Ensembles of probability estimation trees for customer churn prediction

Koen W. De Bock*

Dirk Van den Poel*

*Ghent University, Faculty of Economics and Business Administration, Department of Marketing, Tweekerkenstraat 2, B-9000 Gent

Abstract

Customer churn prediction is one of the most important elements of any Customer Relationship Management (CRM) strategy. In this study, a number of strategies are investigated to increase the lift of ensemble classification models. In order to increase lift performance, two elements of a number of well-known ensemble strategies are altered: (i) the potential of using probability estimation trees (PETs) instead of standard decision trees as base classifiers is investigated and (ii) a number of alternative fusion rules for the combination of ensemble member's outputs are proposed and compared. Experiments are conducted for four popular ensemble strategies (Bagging, the Random Subspace Method, SubBag and AdaBoost) on five real-life churn data sets. The results demonstrate the value of using PETs over standard decision trees in order to increase lift. Overall, the effect of the proposed strategies heavily depends on the chosen ensemble algorithm in which they are implemented.

Keywords

CRM, database marketing, churn prediction, PETs, probability estimation trees, ensemble classification, lift

First author: Koen W. De Bock: Koen.DeBock@UGent.be

Second and corresponding author: Dirk Van den Poel: Dirk.VandenPoel@UGent.be ; Tel.: + 32 9 264 89 80; Fax: + 32 9 264 42 79, http://www.crm.UGent.be

1. Introduction

In today’s business environment, an effective Customer Relationship Management strategy is of the foremost importance [1]. One of the most important aspects of CRM is customer retention, i.e. the prevention of customers from ceasing to buy products or services and leaving the company. One often-used strategy in this context is to identify the potential churners in an early stage, and to treat these customers accordingly by offering adapted incentives in order to re-establish their relationship with the company [2]. This practice is generally pursued in churn prediction.

In churn prediction, information from customers that is available in the company database is used to determine their proneness to attrite. Relevant predictive features typically include historical transactions from the customer with the company, demographic information of the customer and so on. Data mining techniques, and more specifically, classification algorithms, are then deployed to generalize the relationship that exists between a customer’s characteristics, and his or her probability to churn [3]. Once built, these models can be used to predict the future behavior of customers and to deliver targeting information for churn-preventing marketing campaigns.

In a churn-prediction model, accuracy is extremely important [4]. In this study, the use of ensemble learning for churn prediction is considered. Applications of ensemble classifiers to churn prediction include Random Forests [5], AdaBoost [6], AdaCost [7], Bagging [8], Stochastic Gradient Boosting [9], ensembles of Artificial Neural Networks [4] and the multi-classifier class-combiner technique [10]. These studies all demonstrate the beneficial impact of using ensemble classifiers over single classifiers for classification performance in the context of churn prediction.

An often-used performance criterion in churn prediction is lift (e.g. [8, 9, 11]). Lift measures how many times the classification model improves the identification of potential churners in the selection, over random guessing. A popular criterion in research on churn is top-decile lift, where the top 10 percent of customers with the highest probabilities to churn are considered. In this study, two strategies to improve lift performance of five well-known ensemble classifiers are presented. A first strategy involves using C4.4 probability estimation trees (PETs) [12] as base classifier in the ensemble classifiers instead of regular C4.5 [13] decision trees. Probability estimation trees are designed to generate better posterior probabilities than regular decision trees. They have been shown to provide better ranking capabilities than regular decision trees, and can hence be expected to improve lift performance when used in ensembles. A second strategy involves altering the fusion rules that are used to combine the predictions of individual ensemble member classifiers into aggregate predictions. A comparison is made between the standard fusion rules used in ensemble methods, as majority voting or average aggregation, to a weighted approach based on lift performance measures of member classifiers. In an experimental validation, the effect of both strategies will be investigated for Bagging [14], the Random Subspace Method (RSM; [15]) and AdaBoost [16]. Experiments are conducted for five real-life churn data sets from various European companies, belonging to different sectors. Also, in addition to top-decile lift, a range of alternative selection percentages is considered for the calculation of lift measures.

