

A STUDY OF DECISION TREE ENSEMBLES AND FEATURE SELECTION FOR STEEL PLATES FAULTS DETECTION

Sami M. Halawani

Faculty of Computing and Information Technology,
King Abdulaziz University, Jeddah, Saudi Arabia.

Abstract— The automation of fault detection in material science is getting popular because of less cost and time. Steel plates fault detection is an important material science problem. Data mining techniques deal with data analysis of large data. Decision trees are very popular classifiers because of their simple structures and accuracy. A classifier ensemble is a set of classifiers whose individual decisions are combined in to classify new examples. Classifiers ensembles generally perform better than single classifier. In this paper, we show the application of decision tree ensembles for steel plates faults prediction. The results suggest that Random Subspace and AdaBoost.M1 are the best ensemble methods for steel plates faults prediction with prediction accuracy more than 80%. We also demonstrate that if insignificant features are removed from the datasets, the performance of the decision tree ensembles improve for steel plates faults prediction. The results suggest the future development of steel plate faults analysis tools by using decision tree ensembles.

Index Terms— Material informatics, Steel plates faults, Data mining, Decision trees, Ensembles.

I. INTRODUCTION

Materials informatics [1] is a field of study that applies the principles of data mining techniques to material science. It is very important to predict the material behavior correctly. As we cannot do the very large number of experiments without high cost and time, the prediction of material properties by computer methods is gaining ground. A fault is defined as an abnormal behavior. Defects or Faults detection is an important problem in material science [2]. The timely detection of faults may save lot of time and money. The steel industry is one of the areas which have fault detection problem. The task is to detect the type of defects, steel plates have. Some of the defects are Pastry, Z-Scratch, K-Scatch, Stains, Dirtiness, and Bumps etc. One of the tradition methods is to have manual inspection of each steel plate to find out the defects. This method is time consuming and need lot of efforts. The automation of fault detection technique is emerging as a powerful technique for fault detection [3]. This process relies on data mining techniques [4] to predict the fault detection. These techniques use past data (the data consist of features and the output that is

to be predicted by using features) to construct models (which are also called classifiers) and these models are used to predict the faults.

Classifiers ensembles are popular data mining techniques [5]. Classifier ensembles are combination of base models; the decision of each base model is combined to get the final decision. A decision tree [6] is very popular classifier which has been successfully used for various applications. Decision tree ensembles have been very accurate classifiers and have shown excellent performance in different applications [7].

It has also been shown that not all features that are used for prediction are useful [8]. Removing some of the insignificant features may improve the performance of the classifiers. Hence, it is important to analyze the data to remove insignificant features.

Various data mining techniques [9, 10] have been used to predict the steel plate faults, however, there is no detailed study to show the performance of different kinds of decision tree ensembles for predicting steel plate faults. The advantage of the decision tree ensembles is that they give accurate predictions [11]. Hence, we expect that decision tree ensembles will perform well for steel plate fault predictions. As there has been no study to see the effect of removing insignificant features on prediction accuracy for steel plate faults, it will be useful to do this study for better prediction results. There are two objectives of the paper;

1- There are many decision tree ensemble methods. In this paper we will find out which method is the best for steel plates prediction problem.

2- To investigate the effect of removing insignificant features on the prediction accuracy for decision tree ensembles for steel plates faults.

II. RELATED WORKS

In this section, we will discuss various data mining techniques that were used in this paper.

Decision trees

Decision trees are very popular tools for classification as they produce rules that can be easily analyzed by human

beings. A decision tree can be used to classify an example by starting at the root of the tree and moving through it until a leaf node, which provides the rules for classification of the example. There are various methods to produce decision trees, however C4.5 [6] and CART trees [12] are the most popular decision tree methods.

Classifier Ensembles

Ensembles [5] are a combination of multiple base models; the final classification depends on the combined outputs of individual models. Classifier ensembles have shown to produce better results than single models, provided the classifiers are accurate and diverse.

Several methods have been proposed to build decision tree ensembles. In these methods, randomization is introduced to build diverse decision trees. Bagging [13] and Boosting [14] introduce randomization by manipulating the training data supplied to each classifier. Ho [15] proposes Random Subspaces that selects random subsets of input features for training an ensemble. Breiman [16] combines Random Subspaces technique with Bagging to create Random Forests. To build a tree, it uses a bootstrap replica of the training sample, then during the tree growing phase, at each node the optimal split is selected from a random subset of size K of candidate features, then during the tree growing phase, at each node the optimal split is selected from a random subset of size K of candidate features. We will discuss these techniques in detail.

