Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Expert Systems with Applications 34 (2008) 1721–1732
Expert Systems with Applications
www.elsevier.com/locate/eswa

Random Forests for multiclass classification: Random MultiNomial Logit ☆

Anita Prinzie *, Dirk Van den Poel

Department of Marketing at Ghent University, Ghent, Belgium

Abstract

Several supervised learning algorithms are suited to classify instances into a multiclass value space. MultiNomial Logit (MNL) is recognized as a robust classifier and is commonly applied within the CRM (Customer Relationship Management) domain. Unfortunately, to date, it is unable to handle huge feature spaces typical of CRM applications. Hence, the analyst is forced to immerse himself into feature selection. Surprisingly, in sharp contrast with binary logit, current software packages lack any feature-selection algorithm for MultiNomial Logit. Conversely, Random Forests, another algorithm learning multiclass problems, is just like MNL robust but unlike MNL it easily handles high-dimensional feature spaces. This paper investigates the potential of applying the Random Forests principles to the MNL framework. We propose the Random MultiNomial Logit (RMNL), i.e. a random forest of MNLs, and compare its predictive performance to that of (a) MNL with expert feature selection, (b) Random Forests of classification trees. We illustrate the Random MultiNomial Logit on a cross-sell CRM problem within the home-appliances industry. The results indicate a substantial increase in model accuracy of the RMNL model to that of the MNL model with expert feature selection.

Keywords: Multiclass classifier design and evaluation; Feature evaluation and selection; Data mining methods and algorithms; Customer relationship management (CRM)

1. Introduction

Multiclass problems, i.e. the classification of instances into multiple discrete classes, are omnipresent and hence the study object of ongoing research in a plenitude of domains: pattern recognition (e.g. image annotation for supporting keyword retrieval of images (Goh, Chang, & Li, 2005)), text classification (Novovicova & Malik, 2003), biology (e.g. plankton image recognition (Luo et al., 2004)), medicine (e.g. distinguishing different disease types (Huang et al., 2005), psychology (Devillers, Vidrascu, & Lamel, 2005)), and last but not least marketing/CRM, i.e. Customer Relationship Management (e.g. next-product to buy models (Prinzie & Van den Poel, 2005)). Several learning algorithms have been proposed to handle such multiclass classification. While some algorithms are merely an extension or combination of intrinsically binary classification methods (e.g. multiclass SVM as one-versus-one or one-versus-all binary SVMs), other algorithms like nearest shrunken centroids and MultiNomial Logit (MNL) are specifically designed to map features to a multiclass output vector. MNL's robustness (Agrawal & Schorling, 1996) is greatly appreciated and therefore, MNL has a proven track record in many disciplines amongst them transportation research (Anas, 1983) and CRM (Baltas & Doyle, 2001). Unfortunately, MNL suffers from the curse of dimensionality thereby implicitly necessitating feature selection, i.e. the selection of a best subset of variables of the input feature set. In glaring contrast to binary logit, to date, software

☆ This work was supported in part by Flemish Research Fund (FWO, Grantno. G0055.01).

* Corresponding author. Tel.: +32 9 264 35 20.
E-mail addresses: Anita.Prinzie@UGent.be (A. Prinzie), Dirk.VandenPoel@UGent.be (D. Van den Poel).
URL: www.crm.ugent.be

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.01.029

packages mostly lack any feature-selection algorithm for MNL. This absence constitutes a serious problem as (a) the CRM domain is, similar to fields such as text categorization (Leopold & Kindermann, 2002) and image retrieval (Swets & Weng, 1996), characterized by many input variables, often hundreds, with each one containing only a small amount of information and (b) human expert feature selection is known to be suboptimal (Liu & Yu, 2005).

Recently, Random Forests (Breiman, 2001), i.e. a classifier combining a forest of decision trees grown on random input vectors and splitting nodes on a random subset of features, have been introduced for the classification of binary *and* multiclass outputs. The majority of papers employs Random Forests for the prediction of a *binary* target and adduces further proof of Random Forests' high accuracy (Buckinx & Van den Poel, 2005; Lunetta, Hayward, Segal, & Eerdewegh, 2004; Schwender, Zucknick, Ickstadt, & Bolt, 2004). To the best of our knowledge, only two studies (Huang et al., 2005; Luo et al., 2004) investigated the potential of Random Forests in a multiclass setting. Luo et al. (2004) compare the predictive performance of Random Forests for the recognition of plankton images with that of C4.5, a cascade correlation neural network, bagged decision trees and SVM. In a simulation study, Huang et al. (2005) assess that Random Forests have a slight edge to partial least squares, penalized partial least squares, LASSO and nearest shrunken centroids. Random Forests eliminates decision trees' instability disadvantage but shares its capacity to cope with huge features spaces. In fact, feature selection is implicitly incorporated during each tree construction. At each node of one of the decision trees in the forest, the best variable to split on out of a random subset of variables is selected. During classification, just those features needed for the test pattern under consideration are involved (Jain, Duin, & Mao, 2000).

Given Random Forests' robustness and competence for analyzing large feature spaces *and* MNLs weakness in the latter, why not applying the Random Forests approach to MNL, i.e. building a forest of MNLs, to unite the best of both worlds? To this end, this paper proposes a new method, the Random MultiNomial Logit (RMNL), a Random Forest of MultiNomial Logits. We explore the potential of RMNL and compare it with the traditional MNL with human expert feature selection. Our new innovative RMNL method is demonstrated on a cross-sell case within the home-appliances industry.

The remainder of this paper is structured as follows. In the methodology section, we explain the technical features of Random Forests and MNL. Next, we elaborate on how we adopt the Random Forests ideas and apply them on MNL to construct a Random Forest of MNLs, i.e. a Random MultiNomial Logit. In Section 3, we describe the CRM application in which we illustrate our new method. Section 4 discusses the main findings. Finally, in the last Section final conclusions are drawn and several avenues for further research suggested.

2. Methodology

2.1. Problem definition

We consider a supervised multinomial classification problem. Let (X, Y) be a pair of random variables. Given X the feature space containing M predictors, the X values are vectors of the form $\langle x_1, x_2, \ldots, x_M \rangle$ taking discrete (e.g. total number of products bought) or real values (e.g. total monetary value). The Y values are class values drawn from a discrete set of classes $\{1, 2, \ldots, K\}$. A learning algorithm c, in this paper Random Forests, MultiNomial Logit or Random MultiNomial Logit, makes a hypothesis h_c about the true function f; $y = f(x)$ based on N_1 labeled training instances $\{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$. Hence, the learning algorithm outputs a *classifier* $h_c : X \to \{1, 2, \ldots, K\}$. Given a loss function L, in this paper defined as the total number of misclassified observations (1), the objective is to find a classifier h_c or a set of classifiers $\{h_1, \ldots, h_C\}$ (i.e. ensemble) minimizing the loss function for N_2 new unseen instances, i.e. test data.

$$L = \sum_{i=1}^{N_2} (h_C(X_i) \neq Y|X_i) \tag{1}$$

2.2. Classification trees and Random Forests

A classification tree is a non-parametric classifier h_c recursively partitioning the observations into subgroups with a more homogeneous categorical response (Breiman, 1984). Hence, classification trees make no assumption about the form of the underlying relationships between the predictor variables and the response, i.e. no assumptions about f; $y = f(x)$. Therefore, this learning algorithm might be very useful given non-linearly associated predictors. Besides this functional flexibility, decision trees are well adapted to deal with heterogeneity, as separate models are automatically fit to subsets of data defined by early splits in the tree. Finally, classification trees are very efficient at selecting from large numbers of predictor variables, which is characteristic of CRM data mining and is reflected in the profusion of papers addressing the feature-selection problem (Buchtala, Klimek, & Sick, 2005). Despite these advantages, early methods for constructing decision trees have been found to be unstable; i.e. a small perturbation in the data causes large changes in predictions (Breiman, 1996). To provide stability, researchers introduced new decision tree methods making use of ensembles of predictors, amongst them boosting and bagging (Breiman, 1996) (bagged predictors). Bagging; bootstrap aggregation, builds an ensemble of classifiers, here decision trees, grown on a random selection (with replacement) of examples in the training set; bootstrap sample S_T. The T decision trees' hypotheses $\{h_1(X, S_1), h_2(X, S_2), \ldots, h_T(X, S_T)\}$ are combined into an aggregated hypothesis h_c, i.e. a bagged predictor, by letting the T decision trees vote for the most popular class.

