DECISION TREE ENSEMBLE TECHNIQUES FOR DATA STREAMS CLASSIFICATION

Monika arya
Dept.of CSE
BIT, Durg

Dr.G. Hanumat Sastry
School of Computer Science
UPES,Dehradun

Abstract

The technological advancement has led to the outburst of data. There has been trouble in managing and analyzing such voluminous data streams with the traditional data classification approaches. Decision trees are commonly used, most efficient and well known classification method for data streams. Data stream classifiers are different from traditional classifiers as they need to adapt unforeseen changes in data streams. These unforeseen changes in data streams are termed as concept drifts. Ensemble classifiers have therefore become an interesting research area in Data stream mining due to the fact that they offer a natural way to adapt changes due to their modular nature. An Ensemble is a group of base classifiers which are combined with an aim to achieve better accuracy and performance than those of achieved by single classifier. A forest is an ensemble whose members are learned by decision tree learning method [26]. Two of the most popular techniques for constructing ensembles of decision tree are bagging and boosting [25]. Both of these methods operate by taking a base learning algorithm and invoking it many times with different training sets. Despite its remarkable performance, these ensemble methods have certain limitations. These limitations restrict their applicability in mining data streams with concept drift. Hence, this study comprehensively compares the decision tree ensemble techniques adapted for data streams .The study also surveys the effect decision tree pruning parameters on its accuracy and efficiency. Future research directions in this field are determined based on opportunities of pruning a forest of decision trees. These research directions facilitate the development of an ensemble classifier that can efficiently classify data streams and can adapt to changes and drifts in the evolving stream.

Keywords: Data streams, decision tree, ensemble classifiers, concept drift, forest

1 INTRODUCTION

The traditional data was limited and manageable by relational data base management system whereas the amount and speed at which world is creating data stream is unlimited. These data streams can be characterized by huge volume, variety, veracity and velocity. The underlying difference between the traditional data and data streams led to a distinct set of technological approaches for analyzing and managing the data streams [27]. Classification algorithms for data streams also have new requirements in terms of memory usage, processing time, single scan of incoming data etc as compared to traditional static data. Data stream classifiers are different from traditional classifiers as they can adapt to unforeseen changes in stream's of data. These unforeseen changes are known as concept drift. Concept drift is the change in characteristic of streams and target concepts over time due to various conditions. On the basis of its influence on classification performance, concept drift are of two types: virtual drifts and real drifts [28]. Virtual concept drift has no impact on the decision boundaries thus do not directly influence the classifier being used. Whereas the real concept drift has impact on decision boundaries and thus may significantly influence on the performance of the classifier. Another classification of concept drift is based on severity and speed of changes. The concept drift can be of five types: sudden drift, incremental drift, gradual drift, recurring drifts and blips. There can also be mixed concept drift which can exhibit hybrid characteristics. The presence of concept drift deteriorates the predictive accuracies of the classifier as they affect the properties of the classes that are used to train the current classifiers and thus may result in accuracy drop of the classifier over time. Some real life examples of concept drifts are spam categorization, weather predictions, monitoring systems, financial fraud detection etc. In literature, three general solutions have been proposed to handle the presence of concept drift[1]:-

