Efficient Multistream Regression

Anonymous Authors

Abstract—**In the data stream regression problem, the predicted values of data instances are generated from a non-stationary process. According to past studies, scholars were focusing on problems under different scenarios such as covariant shift or concept drift. However, lots of these theories are discussing one scenario while keeping others consistent. For example, the training and testing data distribution are assumed to be similar, and true output value related to stream instances would be available. However, in practice these assumptions are not always valid. For most of the stream regression problems, only a fraction of values may be available soon after the prediction. If we only use a small fraction of data to train the model, sampling bias, or shift of the input distribution may be introduced accordingly. Thus the prediction generated by this data sample would not be accurate, so how to appropriately handling this issue is a challenge for recent work. Moreover, if the conditional probability between the training data sample and testing data sample is different under the data stream setting, more estimation biases will be introduced as well.**

In this paper, we present a setting of multistream regression problem, involving two independent non-stationary data generating process. A source stream continuously generates data instances with values of output while the target stream generates data instances without values of output. Notice here the distribution of instances may vary. Hence the issue we want to address here, which is called Multistream Regression, is to estimate the output of each instance in target stream, while using output available on the source stream. Since the concept drift may occur between these two streams, a concept drift detection framework is designed and implemented. An assemble model is built to update the model built before the concept drift point in the source stream. We empirically evaluate the multistream regression's performance on real-world datasets and artificially generated datasets with synthetic biases.

1. Introduction

Recent proliferation of Internet technology in daily life has been creating numerous stream data. These data streams may comes from a variety of sources, such as social networks, online businesses, sensors, military surveillance and so on. Data streams exhibit dynamic characteristics as opposed to static data (e.g. a data warehouse). Hence, unlike traditional datasets, the stream of data flows into a computer system continuously with varying arrival rates. They are time stamped, fast changing, massive, and potentially infinite. Therefore, traditional database technology has limited capability to store and process an entire data stream due to its tremendous volume. Relative mining or analysis algorithms must process the data in real time and update the parameters as the stream progresses.

Regression analysis is a widely used technique for modeling the relationship between dependent variables and independent variables [9]. Several recent approaches focus on applying regression analysis techniques to model and predict the behavior of the stream data. Traditionally, the main task of regression is to learn a function with the input of several attributes from the past stream data, and predicting the value of a numerical target attribute in the future stream data [10], [20]. It is commonly assumed that using the source data with output available to train the linear regression model (pairs of input vectors and output scalars) are representative of the target data. Specifically, in real world, non-stationary data generation process may induce arbitrary changes in data distribution over time. In the past time, series of studies have proposed techniques to address these challenges. Bifet [2] proposed sliding windows to address the shift problem, the window size could be recomputed online according to the rate of change observed from the data in the window itself. However, these techniques make two strong assumptions on the ability of applications that deploying them. One of the assumptions is that true output value of data instances along the stream will be available soon after prediction. The other assumption is that sample data distribution of the training and test mini-batches are equal during regression, this assumption may often be violated in many real-world applications. Also, in real world, continuous output value are normally obtained using mechanisms such as sensors under most of the situations, so it may not occur at a fast enough speed. In this case, a sampling data instances for prediction may be biased, this induces a difference in data distribution between the training and test data [4], [7], [14]. A regression model trained on such a biased training dataset may perform poorly on future unbiased test data. This is known as data shift [11].

Except data shift, there is another challenge called concept drift [22], which occurs in a stream when the underlying concept of data changes over time. So, the regression model needs to be updated regularly to adapt to the most recent concept. The vast majority of concept drift researches assume that true output values of test instances will be readily available to update the regression as soon as they are tested. However, typical feedback are collected from sensors and validated by the human expert, which is a costly and time consuming process. More often, true values are available only for a limited amount of data. Therefore, their

assumption suffer in this scenario.

There are very few effective methods that address regression algorithm under the data shift environment. At first, people use re-weight method to solve previous problem, but this method introduces strong inductive biases that are highly extrapolative. Recently, a more improved method is proposed in the paper [6], [8], [14], here the author established a robust bias-aware regression to solve the covariate shift problem that embraces the resulting uncertainty. Their approach assumes that the stochastic mapping from inputs to the output variable is as similar as possible to a "zero-knowledge" reference distribution. Then they apply minimax relative log-loss optimization approach to minimize relative loss in the worst case, and conditional Kullback-Leibler divergence. Compared with a range of existing regression models on both synthetic and natural biased datasets, their experiment's error value (also log-loss rate) performs best. However, scenario assumed by this approach fail to take consider that there is concept drift occurs. Furthermore, their data stream only consist of single stream, not multi-stream.

