A Scalable Algorithm for Classification of Streaming Data

Bezzam Varun	312212104020
Kiran Sudhir	312212104044
Mayanka Pachaiyappa	312212104054

BE CSE, Semester 8

T.T. Mirnalinee
Supervisor

Project Review: 1 (10 February 2016)

Department of Computer Science and Engineering
SSN College of Engineering

---

1 Motivation

Stream mining is the capability of extracting useful information from these large datasets or streams of data. New mining techniques are required due to the large volume, velocity and variability of such data. In this project, a fast, scalable architecture based on Very Fast Decision Trees (VFDT) and Early Drift Detection Method (EDDM) is designed for processing stream data. Existing systems that integrate online learning models with drift detection methods suffer from various drawbacks such as the requirement to maintain a variable size sliding window over the data. Another disadvantage is that they assume that a large number of training examples can be stored in memory. Finally, we experimentally compare the proposed system to existing systems using benchmark data, and also demonstrate the utility of the system using real time stream data extracted from Twitter feeds.

2 Problem statement

Since most real world applications today generate data streams on a daily basis, efficient processing of streaming data is the need of the hour. The most common issues that are faced while handling stream data are accuracy, time constraints and storage space restrictions. While the VFDT algorithm provides efficient handling of streaming data, it does not account for concept drift. Concept drift means that the statistical

properties of the target variable, which the model is trying to predict, change over time in unforeseen ways. This causes problems because the predictions become less accurate as time passes. Moreover, concept drift based techniques such as those proposed by Gama et al. do not clearly define how to accurately determine when the error rate increases. However, with an online learning model such as VFDT, even a very small number of training examples can be used to develop the incremental decision tree model, so the error rate can be easily determined. A novel algorithm is developed that integrates the detection of concept drift within VFDT algorithm to handle the Velocity and Variability of the data. In order to handle the volume of the data, we include a preprocessing step within the stream that samples the stream data and produces only a small representative stream of records.

3 Literature survey

Previous work on scaling up decision tree learning produced systems such as SLIQ [1], SPRINT [2] and Rainforest [3]. These systems perform batch learning of decision trees from large data sources in limited memory by performing multiple passes over the data and using external storage. Such operations are not suitable for high speed stream processing. While VFDT works well for online learning of streaming data without storing any examples in memory, it does not handle concept drift. Concept-changing VFDT (CVFDT) [5] was proposed as an improvement to VFDT that is capable of handling abrupt concept drift but does not accurately handle gradual concept drift. However, it requires maintaining a sliding window over the incoming training examples and growing an alternate decision tree over the newer examples. This approach depends on the size of the sliding window and also assumes that the sliding window can always fit in memory.

The Drift Detection Method (DDM) [6] proposed by Gama et al. to solve the problem of concept drift is incapable of handling gradual concept drift. Other systems such as On-Line Information Networks (OLIN) [8] based on info-fuzzy networks adapt automatically to concept drift by repeatedly constructing a new model from a sliding window of latest examples. Repeated construction of a new model is computationally expensive and adversely affects performance. Moreover, similar to CVFDT, such systems make assumptions regarding the size of the sliding window and the ability to store the window of training examples in memory.

3.1 Very Fast Decision Trees

Hoeffding trees were introduced by Domingos and Hulten in the paper Mining High-Speed Data Streams [4]. They refer to their implementation as VFDT, an acronym for Very Fast Decision Tree learner. The key idea depends on the use of Hoeffding bounds. Hoeffding trees are being studied because they represent current state of-the-art for classifying high speed data streams. The algorithm fulfills the requirements necessary for coping with data streams while remaining efficient, an achievement that was rare prior to its introduction. The Hoeffding tree induction algorithm induces a decision tree from a data stream incrementally, briey inspecting each example in the stream only once, without need for storing examples after they have been used to update the tree. The only information needed in memory is the tree itself. The measure used to split a node is the entropy or information gain, given by

$$entropy(p_1, p_2, ..., p_n) = \sum_{i=1}^{n} -p_i \log_2 p_i$$

The Hoeffding bound states that with probability 1-δ, the true mean of a random variable of range R will not differ from the estimated mean after n independent observations by more than:

$$\epsilon = \sqrt{R^2 \ln(1/\delta)/2n}$$

A simple decision tree generated by VFDT can be seen in Figure 1.

