BoostEMM — Transparent Boosting using Exceptional Model Mining

Simon van der Zon[1], Oren Zeev Ben Mordehai[1], Tom Vrijdag[1], Werner van Ipenburg[2], Jan Veldsink[2], Wouter Duivesteijn[1], and Mykola Pechenizkiy[1]

[1] Eindhoven University of Technology, the Netherlands,
{s.b.v.d.zon, o.zeev.ben.mordehay, w.duivesteijn, m.pechenizkiy}@tue.nl
t.s.vrijdag@student.tue.nl
[2] Cooperatieve Rabobank U.A., the Netherlands,
{werner.van.ipenburg, jan.veldsink}@rabobank.nl

Abstract. Boosting is an iterative ensemble-learning paradigm. Every iteration, a weak predictor learns a classification task, taking into account performance achieved in previous iterations. This is done by assigning weights to individual records of the dataset, which are increased if the record is misclassified by the previous weak predictor. Hence, subsequent predictors learn to focus on problematic records in the dataset. Boosting ensembles such as AdaBoost have shown to be effective models at fighting both high variance and high bias, even in challenging situations such as class imbalance. However, some aspects of AdaBoost might imply limitations for its deployment in the real world. On the one hand, focusing on problematic records can lead to overfitting in the presence of random noise. On the other hand, learning a boosting ensemble that assigns higher weights to hard-to-classify people might throw up serious questions in the age of responsible and transparent data analytics; if a bank must tell a customer that they are denied a loan, because the underlying algorithm made a decision specifically focusing the customer since they are hard to classify, this could be legally dubious. To kill these two birds with one stone, we introduce BoostEMM: a variant of AdaBoost where in every iteration of the procedure, rather than boosting problematic records, we boost problematic subgroups as found through Exceptional Model Mining. Boosted records being part of a coherent group should prevent overfitting, and explicit definitions of the subgroups of people being boosted enhances the transparency of the algorithm.

Keywords: Boosting, class imbalance, Exceptional Model Mining, model transparency, responsible analytics

1 Introduction

Arguably, AdaBoost [10] can be described as one of the best-performing out-of-the-box classifiers. By evaluating the performance of simple base classifiers, the algorithm learns which records of a dataset are easy/hard to classify. Subsequent base classifiers focus on more problematic records. By properly weighing the decisions of the earlier, blunter base classifiers with the later, more specific ones, an ensemble classifier is built that performs well overall. Hence, AdaBoost as a mechanism shines in devoting appropriate attention of a classifier to those records of the dataset that prove to be problematic.

For all its strenghts, AdaBoost also comes with two weaknesses. The records proving problematic for the classifier might encompass outliers, which could lead to overfitting. Moreover, in the age of responsible data analytics, we want to know not only *what* our algorithm does, but also *why it is reasonable*. A recent European Parliament resolution on the future of robotics and artificial intelligence in Europe contains the following [9, Section "Ethical principles", point 12]:

> [. . .] it should always be possible to supply the rationale behind any decision taken with the aid of AI that can have a substantive impact on one or more persons' lives; [. . .] it must always be possible to reduce the AI system's computations to a form comprehensible by humans;

If the decision in a dataset on loan applications is outsourced to AdaBoost, the customer might demand insight in the reasoning behind rejection. When this resolution is put into law, it is likely that a customer demanding insight in the reasoning AdaBoost deployed behind its decision (which has a substantive impact on the customer's life), must be presented with not only *how* AdaBoost decides where to focus its extra attention, but also *why*. Hence, in the near future, financial institutions will shy away from the liability associated with using AdaBoost if there is no transparent form of boosting.

In this paper, we fill that void by proposing *BoostEMM*. This method combines AdaBoost-style iterative learning of base classifiers, where the focus shifts towards problematic parts of input space, with Exceptional Model Mining (EMM) [18,5]. This is a local pattern mining method, designed to find subgroups (subsets of the dataset at hand) that satisfy two conditions. On the one hand, subgroups must be interpretable. This is typically enforced by only allowing subgroups that can be defined as a conjunction of few conditions on input attributes of the dataset; hence subgroups come in terms that any domain expert can understand. On the other hand, subgroups must be exceptional. This typically is formalized in terms of an unusual interaction between several target attributes; hence subgroups represent unusual behavior in the dataset. We choose targets that represent the actual class label and the label predicted by base classifiers, and define several quality measures that gauge unusual interaction between those targets. Hence, for various types of bad base classifier performance, EMM finds coherent parts of the classifier input space where this behavior occurs.

1.1 Main Contributions

We provide *BoostEMM*, a method encompassing core ideas from both AdaBoost and Exceptional Model Mining. By dovetailing these techniques, BoostEMM achieves two benefits over AdaBoost:

1. by defining specific EMM variants (cf. Section 3.3), we steer boosting to punish specific kinds of bad behavior (error rate, class imbalance, FPR/TPR), which is relevant for cost-sensitive applications;
2. by dovetailing an EMM run with every iteration of the boosting algorithm, on every step we can report subgroups where extra boosting is needed. This adds transparency to the boosting process (cf. Section 3.4), which is relevant in the light of looming EU law.