The paper is organized as follows. In Section 2, an overview is presented of probability estimation trees, ensemble classification and the ensemble classifiers considered in this study, and lift. Section 3 includes an experimental validation of the proposed algorithm variations. This section presents an overview of the churn data set selection, an overview of the conditions for the experimental comparison, and the results. In a final section, a conclusion is formulated, and limitations to the study and directions for future research are provided.

2. Methodology

2.1 Probability estimation trees

Tree induction algorithms, such as C4.5 [13], CART [17] and CHAID [18] are primarily designed to generate ‘crisp’ classifications: they map an instance, based on its values on a set of features, to precisely one class. However, many applications, such as churn prediction benefit from the availability of an estimation of confidence in a class prediction, such as a class membership probability. An alternative to standard classification trees in this aspect are probability estimation trees (PETs) that estimate class membership probabilities. Many PETs have been introduced in literature [12, 19-21]. In its most basic form, maximum likelihood probability estimates are generated as follows. Denote a decision tree that represents a C-class classification problem. At the end leaf for class $c \in C$ there are N instances belonging to C different classes, and k instances belonging to class c. The maximum likelihood posterior class membership probability for class c is then equal to $\frac{k}{N}$ [22]. It has been shown that PETs based on this rule often perform poorly at estimating class membership probabilities [12, 23]. In [12], Provost and Domingos identify a number of reasons for this. They argue that maximum likelihood probabilities are potentially highly inaccurate, especially if the number of training instances at an end leaf is small. Further, they argue that pruning, aimed at constructing small but accurate trees, results in lower quality of estimated class probabilities. In order to generate better PETs, they suggest C4.4, an adapted version of the C4.5 tree-building algorithm. In C4.4, maximum-likelihood estimates are smoothed by using the Laplace correction, which adjusts probability estimates in order to make them less extreme. The Laplace estimate used for C4.4 is given by $\frac{k+1}{N+C}$. Further, in C4.4, no pruning is applied, and 'collapsing', a secondary pruning strategy inherent to C4.5, is no longer performed.

In [12], Provost and Domingos applied Bagging to C4.4 PETs and demonstrated in their experiments that an ensemble of PETs substantially improved AUC performance for a majority of the examined data sets. However, they did not consider alternative ensemble strategies. Moreover, the lift performance of PETs and particularly ensembles of PETs has, to the best of our knowledge, never been analyzed in the context of customer churn prediction.

2.2 Ensemble classification

Ensemble classification has been a popular field of research in recent years. Multiple studies have demonstrated the beneficial effect of combining many classification models into aggregated ensemble classifiers on classification accuracy (e.g., [24-26]). While several algorithms have been proposed in literature, many are inspired by two classical ensemble strategies: Bagging [14] and Boosting [27]. In Bagging (Bootstrap aggregating), each member classifier in the ensemble is trained on a bootstrap sample (i.e., a random sample taken with replacement and with the same size) of the original training data. Member outputs are aggregated using majority voting (also known as plurality voting): instances are assigned the class that is most frequently assigned by the ensemble members [14]. Bagging can introduce a significance improvement in accuracy as a result of a reduction of variance versus individual decision trees. The most well-known boosting algorithm is AdaBoost [16, 27]. In AdaBoost, instances that are mislabeled receive higher weight during consecutive training iterations and hence, the classifier is forced to concentrate on hard-to-predict instances. A related method to Bagging is the Random Subspace Method (RSM, [15]), also known as attribute bagging [28]. In RSM, a random feature subset is sampled for the training of an ensemble member.