In this paper, we introduce a new problem setting consisting of different types of data streams over the same domain, thereby relaxing the above-mentioned assumptions. This approach is motivated by MSC algorithm [5]. We assume the existence of two streams. One stream is referred to as the source stream. This stream consists of data instances, generated by a non-stationary process from a particular domain. Another stream of data instances from the same domain is assumed to be generated by an independent non-stationary process, referred to as target stream. The regression problem is to predict value of data instances in the concept drifting target stream while leveraging real information available in the concept drifting source stream. Since this regression problem involves two types of data streams.For example, source stream with real value and target stream without value, we call it as Multistream Regression. This setting could be illustrated in Figure 1.

The multistream problem proposed resonates with that of domain adaptation [1] and transfer learning [21] when considering a set of observations in each stream within a particular time window. Here, the training and test distributions are assumed to have equal conditional class distribution with unequal marginal distribution. In this situation under the multistream regression environment, we apply a correction mechanism to weigh training data instances so that their weighted data distribution resembles that of the observed test data distribution. Here are major contributions of this paper:

1. We introduce a new data stream regression setting consisting of two independent non-stationary streams. Output value are predicted in target stream using the data instances from source stream. We call this problem as Multi-Stream Regression (MSR).

2. We adapt a Robust Covariate-Shift-aware Regression (RCSR) to address challenges in adapting the source data stream distribution, which may be biased, towards the target data stream distribution in a non-stationary environment.

3. We perform concept drift detection without requiring true dependent variable from the target stream, using dependent variable observed on the source stream.

4. We evaluate this proposed approach empirically over several benchmark datasets, and compare the results with baseline methods. One baseline method, RCSR, is from the paper [6]. The other baseline method pRCSR is periodically update model by using RCSR. Our method Efficient Multi-stream Regression (EMR) significanlty improved performance of baseline methods.

The rest of the paper is organized as follows: Section 2 discusses the related work and Section 3 provides a background on multiple stream regression. Section 4 then describes the proposed approach, and Section 5 reports the experiments and results. Finally, Section 6 concludes with directions to future work.

2. Relative Work

In this section, we present a brief discussion on covariate shift adaptation, concept drift and multistream regression. We also discuss some of the related work in this area.

2.1. Covariate Shift Adaptation

One of the fundamental ideas of data mining technology is that both the training and test data represent the same data distribution, known as the "stationary distribution assumption". In the previous statement, we described that this assumption may violates in real-world application. Addressing an arbitrary difference between training and test distribution is a very difficult problem [15]. Hence, most approaches assume that the source and target data distributions P_{src} and P_{tgt} are related through a covariate shift assumption. As previous describe, the relationship between the source and target data distributions is such that $P_{src}(y|\mathbf{x}) = P_{tgt}(y|\mathbf{x})$ and $P_{src}(\mathbf{x}) \neq P_{tgt}(\mathbf{x})$,where x and y denote the set of covariate values and target value of the data instance respectively.

Recent studies focus on developing correction mechanisms for covariate shift, one algorithm called Kernel Mean Matching (KMM) is proposed in paper [13]. In their algorithm, covariate shift between training and test data distribution is accounted by computing an importance weight $\beta(x) = \frac{P_{tgt}(\mathbf{x})}{P_{src}(\mathbf{x})}$. For each source instance x, we could use this algorithm in the learning process. Except KMM algorithm, KLIEP [23],and unconstrained Least Square Importance Fitting (uLSIF) [16], are among the available techniques in the literature for handling covariate shift in data. However, these approaches work only for fixed-size training and test data. For stream data, Kawahara and Sugiyama [17] extended previous KLIEP to direct online density ratio estimation and get good result. But it works on a single stream of data, where set of source/reference and target data instances are determined by a sliding window. In our paper, we consider multiple streams of data, where new data instance may arrive arbitrarily in any stream.

2.2. Concept Drift

Concept drift refers to the scenario when the relation between input data X, feature values, and the target variable y will be changed over time. Formally, concept drift between time t_0 and t_1 can be defined as

$$\exists X : P_{t_0}(X, y) \neq P_{t_1}(X, y) \quad (1)$$

Where $P_{t_0}(X, y)$ represents the joint distribution of the set of input variables X and the target value y at time t_0, and $P_{t_1}(X, y)$ is the joint distribution at time t_1 [12]. The components of this equation include the prior probability of the target variable P_y, the conditional probability $P(X|y)$, distribution of incoming data P_X, and the posterior probability $P(y|\mathbf{x})$. Real concept drift refers to the changes in posterior probability $P(y|\mathbf{x})$. It may occur due to several reasons including changes in incoming data distribution P_X.

2.3. Stream Regression

Many methods can be found in the literature for solving classification tasks on streams, while only few exists for regression tasks. Especially, there are limited number of paper talking about regression under stream setting with concept drift. An incremental algorithm that scales linearly with the number of examples is presented to solve an incremental model tree problem [9]. The authors present an incremental node splitting rule, together with incremental methods for stopping the growth of the tree and pruning.

