AdaBoost as Classifier Ensemble in Classification Problems

Jasmina Đ. Novakovic
Belgrade business school,
Higher education institution for applied studies,
Belgrade, Serbia,
jnovakovic@sbb.rs

Alempije Veljovic
Faculty of technical science Cacak,
University of Kragujevac,
Cacak, Serbia,
alempije@beotel.net

Abstract— **In this paper, we suggest AdaBoost as classifier ensemble that can incorporate different base classifiers into classifier ensembles models for classification problems. This paper investigates the impact of using different base classifiers on classification accuracy of AdaBoost classifier ensemble. Classifier ensembles with five base classifier has used on five medical data sets. These results evaluated and compared choosing different type of decision tree algorithms for base classifier.**

Keywords- AdaBoost; classification accuracy; classifier ensembles; decision tree

I. INTRODUCTION

Machine learning involves the development of programs that learn from previous data. It is a field of artificial intelligence that deals with the construction of adaptive computing systems that are able to improve their performance by using information from experience. Machine learning is the discipline that studies the generalization and construction and analysis of algorithms that can generalize.

The machine learning method can achieve good performance in many areas, such as speech recognition, hand-written text, driving a car, and so on. But as much as the applications of machine learning were diverse, there are tasks that are repetitive. Therefore, it is possible to talk about the types of learning tasks that often occur. One of the most common tasks of learning that occurs in practice is classification. Classification is an important recognition of object types, for example whether a particular tissue represents malignant tissue or not. In this research we explore classification problems in machine learning.

In machine learning, in many fields multiple classifier system is more accurate and robust than an excellent single classifier, because one single classification system cannot always provide high classification accuracy [1-9]. Classifier combination is an active field of research for the reason that a lot of theoretical and practical studies present the advantages of the combination paradigm over the individual classifier models. A great deal of study has gone into designing multiple classifier systems that are commonly called classifier ensembles.

The main aim of this paper was to experimentally verify the impact of different base classifier on classification accuracy of AdaBoost. We use classifier ensembles, instead of individual classifier, because in many fields, multiple classifier system is more accurate and robust than an excellent single classifier.

In this study, we suggest classifier ensembles that can incorporate different base classifier into AdaBoost classifier ensembles models. The goal of this research is also to present and compare different algorithmic approaches for constructing and evaluating systems that learn from experience to make the decisions and predictions and minimize the expected number or proportion of mistakes.

The paper is organized as follows. In the next section we briefly described general issues concerning AdaBoost. Section 3 gives a brief overview of description of data sets which are used in this experiment. Section 4 discusses the results and investigates the performance of the proposed technique. Finally, concluding remarks are discussed in section 5.

II. ADABOOST

Boosting is a family of methods for improving the performance of a "weak" classifier by using it within an ensemble structure, the most prominent member of which is AdaBoost. In Boosting methods, a set of weights is maintained across the objects in the data set, so that objects that have been difficult to classify acquire more weight, forcing subsequent classifiers to focus on them. These methods works by repeatedly running a learning algorithm on various distributions over the training data, and then combining the classifiers produced by the learner into the single composite classifier.

The Boosting algorithm takes as input a training set of m examples $S = \langle (x_1, y_1), \ldots, (x_m, y_m) \rangle$ where x_i is an instance drawn from some space X, and $y_i \in Y$ is the class label associated with x_i. In this research, is assumed that the set of possible labels Y is of finite cardinality k. The Boosting algorithm calls weak learning algorithm repeatedly in a series of rounds. On round t, the booster provides weak learning algorithm with a distribution D_t over the training set S. Weak learning algorithm computes a classifier or hypothesis $hypothesis\ h_t: X \to Y$, which should misclassify a non trivial fraction of the training examples, relative to D_t. The goal of weak learner is to find a hypothesis h_t that minimizes the training error $\epsilon_t = Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$. Training error is measured with respect to the distribution D_t that was provided

to the weak learner. This process continues for T rounds. At last, the booster combines the weak hypotheses $h_1, \ldots, h_T$ into a single final hypothesis h_{fin}. In the Boosting algorithm the manner in which D_t is computed on each round, and how h_{fin} is computed are unspecified and these questions solve different Boosting schemes in different ways.

