ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Special Issue 14, May 2016

Adaptive Modelling for Streaming Data Classification

Srilakshmi Annapoorna P.V
PG Scholar
Department of CSE
SSN College of Engineering
Chennai, India
annu7552@gmail.com

Mirnalinee T.T
Professor
Department of CSE
SSN College of Engineering
Chennai, India
mirnalineett@ssn.edu.in

Abstract—In streaming data scenario, making analysis of data is really challenging. Concept drift should be handled effectively. When data arrives in the form of stream, the characteristics of the data changes over time. In case of ensemble methods, models are updated by constructing new models for new data and redistributing the instances that belong to old model according to the change in pattern. Memory requirements are bounded while handling streaming data. As data keep on coming and models are constructed continuously, memory overhead may occur. This can be handled by discarding a poorly performing classifier from an ensemble. While discarding a poor classifier, the accuracy of classification is also improved since it discards a poorly performing classifier. And also it solves the problem of memory overhead. Testing time is also the important factor to be considered in streaming scenario. By discarding, the overall ensemble size decreases to some extent and hence testing time also reduces. A novel algorithm for dynamically discarding the classifier in random forest construction at the time of training is proposed. Experimental analysis have made using random forest algorithm and observed that discarding the classifier leads to improvement in overall accuracy of random forest.

Index Terms—Streaming data, Adaptive model, Tree pruning, Random Forest, Classification.

I. Introduction

Data stream mining is the process of deriving meaningful information from the data that is arriving in the form of streams. In streaming scenario, the nature of data that arrives is unknown, hence gathering knowledge and performing classification is more difficult. Concept drift is the major factor that affects the classification performance while talking about data streams. Changes in pattern of data is highly probable. Models need to be adapted accordingly as the concept change occurs. As streaming data analysis is performed online, the memory available to store the data to be processed is bounded. So there is high chance of facing memory overhead problems. Those problems must be addressed by using efficient strategy. At the same time the accuracy of the classification must be improved. In order to solve those problems various methods are available. But still those methodologies needs to be enhanced in order to provide better results.

Some of the existing system are discussed as follows. In [5] , an online random forest algorithm has been proposed. Bagging and decision tree growing procedure has been given. They have used out of bag estimate in order to discard the trees online. While constructing trees in online, the trees are randomly discarded where the probability of discarding a tree depends upon the out of bag error value. In [1], to address the problem of concept drift, feature importance metrics such as Mean Decrease in Accuracy(MDA) and Mean Decrease in Gini impurity(MDG) has been adapted to online random forest. By using MDA as metric, it permutes the class values for the incoming test observations and derives confusion matrix for each observation, when there is no change in accuracy it is treated as unimportant feature. By periodically checking the validity of splits it prunes the split which is very impure based on gini index. Tree discarding is performed here by using random probability function where the function depends on tree's test stream error rate.

In [3] , basic classification of pruning methodologies are discussed. Static pruning and dynamic pruning are the major classification in pruning methodologies. In static pruning, trees are pruned if and only if all the models are constructed and can provide effective classification. But in dynamic pruning, tree pruning is performed online while constructing the trees. Each time a new tree is constructed it checks for the the classifier that is performing poorly based upon some heuristic strategy.

In [4] , global pruning methodology has been proposed. Leaf vector and indicator vector are used to get information about predictions and whether a data point belong to the corresponding leaf or not respectively. Pruning can be performed by combining least significant leaves which is chosen based on the norm values of the leaf vector of each leaves. After combining, the statistics of new leaves are updated through the indicator vector by removing the values of old leaves and updating the values of new leaves.

In [2] , comparision between weighted majority voting and majority voting has been performed. The trees here are weighted based on the concept of margin functions. Margin funtions here describes, how much extent a classifier will have more votes for an example to be classified as right class. Based on this, trees are given weights and only those trees are used for making decision of final vote. The weights of tree also based on out of bag estimate. Better classification results are obtained through weighted majority voting rather using simply

majority voting.