Multiple Linear Regression (MLR) analysis is also discussed for time series data stream. It is based on the OLAP technology for streaming data. This system enables an online computation of linear regression over multiple dimensions, and tracks the change of trend according to user's interest. However, none of the researches mentioned above takes sudden, unusual change of the conditional probability distribution, known as concept drift, in to consideration. Specifically, if we have a multistream setting with concept drift, none of these methods described can learn the change in hidden variables or in the intrinsic properties of the observed variables. Hence the study of this problem is very important.

3. Preliminaries

In this section, we formalize the multistream regression problem and present challenges of performing regression over drifting data streams in this context

3.1. Multiple Stream Regression

Data stream regression is a challenging task due to its inherent properties, like infinite length, concept drift. Infinite length problem is addressed by dividing the whole stream into fixed-size mini-batches. However, previous work suggesting for regression data stream are typically based on a single data stream. Recently, multiple data streams model (MDS) and some related issues are discussed in several papers. Although research on multiple data streams model is not started for a long time, multiple data streams has been involved in many of the data stream applications, today such as advances in micro-electro-mechanical system, wireless communication technology and so on.

From the above introduction, multiple data streams model is different from that single stream model in the following aspects [3].The first aspect is multiple data streams model adopts multiple data domain to generate distributed data streams independently. What's more, there are requirements over all the data streams such as comparing local pattern and regional/global pattern, and synchronizing data points from these data streams. Finally, it is difficult or impractical to simply view and treat the data from different data sources as a single data stream with multiple marginal probability, which means that source and target data represent the same data distribution.

Here we propose a new problem setting called Multistream Regression, in this situation, two data streams over the same domain are considered relaxing the following assumptions: first, true values of data instances along the stream are not available soon after prediction. Second, with previous statement, source and target data represent different distribution.

TABLE 1. NOTATIONS

Symbol	Meaning
D	Domain
x	Independent variables of m dimensions
y	Value of a data instance
S	Source data stream with actual y
T	Target data stream without actual y
M	Parameter matrix of regression model
θ	Vectorized
f_{src}	Probability distribution function of source data
f_{tgt}	Probability distribution function of target data
f_{tm}	Probability distribution of target mini batch
$\hat{Y}$	Predicted value of a data instance
N_{max}	Maximum allowable size for dynamic window

3.2. Notations and Problem Statement

In this paper, notations listed in Table 1 are used to describe concepts of the data stream. A continuous stream of data instances is generated from domain D. A data instance is denoted as (x, y), where x is a m-dimensional vector, and y is the corresponding value. Two processes S and T in domain D are used to represent the source and target data stream. In process S, values of both x and y are available, while in process T, only x is available. Thus, a $Multi-streamRegression$ problem can be defined as follows:

Suppose X_s is a set of n-dimensional vectors and Y_s is the corresponding true value in a certain domain D. Meanwhile, X_t is a set of n-dimensional vectors in a target data stream in the same domain. We want to build a regression model f_{ens} to estimate the value that corresponds X_t using X_s, Y_s and X_t.

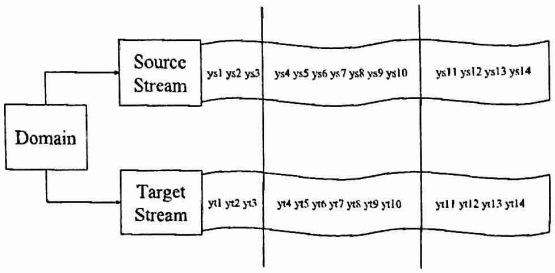

Figure 1. Demonstration of synchronous concept drift on data streams

3.3. Challenges

Normally we assume that inputs in the source set follow the same distribution as the inputs in the target set, so that source data provides solid information in predicting instances in the target set. However, this assumption is not always valid in a multistream setting. Equivalently, consider sample probability distributions of S and T within a certain time period. $P_S^t(y|\mathbf{x}) = P_T^t(y|\mathbf{x})$ and $P_S^t(\mathbf{x}) \neq P_T^t(\mathbf{x})$ at time t. In this condition, the distribution of $P(\mathbf{x})$ is biased in source compared to that in target, and this condition is called *covariate shift*.

Meanwhile, in a real time regression analysis setting such as prediction to data stream, there is another challenge referred as *concept drift*. It describes the phenomenon that data pattern evolves over time, or more formally, the conditional probability distribution changes over time in the target stream. In our problem, it can be expressed mathematically as $P_S^t(y|\mathbf{x}) \neq P_T^t(y|\mathbf{x})$ at time t. In this paper, we assume that two data streams have synchronous data drifts, which can be seen in Figure 1.

