Dima Alberg
Department of Industrial Engineering and Management,
SCE - Shamoon College of Engineering,
Beer-Sheva, Israel

Incremental Interval Regression Tree Learning with Mean Variance Numerical Data Streams

UDC: 005.521 ; 005.82
DOI: 10.7595/management.fon.2012.0013

XIII International Simposium SymOrg 2012, 05-09 June 2012, Zlatibor, Serbia

In this paper, we present a novel method for interval regression tree incremental learning with mean variance patterned numerical data streams. The proposed Mean Variance Interval Regression Tree (MVIRT) algorithm transforms continuous temporal data into two statistical moments according to a user-specified time resolution and builds a regression model tree for estimating the prediction interval of the target variable. The algorithm main properties are time - based incremental mean variance tree induction algorithm accompanying novel time resolution and outliers detection mechanism. Results of real world data stream show that the MVIRT algorithm produces more accurate and easily interpretable prediction models than other state-of-the-art batch incremental model tree methods.

Keywords: prediction, regression tree, incremental learning, data stream mining, interval prediction

1. Introduction

A sequence of instances, such as sensor transmitted data, geospatial location, tick - by - tick stock prices and exchange rates, electric signals and voltage, the values of which may be recorded and observed continuously in time, is called a *data stream*. The sources of data streams include meteorological and financial data, network monitoring, web applications, sensor networks, etc.

In regression, the attribute for which values are being predicted is numeric (continuous-valued and ordered) rather than categorical (discrete valued and unordered). This attribute can be referred to as the predicted attribute. Note that regression[7] can also be viewed as a mapping function, $Y=f(X)$, where X is the input and the output is a continuous or ordered value Y. Thus, the regression problem in data streams will be considered as the problem of learning a function of several input attributes from the past data stream with the goal of estimating as accurately as possible the value of a numerical target attribute in the future data stream.
The classical batch regression tree methods for predicting a numerical target variable build upon the supervised divide-and-conquer approach. In terms of the terminal node (leaf) representation, these methods can be classified into following categories:

- constant or mean value
- model
- interval prediction

The main difference between the methods is that the model and the interval representations, unlike the mean value approach, employ more complex prediction models with the aim of improving the target variable prediction accuracy. An efficient batch regression model tree algorithm ensures that the model construction and prediction can be processed with consistently improving accuracy, as new information is incorporated. Moreover, smaller trees, i.e., those which do not have too many splits, are easier to get and to interpret. In many real world applications, which contain large data masses the regression model tree fast processing and ease of interpretation is just as important as predictive accuracy [6, 10].

According to Holmes et al.[8] the batch - incremental algorithms place hard restrictions on regression tree learning. First, a model must be induced incrementally. Second, the instances processing time must keep up with their speed of arrival. Third, a model may only use a constant amount of memory while providing an

accurate and updated prediction model at any point in time. Therefore to overcome these restrictions Bifet et al.[3, 4] introduced the new class of the ensemble batch - incremental algorithms. These algorithms usually apply an ensemble of sliding window methods and split the data stream into a stream of disjoint batches whereas each data batch can be processed in the order of its arrival using one of batch regression tree algorithms, which are discussed much deeper in the next section.

Mining data streams with model

2. Mining data streams with model tree regression methods

Although regression trees are well-studied class of learners, little research has been done in the area of incremental regression tree induction. The problem becomes even harder if the learning must deal with continuous data streams. That means that the regression tree learner becomes fully incremental and must be updated with every new arriving instance. In an incremental learning system, the data instances arrive sequentially in time. Each instance represents a snapshot of the changing environment system at time *t*. Since data streams are very long and often open-ended, the system must be updated using one instance at a time. Each instance is discarded as soon as it is used for updating.

Potts[11, 12] brings together the batch and incremental versions of the two splitting rules of regression linear model tree with two uni□ed frameworks Online - RD and RA. The proposed tree models are built from the top down, using one of two statistical tests to determine both the split point and whether to carry on splitting. They use the Chow[4] test, a standard statistical test for homogeneity amongst sub-samples.