II. Background Knowledge

Data streams flow with high velocity and flow of data is continuous. When compared to the data generated in traditional static scenarios, streaming data are quite different. They have data distributions that modify with time and they are unbounded. Data streams are generally generated by many applications such as network monitoring and traffic management, click streams of web , data from sensor applications, email messages, and others. Data generation has become rapid and is expected to be very huge which leads to designing of new techniques in order to handle huge data. Storage, communication and computation capacity in the computing systems are questioned by this quick production of non-stop data streams. It is quite demanding to store, mine and query these data sets. The great challenge in mining data streams are about pulling out knowledge structures present in models and patterns in the uncontrollable streams of information. The demands and the research problems faced in the data stream mining is highly motivational. In response to the continuous data problem, the data stream paradigm has recently emerged. Data stream classification involves algorithm being written which can naturally cope with data sizes many times greater than memory, and can be extended to real-time applications which was not previously tackled by machine learning or data mining and is considered to be more challenging. The training examples can be briefly processed a single time only, then those examples must be discarded to make space for subsequent examples which is the core assumption of data stream classification. There is no control over the order of the examples seen, and the algorithm processing the stream must update its model incrementally as each example is processed. It is very difficult to process the huge data in a short period of time. Hence preprocessing is required in order to construct a generic model and test the accuracy of the model using test data. There are many challenges to be faced while handling high velocity data and those challenges must be addressed optimally in order to improve the efficiency.

A. Issues in mining datastreams

Mining datastream means deriving useful information from the continuosly arriving data. Extracted knowledge can be useful for many purposes such as business intelligence etc. There are several challenges to be handled while processing the continuous streams. This lead to design and enhancement of existing algorithms to make it adaptable to handle high velocity data and improve the accuracy of algorithms still more. Some of the challenges faced generally while handling streaming data is listed below.

1) Handling higher velocity and continuos data: Streaming data is referred to as continuous arrival of data at higher speed. Managing data is one the most significant concern in streaming scenario. Traditional systems for data management cannot handle such higher data rates. Keeping all such data in the media is unreasonable. Also, It is very high-priced to check the data several times . Capturing those data and processing it online is much difficult and challenging.

2) Data reduction and synopsis construction: The data stream analysis, classification, querying and clustering applications will need some type of specification techniques to comply with the earlier mentioned problems. Fairly accurate answers from huge data sets generally by means of synopsis construction and data reduction can be acquired using these techniques. By choosing a subset of incoming data or by making use of sketching, aggregation techniques, load shedding, this can be done.

3) Memory requirements for managing Unbounded data : In streaming scenario, data is huge and boundary conditions for size of data does not exists. Memory constraints are significant challenge need to be faced, Since processing data, as it arrives, deal with storing those data in main memory instead of secondary memory.

Data streams have unbound data, so the storage that can be used to find or preserve frequent item sets is inadequate. It is extremely essential to design space effective techniques that can have one look or less over the incoming stream because of this large number of produced streams. Based on time parameter, the frequency of an item set depends; This is another result of unbounded data. From unbounded data, it is very demanding to use inadequate storage and find dynamic frequent item sets.

4) Handling Concept drifts: The nature of data stream keeps on changing. So, the model should be able to adapt to such changes quickly. This change would help many temporal-based analysis applications like emergency and video-based surveillance, disaster recovery. Concept drift occurs when statistical properties of data changes vary time to time. Such drifts needed to be detected to adapt the models to quick changes. When a drift occured has not taken into account it will lead to poor results and accuracy.

5) Handling queries and response-time: Analysis on recent data and find the appropriate information corresponding to the query raised and returning the result back are some of the requirements of user defined querying. In short period of time these processing must be done as data is lost after a particular time interval when new data arrives. All memory and time constraints must be taken into account and algorithm must be designed to handle such scenarios and work effectively. As the data stream applications are very time specific, there is a demand on response time. Algorithms that come slower than the data arriving rate in constrained situations are of no use.

6) Visualization of results: Research is still on in revelation of traditional data mining results on a desktop. It

is a real challenge to see visualization in the small screens of the PDA. If a businessman is seeing the results of data being streamed and analysed on his PDA, the results should be so effective that it should facilitate him to take quick decisions.