4. Multistream Regression

In this paper, we proposed a framework for multi-stream regression, referred as MSR. As we discussed before, in the multi-stream setting, the assumption of covariate shift holds valid between source and target streams until concept drift occurs in the target stream. Hence, the objective of setting a MSR is to predict the output values in target stream efficiently using the information available in the source stream. To archive this goal, we establish two models. The first model is a regression model in the source stream with both input features and output values available. Parameters trained by this model can be used to make accurate prediction in source and target stream. We also applied a change detection technique (CDT) to detect concept drift in target stream. Once a concept drift is detected in the target stream, we will use the most recent data instances in the source stream to update the model, so that the covariate shift assumption can be restored. The flow chart of this algorithm is shown in Figure 2.

As shown in Figure 2, data instances in both source and target streams are generated continuously at the same time. The ensemble update is initialized by source and target

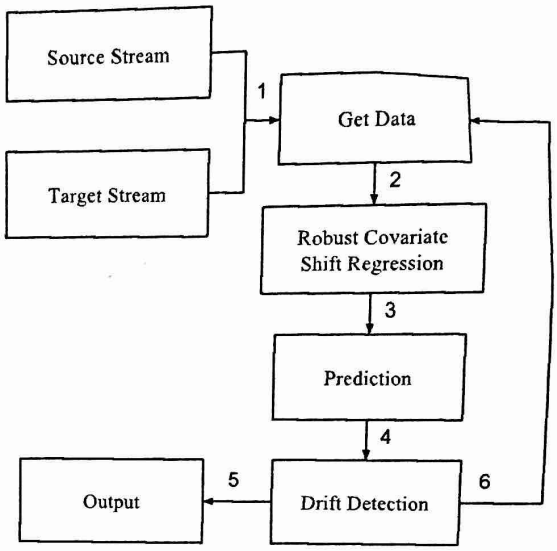

Figure 2. Multistream Regression Overview

streams, and training process starts on an initial mini-batch of data instances collected from head of source stream (Step 1 and 2). Then estimated output value of target stream is predicted via the trained model (Step 3). Then, the generated predictions is used as an input for Drift Detection phase (Step 4). If a concept drift in target stream is detected by the algorithm, a new regression model is trained based on the latest minibatch data available in the source stream, so that bias in the stream can be removed (Step 6). The parameters of this ensemble model are further used for generating the output value of target stream.

Algorithm 1 Multistream Regression

procedure MSR
begin:
  $B_s, B_t \leftarrow readData(S, T, I)$
  $M \leftarrow buildModel(B_s, B_t)$
  while S or T *exists* **do**
    $B_s, B_t \leftarrow readData(S, T, 1)$
    $W_s \leftarrow getError(E_{param}, B_s)$
    $W_t, \hat{Y} \leftarrow preData(E_{param}, B_t)$
    if $z \leftarrow checkDrift(W_t)$ **then**
      $B_s, W_s \leftarrow updateBuffer(z, B_s, W_s)$
      $M \leftarrow buildSourceModel(B_s)$
    $\hat{y} \leftarrow getPrediction(E_{param}, B_t)$
    print $getAccuracy(\hat{y}_t, B_t)$

4.1. Initialization and Covariate Shift Correction

In our proposed framework of MSR, data instances in source and target streams are stored in data buffer B_s and B_t respectively. Due to the nature of multi-stream regression setting, data in B_s comes with output value, and data in B_t arrives without output value.

The RCSR algorithm is derived from minimax robust estimation [6], which generates the predictor that minimizes the worst-case prediction loss. If the log loss function is employed as the loss function, this approach reduces to the principle of maximum entropy. A natural loss function used to evaluate the conditional logloss on the target distribution is shown below:

$$\begin{aligned} &\text{reloss}_{f_{tgt}(\mathbf{x})}(f(y|\mathbf{x}), \hat{f}(y|\mathbf{x}), f_0(y|\mathbf{x})) \\ &= E_{f_{tgt}(\mathbf{x})f(y|\mathbf{x})}\left[-\log\frac{\hat{f}(y|\mathbf{x})}{f_0(y|\mathbf{x})}\right] \end{aligned} \tag{2}$$

Here this conditional logloss evaluates the level of covariate shift by measuring the "surprise" to tell samples from $f_{tgt}(\mathbf{x})f(y|\mathbf{x})$ while actually is from $f_{tgt}(\mathbf{x})\hat{f}(y|\mathbf{x})$ [18].Thus, if there isn't significant covariate shift from the target set to the source set, the solution could be a minimax robust estimation approach subjects to quadratic squares solution for regression. Finally, the regression estimator should be the one that is robust to data set that contains the most significant covariate shift. We define the difference in conditional logloss between an estimator $\hat{f}(y|\mathbf{x})$ and a baseline condition distribution $f_0(y|\mathbf{x})$ on the target stream distribution $f_{tgt}(x)f(y|\mathbf{x})$ as relative loss in this equation.