Random Subspaces

Ho [15] presents Random Subspaces (RS) ensembles. In this method, diverse datasets are created by selecting random subsets of a feature space. Each decision tree in an ensemble learns on one dataset from the pool of different datasets. Results of these trees are combined to get the final result. This simple method is quite competitive to other ensemble methods. Experiments suggest that RS is good when there is certain redundancy in features. For datasets where there is no redundancy, redundancy needs to be introduced artificially by concatenating new features that are linear combinations of original features to the original features and treating this as the data.

Bagging

Bagging [13] generates different bootstrap training datasets from the original training dataset and uses each of them to train one of the classifiers in the ensemble. For example, to create a training set of N data points, it selects one point from the training dataset, N times without replacement. Each point has equal probability of selection. In one training dataset, some of the points get selected more than once, whereas some of them are not selected at all. Different training datasets are created by this process. When different classifiers of the ensemble are trained on different training datasets, diverse classifiers are created. Bagging does more to reduce the variance part of the error of the base classifier than the bias part of the error.

Adaboost.M1

Boosting [14] generates a sequence of classifiers with different weight distribution over the training set. In each iteration, the learning algorithm is invoked to minimize the weighted error, and it returns a hypothesis. The weighted error of this hypothesis is computed and applied to update the weight on the training examples. The final classifier is constructed by a weighted vote of the individual classifiers. Each classifier is weighted according to its accuracy on the weighted training set that it has trained on. The key idea, behind Boosting is to concentrate on data points that are hard to classify by increasing their weights so that the probability of their selection in the next round is increased. In subsequent iteration, therefore, Boosting tries to solve more difficult learning problems. Boosting reduces both bias and variance parts of the error. As it concentrates on hard to classify data points, this leads to the decrease in the bias. At the same time classifiers are trained on different training data sets so it helps in reducing the variance. Boosting has difficulty in learning when the dataset is noisy.

Random Forests

Random Forests [16] are very popular decision tree ensembles. It combines bagging with random subspace. For each decision tree, a dataset is created by bagging procedure. During the tree growing phase, at each node, k attributes are selected randomly and the node is split by the best attribute from these k attributes. Breiman [16] shows that Random Forests are quite competitive to Adaboost. However, Random Forests can handle mislabeled data points better than Adaboost can. Due to its robustness of the Random Forests, they are widely used.

Feature Selection

Not all features are useful for prediction [17, 18]. It is important to find out insignificant features that may cause error in prediction. There are many methods for feature selection, however, Relief [19] is a very popular feature selection method. Relief is based on feature estimation. Relief assigns a value of relevance to each feature. All features with higher values than the user given threshold value are selected.

III. RESULTS AND DISCUSSION

All the experiments were carried out by using WEKA software [20]. We did the experiments with Random Subspace, Bagging, AdaBoost.M1 and Random Forests modules. For the Random subspace, Bagging and AdaBoost.M1 modules, we carried out experiments with J48 tree (the implementation of C4.5 tree). The size of the ensembles was set to 50. All the other default parameters were used in the experiments. We also carried out experiment with single J48 tree. Relief module was used for selecting most important features. 10-cross fold strategy was used in the experiments. The experiments were done on the dataset taken from UCI repository [21]. The

dataset has 27 independent features and 7 classes; Pastry, Z-Scratch, K-Scratch, Stains, Dirtiness, and Bumps and Other-Deffects. The information about the features is provided in Table 1. We compared the prediction error of all the methods. Low error suggests better performance.

Classification Method	Classification Error in %
Random Subspace	20.60
Bagging	19.19
AdaBoost.M1	18.08
Random Forests	20.04
Single Tree	23.13

Table 1. – The information about the features of steel plates

Feature Number	Feature Name	Feature Number	Feature Name
1	X_Minimum	15	Edges_Index
2	X_Maximum	16	Empty_Index
3	Y_Minimum	17	Square_Index
4	Y_Maximum	18	Outside_X_Index
5	Pixels_Areas	19	Edges_X_Index
6	X_Perimeter	20	Edges_Y_Index
7	Y_Perimeter	21	Outside_Global_Index
8	Sum_of_Luminosity	22	LogOfAreas
9	Minimum_of_Luminosity	23	Log_X_Index
10	Maximum_of_Luminosity	24	Log_Y_Index
11	Length_of_Conveyer	25	Orientation_Index
12	TypeOfSteel_A300	26	Luminosity_Index
13	TypeOfSteel_A400	27	SigmoidOfAreas
14	Steel_Plate_Thickness		

Table 4- Classification error of various methods with 15 most important features, calculated from Relief method, used for the training and prediction.