III. Proposed System Design

To address the challenges related to memory and model adaptation, new system has been proposed. Initially data has

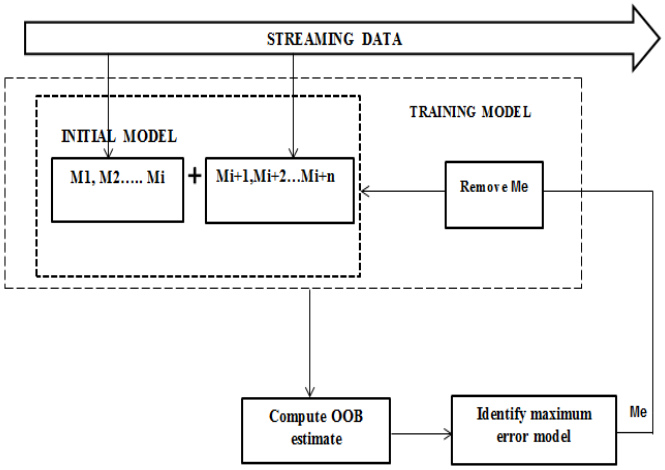

Fig. 1. Adaptive training model construction

to be preprocessed in order to provide higher quality samples. Effective preprocessing techniques must be used to improve the accuracy of classification. The architecture diagram of the proposed system has been illustrated in the figure 1.

Model construction is the most important step in classification. In Streaming Scenario, the pattern of incoming data will change from time to time. Adaptive model construction is needed. Adaptive model construction here refers to the discarding the outdated models and when new data arrives, constructing the new models that adapt to change in pattern. Also we have to make sure that models must be adapted according to frequent concept drifts that occur in streams. Model must be adapted quickly and accurately. Random forest have been used to construct model which uses tree as base classifier. Initially, some n models are constructed with the arriving data. Then based upon the rate at which data arrives and size of data, number of examples to construct a single tree is choosen. When number of examples that arrives become equal to number of size of data with respect to its rate, then a new model is built for the next set of examples. At each time, a model is created, OOB estimate is calculated for existing trees and poorly performing classifiers are discarded. Models M1 to Mn are constructed and combined toghether for classification.

Ensembling strategy is used in order to perform classification which combines the output of each classifier by averaging or majority voting strategy.

A. Adaptive Modelling

In Adaptive modelling, when data streams arrive the concept and characteristics of data probably will change from time to time. When characteristics of data is changed, the old model which is constructed using old data classifies poorly for the new data. Hence the over all performance of classifier degrades. In order to handle this, an efficient methodology must be designed so that outdated classifiers are discarded and new well performing classifiers are added to model. There are various ways handling concept drift. Some of the ways are incrementing model and discarding trees that are outdated. Incrementing model deals with finding the concept drift from incoming stream of data at first. When occurrence of drift is identified, then distribution of tuples in the tree which is poorly classifying is redistributed according to the concept change. Model has been incremented over here. The second is discarding an outdated classifier. Based on some heuristic measure, each classifiers performance is evaluated and some strategy is used to remove the under performing classifier so that model is being adapted to changes in concept of data.
While dealing with data streams, Concept drift is the most important factor to be considered in order to handle high velocity data with changing characteristics. Concept drift is defined as the change that occurs in the characteristics of incoming data time to time. There are many kinds of drifts to be considered while handling high velocity data like gradual drift, sudden drift, blip drift etc. When pattern of data changes model needs to be refined accordingly.

B. Random Forest with Adaptive Modelling

Random forest is an ensemble based classification algorithm. Ensemble is the concept of constructing set of classifiers and when an input data is arriving, each classifier predicts the class of incoming example and finally label of the class is decided by averaging or voting. As streaming data deal with concept evolution, learning and adapting the classifier according to change in concept is necessary. This indicates that model needs to be updated. Model updation is performed by constructing new trees when new pattern arises in stream of incoming data. The base classifier used in random forest is Decision tree. Since random forest deal with ensemble approach, while the data arrives in the form of stream it starts constructing trees. Each tree is refered as a model. When there exists a concept drift it can be handled by constructing new tree with the examples in the oudated tree and new examples that arrive. When new tree is constructed, each time the memory overhead increases, hence we have to go for discarding models in ensemble based on certain criteria. Discarding or pruning the poor performing classifier is one of the way where oudated trees are rejected and over all performance is improved. And the other way to adapt the model is, consider some K trees are constructed then there exists a situation where all K trees or maximum number of trees are performing poorly for the upcoming stream then, a new model has to be constructed in such a way that examples that are available in the old trees are stored and must be

redistributed according to new pattern by construction of new tree.