Random Forests builds on this bagging technique. Random Forests (Breiman, 2001) is a bagged classifier h_c combining a collection of T classification or regression trees (i.e. forest of trees), here T classification trees. Each tree t is grown on a different bootstrap sample S_t containing N_1 randomly drawn instances with replacement from the original training sample. Besides bagging Random Forests also employs random feature selection. At each node of the decision tree t, m variables are selected at random out of the M input vectors and the best split selected out of these m. Each decision tree t is grown using CART methodology to the largest extent possible. To classify a new instance, put the input vector down the T trees in the forest. Each tree votes for the predicted class. Finally the bagged predictor is obtained by majority vote, i.e. the instance is classified into the class having the most votes over all T trees in the forest.

The two sources of randomness, random inputs and random features, make Random Forests accurate classifiers in different domains (Huang et al., 2005; Wu et al., 2003). Using random feature selection to split each node yields error rates that compare even favorably to Adaboost (Freund & Schapire, 1996), but are more robust with respect to noise (Dietterich, 2000). This noise robustness is of the utmost importance to CRM applications typically tormented by noisy data.

The forest error rate depends on the correlation between any two trees in the forest and the strength of each individual tree in the forest. There is a positive relationship between m and the strength and correlation. Breiman (2001) estimates the error rate on out-of-bag data (i.e. oob data). Each tree is constructed on a different bootstrap sample S_t. As in each bootstrap training set about one third of the instances are left out (i.e. out-of-bag), we can estimate the test set classification error by putting each case left out of the construction of the tth tree down the tth tree. The oob error estimate is the misclassification proportion on oob data. Hence, in Random Forests there is no need for cross-validation or a separate test set. Another application of this oob data is to determine variable importance. To measure the importance of the mth variable, randomly permute this variable in the oob data and put the data down the corresponding tree. Subtract the number of votes for the correct class in the variable-m-permuted data from the number of correct votes in the untouched data and average over all trees T in the forest. This is the *raw importance score* for variable m and represents the percent increase in misclassification rate as compared to the oob rate on non-permuted data. Assuming this score is independent from tree to tree, a z-score is obtained by dividing the raw score by its standard error computed in the classical way.

2.3. MultiNomial Logit and Random MultiNomial Logit

Besides decision trees and Random Forests, MultiNomial Logit (MNL) is another learning algorithm suitable to supervised learning a multiple class problem. We rely on multinomial-discrete choice modeling (Ben-Akiva & Lerman, 1985). The random-utility function U_{ik} of individual i for choice k belonging to choice set D with $K > 2$ (cf. multiclass) is decomposed into a deterministic and stochastic component (2):

$$U_{ik} = \boldsymbol{\beta}' x_{ik} + \boldsymbol{\varepsilon}_{ik} \tag{2}$$

where x is a matrix of observed attributes which might be choice (e.g. price of product) or individual specific (e.g. age of customer), β' is a vector of unobserved marginal utilities (parameters) and ε_{ik} is an unobserved random error term (i.e. disturbance term or stochastic component). Different assumptions on the error term of the random-utility function U_{ik} of individual i for choice k belonging to choice set D_K ($K > 2$), give rise to different classes of models. In this paper, we apply the MultiNomial Logit (MNL, independent and i.i.d. disturbances). We renounce from the estimation of a heteroscedastic-extreme-value (hev, independent and non-identical error terms) or probit model (correlated and non-identical error terms), as in (Prinzie & Van den Poel, in press) we assess no significant improvement in accuracy compared to the logit model, this on the same data as we use in this paper. The probability of choosing an alternative k among K_i choices for individual i can be written as in (3). The classifier predicts the class with the highest posterior probability for a given individual i.

$$P_i(k) = \frac{\exp(x'_{ik}\boldsymbol{\beta})}{\sum_{k \in k_l} \exp(x'_{ik}\boldsymbol{\beta})} \tag{3}$$

We compare the accuracy of above described MNL with that of a Random MultiNomial Logit (RMNL). Inspired by Random Forests principles and motivated by its high performance, we hypothesize that other methods than classification and regression trees might also benefit from injecting randomness. In this paper, we propose Random MultiNomial Logit, exploring the effects of random inputs (cf. bagging) and random feature selection on the accuracy of a traditional MNL. Because of reasons indicated above, the current paper is limited to the multinomial *logit* formulation. However, by no means does this imply that the proposed Random MultiNomial Logit framework could not be applied on a multinomial heteroscedastic-extreme-value or probit model. The RMNL is a bagged predictor h_c combining a collection of R MultiNomial Logits (i.e. forest of MultiNomial Logits). Firstly, just like Random Forests builds T classification or regression trees on different bootstrap samples S_t, in Random MultiNomial Logit each MultiNomial Logit r is grown on a different bootstrap sample S_r containing N_1 randomly drawn instances with replacement from the original training sample. Secondly, similar to Random Forests, this bagging is used in tandem with random feature selection. Each of the MNLs is estimated given a random selection of m out of M explanatory variables. To classify an observation, put the input vector down the R MultiNomial Logits in the forest. Each MNL votes for its predicted class. Finally the bagged predictor is obtained by majority vote, i.e. the instance is classified into the class having the most votes over all R MNLs in the forest.

Analogous to Breiman (2001) we utilize the out-of-bag data to estimate the test set error as well as the variable importances. By presenting each case left out of the construction of the *r*th multinomial regression to the *r*th MNL, we obtain a misclassification rate on oob data. Identical to Random Forests, Random MultiNomial Logit makes cross-validation or a test set redundant. To measure the importance of the *m*th variable in RMNL, we randomly permute this variable in the oob data and put the data down the corresponding MNL. Subtract the number of votes for the correct class in the variable-*m*-permuted data from the number of correct votes in the untouched data and average over all *R* MNLs in the forest. This is the *raw importance score* for variable *m* from which we infer the standardized importance score *z*. See Fig. 1. for an overview of the RMNL algorithm.

