Enhancing Classification of Ecological Momentary Assessment Data using Bagging and Boosting

Gerasimos Spanakis*, Gerhard Weiss* and Anne Roefs†
*Department of Data Science and Knowledge Engineering
Maastricht University, Maastricht, Netherlands, 6200MD
Email: {jerry.spanakis, gerhard.weiss}@maastrichtuniversity.nl
†Faculty of Psychology and Neuroscience
Maastricht University, Maastricht, Netherlands, 6200MD
Email: a.roefs@maastrichtuniversity.nl

***Abstract*—Ecological Momentary Assessment (EMA) techniques gain more ground in studies and data collection among different disciplines. Decision tree algorithms and their ensemble variants are widely used for classifying this type of data, since they are easy to use and provide satisfactory results. However, most of these algorithms do not take into account the multiple levels (per-subject, per-day, etc.) in which EMA data are organized. In this paper we explore how the EMA data organization can be taken into account when dealing with decision trees and specifically how a combination of bagging and boosting can be utilized in a classification task. A new algorithm called BBT (standing for Bagged Boosted Trees) is proposed which is enhanced by an over/under sampling method leading to better estimates of the conditional class probability function. BBT's necessity and effects are demonstrated using both simulated datasets and real-world EMA data collected using a mobile application following the eating behavior of 100 people. Experimental analysis shows that BBT leads to clear improvements with respect to prediction error reduction and conditional class probability estimation.**

***Index Terms*—Ecological Momentary Assessment, Classification Trees, Bagging, Boosting**

I. Introduction

Decision trees (classification or regression) have been widely used for exploring and predicting different categories of datasets. Tree ensembles (based on bagging or boosting) have been utilized in order to overcome the two main disadvantages of single decision trees, namely their moderate accuracy and the difficulty to scale for big datasets.

This work will focus on classification trees and how their ensembles can be utilized in order to set up a prediction environment using Ecological Momentary Assessment (EMA) data from a real-world study. EMA [1] refers to a collection of methods used in many different disciplines by which a research subject (i.e. human, plant, sample depending on the study) repeatedly reports on specific variables measured close in time to experience and in the subject's natural environment (e.g. experiencing food craving is measured again and again on the same subject). EMA aims to minimize recall bias, maximize ecological validity and allow microscopic analysis of influence behavior in real-world contexts. EMA data have a different structure than normal data and account for several dependencies between them, since e.g. many samples belong to the same subject so they are expected to be correlated. However, most decision trees that deal with EMA data do not take these specificities into account.

This paper combines boosted trees with a form of bagging that creates bootstrap samples which take into account the correlation in within-subject data. Moreover, the boosting algorithm is modified so as to provide an optimal number of trees taking into account the bagging function. Also, problems related to the calculation of the conditional class probability function are tackled: classifying at the 1/2 quantile is not always optimal for classification problems and also sometimes a trained classifier has to be mapped to a base rate for the two classes different than the one in the training data.

The rest of the paper is organized as follows: Section II presents related work and Section III introduces the proposed Bagged Boosted Trees method to classify EMA data. Experimental results are presented in Section IV and finally Section V concludes the paper.

II. Related Work

EMA data particular type of structure is a major focus in order to analyse this kind of data. Classical statistics often assume that observations are drawn from the same general population and are independent and identically distributed [2]. This is not the standard for EMA data and most decision tree algorithms do not take that into account when treating these data [3], despite many improvements (like combinations of multiple trees) have already been proposed [4].

Many efforts have been made to adjust and further improve decision trees. Convex-optimization methods have been used in cases like neural trees where the objective function computes the errors before non-linear activation functions instead of after them as is usually the case. In this way, the local optimum avoidance is ensured and global optimum is obtained by solving a square system of linear equations [5].

Bagging [6] involves having each tree in the ensemble vote with equal weight. In order to promote model variance, bagging trains each model in the ensemble using a randomly drawn subset of the training set. As an example, the random forest algorithm [7] combines random decision trees with bagging to achieve very high classification accuracy. Boosting

[8] involves incrementally building an ensemble by training each new model instance to emphasize the training instances that previous models mis-classified. Major differences between bagging and boosting are that (a) boosting changes the distribution of training data based on the performance of classifiers created up to that point (bagging acts stochastically) and (b) bagging uses equal weight voting while boosting uses a function of the performance of a classifier as a weight for voting.

Boosted trees have been shown to provide best results in supervised problems [9]. They have been modified in order to provide accurate results for class probability estimation [10] and have also been applied to many domains [11]. Bagged trees are used extensively in statistics and provide a plurality voting classification criterion [12].

There are limited studies on combining bagging and boosting. Authors in [13] apply this combination but only to regression trees due to the nature of the algorithm (use of alternate bias-variance reduction approaches). For classification problems, authors in [14] propose a combination of both approaches using a voting criterion of bagging and boosting ensembles using $C4.5$ algorithm [15] as a base classifier. Their results show improvement of 15% on average. Work in [16] has also utilized a combination of bagging and boosting with imbalanced data handling and variable preprocessing. Their results are promising but not satisfactory when it comes to aggressive down-sampling and concerns are raised about the number of bagging iterations. However, none of these approaches have been applied to longitudinal data or take into account the EMA structure.