Thus, the optimization problem of robust covariate shift regression can be interpreted as a two-player game in which the estimator first chooses $\hat{f}(y|\mathbf{x})$ to minimize relative loss and then the adversarial evaluation player chooses $f(y|\mathbf{x})$ to maximize relative loss. The estimator $\hat{f}(y|\mathbf{x})$ is the saddle point solution of the following minimax optimization:

$$\text{minimax } reloss_{f_{tgt}(\mathbf{x})}(f(y|\mathbf{x}), \hat{f}(y|\mathbf{x}), f_0(y|\mathbf{x})) \tag{3}$$

Notice that the objective function of this optimization is convex and the constraints are each affine. Hence, standard tools from convex optimization can be deployed to obtain the solution for the constrained optimization.

The robust covariate shift regression for the target distribution $f_{tgt}(\mathbf{x})$ predicted using constraints from the source distribution $f_{src}(\mathbf{x})$ takes the form:

$$\hat{f}_\theta(y|x) = f_0(y|x)e^{-\frac{f_{src}(\mathbf{x})}{f_{tgt}(\mathbf{x})}\theta^T\Phi(\mathbf{x},y)} \tag{4}$$

$$where:\ \Phi(\mathbf{x},y) = \begin{bmatrix} y \\ \mathbf{x} \\ 1 \end{bmatrix}\begin{bmatrix} y & \mathbf{x} & 1 \end{bmatrix} \tag{5}$$

with parameters obtained via target distribution maximum conditional log likelihood estimation:

$$\theta = \arg\max_\theta E_{f_{tgt}(\mathbf{x})f(y|\mathbf{x})}\left[\log\hat{f}_\theta(y|\mathbf{x})\right] \tag{6}$$

$$\theta^T vector(\Phi(\mathbf{x},y)) = \begin{bmatrix} y \\ \mathbf{x} \\ 1 \end{bmatrix}^T M \begin{bmatrix} y \\ \mathbf{x} \\ 1 \end{bmatrix} \tag{7}$$

Thus the robust covariate shift regression is a conditional Gaussian distribution:

$$\begin{aligned} &\hat{f}_M(y|\mathbf{x}) \sim N(\mu(\mathbf{x},M), \sigma^2(\mathbf{x},M)) \\ &where:\ M = \begin{bmatrix} M_{(y,y)} & M_{(y,x_1)} \\ M_{(x_1,y)} & M_{(1,1)} \end{bmatrix} \end{aligned} \tag{8}$$

$$\begin{aligned} \mu(\mathbf{x},M) = &\left(2\frac{f_{src}(\mathbf{x})}{f_{tgt}(\mathbf{x})}M_{(y,y)} + \frac{1}{\sigma^2}\mu_0\right)^{-1} \\ &\left(-2\frac{f_{src}(\mathbf{x})}{f_{tgt}(\mathbf{x})}M_{y,x_1}\begin{bmatrix}\mathbf{x}\\1\end{bmatrix} + \frac{1}{\sigma^2}\mu_0\right) \end{aligned} \tag{9}$$

4.2. Value Prediction

Now we can use the matrix derived in Section 3.3 to predict the value for every instance in the dataset. This prediction is based on the Minimax estimation formulation, and it minimizes the target distribution conditional $Kullback - Leiber$ divergence. Hence the Covariate Shift problem is addressed by this algorithm. Also, the logloss is evaluated via the $getLogloss$ function in Algorithm 1. Now the data in both source and target dataset are unbiased.

4.3. Drift Detection

In this section, a $detectDrift$ method is introduced to label those points in a data stream where significant changes happen. These changes are referred as concept drift. We can use the generated $\hat{y}$ in the target set, along with those instances with y in the source set, to figure out the drift point in the target stream. A concept drift along S could significantly impact the performance of Multistream regression model. Here, the concept drift we mentioned here specifically indicates a $within - stream$ drift. This type of concept drift can cause change in the data distribution. Once the change point is detected, the problem could be handled by training another regression model so that the regression parameters can be updated. We expect the prediction accuracy will be greatly improved once we mark the drift points and update the regression model accordingly.

4.3.1. Window Management. In order to detect the change point accurately, two sliding windows on both source stream and target stream are maintained. The reason to do this is that we want to generate feedback on most recent data. The predicted value of data instances is generated by the regression model, and is inserted into W. In the window, there is an ensemble confidence score calculated from regression of data instances in T. Then values of confidence are generated within a range of [0,1]. Based on the nature of the confidence level (confidence score value is usually low until there is a concept drift), here we apply a $beta$ distribution to model the confidence level. Note that the window management schema for source stream is to follow the corresponding window in the target stream, so that most recent data can be used for training if there is a concept drift in the stream.

4.3.2. Change Detection. We propose a change point detection method to check for significant change in the target stream. The reason of window management is to make the model check for concept drift, then update the prediction using most recent data automatically.