An important element of any ensemble classifier algorithm is the fusion rule, used to aggregate ensemble member's outputs to ensemble predictions. A categorization is often made between fusion rules for label outputs and fusion rules for continuous outputs [22]. A well-studied and simple combiner is plurality voting, as described earlier in the case of Bagging. RSM and Random Forest implement average aggregation (sometimes referred to as mean combination rule), which takes the average of the ensemble members' outputs. For both methods, weights can be assigned to ensemble members which score higher on a performance metric of choice to obtain weighted majority voting or weighted average aggregation [22]. Training error rates are often used as weights. In this study, top-p-th percentile and derivations are introduced as weights for weighted average aggregation fusion rules.

2.3 Lift

In the context of churn prediction, lift focuses on the segment of customers with the highest risk to the company, i.e. customers with the highest probability to churn. The definition of lift depends upon the percentage of riskiest customers one is considering for a retention campaign. Suppose that a company is interested in the top *p*-th percentile of most likely churners, based on predicted churn probabilities. The top *p*-th percentile lift then equals the ratio of the proportion of churners in the top *p*-th percentile of ordered posterior churn probabilities, $\pi_{p\%}$, to the churn rate in the total customer population, π ; *top p-th percentile lift* = $\frac{\pi_{p\%}}{\pi}$. As the proportion of customers that a company is able and willing to target depends on the specific content, the experiments will calculate lift performance for different percentiles. The concept of directly optimizing the lift measure is similar to the one proposed in [29]. The authors maximize the number of purchases at a given mailing depth (used as a constraint) for database marketing.

3. Experimental validation

3.1 Data

To investigate the suitability of probability estimation trees as base classifiers in ensemble classifiers for increasing lift in churn prediction, this study considers five real-life churn data sets from different business contexts and for different products or services. The characteristics of these data sets are provided in Table 1.

Table 1: data set description

Data set			
	Instances	*Features*	*Minority class percentage*
Bank1	23,562	236	3.52
Bank2	42,783	164	11.14
Supermarket chain	32,371	46	25.15
DIY chain	3,827	15	28.14
Bank3	20,456	137	5.99

As shown in Table 1, churn data sets are typically characterized by high dimensionality, both in terms of number of features and number of instances. Another issue is the class imbalance of the data. Churn is usually a rare event [30]. Many techniques have been proposed to deal with class imbalance in churn prediction [8, 9]. In [31], the effect of class imbalance on a number of performance metrics is analyzed for probability estimation trees. This study indicates the importance of an appropriate treatment for the problem of class imbalance for the quality of probability estimates of PETs. While the authors suggest a wrapper method to determine an optimal sampling level of undersampling, in this study, majority class instances in the training data sets are undersampled in order to obtain balanced class distributions.

3.2 Experimental settings

In this study, we consider three ensemble classifiers: Bagging, RSM and AdaBoost. C4.4 and C4.5 base classifiers are implemented using the J48 classifier, which is a C4.5 implementation, available in WEKA [32]. C4.4 is implemented as in [12]. Bagging, RSM, AdaBoost and the proposed variations are implemented in MATLAB. Ensemble sizes are set to 100 constituent ensemble members. One final parameter is the random feature subset size for RSM. This parameter is set equal to the square root of the number of features in the respective data sets, as suggested in [25] and [33].

A first comparison involves the use of C4.4 PETs versus standard C4.5 classification trees as base classifiers in ensemble classifiers. In a second comparison, the influence of the introduction of alternative fusion rules based on top *p*-th percentile lift is investigated. Throughout the comparisons, different lift definitions are considered, with *p* ranging from 50 to 5. More specifically,

$p \in \{50,40,30,20,10,5, p_r\}$ where p_r is the (rounded) actual churn rate as observed in the training data as provided in Table 1. This additional percentile represents the plausible strategy of companies to target as many potential churners as they expect to emerge based on past experience.