On the other hand, there have been some efforts to apply decision tree based methods to EMA data [17] in order to overcome dependencies between data but results were not promising. Other approaches that have been developed include the introduction of random factors (to account for the variability within the data) but on the one hand they are only applied to regression trees ([18], [19], [20]), and on the other hand they do not use bagging or boosting for improving performance. Work in current paper aims at bridging this gap by combining bagging and boosting with the longitudinal data structure.

III. The proposed algorithm

A. Growing Bagged Boosted Trees (BBT)

Let the training data be $x_1, ..., x_n$ and $y_1, ..., y_n$ where each x_i is a d-dimensional vector and $y_i \in \{-1, 1\}$ is the associated observed class label. To justify generalization, it is usually assumed that training data as well as any test data are *iid* samples from some population of (x, y) pairs. Our goal is to predict $\hat{y}_i$ given x_i where in the case of classification problems we apply logistic regression rules, i.e., $1/(1+exp(-\hat{y}_i)$ is the predicted probability of the instance belonging to the positive class. Learning is achieved through a model (say Θ) and that has a clear objective to minimize:

$$Obj(\Theta) = L(\Theta) + \Omega(\Theta) \tag{1}$$

where L measures the training loss, i.e., how well model fits on training data and Ω is the regularization which measures the complexity of the model. Since we are dealing with a classification problem, logistic regression takes the following form:

$$\begin{aligned} Obj(\Theta) = \textstyle\sum_{i=1}^{n} [y_i ln(1 + e^{-w^T x_i}) + \\ +(1 - y_i) ln(1 + e^{w^T x_i})] + \lambda ||w||^2 \end{aligned} \tag{2}$$

where w are the regression weights and λ is the regularization parameter.

Large classification trees have high variance and low bias [21] and are therefore well suited to enhancement by ensemble methods like bagging or boosting. The proposed method, namely BBT (Bagged Boosted Trees) is described below.

The first step to fit a BBT is to select the loss function, which in the case of a classification problem is defined by Equation 2. Parameters to be selected include the number of trees to be grown in sequence, the shrinkage (or learning) rate, the size of individual trees and the fraction of the training data sampled. There are several guides on how to select these parameters [22], since they need to be selected in advance by the user. Shrinking (or learning rates) rates of 0.1 to 0.001 are values that are normally used and generally smaller values yield lower prediction error (PE) but require proportionally more computation [23]. The fraction of training data sampled is typically set in the range $(0.4-0.6)$ and is rarely varied [24]. In our case, after the parameter selection, we grow the Boosted Bagged Trees (BBT) (say using M trees) on the training data using the following process and by growing single Boosted Trees (BT):

1) Divide the data into B (typically $5 - 10$) subsets and construct B training data sets each of which omits one of the B subsets (the 'out-of-bag' data). See Section III-B for the method to select the subsets
2) Grow B BT; one for each of the B training sets.
3) Calculate the PE for each BT for tree sizes 1 to M from the corresponding out-of-bag data and pool across the B boosted trees. Predictions for new data are computed by first predicting each of the component trees and then aggregate the predictions (e.g., by averaging), like in bagging.
4) The minimum PE estimates the optimum number of trees $m*$ for the BT. The estimated PE of the single BT obtained by cross-validation can thus also be used to estimate PE for the BBT. BBT thus require minimal additional computation beyond estimation of $m*$.
5) Reduce the number of trees for each BT to $m*$.

It has been observed repeatedly that the performance of the procedure, with respect to prediction error, is quite insensitive to the choice of M and tends to result in small error rates (relative to competing methods) across a wide range of applications, especially in high dimensions [26]. In many real-world examples, large values of M work very well [27].

The (general) algorithm for boosting (used to grow the BT and based on AdaBoost [8]) is as follows. First let $F_0(x_i) = 0$

for all x_i and initialize weights $w_i = 1/d$ for $i = 1, ..., d$. Then repeat the following for $m = 1, ..., M$ for each one of the B BT:

- Fit the decision tree g_m to the training data sample using weights w_i where g_m maps each x_i to -1 or 1.
- Compute:
  - the weighted error rate $\epsilon_m = \sum_{i=1}^{n} w_i I\{y_i \neq g_m(x_i)\}$
  - half its log-odds and derive $\alpha_m = \frac{1}{2} \log \frac{1-\epsilon_m}{\epsilon_m}$
- Let $F_m = F_{m-1} + \alpha_m g_m$.
- Replace the weights w_i with $w_i = w_i e^{-\alpha_m g_m(x_i) y_i}$ and then renormalize by replacing each wi by $wi/(\sum w_i)$.

Given the fact that we are solving a classification problem, we are exploring a way to connect the score function of BBT (i.e., $F_m(x)$) and additive logistic regression. Following [28]'s work, we use as an estimate $p_m(x)$ of the Conditional Class Probability Function (CCPF) $p(x)$ that can be obtained from F_m through a logistic link function:

$$p_m(x) = p_m(y = 1|x) = \frac{1}{1 + \exp(-2F_m(x))} \tag{3}$$

Using this link function, it has been shown [29] that the exponential loss of boosting process can be mapped to a score function (or loss function) on the probabilities similar to a maximum likelihood criterion, which can be used instead of Equation 2 to estimate the fitness of CCPF. This means that each iteration of boosting uses the current probability estimates to minimize this criterion (see the actual formula in Section IV) and this process is followed on every step.

