

A Study Of Bagging And Boosting Approaches To Develop Meta-Classifier

G.T. Prasanna Kumari

Associate Professor, Dept of Computer Science and Engineering, Gokula Krishna College of Engg,
Sullurpet-524121, AP, INDIA, Email:tabi_prasanna@yahoo.com

Abstract--Classification is one of the data mining techniques that analyses a given data set and induces a model for each class based on their features present in the data. Bagging and boosting are heuristic approaches to develop classification models. These techniques generate a diverse ensemble of classifiers by manipulating the training data given to a base learning algorithm. They are very successful in improving the accuracy of some algorithms in artificial and real world datasets. We review the two popular ensemble algorithms such as Boosting, Bagging.

In this paper we study the classification task with more emphasis on boosting and bagging methods classification. In this paper, we present online versions of bagging and boosting that require only one pass through the training data. Boosting induces the ensemble of weak classifiers together to create one strong classifier. In boosting successive models give extra weights to the earlier predictors. While In bagging, each model is independently constructed using a bootstrap sample of the data set. In the end, overall prediction is made by majority voting.

Keyword-Bagging; boosting; ensemble learning

I.INTRODUCTION

Data Mining is an iterative and multi step process of knowledge discovery in databases with the intention of uncovering hidden patterns. The huge amount of data to process is more and more significant in the world. Modern data-mining problems involve streams of data that grow continuously over time that includes customer click streams, telephone records, large sets of web pages, multimedia data, sets of retail chain transactions, assessing credit risks, medical diagnosis, scientific data analysis, music information retrieval and market research reports. Classification algorithm is a robust data mining tool that uses exhaustive methods to generate models from a simple to highly complex data. The induced model is used to classify unseen data instances. It can be referred as supervised learning algorithms because it assigns class labels to data objects. There are many approaches to develop the classification model including decision trees, meta algorithms, neural networks, nearest neighbor

methods and rough set based methods . The Meta classifiers is the most commonly used classification algorithms, because of their ease of implementation and easier to understand compared to other classification algorithms. Meta Learning is used in the area of predictive data mining, to combine the predictions from multiple models. It is significantly useful when the types of models are very different in their nature.

Bagging and boosting are two of the most well-known ensemble learning methods due to their theoretical performance guarantees and strong experimental results. However, these algorithms have been used mainly in batch mode, i.e., they require the entire training set to be available at once and, in some cases, require random access to the data. Bagging is useful for weak and unstable classifiers with a non-decreasing learning curve and critical training sample sizes. Boosting is beneficial only for weak, simple classifiers, with a non-decreasing learning curve, constructed on large training sample sizes. In this paper, we present online versions of bagging and boosting that require only one pass through the training data. We also compare the online and batch algorithms in terms of accuracy and running time.

II.META-CLASSIFIER: BAGGING ALGORITHM

Bagging frequently improves the predictive performance of a model. An online version has recently been introduced, which attempts to gain the benefits of an online algorithm while approximating regular bagging. However, regular online bagging is an approximation to its batch counterpart and so is not lossless with respect to the bagging operation. By operating under the Bayesian paradigm, we introduce an online Bayesian version of bagging which is exactly equivalent to the batch Bayesian version, and thus when combined with a lossless learning algorithm gives a completely lossless online bagging algorithm. We also note that the Bayesian formulation resolves a theoretical problem with bagging, produces less variability in its estimates, and can improve predictive performance for smaller data sets.

Bagging is a machine learning method of combining multiple predictors. It is a model averaging approach. Bagging is a technique generating multiple training sets by sampling with replacement from the available training data. It is also known as bootstrap aggregating. Bootstrap aggregating improves classification and regression models in terms of stability and accuracy. It also reduces variance and helps to avoid overfitting. It can be applied to any type of classifiers. Bagging is a popular method in estimating bias, standard errors and constructing confidence intervals for parameters. To build a model,

1. split the data set into training set and test set.
2. Get a bootstrap sample from the training data and train a predictor using the sample.