Weighted average aggregation is applied using as weights: (i) top *p*-th percentile lift, further referred to as $lift_p$, (ii) $(lift_p - 1)$,, and (iii) rescaled $lift_p$, defined as $\frac{lift_p - \min(lift_p)}{\min(lift_p)}$. Alternative (ii) is inspired by fact that a lift of one implies a model not outperforming random guessing. Only if the classifier outperforms random guessing is $(lift_p - 1)$ a positive weight for the respective classifier. In alternative (iii), the minimal observed lift is set as a minimum threshold for classifiers to receive a positive weight. Finally, in the case of Bagging, average aggregation is added as an additional fusion rule to investigate its value for lift performance over plurality voting.

Reported results are averaged over a 5x2-fold cross-validation[1]. Within a 2-fold cross-validation, the training set is randomly split into two parts; the first part is used for model training, while the second part is used for model validation and vice versa.

3.3 Results

Results are reported as counts of wins, losses and ties. When a reference algorithm performs better (worse) in terms of lift performance, a win (loss) is registered, while equal lift performance between a reference and a benchmark algorithm results in a tie. Tables 2 to 4 provide wins-losses-ties counts for variations based on Bagging, RSM and AdaBoost.

Table 2: Experimental results for Bagging: wins-losses-ties

Algorithm	*Bagging (C4.5 + averaging)*	*Bagging (C4.5 + lift weights)*	*Bagging (C4.5+ (lift-1) weights)*	*Bagging (C4.5 + rescaled lift weights)*	*Bagging (C4.4 + averaging)*	*Bagging (C4.4 + lift weights)*	*Bagging (C4.4 + (lift-1) weights)*	*Bagging (C4.4 + rescaled lift weights)*
Bagging (C4.5 + averaging)	-	1/1/32	1/2/32	3/0/32	0/10/25	0/9/26	0/11/24	0/9/26
Bagging (C4.5 + lift weights)	1/1/32	-	2/1/32	3/0/32	0/9/26	0/9/26	0/10/25	0/9/26
Bagging (C4.5+ (lift-1) weights)	2/1/32	1/2/32	-	2/0/33	0/9/26	0/9/26	0/9/26	0/8/27
Bagging (C4.5 + rescaled lift weights)	0/3/32	0/3/32	0/2/33	-	0/11/24	0/10/25	0/10/25	0/9/26
Bagging (C4.4 + averaging)	10/0/25	9/0/26	9/0/26	11/0/24	-	0/0/35	0/0/35	5/1/29
Bagging (C4.4 + lift weights)	9/0/26	9/0/26	9/0/26	10/0/25	0/0/35	-	0/0/33	4/2/29
Bagging (C4.4 + (lift-1) weights)	11/0/24	10/0/25	9/0/26	10/0/25	0/0/35	0/0/33	-	3/2/30
Bagging (C4.4 + rescaled lift weights)	9/0/26	9/0/26	8/0/27	9/0/26	1/5/29	2/4/29	2/3/30	-

Table 3: Experimental results for RSM: wins-losses-ties

Algorithm	*RSM (C4.5)*	*RSM (C4.5 + lift weights)*	*RSM (C4.5+ (lift-1) weights)*	*RSM (C4.5 + rescaled lift weights)*	*RSM (C4.4)*	*RSM (C4.4 + lift weights)*	*RSM (C4.4 + (lift-1) weights)*	*RSM (C4.4 + rescaled lift weights)*
RSM (C4.5)	-	0/8/27	0/14/21	0/14/21	0/2/33	0/2/33	0/4/31	0/4/31
RSM (C4.5 + lift weights)	8/0/27	-	1/7/27	1/6/28	0/1/34	0/0/35	0/1/34	0/1/34
RSM (C4.5+ (lift-1) weights)	14/0/21	7/1/27	-	3/0/32	2/0/33	0/1/34	0/1/34	0/1/34
RSM (C4.5 + rescaled lift weights)	14/0/21	6/1/28	0/3/32	-	1/1/33	0/2/33	0/1/34	0/0/35
RSM (C4.4)	2/0/33	1/0/34	0/2/33	1/1/33	-	1/15/19	0/14/21	0/15/20
RSM (C4.4 + lift weights)	2/0/33	0/0/35	1/0/34	2/0/33	15/1/19	-	0/10/25	1/12/22
RSM (C4.4 + (lift-1) weights)	4/0/31	1/0/34	1/0/34	1/0/34	14/0/21	10/0/25	-	4/1/29
RSM (C4.4 + rescaled lift weights)	4/0/31	1/0/34	1/0/34	0/0/35	15/0/20	12/1/22	1/4/29	-