Two issues arise with the introduction of BBT. The first one is how to part the dataset in order to construct the B BT and the second is how to handle the CCPF in order to correctly estimate the probability of a sample to belong to the positive class. These issues are handled in the following subsections and highlight the advantages of BBT for EMA data compared to other algorithms.

B. Selecting subsets of EMA data

Lately, in many medical/heath related applications, longitudinal datasets are available. Bagging and boosting algorithms do not consider any dependency structure in the data, which can clearly have a negative effect on the classification performance, because e.g. observations can be highly correlated since they stem of the same person [30]. Here, we further elaborate the idea of subject based bootstrapping [17]) and introduce a refined strategy in order to correctly select subsets of EMA data used in building the BBT.

We will focus on examples deriving from the dataset used in experiments (see Section IV) but the same process can easily be extended to any other EMA data with longitudinal structure. We consider that data points are consisted of repeated measurements belonging to P different subjects and use the following notation for the dataset:

$$\begin{matrix} \mathcal{L} = \left\{(y_i^{j(i)}, x_i^{j(i)})\right\} \\ i = 1, ..., P \qquad j(i) = 1, ..., J_i \end{matrix} \tag{4}$$

where $x_i^{j(i)} = (x_{i1}^{j(i)}, ..., x_{id}^{j(i)})$ is a d-dimensional predictor vector, $y_i^{j(i)} \in \{-1, 1\}$ is the class variable for the equivalent vector and J_i denotes the number of data samples per person i.

In order to create learning sets for the individual boosted trees (BT), B bootstrap samples of the set of subjects $\mathcal{S} = 1, ..., P$ are drawn with the drawn subjects denoted as i^*. To create the learning set we introduce the strategy $\mathcal{S}$ according to which one observation is drawn per subject. Thus, learning sets $\mathcal{L}_{b,s}^*$ of the B individual trees are defined as:

$$\begin{matrix} \mathcal{L}_{b,s}^* = \{(x_i, y_i), & i = 1, ..., P\} \\ & b = 1, ..., B \end{matrix} \tag{5}$$

where:

- (x_i, y_i) consists of the d-dimensional measurement x_i and class variable y_i,
- i is an index for the subjects included in the subset,
- P is the number of subjects,

Above process summarizes step 1 in the algorithm presented in Section III-A. The strategy where one random observation is used per tree is supported by relevant literature [17] but also is based on a simple rationale: When only one observation per subject is selected, the probability that different observations are used for the training of different trees is increased, although the same subjects might be selected which further reduces similarity between trees. By this way, we manage to incorporate advantages of subject based bootstrapping and observation based bootstrapping into the final BBT ensemble. Also, this approach can be applied to unbalanced data points per subject.

C. Balancing the data and CCPF computing

Classifying at the 1/2 quantile of the CCPF works well for binary classification problems but in the case of EMA data, sometimes classification with unequal costs or, equivalently, classification at quantiles other than 1/2 is needed. Strategies about correctly computing the CCPF are considered by over/under-sampling using the following process that converts a median classifier (like the BT in our case) into a q-classifier. The steps are the following:

- Let N_{+1} and N_{-1} be the marginal counts of positive and negative classes respectively. Choose values for k_{+1},$k_{-1} > 0$ so that $\frac{k_{+1}}{k_{-1}} = \frac{N_{+1}}{N_{-1}} / \frac{q}{1-q}$.
- Pick k_{+1} samples from the training set for which $y_i = 1$ such that each observation has the same chance of being selected.
- Pick k_{-1} samples for which $y_i = -1$ such that each observation has the same chance of being selected.
- Obtain a classifier using the usual process described before from the combined sample of $k_{+1} + k_{-1}$ points. Assume its output is a score function $F_m(x)$ such that

$F_m(x) > 0$ estimates $p(x) > 1/2$.

- Estimate x as having $p(x) > q$ if $F_m(x) > 0$.

In the case of $k < N$ (i.e., under-sampling), selection can be done by random sampling with or without replacement and in the case of $k > N$ (i.e., over-sampling) selection can be done either by sampling with replacement or by simply replicating observations (data augmentation). More details for this re-weighting/re-sampling scheme can be found in [31].

The next step is to convert the q-classifier to an estimator of the conditional class probabilities. Algorithm used was based on [10] and more details can be found there. The goal is to produce an estimate $\hat{p}(x)$ for the CCPF $p(x)$ given a q-classifier or more simply the estimation of the region $p(x) > q$ in which observations are classified in the positive class. First, a quantization level $\delta > 2$ is fixed. In our experiments, δ was set to 10, thus our estimate $\hat{p}(x)$ for $p(x)$ at any x will be one of $\{0.05, 0.15, ..., 0.95\}$ which requires our algorithm pipeline to run on nine artificial training datasets. Next, q-classification is carried out on the data (using our normal process as described in the previous sections) for the range of quantiles $q = 1/\delta, 2/\delta, ..., 1 - 1/\delta$. For each q and every x this provides an estimate of $I\{p(x) \geq q\}$ which we will denote as $\hat{\Delta}_q(x) \in \{0, 1\}$. In order to provide an estimate of $p(x)$ from the quantile-based estimates $\hat{\Delta}_q(x)$ we use the following process which ensures monotonicity of the level sets.