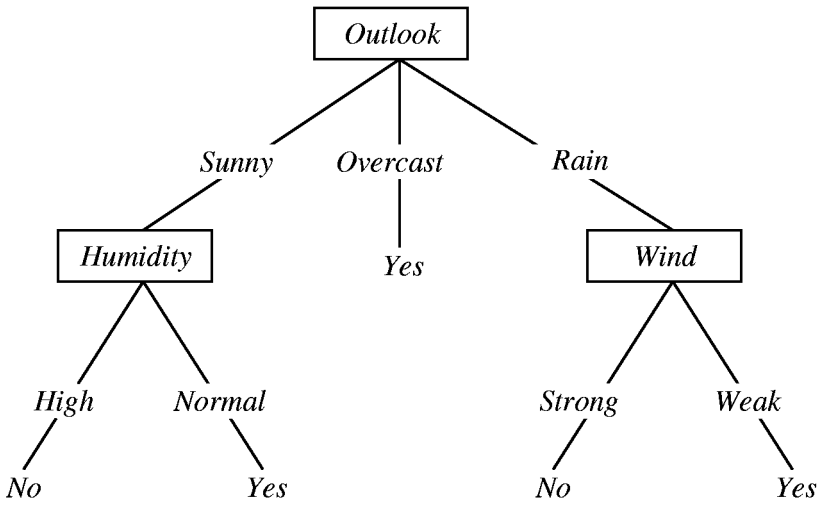

Figure 1: Example of a Decision Tree

3.2 Early Drift Detection Method

The Early Drift Detection Method (EDDM) [7] improves the detection in presence of gradual concept drift. At the same time,it keeps a good performance with abrupt concept drift. The basic idea is to consider the distance between two errors classification instead of considering only the number of errors. While the learning method is learning, it will improve the predictions and the distance between two errors will increase. We can calculate the average distance between two errors (p'_i) and its standard deviation (s'_i). What is stored are the values of p'_i and s'_i when $p'_i + 2s'_i$ reaches its maximum value (obtaining p'_{max} and s'_{max}). Thus, the value of $p'_{max} + 2s'_{max}$ corresponds with the point where the distribution of distances between errors is maximum. This point is reached when the model that is being induced best approximates the current concepts in the dataset. The method defines two thresholds based on the value of $(p'_i + 2s'_i)/(p'_{max} + 2s'_{max})$, henceforth known as Gama's ratio (GR) :
- $GR < \alpha$ for the drift indication threshold. Beyond this threshold, the examples are stored in advance of a possible change of context.
- $GR < \beta$ for the drift confirmation threshold. Beyond this limit, the concept drift is supposed to be true, the model induced by the learning method is reset and a new model is learnt using the examples stored since the warning level triggered. The values for p'_{max} and s'_{max} are reset too.

4 Proposed system

The proposed system presents a novel method to handle concept drift within the VFDT framework. The system architecture is as follows:

4.1 Preprocessing

In the preprocessing step, the stream data is sampled and a small representative stream of records is produced. The sampling step significantly reduces the volume of the data to be processed. In the proposed system, three different sampling strategies have been implemented- Simple Random Sampling, Reservoir Sampling and Stratified Sampling to produce a smaller subset of the input stream data. A Simple Random Sample (SRS) is a sample where every possible subset of sampling units has the same probability of being the sample. Reservoir sampling is a family of randomized algorithms for randomly choosing a sample of k items from a list S containing n items, where n is very large or an unknown number. Reservoir sampling is particularly well suited

to the sampling of streaming data because it ensures that every item has an equal probability of being present in the final sampled dataset. Stratification is the process of dividing members of the population into homogeneous subgroups before sampling. The strata should be mutually exclusive: every element in the population must be assigned to only one stratum. The strata should also be collectively exhaustive: no population element can be excluded. Then simple random sampling is applied within each stratum.

