Bagging Random Trees for Estimation of Tissue Softness

S. B. Kotsiantis[1], G. E. Tsekouras [2], P. E. Pintelas[1]

[1] Educational Software Development Laboratory
Department of Mathematics, University of Patras, Greece.
{sotos, pintelas}@ math.upatras.gr
[2] Department of Cultural Technology and Communication
University of the Aegean, Mytilene, Greece
gtsek@ ct.aegean.gr

Abstract. We present an ensemble of classifiers that can be used to predict quality characteristics of an important process in pulp and paper industry: the tissue softness estimation. This classification problem is a difficult one since, with respect to our data set, the accuracy of all the well-known classifiers is below 68%. Contrary to that, the bagging random trees ensemble model is able to increase the accuracy up to 75%.

1. Introduction

A process in pulp and paper industry is the tissue softness estimation. The production of tissue paper, softness is one the key attributes to improving the marketability of the final product. Consistent production of soft tissue is the key in ensuring production; sales and cost targets are met.

A problem related to tissue softness estimation is that the quality characteristics of the final product cannot be objectively measured due to the lack of proper sensors and measuring devices [10]. Thus, in order to estimate the quality, an expert is usually employed to carry out the classification by assigning the tissue softness characteristics to certain predefined categories. However, in most of the cases the use of an expert to perform the above classification requires the interruption of the process in order to collect samples. Another issue is that the classification of the quality is subjective and may lead to errors, especially when the same expert is not always employed to perform the classification.

In this paper we use supervised machine learning algorithms to automatically determine the tissue softness characteristics. The implementation of various well-known classifiers to the available data set yielded accuracy below 68%. This fact directly implies that the process is a very difficult one. To increase the accuracy and to improve the classification efficiency we use a technique called bagging random decision trees, which can provide more accurate results. After a number of experiments we showed that, in contrast to the most well known classifiers and ensemble methods, the used technique can significantly increase the classification accuracy of the process.

The following section describes in brief the problem and the dataset of our study. In section 3 we present the basic design issues of the supervised machine learning techniques. In section 4, we present the used ensemble method. Section 5 compares the experimental results obtained by other well-known techniques with the used ensemble, while the concluding remarks are given in section 6.

2 Problem and data description

Consumer acceptance of tissue is strongly influenced by the level of softness. Softness is a complex human perception that is influenced by both physical properties and psychological factors. Both bulk softness and surface softness are factors in the overall perception of softness.

Tissue is used mainly for body care, so quality issues in tissue making are more important than in any other type of paper production. There are several parameters, which affect the tissue softness, including the constitution and consistency of pulp, pulp refining and thickness of stock in the paper machine. However, the most important parameters are the ones related to the drying process, which is designed so that the sheet detaches from the Yankee dryer prior to contact with the doctor blade and forms a loop (the microfold) [6]. These parameters are the doctor blade angle and wear, the crepe ratio (the ratio of the speed of the sheet on the Yankee dryer to the speed of the sheet on the reel) and the rates of the chemicals, which adjust coating and release. Due to this process, the produced tissue becomes softer, bulkier and more absorbent. Tissue softness is usually measured in a subjective manner by the touch of an expert who compares it with some scaled samples.

For this reason, a model that infers tissue softness from other variables, which are accurately measured, could be of great use. A model using supervised machine learning techniques can be developed for a tissue-producing process, using the tissue softness as the output variable and the following parameters as input variables:

- x1: cross directional tensile of tissue
- x2: machine directional tensile of tissue
- x3: machine directional stretch of tissue
- x4: Yankee coating rate (ml/min)
- x5: Yankee release rate (ml/min)

The available data consisted of 375 input–output pairs. More information about data set can be found in [18].

3 Machine learning techniques and estimation of tissue softness

Supervised machine learning is the exploration for algorithms that reason from externally supplied instances to produce general hypotheses, which will make predictions about future instances. In other words, the goal of supervised learning is to build

a concise model of the distribution of the class label in terms of the predictor features. The resulting classifier is then used to assign class labels to the testing instances where the values of the predictor features are known but the value of the class label is unknown.