Besides improving the accuracy of the traditional MNL, the Random Forests framework addresses the

```
Algorithm Random MultiNomial Logit (RMNL)

[wPCC, PCC, AUC, wPCC_Oob, PCC_Oob, AUC_Oob, rawimp, standimp]=RMNL(N, R, m, cm)
Input:  N    data instances $\langle (x_1,y_1),...,(x_N,y_N) \rangle$ with labels y∈ {1, 2, …, K}
        R    number of bootstrap samples to draw from N and hence, number of MNLs to combine
        m    number of features to randomly select from total M features
        cm   combination method, e.g. MV or aMV
Output: wPCC, PCC, AUC                weighted PCC, PCC and AUC on data N
        wPCC_Oob, PCC_Oob, AUC_Oob    weighted PCC, PCC and AUC on out-of-bag data
        rawimp                        [m× 1] vector indicating raw importance score for each of the m
                                      selected features
        z                             [m× 1] vector indicating standardized importance score for each of
                                      the m selected features
Method:
for r=1 to R do
        $S_r = bootstrap(N)$
        $S_{r_m} = select(m, M, S_r)$   /* Select m out of M input features from S_r
        $Oob_{r_m} = N \notin S_{r_m}$        /* Oob data with m randomly selected features
        $[\beta_{r_m}] = MNL(S_{r_m})$
        $[wPCC_{Oob_{r_m}}, PCC_{Oob_{r_m}}, AUC_{Oob_{r_m}}] = evaluate(\beta_{r_m}, Oob_{r_m})$
        if r=R then
                $[wPCC, PCC, AUC] = evaluate(\langle \beta_{1_m}, ..., \beta_{R_m} \rangle, N, cm)$   */Apply R MNL models with m
                                                    randomly selected features, on data set N using cm
                $[wPCC_{Oob}, PCC_{Oob}, AUC_{Oob}] = evaluate(\langle \beta_{1_m}, ..., \beta_{R_m} \rangle, \langle Oob_{1_m}, ..., Oob_{R_m} \rangle, cm)$
                                                    /*Apply R MNL models with m randomly selected features on oob
                                                    data using combination method cm
            for v=1 to m
                    for (mr=1 to MR) : $(r \in R, v \in m_r)$    /* For all mr bootstrap samples where
                                                               feature v belongs to  selected m features
                    $Oob_{r_{m_v}} = permute(Oob_{mr}, v)$   /* Permute the values of feature v in oob data
                    $[wPCC_{Oob_{r_{m_v}}}, PCC_{Oob_{r_{m_v}}}, AUC_{Oob_{r_{m_v}}}] = evaluate(\beta_{mr_m}, Oob_{r_{m_v}})$
                        if mr=MR then
                                $rawimp[v] = \frac{\sum_{mr=1}^{MR} PCC_{Oob_{mr_r}} - PCC_{Oob_{mr_{r_{m_v}}}}}{MR}$
                                $z[v] = \frac{rawimp[v]}{se}$
                        endif
                    endfor
            endfor
        endif
endfor
```

Fig. 1. RMNL Algorithm.

feature-selection problem. Similar to fields such as genomic analysis (Xing, Jordan, & Karp, 2001), image retrieval (Swets & Weng, 1996), speech analysis (Chetouani, Faundez-Zanuy, Gas, & Zarader, 2005) and text categorization (Leopold & Kindermann, 2002), the CRM domain is burdened with huge feature spaces on the one hand and limited time and computational resources on the other hand. Therefore, feature selection is not only applied to increase accuracy, to limit practical costs of implementing the final model (Prinzie & Van den Poel, 2005) or to increase model comprehensibility, the CRM analyst is often forced to make a selection of input features in order to get *any* output from the learning algorithm. However as even human experts are ineffective at formulating hypotheses when data sets have large numbers of variables, feature-selection algorithms are imperative (Liu & Yu, 2005). Feature-selection algorithms reduce the dimensionality to enable multiple class data mining algorithms to work effectively on high-dimensional data. In short, we believe that the random feature selection might be extremely valuable to the CRM domain. In absence of feature-selection algorithms integrated into supervised *multiple* class learning problems in statistical software packages, this in glaring contrast to binary classification, random feature selection might be a welcome way out.

2.4. Predictive model evaluation: wPCC and AUC

The distinct learning algorithms (Random Forests, MNL and Random MultiNomial Logit) are evaluated on a separate test set, i.e. a data set of instances not used for model estimation. Although we could rely on the out-of-bag error estimates of Random Forests and Random MultiNomial Logit, a test set is needed for estimating the generalization error of the traditional MNL. Hence, to compare the three algorithms, each model's accuracy on the same test set is determined. We assess the models on their predictive performance as measured by the wPCC and the AUC.

In absence of a specific predictive objective, e.g. predict classes $k = 1$ and $k = 3$ well, we evaluate the algorithms in terms of their ability to correctly classify cases in all classes K. Given this objective and the small class imbalance of the dependent (i.e. differences in class prior probabilities biasing predictions towards the dominant class, cf. infra), it is inappropriate (Barandela, Sánchez, Garcia, & Rangel, 2003) to express the classification performance in terms of the average accuracy like the Percentage Correctly Classified (PCC), i.e. the total number of correctly classified relative to the total number of predicted instances (Morrison, 1969). The predictive evaluation of the models should therefore take the distribution of the multinomial dependent variable into consideration (Morrison, 1969). Firstly, we will weigh the class-specific PCCs with regard to the prior class distribution. Each class k ($k \in K$) of the dependent variable has a strict positive weight w_k (5), with f_k referring to the relative frequency of the class on the dependent variable. The class-specific weights sum to one as in (4). Given the weights, the weighted PCC is

$$\sum_{k=1}^{K} w_k = 1 \tag{4}$$

$$w_k = \frac{1 - f_k}{\sum_{k=1}^{K} 1 - f_k} \tag{5}$$

$$\text{wPCC} = \frac{\sum_{k=1}^{K} \text{wPCC}_k}{K} \tag{6}$$

with $\text{wPCC}_k = w_k * \text{PCC}_k$. The weighted PCC favors a model with a smaller PCC but with a greater number of correctly classified on smaller classes, to a model having a higher PCC due to predicting most cases to over represented classes. Similarly, the weights w_k and hence the wPCC could be tailored to maximize performance for a specific class. We penalize models predicting several alternatives (cf. ties on maximum probability) by equally dividing the 100% classified over all alternatives predicted. Secondly, we benchmark the model's performance to the proportional chance criterion (Cr_{pro}) rather than the maximum chance criterion (Cr_{max}) (Morrison, 1969):

$$\text{Cr}_{\text{pro}} = \sum_{1}^{K} f_k^2 \tag{7}$$

Besides this wPCC, the predictive performance of each classification model is assessed by the Area Under the receiver Operating Curve (AUC). The Receiver Operating Characteristics curve plots the hit percentage (events predicted to be events) on the vertical axis versus the percentage false alarms (non-events predicted to be events) on the horizontal axis for all possible cut-off values. The predictive accuracy of a model is expressed by the area under the ROC curve (AUC). The AUC statistic ranges from a lower limit of 0.5 for chance (null-model) performance to an upper limit of 1.0 for perfect performance (Green & Swets, 1966). Although the AUC measure is essentially designed to measure the degree to which a classifier can discriminate between two classes, we apply this binary measure to assess the multiclass classification predictive performance by adopting a one-versus-all (k-versus-$K\backslash k$) approach and averaging these K AUCs to an overall multiclass AUC.

2.5. Testing the statistical difference between AUCs on the test data

The primary objective of this study is to evaluate the predictive impact of adopting the Random Forest ideas to MultiNomial Logit. Therefore, we determine if the predictive performances of (a) a MNL with expert feature selection and (b) a Random MultiNomial Logit, are statistically different. We employ the non-parametric test by DeLong, DeLong, and Clarke-Pearson (1988) to determine whether the areas under the ROC curves (AUCs) are significantly different.

3. A cross-sell application

The methodological framework (Random Forests, MNL and RMNL) is applied to a CRM database of a major home-appliances retailer (a specialist multiple) containing scanner data for over one million customers making purchases from a very broad and deep product assortment ranging from small appliances like food processors to big appliances like dish washers. We analyze these scanner data employing the different learning algorithms to amass knowledge on customers' cross-buying patterns in order to support cross-sell actions. Cross-sell strategies are decision support systems aimed at the augmentation of the number of products/services customers use from the firm.