Alberg and Last[1,2] introduced MOPT (Mean Output Prediction Tree) algorithm for interval prediction of numeric target variables from temporally aggregated numerical data, where each aggregated data instance is represented by its mean and variance. The proposed algorithm differs from the state-of-the-art regression algorithms in the splitting of each input and output feature to two moments according to the input time resolution and it can also identify the most appropriate prediction time resolution that minimizes the prediction error and builds more compact interval based regression tree. The main disadvantages of the MOPT algorithm are posed by the fact that is not using an explicit time resolution detection mechanism and thus is not suitable for dealing with massive data streams, which may include distribution change patterns and can require excessive memory and processing power resources.

Ikonomovska et al.[9] describe the FIMT-DD (Fast and Incremental Model Tree with Drift Detection) algorithm which is an advanced adaptation of the FIMT (Fast and Incremental Model Tree) and FIRT (Fast and Incremental Regression Tree) algorithms and uses an explicit change detection method (DD) for dynamic environments and time-changing distributions. The main difference between FIRT and FIMT algorithms lies in the fact that FIRT has no linear models in the leaves. According to the authors, the main advantages of the FIMT-DD are that it is competitive with batch algorithms in terms of accuracy, enables local change detection, and avoids the costs of re-growing the whole tree when only local changes are necessary.

The common disadvantage of the represented methods that they are unable to detect changes properly and adapt their tree models with a minimal loss of accuracy. A simple way to cope with this problem is to calculate every possible splitting point. As a result, the task becomes computationally expensive and it has a negative effect on the algorithm scalability. This non-trivial problem raise the need of incremental algorithms with fast execution and response time, which are able to detect changes properly and adapt their tree models with a minimal loss of accuracy.

3. The Mean Variance Interval Regression Tree MVIRT Methodology

The Mean Variance Interval Regression Tree (MVIRT) algorithm introduced in this paper requires continuous aggregated temporal variables, represented in the form of two unbiased estimators (sample average and variance) and produces an interval batch incremental interval regression tree for a numeric target variable *Y*.

In our algorithm the average and variance of each variable will be mapped in univariate Mahalanobis distance

based on auxiliary control variable *M(·)* that should respond to changes in both statistical moments. The proposed approach enables to ignore outliers, which cause prediction instability and model overfitting effect, and thus achieve a considerable reduction in the size of the induced tree.

$$X_i \sim \left\{\bar{x}_i(r);\ \hat{s}^2_{x_i}(r)\right\}, where\ i \epsilon (1, \dots, N) \quad (1)$$

Suppose that each instance *i* of input variable *X* is represented by two aggregated mean and variance estimators $\{\bar{x}_i(r)\}$ and $\left\{\hat{s}^2_{x_i}(r)\right\}$ for a given temporal measurement aggregation resolution *r*. Let x_A $\{\bar{x}_i(r)\}$ and x_S $\left\{\overline{\hat{s}^2_{x_i}}(r)\right\}$ be the sample average values over all instances, and let V_A $\{\hat{s}^2(x_i(r))\}$ and V_S $\left\{\hat{s}^2\left(\hat{s}^2_{x_i}(r)\right)\right\}$ be the sample variance of the corresponding unbiased estimators. The corresponding sample covariance between $\left\{\bar{x}_i(r);\ \hat{s}^2_{x_i}(r)\right\}$ is denoted by V_{AS} $\left\{\hat{s}^2\left(\bar{x}_i(r), \hat{s}^2_{x_i}(r)\right)\right\}$ and the Mahalanobis distance between the two measured statistical moments of the input variable *X* is calculated by:

$$M_i\left(\bar{x}_i(r);\ \hat{s}^2_{x_i}(r)\right) - \frac{N}{V_A V_S - V_{AS}} \cdot (*), where$$
$$(*) = V_S \cdot (\bar{x}_i(r) - x_A)^2 + V_A \cdot \left(\hat{s}^2_{x_i}(r) - x_S\right)^2 - 2\sqrt{V_{AS}} \cdot (\bar{x}_i(r) - x_A) \quad (2)$$