1) We begin with the median $\hat{\Delta}_{0.5}(x)$.
2) - If $\hat{\Delta}_{0.5}(x) = 1$ then
   $\hat{p}(x) = min\{q > 0.5 : \hat{\Delta}_q(x) = 0\} - \frac{1}{2\delta}$. If no such q is found then $\hat{p(x)} = 1 - 1/2\delta$.
   - If $\hat{\Delta}_{0.5}(x) = 0$ then
   $\hat{p}(x) = max\{q < 0.5 : \hat{\Delta}_q(x) = 1\} + \frac{1}{2\delta}$. If no such q is then take $\hat{p(x)} = 1/2\delta$.

IV. Experiments

A. Simulated dataset

As a first way to assess the usefulness and effectiveness of BBTs we conduct a series of experiments on simulated datasets, since a large-scale, real-world longitudinal dataset is not available. Prediction error is mostly used for assessing performance in longitudinal data, although not extensive research work exists [32]. The datasets contain different number of subjects (P = 50, 100, 500, 1000 or 2000) and different number of observations per subject (T = 10, 25, 50 or 100 observations per subject). Datasets created using the transactions dataset of Amazon [33] which originally contains 9484 transactions for 250 distinctive software titles, thus there are 250 different subjects with a varying number of observations per subject. Target variable (price) was converted to a two-class classification problem by dichotomizing it using different ways (in order to create balanced and unbalanced classes) and also some of the 20 numerical features were converted to categorical in order to demonstrate performance using both types of data.

In order to accurately measure out-of-sample performance for subjects present in the training dataset we use 75% of observations of P subjects for training and then we predict future observations for these subjects in order to estimate the out-of-sample performance (using the rest 25% as testing sample). During the splitting process, we make sure that there are enough samples for all subjects both in the training and the testing dataset.

The exponential loss function is used as a measure to assess the performance of class probability estimators (as it was mentioned in Section III-C). The formula used for computing the loss is shown in the following Equation:

$$\sum_{i=1}^{n^*} \left[p(x_i^*) \sqrt{\frac{1 - \hat{p}(x_i^*)}{\hat{p}(x_i^*)}} + (1 - p(x_i^*)) \sqrt{\frac{\hat{p}(x_i^*)}{1 - \hat{p}(x_i^*)}} \right] \quad (6)$$

where $\hat{p}$ is the estimate probability and p is the actual probability computed on the hold-out sample x_i^*.

Prediction error results can be found in Table I. Standard optimized implementations of the state-of-the-art algorithms (Single Classification Tree (SCT), Bagging, Boosting, Random Forest) were used utilizing "rpart", "adabag" and "party" packages in R and for the B&B combine method implementation details can be found in [16]. For a relatively small number of P, BBT performs comparably with B&B combination and better than the other algorithms. As P increases (i.e. data derive from different subjects), performance of BBT is improving and outperforms all other methods (B&B combine included). Experiments were also conducted for different T values (different observations per subject). For large T, algorithms perform better because there is enough availability of data for all subjects within the study but for smaller T performance of BBT is superior. Results for $P = 2000$ and different values of T are presented in Table II.

The exponential loss function comparison for different algorithms is presented in Figure 1. This Figure shows the results for $P = 2000$, $T = 200$ and a relatively unbalanced target class (70% for positive class and 30% for negative class) but similar results were obtained for other values as well. From this Figure it becomes apparent that boosting algorithms (except for BBT) tend to overfit the loss function, thus not correctly computing the CCPF (bagging algorithms were excluded from this experiment as they are not expected to overfit). When performance of most algorithms improves (in terms or prediction error) after 200 iterations, the loss function deteriorates and experiments showed that this trend is more prevalent when the imbalance in the output class is increasing.

B. Experiments on a real-world EMA dataset

In order to illustrate the effect of BBT, we now apply this method to our EMA dataset obtained by a study designed by the authors [34]. The EMA study followed 100 participants over the course of 14 days. Every day, subjects were randomly notified by a beeper (random sampling) between 0730 and 2230 with an interval of two hours. Besides that, when they

TABLE I: Prediction Error (%) for different algorithms and different number of subjects (using all available observations per subject)

	P=50	P=100	P=500	P=1000	P=2000
SCT	0.47	0.46	0.48	0.48	0.49
Bagging	0.41	0.42	0.45	0.44	0.46
Boosting	0.34	0.35	0.37	0.36	0.38
Random Forest	0.34	0.35	0.34	0.32	0.34
B&B Combine	**0.32**	**0.32**	0.32	0.33	0.35
BBT	0.34	**0.32**	**0.31**	**0.29**	**0.28**

TABLE II: Prediction Error (%) for different algorithms and different number of observations per subject (using P=2000)

	T=10	T=25	T=100	T=200
SCT	0.51	0.49	0.44	0.42
Bagging	0.38	0.36	0.44	0.40
Boosting	0.41	0.4	0.38	0.37
Random Forest	0.37	0.38	0.36	0.35
B&B Combine	0.38	0.37	0.34	0.31
BBT	**0.33**	**0.31**	**0.29**	**0.28**