2 Related Work

The groundbreaking paper on AdaBoost in its traditional form is [10]; this form is also explained in Section 3. A version incorporating class probabilities was introduced in [11]. Convex potential boosting algorithms (which includes AdaBoost) cannot handle random classification noise well [20]; boosting tends to overfit towards the noisily labeled records of the dataset, reducing generality of the learned classification model.

The resurgence of neural networks through the deep learning hype has led to a reappreciation of complex classifiers, that perform extremely well on the task for which they have been designed, but whose internal reasoning is far too complex for a human to fully understand. As a reaction, papers emerge that take a peek into the black box. Some of the first such papers include [12,13] for hard classifiers, and [8] for soft classifiers. These papers share the objective of transparency with BoostEMM, but they do not (as BoostEMM does) loop back the interpretable results into the classification process to improve performance.

The study of how local patterns can aid global models was the topic of the LeGo workshop [16]. This workshop encompasses both papers *enhancing* existing classifiers with local patterns, or *combining* local patterns *into* a classifier. A few years later, LeGo-EMM [7] enhanced multi-label classifiers with local patterns found through Exceptional Model Mining with the Bayesian networks model class [6], which improved multi-label SVN classification. These methods have in common that the learning process is a single line: first local patterns are mined, then a subset of those patterns is selected, and finally that subset is used to enhance/replace the feature set of a classifier, with which subsequently predictions are made. Hence, LeGo papers share the incorporation of local pattern knowledge in classification with BoostEMM, but they do not (as BoostEMM does) loop back the output into an iterative learning process.

Exceptional Model Mining seeks to find subgroups of a dataset where several targets interact in an unusual manner. A simpler cousin of EMM is Subgroup Discovery (SD) [15,21,14]: the task of finding subgroups of a dataset where a single target displays an unusual distribution. SD is closely related to Contrast Set Mining (CSM) [1] and Emerging Pattern Mining (EPM) [3]; the results of the latter technique have also been exploited to enhance classification [4], in the style of the LeGo workshop papers discussed in the previous paragraph. The relation between SD, CSM, and EPM is explored in detail in [17].

3 The BoostEMM Method

Given a dataset Ω, which is a bag of N records $r \in \Omega$ of the form $r = (a_1, \ldots, a_k, \ell)$, where $\{a_1, \ldots, a_k\}$ are the input attributes of the dataset, taken from some collective domain $\mathcal{A}$, and ℓ is the class label, typically taken to be binary. If we need to refer to a specific record (or corresponding data components), we do so by superscripts: the i^{th} record is denoted r^i, ℓ^i is its class label, and the value of its j^{th} input attribute is a^i_j. We also use the shorthand $\mathfrak{a}^i$ to denote the collective input attribute values of r^i: $\mathfrak{a}^i = (a^i_1, \ldots, a^i_k)$.

The goal of classification is to find a mapping P from the input attribute space to the class label; $P : \mathcal{A} \to \{0, 1\}$, such that we can predict the latter for unseen instances of the former. In boosting, these predictions are improved through the following methodology. We iteratively build one *strong learner* or *expert* $\mathcal{P} = (E, \mathfrak{W})$. This is done by constructing an *ensemble* E of h weak learners or *predictors* (i.e. classifiers that perform (slightly) better than random) $E = (P_1, \ldots, P_h)$, and an associated tuple $\mathfrak{W} = (\mathfrak{w}_1, \ldots, \mathfrak{w}_h)$ of weights related to the performance of each predictor. AdaBoost obtains these weights for each weak classifier P_i by a transformation (cf. Section 3.2) of its error rate err_i. In each iteration of the boosting process, a new weak learner P_j is constructed, taking into account the whole training set but also the up-to-date priorities, or weights, of the records. These weights are maintained as another tuple of N weights W, associated with the records of the dataset: $W = (w^1, \ldots, w^N)$. In the first iteration, all training data (unless given initial weights by the end user) are initialized with equal weights $w^i = {}^1\!/_N$, and the first weak learner is trained. In subsequent iterations, the weights are increased for all samples that are classified incorrectly by $\mathcal{P}$, after which the weights are normalized. In later sections, we replace the selection mechanism for the erroneous samples by an equivalent EMM function defining the subgroups to be boosted. AdaBoost updates these weights W in a manner similar to the weights $\mathfrak{W}$, by a transformation (cf. Section 3.2) based on err_j.