In this paper, the objective is to build the best possible 'Next-Product to Buy' model (NPTB model, (Knott, Hayes, & Neslin, 2002)) predicting in what product category the customer will acquire his next durable. We partition the home-appliance product space into nine product categories based on four underlying needs: cleaning, communication, cooking and entertainment (Barsalou, 1991; Corfman, 1991; Johnson, 1984). Hence, Y takes discrete values $\{1, 2, \ldots, 9\}$, $K = 9$ and has following prior distribution: $f_1 = 9.73\%$, $f_2 = 10.45$, $f_3 = 20.49$, $f_4 = 12.64$, $f_5 = 11.70$, $f_6 = 9.74$, $f_7 = 8.67$, $f_8 = 8.13$ and $f_9 = 8.45$. We select customers having at least two previous purchase events and maximum 16 purchase events thereby excluding outliers (median length of $2 + 3$ times $\sigma = 4.88$). Companies are deleted from the analysis. We randomly assigned 37,276 (N_1) customers to the estimation sample and 37,110 (N_2) customers to the test sample. For each of these customers we constructed a number of predictors X on: (1) monetary value, depth and width of purchase behavior – 5 features, (2) number of home-appliances acquired at the retailer – 13 features, (3) socio-demographical information – 5 features, (4) brand loyalty – 21 features, (5) price sensitivity – 25 features, (6) number of home-appliances returned – 3 features, (7) dominant mode of payment – 1 feature, (8) experience of a special life-event – 1 feature, (9) the order of acquisition of durables (ORDER) – 12 features, (10) the time to a first-acquisition or a repeated-acquisition event for a durable (DURATION) – 2 features. The first nine blocks of features are meant to build a general customer profile (NULL). The last two blocks aim to capture sequential patterns (ORDER or DURATION between acquisitions) in customers' purchase behavior. For a more in-depth discussion of these covariates, we refer the interested reader to Prinzie and Van den Poel (2005). In the current paper, we assess the ability of the three learning algorithms C, Random Forests C_1, MultiNomial Logit C_2 and Random MultiNomial Logit C_3, to make a hypothesis $h_c : X \rightarrow \{1, 2, \ldots, 9\}$ closely resembling the true function f; $y = f(x)$ based on N_1 (37,276) labeled training instances $\{(X_1, Y_1), (X_2, Y_2), \ldots, (X_{37,276}, Y_{37,276})\}$. The objective is to find a classifier h_c minimizing the misclassification error for N_2 (37,110) new unseen instances, i.e. test data. The latter boils down to obtaining the best possible NPTB model for all classes K. The predictive performance of the NPTB model could be improved by optimizing towards particular cells of the confusion matrix to support the retailer's specific product-category management (Prinzie & Van den Poel, 2005).

4. Results

4.1. Estimation of Random Forests for classification

To predict in what product category the customer will buy next, we estimated several Random Forests for Classification on the estimation data. Two series of analyses are conducted: (1) unbalanced and (2) balanced. Breiman (Breiman, 2001) offers a possibility to balance prediction error. In presence of class imbalances (e.g. even small ones like in our data), Random Forests, trying to minimize the overall error rate, will keep the error rate low for larger classes, while letting under represented classes have a higher error rate. Error balancing is achieved by setting different weights for the classes. Smaller classes are assigned higher weights in order to reduce their error rate. The weights are determined as in (8) and simulate a uniform class distribution.

$$w_{\mathrm{RF}_k} = \frac{\frac{100}{K}}{f_k} \tag{8}$$

Given the prior class distribution reported above, the Random Forests class weights are: $w_{\mathrm{RF1}} = 1.14$, $w_{\mathrm{RF2}} = 1.06$, $w_{\mathrm{RF3}} = 0.54$, $w_{\mathrm{RF4}} = 0.88$, $w_{\mathrm{RF5}} = 0.94$, $w_{\mathrm{RF6}} = 1.14$, $w_{\mathrm{RF7}} = 1.28$, $w_{\mathrm{RF8}} = 1.37$ and $w_{\mathrm{RF9}} = 1.31$. Obviously, in the unbalanced case the weights are the prior class distributions. For each of these unbalanced and balanced scenarios, we ran Random Forests with 500 classification trees (i.e. default setting). This number should be large enough to let the generalization error of the Random Forest converge. In fact, the only adjustable parameter to which Random Forests is somewhat sensitive is the number of random features m to split each node on. To this end, we evaluate the performance of Random Forests with 500 trees, either balanced or unbalanced, on a broad range of m values. By default, Random Forests determines m to be the square root of M; $m = 441^{\wedge 1/2}$. We engage in a grid search with main step size 1/3 of the default setting. Table 1 reports our results for the unbalanced as well as for the balanced Random Forests with $T = 500$ and m ranging from 7 to 399 variables randomly selected from the 441 X variables. Firstly, for both the unbalanced as balanced scenario, the results confirm the sensitivity of Random Forests to the m parameter (Fig. 2). It is clear that the default setting of m would not deliver optimal model performance. Secondly, the results illustrate that balancing prediction errors by attaching weights to the classes increases the performance of Random Forests on the weighted PCC but decreases the overall PCC. This finding does not come as a surprise, as Breiman (Breiman, 2001) also remarks that in order to obtain a more balanced

Table 1
Random Forests for Classification

m	Unbalanced			Balanced		
	wPCCe	PCCe	AUCe	wPCCe	PCCe	AUCe
7	17.62	23.19	0.5992	19.27	19.94	0.5999
14	18.07	23.30	0.6003	19.43	20.10	0.6010
21	18.49	23.53	0.6008	19.74	20.38	0.6007
28	18.56	23.63	0.6028	19.86	20.49	0.6026
35	18.83	23.82	0.6055	20.22	20.84	0.6042
42	18.99	24.01	0.6049	20.20	20.88	0.6057
49	19.02	24.04	0.6048	20.48	21.08	0.6054
56	18.9	23.92	0.6069	20.42	21.10	0.6047
63	19.07	24.08	0.6073	20.56	21.23	0.6071
70	19.09	24.03	0.6074	20.50	21.13	0.6053
77	19.27	24.21	0.6082	20.59	21.20	0.6072
84	19.23	24.13	0.6084	20.63	21.25	0.6061
91	19.33	24.26	0.6096	20.61	21.20	0.6067
98	19.32	24.27	0.6092	20.93	21.50	0.6080
105	19.32	24.31	0.6089	20.50	21.11	0.6070
112	19.16	24.18	0.6099	20.58	21.24	0.6090
119	19.22	24.20	0.6100	20.55	21.20	0.6079
126	19.17	24.15	0.6097	20.70	21.36	0.6079
147	19.32	24.32	0.6099	20.82	21.43	0.6078
168	19.41	24.36	0.6124	20.98	21.59	0.6089
189	19.63	24.54	0.6130	20.74	21.36	0.6101
210	19.51	24.45	0.6100	20.84	21.55	0.6095
231	19.55	24.48	0.6115	20.86	21.56	0.6090
252	**19.76**	**24.66**	0.6129	20.71	21.34	0.6095
273	19.48	24.40	0.6126	20.92	21.50	0.6094
294	19.62	24.59	0.6123	21.01	21.69	**0.6114**
315	19.61	24.55	0.6113	20.64	21.34	0.6081
336	19.38	24.34	0.6122	**21.04**	**21.67**	0.6097
357	19.53	24.44	0.6130	20.85	21.50	0.6085
378	19.54	24.48	0.6112	20.91	21.56	0.6104
399	19.72	24.60	**0.6142**	20.70	21.37	0.6100

Figures in bold indicate maximum value for a given column.

prediction error, the overall error rate (i.e. 1-PCC) will go up. Given the objective to build a NPTB model able to discriminate among *all* K classes, we select the best Random Forest model using the wPCC criterion. Based on the training data, we conclude that a balanced Random Forest with 500 trees and $m = 336$ delivers the best predictive performance: wPCCe = 21.04%, PCCe = 21.67% and AUCe = 0.6097. Note that the overall PCC should be compared to the proportional chance criterion $\mathrm{Cr}_{\mathrm{pro}} = 12.28\%$. With regard to the PCC, an unbalanced model with $m = 399$ is optimal. However, remember that overall accuracy (PCC) is of secondary importance; our first concern is to build a classifier able to predict all K classes equally well.