Repeat the steps at random number of times. The models from the samples are combined by averaging the output for regression or voting for classification. Bagging automatically yields an estimate of the out of sample error, also referred to as the generalization error. Bagging works well for unstable learning algorithms like neural networks, decision trees and regression trees. But it works poor in stable classifiers like k-nearest neighbors. The lack of interpretation is the main disadvantage of bagging. The bagging method is used in the unsupervised context of cluster analysis.

III. META-CLASSIFIER: BOOSTING ALGORITHM

Adaptive Boosting technique has been proposed in order to increase the accuracy of the ensemble. The basic idea of boosting is to build a series of classifiers so that the later classifier will focus more on the misclassified tuples of the previous round. Based on this way, an ensemble of classifiers with high accuracy will be produced since classifiers in the ensemble complement each other. Boosting is a general method for improving the accuracy of any given learning algorithm.

Adaptive boosting is a popular and powerful meta ensemble algorithm. “Boosting” is an effective method for the improvement in the performance of any learning algorithm. It is also referred as “stage wise additive modeling”. The model is a more user friendly algorithm. The algorithm does not suffer from overfitting. It solves both the binary classification problems as well as multiclass problems in the machine learning community. AdaBoost also gives an extension to regression problems. Boosting algorithms are stronger than bagging on noise free data. The algorithm depends more on data set than type of classifier

algorithms. The algorithm puts many weak classifiers together to create one strong classifier. It is a sequential production of classifiers.

To construct a classifier:

1. A training set is taken as input
2. A set of weak or base learning algorithms are called repeatedly in a series of rounds to maintain a set of weights over the training set. Initially, all weights are set equally, but on each round, the weights of incorrectly classified examples are increased so that the weak learner is forced to focus on the hard examples in the training data.
3. This boosting can be applied by two frameworks, i) boosting by weighting ii) boosting by sampling. In boosting by weighting method, the base learning algorithms can accept a weighted training set directly. With such algorithms, the entire training set is given to the base learning algorithm. And in boosting by sampling examples are drawn with replacement from the training set with probability proportional to their weights.
4. The stopping iteration is determined by cross validation.

The algorithm does not require prior knowledge about the weak learner and so can be flexibly combined with any method for finding weak hypotheses. Finally, it comes with a set of theoretical guarantees given sufficient data and a weak learner that can reliably provide only moderately accurate weak hypotheses. The algorithm is used on learning problems having either of the following two properties. The first property is that the observed examples tend to have varying degrees of hardness. The boosting algorithm tends to generate distributions that concentrate on the harder examples, thus challenging the weak learning algorithm to perform well on these harder parts of the sample space. The second property is that the algorithm is sensitive to changes in the training examples so that significantly different hypotheses are generated for different training sets.

IV. ON-LINE ENSEMBLE LEARNING

Many traditional machine learning algorithms generate a single model (e.g., a decision tree or neural network). An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way (typically by weighted or unweighted voting) to classify new examples. There are variety of approaches to construct an ensemble i.e., Bagging and Boosting. The idea of ensemble learning is to employ multiple learners and combine their predictions. If we have a committee of M models with uncorrelated errors, simply by averaging

them the average error of a model can be reduced by a factor of M . Unfortunately, the key assumption that the errors due to the individual models are uncorrelated is unrealistic; in practice, the errors are typically highly correlated, so the reduction in overall error is generally small. However, by making use of Cauchy's inequality, it can be shown that the expected committee error will not exceed the expected error of the constituent models. The concept of a committee arises naturally in a Bayesian framework. For a Bayesian—someone who is willing to put a probability on a hypothesis—the task of ensemble learning is, in principle, straightforward: one should use Bayesian model averaging (BMA). This involves taking an average over all models, with each model's prediction weighted by its posterior probability. In BMA it is assumed that a single model generated the whole data set, and the probability distribution over our candidate models simply reflects our uncertainty as to which model that is. By contrast, when we combine multiple models, we are assuming that different data points within the data set can potentially be generated from different models. Ensemble learning methods instead generate multiple models. Given a new example, the ensemble passes it to each of its multiple base models, obtains their predictions, and then combines them in some appropriate manner (e.g., averaging or voting). Ensemble learning traditionally has required access to the entire dataset at once, i.e., it performs batch learning. However, this is clearly impractical for very large datasets that cannot be loaded into memory all at once. (Oza and Russell, 2001; Oza, 2001) apply ensemble learning to such large datasets. In particular, this work develops online bagging and boosting, i.e., they learn in an online manner. That is, whereas standard bagging and boosting require at least one scan of the dataset for every base model created, online bagging and online boosting require only one scan of the dataset regardless of the number of base models.

