.

T-test shows that the proposed EMCM algorithm is significantly better ($p<0.05$) than QBC at nearly half of the check points and statistically outperforms ($p<0.05$) Greedy and RAND most time, suggesting that the proposed EMCM algorithms are effective in choosing the most informative examples and robust to various types of data distributions.

VI. Conclusions and Future Work

In this paper, a new active learning framework, Expected Model Change Maximization (EMCM), is proposed for regression. It measures model change as the difference between the current model parameters and the parameters trained with expanded training set. Inspired by Stochastic Gradient Descent updating rules, we use the gradient of the loss to estimate the model change. Under this framework, we first derive a novel active learning algorithm for linear regression, which aims to choose the examples that maximally change the current model. The learner for nonlinear regression is Gradient Boosting Decision Tree (GBDT), a well-known nonlinear regression model. The GBDT is approximated as a linear regression model through feature mapping. We then propose an active learning algorithm for GBDT to choose the examples which result in the largest change to the model with respect to super features. Extensive experimental results on benchmark data sets from UCI machine learning repository have demonstrated that the effectiveness and robustness of the proposed active learning algorithms.

In this study, the proposed algorithms perform batch mode active learning, i.e. selecting the top k informative examples. The correlation or similarity among the selected examples at each batch is not considered. A possible extension of the work is to consider the diversity of the selected data set to further minimize labeling cost.

Acknowledgments

This research was supported by National Natural Science Foundation of China (No. 61003107), the High Technology Research and Development Program of China (2011AA01A107, 2012AA011702), and Shanghai Science and Technology Rising Star Program (No. 11QA1403500).

References

[1] D.D. Lewis and W.A. Gale, "A Sequential Algorithm for Training Text Classifiers," In *Proc. 17th ACM Int'l SIGIR Conf. (SIGIR'94)*, pp. 3-12, 1994.

[2] Y. Freund, H.S. Seung, E. Shamir, and N. Tishby, Selective Sampling Using the Query by Committee Algorithm, *Machine Learning*, vol.28, pp. 133-168, 1997.

[3] B. Settles and M. Craven, "An Analysis of Active Learning Strategies for Sequence Labeling Tasks," In *Proc. Int'l Conf. Emprical Methods in Natural Language Processing (EMNLP'08)*, pp. 1070-1079, 2008.

[4] N. Abe and H. Mamitsuka, "Query Learning Strategies using Boosting and Bagging," In *Proc. 15th Int'l Conf. Machine Learning (ICML'98)*, pp. 1-10, 1998.

[5] B. Settles, "Active Learning," *Morgan & Claypool*, 2012.

[6] N. Roy and A. McCallum, "Toward Optimal Active Learning through Sampling Estimation of Error Reduction," In *Proc. 18th Int'l Conf. Machine Learning (ICML'01)*, pp. 441-448, 2001.

[7] D.A. Chon, Z. Ghahramani, and M.I. Jordan, "Active Learning with Statisitical Models," *Journal of Artificial Intelligence Research*, vol.4, pp. 129-145, 1996.

[8] S. Tong and D. Koller, "Support Vector Machine Active Learning with Applications to Text Classification," *Journal of Machine Learning Research*, vol.2, pp. 45-66, 2001.

[9] J. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics*, pp. 1189-1232, 2001.

[10] H. Yu and S. Kim, "Passive Sampling for Regression," In *Proc. 10th Int'l Conf. Data Mining (ICDM'2010)*, pp. 1151-1156, 2010.

[11] B. Settles, M. Craven, and S. Ray, "Multiple-Instance Active Learning," In *Proc. Advances in Neural Information Processing Systems (NIPS'08)*, pp. 1289-1296, 2008.

[12] R. Castro, R. Willett, and R. Nowak, "Faster Rates in Regression via Active Learning," In *Proc. Advances in Neural Information Processing Systems (NIPS'05)*, pp. 179-186, 2005.

[13] M. Sugiyama, "Active Learning in Approximately Linear Regression Based on Conditional Expectation of Generalization Error," *Journal of Machine Learning Research*, vol.7, pp. 141-166, 2006.

[14] M. Sugiyama and S. Nakajima, "Pool-based Active Learning in Approximate Linear Regression," *Machine Learning*, vol.75, pp. 249-274, 2009.

[15] P. Donmez and J.G. Carbonell, "Optimizing Estimated Loss Reduction for Active Sampling in Rank learning," In *Proc. 25th Int'l Conf. Machine Learning (ICML'08)*, pp. 248-255, 2008.

[16] T. Fushiki, "Bootstrap Prediction and Bayesian Prediction under Misspecified Models," *Bernoulli*, vol.11, no.4, pp. 747-758, 2005.

[17] R. Burbidge, J.J. Rowland, and R.D. King, "Active Learning for Regression Based on Query by Committee," In *Proc. 8th Int'l Conf. Intelligent Data Engineering and Automated Learning (IDEAL'07)*, pp. 209-218, 2007.

[18] H.T. Nguyen and A. Smeulders "Active learning using pre-clustering," In *Proc. 21th Int'l Conf. Machine Learning (ICML'04)*, pp. 623-630, 2004.