One of implementation of AdaBoost is AdaBoost.M1 algorithm. AdaBoost.M1 algorithm uses the simple rule present in Fig. 1, where the initial distribution D_1 is uniform over S so $D_1(i)=1/m$ for all i. In this algorithm to update distribution, the weight of example i is multiplied by some number $\beta_t \in [0,1]$ if h_t classifies x_i correctly, and otherwise the weight is left unchanged, and also divide by the normalization constant Z_t. Thus, "hard" examples, which tend often to be misclassified, get higher weight, and "easy" examples that are correctly classified by many of the previous weak hypotheses get lower weight. Accordingly, AdaBoost.M1 focuses the most weight on the examples that seem to be hardest for weak learning algorithm.

Input: sequence of m examples $\langle (x_1, y_1), \ldots, (x_m, y_m) \rangle$ with labels $y_i \in Y = \{1, \ldots, k\}$
weak learning algorithm
integer T specifying number of iterations
Initialize $D_1(i)=1/m$ for all i.
Do for $t = 1,2, \ldots, T$
1. Call weak learning algorithm, providing it with the distribution D_t.
2. Get back *hypothesis* $h_t: X \to Y$.
3. Calculate the error of h_t: $\epsilon_t = \sum_{i:h_t(x_i) \neq y_i} D_t(i)$. If $\epsilon_t > 1/2$, then set $T = t - 1$ and abort loop.
4. Set $\beta_t = \epsilon_t/(1 - e_t)$.
5. Update distribution D_t: $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t & \text{if } h_t(x_i) = y_i \\ 1 & otherwise \end{cases}$
where Z_t is a normalization constant (chosen so that D_{t+1} will be a distribution).
Output the final hypothesis: $h_{fin}(x) = \arg\max \sum_{t:h_t(x)=y} log \frac{1}{\beta_t}$

Figure 1. AdaBoost.M1 algorithm [3]

The number β_t is a function of ϵ_t and the final hypothesis h_{fin} is a weighted vote of the weak hypotheses. The weight of hypothesis h_t is defined to be $ln1/\beta_t$ so that greater weight is given to hypotheses with lower error.

The success of AdaBoost algorithm has been explained, among others, with its diversity creating ability, which is an important property of a classifier ensemble [4]. This algorithm creates inaccurate classifiers by forcing them to concentrate on difficult objects and ignore the rest of the data, which led to large diversity that boosted the ensemble performance, often beyond that of Bagging. This leads us to the famous accuracy-diversity dilemma, because it seems that classifiers cannot be both very accurate and have very diverse outputs.

III. DESCRIPTION OF DATA SETS

Five real data sets in medical domains were used for tests, taken from the UCI repository of machine learning databases [10]. We used these data sets to compare results of classification with data dimensionality reduction by AdaBoost in medical diagnosis. In the following, we provide the details for the benchmark data sets we have used from UCI repository of machine learning databases.

Hepatitis (HE): The main aim of this data set is to predict whether hepatitis patients will die or not. In this data set, there are two classes: live (123 instances) and die (32 instances). Fig. 2 presents liver tissue and the pathological changes in it due to the presence of chronic hepatitis C.

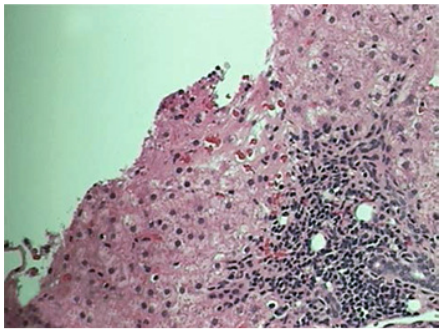

Figure 2. Liver tissue and the pathological changes in it due to the presence of chronic hepatitis C [http://www.cpmc.org/advanced/liver/patients/topics/HepatitisC-profile.html]

Figure 3. Alcohol-damaged liver [http://www.treatment4addiction.com/addiction/alcohol/liver-damage/]

Figure 4. Arizona Pima Indians [http://indiancountrytodaymedianetwork.com/article/mexico-vs.-arizona-pima-indians-3258]

Liver (LI): In this data set, the first five variables are all blood tests, which are thought to be sensitive to liver disorders that might arise from excessive alcohol consumption. Each row in this data set constitutes the record of a single male individual. Alcohol-damaged liver is presented n Fig. 3.