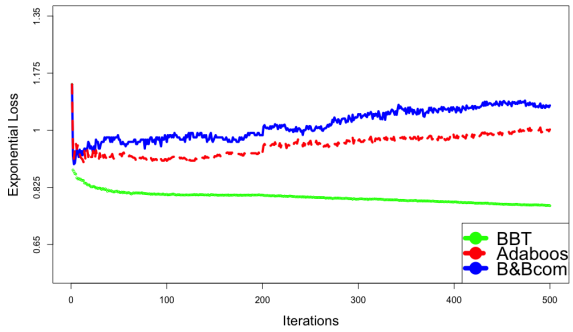

Fig. 1: Exponential loss for different algorithms for simulated dataset

were about to eat something they fill out a similar questionnaire which also contained the food information. This process resulted in an average of 10 responses (including random samples and eating events) per user per day. The dataset is multi-level and complex containing information about users and their eating events, emotions, circumstances, locations for several time moments during each day that they participated in the study. An overview of the variables involved in the study is presented in Table III.

user	Date/time	crv	negE	posE	sp_cr	time	week	circ	loc	sp_eat
pp5	26/01/15 23:57	LOW	NO	LOW	N	evening	NO	LowLevel	Home	N
pp5	27/01/15 09:32	LOW	NO	HIGH	N	morning	NO	LowLevel	Home	N
pp5	27/01/15 12:17	MID	NO	HIGH	H	noon-after	NO	ComputerRelated	Work	U
pp5	27/01/15 14:43	LOW	YES	MID	N	noon-after	NO	Work	Work	H

user	crv	negE	posE	sp_cr	time	week	circ	loc	sp_eat	NextEating (y class)
pp5	LOW	NO	HIGH	N	morning	NO	LowLevel	Home	N	U
pp5	MID	NO	HIGH	H	noon-after	NO	ComputerRelated	Work	U	H

Fig. 2: Data conversion example for early prediction

Each data point is used to predict whether the next data point (provided that they both occur on the same day) will be a healthy or an unhealthy eating moment. Figure 2 shows an example of how data points (belonging to user "pp5") are converted and combined in order to enable early prediction using a classification algorithm.

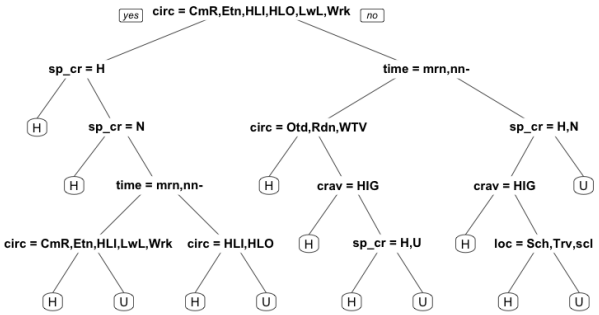

Fig. 3: Classification tree structure

Using the data points of Figure 2 (from 100 people) as observations, we want to predict under which conditions (i.e. combinations of attributes) people are led to unhealthy eating (class variable, y), thus the BBT will be applied to a binary classification problem (class can be either "healthy" or "unhealthy"). There are 9 variables which represent user status at "current" time-point will be used to predict the eating event at the "next" time-point. In total, there are 5041 data points deriving from 100 different users which will be used for training and testing different decision trees. Splitting to training and testing was performed according to the methodology presented in the previous Subsection (75% training and 25% testing) but also by sampling in a semi-random way in order to ensure that person-specific data are respectively split between training and testing. Variance reported is an average over all different experiments (for all subjects) conducted (10-fold-cross-validation) and results were found significant. Also, within-person variances were computed and values were not significant to be reported (although BBT performed way better).

A single classification tree (SCT) was fitted to the data with the size of tree (14 terminal nodes) selected by cross-validation (Figure 3). The 13 splits were based on five out of the nine variables and their importance was led by "circ", "sp_cr" and "time" (around 15%) but also showed substantial variation for all predictors. Prediction Error (PE) for this SCT was 37.3%.

A series of BBT were then fitted to the data. The first BBT, comprising trees with 10 splits and using all nine predictors, had PE of 28.6% which is an important improvement on the SCT. The partial dependency plots of the single predictors are presented in Figure 4, along with the importance values which are depicted on top of each plot and suggest a different

TABLE III: Thinkslim dataset attributes

Attribute	Short	Discretized values	Details
Craving	crv	Low, Mid, High	
Negative Emotions	negE	No, Yes	sad, bored, stressed, angry
Positive Emotions	posE	Low, Mid, High	happy, relaxed
Location	loc	Home, School, Traveling, Work, Social, Other	
Circumstances	circ	ComputerRelated	Phone / Internet / Computer
		Eating	Eating / Non-social drinking
		HighLevelIn	Preparing food, cleaning, sanitary, etc.
		HighLevelOut	Exercising, hobby, leisure, shopping, etc.
		LowLevel	Relaxing, waiting, lying in bed, etc.
		WatchingTV	
		Reading	Studying, thinking, etc.
		Socializing	Having a drink, etc.
		Outdoors	traveling, etc.
		Working	administration, work activities, etc.
Time of day	time	morning, noon-afternoon, evening	
Weekend	week	NO, YES	
Specific Craving	sp_cr	N, H, U	Nothing, Healthy, Unhealthy
Specific Eating	sp_eat	N, H, U	Nothing, Healthy, Unhealthy