Algorithm 2 drafts our proposed CDT. If at any time point, the average feedback is below 0.5, or size of the the window exceeds N_{max}, the ensemble model is updated immediately regardless of any distribution change. Otherwise, this detection technique divides the window W_h for each q between γ and $n - \gamma$, where n is the total number of data instances and γ is the cushion size. Thus, $W_h^b = W[1:q]$ is composed of older observations and $W_h^a = W[q+1:n]$ is composed of recent ones. Let θ_a and θ_b be the estimated distribution parameters from W_h^a and W_h^b, we calculate the sum of log likelihood ratios as follows:

$$S(q,n) = \sum_{i=q+1}^{n} \log\left(\frac{P(W_h[i]|\theta_a)}{P(W_h[i]|\theta_b)}\right) \tag{10}$$

where $P(W_h[i]|\theta)$ is the probability density function ,given a set of parameters θ, applied on the i^{th} instance stored in W_h. Using this output, ω_n can be calculated accordingly to determine whether a change point is detected.

Algorithm 2 Concept Drift Detection

```
procedure DETECTDRIFT
begin:
    Threshold ← −log(α_d), n ← size of W_h,
    and ω_n ← 0
    if n ≤ N_max & mean(W_h[1 : n] > 0.5 then
        for q ← γ : n − γ do
            Estimate pre and post change distribution
            parameters, θ_α and θ_β from W_x[1 : q] and
            W_h[q + 1 : n] respectively
            Calculate S(q, n) using Equation (3)
        ω_n = max_{γ≤q≤n−γ} S(q, n)
        if ω_n ≥ Threshold then
            return kmax, where S_kmax = ω_n
        else
            return -1
    else
        return 0
```

4.4. Drift Adaption

As we mentioned before, parameters generated by the regression model will be updated once a concept drift is detected. If a drift is detected in the T stream, new regressions are trained accordingly using a minibatch of recent data instances, which represents a new data distribution. However, the significance of each drift varies. Updates for each drift are not always necessary as it will increase the computational complexity dramatically. Here, we assume that the covariate shift assumptions hold between the source and target stream initially. Since the drift occurs simultaneously on S and T, for each detected drift in a stream, the associated data buffer (B_s for B_t) and feedback buffer (W_s or W_t) of the stream are updated by removing data instances before the change point, thereby simulating dynamic size of buffer. This is performed using $updateBuffers$ in Algorithm 1.

5. Experiment

5.1. Datasets

5.1.1. Artificial Dataset. To evaluate the effectiveness of the algorithm, one way is to use artificially generated data. Since normally there are two categories of concept drift, we generate our dataset based on the following two criteria, 1. local or global; 2. abrupt or gradual.

For local concept drift, the data distribution changes only over a constrained region of instance space. On the other hand, for global concept drift, the distribution changes over the whole region of instance space, that is, for all the possible values in the target stream. Meanwhile, abrupt and gradual concept drift is the description of the sharpness of a specific drift. If the change happens suddenly and leads to a large drift within a short period, it can be referred as an abrupt drift. Otherwise, changes can be referred as a gradual drift.

Then, we start to simulate and study three scenarios for concept drift:

1. *Global gradual drift*: The first type of simulated drift is global and gradual. The only occurrence of the gradual concept drift is at 1/2 of example of both source and target streams. Starting from this drift point, instances from the new concept are being gradually introduced among the examples to the old concept.
2. *Global abrupt drift*: The second type of simulated drift is global and abrupt. The concept drift appears over the whole instance space. There are one points of concept drift, which occurs at 1/2 of the example.
3. *Local abrupt drift*: The third type of simulated drift is local and abrupt. Here, the concept drift appears in two distinct regions of the instance space. There are three points of abrupt change of both source and target streams, the first one at 1/4 of the examples, the second one at 1/2 of the examples, and the last one at 3/4 of the example.

TABLE 2. DATASET

Dataset	No. of features	No. of instances
globalGradual	5	100,000
globalAbrupt	5	100,000
localAbrupt	5	100,000
beijingPM2.5	12	43824
bikeSharing	16	17389

5.1.2. Real World Data. We deploy publicly available regression datasets from the UCI repository [19] to evaluate our approach. The number of instances and features can be found in Table 2 as well.