Pima Indians diabetes (PI): In this data set (Fig. 4) the diagnostic is whether the patient shows signs of diabetes according to World Health Organization criteria (i.e., if the 2 hour post-load plasma glucose was at least 200 mg/dl at any survey examination or if found during routine medical care).

Statlog Heart (SH): The task is to predict absence or presence of heart disease (Fig. 5). This data set contains 13 features (which have been extracted from a larger set of 74).

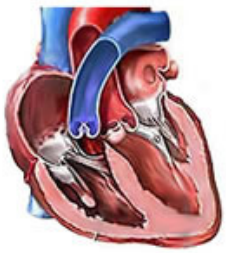

Figure 5. Heart [http://archive.ics.uci.edu/ml/datasets/Statlog+(Heart)]

Mammographic mass (MM): The task is to predict the severity (benign or malignant) of a mammographic mass lesion from BI-RADS features and the patient's age. Fig. 6 presents extracting contours based on the shadow of mammographic masses.

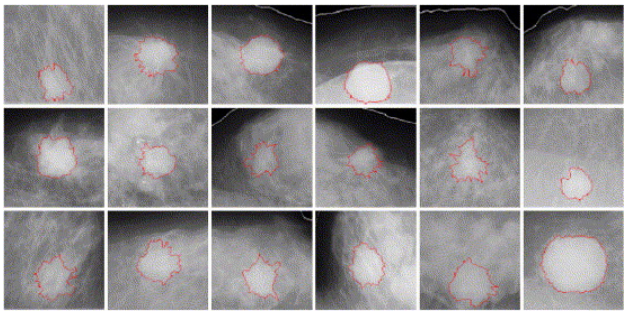

Figure 6. Extracting contours based on the shadow of mammographic masses [11]

IV. EXPERIMENT AND RESULTS

The experiment was performed using WEKA (Waikato Environment for Knowledge Analysis) tools for data preparation and research developed at the University of Waikato in New Zealand. When searching for the model that best approximates the target function, it is necessary to provide measures of quality models and learning. Different measures can be used depending on the problem, in our experimental studies; we used the classification accuracy as a measure of the quality of the model.

Our implementation is as follows. To get a more reliable evaluation of the learned knowledge, we used the cross-validation, where we have a full set of data that we had shared to *n* approximately equal subsets. In doing so, we have a subset of the training carried out and pulled the other *n-1* subsets, and after training, the quality of the learned knowledge assessed in a separate subset. Procedure described above are repeated for all other subsets extracted as a final quality score obtained by taking the average score for each of the subsets. In our experimental study we take the value of *n* is 10. Cross-validation was used in our experimental study, because the procedure gives stable quality evaluation, the advantage of this method is that each of the *n* steps of cross validation using a large amount of data in their training and all available instances at one time were used to test.

In this section, we will investigate the impact of AdaBoost as classifier ensembles on classification accuracy. Consequences of choosing different base classifier are monitored. In our case we used different type of decision tree algorithm, such as DecisionStump, J48, ADTree, LADTree and BFTree. Later on, comparisons of results of measuring the performance of classifiers are presented.

Decision trees have various advantages amongst other methods, such as:

- simple to understand and interpret,
- able to handle both numerical and categorical data,
- requires little data preparation,
- possible to validate a model using statistical tests,
- performs well with large datasets,
- robust, which means that performs well even if its assumptions are somewhat violated by the true model from which the data were generated.