Additionally, as new data arrives, the ensembles can be updated without reviewing any past data. However, because of their limited access to the data, these online algorithms do not perform as well as their standard counterparts. Theoretical frameworks that can guide us in the development of new ensemble learning algorithms specifically for modern datasets have yet to be developed.

Traditional supervised learning algorithms generate a single model such as a Naive Bayes classifier and use it to classify examples. Ensemble learning algorithms combine the predictions of multiple base models, each of which is learned using

a traditional algorithm. Bagging and Boosting are well-known ensemble learning algorithms that have been shown to improve generalization performance compared to the individual base models. Theoretical analysis of boosting's performance supports these results. Online learning algorithms process each training example once "on arrival" without the need for storage and reprocessing, and maintain a current model that reflects all the training examples seen so far. Such algorithms run faster than typical batch algorithms in situations where data arrive continuously. They are also faster with large training sets for which the multiple passes through the training set required by most batch algorithms are prohibitively expensive. There have been other recent efforts to develop online ensemble learning algorithms. An online bagging algorithm was developed in which the user chooses the probability that each training example is chosen for inclusion in each base model's training set. After considerable tuning, it performed comparably to online bagging. An online boosting algorithm that attempts to duplicate Breiman's Arc-x4 algorithm. A lossless online bagging algorithm that draws training example weights for each base model according to a Gamma distribution. However, their algorithm is lossless relative to a batch bagging algorithm that uses a Dirichlet distribution to draw its weights rather than the original bagging algorithm.

A. On-Line Bagging

Online bagging is a good approximation to batch bagging to the extent that their base model learning algorithms produce similar models when trained with similar distributions of training examples. If the same original training set is supplied to the two bagging algorithms, then the distributions over the training sets supplied to the base models in batch and online bagging converge as the size of that original training set grows to infinity. We have also proven that the classifiers returned by bagging and online bagging converge to the same classifier given the same training set as the number of models and training examples tends to infinity under two conditions. The first is that the base model learning algorithms return classifiers that converge toward the same classifier as the number of training examples grows. The second is that, given a fixed training set T , the online and batch base model learning algorithms return the same classifier for any number of copies of T that are presented to the learning algorithm. For example, doubling the training set by repeating every example in T yields the same classifier as T would yield. For example, this condition is true of decision trees and Naïve Bayes classifier.

Given a training dataset T of size N , standard batch bagging creates M base models. Each model is trained by calling the batch learning algorithm L_b on a bootstrap sample of size N created by drawing random samples with replacement from the original training set.

Figure 1 gives the pseudocode for bagging.

```

Bagging( $T, L_b, M$ )
  For each  $m \in \{1, 2, \dots, M\}$ ,
     $T_m = \text{Sample\_With\_Replacement}(T, |T|)$ 
     $h_m = L_b(T_m)$ 
  Return  $\{h_1, h_2, \dots, h_M\}$ 
    
```

Figure 1. Bagging Algorithm

Each base model's training set contains K copies of each of the original training examples where

$$P(K = k) = \binom{N}{k} \left(\frac{1}{N}\right)^k \left(1 - \frac{1}{N}\right)^{N-k}$$

which is the Binomial distribution. As $N \rightarrow \infty$, the distribution of K tends to a Poisson(1) distribution: $P(K=k) \sim \exp(-1)/k!$. We can perform bagging online as follows: as each training example $d=(x,y)$ is presented to our algorithm, for each base model, choose the example $K \sim \text{Poisson}(1)$ times and update the base model accordingly using the online base model learning algorithm L_o (see Figure 2). New examples are classified the same way in online and batch bagging---by unweighted voting of the M base models.

```

OnlineBagging( $\mathbf{h}, L_o, d$ )
  For each base model  $h_m \in \mathbf{h}, m \in \{1, 2, \dots, M\}$ ,
    Set  $k$  according to Poisson(1).
    Do  $k$  times
       $h_m = L_o(h_m, d)$ .
    
```

