Technical Report

Understanding Where Your Classifier Does (Not) Work — the SCaPE Model Class for Exceptional Model Mining

Wouter Duivesteijn, Julia Thaele

09/2014

technische universität
dortmund

SFB 876 Verfügbarkeit von
Information durch Analyse unter
Ressourcenbeschränkung

Part of the work on this technical report has been supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876 "Providing Information by Resource-Constrained Analysis", project C3.

Speaker: Prof. Dr. Katharina Morik
Address: TU Dortmund University
Joseph-von-Fraunhofer-Str. 23
D-44227 Dortmund
Web: http://sfb876.tu-dortmund.de

Understanding Where Your Classifier Does (Not) Work — the SCaPE Model Class for Exceptional Model Mining

Wouter Duivesteijn
Fakultät Informatik, LS VIII
Lehrstuhl für Künstliche Intelligenz
Technische Universität Dortmund, Germany
wouter.duivesteijn@tu-dortmund.de

Julia Thaele
Fakultät Physik, LS E5B
Lehrstuhl für Astroteilchenphysik
Technische Universität Dortmund, Germany
julia.thaele@tu-dortmund.de

Abstract—**FACT, the First G-APD Cherenkov Telescope, detects air showers induced by high-energetic cosmic particles. It is desirable to classify a shower as being induced by a gamma ray or a background particle. Generally, it is nontrivial to get any feedback on the real-life training task, but we can attempt to understand how our classifier works by investigating its performance on Monte Carlo simulated data. To this end, in this paper we develop the SCaPE (Soft Classifier Performance Evaluation) model class for Exceptional Model Mining, which is a Local Pattern Mining framework devoted to highlighting unusual interplay between multiple targets. In our Monte Carlo simulated data, we take as targets the computed classifier probabilities and the binary column containing the ground truth: which kind of particle induced the corresponding shower. Using a newly developed quality measure based on ranking loss, the SCaPE model class highlights subspaces of the search space where the classifier performs particularly well or poorly. These subspaces arrive in terms of conditions on attributes of the data, hence they come in a language a domain expert understands, which should aid him in understanding where his/her classifier does (not) work. Additional experiments are carried out on nine UCI datasets. Found subgroups highlight subspaces whose difficulty for classification is corroborated by astrophysical interpretation, as well as subspaces that warrant further investigation.**

Keywords—Astrophysics, Exceptional Model Mining, Cherenkov radiation, soft classifier.

N.B.: A condensed version of this paper will appear as a short paper at the 14th IEEE International Conference on Data Mining [1].

I. Introduction

The FACT telescope ([2], [3]) is an Imaging Air Cherenkov Telescope, designed to detect light emitted by secondary particles, generated by high-energetic cosmic particles interacting with the atmosphere of the Earth. For astrophysical reasons, it is important to classify the light as resulting from the atmosphere being hit by a gamma ray or a proton; the latter occur much more frequently, but the former are the more interesting in gamma astronomy (which will be discussed later in the paper). Currently, one of the used classifiers is a random forest, whose performance needs our detailed attention.

The problem with training a classifier on real astrophysical data is that there is no clear feedback. Based on the observed light, we could deduce whether the inducing particle is a gamma ray or a proton. Then, we can look in the direction from which the particle originated, and strive to find an astrophysical source generating gamma rays. But even if we find such a source, there is no certain way of telling what kind of particle induced the original observation. Effectively, we are dealing with a feedbackless learning task, and it is typically hard to finetune a classifier without feedback.

To study our learning performance, we turn to Monte Carlo data. We simulate particle interactions with the atmosphere, as well as reflections of the resulting Cherenkov light with telescope mirrors on the one hand and the FACT camera electronics on the other hand. Thus, we can simulate the images in the FACT camera with known parameters, including the type of inducing particle. This gives us a dataset of camera images that is equivalent in form to a dataset we would get from real astrophysical observations, except that we also know the true label of our classification task. By training our random forest on this dataset, we obtain the soft classifier probabilities for each record. Through studying the interaction between the binary ground truth that we already knew and the soft classifier probabilities we learned from the data, we can understand where our classifier performs exceptionally well or exceptionally poorly.