Table 4: Experimental results for SubBag: wins-losses-ties

Algorithm	*SubBag (C4.5)*	*SubBag (C4.5 + lift weights)*	*SubBag (C4.5+ (lift-1) weights)*	*SubBag (C4.5 + rescaled lift weights)*	*SubBag (C4.4)*	*SubBag (C4.4 + lift weights)*	*SubBag (C4.4 + (lift-1) weights)*	*SubBag (C4.4 + rescaled lift weights)*
SubBag (C4.5)	-	1/4/30	2/5/28	1/4/30	2/0/33	2/3/30	4/3/28	3/3/29
SubBag (C4.5 + lift weights)	4/1/30	-	1/6/28	1/4/30	2/0/33	2/1/32	3/4/28	3/3/29
SubBag (C4.5+ (lift-1) weights)	5/2/28	6/1/28	-	0/0/31	2/0/33	2/0/33	2/3/30	2/1/32
SubBag (C4.5 + rescaled lift weights)	4/1/30	4/1/30	0/0/31	-	2/0/33	2/0/33	2/3/30	2/1/32
SubBag (C4.4)	0/2/33	0/2/33	0/2/33	0/2/33	-	0/6/29	0/6/29	0/6/29
SubBag (C4.4 + lift weights)	3/2/30	1/2/32	0/2/33	0/2/33	6/0/29	-	1/4/30	1/3/31
SubBag (C4.4 + (lift-1) weights)	3/4/28	4/3/28	3/2/30	3/2/30	6/0/29	4/1/30	-	2/0/26
SubBag (C4.4 + rescaled lift weights)	3/3/29	3/3/29	1/2/32	1/2/32	6/0/29	3/1/31	0/2/26	-

Table 5: Experimental results for AdaBoost: wins-losses-ties

Algorithm	*AdaBoost (C4.5)*	*AdaBoost (C4.5 + lift weights)*	*AdaBoost (C4.5+ (lift-1) weights)*	*AdaBoost (C4.5 + rescaled lift weights)*	*AdaBoost (C4.4)*	*AdaBoost (C4.4 + lift weights)*	*AdaBoost (C4.4 + (lift-1) weights)*	*AdaBoost (C4.4 + rescaled lift weights)*

AdaBoost (C4.5)	-	5/5/25	11/1/23	13/2/20	3/23/9	0/21/14	0/21/14	0/21/14
AdaBoost (C4.5 + lift weights)	5/5/25	-	18/0/17	20/0/15	4/24/7	1/24/10	2/24/9	2/24/9
AdaBoost (C4.5+ (lift-1) weights)	1/11/23	0/18/17	-	15/0/20	1/25/9	0/24/11	0/24/11	0/24/11
AdaBoost (C4.5 + rescaled lift weights)	2/13/20	0/20/15	0/15/20	-	2/25/8	1/25/9	0/25/10	0/25/10
AdaBoost (C4.4)	23/3/9	24/4/7	25/1/9	25/2/8	-	3/12/20	3/11/21	3/11/21
AdaBoost (C4.4 + lift weights)	21/0/14	24/1/10	24/0/11	25/1/9	12/3/20	-	1/0/34	1/0/34
AdaBoost (C4.4 + (lift-1) weights)	21/0/14	24/2/9	24/0/11	25/0/10	11/3/21	0/1/34	-	0/0/23
AdaBoost (C4.4 + rescaled lift weights)	21/0/14	24/2/9	24/0/11	25/0/10	11/3/21	0/1/34	0/0/23	-