3.1 Exceptional Model Mining

Given a dataset Ω, which is a bag of N records $r \in \Omega$ of the form $(a_1, \ldots, a_k, t_1, \ldots, t_m)$, where $\{a_1, \ldots, a_k\}$ are the descriptors of the dataset, and $\{t_1, \ldots, t_m\}$ the targets. The goal of EMM is to find subgroups of the dataset at hand, defined in terms of a conjunction of a few conditions on single descriptors of the dataset (e.g.: $a_7 \leq 3 \wedge a_3 = \text{true}$), for which the targets interact in an unusual manner.

Definition 1 (Subgroup). *A* subgroup *corresponding to a description D is the bag of records $G_D \subseteq \Omega$ that D covers, i.e.*

$$G_D = \{\ r^i \in \Omega \mid D(\mathfrak{a}^i) = 1\ \}$$

From now on we omit the D if no confusion can arise, and refer to the *coverage* of a subgroup by $n = |G|$.

In order to objectively evaluate a candidate description in a given dataset, we need to define a quality measure. For each description D in a user-defined description language $\mathcal{D}$, this function quantifies the extent to which the subgroup G_D corresponding to the description deviates from the norm.

Definition 2 (Quality Measure). *A* quality measure *is a function* $\varphi : \mathcal{D} \to \mathbb{R}$ *that assigns a numeric value to a description* D*.*

3.2 Mining Descriptions for Boosting and Updating the Weights

The input attributes in classification/boosting correspond to the descriptors in EMM. Having trained a weak learner P_j, we generate targets that reflect how well the classification performs on each record. We explore several choices for unusual interaction between these targets in Section 3.3. Having thusly defined a model class for EMM, we run the beam search algorithm [5, Algorithm 1] to generate a set $\mathfrak{D}_{\text{top-}q}$ of subgroups G_D with their associated descriptions D. AdaBoost constructs the subset to be boosted by simply picking all erroneously classified samples. Instead, BoostEMM picks every record that is covered by at least one of the top subgroups we found with EMM. Hence, BoostEMM adheres to the following scheme:

$$\begin{aligned}
err_j &= \frac{1}{\sum\limits_{i=1}^{N} w^i} \sum_{i=1}^{N} w^i \mathbb{I}(\ell^i \neq P_j(\mathfrak{a}^i)) \\
\alpha_j &= \log \frac{1 - err_j}{err_j} \\
w^i &\leftarrow w^i \cdot \exp\left(\alpha_j \cdot \mathbb{I}\left(\left\{ \mathfrak{a}^i \in \Omega \;\middle|\; \exists_{D_j \in \mathfrak{D}_{\text{top-}q}} : D_j(\mathfrak{a}^i) = 1 \right\} \right) \right) \\
\mathcal{P}(\mathfrak{a}) &= \operatorname*{arg\,max}_{\ell \in \{0,1\}} \sum_{m=1}^{M} \alpha_j \cdot \mathbb{I}(P_j(\mathfrak{a}) = \ell)
\end{aligned}$$

In AdaBoost, the weight update function is instead given by:

$$w^i \leftarrow w^i \cdot \exp(\alpha_j \cdot \mathbb{I}(\ell^i \neq P_j(\mathfrak{a}^i)))$$

3.3 The Transparent Boosting Model Class for EMM

The missing ingredient in the description of BoostEMM in the previous section, is: how do we find the subgroup set $\mathfrak{D}_{\text{top-}q}$? We do so by Exceptional Model Mining. Within BoostEMM, every EMM run is encompassed by the boosting process. Hence, we have just trained a weak learner P_j to predict a specific class label ℓ for every possible input attribute vector $\mathfrak{a} \in \mathcal{A}$. Within this setting, in order to employ EMM, we need to cast the available building blocks into a form that fits the EMM problem specification, as outlined in Section 3.1. We need to describe the dataset in terms of descriptors and targets, formulate a model class over the targets, and define a quality measure over this model class.

For the descriptors in EMM we take the input attributes $a_1^i, \ldots, a_k^i$ of the classification task given at the start of Section 3. In the Transparent Boosting model class for EMM there are two targets: the original class label, $t_1^i = \ell^i$, and the class label predicted by the available weak learner P_j, $t_2^i = P_j(\mathfrak{a}^i)$. The kind of interaction in which the Transparent Boosting model class is interested, is an exceptional discord between the original class label and the predicted class label: where does our weak learner perform not so well?

The last question can be answered in many reasonable manners. Which answer we choose depends on what kind of boosting we want to achieve. In EMM, the *quality measure* governs what exactly we find interesting within the kind of interaction defined by the model class. As is common in EMM, we build up the quality measure from two components: $\varphi_{\mathrm{TB}}(D) = \varphi_{\mathrm{size}}(D) \cdot \varphi_{\mathrm{dev}}(D)$. The latter component measures the exceptionality degree of target interaction. Since a large value for this can easily be obtained in tiny subgroups, we need to prevent overfitting by multiplying with a component explicitly representing subgroup size. For this, we take $\varphi_{\mathrm{size}}(D) = \log(|G_D|)$. We employ the logarithm here, since we do not want to put a penalty on medium-sized subgroups compared to large subgroups; this component is only meant to discourage tiny subgroups. For the deviation component $\varphi_{\mathrm{dev}}(D)$, we develop four alternatives.