The goal of decision tree is to create a model that predicts the value of a target variable based on several input variables. Each interior node corresponds to one of the input variables; there are edges to children for each of the possible values of that input variable. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf. This type of learning is one of the most successful techniques for supervised classification learning. We can assume that all of the features have finite discrete domains, and there is a single target feature called the classification. Each element of the domain of the classification is called a class. Classification tree is a tree in which each internal node is labeled with an input feature. The arcs coming from a node labeled with a feature are labeled with each of the possible values of the feature. In classification tree each leaf of the tree is labeled with a class or a probability distribution over the classes. Classification tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is stopped when the subset at a node has all the same value of the target variable, or when splitting no longer adds value to the predictions.

To achieve the goal of classifier ensemble to produce a model (based on the training data) which predicts the target values of the test data given only the test data features; the following procedure is used. We set classifier ensembles in following way:

- AdaBoost.M1 algorithm is used, which use the base classifier DecisionStump (AdaBoost_DS) and reweighting, the number of iterations is set on 10, and weight threshold for weight pruning is set on 100.
- AdaBoost.M1 algorithm is used, which use the base classifier J48 (AdaBoost_J48) and reweighting, the number of iterations is set on 10, and weight threshold for weight pruning is set on 100.
- AdaBoost.M1 algorithm is used, which use the base classifier ADTree (AdaBoost_ADT) and reweighting, the number of iterations is set on 10, and weight threshold for weight pruning is set on 100.
- AdaBoost.M1 algorithm is used, which use the base classifier LADTree (AdaBoost_LADT) and reweighting, the number of iterations is set on 10, and weight threshold for weight pruning is set on 100.
- AdaBoost.M1 algorithm is used, which use the base classifier BFTree (AdaBoost_BFT) and reweighting, the number of iterations is set on 10, and weight threshold for weight pruning is set on 100.

The classification accuracy is measured by applying AdaBoost with different base classifiers. AdaBoost as classifier ensembles were used for the good performance shown by the preliminary study, the high classification accuracy and high speed operation. After that, was analyzed the time taken to built model of each meta classifiers.

Results of classification accuracy, as a method for measuring the performance of AdaBoost for five data sets in medical domains, are presented in Table 1 and on Fig. 7. Results of the time taken to built model of each meta classifiers, are presented in Table 2 and on Fig. 8.

TABLE I. CLASSIFICATION ACCURACY OF ADABOOST

	Classification accuracy of AdaBoost				
Data set	AdaBoost_DS	AdaBoost_J48	AdaBoost_ADT	AdaBoost_LADT	AdaBoost_BFT
PI	74.35	72.40	72.79	73.70	70.96
SH	80.00	80.37	75.56	79.63	78.52
MM	82.62	79.81	81.37	81.89	76.27
LI	66.09	71.59	71.59	71.88	65.80
HE	82.58	84.52	77.42	82.58	82.58

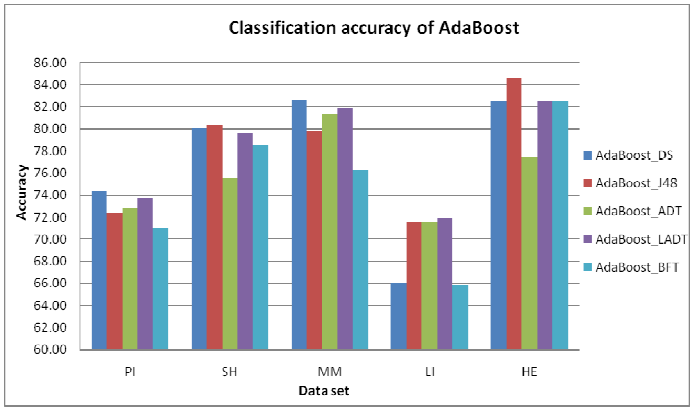

Figure 7. Impact of different base classifiers on classification accuracy

Selecting appropriate base classifier for a given data set, the reliability of classification for most of data sets and classifier ensembles is increased.

For PI and MM data sets, among all base classifier algorithms, using DS as base classifier, the highest classification accuracy are achieved. Using J48 as base classifier, the highest classification accuracy are achieved for SH and HE data sets. For LI data set, using LADT as base classifier, the highest classification accuracy is achieved.

