Student: Ha Le

ID: 1344084

25

Homework 3: Write an introduction for the paper: "Online Ensemble Learning: An Empirical Study."

Introduction

Nowadays, ensemble methods, which use multiple learning algorithms to obtain better predictive performance, have become an extremely popular research area. With the help of ensemble methods such as boosting, bagging ..., we may gain better results than other machine learning algorithms. Ensemble methods can be classified into two groups: offline learning methods and online learning methods. In the past decades, offline learning methods have proved it superior performance in many machine learning problems. Freund et al [3] presented a variant of the original boosting algorithm [5], called boosting by filtering, for improving the accuracy of algorithms for learning binary concepts. The improvement is archived by combining a large number of hypotheses, each of which is generated by training the given learning algorithm on a different set of examples. Breiman et al [1] introduced a bagging predictor technique, and proved it better performance on a linear regression example. Freund and Schapire et al [4] introduced a new boosting algorithm called ada-boost which reduces the error of learning algorithm that consistently generates classifiers whose performance is a little better than random guessing. He performed an experiment on OCR problem and showed the improvement in classifying performance. In contradiction to offline learning, online learning has not been received much attention due to its constraint on storage as well as processing time.

In this paper, we proposes two online learning algorithms that evolving from offline learning algorithms, which focusing on time and storage constraints. First, we introduce a general multiple-update algorithm based on the boosting by filtering algorithm [3]. In this general algorithm, instead of storing all the previous instances and learning new hypotheses, we only update the learnt hypotheses ~~based~~ on the new coming instance. P To reduce updating time, we can update learnt hypotheses in parallel. Second, an algorithm based on bagging technique is introduced. Our online bagging algorithm is similar to the general online boosting algorithm. However, we ignore weight updating in some instance to give each hypothesis a different training sequence. Our second online learning algorithm is based on the offline

What was changed and why was it changed.

algorithm Arc-x4 [2]. In this method, we used the same weight updating function as in Arc-x4 to emphasize selection of instances with many wrong prediction from previous hypotheses.

To evaluate our proposed online learning algorithm, we use our variant of ID4 online decision tree algorithm as the base learner and test on both the branch prediction domain and online variants of three familiar machine learning benchmarks. The experimental results show that our online boosting method based on Arc-x4 weight updating scheme outperforms our online bagging method as well as single base learners. The results also show that online learning methods with ensembles of small trees often have better performance than a single large tree. This result indicates that our online ensemble learning methods are very effective in tight space constraint problems, such as the branch prediction problem.

The remainder of this paper is organized as follows. In Section 2, we briefly discuss the motivating problem of branch prediction domain. In Section 3, we describe online ensemble learning and present our proposed algorithms. Section 4 discusses our online decision tree base learner. Section 5 and Section 6 show the empirical results for our proposed methods. Finally, Section 7 is our conclusions and future work.

References

1. Breiman, Leo. "Bagging predictors." Machine learning 24.2 (1996): 123-140.

2. Breiman, L., Variance Bias, and Arcing Classifiers. "Technical Report 460." Statistics Department, University of California, CA (1996).

3. Freund, Yoav. "Boosting a weak learning algorithm by majority." Information and computation 121.2 (1995): 256-285.

4. Freund, Yoav, and Robert E. Schapire. "Experiments with a new boosting algorithm." ICML. Vol. 96. 1996.

5. Schapire, Robert E. "The strength of weak learnability." Machine learning 5.2 (1990): 197-227.

Overall well written!
Misses some specific contribution of the paper, e.g.
a. (1) enhancement of ID4
b. parallel training of ~~hy~~ hypotheses

23

Scientific context could be established more clearly. I expected a more detailed description what the contribution of the paper is to solve these problem

Arthur Dunbar

Online Ensemble Learning: An Empirical Study

Introduction

Ensemble methods have been shown to greatly improve the accuracy of offline learners, but at the moment, they are rarely ported over to their online cousins. To address this imbalance and motivate further study, we introduce two new online ensemble learning algorithms: online boosting and online Arc-x4. Both methods use decision tree base ~~learners and work on binary features~~. ~~Our primary~~ motivation behind developing online ensemble learners is to better predict the outcomes of conditional branches.

