A Comparative analysis on persuasive meta classification strategy for web spam detection

R. MahaLakshmi
Research Scholar,
Manonmanium University,
Tirunelveli, Tamil Nadu, India
mani_ram01@yahoo.co.in

***Abstract*— Classification is a data mining technique used to predict group membership for data instances[2]. In this paper, we are analyzing various basic classification techniques under the meta classifier. There are Several major kinds of classification methods including decision tree induction, Bayesian networks, k-nearest neighbor classifier, case-based reasoning, genetic algorithm and fuzzy logic techniques are available. The objective of this paper is to apply meta learning techniques and to provide a comprehensive review of different classification techniques in meta classification.The number of cases classified correctly provides us with an estimate of the accuracy of the model. our aim is to find highly accurate models that are easy to understand and achieve efficiency when dealing with large Datasets.**

Key Words: ***Data Mining, Classification, Performance***

I. INTRODUCTION

Data mining is the extraction of hidden predictive information from large databases [1]. It uses well established statistical and machine learning techniques to build models that predict some behavior of the data. Data mining tasks can be classified into two categories: Descriptive and predictive data mining. Descriptive data mining provides information to understand what is happening inside the data without a predetermined idea. Predictive data mining allows the user to submit records with unknown field values, and the system will guess the unknown values based on previous patterns discovered from the database.

Classification and prediction are predictive models, but clustering and association rules are descriptive models. Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends.

II. ENSEMBLE METHODS

Ensemble methods are used to improve the accuracy of the classifiers and predictors .Bagging and Boosting are two such techniques that use combination of models. Each combines a series of K learned models(classifiers/Predictors),M1,M2,M3…Mk, with the aim of creating an improved composite model M*.
Bagging, boosting and dagging are well known re-sampling ensemble methods that generate and combine a diversity of classifiers using the same learning algorithm for the base-classifiers. Boosting algorithms are considered stronger than bagging and dagging on noise-free data. However, there are strong empirical indications that bagging and dagging are much more robust than boosting in noisy settings. For this reason, in this work we performed a comparison with simple bagging, boosting and dagging ensembles, and DECORATE which are the base Meta Classifiers algorithms, on standard benchmark dataset.

A.Bagging

Given a set, D, of tuples, bagging works as follows. For iteration i (i = 1,2…k), a training set, Di, of d tuples is sampled with replacement from the original set of tuples, D. Note that the term bagging stands for bootstrap aggregation. Each training set is a bootstrap sample. Because sampling with replacement is used, some of the original tuples of D may not be included in Di, whereas others may occur more than once. A classifier model, Mi, is learned for each training set, Di. To classify an unknown tuple, X, each classifier, Mi, returns its class prediction, which counts as one vote. The bagging can be applied to the prediction of continuous values by taking the average value of each prediction for a give test tuple.

The bagged classifier often has significantly greater accuracy than a single classifier derived from D, the original training data. It will not be considerably worse and is more robust to the effects of noisy data. The increased accuracy occurs because the composite model reduces the variance of the individual classifiers. For prediction, it was theoretically proven that a bagged predictor will always have improved accuracy over a single predictor derived from

B) Boosting

In boosting, weights are assigned to each training tuple. A series of k classifiers is iteratively learned. After a classifier Mi is learned, the weights are updated to allow the subsequent

classifier, Mi+1, to pay more attention to the training tuples that were misclassified by Mi. The final boosted classifier, M*, combines the votes of each individual classifier, where the weight of each classifier's vote is a function of its accuracy. The boosting algorithm can be extended for the prediction of continuous values.

Adaboost is a popular boosting algorithm. Suppose We would like to boost the accuracy of some learning method. WE are given D, a data set of d class –labeled tuples,(x1,y1),(x2,y2),…..,(xd,yd), where yi is the class label of tuple xi. Initially, adaboost assigned each training tuple an weight of 1/d . Generating k classifiers for the ensemble requires K rounds through the rest of the algorithm .In round i, the tuple from D are sampled to form a triangle set ,Di of size d. sampling with replacement is used –the same tuple may be selected more ,Mi is derived from the training tuple are then adjusted according to how they were classified. If a tuple was incorrectly classified its weight is increase. If a tuple was correctly classified its weight is decrease .A tuple weight reflect how hard it is to classify-the higher the weight, the more often it has been misclassified, These weights will be used to generated of he next round. The basic idea is that when we build a classifier, we want it to focus more on the misclassified tuples of the previous round. Some classifiers may be better at classifying some hard tuples than others. In this way, we build a series of classifiers that complement each other.