Classification Method	Classification Error in %
Random Subspace	21.32
Bagging	21.70
AdaBoost.M1	21.38
Random Forests	22.35
Single Tree	24.67

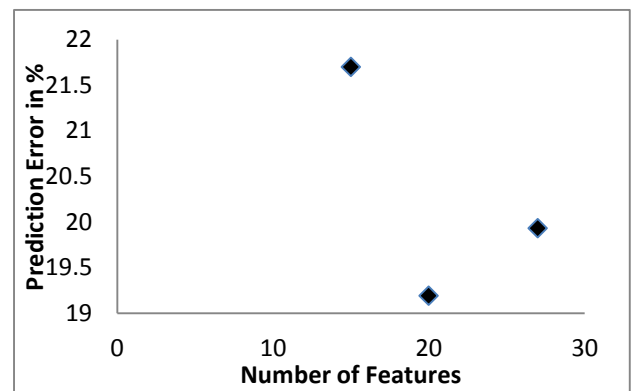


Fig. 1. Prediction error vs number of features graph for Bagging method.

Table 2- Classification error of various methods with all the features used for the training and prediction.

Classification Method	Classification Error in %
Random Subspace	19.62
Bagging	19.93
AdaBoost.M1	19.83
Random Forests	20.76
Single Tree	23.95

Table 3- Classification error of various methods with 20 most important features, calculated from Relief method, used for the training and prediction.

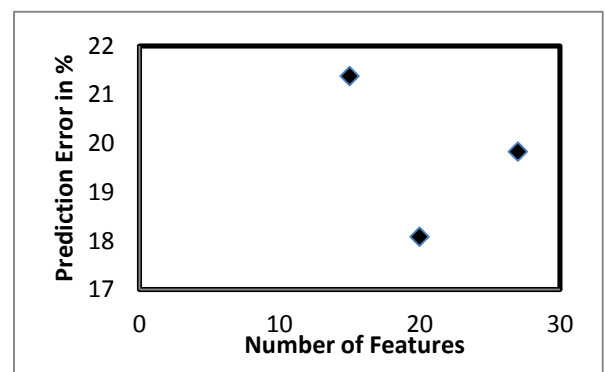


Fig. 2. Prediction error vs number of features graph for AdaBoost.M1 method.

Discussion

The results with all features, 20 most important features and 15 most important features are shown in Table 2, Table 3 and Table 4 respectively. Results suggest that with all features, Random Subspace performed best. Adaboost.M1 came second. Results for all the classifier methods except Random Subspace improved when the best 20 features were selected. For example, AdaBoost.M1 had 18.08 % classification error with 15 features, whereas with all features the error was 19.83 %. It suggests that some of the features were insignificant so removing those features improved the performance. As discussed for Random Forests, the ensemble method is useful when we have large number of features, hence removing features had adverse effect on the performance. With 10 most important features, the performance of all the classification methods degraded, it suggests that we removed some of the significant features from the datasets that had adverse effect on the performance of the classification methods. Results for Bagging and AdaBoost.M1 are presented in Fig. 1 and Fig. 2 respectively. These figures show that by removing the insignificant features the error first decreases and then increases. This suggests that the dataset has more than 15 important features.

The best performance is achieved with AdaBoost.M1 (18.08 % error) with 20 features. It shows that a good combination of ensemble method and feature selection method can be useful for steel plates defects. Results suggest that all the ensembles method performed better than single trees. This demonstrates the efficacy of the decision tree ensemble methods for steel plates defects problem.

IV. CONCLUSION AND FUTURE WORK

Data mining techniques are useful for predicting material properties. In this paper, we show that decision tree ensembles can be used to predict the steel plate faults. We carried out experiments with different decision tree ensemble methods. We found that AdaBoost.M1 performed best (18.08 % error) when we removed 12 insignificant features. In this paper, we showed that decision tree ensembles particularly Random subspace and AdaBoost.M1 are very useful for steel plates faults prediction. We also observed that removing 7 insignificant features improves the performance of decision tree ensembles. In this paper, we applied decision tree ensembles methods with Relief feature selection method for predicting steel plates faults, in future we will apply ensembles of neural networks [22] and support vector machines [23] for predicting for predicting steel plates faults. In future, we will use other feature section methods [17, 18] to study their performance for steel plate faults prediction.