1. Rebuild the classifier every time new instance arrives.

2. Monitor the changes in characteristics and update the model.

3. Using adaptive learning algorithm that can adapt to new instance and forget old ones.

Decision trees are commonly used, most efficient and well known mining method in classification of data stream. The decision tree has many real world decision making applications like radar signal classification, credit approval, medical diagnosis, weather prediction, fraud detection, and customer segmentation etc. A decision tree consists of three main parts: nodes, leaves, and edges. Each node represents an attribute by which the data is to be partitioned. Number of edges can emerge from Each node. Edges are labeled according to possible values of the attribute. An edge connects either two nodes or a node and a leaf. Leaves are labeled with a decision value for categorization of the data. To make a decision using a decision Tree, start at the root node and follow the tree down the branches until a leaf node representing the class is reached. Each decision tree represents a rule set, which categorizes data according to the attributes of dataset. There are many advantages to the use of decision trees for classification task: Rules generation using decision tree are easy to understand, the tree size is independent of the database size, and the tree can be constructed for large dataset with many attributes. The time and space complexity of constructing a decision tree depends on the size of the data set, the number of attributes in the data set, and the shape of the resulting tree. Over-fitting is one of the major difficulties for decision tree. Growing the tree beyond a certain level of complexity leads to over-fitting. Pruning a decision tree help us to avoid over-fitting. There are several approaches to avoid over-fitting in building decision tree.

- Pre-Pruning-This method stop growing the tree earlier, before it perfectly classifies the training set.
- Post-Pruning-This method allows the tree to perfectly classify the training set, and then post prune the tree.

We can avoid overfitting by changing the parameters like max_leaf_nodes (reduces number of leaf nodes), min_samples_leaf(restrict the size of sample leaf) and max_depth (reduce the depth of the tree to build a generalized tree). These parameters thus act as pruning parameters. By tuning these parameters the size and accuracy of the decision tree model can be controlled. The algorithms to build a decision first create a tree and then prune the tree to reduce its size without compromising with its performance. Pruning removes a portion of tree to reduce the overall size of the tree thus reducing the complexity of the decision tree. The challenge of pruning is to reduce the size of the base classifiers and finally the ensemble by still maintaining or even improving, the performance of the ensemble. There are various parameters for pruning the decision tree like max_depth,max_leaf_nodes etc.

Ensemble of decision trees is a more promising approach in data stream classification in which several decision trees are combined to produce better predictive performance than utilizing a single decision tree. The main principle behind the ensemble model is that a group of weak learners come together to form a strong learner. Ensemble classifiers have become an interesting research area in Data stream mining due to the fact that they offer a natural way to adapt changes due to their modular nature. Ensemble classifiers can adapt by adding new classifiers components which are trained on recent data and by removing the old classifier components which are based on outdated concepts. Despite its remarkable performance, ensemble methods have certain limitations to its applicability in stream data mining. As combining large number of classifiers to build an ensemble model also brings on large computational requirements, including the training costs, the storage needs and the prediction time. The poor prediction accuracy of the base classifier can negatively affect the overall performance of the ensemble classifier. There are various techniques to build the ensemble of the decision tree. Few of them are:

- Bagging-Bagging technique employs building multiple models (typically of the same type) from different subsamples of the training dataset. **T**he training set is divided into number of subsets and each subset is used to train their decision tree. As a result, we end up with an ensemble of different models. Now average of all the predictions is used for final prediction. In this way the ensemble model is more robust than single decision tree.
- Boosting-**Boosting** is another ensemble technique in which the classifiers learn sequentially by learning from errors made by previous model. So boosting employs building multiple models (typically of the same type) each of which learns to fix the prediction errors of a prior model in the chain. By combining the whole set at the end converts weak classifier into better performing model.

In this paper we will experimentally compare the advance bagging and boosting algorithm which are specifically designed to classify data streams and study the effect of tuning different parameters on accuracy and overall efficiency of the decision tree classifier.