Decision trees are trees that classify instances by sorting them based on attribute values. Each node in a decision tree represents an attribute in an instance to be classified, and each branch represents a value that the node can take. A recent overview of existing work in decision trees is provided in [14]. In rule induction systems, a decision rule is defined as a sequence of Boolean clauses linked by logical AND operators that together imply membership in a particular class [10]. The general goal is to construct the smallest rule-set that is consistent with the training data.

Artificial Neural Networks (ANNs) are another method of inductive learning and they all based on computational models of biological neurons [13]. A multi layer neural network consists of large number of units (neurons) joined together in a pattern of connections. First, the network is trained on a set of paired data to determine the input-output mapping. The weights of the connections between neurons are then fixed and the network is used to determine the classifications of a new set of data.

Naive Bayes classifier is the simplest form of Bayesian network [7]. This algorithm captures the assumption that every attribute is independent from the rest of the attributes, given the state of the class attribute. The assumption of independence is clearly almost always wrong. However, a large-scale comparison of Naive Bayes classifier with state-of-the-art algorithms on standard benchmark datasets found it sometimes to be superior to each of the other learning schemes [7].

Instance-based learning algorithms belong in the category of lazy-learning algorithms [13], as they delay the induction process until classification is performed. One of the most straightforward instance-based learning algorithms is the nearest neighbour algorithm [1]. K-Nearest Neighbour (kNN) assumes that the instances within a data set will generally exist in close proximity with other instances of the similar class.

The SVM technique revolves around the notion of a ‘margin’ that separates two data classes. Maximizing the margin, and thereby creating the largest possible distance between the separating hyperplanes can reduce the upper bound on the expected generalization error [5]. However, most real-world problems involve non-separable data for which no hyperplane exists that successfully separates the positive from negative instances in the training set. The solution is then to map the data into a higher-dimensional space and define a separating hyperplane there.

For the purpose of this study, a representative algorithm for each described learning technique was used. The most commonly used C4.5 algorithm [17] was the representative of the decision trees in our study. The most well known learning algorithm to estimate the values of the weights of a neural network - the Back Propagation (BP) algorithm [13] - was the representative of the ANNs. The Naive Bayes algorithm, which we used, is based on estimating marginal Gaussian estimators for numerical attributes. The 3-NN algorithm that combines robustness to noise and less time for classification than using a larger k for kNN was also used [1]. Finally, the Sequential Minimal Optimization (or SMO) algorithm was the representative of the SVMs as one of the fastest methods to train SVMs [16].

All accuracy estimates were obtained by averaging the results from stratified 10-fold cross-validation in our dataset. It must be mentioned that we used the free available source code for our experiments by the book [12]. As one can see, this classification problem is very difficult since the accuracy of all the well known classifiers in this domain is below 68% according to our dataset.

Table 1. Accuracy of simple models in our dataset

	NB	C4.5	3NN	BP	RIPPER	SMO
Accuracy	54.93%	67.20%	60.53%	58.93%	64.27%	60%

For this reason, there is a need for a more accurate technique. Combining classifiers is proposed as a new direction for the improvement of the classification accuracy [2]. When multiple classifiers are combined using voting methodology, we expect to obtain good results based on the belief that the majority of experts are more likely to be correct in their decision when they agree in their opinion. Voters can express the degree of their preference using a confidence score i.e. the probabilities of classifiers prediction.

4 Used technique

Random Forests, recently introduced by Breiman [4], have been shown to be a powerful classification technique. In bagging, single models are induced over bootstrap samples of the training data, and the classification is made by voting. Random Forests is a particular implementation of bagging in which each model is a Random Tree. In a Random Tree for each split, rather than considering all possible splits, a tournament is held with a small group of randomly selected splits, and the best split of this smaller group is chosen. Therefore randomness enters the Random Forest algorithm in two places: the bootstrap sample for each tree, and choice of splits for participation in tournament selection. Random Forests exhibit many desirable characteristics: parallelism, high accuracy, fast to train, built-in error predictor, and a tendency not to overfit [4].