A first set of observations is derived for the Bagging variations. The results in tables indicate how the lift performance of Bagging can be increased by replacing standard C4.5 decision trees with C4.4 PETs, and confirm findings in [12]. The infl

Table 3 presents results for the Random Subspace Method. Here different results are found. The introduction of C4.4 as base classifier in RSM does not improve performance over RSM with C4.5 base classifiers. However, fusion rules based on top *p*-th percentile lift invoke substantial improvement of lift performance over average aggregation for RSM with C4.4 base classifiers. All lift-based fusion rules demonstrate equivalent performance.

Finally, Table 4 shows wins, losses and ties based on experimental results for AdaBoost. Here, both modified base learners and adapted fusion rules contribute to an improvement of lift performance. The introduction of C4.4 base learners generates a marked improvement over using C4.5 base classifiers. The use of lift-based fusion rules invokes a further increase. The best performance is observed for regular lift weights.

Conclusion

Customer retention and churn prediction are important elements of Customer Relationship Management (CRM) strategies. An often-used evaluation metric for churn prediction models is lift, which measures the degree to which the model is better in identifying the customers most likely to churn, over random guessing. This study investigates strategies to improve lift performance of three well-known ensemble algorithms. The first is the adoption of C4.4 probability estimation trees (PETs) as base classifiers, instead of regular C4.5 classification trees. Probability estimation trees are especially designed to generate class membership probabilities of higher quality. C4.4 trees are a variation of C4.5, implementing Laplace correction and avoiding pruning strategies. The second strategy involves replacing standard fusion rules for ensemble members' outputs by weighted average aggregation, using three alternative lift-based weight sets. Both strategies are applied to three well-known ensemble algorithms: Bagging, the Random Subspace Method (RSM) and AdaBoost. Experiments on five real-life churn prediction data sets are conducted to compare C4.5 and C4.4 trees as base classifiers, and original versus the proposed fusion rules.

The results indicate variation in the effect of the proposed strategies on lift performance depending on the nature of the ensemble algorithm: (i) Bagging greatly benefits from adopting C4.4 base classifiers and average aggregation as a fusion rule. Lift-based weighted averaging does not substantially improve lift performance; (ii) while RSM does not benefit from adopting C4.4 trees as base classifiers, weighted average aggregation is a viable strategy to increase lift performance; and (iii) Lift performance of AdaBoost can be substantially improved by implementing C4.4 base classifiers and weighted average aggregation based on lift. Overall, there is no clear dominance of any of the proposed lift-based weights.

Future work includes the adoption of different PET algorithms and alternative ensemble strategies, such as the recently proposed Rotation Forests [26] and application of error decomposition and techniques to gain insight in the accuracy-diversity trade-off to gain insight in the specific conditions that need to be fulfilled to successfully ensemble PETs for increasing lift performance.

Acknowledgements
The authors thank the reviewers for reviewing the paper and Ghent University for funding the PhD project of Koen W. De Bock.

References

1. Reinartz, W., Kumar, V.: The mismanagement of customer loyalty. Harvard Business Review **80** (2002) 86-94

2. Burez, J., Van den Poel, D.: CRM at a pay-TV company: Using analytical models to reduce customer attrition by targeted marketing for subscription services. Expert Systems with Applications **32** (2007) 277-288

3. Shaw, M.J., Subramaniam, C., Tan, G.W., Welge, M.E.: Knowledge management and data mining for marketing. Decision Support Systems **31** (2001) 127-137

4. Kim, Y.S.: Toward a successful CRM: variable selection, sampling, and ensemble. Decision Support Systems **41** (2006) 542-553

5. Larivière, B., Van den Poel, D.: Predicting customer retention and profitability by using random forests and regression forests techniques. Expert Systems with Applications **29** (2005) 472-484

6. Jinbo, S., Xiu, L., Wenhuang, L.: The application of AdaBoost in customer churn prediction. 2007 International Conference on Service Systems and Service Management (ICSSSM'07) (2007) 513-518