To identify outlying values of M_i we need to determine its probability distribution. The proposed distance metric under the null hypothesis (which includes the assumption of multivariate normality of *X* variable) has a chi square - distribution with two degrees of freedom and express Mahalanobis multivariate standardized distance between the values of the current two first moments. For example, if the input variable averages remain at the values x_A and x_S then values of *M(·)* should be less than the $\chi^2_{\alpha,2}$ and greater than *0*, where $\chi^2_{\alpha,2}$ is the upper α percentage point of the chi-square distribution with two degrees of freedom. If at least one of the averages shifts to some new value, then the probability that the statistical moment $\chi^2_{\alpha,2}$ exceeds the limit increases. In the tree induction algorithm the confidence interval limits of the *M* distance metric are calculated by:

$$UCL(M_i) = \frac{2(r-1)(n-1)}{rn-r-1} F_{\alpha, 2, rn-r+1}$$
$$LCL(M_i) - \frac{2(r-1)(n-1)}{rn-r-1} F_{1-\alpha, 2, rn-r+1} \quad (3)$$

where *r* is a temporal aggregation resolution.

4. The MVIRT Tree Splitting Procedure

In the instance splitting stage of the MVIRT algorithm we assume that we have a set of *n* training instances at a given node. The pseudo code in Figure 1 finds the best split for predicting the mean of a numeric target variable. This procedure applies to splitting the values of bivariate numeric input variables, where each variable is represented by the sample mean *AVG(X)* and variance *VAR(X)* according to the predefined temporal aggregation resolution *r*.

The splitting procedure includes three main steps. The first step consists of the Mahalanobis distance calculation for numeric input variable *X* in every instance (see Equation 2) and outliers detection procedure. The second step contains incremental time resolutions logic mechanism which increases current time resolution *TR* when all input variables instances are outliers. It should be noted that if the number of outliers equals to the number of training instances, then the algorithm ignores a given input variable and shifts to the next one or returns the tree. The third final step is aimed at selecting the best estimator (sample average or variance) for input variable. In this step, the algorithm calculates the absolute differences ratio between the value of *MXY* and the values of the estimators *MAVG* and *MVAR* in the best splitting instance of *X* and chooses the best node estimator (Best_Contributor), which minimizes that difference ratio.

MVIRT(α, TR, X, Y) splitting procedure		
Input Args:	User defined , *α* Current time resolution, *r* The mean-variance input variables, *X* The mean-variance target variable, *Y*	
Output:	The best split point for input attribute *X*	
Pseudocode body:		

# Calculate the Mahalanobis distance vector for input variable MX 	
For each instance Do:
MX = M (AVG(X),VAR(X)) (formula 2)
Next
#Outliers detection & droping procedure
{C} = null
If MX(*α*) is not outlier (formula 2) Then
{C} + + # Non outliers data instances collection
End If
		(1)

# Time resolution setting 	
If {C} is empty Then
#Shift to the next input variable
Call recursively MVIRT(α , TR, X, Y)
If all input variables instances are outliers (concept drift is detected) Then
# Increase time resolution TR for current input variable 	
TR = *r* - -
If TR is empty Then
Return MVIRT Tree
Else
Call recursively MVIRT(α, TR, X, Y)
End If
End If
		(2)

# Best contribution variable detection 	
For each instance in {C} Do:
MY = M (AVG(Y), VAR(Y))
MXY = M (M(X), M(Y))
#Mahalanobis distance between averages of X and Y
MAVG = M (AVG(X), AVG (Y)) (formula 2)
#Mahalanobis distance between variances of X and Y
MVAR = M (VAR(X), VAR (Y)) (formula 2)
Next
Best_Contribution_Ratio = Max(|MXY - MAVG |,| MXY - MVAR|)/ |MXY|
#Best split value for current input variable X in time resolution TR
Return Split (TR; Best_Contributor (Avg/Var) ; Best_Split_Value)
		(3)