Online learning algorithms are very useful for branch prediction because the predictions made need to change based upon what the computer is currently doing and the footprint of the algorithm needs to be light because the predictions need to be done quickly. Normal tabled approaches grow exponentially with the number of features considered. To fix this problem, our ensembles use depth bounded decision trees which merely grow linearly with the number of features considered. Good

In a naive sense, it is easy to construct an online algorithm from any offline algorithm by storing what has been seen so far. This causes the learner to have a large foot print, and in online settings, resources are usually constrained to an extent that makes this approach infeasible.

We also wish to avoid the single-update/sequential approach. Offline learners allow a single training instance to contribute to many ensembles, and we seek this attribute for an online learner. This is very important when considering concept drift, which, by its nature, a branch predictor must take in to account. Additionally, sequential methods require a difficult to design method that decides when to freeze a member of the ensemble and to start on another one.

Not much of a paragraph

Computers now typically have multiple cores allowing many processes to run simultaneously. To take advantage of this, we design our algorithms to be efficient when run in parallel.

mixes many things

We show empirically that our online Arc-x4 significantly outperforms our online boosting algorithm and individual online decision trees provided Arc-x4 has sufficient time to 'warm up.' The warm up is required because online ensemble learners suffer from a poor early performance but achieve better performance with more examples. This can be mitigated by only using approximately 25 trees. Unlike online Arc-x4, online bagging gives poor results (possibly because of 0/1 instance weighting).

Well depth of what

When testing the performance of Arc-x4, we varied both its depth for each tree in the ensemble and the number of trees included. We found that both increased the accuracy of Arc-x4 and we found that increasing the depth allowed the algorithm to better exploit the ensemble size. We also discovered that increasing depth gives benefits that increasing the number of trees can never give and vice versa. Despite this, increased depth is much more difficult to achieve because of the shear number of nodes it adds, while increasing the number of trees in the ensemble is relatively easy. For this reason, ensemble size is generally a better use of nodes than tree depth.

(After this the introduction wasn't blacked out for a paragraph, so I used it as the last paragraph) This paper is organized as follows. In section 2 we discuss the motivating problem of branch prediction. In section 3 we introduce online ensemble learning and our two algorithms. In section 4 we describe our online decision-tree base learner. In Sections 5 and 6, we give our empirical results for boosting and bagging ensembles.

First 2 paragraphs very good! The paper's contributions are presented at a somewhat abstract

Homework 3: Online Ensemble Learning: An Empirical Study

Name: Kunal Parmar

UHID: 1329834

Introduction

Ensemble learning is a very popular technique in machine learning where multiple weak learners are strategically combined to produce a strong learner. Ensemble methods have gained popularity because many researchers have demonstrated their superior prediction performance relative to single models on a variety of problems. However, these ensemble learning methods have been applied only in offline settings where the entire set of training examples is repeatedly processed as a whole. Using these ensemble methods in an online fashion would be useful when data is being generated continuously. In this paper, we apply ensemble approaches in an online learning setting. Online machine learning is used when training data becomes available in a sequential order and the algorithm updates some hypothesis after each example is presented to it. Cannot read data again!

~~cannot read exactly~~

~~ago~~

The online ensemble learning approach is applied to a conditional branch outcome prediction problem in computer architecture. To increase the performance of prefetching instructions in a microprocessor, conditional branch instructions need to be predicted correctly well in advance. Based on the history of the outcomes of a conditional branch in previous encounters, the processor predicts the value of the branch for the current state. Branch prediction is an interesting problem for online ensemble learning due to the time and space constraints imposed in the problem and the highly-parallel hardware architecture implementation. The goal of the paper is not to improve or beat the current state-of-the-art branch prediction architectures but rather explore the utility of online ensemble learning methods.

proposed

The paper ~~covers~~ online versions of two offline ensemble learning algorithms; the bagging algorithm and the Arc-x4 algorithm (Breiman 1996b), which is a boosting algorithm. In general, an online ensemble learning algorithm outputs an updated ensemble when provided with inputs of an ensemble, a training instance and an online learning algorithm. The online bagging algorithm applies a base learner to different sets of training ~~instances in~~ parallel. The online Arc-x4 algorithm is similar to its offline version except the ensemble vote update function used which is similar to the online bagging algorithm.