7. Glady, N., Baesens, B., Croux, C.: Modeling churn using customer lifetime value. European Journal of Operational Research **197** (2009) 402-411

8. Lemmens, A., Croux, C.: Bagging and boosting classification trees to predict churn. Journal of Marketing Research **43** (2006) 276-286

9. Burez, J., Van den Poel, D.: Handling class imbalance in customer churn prediction. Expert Systems with Applications **36** (2009) 4626-4636

10. Wei, C.P., Chiu, I.T.: Turning telecommunications call details to churn prediction: a data mining approach. Expert Systems with Applications **23** (2002) 103-112

11. Coussement, K., Benoit, D.F., Van den Poel, D.: Improved marketing decision making in a customer churn prediction context using generalized additive models. Expert Systems with Applications **37** (2010) 2132-2143

12. Provost, F., Domingos, P.: Tree induction for probability-based ranking. Machine Learning **52** (2003) 199-215

13. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kauffman Publishers, San Mateo, CA. (1993)

14. Breiman, L.: Bagging predictors. Machine Learning **24** (1996) 123-140

15. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 832-844

16. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Saitta, L. (ed.): Thirteenth International Conference on Machine Learning. Morgan Kauffman (1996) 148–156

17. Breiman, L., Friedman, J.H., Olsen, R.A., Stone, C.J.: Classification and regression trees. Chapman & Hall / CRC (1984)

18. Kass, G.V.: An exploratory technique for investigating large quantities of categorical data. Applied statistics **29** (1980) 119-127

19. Clemençon, S., Vayatis, N.: Tree-Based Ranking Methods. IEEE Transactions on Information Theory **55** (2009) 4316-4336

20. Zhang, H., Su, J.: Learning probabilistic decision trees for AUC. Pattern Recognition Letters **27** (2006) 892-899

21. Jiang, L.X., Li, C.Q., Cai, Z.H.: Learning decision tree for ranking. Knowl. Inf. Syst. **20** (2009) 123-135

22. Kuncheva, L.I.: Combining pattern classifiers: methods and algorithms. John Wiley & Sons, Hoboken, New Jersey (2004)

23. Provost, F., Fawcett, T., Kohavi, R.: The Case against Accuracy Estimation for Comparing Induction Algorithms. In: Shavlik, J. (ed.): 15th International Conference on Machine Learning (ICML-1998). Morgan Kaufman (2000) 445-453

24. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning **36** (1999) 105-139

25. Breiman, L.: Random forests. Machine Learning **45** (2001) 5-32

26. Rodriguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation forest: A new classifier ensemble method. IEEE Transactions on Pattern Analysis and Machine Intelligence **28** (2006) 1619-1630

27. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences **55** (1997) 119-139

28. Bryll, R., Gutierrez-Osuna, R., Quek, F.: Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. Pattern Recognition **36** (2003) 1291-1302

29. Prinzie, A., Van den Poel, D.: Constrained optimization of data-mining problems to improve model performance: A direct-marketing application. Expert Systems with Applications **29** (2005) 630-640

30. Neslin, S.A., Gupta, S., Kamakura, W., Lu, J.X., Mason, C.H.: Defection detection: Measuring and understanding the predictive accuracy of customer churn models. Journal of Marketing Research **43** (2006) 204-211

31. Cieslak, D., Chawla, N.: Analyzing PETs on imbalanced datasets when training and testing class distributions differ. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.): 12th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Vol. 5012 (2008) 519-526

32. Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations **1** (2009)

33. Bernard, S., Heutte, L., Adam, S.: Influence of hyperparameters on Random Forest accuracy. In: Benediktsson, J.A., Kittler, J., Roli, F. (eds.): 8th International Workshop on Multiple Classifier Systems (MCS 2009). Springer-Verlag, Reykjavik, Iceland (2009) 171-180