Figure 1: MVIRT splitting criterion pseudo code

5. The MVIRT Tree Leaf Construction Procedure

In each terminal node the MVIRT algorithm computes the bounds of a prediction interval for the corresponding tree leaf with a user-defined confidence level 1-α using the following equations:

$$\begin{cases} \bar{y}_i \mp t_{1-\alpha/2,\, n_i-1} \cdot \hat{s}_{y_i} \sqrt{\left(1 + 1/n_i\right)}, & n_i \leq 30 \\ \bar{y}_i \mp z_{\alpha/2} \cdot \hat{s}_{y_i} \sqrt{\left(1 + 1/n_i\right)}, & n_i > 30 \end{cases} \quad (4)$$

where $i \in (1,..,n_i)$ are tree leaf instances and $\bar{y}_i$, $\hat{s}_{y_i}$, represent mean and standard deviation estimators of the tree leaf. Thus, when the confidence level equals to zero (i.e. α =100%) then the corresponding values of $t_{0.5}$ and $z_{0.5}$ distributions are equal to zero, the MVIRT model transforms the interval representation of a tree leaf into the sample average of the target variable. This fact is very useful in the sense of experimental comparison between MVIRT tree and other, point estimation regression tree algorithms.

6. El Nino Data Set

The El Nino data stream is available at the UCI KDD Archive (http://www.ics.uci.edu). This data was collected with the Tropical Atmosphere Ocean (TAO) array, which was developed by the international Tropical Ocean Global Atmosphere (TOGA) program (http://www.pmel.noaa.gov). The TAO array consists of nearly 70 moored buoys spanning the equatorial Pacific, measuring oceanographic and surface meteorological variables critical for improved detection, understanding and prediction of seasonal to inter-annual climate variations originating in the tropics, most notably those related to the El Nino/Southern Oscillation (ENSO) cycles. This data stream was collected on a daily basis and contains 178,080 numerical instances from March 1980 to June 1998. Each instance in the data stream has the following numerical attributes: date, latitude, longitude, zonal winds (west<0, east>0), meridional winds (south<0, north>0), relative humidity, air temperature, sea surface temperature and subsurface temperatures down to a depth of 500 meters. The latitude and longitude in the data showed that the buoys moved around to different locations. The wind data, both the zonal and meridional winds fluctuated between -10 m/s and 10 m/s. Relative humidity values in the tropical Pacific were typically between 70% and 90%. Both air temperature and sea surface temperature fluctuated between 20 and 30 degrees Celsius. The target (prediction) attribute in the El Nino data stream is sea surface temperature (SST), which is identified as warmer than normal sea surface temperatures.

There are missing values in the data. As mentioned earlier, not all buoys are able to measure current attributes because these values may be missing dependent on an individual buoy. The missing values replacement operation was performed with both time series neighbors average value interpolating procedure. Finally in order to evaluate the predictive performance, the set of all examples was split into learning and testing examples sets in proportion 70:30.

The MVIRT algorithm performance is compared to three state-of-the-art incremental model tree algorithms implemented by the Rapid Miner and tuned with time series plug-in: M5P[13], M5-Rules[14], RepTree[14]. Due to the memory and time constraints, it was very important to evaluate the capabilities of all algorithms to learn incrementally and correctly and at the same time to construct an appropriately small model tree. Therefore in each experiment we applied the constant months sliding window mechanism. These algorithms do not typically provide interval predictions, therefore, to overcome this limitation we have used the average sliding windows mean estimator to make point predictions in our comparative evaluation experiment. Finally, in order to improve the algorithm scalability, we tuned M5P and RepTree with bagging evaluation mechanism which was implemented in Java API of WEKA package.

The results in Table 1 show that under Average Root Mean Square Error (A[RMSE]) and Average Explained Variability (A[EV]) criteria the MVIRT and the RETIS-M algorithms are more accurate than other proposed algorithms in terms of pair-wise t-Student test difference. We have denoted by * the cases where the *p* value of the difference between MVIRT and other algorithms is smaller than or equal to 5%. The MVIRT algorithm outperforms significantly the other algorithms in terms of Average Cost Complexity Measure (A[CCM]). Fi-

nally, we maintain that our proposed MVIRT tree models are more interpretable than RETIS tree models in terms of Average Tree Size (A[TS]) measure (7 vs. 23).