Also propose improvements of ID4)

The algorithms are trained using an online decision tree algorithm used as a base learner. The paper uses a variant of the ID4 algorithm (Schlimmer and Fisher, 1986) which allow an existing tree to be updated using only new individual data instances, without having to re-process past instances. The algorithms are applied to generate ensembles of decision trees used on problems from machine learning benchmarks and the domain of branch prediction. The online Arc-x4 algorithm significantly improves the prediction accuracy over single decision trees. Also, boosting produces ensembles of small trees that often outperform large single trees with the same number of nodes.

not very specific with respect to contribution

The rest of the paper is organized as below. In Section 2, we discuss the motivating problem of branch prediction. In Section 3, we introduce online ensemble learning and our two algorithms In Section 4, we describe our online decision tree base learner. In Sections 5 and 6, we give our empirical results for boosting and bagging ensembles.

References:

Breiman, L. (1996b). *Arcing Classifiers* (Technical Report). Dept of Statistics, Univ. of California, Berkeley, CA.

Schlimmer, J.C., & Fisher D. (1986). A case study of incremental introduction. *Proceedings of the Fifth National Conference in Artificial Intelligence* (pp. 496-501). San Francisco: Morgan Kaufmann.

Somewhat abstract, Not very specific with respect to contribution!

Does not introduce the scientific and applied context of the work clearly.

20

Assigned Paper: Online Ensemble Learning: An Empirical Study (2003)

Student: Pham Dinh Nguyen

26

Introduction

In computer architecture and CPU design, branch prediction is critical for the performance of the execution pipeline. A good branch predictor implementation usually employs a combination of various methods, but currently, very few of them are based on Machine Learning algorithms. The first Machine-learning-based proposal is the Neural Branch Predictor (Towards a High Performance Neural Branch Predictor, 1999), which was then followed up by other works. In this study, we initiate the application of Ensemble learning in solving branch prediction problem. Moreover, until recently, Ensemble methods have not been widely applied in online scenarios. Thus the problem is twofold interesting: how Ensemble methods can fit into a resource-limited and real-time branch predictor, and how hardware-level implementation can take advantage of Ensemble learning.

Good motivation

Ensemble learning is the family of machine learning methods where a base learner (or hypothesis) is trained multiple times (on the same input data with some mechanism for variation). The results are combined into the ensemble hypothesis (by a majority vote, or by weights,...). Theoretical and empirical studies ~~showed~~ have shown that Ensemble methods can achieve better performance compared to other standalone learning methods. Another big advantage is that Ensemble learner is very resistant to overfitting ~~problem~~, which makes it attractive for online adaptive learning.

→ examples can only be read once

However, applying Ensemble in branch prediction is not straightforward. The implementation must reside on the CPU, thus there is not enough space to track the states of complex ensemble hypothesis. This is known as the resource constrain. The second limitation is the time constrain, the prediction must be given as soon as a branching instruction is encountered, which means only a few pipeline cycles (in order of nanoseconds).

brad)

We only consider two-label classification, equivalent to branch ~~taken and~~ branch not taken decisions. Though not the most general ensemble scenario, this setting is typical for prediction problem. Moreover, we use the same feature space used by state-of-the-art predictors, to allow a meaningful performance comparison.

Due to the resource and time constrain, we choose the following setting for the study: Two online ensemble methods which are the online Bagging and the online Arc-x4, one base learner which is the decision tree ID4. We briefly discuss the reasoning as follow.

Offline ensemble methods are sequential, where a new hypothesis is learned at each iteration. If T is the number of iteration, the ensemble will consist of T hypothesizes. As the sequential approach will not meet the time constrain, we propose online ensembles where each hypothesis can be learned independently and in parallel. The online algorithm is thus: maintaining a set of T hypothesizes, with T weights (or some combination rule, i.e. majority vote). The other idea is to employ a predefine weight function, such that when a new example arrives, we can generate different examples to train each hypothesis in parallel. Our propose approach, for both online Bagging and online Arc-x4, has the space complexity of O(T) and the time complexity of O(1).

of few members in the ensemble.

For the base learner, we only deal with two-class problem, thus a simple decision tree is good enough. Given the resource constrain, the tree should not be too deep. Moreover, as we aim at hardware implementation, the tree must have static allocation. Thus the well-known offline algorithm ID3 is not suitable for the task. We consider ID4 and ID5, which are the online decision tree algorithms, and propose a variation of ID4 with compromised complexity but significantly improved accuracy.