4.2. MultiNomial Logit

Besides Random Forests, we also applied the MultiNomial Logit algorithm to our home-appliance scanner data to predict in what category k with $K = \{1, 2, \ldots, 9\}$ the customer will buy next. In a first step, we estimated a MNL model with all M non-choice specific parameters (89 corresponding to the 441 Random Forest features). This turned out to be a fruitless attempt, as the model did not converge even after numerous attempts. It confirms our experience that in presence of large feature spaces, MNL forces the researcher to engage in feature selection.

In a second step, we preceded the final MNL estimation with feature selection based on our own human expert know-how of the CRM domain, from now on referred to as ‘MNL with expert feature selection’. In absence of an integrated feature-selection algorithm within the MNL algorithm, we adopted the *wrapper* approach (Kohavi & John, 1997), selecting features improving the MNL algorithm’s performance. We adopted a two-phase approach. *Firstly,* we selected the best features within each of the three different types of covariates (cf. supra Section 2, A Cross-sell application), i.e. within the NULL, ORDER and DURATION blocks. Secondly, we compared four NPTB models on their predictive accuracy wPCC, PCC and AUC on the estimation sample: (1) BEST5 model including only 5 selected NULL features, (2) BEST5 + ORDER including only the selected NULL features and selected ORDER feature, (3) BEST5 + DURATION including only the selected NULL features and selected DURATION feature, and (4) BEST5 + ORDER + DURATION including only the selected NULL features, selected ORDER feature and selected DURATION feature. The BEST5 model with Duration (wPCCe = 19.75%, PCCe = 22.00% with $\mathrm{Cr}_{\mathrm{pro}} = 12.28\%$ and AUCe = 0.5973) delivered the best possible NPTB model employing this two-phase (best 2 within blocks followed by best 5 across blocks) expert feature-selection procedure. Notwithstanding this high level of accuracy its wPCCe and AUCe are lower than that of the best Random Forests (wPCCe = 21.04%, PCCe = 21.67% and AUCe = 0.6097), the expert feature selection is very resource consuming (time and computer power) and moreover it cannot guarantee that the optimal subset of features is selected. Could a more accurate NPTB MNL model be obtained by adopting the Random Forests approach to the MNL framework? After all, as Liu and Yu (2005) state, use of randomness helps to escape local optima in the search space and optimality of the selected subset dependent on the available resources.

4.3. Random MultiNomial Logit (RMNL)

We adopt the Random Forest approach to the MNL framework to explore if injecting randomness could improve the accuracy of a MNL with expert feature selection. *Firstly,* just like Random Forests builds T classification trees on different bootstrap samples S_t, in Random MultiNomial Logit each multinomial logit r is grown on a different bootstrap sample S_r containing N_1 randomly drawn instances with replacement from the original training sample. *Secondly,* similar to Random Forests, this bagging is used in tandem with random feature selection. Each of the MNLs is estimated given a random selection of m out of M explanatory variables. *Thirdly,* analogous to Random Forests, the R MNLs are combined with a Majority Voting (MV) combination scheme. Hence, each MNL casts

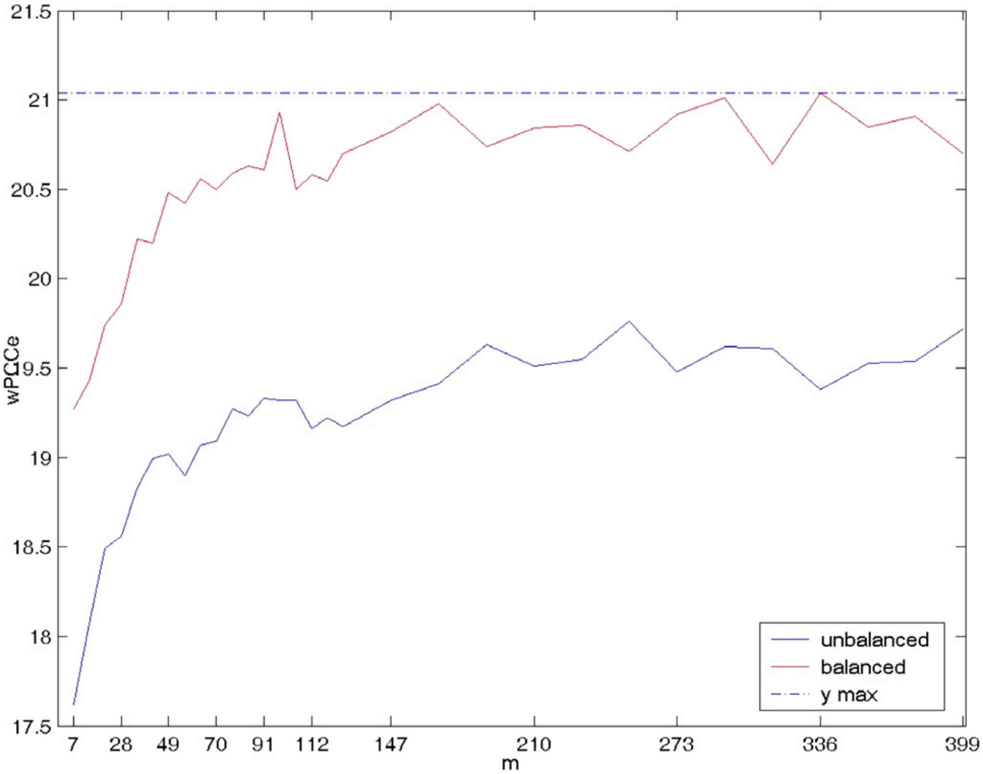

Fig. 2. Sensitivity of Random Forests to m parameter. wPCCe refers to weighted PCC on estimation sample.