1) Testing: Prediction of an incoming has to done based on the constructed model. Since set of classifiers is used to build random forest it must aggregate the predictions by using the following methods .

Averaging

The prediction is given by average of values while considering the real value.

Majority voting

Vote for a single class is derived from each tree's prediction. The class which receives the most votes is said to predicted label.

C. Pruning and Discarding trees

While constructing ensemble of classifiers using streaming data, multiple trees are constructed and the output for test data obtained by majority class. Hence the over all performance of the classifier always depends on the individual performance of each tree. The performance of each tree can be evaluated using the Out of Bag error estimate. This measure is used to identify each tree's performance. The larger the estimate value denotes that performance of tree is becoming poorer. Such trees are also called as outdated trees. When data arrives continuously in the form of streams the pattern of data gets changed time to time which results in concept evolution. When concept drift occurs the old model which are constructed may get outdated and hence those models need to be discarded in order to reduce the effect of poor classification.

1) Out-of-bag error: Out of bag estimate is defined as the prediction error determined by testing the tree with training sample that do not contain the bootstrap aggregate. And hence error or performance of tree in terms of error is obtained. Each tree can be tested with this estimate to obtain prediction error of the tree. By performing evaluation on the observations which are not used to build the tree, prediction performance of each tree can be determined which is useful to take important decision. There is no need of separate cross validation sets in order to get the unbiased estimate of test set. This is one of the advantage in random forest.

During the run of random forest this is internally measured. By using different bootstrap sample from incoming data each time this is internally measured during run of random forests. In the construction of kth tree, about one third of the bootstrap samples are left out and not used in the construction. Hence to get a prediction error, classification is performed using the left out samples by passing down those samples to the kth tree. In this way, a test set classification is obtained. In our module we are implementing the concept of pruning when the maximum error value occurs repeatedly. The reason to consider the repetition of error is, in most of streaming scenario there will be a sudden drift in data which occurs only once or twice. At that time choosing the maximum error and discarding will become a local maxima. And hence it has to be checked for certain number of repetitions. When it is found that certain error occurs repeatedly, then it must be removed to get improvement in classification performance.

Just straightly taking the maximum error is not an optimal strategy because it highly tends to struck in local maximum and also in streaming data analysis the error once occurred need not to occur other time since characteristics of data changes over time. Hence if and only if the error occurs more than a specific number of times that corresponding tree should be discarded.

IV. Experimental results and Analysis

Our implementation for preprocessing has been carried out using random forest algorithm in spark with preprocessing techniques such as sampling and filtering. It has carried out using ubuntu os on a Intel core i3 processor with memory requirements 2GB. Scala language has been used in spark. In order to achieve parallel processing. In order to implement the pruning algorithm in random forest, eclipse is used.

We have used bench mark dataset name Forest Cover Type which consists of 581012 instances with 54 attributes which in turn contains 10 quantitative variables, 4 binary wilderness areas, 40 binary soil type variables and no missing attributes. This data set contains 7 class distributions.

Random forest is implemented in java using eclipse IDE. Sampled data is given as input.

Our goal is to make sure that the classification accuracy will not be affected even when the model is build using samples and filtered tuples. Random forest algorithm has been used to build the model. In this algorithm, while building each tree, error for each tree is calculated using out of bag estimate in order to evaluate the performance and finally calculates the overall accuracy of random forest. This algorithm also calculates the number of correctly classified examples for each tree build.