In the case of Random Forests, tree decorrelation is attained through randomness in choosing the bootstrap sample to train each tree, and tournament selection in growing each tree. As tournament size F approaches the total number of possible splits in the training set, the procedure becomes equivalent to bagging decision trees (where each tree is grown deterministically). Any other value of F corresponds to Random Forests.

In our implementation, instead of select possible splits at random, evaluate them and choose best, we evaluate all splits, and choose at random one of the best. This modification would be expected to slightly increase the correlation between individual trees. We also do not use pruning, as the bagging-like procedure can only reduce the variance, but not improve on high bias [2]. Therefore one probably want a low bias learner, and no pruning means lower bias.

Finally, the used technique is presented in Fig. 1.

MODEL GENERATION

For i = 1 to *n*:

- Draw a bootstrap sample from the data.
- Grow a "random" tree, where at each node, the split is a random variable of the 3 best variables. The tree is grown to maximum size and not pruned back.
- Store the resulting models.

CLASSIFICATION

For each of the *n* models:

Predict class of instance for each of the *n* models.

Return class that has been predicted most often.

Fig. 1. The used ensemble

The used ensemble has a free parameter: the numbers of models (*n*). In our experiments, we used 90 models since most of the researchers used 100 models for their ensembles [2], [16], [20] and we wanted this technique to use less time for training. It must be also mentioned that the used ensemble can be easily parallelized.

The computations required to obtain the classifiers in each bootstrap sample are independent of each other. Therefore we can assign tasks to each processor in a balanced manner. By the end each processor has obtained a part of the ensemble. In the case we use the master-slave parallel programming technique, the method starts with the master splitting the work to be done in small tasks and assigning them to each slave. Then the master performs an iteration in which if a slave returns a result (this means it finished its work) then the master assigns it another task if there are still tasks to be executed. Once all the tasks have been carried out the master process obtains the results and orders the slaves to finish since there are not more tasks to be carried out. This parallel execution of the presented ensemble can achieve almost linear speedup.

In the following section, we briefly describe the most well known ensembles techniques and we compare the used technique with the other well known ensembles.

5 Experiments Results in relation to other ensembles techniques

Boosting [8] is similar in overall structure to bagging, except that keeps track of the performance of the data mining algorithm and concentrates on instances that have not been correctly learned. Instead of choosing the t training instances randomly using a uniform distribution, it chooses the training instances in such a manner as to favor the instances that have not been accurately learned. AdaBoost is a practical version of the boosting approach [8]. It was subsequently observed that Adaboost is in effect approximating a stagewise additive logistic regression model by optimising an exponential criterion [9]. This leads us to new variants of Adaboost that fit additive models

directly. One such variant is Logitboost, which uses the Newton-like steps to optimise the loss criterion [16].

MultiBoosting [20] is another method that can be considered as wagging committees formed by AdaBoost. Wagging is a variant of bagging; bagging uses resampling to get the datasets for training and producing a weak hypothesis, whereas wagging uses reweighting for each training example, pursuing the effect of bagging in a different way. Another meta-learner (DECORATE, Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples) is presented in [12] that uses a learner to build a diverse committee. This is accomplished by adding different randomly constructed examples to the training set when building new committee members. Finally, LMT method builds logistic model trees, which are classification trees with logistic regression functions at the leaves [11].

Firstly, we compared the used technique with the previous referred well known ensembles. All accuracy estimates were obtained by averaging the results from stratified 10-fold cross-validation in our data set. The t-test was also used to statistically compare the algorithms. Throughout, we speak of two results for the dataset as being "significant different" if the difference is statistical significant at the 1% level according to the corrected resampled t-test [15], with each pair of data points consisting of the estimates obtained in one of the 100 folds for our method and the other ensemble being compared.

The presented ensemble is significantly more accurate than Bagging C4.5 with 100 models, Boosting C4.5 with 100 models, Boosting NB with 100 models, Random Rorest with 100 trees and Decorate C4.5 with 100 iterations, according to the corrected resampled t-test [15] (see Table 2). We used 100 models for the ensembles methods such as [2].