II RELATED WORK

Many learning algorithms were used as a base models in ensemble classifiers for handling concept drift. One of the most popular is the decision tree. Domingos and Hulten proposed algorithm called a Very Fast Decision Tree (VFDT) [1].VFDT is capable of growing decision tree from streaming data but it has a limitation that this model works without any explicit detection of changes i.e. blind adaptation and deals with only categorical attributes. Gama, J. et al. in his work proposed another extension of VFDT called VFDTc. It is able to deal with categorical as well as numerical attributes [4]. Hulten, Spencer and Domingos [3]in their paper presented an improved version called a Concept- adapting Very Fast Decision Tree learner (CVFDT). CVFDT is able to adapt to concept-drift in streams. Jankowski, D. et al.[2] in his work addresses a data mining task of classifying data stream with concept drift. He proposed an algorithm, named Concept-adapting Evolutionary Algorithm for Decision Tree that does not require any knowledge of the environment such as numbers and rates of drifts. The novelty of the approach is combining tree learner and evolutionary algorithm, where the decision tree is learned incrementally and all information is stored in an internal structure of the trees population. Chen et al. [7] studied the task of online boosting--combining online weak learners into an online strong learner which rely on the smooth boosting algorithm. These algorithms are not adaptive as they require prior knowledge of γ as a parameter .Beygelzimer et al.[6] designed an adaptive online boosting algorithms using the theory of online loss minimization and does not require knowing γ in advance. The OzaBagADWIN algorithm proposed by Bifet et al. [22] is a Bagging algorithm adaptation. The idea of this proposal is to add a drift detector called Adaptive Windowing (ADWIN) [23] to the incremental version of the Bagging algorithm [21]. The adaptation mechanism is based on replacing the worst of the classifiers in an instant of time with a new base classifier created more recently. The Adaptive-Size Hoeffding Tree (ASHT) [24]is derived from the Hoeffding Tree algorithm with few changes like fixed maximum size of the tree and if the size exceeds the maximum limit than the size is reduced by deleting some nodes. Gomes, H.M. et al.[5]In their work presented the adaptive random forest (ARF) algorithm. ARF can adapt with different types of concept drifts by using an effective re-sampling method .

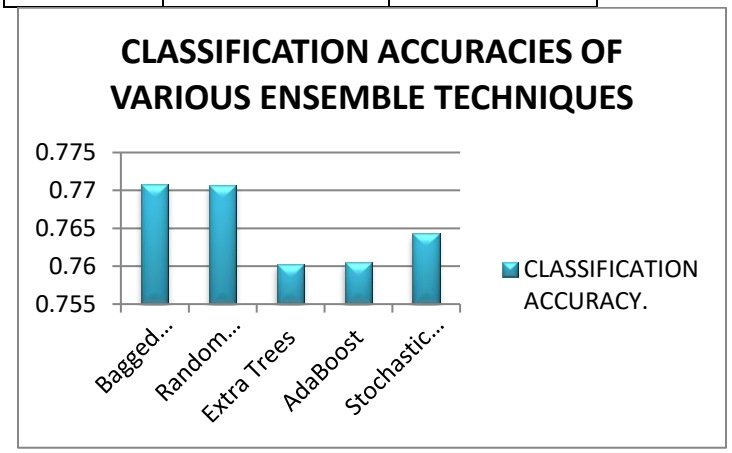
III EXPERIMENTAL SETUP

Initially, the various traditional algorithms of Bagging and boosting are compared in terms of classification accuracy. These experiments were performed on data set in python. The prima Indians onset of diabetes data is used from UCI Machine Learning Repository for each ensemble algorithm. The algorithms used 10 fold cross validation. It is a standard technique used to estimate the performance of any machine learning algorithm on unseen data. It is a binary classification problem where all of the input variables are numeric.

TABLE 1

CLASSIFICATION ACCURACY OF TRADITIONAL ENSEMBLE TECHNIQUES

ENSEMBLE TECHNIQUE	MODEL USED	CLASSIFICATION ACCURACY
BAGGING	Bagged Decision Trees	0.770745044429
	Random Forest	0.770727956254
	Extra Trees	0.760269993165
BOOSTING	AdaBoost	0.76045796309
	Stochastic Gradient Boosting	0.764285714286



The next set of experiments were performed to compare many advance algorithms of bagging and boosting like OzaBagADWIN, ASHT ,meta-OzaBag

with various adaptations to classify data streams and experimentally tested their performance (accuracy, precision, evaluation time, F1 Score and recall value).