We study this interaction with an Exceptional Model Mining (EMM) ([4], [5]) approach. This is a Local Pattern Mining framework, specialized in finding coherent subsets of the dataset where multiple targets interact in an unusual way. In this paper, we introduce the SCaPE (Soft Classifier Performance Evaluation) model class for EMM, seeking subgroups for which a soft classifier represents a ground truth exceptionally well or poorly. This should allow a domain expert to understand where his/her classifier does (not) work.

II. Preliminaries

Before we can introduce the new contributions of this paper, we need to cover a lot of preliminary ground. The preliminaries have been split up into three parts: an introduction of astrophysical concepts, a short note on the alignment of soft and hard classifiers, and a summary of Local Pattern Mining methods including EMM. Feel free to skip the corresponding subsections if you are familiar with these fields.

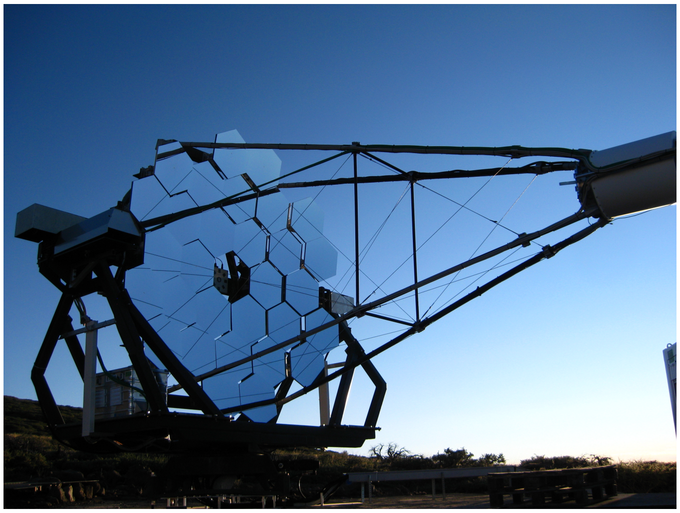


Fig. 1. The FACT telescope.

A. The FACT Telescope

An important task in astroparticle physics is observing distant astrophysical sources such as Supernova Remnants (SNR) or Active Galactic Nuclei (AGN) in multiple energy ranges (optical, radio, X-ray, gamma rays), since combining such observations helps us understand (amongst others) the cosmic particle acceleration and radiation emission mechanisms of these sources [2]. Each energy range demands different detector techniques, hence dedicated telescopes are required. In the high-energy regime, we are interested in (ultra-)relativistic cosmic particles such as gamma rays, neutrinos, and protons, which are assumed to be accelerated by astrophysical sources (such as SNR and AGN). Gamma rays are interesting because of their neutral electric charge, which causes them to travel undeflected by intergalactic magnetic fields. This means that the direction from which the primary gamma rays are coming, necessarily points directly to the astrophysical source.

The Earth's atmosphere is only transparent in optical and radio wavelengths. This prohibits observing low-energy gamma rays on Earth, but we can make these observations with dedicated satellites. Since the flux of gamma rays, which is the amount of particles per area and time, decreases with higher energy, the detection of gamma rays in higher energy ranges would require either a bigger detection area (in the satellite) or more time. Both solutions are not satisfying, as time and bigger satellites are prohibitively cost-intensive. Instead, we can exploit an effect caused by the, otherwise detrimental, passage through the atmosphere a particle makes, which allows to observe very high-energy (VHE) gamma rays by ground-based telescopes.

When very high-energetic cosmic particles such as gamma rays and protons interact with the atmosphere of the Earth, they induce an extensive air shower consisting of secondary relativistic particles, which can be charged. The charged particles emit Cherenkov radiation [6], a blueish light which can be detected by Imaging Air Cherenkov Telescopes (IACT). One such telescope is FACT, the First G-APD Cherenkov Telescope (cf. Figure 1). It is located on La Palma, Canary Islands, Spain at 2200m above sea level, and is operational since October 2011 [3]. With a total reflective surface of $9.5\,\mathrm{m}^2$, it is a rather small telescope. FACT is the first IACT using Geiger-mode Avalanche PhotoDiodes (G-APD) (also known as silicon photomultipliers) as photosensors to detect Cherenkov light. Contrary to conventional detector techniques of IACTs, G-APDs allow to observe even during strong moonlight and thus increase the effective observation time. This is especially interesting for source detection by small telescopes, but also very important for long-term monitoring of sources.