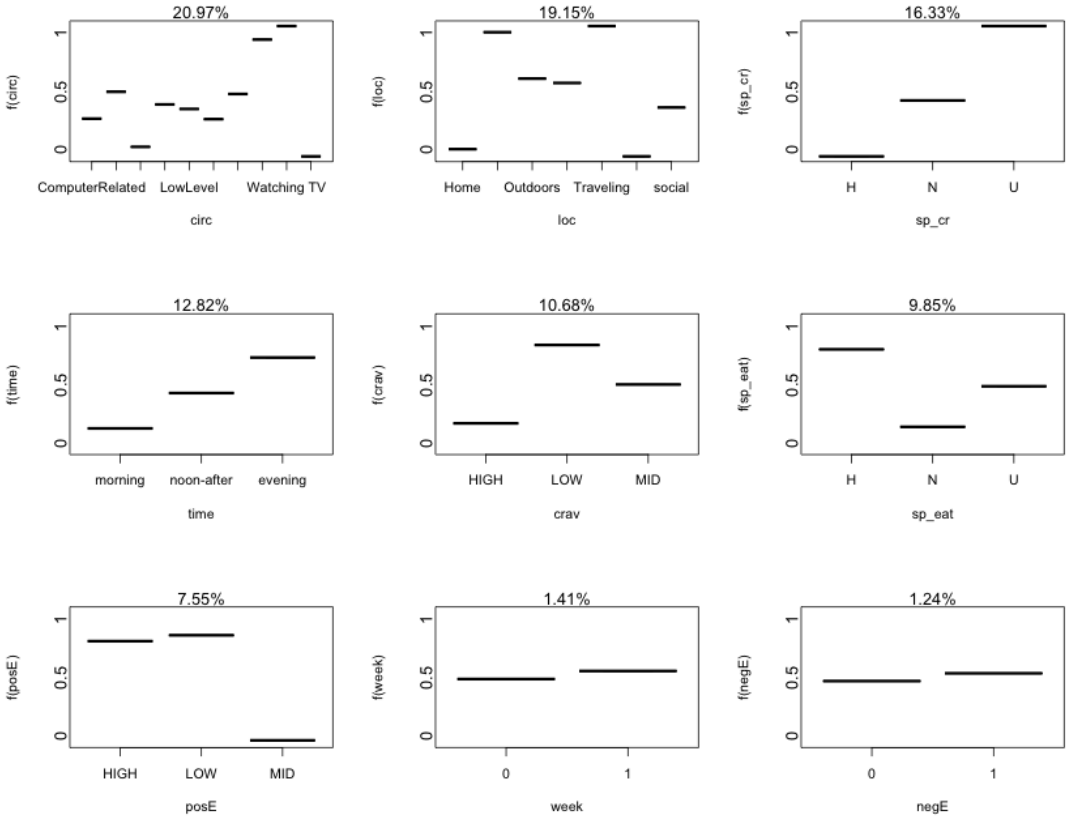

Fig. 4: Bagged Boosted Tree (BBT) analysis

order than the SCT: They show strong effects for "circ", "loc", "sp_cr", moderate effects for "time", "crav", "sp_eat" and "posE", and weak effects for "week" and "negE".

Next experiments were done to determine the degree to which predictors interact in providing the response. Firstly, "week" and "negE" were dropped from the model and reduced PE to 23.3%, something that was expected from the partial dependency plots (since they had the lowest effect). Then, we fitted a BBT by forcing individual trees having only single splits (meaning that the estimated response would depend only on main effects) and the PE was increased to 26.7%, indicating that interactions accounted for 3.4% of PE. Finally, BBT with individual trees of two splits (including first-order interactions) yielded a PE of 25.9%, indicating that interactions higher than first-order could be neglected.

In the comparison between methods, BBT gave a PE of 23.3%, whereas the single classification tree (37.3%), bagged trees (28.9%), boosted trees (adaboost) (25.9%) and random forests (26.8%), all having higher PE than the BBT. Figure 5 summarizes these results and also presents a series of