The OzaBagADWIN algorithm proposed by Bifet et al. [22] is a Bagging algorithm adaptation. The idea of this proposal is to add a drift detector called Adaptive Windowing (ADWIN) [23] to the incremental version of the Bagging algorithm [21]. The adaptation mechanism is based on replacing the worst of the classifiers in an instant of time with a new base classifier created more recently.

The Adaptive-Size Hoeffding Tree (ASHT) [24]is derived from the Hoeffding Tree algorithm with few changes like fixed maximum size of the tree and if the size exceeds the maximum limit than the size is reduced by deleting some nodes. The idea behind reducing the size is the concept that smaller trees adapt more quickly to changes. When the tree size exceeds the maximum size value either the oldest node is deleted or all the nodes are deleted and re-started from the first node.This new method was adopted for improving the performance of bagging as it increases the diversity among trees.

The meta-OzaBag is an Incremental on-line bagging of Oza and Russell [21] which is an online versions of bagging and boosting for Data Streams. Their study was based on the observation that the probability of an example chosen from training data for replication will tend to aPoisson(1) distribution .The RandomTreeGenerator is used to generate a stream based on randomly generated tree. ConceptDriftStream is used to add concept drifts to the examples in stream.

All experiments are carried out in Moa. The Ensemble size was fixed to 10 and number of evaluation instances for each ensemble classifier was taken 100000. The base learner used was Hoeffding Tree.

TABLE 2

PERFORMANCE COMPARISON OF DIFFERENT ENSEMBLE CLASSIFIERS

CLASSIFIERS	Purpose	Evaluation Time (CPU Sec)	Classification Correctness(%)	F1-Score (%)	Precision (%)	Recall (%)
Multilabel-meta-OzaBagAdwin	Bagging of evolving data streams using Adwin	38.6258	84.8999	84.7989	85.1	84.5
Multilabel-meta-OzaBagADWIN	Bagging of evolving data streams using Adwin	21.9181	91.2222	91.024	91.149	90.9
Meta-OnlineSmoothBoost	Incremental on-line boosting	24.180	91.3322	01.374	91.45	91.3
Meta-OzaBag	Incremental on-line bagging	19.8277	92.0	91.55	91.75	91.45
Meta-OzaBagASHT	Bagging using trees of different size	24.2113	89.222	89.1466	89.7	88.6
Multilabel-meta-OzaBagAdwinML	Bagging of evolving data streams using Adwin	31.761	84.899	84.7989	85.1	84.5
Meta-OzaBag Adwin	Bagging of evolving data streams using Adwin	23.290	92.00	91.599	91.75	91.45

The next set of experiment is to study the affect of different parameter on accuracy and efficiency of the decision tree classifier.The data set used in the experiment was generated to model psychological experimental results. It is a multivariate data having categorical attributes and the number of attributes is 5.Number of instances is 625. These experiments were performed on data set in python. In our experiment we have tuned the pruning parameters like max_leaf_nodes, min_samples_leaf and max_depth. Experiments show the accuracy and size

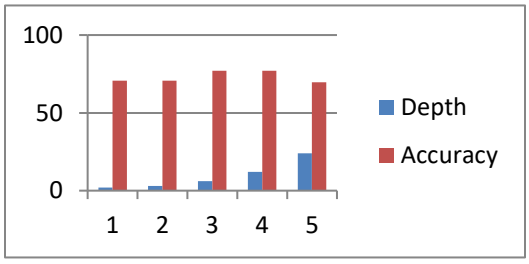
of tree for particular parameter. We have also compared the other efficiency measure of classifiers like precision,recall f1-score and support.Table 3 shows the affect of tuning the parameters on the accuracy and overall efficiency of the decision tree classifier.