Figure 2. Online Bagging Algorithm

Online bagging is a good approximation to batch bagging to the extent that their base model learning algorithms produce similar models when trained with similar distributions of training examples. If the same original training set is supplied to the two bagging algorithms, then the distributions over the training sets supplied to the base models in batch and online bagging converge as

the size of that original training set grows to infinity. The classifiers returned by bagging and online bagging converge to the same classifier given the same training set as the number of models and training examples tends to infinity under two conditions. The first is that the base model learning algorithms return classifiers that converge toward the same classifier as the number of training examples grows. The second is that, given a fixed training set T , the online and batch base model learning algorithms return the same classifier for any number of copies of T that are presented to the learning algorithm. For example, doubling the training set by repeating every example in T yields the same classifier as T would yield.

B. Online Boosting

Our online boosting algorithm is designed to be an online version of AdaBoost. M1 (the pseudocode is given in Figure 3). AdaBoost generates a sequence of based models h_1, h_2, \dots, h_M using weighted training sets (weighted by D_1, D_2, \dots, D_M) such that the training examples misclassified by model h_{m-1} are given half the total weight when generating model h_m and the correctly classified examples are given the remaining half of the weight.

```

AdaBoost( $\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}, L_b, M$ )
  Initialize  $D_1(n) = 1/N$  for all  $n \in \{1, 2, \dots, N\}$ .
  For all  $m \in \{1, 2, \dots, M\}$ ,
     $h_m = L_b(\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}, D_m)$ .
     $\epsilon_m = \sum_{n=1}^N D_m(n) I(h_m(x_n) \neq y_n)$ .
    If  $\epsilon_m \geq 1/2$  then set  $M = m - 1$  and abort loop.
    For all  $n \in \{1, 2, \dots, N\}$ ,
       $D_{m+1}(n) = D_m(n) \times \begin{cases} \frac{1}{2(1-\epsilon_m)} & \text{if } h_m(x_n) = y_n \\ \frac{1}{2\epsilon_m} & \text{otherwise} \end{cases}$ 
  Return  $h_{fin}(x) = \arg \max_{y \in Y} \sum_{m=1}^M \log \left( \frac{1-\epsilon_m}{\epsilon_m} \right) I(h_m(x_n) = y_n)$ 
    
```

Figure 3. AdaBoost Algorithm

```

Initial conditions: For all  $m \in \{1, 2, \dots, M\}$ ,
 $\lambda_m^{sc} = 0, \lambda_m^{sw} = 0.$ 
OnlineBoosting( $\mathbf{h}, L_o, (x, y)$ )
  Set  $\lambda = 1.$ 
  For each base model  $h_m \in \mathbf{h}, m \in \{1, 2, \dots, M\}$ ,
    Set  $k$  according to Poisson( $\lambda$ ).
    Do  $k$  times
       $h_m = L_o(h_m, (x, y)).$ 
    If  $y = h_m(x)$ 
       $\lambda_m^{sc} \leftarrow \lambda_m^{sc} + \lambda$ 
       $\epsilon_m \leftarrow \frac{\lambda_m^{sw}}{\lambda_m^{sc} + \lambda_m^{sw}}$ 
       $\lambda \leftarrow \lambda \left( \frac{1}{2(1 - \epsilon_m)} \right)$ 
    else
       $\lambda_m^{sw} \leftarrow \lambda_m^{sw} + \lambda$ 
       $\epsilon_m \leftarrow \frac{\lambda_m^{sw}}{\lambda_m^{sc} + \lambda_m^{sw}}$ 
       $\lambda \leftarrow \lambda \left( \frac{1}{2\epsilon_m} \right)$ 
  end
  To classify a new example with input  $x$ , return :

$$h_{\text{final}}(x) = \arg \max_{y \in Y} \sum_{m=1}^M \log \left( \frac{1 - \epsilon_m}{\epsilon_m} \right) I(h_m(x) = y).$$


```

Figure 4. Online Boosting Algorithm

Our online boosting algorithm (Figure 4) simulates sampling with replacement using the Poisson distribution just like online bagging does. The only difference is that when a base model misclassifies a training example, the Poisson distribution parameter λ associated with that example is increased when presented to the next base model; otherwise it is decreased. Just as in AdaBoost, our algorithm gives the examples misclassified by one stage half the total weight in the next stage; the correctly classified examples are given the remaining half of the weight. This is done by keeping track of the total weights of each base model's correctly classified and misclassified training examples (λ_m^{sc} and λ_m^{sw} respectively) and using these to update each base model's error ϵ_m . At this point, a training example's weight is updated the same way as in AdaBoost. One area of concern is that, in AdaBoost, an example's weight is adjusted based on the performance of a base model on the entire training set while in online boosting, the weight adjustment is based on the base model's performance.