Error-based boosting with $\boldsymbol{\varphi_{\mathrm{err}}}$ If one would merely be interested in the error rate, we define the target interaction exceptionality of the subgroup as follows.

$$\varphi_{\mathrm{err}}(D) = \frac{\sum\limits_{i:D(\mathfrak{a}^i)=1} w^i \mathbb{I}(\ell^i \neq P_j(\mathfrak{a}^i))}{\sum\limits_{i:D(\mathfrak{a}^i)=1} w^i}$$

This quality measure computes the error rate, but only on the records covered by the subgroup. Hence, unlike AdaBoost, BoostEMM will *also* boost records of the dataset that were classified *correctly* by the weak learner. *This is deliberate,* since this ought to reduce the overfitting effect from which AdaBoost suffers.

Kappa-based boosting with $\boldsymbol{\varphi_{\kappa}}$ In the presence of class imbalance, optimizing for the Kappa statistic is more appropriate than the error rate.

$$\varphi_\kappa(D) = \frac{\mathrm{ACC}_{\mathrm{OBS}}(D) - \mathrm{ACC}_{\mathrm{EXP}}(D)}{1 - \mathrm{ACC}_{\mathrm{EXP}}(D)}, \text{ where} \qquad \varSigma_w(D) = \sum_{i:D(\mathfrak{a}^i)=1} w^i$$

$$\mathrm{ACC}_{\mathrm{OBS}}(D) = {}^{1}\!/_{\varSigma_w(D)} \cdot \sum_{i:D(\mathfrak{a}^i)=1} w^i \mathbb{I}(\ell^i = P_j(\mathfrak{a}^i))$$

$$\mathrm{ACC}_{\mathrm{EXP}}(D) = {}^{\left(\mathrm{pos}(D)\cdot\mathrm{pos}_p(D)+\mathrm{neg}(D)\cdot\mathrm{neg}_p(D)\right)}\!/_{(\varSigma_w(D))^2}$$

$$\begin{array}{ll} \mathrm{pos}(D) = \sum\limits_{i:D(\mathfrak{a}^i)=1} w^i \cdot \mathbb{I}(\ell^i = 1) & \mathrm{pos}_p(D) = \sum\limits_{i:D(\mathfrak{a}^i)=1} w^i \cdot \mathbb{I}(P(\mathfrak{a}^i) = 1) \\ \mathrm{neg}(D) = \sum\limits_{i:D(\mathfrak{a}^i)=1} w^i \cdot \mathbb{I}(\ell^i = 0) & \mathrm{neg}_p(D) = \sum\limits_{i:D(\mathfrak{a}^i)=1} w^i \cdot \mathbb{I}(P(\mathfrak{a}^i) = 0) \end{array}$$

Cost-sensitive boosting with $\varphi_{\mathbf{FNR}}$ and $\varphi_{\mathbf{FPR}}$ Based on the dataset domain at hand, one might be interested in cost-sensitive classification. When the cost of false negatives and false positives is substantially skewed, one would desire to find subgroups that boost for either of these components. Hence, we employ each type of classification mistake as a deviation component of its own.

$$\varphi_{\text{FNR}}(D) = {}^1\!/\!_n \sum_{i:D(\mathfrak{a}^i)=1} w^i \cdot \mathbb{I}(\ell^i \neq P_j(\mathfrak{a}^i) \wedge \ell^i = 1)$$

$$\varphi_{\text{FPR}}(D) = {}^1\!/\!_n \sum_{i:D(\mathfrak{a}^i)=1} w^i \cdot \mathbb{I}(\ell^i \neq P_j(\mathfrak{a}^i) \wedge \ell^i = 0)$$

3.4 Model Transparency

After one has trained an ensemble using boosting, it is insightful to know how the ensemble was constructed (i.e. which data was emphasized most during the boosting process). Especially when boosting for various quality measures, it can be interesting to see how the various boosting strategies behave. We present a visualization that shows the user exactly which regions have (successfully) been boosted the most. Our method can show a high number of descriptions by visualizing them in a tree. The tree is constructed by looping over the descriptions. For each description we create a branch:

1. A branch consists of nodes represented by the literals of a description, and the root of the branch is the first literal (which makes sense, since each following literal is a refinement on the description).
2. The last node (literal) of the branch stores the weight of the description. The weight corresponds to the error of the weak learner that was constructed from this description ($\mathfrak{w}$).
3. If a literal already exists during creation of the branch, we increase the weight of the existing leaf by the weight of the literal, because the description was used more heavily (i.e. by multiple classifiers). We proceed the creation of the branch using the existing path.