TABLE II. ADABOOST AND THE TIME TAKEN TO BUILT MODEL OF EACH META CLASSIFIERS

	AdaBoost and time taken to build model (seconds)				
Data set	AdaBoost_DS	AdaBoost_J48	AdaBoost_ADT	AdaBoost_LADT	AdaBoost_BFT
PI	0.16	0.41	0.50	0.94	1.06
SH	0.02	0.08	0.20	0.34	0.22
MM	0.02	0.08	0.33	0.53	1.26
LI	0.00	0.03	0.16	0.23	0.23
HE	0.00	0.05	0.16	0.45	0.25

For all data sets, among all base classifier algorithms, using DS as base classifier, we achieved the minimum time required to build models.

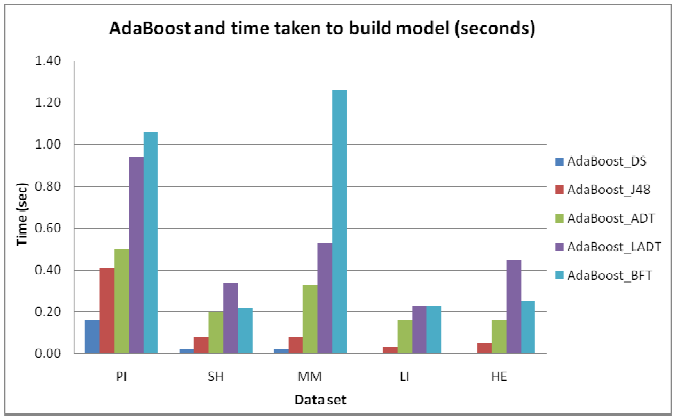

Figure 8. Impact of different base classifiers on time taken to build model

V. CONCLUSIONS

Five approaches for constructing AdaBoost classifier ensembles are presented, which have been found to be accurate and computationally feasible across various data domains.

The reliability of classification for most of data sets and classifier ensembles is increased when we select appropriate base classifier. For all data sets, using DS as base classifier, we achieved the minimum time required to build models.

REFERENCES

[1] L. Breiman, "Bagging predictors", Machine Learning, vol. 24, no. 2, pp. 123-140, 1996.

[2] L. Dong, Y. Yuan, Y. Cai, "Using bagging classifier to predict protein domain structural class", Journal of Biomolecular Structure & Dynamics, Volume 24, Issue Number 3, 2006.

[3] Y. Freund, R.E. Schapire, "Experiments with a new boosting algorithm", ICML, 1996.

[4] L. Kuncheva, “Diversity in multiple classifier systems” (editorial), Information Fusion, vol. 6, no. 1, 3-4, 2004.

[5] J.J. Rodriguez, L.I. Kuncheva, C.J. Alonso, “Rotation forest: a new classifier ensemble method”, In: IEEE Transactions on pattern analysis and machine intelligence, vol. 28, no. 10, October 2006.

[6] K.M. Ting, I.H. Witten “Stacking bagged and dagged models”, In: Fourteenth international Conference on Machine Learning, San Francisco, CA, pp. 367-375, 1997.

[7] Prem Melville, Raymond J. Mooney, “Constructing diverse classifier ensembles using artificial training examples”, Proceedings of the IJCAI-2003, pp. 505-510, Acapulco, Mexico, August 2003.

[8] G. I. Webb, “MultiBoosting: a technique for combining boosting and wagging”, Machine Learning, 40, pp. 159–39, Kluwer Academic Publishers, Boston, 2000.

[9] J. Friedman, T. Hastie, R. Tibshirani, “Additive logistic regression: a statistical view of boosting”, The Annals of Statistics 2000, vol. 28, No. 2, pp. 337–407.

[10] A. Frank, A. Asuncion, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml], Irvine, CA: University of California, School of Information and Computer Science, 2010.

[11] T. Nakagawa, T. Harab, H. Fujitab, T. Iwasec, T. Endod, K. Horitae, “Automated contour extraction of mammographic mass shadow using an improved active contour model”, Elsevier, International Congress Series, Volume 1268, June 2004, pp. 882–885.