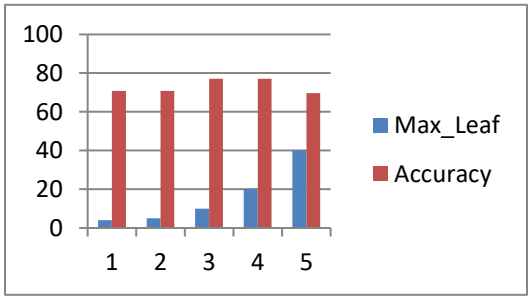
TABLE 3

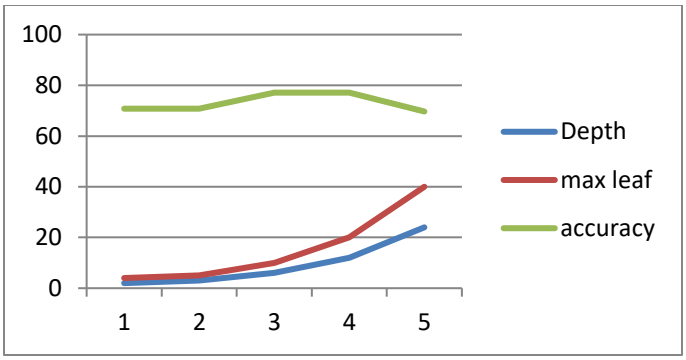
EFFECT OF TUNING PARAMETERS

(Results Using Entropy)

Depth	Max_Leaf	Accuracy	Precision	Recall	F1-score	support
2	4	70.74468085106383	0.66	0.71	0.68	188
3	5	70.74468085106383	0.66	0.71	0.68	188
6	10	77.12765957446808	0.74	0.77	0.76	188
12	20	77.12765957446808	0.72	0.77	0.74	188
24	40	69.68085106382979	0.67	0.70	0.67	188







IV CONCLUSION

In this paper we have experimental comparison of the traditional decision tree ensemble techniques and it was observed that different techniques have different level of accuracy and efficiency. The experimental results shows that the classification accuracy of bagging technique using random forest is better as compared to other algorithms of baaging and boosting. Apart from traditional techniques for static dataset, experiments were also performed on the data streams by various algorithms which are specifically adapted to handle concept drift and their performance was compared. The experiments shows that the algorithms using fixed size of decision tree (like Multilabebel-meta-OzaBagML, Meta-OnlineSmoothBoost, Meta-OzaBag) perform better than the algorithm having different size of decision trees(Meta-OzaBagASHT). We have also done experiments to study the effect of different pruning parameters on the accuracy and efficiency of the decision tree. The results shows that the accuracy changes by tuning the parameters like depth of the tree, maximum leaf etc. The experimental results shows that the parameters like depth of the tree, maximum leaf etc have direct impact on the accuracy of the decision tree. Increasing the depth and max_leaf parameters the accuracy of the classifier also improves. But beyond certain point it becomes constant which indicates that even if the tree grows in size after this point there would be no contribution in the accuracy improvement and still if the tree grows beyond certain level the accuracy may degrade. Therefore there is a limit of pruning an individual tree. However, a forest is an ensemble of a decision tree. Thus it can be concluded that in future the study can be focused in the direction of pruning a forest .Pruning a forest can finally result in an ensemble classifier that can efficiently classify data streams and can adapt to changes and drifts in the evolving stream.

REFERENCES

1.Domingos, P., & Hulten, G. (2000, August). Mining high-speed data streams. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 71-80). ACM.

2.Jankowski, D., Jackowski, K., & Cyganek, B. (2016). Learning Decision Trees from Data Streams with Concept Drift. *Procedia Computer Science*, *80*, 1682-1691.

3. Hulten, G. et al.: Mining time-changing data streams. Proc. seventh ACM SIGKDD Int. Conf. Knowl. Discov. data Min. KDD 01. 1, 97–106 (2001).

4. Gama, J. et al.: Accurate decision trees for mining high-speed data streams. Proc. ninth ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD '03. 523 (2003).

5. Gomes, H. M., Bifet, A., Read, J., Barddal, J. P., Enembreck, F., Pfharinger, B., ... & Abdessalem, T. (2017). Adaptive random forests for evolving data stream classification. *Machine Learning*, *106*(9-10), 1469-1495.