a vote for its predicted class, inferred from selecting the maximum posterior probability across the K classes. The instance is assigned to the class that collects the most votes. Analogous to Random Forests, the AUC statistic under the Majority Voting combination scheme is calculated on the percentage votes per class. Additionally, we assess the predictive performance of the RMNL by combining the MNLs employing the adjusted Majority Vote (aMV) as, unlike Random Forests, the MNLs output *continuous* posterior probabilities. The adjusted Majority Vote is a refinement of the MV algorithm suited to this continuous case where class probabilities are predicted by the base-level classifiers (e.g. MNL classifiers). Let $p_r(x)$ be the class probability distribution predicted by MNL r on example X. The probability distribution vectors returned by the R MNLs are summed to obtain the class probability distribution of the meta-level voting classifier RMNL (9). The AUC statistic is applied on the probabilities from (9)

$$P_{R_{\mathrm{RMNL}}} = \frac{1}{R}\sum_{r=1}^{R} p_r(x) \qquad (9)$$

4.3.1. RMNL combining 100 MNLs

Initially, we estimated a series of RMNLs combining 100 ($R = 100$) MNLs with m randomly selected features. Just like Random Forests, we take the square root of M; $m = 89^{\wedge 1/2}$, as default parameter setting and, subsequently, engage in a grid search with main step size 1/3 of the default setting. This way m spans a range from 3 to 84. Unfortunately, MNL models with more than 48 variables failed to estimate for the same reason (multicollinearity) that we were unable to estimate full MNL model (cf. Section 4. Results, 4.2. MultiNomial Logit). Table 2, RMNL ($R = 100$) gives an overview of the results. The first column indicates the combination method, i.e. MV or aMV. The results demonstrate that the aMV combination method surpasses the MV method in AUC predictive performance. Amongst the RMNLs with $R = 100$, the highest predictive accuracy is observed for $m = 48$ (wPCCe = 21.25, PCCe = 26.87, AUCe = 0.6491). This performance is considerably higher than that of the best MNL with expert feature selection (wPCCe +1.5 percentage points (from now on abbreviated to pctp), PCCe +4.87 pctp, AUCe + 3.53 pctp). Compared to the best Random Forest, the wPCCe is similar ($\mathrm{wPCC}_{\mathrm{RF}} = 21.04$), but the PCC and AUC are impressively higher (PCCe +5.2 pctp; AUCe = +3.94 pctp).

4.3.2. RMNL combining MNLs with 10% highest wPCC (10_Best)

Although all 100 MNLs have a performance better than $\mathrm{Cr}_{\mathrm{pro}} = 12.28\%$, combining only very accurate classifiers might improve the performance of the RMNL even more (Dietterich, 1997). Therefore, in a second step, we combined only the MNL models for a given m with the 10% highest wPCCe, i.e. 10_Best. We refrain from evaluating the potential of combining the 10% with highest PCCe or AUCe, as the wPCCe is our main performance criterion. The same range of m values as for RMNL with $R = 100$

Table 2
RMNL

Combi	m	RMNL ($R=100$)			RMNL ($R=10$)			Single_Best		
		wPCCe	PCCe	AUCe	wPCCe	PCCe	AUCe	wPCCe	PCCe	AUCe
MV	3	10.03	20.55	0.5972	18.57	23.30	0.5582	19.43	22.35	0.5884
aMV	3	11.53	21.41	0.6163	19.30	23.93	0.6232			
MV	6	15.00	23.10	0.5922	18.66	24.20	0.5717	18.93	23.67	0.6151
aMV	6	13.41	22.45	0.6225	18.76	24.42	0.6262			
MV	9	16.68	23.76	0.5894	19.42	24.54	0.5735	19.53	24.09	0.6237
aMV	9	15.60	23.62	0.6270	19.69	24.98	0.6315			
MV	12	17.70	24.17	0.5873	19.46	24.74	0.5772	20.36	25.42	0.6286
aMV	12	17.52	24.53	0.6300	19.94	25.25	0.6342			
MV	15	18.33	24.71	0.5891	20.39	26.06	0.6036	20.57	26.12	0.6356
aMV	15	18.36	24.98	0.6328	20.56	26.33	0.6403			
MV	18	18.73	25.10	0.5943	20.38	26.24	0.6047	20.32	25.56	0.6306
aMV	18	18.62	25.15	0.6359	20.42	26.35	0.6419			
MV	21	19.29	25.46	0.5964	20.95	26.59	0.6085	21.46	26.2	0.6398
aMV	21	19.33	25.56	0.6390	21.09	26.78	0.6436			
MV	24	19.51	25.64	0.5971	20.99	26.63	0.6114	20.86	26.17	0.6316
aMV	24	19.48	25.64	0.6404	21.12	26.77	0.6425			
MV	27	19.64	25.72	0.5995	21.11	26.59	0.6049	21.03	26.12	0.6340
aMV	27	19.74	25.90	0.6423	21.14	26.63	0.6435			
MV	30	20.13	26.09	0.6047	21.45	27.01	0.6058	21.57	26.82	0.6400
aMV	30	20.17	26.18	0.6443	21.45	27.04	0.6461			
MV	33	20.40	26.32	0.6046	21.54	27.04	0.6099	21.54	26.87	0.6404
aMV	33	20.37	26.35	0.6458	21.59	27.13	0.6468			
MV	36	20.72	26.51	0.6061	21.51	26.98	0.6076	21.49	26.86	0.6410
aMV	36	20.66	26.54	0.6467	21.55	27.05	0.6469			
MV	39	20.70	26.50	0.6092	21.85	27.29	0.6123	21.78	26.89	0.6441
aMV	39	20.73	26.58	0.6472	21.75	27.23	0.6480			
MV	42	20.94	26.65	0.6077	21.83	27.30	0.6070	21.75	26.89	0.6436
aMV	42	20.91	26.69	0.6480	21.82	27.31	0.6477			
MV	45	21.02	26.64	0.6101	21.88	27.24	0.6132	21.78	26.99	0.6436
aMV	45	21.03	26.73	0.6485	21.87	27.27	0.6478			
MV	48	21.22	26.77	0.6124	21.98	27.297	0.6108	**21.9**	**27.02**	**0.6438**
aMV	**48**	**21.25**	**26.87**	**0.6491**	**22.01**	**27.33**	**0.6489**			

is considered. Table 2, column RMNL ($R=10$) reports the results. Analogous to RMNL with $R=100$, the aMV combination method outperforms the MV scheme in terms of AUC performance. Furthermore, the highest predictive performance is also attained for $m=48$ and this performance is, as far as it concerns the wPCCe and the PCCe, slightly higher than the performance of RMNL $R=100$ (wPCCe +0.76 pctp, PCCe +0.46), however the AUC statistic is slightly worse (AUCe –0.016 pctp). In sum, combining only a selection of more accurate MNL models improves upon the performance of a RMNL combining all 100 MNLs.

4.3.3. *Select Best (Single_Best)*

According to Džeroski and Ženko (2004) the performance of an ensemble like RMNL should be compared to that of the best individual base-level classifier. For each value of m, we selected the MNL out of the 100 MNLs of the RMNL, scoring best on the wPCCe. Table 2, last column (Single_Best) presents the results. In general, we find that the Single_Best procedure outperforms the RMNL with $R=100$ on both wPCCe and PCCe criterion, while the AUCe statistics are just slightly lower. Compared to 10_Best, the performance of the Single_Best procedure is slightly less or occasionally identical. For $m=48$, the 10_Best RMNL outperforms the Single_Best on all three criteria, although the differences are rather small (wPCCe + 0.11 pctp, PCCe +0.31 pctp, AUCe +0.51 pctp).

In conclusion, the estimation results demonstrate the superiority of our new Random MNL framework over the MNL with expert feature selection. This is promising because on the one hand RMNL improves the performance and on the other hand, it accommodates for the feature-selection problem of MNL. However, can we replicate these findings when applying the algorithms to new unseen data (i.e. test data)?

4.4. *Predictive model evaluation on test data: Random Forests, MNL and RMNL*

We assess the robustness of the results on the estimation sample by applying and evaluating the best Random Forest ($m=336$, balanced), the best MNL with expert feature selection (best5+Duration), the best RMNL ($m=48$, 10_Best, aMV) and the single best MNL from RMNL ($m=48$, Single_Best) on a separate test sample, i.e. a dataset of instances not used for estimation ($N_2=37{,}110$). Table 3 and Fig. 3 present the results.