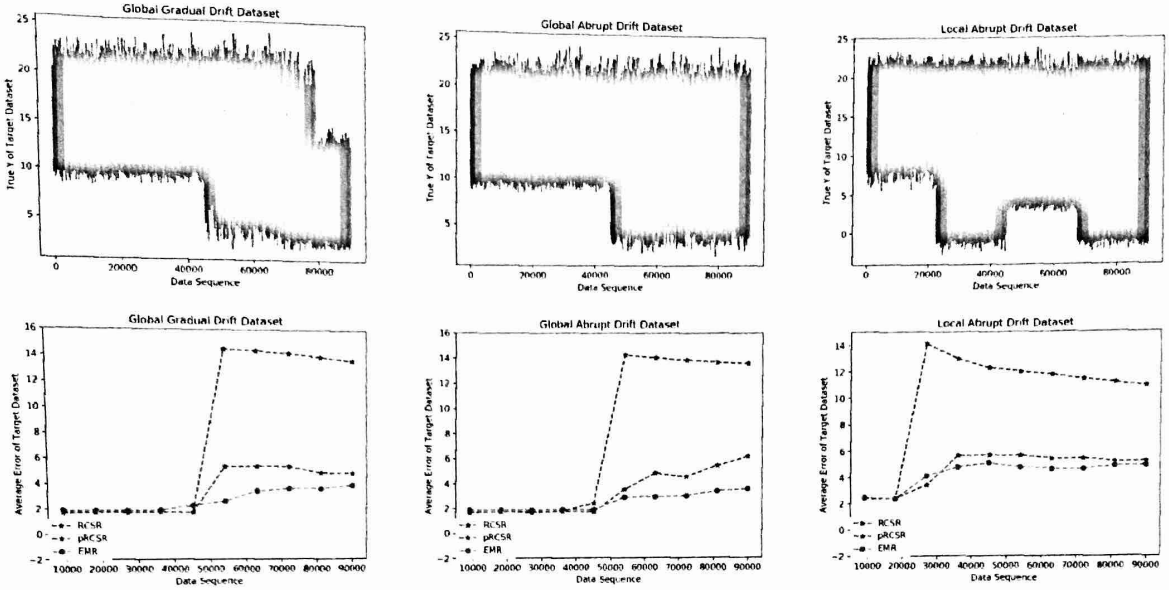

Figure 3. Stream: MAE on artificial datasets

TABLE 3. STREAM: MAE ON ARTIFICIAL DATASETS

	globalAbrupt			globalGradual			localAbrupt		
Iteration	RCSR	pRCSR	EMR	RCSR	pRCSR	EMR	RCSR	pRCSR	EMR
900	**1.69**	1.74	1.90	**1.71**	1.76	1.91	2.50	**2.42**	2.50
1800	1.75	**1.73**	1.90	**1.75**	1.76	1.91	2.42	2.42	**2.40**
2700	**1.68**	1.74	1.90	**1.74**	1.76	1.91	14.15	**3.42**	4.10
3600	1.77	**1.73**	1.91	**1.75**	1.76	1.92	12.97	5.66	**4.80**
4500	2.43	**1.72**	1.92	**1.76**	1.76	2.31	12.29	5.66	**5.06**
5400	14.27	3.51	**2.89**	14.55	5.37	**2.64**	12.02	5.66	**4.78**
6300	14.03	4.79	**2.95**	14.40	5.37	**3.43**	11.76	5.42	**4.62**
7200	13.85	4.52	**2.98**	14.21	5.37	**3.64**	11.43	5.42	**4.66**
8100	13.69	5.41	**3.42**	13.94	4.83	**3.94**	11.17	5.19	**4.90**
9000	13.56	6.13	**3.53**	13.61	4.83	**3.88**	10.91	5.19	**4.90**

5.2. Experiments

5.2.1. Baseline Method. Since there aren't existing studies about regression problems under a multistream setting, here, we designed two baseline methods, including Robust Covariate Shift Regression (RCSR) and periodically update Robust Covariate Shift Regression (pRCSR). RCSR is the method described in Section 2 to eliminate the impact of covariate shift in between the source and target stream. And according to Chen's paper, the RCSR method has better performance on dataset with shift compared to normal method such as Multiple Linear Regression (MLR). Hence two baseline methods are designed as follows:

1. We follow the setting of RCSR to train regression parameters M. Just as the first baseline method, all the parameters are generated on an initial set of source and target data instances in both source and target stream. Then M is used to predict all instances in T. Hence the difference between this baseline method and our proposed method in this paper is that this baseline method doesn't update the model based on the shift of probability density of data streams. We denote this baseline method as RCSR.

2. We update the RCSR model with periodically update. Generally speaking a new RCSR model is trained using the latest data instances in both S and T data streams. We assume the periodically update will improve the prediction accuracy of the model proposed. Here we denote this baseline method as pRCSR.

5.3. Complexity Analysis

At every iteration of MSR, a new data instance is obtained from both S and T streams. Hence the time complexity of this framework is highly depended on the regression we use. Since the prediction of output values is just a simple matrix multiplication, hence, the training process, which includes RCSR and drift detection, determines the complexity of the whole algorithm.

RCSR is a minimax learning process trained by B_s. Since there isn't any training on the target stream, we

error of RCSR model is high after the drift point in all datasets. Meanwhile, the pRCSR model performs better than the RCSR due to the periodical updates. The updated coefficients is a significant improvement of prediction accuracy regarding the stream data. For example, in the Bike Sharing Dataset, the MAE of pRCSR stabilizes around 0.20 after the drift point, however, the RCSR reached to 0.26 by the end of 13,137 data instances in the target stream.