After tree construction, we merge sibling leaf nodes defined on the same attribute that originate from the same boosting iteration. For instance, two sibling leaf nodes "age $<$ 20" and "age $<$ 30" can be merged into a single leaf node "age $<$ 30", since all descriptions from the same round are boosted together.

Figure 1 shows the descriptions encountered in a run of the BoostEMM process. The size of the nodes represents the weight, indicating the degree to which the constraint contributes to the selection of samples for boosting.

4 Experiments

Three datasets with a binary classification task were used for the experiments (cf. Table 1). The well-known Adults dataset stems from the UCI repository [19];

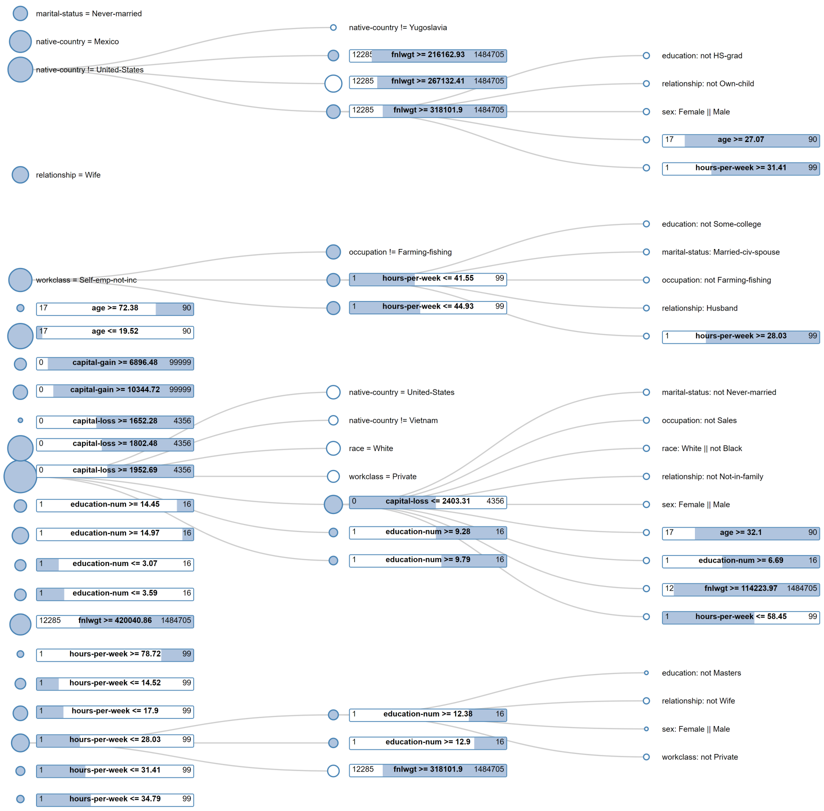
marital-status = Never-married
native-country = Mexico
native-country != United-States
relationship = Wife
workclass = Self-emp-not-inc
17 age >= 72.38 90
17 age <= 19.52 90
0 capital-gain >= 6896.48 99999
0 capital-gain >= 10344.72 99999
0 capital-loss >= 1652.28 4356
0 capital-loss >= 1802.48 4356
0 capital-loss >= 1952.69 4356
1 education-num >= 14.45 16
1 education-num >= 14.97 16
1 education-num <= 3.07 16
1 education-num <= 3.59 16
12285 fnlwgt >= 420040.86 1484705
1 hours-per-week >= 78.72 99
1 hours-per-week <= 14.52 99
1 hours-per-week <= 17.9 99
1 hours-per-week <= 28.03 99
1 hours-per-week <= 31.41 99
1 hours-per-week <= 34.79 99
native-country != Yugoslavia
12285 fnlwgt >= 216162.93 1484705
12285 fnlwgt >= 267132.41 1484705
12285 fnlwgt >= 318101.9 1484705
occupation != Farming-fishing
1 hours-per-week <= 41.55 99
1 hours-per-week <= 44.93 99
native-country = United-States
native-country != Vietnam
race = White
workclass = Private
0 capital-loss <= 2403.31 4356
1 education-num >= 9.28 16
1 education-num >= 9.79 16
1 education-num >= 12.38 16
1 education-num >= 12.9 16
12285 fnlwgt >= 318101.9 1484705
education: not HS-grad
relationship: not Own-child
sex: Female || Male
17 age >= 27.07 90
1 hours-per-week >= 31.41 99
education: not Some-college
marital-status: Married-civ-spouse
occupation: not Farming-fishing
relationship: Husband
1 hours-per-week >= 28.03 99
marital-status: not Never-married
occupation: not Sales
race: White || not Black
relationship: not Not-in-family
sex: Female || Male
17 age >= 32.1 90
1 education-num >= 6.69 16
12 fnlwgt >= 114223.97 1484705
1 hours-per-week <= 58.45 99
education: not Masters
relationship: not Wife
sex: Female || Male
workclass: not Private

Fig. 1. Inspection of descriptions used for boosting (size indicates ensemble weight $\mathfrak{w}_i$). Descriptions correspond to a conjunction of literals, from root to leaf.

the positive class is high earners. Credit-card fraud [2] can be found at Kaggle; the task is to detect fraudulent transactions. Metadata for both these datasets is readily available online. The third dataset, however, is proprietary: Rabobank provided an anonymized real-life dataset related to on-line fraud.