To compute the error rate of model Mi, we sum the weights of each of the tuples in Di that Mi misclassified.

$$\text{i.e., } error(Mi) = \sum_{j}^{d} W_j * err(X_j)$$

where err(Xj) is the misclassification error of tuple Xj. If the tuple was misclassified, then err(Xj) is 1. Otherwise, it is 0. If the performance of classifier Mi is so poor that its error exceeds 0.5, then we abandon it. Instead, we try again by generating a new Di training set, from which we derive a new Mi.

The error rate of Mi affects how the weights of the training tuples are updated. If a tuple in round i was correctly classified, its weight is multiplied by error(mi) / (1 – error (Mi)). Once the weights of all of the correctly classified tuples are updated, the weights for all tuples are normalized so that their sum remains the same as it was before. To normalize a weight, we multiply it by the sum of the old weights, divided by the sum of the new weights. As a result, the weights of misclassified tuples are increased and the weights of correctly classified tuples are decreased, as described above.

"Once boosting is complete, the ensemble of classifiers used to predict the class label of a tuple, X ,Unlike bagging, where each classifier was assigned an equal vote, For each class, c, we sum the weights of each classifier that assigned class c to X. The class with the highest sum is the "winner" and is returned as the class prediction for tuple X.

C) DAGGING

This meta classifier creates a number of disjoint, stratified folds out of the data and feeds each chunk of data to a copy of the supplied base classifier. Predictions are made via majority vote, since all the generated base classifiers are put into the Vote meta classifier. Useful for base classifiers that are quadratic or worse in time behavior, regarding number of instances in the training data[7].

D) MultiBoosting

MultiBoosting is another classifier method with same category that can be considered as Wagging. Wagging is a variant of Bagging. Bagging uses resampling to get the datasets for training and producing weak hypothesis where as Wagging uses reweighting for each training example, pursuing the effect of bagging in a different way[7].

E) DECORATE

DECORATE(Diverse Ensemble Creation by Oppositional, Relabeling of Artificial Training Examples) is presented that uses a learner (one that provides high accuracy on the training data) to build a diverse committee. This is accomplished by adding different randomly constructed examples to the training set when building new committee members. These artificially constructed examples are given category labels that disagree with the current decision of the committee, thereby directly increasing diversity when a new classifier is trained on the augmented data and added to the committee[7].

III. PERFORMANCE EVALUATION

Data set Description

The dataset Spam email collected from UCI repository [8]. It has 4601 instances, out of which 2788 genuine and 1813 spam mails. There are 58 attributes in each instance , out of which 57 are continuous and 1 has nominal class label. Most of the attributes represent the frequency of a given word or character in the email that corresponds to the instance.

Attribute Information:[8]

- 48 attributes of type word_freq_WORD describing the frequency of word w, the percentage of words in the email.
- 6 attributes of type char_freq_CHAR describing the frequency of a character c, defined in the same way as word frequency.
- 3 attributes describing the longest length, total numbers of capital letters and average length.
- 1 nominal {0,1} class attribute of type spam describing whether the e-mail was considered spam (1) or not (0)

The performance of the classifiers depends on the characteristics of the data to be classified. Various pragmatic tests can be performed to compare the classifier like holdout,

random sub-sampling, k-fold cross validation and bootstrap method. In our study, we have selected k-fold cross validation for evaluating the classifiers.

In k-fold cross validation, the initial data are randomly partitioned into k mutually exclusive subset or folds $d_1,d_2,\ldots,d_k$, each approximately equal in size. The training and testing is performed k times. In the first iteration, subsets d_2, …, d_k collectively serve as the training set in order to obtain a first model, which is tested on d_1; the second iteration is trained in subsets d_1, $d_3,\ldots$, d_k and tested on d_2; and so no[2].

WEKA 3.6.5 tool is used to study the Performance of the chosen algorithms and the results are used to measure the Accuracy, Sensitivity, Specificity and Error rate from the confusion matrix 2x2 obtained.
The following are the formulae used to calculate the Accuracy, Sensitivity, Specificity and Error rate of the used ensemblers.

Accuracy =(TP+TN)/(TP+FP+TN+FN)
Sensitivity=TP/(TP+FN)
Specificity=TN/(TN+FP)
Error rate=(FP+FN)/(TP+FP+TN+FN) Where

TP is the number of True positives,
TN is the number of True negatives,
FP is the number of false positives,
FN is the number of false negatives.