When comparing our proposed EMR approach with pRCSR, we can see that the prediction accuracy is even better. Our proposed EMR model consistently over perform the periodically updated RCSR proposal on every dataset. The reason is that even though pRCSR updates the model periodically, in most cases the update point won't match the drift point. That means there is a period between the drift point and periodical update point, predication is generated by old model coefficiency. This fact will lead to an increasing MAE. On the other hand, due to the drift detection method applied in the model, which helps EMR to update the model coefficiency once drift is detected, the regression model in EMR will be updated on a case-by-case basis. That means the regression model will only be updated when needed, and the time complexity is much less than update the model with higher frequency. Thus, the very existence of concept drift detection algorithm helps the RCSR model to get more accurate predictions in an efficient way.

6. Conclusion and Future Work

In this paper, we propose an efficient multistream regression framework to address the prediction problem in a multistream setting Generally speaking, true value of dependent variable (Y_s) is only available in the source stream, and is used to make prediction for dependent variable of the target stream along with independent variables of both streams (X_s and X_t). Challenges of solving problems having covariate shift and concept drift simultaneously in two data streams is widely discussed. Our solution involves a minimax approach for regression under covariate shift, and concept drift detection techniques to correct drift. Following experiments show that our solution has generated much better performance in terms of MAE on various datasets, compared to baseline models.

There are some future work that we can think of to extend the use of the proposed framework in this paper. For example, in this paper we assume that concept drift happens simultaneously in both source and target data stream. However, concept drift can occur asynchronously as well. Hence there are opportunities to discuss the effect of asynchronous drift on the prediction error as well.

References

[1] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, no. 1-2, pp. 151–175, May 2010. [Online]. Available: http://dx.doi.org/10.1007/s10994-009-5152-4

[2] A. Bifet and G. Ricard, "Learning from time-changing data with adaptive windowing," *Proceedings of the 2007 SIAM International Conference on Data Mining*, pp. 443–448, 2007.

[3] M. Bowman, S. K. Debray, and L. L. Peterson, "Reasoning about naming systems," *ACM Trans. Program. Lang. Syst.*, vol. 15, no. 5, pp. 795–825, November 1993.

[4] J. Braams, "Babel, a multilingual style-option system for use with latex's standard document styles," *TUGboat*, vol. 12, no. 2, pp. 291–301, June 1991.

[5] S. Chandra, A. Haque, L. Khan, and C. Aggarwal, "An adaptive framework for multistream classification," *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management - CIKM '16*, 2016.

[6] X. Chen, M. Monfort, A. Liu, and B. D. Ziebart, "Robust covariate shift regression," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 2016, pp. 1270–1279.

[7] M. Clark, "Post congress tristesse," in *TeX90 Conference Proceedings*. TeX Users Group, March 1991, pp. 84–89.

[8] S. Fear, *Publication quality tables in LaTeX*, April 2005, http://www.ctan.org/pkg/booktabs.

[9] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics Springer, Berlin, 2001, vol. 1.

[10] J. Gama, *Knowledge Discovery from Data Streams*, 1st ed. Chapman & Hall/CRC, 2010.

[11] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Brazilian Symposium on Artificial Intelligence*. Springer, 2004, pp. 286–295.

[12] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, p. 44, 2014.

[13] A. Haque, L. Khan, and M. Baron, "Sand: Semi-supervised adaptive novel class detection and classification over data stream," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI'16. AAAI Press, 2016, pp. 1652–1658. [Online]. Available: http://dl.acm.org/citation.cfm?id=3016100.3016130

[14] M. Herlihy, "A methodology for implementing highly concurrent data objects," *ACM Trans. Program. Lang. Syst.*, vol. 15, no. 5, pp. 745–770, November 1993.

[15] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Scholkopf, "Correcting sample selection bias by unlabeled data," in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, ser. NIPS'06. Cambridge, MA, USA: MIT Press, 2006, pp. 601–608. [Online]. Available: http://dl.acm.org/citation.cfm?id=2976456.2976532

[16] T. Kanamori, S. Hido, and M. Sugiyama, "A least-squares approach to direct importance estimation," *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1391–1445, 2009.

[17] Y. Kawahara and M. Sugiyama, "Sequential change-point detection based on direct density-ratio estimation," *Statistical Analysis and Data Mining*, vol. 5, no. 2, pp. 114–127, 2012.

[18] J. Lafferty, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." Morgan Kaufmann, 2001, pp. 282–289.

[19] X. Liang, T. Zou, B. Guo, S. Li, H. Zhang, S. Zhang, H. Huang, and S. X. Chen, "Assessing beijing's pm2.5pollution: severity, weather impact, apec and winter heating," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, vol. 471, no. 2182, p. 20150257, 2015.

[20] E. Lughofer, *Evolving Fuzzy Systems - Methodologies, Advanced Concepts and Applications*, 1st ed. Springer Publishing Company, Incorporated, 2011.

[21] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.