Our experiment has been carried out with forest cover type data set. When data arrives in the form of stream, tree construction starts by building each tree for each set of incoming data. In our experiment, we fixed the total number of trees as 75. We fix intial number of trees to be 10. Taking smaller values for fixing initial number of trees is better, because it checks OOB estimate for more iterations. And after some k trees constructed, at the time of $k+1^{th}$ tree construction out of bag estimate is calculated for all trees constructed so far. And then the maximum OOB estimate for the current iteration has been stored in a set. Then, from the set that contains maximum OOB error, the highest value is taken and checked for the occurence of repetition. If the same error occurs more than specific number of time times, then the corresponding classifier is discarded. This process is continued for each time when a new tree is built. Each time new tree construction starts based on time interval and size of incoming data. While building a tree as data arrives, then new tree construction take's place based on sufficient data arrival rate and time interval. And the number of repetition for an error to occur is decided emperically by running the algorithm with values of number

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Special Issue 14, May 2016

of occurrences such as 2, 3, 4, 5, 6, 7 and finally found that discarding a tree with maximum error repeated 3 times is more optimal than other values.

A. *Analysis of results*

The implementation has been carried out in eclipse IDE using Forest cover type data type with small subset of 5762 training examples and test data with 2154 examples. The results are shown as follows.

No. of runs	Accuracy with discarding	Testing time for with discarding (milli seconds)	Accuracy without discarding	Testing time for without discarding(milli seconds)
1.	67 %	123	66 %	155
2.	67 %	127	65 %	136
3.	69 %	127	66 %	156
4.	67 %	123	67 %	139
5.	67 %	117	65 %	140
Average	67.4 %	123.4	65.3 %	145.2

TABLE I
TEST ACCURACY

The experiment has been carried out five times for each with discarding and without discarding trees and it has been inferred that random forest has been giving little more accuracy if trees with with maximum error have been discarded. When there is no occurrence of concept drift in data then discarding classifiers may result in decrease in the classification accuracy. The proposed algorithm can be applied to data set containing more concept drifts. The recorded results of accuracy for with and without discarding can be visualized by the following figure 2.

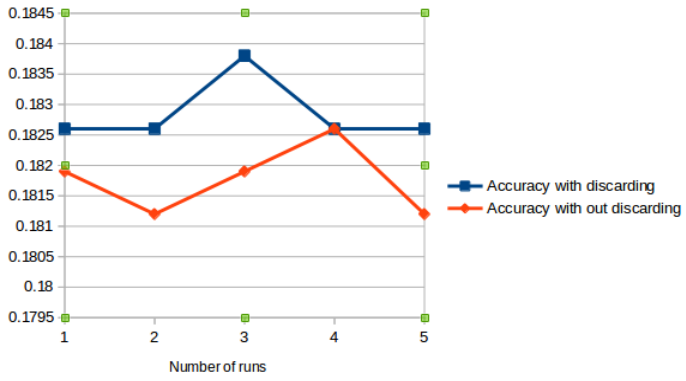

Fig. 2. Result Analysis

V. CONCLUSION

In considering streaming scenario, the characteristics or properties of data change over time. Hence model needs to be adapted according to the current characteristics of incoming data. Thus an algorithm for updating the model by adding trees to existing model and removing trees from the existing model has been proposed. Based upon the maximum tree error the discarding has been performed. Experimental analysis have been made with forest cover type data set and it is found that there is an improvement in the accuracy of classifier with testing time. The accuracy of the classification could be further improved by detecting the concept drift.

REFERENCES

[1] A.P. Cassidy and F.A. Deviney. Calculating feature importance in data streams with concept drift using online random forest. In *Big Data (Big Data), 2014 IEEE International Conference on*, pages 23–28, Oct 2014.
[2] M. El Habib Daho, N. Settouti, M. El Amine Lazouni, and M. El Amine Chikh. Weighted vote for trees aggregation in random forest. In *Multimedia Computing and Systems (ICMCS), 2014 International Conference on*, pages 438–443, April 2014.
[3] V.Y. Kulkarni and P.K. Sinha. Pruning of random forest classifiers: A survey and future directions. In *Data Science Engineering (ICDSE), 2012 International Conference on*, pages 64–68, July 2012.
[4] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. Global refinement of random forest. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 723–730, June 2015.
[5] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof. On-line random forests. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1393–1400, Sept 2009.