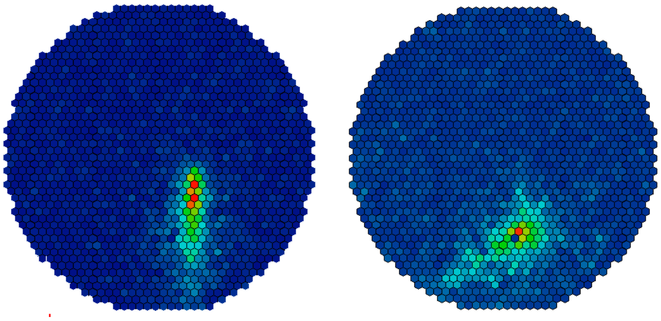


Fig. 2. Camera images of real air showers. The distinctive shapes of different showers helps to classify gamma- and proton-induced showers. The color scale corresponds to the amount of the detected Cherenkov light in each pixel.

As we observe a variability in the gamma ray flux of sources in multiple timescales (both seconds and years) [3], long-term monitoring is required to understand the emission procedures and mechanisms within and surrounding the sources. The primary physics goal of FACT is therefore to observe the brightest known VHE sources on long timescales, which becomes realizable by using G-APDs.

The common method in gamma astronomy to evaluate analysis methods is to compare them with existing methods on data from an astrophysical source called the Crab Nebula. In the year 1054, Chinese astronomers observed a Supernova explosion in the sky. Today, we see the remnant of this explosion as the Crab Nebula. This SNR is one of the brightest sources in gamma ray astronomy. With a distance of about 6,500 light years to Earth it is located near us and inside the Milky Way, making it relatively easy to observe. In 1989 the Whipple Observatory detected the first VHE gamma rays from the Crab Nebula [7]. Since then the Nebula has been studied in detail: the flux of gamma rays from the Crab Nebula was carefully observed on a long timescale. The Crab Nebula is an ideal candidate to calibrate the stability of the analysis of other active sources, since nearly no variability was measured in the flux of HE and VHE gamma rays [8] and serves as a 'standard candle' in gamma astronomy.

The main goal of the analysis method whose results are evaluated in this paper is to find gamma-induced showers. Unfortunately, for the brightest sources, proton showers appear a thousand to ten thousand times more frequently than gamma showers in the source direction [9], which makes the light of the proton-induced showers the biggest background. Therefore, the separation of gamma- and proton-induced showers is very important to be able to detect a source, to increase the sensitivity of the telescope and thus the effective observation time, and finally to measure the spectrum of the source. For the separation, Monte Carlo simulations are necessary, which simulate shower images in the FACT camera with known parameters, such as type and energy of the primary particle that induced the shower. The first step is to simulate particle interactions in the atmosphere and the emission of Cherenkov light with the program MMCS based on CORSIKA [10]. Further processing by a simulation and analyzing tool called MARS [11] includes simulating the reflection of the light on the mirrors of the telescope and the electronics inside the camera. We end up with simulated camera images containing gamma and proton showers. From these camera images the image parameters of the showers are reconstructed. Since gamma- and proton-induced showers have distinctive shapes (cf. Figure 2), the image parameters describing the properties of the shower images are used to distinguish between them. As the information of the primary particles is known in the simulation, the data are labeled as *true* or 1 for gamma showers (signal) and *false* or 0 for proton showers (background).