4.2 Decision Tree Based Model

VFDT is used as the decision tree model and use each training example to update the model. The VFDT algorithm also makes a prediction on the target variable for each training example after training.(This is the train followed by test approach to evaluating online learning models) This is compared against the given label of the data to determine whether this is misclassified (ie) an error or not.

4.3 Computation of Errors

In this step, the EDDM method is used to detect errors and identify concept drift. The values of p'_{max} and s'_{max}, p'_i and s'_i are simultaneously updated for each misclassified data (ie) error, where p'_i is the average number of records between two consecutive errors and s'_i is the standard deviation of p'_i. If the no. of misclassified examples seen so far is greater than a specified minimum threshold, usually set to 30 examples, then the concept drift method starts searching for concept drift. The average distance between any 2 errors is p'_i with standard deviation s'_i. If $p'_i + 2s'_i$ is greater than $p'_{max} + 2s'_{max}$, we update p'_{max} and s'_{max}.

4.4 Model Updation

In this module, the model is updated based on whether concept drift is detected or not. If the Gama's Ratio is less than α, then a drift indication threshold is signaled. From here on, it is assumed that there is a possibility that the concept has changed, so all training examples from here on are stored temporarily. If the Gama's ratio becomes less than β, then a new concept is declared and all the training examples stored are used to train a new VFDT, which replaces the old model. However, if after reaching the drift indication threshold, the ratio moves back above .95, the same old VFDT model is maintained and the stored training examples are discarded.

The above approach has not been implemented so far. Moreover, it is a scalable, incremental learning model that automatically adjusts to concept drift based on the structure of the data without making any assumptions on window sizes and other variables. At any point of time, it only stores a single decision tree model (VFDT) and the two online learning measures p'_{max} and s'_{max} in memory. The architecture of the system can be seen in Figure 2.