Table 1: El Nino data set learners comparison

Learner	A[RMSE]	A[TS]	A[CCM]	A[EV]
B-M5R	0.84*	7	1.01*	0.46*
B-M5P	0.83*	10	1.07*	0.47*
B-REPT	1.57*	5	1.69*	NA
M5 RLS	0.86*	7	1.03*	0.45*
M5P TR	0.84*	8	1.03*	0.46*
MVIRT	0.60	7	0.77	0.62
REPT	1.57*	3	1.64*	NA
RETIS	0.63	23	1.18*	0.60

Conclusions and Future Work

In this paper, we have presented the two moments MVIRT algorithm, which is able to predict values of numeric attributes in massive temporal data sets. The proposed algorithm differs from the state-of-the-art regression algorithms in the splitting of each input continuous feature according to the best mean variance contributor, identifies outliers in the training data and as result builds more compact interval prediction tree. The conducted experiment indicate that the proposed MVIRT algorithm produces more accurate and compact models by comparison to the state-of-the-art regression tree algorithms. In our opinion, the proposed algorithm is only the first step towards a family of truly scalable and fast regression tree algorithms. We see two immediate extensions for the future work. First, our general aim is to construct online accurate and robust multi-r target variable forecasting mechanism, which in turn can be used for massive data streams prediction. Second, we can look into other possible analytical methods for the split point selection, which may reduce the algorithm time and space complexity.

REFERENCES

[1] Alberg, D., Last, M., Neuman, R., & Sharon, A. (2009). Induction of Mean Output Prediction Trees from Continuous Temporal Meteorological Data. *2009 IEEE International Conference on Data Mining Workshops* (pp. 208 - 213). Miami, Florida, USA: IEEE Computer Society.

[2] Alberg, D., Last, M., & Kandel, A. (2011). Knowledge Discovery in Data Streams with Regression Tree Methods. *WIRES Data Mining and Knowledge Discovery* , 69-78.

[3] Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., & Gavaldà, R. (2009). New Ensemble Methods for Evolving Data Streams. *In 15th ACM SIGKDD Intl. Conference on Knowledge Discovery and Data Mining (KDD'09).* Paris, France.

[4] Bifet, A., & Kirkby, R. (2009). *Data Stream Mining A Practical Approach.* COSI. Available at http://www.cs.waikato.ac.nz /~abifet /MOA.

[5] Chow, G. (1960). Tests of Equality between Sets of Coefficients in Two Linear Regressions. *Econometrica*, *28* (3), 591– 605.

[6] Friedman, J. (1991). Multivariate Adaptative Regression Splines. *In Annals of Statistics*, 1 - 19.

[7] Granger, C., & Newbold, P. (1986). *Forecasting in Business and Economics* (2nd Edition ed.). Academic Press.

[8] Holmes, G., Kirkby, R., & Bainbridge, D. (2004). *Batch Incremental Learning for Mining Data Streams.* Hamilton: University of Waikato, Department of Computer Science, New Zealand.

[9] Ikonomovska, E., Gama, J., & Dzeroski, S. (2010). Learning Model Trees from Evolving Data Streams. *Data Mining and Knowledge Discovery Journal*, Springer, (pp. 1 - 41).

[10] Krzanowski, W., & Hand, D. (2007). A Recursive Partitioning Tool for Interval Prediction. *In Proceedings of the ADAC (2007)*, *1*, pp. 241 - 254.

[11] Potts, D. (2004). Incremental Learning of Linear Model Trees. *Proceedings of the 21st International Conference on Machine Learning* (pp. 663 - 670). ACM.
[12] Potts, D., & Sammut, C. (2005). Incremental Learning of Linear Model Trees. *Machine Learning*, 5 - 48.
[13] Quinlan, J. (1993). Combining Instance-Based and Model-Based Learning. *In Proceedings of the 10th International Conference on Machine Learning* (pp. 236 - 243). Morgan Kaufmann.
[14] Witten, I., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques* (2nd ed.). Morgan Kaufmann.

Receieved: April 2012.
Accepted: June 2012.

About the Author

Dima Alberg
Department of Industrial Engineering and Management,
SCE - Shamoon College of Engineering, Beer-Sheva, Israel
e-mail: dimitria@sce.ac.il

Dr. Dima Alberg is lecturer at the Industrial Engineering and Management Department, SCE Shamoon College of Engineering, Israel. He received his B.A and M.A. in Economics and Computer Science from Ben-Gurion University of the Negev. He also received his Ph.D. degree in Information Systems Engineering from the Ben-Gurion of the Negev, Israel. His current research interests are in business intelligence, financial time series data mining, and machine learning.