IV. EXPERIMENTAL RESULTS

Table 1 shows the Accuracy, Sensitivity, Specificity and Error rate of meta classification models.

Figure1 shows the graphical representation of difference in Accuracy.

Figure2 shows the graphical representation of difference in Sensitivity.

Figure3 shows the graphical representation of difference in Specificity.

Figure4 shows the graphical representation of difference in Error rate.

Table 1: Comparison of Classification Models

Algorithms	Accuracy	Sensitivity	Specificity	Error rate
Bagging	94.34	96.05	91.73	5.65
Dagging	89.52	95.65	80.08	10.47 %
AdaBoost	90.06	92.47	86.37	9.93
MultiBoost	84.43	85.68	82.52	15.56
DECORATE	93.72	95.37	91.17	6.28

Figure 1: Comparison graph based on Accuracy

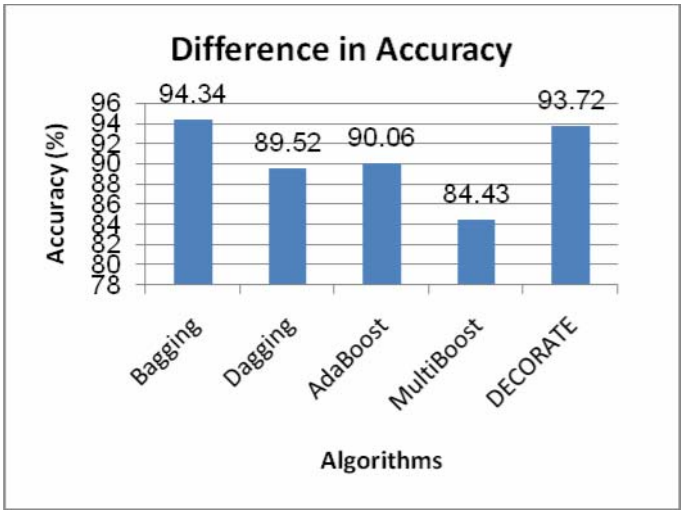


Figure 2: Comparison graph based on Sensitivity

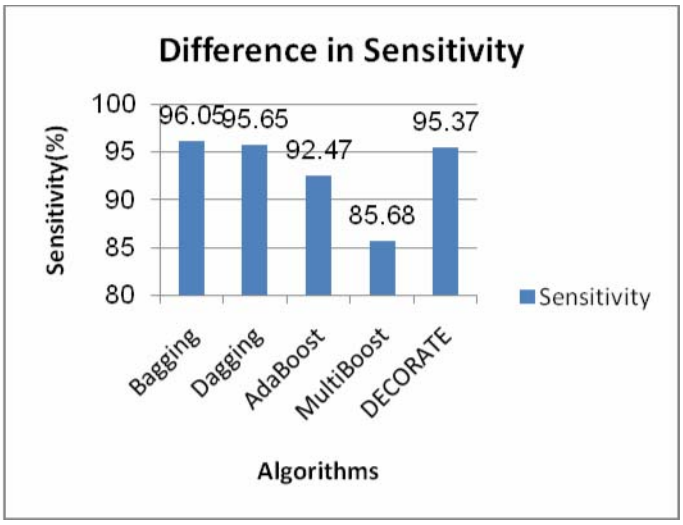


Figure 3: Comparison graph based on Specificity

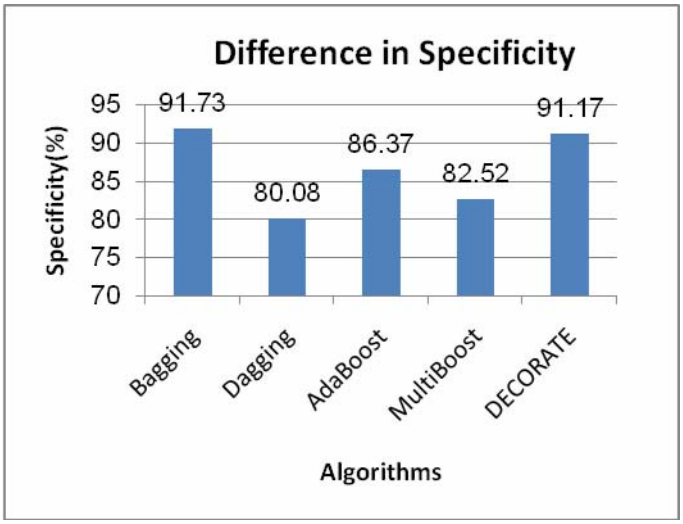
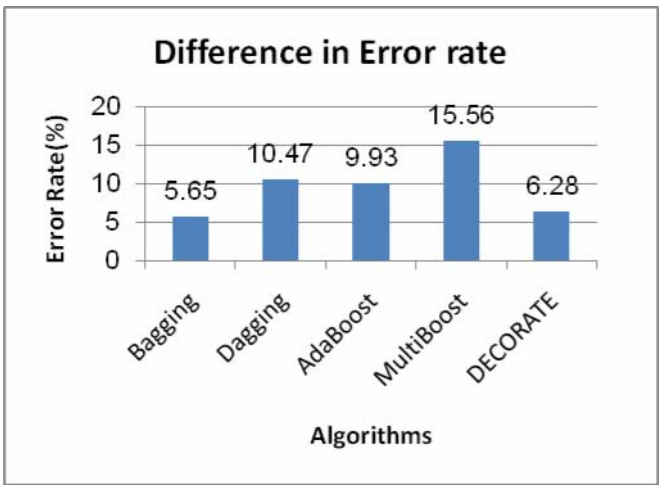