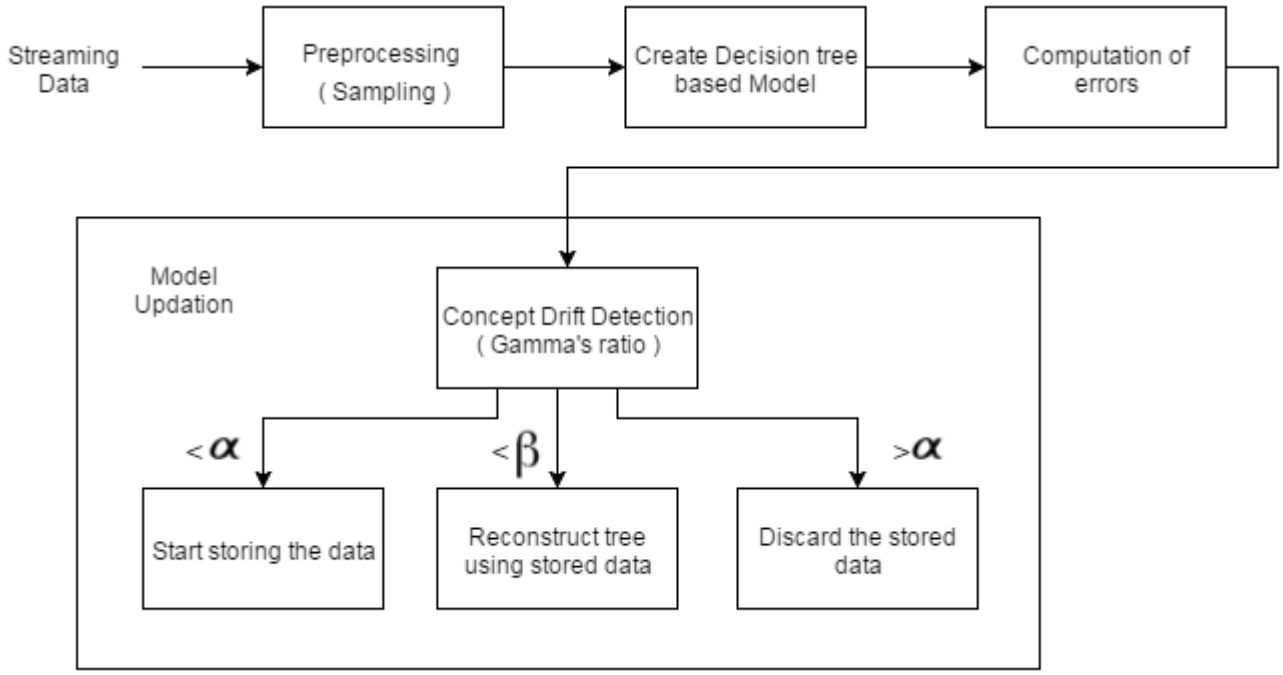

Figure 2: System Architecture

5 Implementation

The preprocessing module and the decision tree creation module have been implemented. A benchmark training dataset (Poker Hand) with 1 million training instances and 25,000 test instances is tested with the proposed system to compare the performance of the system to approaches described in the literature. In the preprocessing step, three different sampling strategies, namely Simple Random Sampling, Reservoir Sampling and Stratified Sampling are implemented to produce subsets of the data of sizes 100000 records and 300000 records respectively. The accuracy and performance of an optimized implementation of the CART algorithm are measured and compared on the representative datasets produced by the preprocessing step. Moreover, three different classifiers are trained using the training dataset, namely CART, Random Forests and a VFDT classifier and the accuracy of these three classifiers trained on the entire

dataset and tested using 25000 test records is compared as well.

6 Results and Analysis

The execution time for the optimised implementation of the CART algorithm reduces significantly on performing sampling. Out of the three samping techniques used, reservoir sampling produced the best results for a sample size of 100,000 records while simple random sampling produced the best results for a larger sample size of 300,000 records. Reservoir sampling was concluded to be the best sampling technique for streaming data because it produced the highest accuracy for 100,000 records and also improved performance by taking the least time to execute.

Out of the three decision tree classifiers implemented, RandomForests has the highest accuracy at 71 percent. However, it cannot be extended easily to handle streaming data. Out of VFDT and the optimised CART, VFDT achieves almost similar accuracy to the CART algorithm. However, VFDT is preferred because it is better suited for streaming data.

A detailed comparison of the sampling strategies can be seen in Figure 3

Sampling Strategy	Size of the dataset		Classification Accuracy		Time to execute	
No Sampling	1,000,000 records		66%		10.96 seconds	
Random Sampling	100,000 records	300,000 records	38%	60%	0.72 seconds	2.24 seconds
Reservoir Sampling	100,000 records	300,000 records	56%	59%	0.6 seconds	2.95 seconds
Stratified Sampling	100,000 records	300,000 records	56%	57%	0.74 seconds	2.80 seconds

Figure 3: Comparison of Sampling Strategies

A comparison of various classification algorithms on the full Poker Hand dataset is shown in Figure 4

Classifier Used	Accuracy
VFDT	65.2%
Decision Tree	66%
Random Forest	71%

Figure 4: Comparison of different classifiers

References

[1] M. Mehta, R. Agrawal, and J. Rissanen, *SLIQ: A fast scalable classifier for data mining*, In Proc. of the Fifth Intl Conference on Extending Database Technology (EDBT), Avignon, France, March 1996.

[2] J. Shafer, R. Agrawal, and M. Mehta. *SPRINT: A scalable parallel classier for data mining*, VLDB 1996.

[3] J. Gehrke, R. Ramakrishnan, and V. Ganti,*Rainforest-A framework for fast decision tree construction of large datasets*, VLDB 1996.

[4] Domingos, P., Hulten, G., 2000, *Mining high-speed data streams*, In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '00).

[5] G. Hulten, L. Spencer, and P. Domingos, *Mining time-changing data streams*, In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '01).

[6] J. Gama , P. Medas, G. Castillo and P. Rodrigues, *Learning with drift detection*, In SBIA Brazilian Symposium on Artificial Intelligence, 2004.

[7] M. Baena-Garcia Jose, J. Del Campo-vila, R. Fidalgo , A. Bifet, R. Gavald and R. Morales-bueno, *Early drift detection method*, In SBIA Brazilian Symposium on Artificial Intelligence, 2005.

[8] L. Cohen, G. Avrahami, M. Last, and A. Kandel, 2008, *Info-fuzzy algorithms for mining dynamic data streams*, Appl. Soft Comput. 8, 4 (September 2008), 1283-1294.