Table 3
Predictive performance on test data

	wPCCt	PCCt	AUCt
RF	20.66	21.39	0.6090
MNL	19.75	21.84	0.5926
RMNL	21.06	**26.41**	**0.6322**
SB	**21.27**	26.21	0.6273

Table 4
Predictive performance on oob data

	wPCC_{oob}	PCC_{oob}	AUC_{oob}
RF	17.16	21.67	
RMNL	21.43	26.7	0.5998
SB	21.16	25.89	0.6331

Table 5
Statistical significance of AUC differences

	RMNL RF	RMNL SB	RMNL MNL
1	81.91	32.10	215.15
2	21.24	17.11	73.01
3	45.99	10.64	58.84
4	11.07	0.06	100.01
		0.80	
5	12.01	21.07	128.63
6	19.50	3.35	24.12
		6.71E−02	
7	16.48	37.37	91.90
8	30.70	28.35	117.49
9	21.01	6.57	81.45

First row indicates $\%^2$. All statistics are significantly different at $a = 0.05$, except those with a row under x^2.

The arrows in Fig. 3 clearly illustrate that the observed higher performance of the best RMNL (RMNL in Fig. 3) as compared to that of the best MNL with expert feature selection on the estimation sample is confirmed on the test sample. Table 3 reveals that wPCCt increases by 1.31 pctp, PCCt by 4.57 pctp and AUCt by 3.96. The best RMNL (10_best, aMV, $m = 48$) also outperforms the single_best in PCCt and AUCt, but not in wPCC. Table 4 reports the out-of-bag performance for the best Random Forest (RF), the best RMNL (RMNL) and the single_best (SB) models. Furthermore, we determine if the AUCs of the MNL with expert feature selection and the best RMNL are statistically different. Per product category, we employ the non-parametric test by DeLong et al. (1988) to determine whether the areas under the ROC curves (AUCs) within a product category are significantly different. Table 5 shows that all AUCs on the test set are statistically significant different except for (a) the AUC difference between RMNL and Single_Best for $k = 4$ and (b) the AUC difference between RMNL and Single_Best for $k = 6$.

To recap, the estimation and test results clearly show evidence of the potential of RMNL to improve the predictive performance of a MNL with expert feature selection. Adopting the Random Forest principles to the Multi-Nomial Logit framework is a valuable research endeavor improving performance as the random feature selection avoids local optima in the feature search space and optimality of the selected subset dependent on the available resources. Moreover, RMNL's time efficiency (cf. 3200 MNLs estimated; $|m| * R$) is still comparable to the MNL with expert feature selection (2613 MNLs estimated).

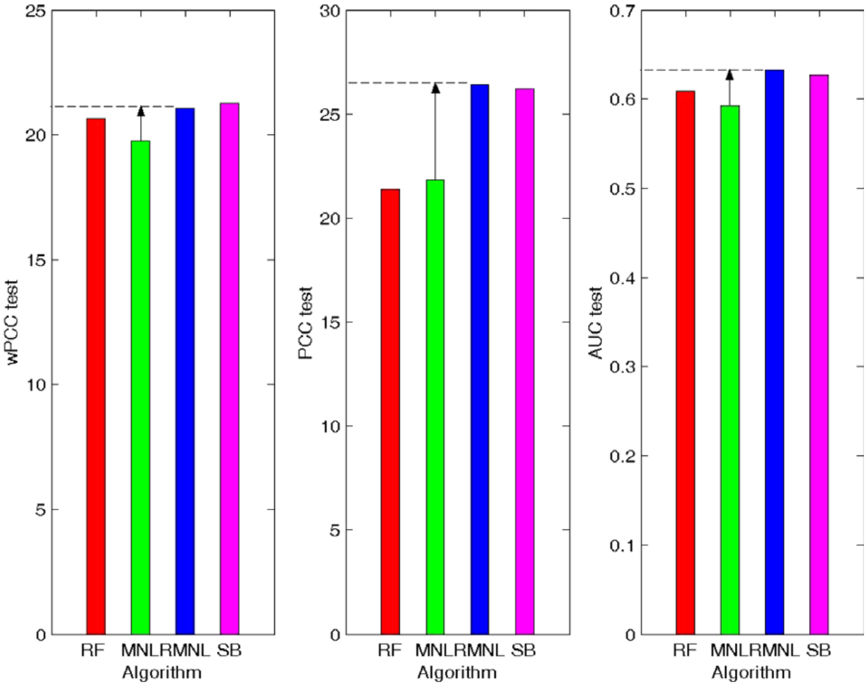

Fig. 3. Predictive performance on test data.

Table 6
Top 20 of features for RMNL (10_Best) compared with rank of RF

Varname	RankRF	RankRMNL	*z*	Block	Description
productnbr_pc		1	29.37	1	monetary, depth and width
diffproduct_pc		2	24.91	1	monetary, depth and width
gender	14	3	19.70	3	socio-demo
ORDER Markov 2nd order		4	16.01	9	order
DURATION (surv)	4	5	9.48	10	duration
ORDER Markov 2nd order		6	9.21	9	order
ORDER dummies	51	7	7.69	9	order
ORDER Markov for Discrimination	7	8	4.86	9	order
language	35	9	4.84	3	socio-demo
nbrdiffbrand	16	10	4.74	4	brand loyalty
loyal_PANASONIC	63	11	4.51	4	brand loyalty
ORDER Markov 2nd order		12	4.44	9	order
rnbrreturns	56	13	4.41	6	returns
nbrabovep90	37	14	4.32	5	price sensitivity
maxdiffprod	36	15	3.96	2	number acquired
nbrbelowql	31	16	3.87	5	price sensitivity
maxprod	46	17	3.74	2	number acquired
maxamount	10	18	3.38	1	monetary, depth and width
DURATION (survdiff)	3	19	3.36	10	duration
ORDER Markov 2nd order		20	3.34	9	order

4.5. Feature importances in NPTB model

From a CRM cross-sell action perspective, it is vital to gain insight in which features drive cross-buying propensities. Therefore, we need to assess the importance of the features in the NPTB model. Analogous to Random Forests we utilize the out-of-bag data. To measure the importance of the mth variable in RMNL, we randomly permute this variable in the oob data and put the data down the corresponding MNL. Subtract the number of votes for the correct class in the feature-m-permuted data from the number of correct votes in the untouched oob data and average over all R MNLs in the forest incorporating feature m. This is the *raw importance score* for feature m from which we infer the standardized importance score z. Table 6 lists the top-20 most important features for RMNL with a reference to the type of covariates they belong to (cf. Section 4 Results, 4.2. MultiNomial Logit). The results indicate a serious loss in predictive accuracy when dropping features on the number of (different) appliances acquired per product category (block 1), the gender of the customer (block 3), the order of acquisition of home appliances (block 9) and the time until a first acquisition within a product category or between repeated acquisition in a product category (block 10).

Table 6 also reports the corresponding ranks of the top-20 features of RMNL in the best Random Forests (a missing indicates that the feature is not selected by Random Forests). In total, there are 59-shared features between the RMNL 10_Best and Random Forests. The Spearman rank-order correlation between the ranks of the shared features is 0.1511 and not significant. This is not surprising given the substantial lower predictive performance for Random Forests as compared to RMNL on wPCCt, PCCt and AUCt.