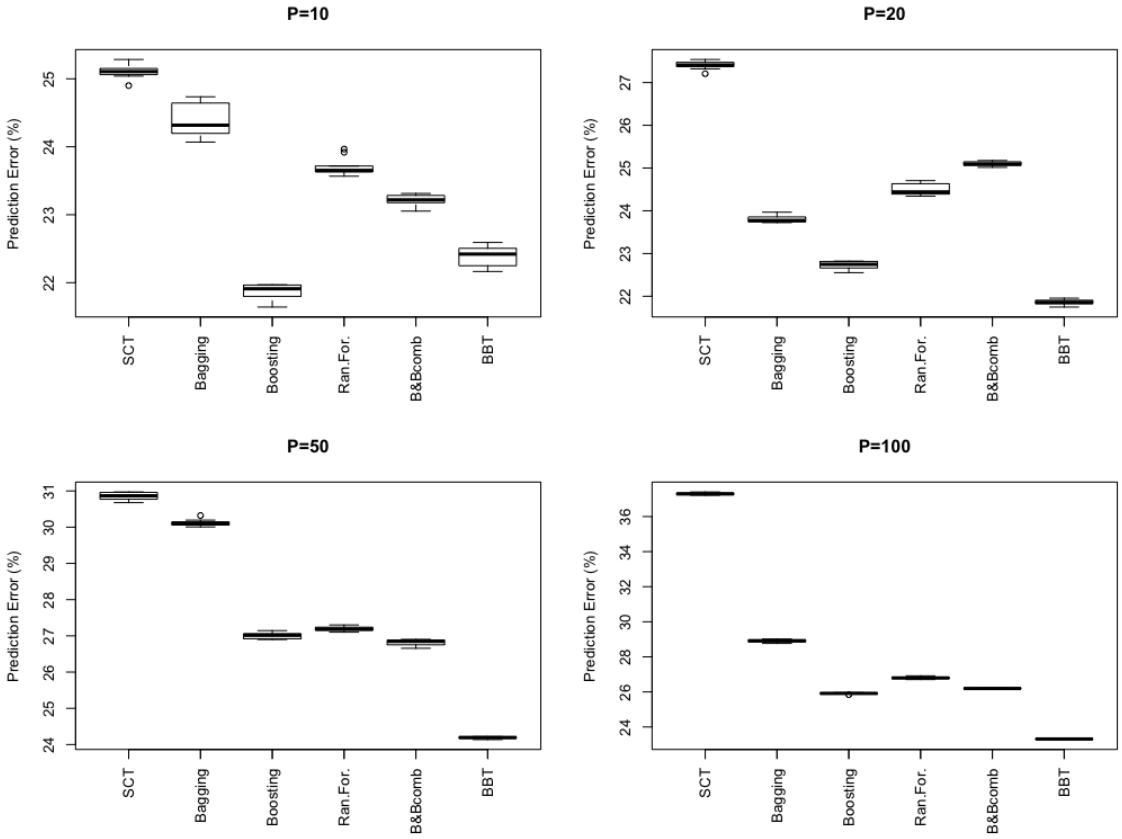

Fig. 5: Prediction Error for different algorithms and different numbers of subjects

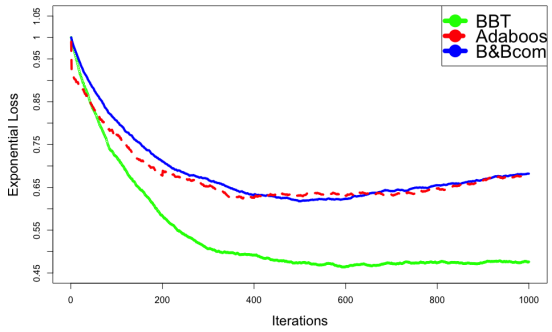

Fig. 6: Exponential loss for different algorithms

experiments made to demonstrate the effectiveness of BBT when the number of different subjects (P) involved in the dataset increases. For relatively small numbers of subjects (10 or 20) performance of BBT and AdaBoost is comparable (although variance increases and the dataset becomes too small for accurate predictions) but as P increases the performance of BBT is clearly better. As P increases (which means that there are more subjects in the dataset), complexity of longitudinal structure increases, thus it is more imperative to take this into account when classifying longitudinal data. This is the reason that BBT performs better than all other algorithms as P increases. However, for small P the effect of different subjects is smaller and this is the reason that Adaboost performs slightly better than all other algorithms. It should be noticed that these observations are in agreement with the results from the experiments on the simulated datasets. Finally, for each algorithm the best tinkered (optimized) result using standard techniques is used for the comparisons in order to overcome individual algorithmic drawbacks.

Regarding the exponential loss, results for the different algorithms are shown in Figure 6 from which it is obvious that B&B combine and Adaboost tend to overfit the loss function which counterbalances their modest accuracy. This is not the case for BBT which using the over/under sampling method described manages to improve the exponential loss and provides an accurate estimate for the CCPF.

V. Conclusion

In this paper an improvement to single trees for handling classification problems in EMA data was presented, namely Bagged Boosted Trees (BBT). BBT are able to modify classification trees and overcome the unsatisfactory performance characteristics in terms of accuracy and in terms of data dependency due to the EMA structure. An estimate of the class probability distribution was presented based on over/under sampling of data.

BBT can outperform the single classification trees but also the boosted or bagged trees as well similar models such as random forests or other combinations of bagging and boosting. Furthermore, they have the advantage of being able to deal with multiple categorical data which raises a scalability issue when dealing with classic models (like generalized linear models) that are widely used in EMA studies. Moreover, BBT can tackle potential nonlinearities and interactions in the data, since these issues are handled through the combination of many different trees of different sizes.

The experimental results of BBT both on simulated and real-world EMA data clearly demonstrate improvement with respect to accuracy in prediction compared to classic decision tree algorithms, while at the same time a better estimate for the conditional class probability function is computed. We see several promising avenues for further improvement and for continuing research using BBT. Application of BBT to other EMA datasets but also to more complex data (involving more numerical or categorical predictors) is a first step to further test BBT. Moreover, adjustment of boosting in order to implement weights based on subjects (and not individual observations) would be an extension with promising results. Finally, faster training is an issue to look into, since the complexity of the model is increased and many steps are needed (bagging, boosting, quantile estimation) in order to achieve the final result.

ACKNOWLEDGMENT

This study is funded by grant 12028 from Stichting Technische Wetenschappen (STW), Nationaal Initiatief Hersenen en Cognitie (NIHC), Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) and Philips under the Partnership programme Healthy Lifestyle Solutions.

REFERENCES

[1] S. Shiffman, A. A. Stone, and M. R. Hufford, "Ecological momentary assessment," *Annu. Rev. Clin. Psychol.*, vol. 4, pp. 1–32, 2008.

[2] C. N. Scollon, C.-K. Prieto, and E. Diener, "Experience sampling: promises and pitfalls, strength and weaknesses," in *Assessing well-being*. Springer, 2009, pp. 157–180.