Figure 4: Comparison graph based on Error rate.



.

V. Conclusion and Scope for Further Enhancements

In this paper, the performance of various ensemble classification methods like Bagging, Dagging, Boosting , and Decorate are compared. The experiments were conducted on the "Web spam Dataset". Classification Accuracy, Sensitivity, Specificity and Error rate is validated by 10-fold cross validation method. Our Studies shows that Bagging turned out to be the best classifier out of the 5 ensemblers. This study was conducted to understand the functionality of various ensemble methods and their performance. In future we may conduct the same experiments with different datasets instead of single dataset, and combine few ensemblers with the single base classifier to study how could the ensemblers combined with the base classifiers boost the performance accuracy.

VI. References

1. W. Frawley and G. Piatetsky-Shapiro and C. Matheus, Knowledge Discovery in Databases: An Overview. AI Magazine, Fall 1992, pgs 213-228

2. Thair Nu PhyuProceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I IMECS 2009, March 18 - 20, 2009, Hong Kong

3. E. Alpaydin, Introduction to machine learning. London: The MIT Press, 2004.

4. R. O. Duda, P. E. Hart, and D. G. Stork, Pattern classification, 2 ed. New York: John Wiley & Sons, Inc., 2001.

5. Group-based Meta-classification Noor A. Samsudin, Andrew P. Bradley,School of Information Technology and Electrical Engineering The University of Queensland, Australia {azah,

6. S A Robust Meta-Classification Strategy for Cancer Diagnosis from Gene Expression Data Gabriela Alexe, Gyan Bhanot, *IBM* Computational Biology Center, IBM T.J.Watson Research.

7. Ms. Bhoomi Trivedi,Ms. Neha Kapadia,INDUS institute of Eng. & Tech,TCET, Kandivali(E),Ahmedabad Modified Stacked generalization withSequential Learning. TCET2012 on IJCA.

8. Suresh Subramanian,Research Scholar , Karpagam University
Coimbatore, Tamil Nadu, India Spam Email classification based on machine learning algorithms

9. J.R. Quinlan, "Induction of decision trees," In Jude W.Shavlik, Thomas G. Dietterich, (Eds.), Readings in Machine Learning. Morgan Kaufmann, 1990. Originally published in Machine Learning, vol. 1, 1986, pp 81–106.

10. Salvatore Ruggieri,"Efficient C4.5 Proceedings of IEEE transactions on knowledge and data Engineering",Vo1. 14,2,No.2, PP.438-444,20025

11. P. Domingos, M. Pazzani, On the optimality of the simple Bayesian classifier under Zero-one loss, Machine learning 29(2-3)(1997) 103-130.11

12. Vapnik, V.N., The Nature of Statistical Learning Theory, 1st ed., Springer-Verlag,New York, 1995.

13. Michael J. Sorich, John O. Miners,Ross A. McKinnon,David A. Winkler, Frank R. Burden, and Paul A. Smith, "Comparison of linear and nonlinear classification algorithms for the prediction of drug and chemical metabolism by human UDP-Glucuronosyltransferase Isoforms"

14. Thair Nu Phyu, "Survey of Classification Techniques in Data Mining MultiConference of Engineers and Computer Scientists" 2009 Vol I IMECS 2009, Hong Kong

AUTHOR'S PROFILE

Ms. R.Mahalakshmi is a Research Scholar in Manonmanium Sundaranar University. Her Research interests include Data and web Mining, search engine Optimization.