English

With these settings, we conduct evaluation using the SPECint95 benchmark suite. Our methods are not good enough compared to the state-of-the-art results, as here is just the first step in this new direction. Focus on understanding the empirical results, our study shows that online Arc-x4 outperforms online bagging methods, and that ensemble decision tree outperforms single large one. Our results provide some insight to the online ensemble problems in resource-constrained scenarios, which can be used as a first step to further investigation.

The paper is organized as follows…….

seems to be novel — What are the ideas here??

Good background and motivation! Match with paper contribution is only partial!

24

Nikolaos Sarafianos (1392654) 2nd year PhD student at CBL

Advanced Machine Learning Prof.: Dr. Eick

Homework3: Writing an Introduction 12.06.2015

to accomplish high accuracy

Ensemble methods are learning algorithms that construct a set of classifiers and then classify new data points by taking a (weighted) vote of their predictions [1]. The power of ensemble algorithms arises from the fact that in order for the classification task to be effective, the individual experts must exhibit some level of diversity among themselves. The diversity in the classifiers – typically achieved by using different training parameters for each classifier – allows individual classifiers to generate different decision boundaries. If proper diversity is achieved, a different error is made by each classifier, strategic combination of which can then reduce the total error [2].

The combined of diverse class

In this work we follow a different path and explore the utility of online ensemble learning approaches [3]. In online approaches training instances are made available one at a time and hypotheses are obtained and updated after each example. However such approaches suffer from resource and space constraints preventing ~~the storage of~~ the stream of the training instances. Motivated by the problem of ~~conditional-branch outcome prediction~~ in ~~computer architecture, we~~ paid particular attention to the aforementioned limitations and introduced a parallel time and space-efficient ensemble learning approach for online settings.

A branch predictor is a digital circuit that tries to guess which way a branch (e.g. an if-then-else structure) will go before this is known for sure. The purpose of the branch predictor is to improve the flow in the instruction pipeline. Thus, binary prediction is a binary feature space two-class concept learning problem in an online setting. Although it's not our primary goal to improve the state-of-the-art branch predictors, our work contributes in this field since our predictors grow linearly with the number of features considered and not exponentially exploiting at the same time the utility of online ensemble methods. We focused on a two-class problem with binary features and showed benefits similar to those shown previously for offline ensembles.

good!

Kind of early!

To address some problems introduced by single-update/sequential approaches we opted for a parallel-generation multiple update approach. Freund [4] described a version of boost-by-majority (BBM) boosting algorithm which improved the accuracy of algorithms for learning binary concepts. We follow a different path and propose an online boosting by filtering framework which can be viewed as a parallel-generation multiple-update variant of the algorithm that uses Arcx4-style instance weighting as proposed by Breiman [5]. Breiman introduced Arcx4 algorithm which is a boosting algorithm to investigate whether the success of AdaBoost roots in its technical details or in the resampling scheme it uses [7].

Moreover

people vague

Besides the aforementioned boosting approach we also discuss an online bagging-style ensemble approach and in both cases we present empirical results ~~using~~ online ID4-like decision tree learners. The proposed decision tree algorithm is a variant of the original ID4 introduced by Sclimmer and

going forth and back!

Our variant?!..

Fisher [6]. ID4 incrementally updates a decision tree by maintaining an estimate of the split criterion of each feature at each node and using these estimates to dynamically select the split feature as well as to prune the tree. Our variant avoids feature discarding which is a common phenomenon in the original ID4 and the empirical results we report for Arc-x4 achieve significantly higher accuracies compared to single trees. Our investigation showed that ensembles of small trees tend to outperform single large trees. ensembles (of) fewer large trees.

The rest of this paper is organized as follows. In Section 2, we offer a discussion the branch prediction which is the problem we motivated by. In Section 3, we introduce ensemble learning and the two algorithms that were employed for the purposes of this paper. Section 4 covers the online decision-tree base learner and finally in Section 5, we provide our empirical results for the two ensemble approaches (boosting and bagging).

References:

[1] Dietterich, Thomas G. "Ensemble methods in machine learning." *Multiple classifier systems*. Springer Berlin Heidelberg, 2000. 1-15.