As it is commonly done in IACT experiments, for instance in the MAGIC [12] and H.E.S.S. [13] experiments, the separation is done with a random forest (RF) algorithm [14]. We employ an implementation available within the RapidMiner analytics platform [15], in particular the one that is equivalent to the implementation available in the WEKA [16] machine learning software. The RF builds a model with the image parameters of the labeled simulated data and tests it on the remaining dataset in a five-fold cross-validation to ensure a stable classification. For this dataset 500 trees were grown, each considering a random subset of 8 out of the 11 available attributes. These 11 attributes contain parameter distributions for gamma and proton showers, which are known to be crudely separable by simple cuts on each parameter relatively successfully. The fact that just a subset of attributes is drawn contributes to the randomized trees needed for a good random forest. Notice that this whole procedure is still under development; these settings do not necessarily represent the final settings of the separation. Each tree classifies an event (one shower) as 1 for signal or 0 for background. Prediction aggregation over all trees is done by averaging, and expressed by the *Signalness*:

$$S = \frac{1}{n_{\text{trees}}} \sum_{i=1}^{n_{\text{trees}}} S_i \qquad \text{with } S_i \in \{0, 1\}$$

This quantity describes the probability or the confidence of the RF for an event to be classified as a gamma shower. For the given FACT dataset the efficiency decreases with a higher Signalness value, but at the same time the purity increases. To separate gamma and proton showers sufficiently while not losing too much data, a cut has to be found which fulfills both conditions and depends on the physics task.

B. Soft and Hard Classifiers

Suppose that we have a binary classification problem: any record of the dataset belongs to exactly one of the two available classes. Let us denote those classes by 0 and 1. For such a problem, two particular types of classification algorithms can be distinguished. On the one hand, a *hard classifier* outputs for each record in the test set a decision to which class it thinks the record belongs: the output is either 0 or 1. On the other hand, a *soft classifier* outputs for each record in the test set a real-valued number, typically a probability: the output can be any value in $\mathbb{R}$, and higher values for the output correspond to a higher confidence that the records should be assigned class 1. These two types of classifiers stem from different philosophies; both have their merits and drawbacks.

A soft classifier can be turned into a hard classifier by thresholding. Denote the real-valued soft classifier output for the i^{th} record by r^i. If we choose any value $v \in \mathbb{R}$, we can convert the soft classifier into a hard classifier by setting our output for the i^{th} record to $\mathbb{I}\left\{r^i > v\right\}$. Here, $\mathbb{I}$ denotes the indicator function, which is equal to 1 if its argument is true and 0 otherwise. In other words, records to which the soft classifier assigns a value higher than v are assigned class 1, and all other records are assigned class 0. By varying v, we can generate as many different hard classifiers from the soft classifier as the number of distinct values for r^i.

Notice that in this process, the *ordering* imposed on the records by the soft classifier outputs is much more important than their actual *values*. Suppose that record x^1 has a higher soft classifier output than record x^2. If the threshold value causes x^2 to be assigned class 1, it will also cause x^1 to be assigned class 1. Conversely, if it causes x^1 to be assigned class 0, is will also cause x^2 to be assigned class 0. This behavior does not depend on how far the soft classifier outputs are apart: rather, their ranking enforces these relations on the hard classifier behavior. We will mirror this emphasis on ordering when we develop our new quality measure, in Section V-B.

In the FACT telescope simulation data, we have the information of the primary particles, which can be viewed as the predictions of a perfect hard classifier. We also learn the signalness of each record, which can be viewed as the output of a soft classifier. We will investigate unusual interplay between these classifiers in an Exceptional Model Mining setting.

C. Exceptional Model Mining

Pattern mining ([17], [18]) is the broad subfield of data mining where only a part of the data is described at a time, ignoring the coherence of the remainder. One class of pattern mining problems is *theory mining* [19], whose goal is finding subsets S of the dataset Ω that are interesting somehow:

$$S \subseteq \Omega \quad \Rightarrow \quad \text{interesting}$$

Typically, not just any subset of the data is sought after: only those subsets that can be formulated using a predefined *description language* $\mathcal{L}$ are allowed. A canonical choice for the description language is conjunctions of conditions on attributes of the dataset. If, for example, the records in our dataset describe people, then we can find results of the following form:

$$\text{Age} \geq 30 \wedge \text{Smoker} = \text{yes} \quad \Rightarrow \quad \text{interesting}$$

Allowing only results that can be expressed in terms of attributes of the data, rather than allowing just any subset, ensures that the results are relatively easy to interpret for a domain expert: the results arrive at his doorstep in terms of quantities with which he should be familiar. A subset of the dataset that can be expressed in this way is called a *subgroup*.