REFERENCES

[1] Rajan, K., Materialstoday, 2005, 8, 10.

DOI: 10.1016/S1369-7021(05)71123-8

- [2] Kelly, A., Knowles, K. M., Crystallography and Crystal Defects, John Wiley & Sons, 2012.
DOI: 10.1002/9781119961468
- [3] Perzyk, M.; Kochanski, A., Kozłowski, J.; Soroczynski, A.; Biernacki R, Information Sciences, 2014, 259, 380-392. DOI
DOI: 10.1016/j.ins.2013.10.019
- [4] Han, J; Kamber, M.; Pei, J; Data Mining: Concepts and Techniques, Morgan Kaufmann, 2011.
DOI: 10.1145/565117.565130
- [5] Kuncheva, L. I., Combining pattern classifiers: methods and algorithms. Wiley-IEEE Press, New York, 2004.
DOI: .wiley.com/10.1002/0471660264.index
- [6] Quinlan, J. R., C4.5: Programs for machine learning. CA: Morgan Kaufmann, San Mateo, 1993.
DOI: 10.1007/BF00993309
- [7] Ahmad, A. Halawani S., and Albidewi, T., Expert systems with Applications, Elsevier, 2012, 39, 7, 6396-6401.
DOI: 10.1016/j.eswa.2011.12.029
- [8] Ahmad, A.; Dey, L., Pattern Recognition Letters, 2005, 26, 1, 43-56.
DOI: 10.1016/j.patrec.2004.08.015
- [9] Buscema, M.; Terzi, S.; Tastle, W.; Fuzzy Information Processing Society (NAFIPS), 2010 Annual Meeting of the North American New Meta-Classifiers, in NAFIPS 2010, Toronto (CANADA), 2010.
DOI: 10.1109/NAFIPS.2010.5548298
- [10] Fakhr, M.; Elsayad, A., Journal of Computer Science, 2012, 8, 4, 506-514.
DOI: 10.3844/jcssp.2012.506.514
- [11] Dietterich, T. G., In Proceedings of conference multiple classifier systems, Lecture notes in computer science, 2000, 1857, pp. 1-15.
DOI: 10.1007/3-540-45014-9_1
- [12] Breiman, L., J. Friedman, R. Olshen and C. Stone, Classification and Regression Trees. 1st Edn., Chapman and Hall, London, 1984.
ISBN: 0412048418
- [13] Breiman, L., Mach. Learn., 1996, 24, 2 123-140.
DOI: 10.1023/A:1018054314350
- [14] Freund, Y. and R.E. Schapire., J. Comput. Syst. Sci., 1997, 55, 119-139.
DOI: 10.1006/jcss.1997.1504
- [15] Ho, T. K.; IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20, 8, 832-844.
DOI: 10.1109/34.709601
- [16] Breiman, L., Machine Learning, 2001, 45, 1, 5-32.
DOI: 10.1023/A:1010933404324
- [17] Saeys, Y.; I. Inza, I.; Larranaga, P, Bioinformatics, 2007, 23, 19, 2507-2517.
DOI: 10.1093/bioinformatics/btm344
- [18] Jain, A.; Zongker, D, IEEE Trans. Pattern Anal. Mach. Intell., 1997, 19, 2, 153-158.
DOI: 10.1109/34.574797

- [19] Kira, K.; Rendell, L.;. AAAI-92 Proceedings, 1992.
ISBN:0-262-51063-4
- [20] Witten, I. H.; Frank, E., Data mining: practical machine learning
tools with java implementations. Morgan Kaufmann, San
Francisco, 2000.
ISBN: 0120884070
- [21] Frank, A.; Asuncion, A.; UCI machine learning repository
Irvine. University of California, 2010.
<http://archive.ics.uci.edu/ml/>
- [22] Bishop, C.M.. Pattern Recognition and Machine Learning. 1st
Edn., Springer-Verlag, New York, ISBN-10: 9780387310732,
2006.
ISBN: 0387310738
- [23] Vapnik, V., The Nature of Statistical Learning Theory.
Springer-Verlag, New York, 1995.