[2] http://www.scholarpedia.org/article/Ensemble_learning

[3] Oza, Nikunj C., and Stuart Russell. "Online ensemble learning." *AAAI/IAAI*. 2000.

[4] Freund, Yoav. "Boosting a weak learning algorithm by majority." *Information and computation* 121.2 (1995): 256-285.

[5] Breiman, Leo. "Bagging predictors." *Machine learning* 24.2 (1996): 123-140.

[6] Schlimmer, Jeffrey C., and Douglas Fisher. "A case study of incremental concept induction." *AAAI*. 1986.

[7] Kuncheva, Ludmila I. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2004.

Creative in some places; not very clear/precise in other places.

Online Ensemble Learning: An Empirical Study

Lifeng Yan
1361158

Abstract

branch prediction ??

Applications such as predicting instructions that will be needed in microprocessor or other computer architecture issues require ~~online learning~~ techniques with time and space constraint. In other words, learning algorithms that can achieve predicting conditional branch outcomes with high hardware efficiency is needed. Some related work has been done, including an online ensemble learning framework called "boost by filtering". This framework has inspired our work, along with the boosting-style algorithm Arc-x4. A variant of ID4 decision tree algorithm is proposed in this paper, which served as a base learner in the ensemble process, which is conducted through two styles of online ensemble methods: boosting and bagging. Experiments are in empirical way since otherwise it will be too computationally expensive and time consuming, so the result may not beat the state of the art. However, th~~e main contribution of~~ this paper ~~is to~~ reach several important conclusions, one is that depth and ensemble size have great influence on the prediction accuracy, and the other is that ensembles of smaller trees outperform single deep trees with the same space requirement. And the experimental results also proved that the ID4 variant algorithm is important for both single-tree and tree-ensemble situations, and can satisfy the constraints set by online approaches.

Not very well writing! but otherwise okay!

??

Short! Not very clear!

Conclusion

?!

?!

A variant ID4 algorithm working in two kinds of ensemble learning approaches: boosting and bagging is proposed in this paper, and considering the hardware implementation requests, the algorithm is parallel and high efficiency just like the framework "boost by filtering" and offline algorithm Arc-x4 on which our method is based, and thus the constraints of space and time of ~~online branch~~ predictions are satisfied, and can be useful in practical applications. The experimental results demonstrate ~~have proven~~ that the proposed method has achieved significant improvement of performance ~~while using in~~ both single-tree and tree-ensemble situations. And on several familiar machine-learning benchmarks, the results also shown that ensembles of small trees outperformed single large trees using same number of nodes, which is an important conclusion that will particularly be useful in applications with prediction tasks with tight spatial constraint.

??

Very Short!

no read out to ole for

?!!

?? use same number of nodes!

Online Ensemble Learning: An Empirical Study

ABSTRACT

Branch Prediction places tight time and space constraints on a learning algorithm due to which predictions must be made in a few nanoseconds. The work in this paper is motivated by this problem of predicting conditional branch outcomes in microprocessors. To achieve processor speed up error rates of branch predictor must be reduced. Time and space efficient online ensemble approach is used to demonstrate ensemble performance gains in online learning settings like that of offline learners. Online Ensemble methods do not store training instances due to resource constraints. Our online versions of ensemble, bagging and boosting algorithms are triggered by boosting by filtering approach, offline bagging and offline Arc-x4 algorithms and they use the parallel-generation multiple update approach wherein it trains multiple ensemble members for each training instance. Online decision tree algorithm is used as base learners to train the ensembles. ID4 algorithm which updates a decision tree is considered along with warm-up, post-pruning and feature-switch suppression like novel extensions introduced to it. Empirical results show that this extensions has significantly improved the ID4 classification accuracy both for single trees and ensembles. We evaluate our boosting and bagging-style online ensemble algorithms against instances of branch prediction problem drawn from computer architecture benchmarks as well as online machine learning benchmarks. Results ~~state~~ shows that ensembles of online trees produced by Arc-x4 consistently performs better than online bagging method and it also improves the error rate of single online decision tree learners. We even show that ensembles achieve less reduced percent error using less space along with large ensemble of small trees than small ensembles of large tress that use the same number of total tree nodes or large single trees. Enhancement in predictive accuracy with respect to ensemble size for online boosting algorithms is indicated. The evaluation demonstrates similar online performance gains and shows that ensemble methods are useful in meeting tight resource constraints.