Rabobank is a financial institution, one of the three biggest banks in the Netherlands. Rabobank supplies a full range of financial products to its customers, including internet and mobile banking. The dataset we work with encompasses over 30 million samples of internet transactions and mobile payments, performed from January 2016 up to February 2017. Most of these samples represent genuine, non-fraudulent transactions, but a tiny fraction (∼2 000) were manually marked as fraudulent by domain experts. Known types of fraud include trojan attacks, phishing and ID takeover. For trojan attacks and phishing, the client takes part in the fraudulent transaction by providing the two-factor authentication to the bank, after being misled by the fraudster to do so on a payment prepared by the fraudster. In ID takeover, the fraudster has stolen the

Table 1. Datasets, with both original and 'preprocessed' feature sets used in practice.

Dataset	#orig. attr.	#prep. attr.	#train Neg./Pos.	#test Neg./Pos.
Adults	14	104	29 741/9 332	7 414/2 355
Credit-card	30	29	227 451/394	56 864/98
Rabobank	2487	200	2 015/385	513/87

credentials of the client and is able to provide the authentication herself. Typically, different attackers and attack types occur in the same period. As attacks are being blocked the modus operandi is changed or renewed within days or weeks. Old attacks are retried over time, new attack vectors show up.

Each record consists of a timestamp, an identification hash, a binary label to indicate fraud, and 1 013 anonymized features. Attributes are masked by renaming. Algorithmically each attribute is inspected; if a sample contains not more then 200 unique values it is considered to be a code which is recoded to a numeric value. Numeric and text values are frequency-based transformed into up to 801 bins. Base attributes are constructed from the current transaction, as well as the history from accountholder and beneficiary. Aggregations found to be useful for the current business rule system were added.

The task in the Rabobank dataset is to predict transactions to be fraudulent (label=1) or not (label=0), having learned from historical data only. In order to be useful alongside the current fraud detection, the bank requires the FPR to be stable and far less then 1:10 000.

4.1 Experimental Setup

All datasets are imbalanced: there are substantially more negative records than positive ones. Since we build on scikit-learn's Python Decision Tree implementation as weak learner, we use dummy variables to handle categorical variables in the Adult dataset. For the Credit-card and Rabobank dataset, all the values were given as numeric in the first place. We discard the 'Time' column in the Credit-card dataset. The beam search algorithm for EMM [5, Algorithm 1] is parametrized with search width $w = 3$, search depth $d = 3$, and incorporates the top-q subgroups into BoostEMM with $q = 6$. We use an AdaBoost implementation with decision stumps (i.e. decision trees with depth 1).

4.2 Experimental Results

We run comparative experiments with seven competitors; results can be found in Table 2. The models are Straw Man (majority class), AdaBoost SAMME [10], AdaBoost SAMME.R [11], and BoostEMM with each of the target interaction exceptionality components (cf. Section 3.3). For each competitor we report accuracy evaluated on a withheld test set. Since all datasets are imbalanced, we also report Kappa and AUC. For all measures, higher is better.

Since Adults is the only non-anonymized dataset, we present descriptions discovered during the BoostEMM training for a qualitative inspection. Subgroups

Table 2. Results of comparative experiments.

ML model	Adults			Credit-card			Rabobank		
	Acc.	κ	AUC	Acc.	κ	AUC	Acc.	κ	AUC
Straw Man	0.759	0.0	0.5	0.998	0.0	0.5	0.855	0.0	0.5
AdaBoost SAMME	0.855	0.566	0.757	0.999	0.758	0.852	0.917	0.647	0.808
AdaBoost SAMME.R	0.869	0.616	0.787	0.999	0.815	0.883	0.91	0.623	0.799
BoostEMM φ_{err}	0.808	0.289	0.606	0.999	0.310	0.592	0.9	0.442	0.66
BoostEMM φ_{κ}	0.759	0.0	0.5	0.999	0.601	0.735	0.855	0.0	0.5
BoostEMM φ_{FNR}	0.241	0.0	0.241	0.999	0.598	0.907	0.145	0.0	0.5
BoostEMM φ_{FPR}	0.759	0.0	0.5	0.999	0.234	0.566	0.855	0.0	0.5

Table 3. Top subgroups found by BoostEMM φ_{err} in first five iterations (Adults).