5. Conclusion

MultiNomial Logit and Random Forests are two algorithms suited for multiclass classification. Given Random Forests' robustness and competence for analyzing large feature spaces *and* MNLs weakness in the latter, this paper explored the potential of extending the Random Forests principles to MNL. Our new innovative Random MultiNomial Logit (RMNL) method is demonstrated on a retail cross-sell application. The empirical results strongly support the proposition that MNL with expert feature selection is suboptimal to RMNL in terms of predictive accuracy (wPCCt increased by 1.31 pctp, PCCt by 4.57 pctp and AUCt by 3.96 pctp!) as well as in terms of time efficiency. Considering (1) the curse of high-dimensional feature spaces and limited resources in many fields amongst which the CRM field, (2) the absence of an integrated feature-selection algorithm in the MNL algorithm and (3) the incompetence of the MNL algorithm to handle huge feature spaces (cf. no results for $m > 48$ in our application), the RMNL seems a very fruitful attempt to assess the feature-selection problem while at the same time improving model accuracy.

This paper applied the Random Forests approach to the MultiNomial Logit method. Future work could adopt the Random Forests approach on a multinomial heteroscedastic-extreme-value or probit model. Moreover, the Random Forests principles might be extended outside the random-utility models to non-parametric multiclass supervised learning algorithms. Another interesting direction for further research constitutes a profound analysis of the conditional effects of injecting randomness on multiclass classification performance. For instance, what is the potential of the Genetic Algorithm, another method besides Random Forests starting from the assumption of the positive effect of randomness, to enhance multiclass supervised learning algorithms like SVM (Peng et al., 2003) and MNL?

Acknowledgement

The authors would like to thank (1) the anonymous home-appliances retailer for providing the data, (2) the Flemish Research Fund (FWO Vlaanderen) and (3) BOF for funding computing facilities (011B5901).

References

Agrawal, D., & Schorling, C. (1996). Market share forecasting: an empirical comparison of artificial neural networks and multinomial logit model. *Journal of Retailing, 72*(4), 383–407.

Anas, A. (1983). Discrete choice theory, information theory and the multinomial logit and gravity models. *Transportation Research Part B – Methodological, 17*(1), 13–23.

Baltas, G., & Doyle, P. (2001). Random utility models in marketing: a survey. *Journal of Business Research, 51*(2), 115–125.

Barandela, R., Sánchez, J. S., Garcia, V., & Rangel, E. (2003). Strategies for learning in class imbalance problems. *Pattern Recognition, 36*(3), 849–851.

Barsalou, L. W. (1991). Deriving categories to achieve goals. In G. H. Bower (Ed.), *The psychology of learning and motivation* (pp. 1–64). New York: Academic Press.

Ben-Akiva, M., & Lerman, S. R. (1985). *Discrete choice analysis: Theory and application to travel demand*. Cambridge: The MIT Press.

Breiman, L. (1984). Classification and regression trees. In L. Breiman, J. H. Friedman, R. A. Olshen, & C. J. Stone (Eds.), *The Wadsworth statistics/probability series* (pp. 358). Belmont, CA: Wadsworth International Group.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Buchtala, O., Klimek, M., & Sick, B. (2005). Evolutionary optimization of radial basis function classifiers for data mining applications. *IEEE Transactions on Systems Man and Cybernetics Part B – Cybernetics, 35*(5), 928–947.

Buckinx, W., & Van den Poel, D. (2005). Customer base analysis: partial defection of behaviourally loyal clients in a non-contractual FMCG retail setting. *European Journal of Operational Research, 164*(7), 252–268.

Chetouani, M., Faundez-Zanuy, M., Gas, B., & Zarader, J. L. (2005). Non-linear speech feature extraction for phoneme classification. Nonlinear speech modeling and applications. *Lecture Notes in Artificial Intelligence, 3445*, 344–350.

Corfman, K. P. (1991). Comparability and comparison levels used in choices among consumer products. *Journal of Marketing Research, 28*(3), 368–374.

DeLong, E. R., DeLong, D. M., & Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics, 44*, 837–845.

Devillers, L., Vidrascu, L., & Lamel, L. (2005). Challenges in real-life emotion annotation and machine learning based detection. *Neural Networks, 18*(4), 407–422.

Dietterich, T. G. (1997). Machine-learning research – four current directions. *AI Magazine, 18*(4), 97–136.

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. *Machine Learning, 40*(2), 1–19.

Džeroski, S., & Ženko, B. (2004). Is combining classifiers with stacking better than selecting the best one? *Machine Learning, 54*(3), 255–273.

Freund, Y., & Schapire, R. (1996). Experiments with a new boosting algorithm. In *Machine learning: Proceedings of the thirteenth international conference* (pp. 148–156).

Goh, K. S., Chang, E. Y., & Li, B. (2005). Using one-class and two-class SVMs for multiclass image annotation. *IEEE Transactions on Knowledge and Data Engineering, 17*(10), 1333–1346.

Green, D., & Swets, J. A. (1966). *Signal detection theory and psychophysics*. New York: John Wiley & Sons.

Huang, X., Pan, W., Grindle, S., Han, X., Chen, Y., Park, S. J., et al. (2005). A comparative study of discriminating human heart failure etiology using gene expression profiles. *Bioinformatics, 6*(205).

Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(1).

Johnson, M. D. (1984). Consumer choice strategies for comparing noncomparable alternatives. *Journal of Consumer Research, 11*(3), 741–753.

Knott, A., Hayes, A., & Neslin, S. A. (2002). Next-product-to-buy models for cross-selling applications. *Journal of Interactive Marketing, 16*(3), 59–75.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence, 97*(1–2), 273–324.

Leopold, E., & Kindermann, J. (2002). Text categorization with support vector machines. how to represent texts in input space? *Machine Learning, 46*(1–3), 423–444.

Liu, H., & Yu, L. (2005). Toward Integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering, 17*(4), 491–502.

Lunetta, K. L., Hayward, L. B., Segal, J., & Eerdewegh, P. V. (2004). Screening large-scale association study data: exploiting interactions using random forests. *BMC Genetics, 5*(32).

Luo, T., Kramer, K., Goldgof, D. B., Hall, L. O., Samson, S., Remsen, A., et al. (2004). Recognizing plankton images from the shadow image particle profiling evaluation recorder. *IEEE Transactions on Systems, Man and Cybernetics – Part B Cybernetics, 34*(4), 1753–1762.

Morrison, D. G. (1969). On the interpretation of discriminant analysis. *Journal of Marketing Research, 6*, 156–163.

Novovicova, J., & Malik, A. (2003). Application of multinomial mixture model to text classification. Pattern recognition and image analysis, Proceedings. *Lecture Notes in Computer Science, 2652*, 646–653.

Peng, S. H., Xu, Q. H., Ling, X. B., Peng, X. N., Du, W., Chen, L. B., et al. (2003). *FEBS Letters, 555*(2), 358–362.

Prinzie, A., & Van den Poel, D. (2005). Constrained optimization of data-mining problems to improve model performance: a direct-marketing application. *Expert Systems with Applications, 29*(3), 630–640.

Prinzie, A., & Van den Poel, D. (in press). Predicting home-appliance acquisition sequences: Markov/Markov for discrimination and survival analysis for modeling sequential information in NPTB models. *Decision Support Systems*, doi:10.1016/j.dss.2007.02.008.

Schwender, H., Zucknick, M., Ickstadt, K., & Bolt, H. M. (2004). A pilot study on the application of statistical classification procedures to molecular epidemiological data. *Toxicology Letters, 151*(1), 291–299.

Swets, D. L., & Weng, J. J. (1996). Using discriminatn eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Learning, 18*(8), 831–836.

Wu, B. L., Abbott, T., Fishman, D., McMurray, W., Mor, G., Stone, K., et al. (2003). Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. *Bioinformatics, 19*(13), 1636–1643.

Xing, E., Jordan, M., & Karp, R. (2001). Feature selection for high-dimensional genomic microarray data. In *Proceedings of the 15th international conference on machine learning* (pp. 601–608).