[3] J. E. Spook, T. Paulussen, G. Kok, and P. Van Empelen, "Monitoring dietary intake and physical activity electronically: feasibility, usability, and ecological validity of a mobile-based ecological momentary assessment tool," *Journal of medical Internet research*, vol. 15, no. 9, p. e214, 2013.

[4] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple classifier systems*. Springer, 2000, pp. 1–15.

[5] A. Rani, G. L. Foresti, and C. Micheloni, "A neural tree for classification using convex objective function," *Pattern Recognition Letters*, vol. 68, pp. 41–47, 2015.

[6] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[7] T. K. Ho, "Random decision forests," in *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, vol. 1. IEEE, 1995, pp. 278–282.

[8] Y. Freund, R. E. Schapire *et al.*, "Experiments with a new boosting algorithm," in *ICML*, vol. 96, 1996, pp. 148–156.

[9] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 161–168. [Online]. Available: http://doi.acm.org/10.1145/1143844.1143865

[10] D. Mease, A. J. Wyner, and A. Buja, "Boosted classification trees and class probability/quantile estimation," *J. Mach. Learn. Res.*, vol. 8, pp. 409–439, May 2007.

[11] G. De'Ath, "Boosted trees for ecological modeling and prediction," *Ecology*, vol. 88, no. 1, pp. 243–251, 2007.

[12] C. D. Sutton, "Classification and regression trees, bagging, and boosting," *Handbook of statistics*, vol. 24, pp. 303–329, 2005.

[13] Y. L. Suen, P. Melville, and R. J. Mooney, "Combining bias and variance reduction techniques for regression trees," in *Machine Learning: ECML 2005*. Springer, 2005, pp. 741–749.

[14] S. Kotsiantis and P. Pintelas, "Combining bagging and boosting," *International Journal of Computational Intelligence*, vol. 1, no. 4, pp. 324–333, 2004.

[15] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[16] J. Xie, V. Rojkova, S. Pal, and S. Coggeshall, "A combination of boosting and bagging for kdd cup 2009-fast scoring on a large database." in *KDD Cup*, 2009, pp. 35–43.

[17] W. Adler, S. Potapov, and B. Lausen, "Classification of repeated measurements data using tree-based ensemble methods," *Computational Statistics*, vol. 26, no. 2, pp. 355–369, 2011.

[18] R. J. Sela and J. S. Simonoff, "RE-EM trees: a data mining approach for longitudinal and clustered data," *Machine learning*, vol. 86, no. 2, pp. 169–207, 2012.

[19] W.-Y. Loh, W. Zheng *et al.*, "Regression trees for longitudinal and multiresponse data," *The Annals of Applied Statistics*, vol. 7, no. 1, pp. 495–522, 2013.

[20] W. Fu and J. S. Simonoff, "Unbiased regression trees for longitudinal and clustered data," *Computational Statistics & Data Analysis*, vol. 88, pp. 53–74, 2015.

[21] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Machine learning*, vol. 36, no. 1-2, pp. 105–139, 1999.

[22] M. Kuhn and K. Johnson, *Applied predictive modeling*. Springer, 2013.

[23] J. Elith, J. R. Leathwick, and T. Hastie, "A working guide to boosted regression trees," *Journal of Animal Ecology*, vol. 77, no. 4, pp. 802–813, 2008.

[24] M. A. Munson and R. Caruana, "On feature selection, bias-variance, and bagging," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2009, pp. 144–159.

[25] P. Bühlmann and B. Yu, "Boosting with the L2 loss: regression and classification," *Journal of the American Statistical Association*, vol. 98, no. 462, pp. 324–339, 2003.

[26] T. J. Hastie and R. J. Tibshirani, *Generalized additive models*. CRC Press, 1990, vol. 43.

[27] P. J. Bickel, Y. Ritov, and A. Zakai, "Some theory for generalized boosting algorithms," *The Journal of Machine Learning Research*, vol. 7, pp. 705–732, 2006.

[28] J. Friedman, T. Hastie, R. Tibshirani *et al.*, "Additive logistic regression: a statistical view of boosting," *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[29] A. Buja, W. Stuetzle, and Y. Shen, "Loss functions for binary class probability estimation and classification: Structure and applications," *Working draft, November*, 2005.

[30] W. Adler and B. Lausen, "Bootstrap estimated sensitivities, specificities and ROC curve," *Comput Stat Data Anal*, vol. 53, no. 3, pp. 718–729, 2009.

[31] D. Mease, A. Wyner, and A. Buja, "Cost-weighted boosting with jittering and over/under-sampling: Jous-boost," *J. Machine Learning Research*, vol. 8, pp. 409–439, 2007.

[32] D. Afshartous and J. de Leeuw, "Prediction in multilevel models," *Journal of Educational and Behavioral Statistics*, vol. 30, no. 2, pp. 109–139, 2005.

[33] A. Ghose, P. G. Ipeirotis, and A. Sundararajan, "The dimensions of reputation in electronic markets," *NYU Center for Digital Economy Research Working Paper No. CeDER-06-02*, 2009.

[34] G. Spanakis, G. Weiss, B. Boh, and A. Roefs, "Network analysis of ecological momentary assessment data for monitoring and understanding eating behavior," in *Smart Health - International Conference, ICSH 2015, Phoenix, AZ, USA, November 17-18, 2015*, 2015, pp. 43–54.