Iter.	D	$\lvert G_D \rvert$	TN	FP	FN	TP
1	capital-gain $\geq 6896.48 \wedge$ age $\geq 19.52 \wedge$ education_7th-8th $\neq 1$	1632	17	0	1615	0
2	capital-loss $\geq 1802.48 \wedge$ marital-status_Married-civ-spouse $= 1 \wedge$ capital-loss ≤ 1952.69	493	16	0	477	0
3	marital-status_Married-civ-spouse $= 1 \wedge$ education-num $\leq 7.72 \wedge$ capital-gain ≤ 3448.24	1847	0	1662	0	185
4	capital-loss $\geq 1802.48 \wedge$ education-num $\geq 7.72 \wedge$ age ≥ 27.07	1041	240	0	801	0
5	capital-gain $\geq 6896.48 \wedge$ age $\geq 19.52 \wedge$ education_7th-8th $\neq 1$	1632	17	0	1615	0

that are deemed most problematic by the four compound quality measures in the first five iterations of the BoostEMM process can be found in Tables 3–6.

5 Discussion

Table 2 shows that BoostEMM can sometimes match the performance of AdaBoost, but sometimes it does not do so well. As expected, φ_{err} mimics AdaBoost best in terms of pure performance; it barely loses accuracy on the Rabobank and Credit-card datasets in comparison with AdaBoost, while it has to cede some ground on the Adults dataset. Interestingly, while BoostEMM with φ_{FNR} leads to substantial accuracy loss in two of the datasets, it performs unexpectedly well on the third. All methods have a high accuracy on the Credit-card dataset, but in terms of AUC, φ_{FNR} outperforms all other methods including AdaBoost.

When we inspect subgroups in more detail (cf. Tables 3–6), we obtain more transparency and hence accountability into the boosting process. This transparency is augmented by the visualization as introduced in Figure 1. Additionally, from Table 6 we learn that BoostEMM suffers from a familiar problem in data mining. This table follows the process of boosting subgroups featuring an unusually high False Postive Rate. As the table shows, the top subgroups found in the first five iterations have an undefined FPR: there are no positives in these subgroups at all. This is caused by the first weak learner assigning all records to the majority class, which is negative: the process only features true and false negatives! In this setting, FPR boosting makes no sense. Therefore, in

Table 4. Top subgroups found by BoostEMM φ_κ in first few iterations (Adults).

Iter.	D	$\lvert G_D \rvert$	TN	FP	FN	TP
1	marital-status_Separated $\neq 1 \wedge$ native-country_Holand-Netherlands $\neq 1$	37824	28562	0	9262	0
2	occupation_Prof-specialty $\neq 1 \wedge$ native-country_Holand-Netherlands $\neq 1$	34088	26982	0	7106	0
3	occupation_Other-service $\neq 1 \wedge$ native-country_Holand-Netherlands $\neq 1$	35157	25992	0	9165	0
4	occupation_Adm-clerical $\neq 1 \wedge$ native-country_Holand-Netherlands $\neq 1$	34592	25858	0	8734	0
5	occupation_Farming-fishing $\neq 1 \wedge$ native-country_Holand-Netherlands $\neq 1$	37892	28695	0	9197	0

Table 5. Top subgroups found by BoostEMM φ_{FNR} in first few iterations (Adults).

Iter.	D	$\lvert G_D \rvert$	TN	FP	FN	TP
1	capital-gain $\geq 6896.48 \wedge$ age $\geq 19.52 \wedge$ education_7th-8th $\neq 1$	1632	17	0	1615	0
2	capital-loss $\geq 1802.48 \wedge$ marital-status_Married-civ-spouse $= 1 \wedge$ capital-loss ≤ 1952.69	493	16	0	477	0
3	capital-gain $\geq 24137.69 \wedge$ marital-status_Married-civ-spouse $\neq 1 \wedge$ age ≥ 19.52	98	1	0	97	0
4	capital-loss $\geq 1802.48 \wedge$ marital-status_Married-civ-spouse $= 1 \wedge$ age > 27.07	887	133	0	754	0
5	capital-gain $\geq 24137.69 \wedge$ marital-status_Married-civ-spouse $\neq 1 \wedge$ age ≥ 19.52	98	1	0	97	0

future work, we plan to tackle this problem by dovetailing the various kinds of boosting BoostEMM has to offer.

A similarly detailed investigation as the on in Tables 3–6 has been made for the Rabobank dataset. Here, the attribute names are all obfuscated; we find them in the form C_0010. However, we presented the resulting subgroups in such tables to domain experts at Rabobank who possess the key to translate obfuscated features back to real-life information. They reported back that the subgroups focus on the historical behavior of the customer of counterparty. Subgroups reported in the first iteration make the initial, rough cut. Subgroups reported in the second iteration give it more detail towards a specific modus operandi. Client confidentiality disallows us to discuss more details about these subgroups, but the domain experts confirm that the problematic areas have clear meaning to them, which provides us with confidence that BoostEMM indeed adds the desired transparency to the boosting process.