Keywords: branch prediction, online ensemble learning, boosting, bagging, decision trees.

Very general!
Could be
be more specific!

Online Ensemble Learning: An Empirical Study

CONCLUSION

designed

Online ensemble learning overcomes the drawbacks in conditional branch outcome predictions which are bounded by time/space constraints. We have ~~implemented~~ two online ensemble parallel-generation algorithms, 'Online Bagging' and 'Online Arc-x4'. These ensemble algorithms provide methods for invoking "base" learning algorithm multiple times thereby combining the resulting hypothesis into an ensemble hypothesis. Online Arc-x4 uses the same instance weight function used by offline Arc-x4 and same accuracy based voting weight function as online bagging to improve prediction accuracy over single trees. These approaches do not require ~~storage of training instances~~. In this paper we extended online decision-tree base algorithm ~~that is~~ ID4 to be used as base learner in our ensemble experiments which was useful in gaining significant performance gains. After evaluating the results on online variants of machine learning and instances of conditional branch outcome prediction it can be concluded that Arc-x4 outperforms the online bagging method in mostly all experiments and improves accuracy over single trees as larger the ensemble size more is the reduction in error and better is the accuracy. Finally, it was shown that large ensembles can improve testing error even when training error has gone to zero and ensemble size mostly depends on number of training instances. Thus it has been demonstrated using empirical experiments that large ensembles of small trees allow us to achieve performance gain over small ensembles ~~of large trees that use the same number of total nodes making them useful in domains with time and space constraints~~. Also, ensembles allows us to use the limited space more effectively by trading off depth for more trees. Henceforth the goal to explore the utility of the online ensemble learning has been achieved. Future work in this area will mostly exploit the linear growth in feature space dimension to explore the use of additional processor state to exceed the current state of art in branch prediction.

8

Enhancement of ID4

Tianxiao 9

Abstract

We proposed online ensemble methods, motivated by the problem of predicting conditional branch outcomes in microprocessors where time and space resources are limited. We demonstrated that familiar ensemble performance gains could be seen in online ensembles. Without loss of generality, binary online concept learning problems with binary features were considered here. The base learner is an online decision-tree extending ID4 specifically designed for our problem.

A variation of 'boosting by filtering' approach which generates ensemble members in parallel is evaluated here. We adapt offline bagging and offline Arc-x4 to online ensemble algorithms, and empirically evaluated our methods against both frequently used computer-architecture benchmarks and several machine-learning benchmarks. Our results show that, first, online Arc-x4 consistantly outperforms online bagging methods. Second, the ensembles of online trees produced by online Arc-x4 "boosting" significantly improve the error rate of single online decision-tree learners. Third, ensembles of smaller trees often outperform larger trees given the same number of tree nodes.

maybe mention what the framework is

Conclusion

Due to that little work has been done on online ensemble learning. We aim to provide a feasible, new online ensemble methods that will provide similar prediction advantage in offline ensemble methods. A problem of branch prediction was studied here. Due to its constraint on resource (time/space), we only consider online ensemble learners that do not store training instances. ~~"Boosting by filtering" approach was evaluated here. Due to its~~ ability to generate ensemble members in parallel, a boost in warming up speed is evident.

Based on the framework on 'boosting by filtering', we tested two new online ensemble methods against both the branch prediction problem and machine-learning benchmarks. We demonstrated that online Arc-x4 using base learner of extended ID4 decision tree algorithm consistently outperform the online bagging method. And the ensemble online trees produced by online Arc-x4 significant improve the error rate against single online decision-tree. Also, we showed that by increasing the tree number and decreasing the depth, the ensemble of smaller trees often generate better result than large trees. Generally, our work demonstrated that ensemble methods could be efficiently adapted in online settings to solve resource constraint problem particularly but not restricted on conditional branch outcome prediction problem.

??

???

not always clear!

number of trees while reducing tree depth

& what was the extension?