Consider running AdaBoost and online boosting on a training set of size 10000. In AdaBoost, the first base model h_1 is trained on all 10000 examples before being tested on, say, the tenth training example. In online boosting, h_1 is trained on only the first ten examples before being tested on the tenth example. Clearly, at the moment when the tenth training example is being tested, we may expect the two h_1 's to be very different; therefore, h_2 in AdaBoost and h_2 in online boosting may be presented with different weights for the tenth training example. This may, in turn, lead to different weights for the tenth example when generating h_3 in each algorithm, and so on. Intuitively, we want online boosting to get a good mix of training examples so that the base models and their normalized errors in online boosting quickly converge to what they are in AdaBoost. The more rapidly this convergence occurs, the more similar the training examples' weight adjustments will be and the more similar their performances will be.

V. CONCLUSION

Online versions perform comparably to their original batch counterparts in terms of classification performance. However, our online algorithms yield the typical practical benefits of online learning algorithms when the amount of training data available is large. Ensemble learning algorithms have become extremely popular over the last several years because these algorithms, which generate multiple base models using traditional machine learning algorithms and combine them into an ensemble model, have often demonstrated significantly better performance than single models. Bagging and boosting are two of the most popular algorithms because of their good empirical results and theoretical support. However, most ensemble algorithms operate in batch mode, i.e., they repeatedly read and process the entire training set. Typically, they require at least one pass through the training set for every base model to be included in the ensemble. The base model learning algorithms themselves may require several passes through the training set to create each base model. In situations where data is being generated continuously, storing data for batch learning is impractical, which makes using these ensemble learning algorithms impossible. These algorithms are also impractical in situations where the training set is large enough that reading and processing it many times would be prohibitively expensive. Unlike the batch versions, our online versions require only one pass through the training examples in order regardless of the number of base models to be combined. We discuss how we derive

the online algorithms perform comparably to the batch versions in terms of classification performance. We also demonstrate that our online algorithms have the practical advantage of lower running time, especially for larger datasets. This makes our online algorithms practical for machine learning and data mining tasks where the amount of training data available is very large.

REFERENCES

- [1] Rich Caruana, Alexandru Niculescu-Mizil, "An Empirical Comparison of Supervised Learning Algorithms", Appearing in Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, 2006.
- [2] Nikunj C. Oza, "Online Ensemble Learning," Ph.D.thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 2001.
- [3] Teyssier, M., & Koller, D. (2005). "Ordering-based search: A simple and effective algorithm for learning bayesian networks", Proceedings of the Twenty-first Conference on Uncertainty in AI (UAI).
- [4] Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40, 139–158.

[5] Alan Fern and Robert Givan, "Online Ensemble Learning: An Empirical Study," The Seventeenth International Conference on Machine Learning, Stanford, CA, pp. 279-286, July 2000.

[6] Herbert K.H. Lee and Merlise A. Clyde, "Lossless Online Bayesian Bagging," *Journal of Machine Learning Research*, Vol. 5, pp. 143-151, 2004.

[7] Nikunj C. Oza and Stuart Russell, "Online Bagging and Boosting," in *Artificial Intelligence and Statistics 2001*, Key West, FL, USA, pp. 105-112. January 2001.

AUTHORS PROFILE



Mrs G.T.Prasanna Kumari is an Associate Professor in the Department of Computer Science and Engineering, Gokula Krishna College of Engineering, Sullurpet. She is pursuing Ph.D., in Computer Science in the area of Distributed Data Mining Systems. She is in teaching since 1999. She presented papers at National and International Conferences and published articles in National and International journals..