References

1. S.D. Bay, M.J. Pazzani. Detecting Change in Categorical Data: Mining Contrast Sets. Proc. KDD, pp. 302–306, 1999.
2. A. Dal Pozzolo, O. Caelen, R.A. Johnson, G. Bontempi. Calibrating Probability with Undersampling for Unbalanced Classification. Proc. SSCI, pp. 159–166, 2015.
3. G. Dong, J. Li. Efficient Mining of Emerging Patterns: Discovering Trends and Differences. Proc. KDD, pp. 43–52, 1999.
4. G. Dong, K. Ramamohanarao. Enhancing Traditional Classifiers Using Emerging Patterns. In: G. Dong, J. Bailey (eds.): Contrast Data Mining: Concepts, Algorithms, and Applications, pp. 187–196, CRC Press, 2013.
5. W. Duivesteijn, A.J. Feelders, A. Knobbe. Exceptional model mining. Data Mining and Knowledge Discovery 30(1):47–98, 2016.
6. W. Duivesteijn, A. Knobbe, A. Feelders, M. van Leeuwen. Subgroup Discovery meets Bayesian networks — an Exceptional Model Mining approach. Proc. ICDM, pp. 158–167, 2010.

Table 6. Top subgroups found by BoostEMM φ_{FPR} in first few iterations (Adults).

Iter.	D	$\|G_D\|$	TN	FP	FN	TP
1	native-country_Holand-Netherlands $\neq 1$	39073	29741	0	9332	0
2	native-country_Holand-Netherlands $\neq 1$	39073	29741	0	9332	0
3	native-country_Holand-Netherlands $\neq 1$	39073	29741	0	9332	0
4	native-country_Holand-Netherlands $\neq 1$	39073	29741	0	9332	0
5	native-country_Holand-Netherlands $\neq 1$	39073	29741	0	9332	0

7. W. Duivesteijn, E. Loza Mencía, J. Fürnkranz, A.J. Knobbe. Multi-label LeGo — Enhancing Multi-label Classifiers with Local Patterns. Proc. IDA, pp. 114–125, 2012.
8. W. Duivesteijn, J. Thaele. Understanding Where Your Classifier Does (Not) Work — The SCaPE Model Class for EMM. Proc. ICDM, pp. 809–814, 2014.
9. European Parliament. Resolution of 16 February 2017 with recommendations to the Commission on Civil Law Rules on Robotics (2015/2103(INL)). http://www.europarl.europa.eu/sides/getDoc.do?pubRef=-//EP//TEXT+TA+P8-TA-2017-0051+0+DOC+XML+V0//EN [accessed July 3, 2017], 2017.
10. Y. Freund, R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1):119–139, 1997.
11. J. Friedman, T. Hastie, R. Tibshirani. Additive logistic regression: a statistical view of boosting. Annals of Statistics 28(2):337–407, 2000.
12. A. Henelius, K. Puolamäki, H. Boström, L. Asker, P. Papapetrou. A peek into the black box: exploring classifiers by randomization. Data Mining and Knowledge Discovery 28(5–6):1503–1529, 2014.
13. A. Henelius, K. Puolamäki, I. Karlsson, J. Zhao, L. Asker, H. Boström, P. Papapetrou. GoldenEye++: A Closer Look into the Black Box. Proc. SLDS, pp. 96–105, 2015.
14. F. Herrera, C.J. Carmona, P. González, M.J. del Jesús. An overview on subgroup discovery: foundations and applications. Knowledge and Information Systems 29(3):495–525, 2011.
15. W. Klösgen. Explora: A Multipattern and Multistrategy Discovery Assistant. Advances in Knowledge Discovery and Data Mining, pp. 249–271, 1996.
16. A. Knobbe, B. Crémilleux, J. Fürnkranz, M. Scholz. From Local Patterns to Global Models: The LeGo Approach to Data Mining. Proc. LeGo: From Local Patterns to Global Models workshop @ ECML/PKDD, pp. 1–16, 2008.
17. P. Kralj Novak, N. Lavrač, G.I. Webb. Supervised Descriptive Rule Discovery: A Unifying Survey of Contrast Set, Emerging Pattern and Subgroup Mining. Journal of Machine Learning Research 10:377–403, 2009.
18. D. Leman, A. Feelders, A.J. Knobbe. Exceptional Model Mining. Proc. ECML/PKDD (2), pp. 1–16, 2008.
19. M. Lichman, UCI Machine Learning Repository, http://archive.ics.uci.edu/ml, University of California, Irvine, School of Information and Computer Sciences, 2013.
20. P.M. Long, R.A. Servedio. Random classification noise defeats all convex potential boosters. Machine Learning 78(3):287-304, 2010.
21. S. Wrobel. An Algorithm for Multi-relational Discovery of Subgroups. Proc. PKDD, pp. 78–87, 1997.