Nacer 9

Abstract

Ensemble methods such as boosting and bagging have been evaluated extensively in offline setting and its benefits have been established. However, very little work has been done in evaluating them in an online setting. In this paper, we consider an online setting that is resource constraint, and it is motivated by the problem of branch conditional prediction. We describe two online ensemble algorithms: one inspired from offline bagging and one inspired by offline Arc-x4. We use ID4 as the base learner in the ensemble experiment. ID4 has been extended to avoid space costs of storing instances. These extensions are the ability ~~to have advanced warm-up~~, post-pruning by subtree monitoring and feature-switch suppression by subtree monitoring. We evaluate both bagging and Arc-x4 on branch prediction. Arc-x4 improves the accuracy over the trees. We use four machine learning benchmarks to and show how the error decreases as ensemble size increases. We compared the bagging algorithm with Arc-x4 which showed weaker performance than the latter.

good

mention

Free down

Conclusion

In this paper, we evaluate ensemble techniques in an online setting. We presented two online ensemble algorithms: one inspired from bagging and the other inspired from Arc-x4. Both have been designed considering memory resource constraints and for tackling the branch conditional prediction. We also extended ID4 base learner to resource bounded online settings which is one of the motivation of this work. The evaluation of these algorithm show that Arc-x4 significantly outperforms bagging even though it didn't necessarily beat the state of the art in many cases. But the goal of this work isn't to beat the state of the art but rather to evaluate ensemble methods in an online setting. This ~~paper sets~~ the ground to using ensemble in ~~an online setting that can be~~ applied to many problems one of which is branch prediction that is ~~presented in this paper.~~

7

not much outreach to other related topics / future work

quite short

By YONGLI ZHANG, 1269577

9

Abstract:

Ensemble methods such as bagging and boosting have gained popularity due to their theoretical performance guarantees and strong experimental results. However, these algorithms are mainly used in offline settings. In this paper, we study online ensemble learning, which is motivated by the problem of predicting conditional branch outcomes in multiprocessor architecture. Chiefly, time and efficiency are considered in our design of online ensemble learning, and our result demonstrates that performance gains similar to those observed in offline settings can be obtained. Moreover, we propose novel "parallel-generation" online ensemble algorithms which are inspired by 'boosting by filtering' proposed by Freund, the offline ensemble method of bagging, as well as the offline boosting-style algorithm Arc-x4. Furthermore, using a novel ID4-variant as the base learner, we empirically evaluate our online ensemble methods against the branch prediction problems and online variants of several familiar machine-learning benchmarks. And our results validate our algorithms in the following aspects: Our extensions to ID4 improve performance in single trees and are critical to obtaining good performance in tree ensembles; Also, our boosting-style algorithm online Arc-x4 consistently outperforms our online bagging methods and significantly improve the error rate compared to single base learners in most of our experiments; Finally, ensembles of small trees often outperform large single trees that use the same total number of tree nodes—similarly, large ensembles of small trees often outperform smaller ensembles of larger trees that use the same number of nodes. These results show that our online ensemble method can be very applicable and effective in domains where tight space and time constraints exist, such as branch prediction.

good!

evaluate

Keywords: ensemble learning, bagging, boosting, online ensemble learning, branch prediction

Conclusion:

This paper has described two novel "parallel generation" online ensemble learning algorithms-generic multiple-update online ensemble learning algorithm and online Arc-x4 algorithm. Inspired by offline ensemble methods of bagging and boosting respectively, the proposed methods have an efficient parallel hardware implementation, which is extremely important for domains such as branch prediction where tight space and time constraints exist. Moreover, we use a novel variant of ID4 as base learner of our decision tree based ensemble methods, and our results show that our extension to ID4 significantly improves performance in both single trees and tree ensembles. Furthermore, in order to justify our methods, we apply them to branch prediction problems and online variants of several familiar machine-learning benchmarks. Our results indicate that our boosting-style algorithm online Arc-x4 consistently outperforms our online bagging methods. Online Arc-x4 is also shown to significantly improve the error rate compared to single base learners in most of our experiments. In addition, we show that ensembles of small trees often outperform large single trees that use the same total number of tree nodes—similarly, large ensembles of small trees often outperform smaller ensembles of larger trees that use the same number of nodes. Last but not least, though the online boosting-style algorithm we present has not been proven to be a boosting algorithm in the theoretical sense, we provide empirical result to justify their effectiveness. Also, as is demonstrated in this paper, online ensemble learning methods have great potentials in domains with tight space and time constraints such as branch prediction. Therefore, in future, we hope our research can motivate more researches about online ensemble learning. ✓

8

What extension?