In the best-known form of theory mining, *frequent itemset mining* [20], the interestingness of a pattern is gauged in an unsupervised manner. Here, the goal is to find patterns that occur unusually frequently in the dataset:

$$\text{Age} \geq 30 \wedge \text{Smoker} = \text{yes} \quad \Rightarrow \quad \text{(high frequency)}$$

In the FACT telescope setting, however, we strive to separate the gamma sources from the proton sources; there is a clear target, hence this setting is supervised. The most extensively studied form of supervised theory mining is known as *Subgroup Discovery* (SD) [21], where one (typically binary) attribute t of the dataset is singled out as the *target*. The goal is to find subgroups for which the distribution of this target is unusual: if the target describes whether the person develops lung cancer or not, we find subgroups of the following form:

$$\begin{aligned} \text{Smoker} = \text{yes} \quad &\Rightarrow \quad \text{lung cancer} = \text{yes} \\ \text{Age} < 30 \quad &\Rightarrow \quad \text{lung cancer} = \text{no} \end{aligned}$$

Exceptional Model Mining (EMM) ([4], [5]) can be seen as the multitarget generalization of SD. Rather than singling out one attribute as the target t, in EMM there are several target attributes $t_1, \ldots, t_m$. Interestingness is not merely gauged in terms of an unusual *marginal* distribution of t, but in terms of an unusual *joint* distribution of $t_1, \ldots, t_m$. Typically, a particular kind of unusual *interaction* between the targets is captured by the definition of a *model class*, and subgroups are deemed interesting when their model is exceptional, which is captured by the definition of a *quality measure*. For example, suppose that there are two target attributes: a person's length (t_1), and the average length of his/her grandparents (t_2). We may be interested in the correlation coefficient between t_1 and t_2; we then say we study EMM with the *correlation model class* [4]. Given a subset $S \subseteq \Omega$, we can estimate the correlation between the targets within this subset by the sample correlation coefficient. We denote this estimate by r^S. Now we can define the following quality measure (tweaked from [4]):

$$\varphi(S) = \left| r^S - r^\Omega \right|$$

EMM then strives to find subgroups for which this quality measure has a high value: effectively, we search for subgroups coinciding with an exceptional correlation between a person's length and his/her grandparents' average length:

$$\text{Lives near nuclear plant} = \text{yes} \quad \Rightarrow \quad \left| r^S - r^\Omega \right| \text{ is high}$$

III. Related Work

Previous work exists on discovering subgroups displaying unusual interaction between multiple targets, for instance in the previously developed model classes for EMM: correlation, regression, Bayesian network, and classification (cf. ([4], [5]), for the Bayesian network model class see also [22]). The classification model class is particularly related to the SCaPE model class, with two major differences. On the one hand, the model class definitions imply a different relation between the subgroup definitions and classifier search space. The classification model class takes both classifier input and output attributes as targets for the EMM run. This disallows those attributes to show up in the descriptions of subgroups found with EMM; exceptional subgroups are described in terms of attributes unavailable to the classifier. By contrast, in the SCaPE model class, all attributes available as input (but not as output!) to the classifier are also available for describing subgroups. Hence, the found unusual subgroups directly correspond to a subspace in the classifier search space. On the other hand, the model classes search for a different underlying concept in the dataset. The classification model class *investigates* classifier *behavior* in the *absence* of a ground truth. The SCaPE model class *evaluates* classifier *performance* in the *presence* of a ground truth. Hence, the two model classes are different means to achieve different ends. The other existing model classes for EMM, including the Bayesian network model class [22], have in common with the SCaPE model class that they find subgroups displaying exceptional interplay between targets. Either the amount of involved targets, or the type of interplay that is being detected (or both) is vastly different from the exceptional interplay being gauged by the SCaPE model class. Hence the resulting subgroups found with these model classes are incomparable to each other, and to those found with the SCaPE model class.