Table 2. Comparing the used technique

	Bagging random trees	Bagging C4.5 (100 models)	Boosting C4.5 (100 models)	Boosting NB (100 models)	Random Rorest (100 trees)	Decorate C4.5
Accuracy	74.40%	69.60%	69.60%	62.67%	69.07%	69.07%

Moreover, the presented ensemble is significantly more accurate than Multiboost C4.5 with 100 models, Logitboost DS with 100 models, Bagging BP with 100 models, Boosting BP with 100 models and LMT classifier, according to the corrected resampled t-test [15] (see Table 3).

Table 3. Comparing the used technique

	Bagging random trees	Multiboost C4.5 (100 models)	Logitboost DS (100 models)	Bagging BP (100 models)	Boosting BP (100 models)	LMT
Accuracy	74.40%	69.60%	60.27%	61.07%	58.67%	68.27%

To sum up, the performance of the presented ensemble is more accurate than the other well-known classifiers and ensembles. The used ensemble can achieve an increase in classification accuracy from 11% to 35% compared to single learners. Moreover, the average relative accuracy improvement of the used methodology is from 7% to 26% in relation to the well known ensembles techniques. This indicates that it is possible to obtain a feasible solution to the problem with the used technique.

6 Conclusion

Tissue is a true consumer product and as such there are continuous demands to improve quality and performance. One of the most commonly sought after improvements is the softness of the tissue. Tissue softness is perceived by the consumer as the primary quality property

In our case study, the classification of tissue softness was a very difficult problem since the accuracy of all the well known classifiers in this domain was below 68% according to our dataset. Contrary to that, the bagging random trees ensemble model was able to increase the accuracy up to 75%. In a following work, we will try to further increase the classification accuracy.

References

1. Aha, D., Lazy Learning. Dordrecht: Kluwer Academic Publishers (1997).
2. Bauer, E. & Kohavi, R., An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning, 36, (1999) 105–139.
3. Breiman L., Bagging Predictors. Machine Learning, 24(3), (1996) 123-140.
4. Breiman, L., Random Forests. Machine Learning 45 (1), (2001) 5-32.
5. Burges, C., A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2, (1998) 1-47.
6. Corboy, W. G., Yankee dryers. In B. A. Thorp, Paper machine operations. In: Pulp and paper manufacture, vol. 7. GA:Joint Textbook Committee of the Paper Industry (1991).
7. Domingos, P. and Pazzani, M., On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, 29, (1997) 103-130.
8. Freund Y. and Schapire R. E., Experiments with a New Boosting Algorithm, Proceedings: ICML'96, p. 148-156.
9. J. H. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. The Annals of Statistics, 28 (2000) 337 – 374.
10. Furnkranz, J., Separate-and-Conquer Rule Learning, Artificial Intelligence Review, Vol. 13, (1999) 3-54.
11. N.Landwehr, M.Hall, E. Frank, Logistic Model Trees, 14th European Conference on Machine Learning ECML (2003) 241-252.
12. P. Melville and R. Mooney, Constructing Diverse Classifier Ensembles using Artificial Training Examples, Proc. of the IJCAI-2003, pp.505-510, Mexico, August 2003
13. Mitchell, T., Machine Learning. McGraw Hill (1997).

14. Murthy, S., Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. Data Mining and Knowledge Discovery, 2 (1998) 345–389.
15. Nadeau, C., Bengio, Y., Inference for the Generalization Error. Machine Learning 52(3), (2003) 239-281.
16. Platt, J., Using sparseness and analytic QP to speed training of support vector machines. In: Kearns, M. S., Solla, S. A. & Cohn D. A. (Eds.), Advances in neural information processing systems 11. MA: MIT Press (1999).
17. Quinlan, J. R., C4.5: Programs for machine learning. Morgan Kaufmann, San Francisco (1993).
18. Tsekouras, G., Sarimveis, H., Raptis, C., Bafas, G., A fuzzy logic approach for the classification of product qualitative characteristics, Computers and Chemical Engineering 26 (2002) 429–438.
19. Witten, I.H., Frank, E., Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, San Mateo, CA, (2000).
20. Webb, G. I., MultiBoosting: A Technique for Combining Boosting and Wagging, Machine Learning, 40, (2000) 159–196, Kluwer Academic Publishers.