6. Beygelzimer, A., Kale, S., & Luo, H. (2015, June). Optimal and adaptive algorithms for online boosting. In *International Conference on Machine Learning* (pp. 2323-2331).

7. Chen, S. T., Lin, H. T., & Lu, C. J. (2012). An online boosting algorithm with theoretical justifications. *arXiv preprint arXiv:1206.6422*.

8. Bifet, A., de Francisci Morales, G., Read, J., Holmes, G., & Pfahringer, B. (2015, August). Efficient online evaluation of big data stream classifiers. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 59-68). ACM.

9. Barddal, J. P., Gomes, H. M., & Enembreck, F. (2015, April). SNCStream: A social network-based data stream clustering algorithm. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing* (pp. 935-940). ACM.

10. Brzezinski, D., & Stefanowski, J. (2014). Combining block-based and online methods in learning ensembles from concept drifting data streams. *Information Sciences*, *265*, 50-67.

11. Gomes, H. M., & Enembreck, F. (2014, March). SAE2: advances on the social adaptive ensemble classifier for data streams. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing* (pp. 798-804). ACM.

12. Parker, B. S., & Khan, L. (2015, January). Detecting and Tracking Concept Class Drift and Emergence in Non-Stationary Fast Data Streams. In *AAAI* (pp. 2908-2913).

13. Žliobaitė, I., Bifet, A., Read, J., Pfahringer, B., & Holmes, G. (2015). Evaluation methods and decision theory for classification of streaming data with temporal dependence. *Machine Learning*, *98*(3), 455-482.

14. Kapoor, P., Rani, R., & JMIT, R. (2015). Efficient Decision Tree Algorithm Using J48 and Reduced Error Pruning. *International Journal of Engineering Research and General Science*, *3*(3).

15. Bhargava, N., Sharma, G., Bhargava, R., & Mathuria, M. (2013). Decision tree analysis on j48 algorithm for data mining. *Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering*, *3*(6).

16. Wang, H., Fan, W., Yu, P. S., & Han, J. (2003, August). Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 226-235). AcM.

17. Patel, N., & Upadhyay, S. (2012). Study of various decision tree pruning methods with their empirical comparison in WEKA. *International journal of computer applications*, *60*(12).

18. Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., & Gavaldà, R. (2009, June). New ensemble methods for evolving data streams. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 139-148). ACM.

19. Parker, B. S., & Khan, L. (2015, January). Detecting and Tracking Concept Class Drift and Emergence in Non-Stationary Fast Data Streams. In *AAAI* (pp. 2908-2913).

20. Serrurier, M., & Prade, H. (2015, June). Entropy evaluation based on confidence intervals of frequency estimates: Application to the learning of decision trees. In *International Conference on Machine Learning* (pp. 1576-1584).

21. Oza Nikunj, C., & Russell Stuart, J. (2001). Online bagging and boosting. Jaakkola Tommi and Richardson Thomas, editors. In *Eighth International Workshop on Artificial Intelligence and Statistics* (pp. 105-112).

22. Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., & Gavaldà, R. (2009, June). New ensemble methods for evolving data streams. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 139-148). ACM

23. Bifet, A., & Gavalda, R. (2007, April). Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining* (pp. 443-448). Society for Industrial and Applied Mathematics.

24. Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., & Gavaldà, R. (2009, June). New ensemble methods for evolving data streams. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 139-148). ACM.

25. Oza, N. C. (2005). Online bagging and boosting.

26. Jiang, X., Wu, C. A., & Guo, H. (2017). Forest pruning based on branch importance. *Computational intelligence and neuroscience*, *2017*.

27. Jiang, N., & Gruenwald, L. (2006). Research issues in data stream association rule mining. *ACM Sigmod Record*, *35*(1), 14-19.

28. Kadwe, Y., & Suryawanshi, V. (2015). A review on concept drift. *IOSR J. Comput. Eng*, *17*, 20-26.