Local Pattern Mining tasks that are similar to SD are Contrast Set Mining [23] and Emerging Pattern Mining [24]. Both these tasks do not consider multiple target attributes simultaneously, and do not directly model unusual interactions. Explicitly seeking a deviating model over a target is performed in Distribution Rules [25], where there is only one numeric target, and the goal is to find subgroups on which the target distribution over the entire target space is the least fitting to the same distribution on the whole dataset. This can be seen as an early instance of EMM with only one target. However, there is no multi-target interaction. Umek et al. [26] do consider SD with multiple targets. They approach the attribute partition in the reverse way of EMM: candidate subgroups are generated by agglomerative clustering on the targets, and predictive modeling on the descriptors strives to find matching descriptions. This work does not allow freely expressing when target interaction is unusual. Redescription Mining [27] seeks multiple descriptions inducing the same subgroup. This models unusual interplay, but on the descriptor space rather than the target space. Furthermore, none of this work concerns explicit evaluation of a classifier.

Automated guidance to improve a classifier has been studied in the data mining subfield of meta-learning. The exact meaning of this term is subject to debate; see [28] for a survey discussing some of the views. A constant factor is that meta-learning hovers around the question how knowledge about learning can be put to use to improve the performance of a learning algorithm. A typical approach is to let the machine compute meta-features characterizing the data, such as correlations between attributes, attribute entropy, and mutual information between class and attributes. These meta-features are then considered in a new classifier training phase, and the hope is that this improves predictive performance. This process is depicted in the self-adaptive learning flow diagram in [28, Figure 2]. The meta-features can also be employed to compare learning algorithms. For instance, Henery [29] provides a set of rules to determine when the one learning algorithm is significantly better than the other. However, in almost all of the existing meta-learning work, the focus is on letting the machine learn how the machine can perform better.

By contrast, Vanschoren and Blockeel [30] express an interest in *understanding* learning behavior. Their paper discusses a descriptive form of meta-learning, proposing an integrated solution (using experiment databases) that aims to explain the behavior of learning algorithms. This explanation is again expressed in terms of meta-features; no investigation takes place of particular subspaces of the search space on which the algorithm performs exceptionally. While Vilalta and Drissi [28, Section 4.3.1] do devote a subsubsection to "Finding regions in the feature space [. . .]", this is again in the context of algorithm selection. Their innovation lies in allowing different learning algorithms for different records of the dataset. Meta-learning is related to the goals we strive to achieve with the SCaPE model class for EMM, but two things set these approaches apart: meta-learning focuses on meta-features, while the SCaPE model class focuses on coherent subspaces of the original search space, and meta-learning focuses on letting the machine improve the predictive performance of the machine, while the SCaPE model class focuses on providing *understanding* to the domain expert where his/her classifier works well or fails. As such, the SCaPE model class for EMM provides progress on the path sketched by Vanschoren and Blockeel in the conclusions of their paper [30, Section 5]: "We hope to advance toward a meta-learning approach that can explain not only *when*, but also *why* an algorithm works or fails [. . .]".

A very recent first inroad towards peeking into the classifier black box is the method by Henelius et al. [31], who strive to find groups of attributes whose interactions affect the predictive performance of a given classifier. This is more akin to the classification model class for EMM. While Henelius et al. study hard classifiers, the SCaPE model class is designed for soft classifiers.

IV. Main Contribution

The main contribution of this paper is the development of a new model class with associated quality measure for Exceptional Model Mining: the SCaPE (Soft Classifier Performance Evaluation) model class. In this model class, two targets are identified: a binary target b describing the ground truth, and a real-valued target r containing the output of a soft classifier that strives to approximate b. The goal in this model class is to find subgroups for which this soft classifier represents the ground truth exceptionally well or exceptionally poorly. Notice that, SCaPE being an EMM model class, the focus is on easily-interpretable subgroups. Hence, our primary goal is not to let the machine improve the machine, but to let the domain expert *understand* where his/her classifier does or does not work.

V. The SCaPE Model Class for EMM

In the SCaPE model class for EMM, we assume a dataset Ω, which is a bag of N records of the form $x = (a_1, \ldots, a_k, b, r)$. We call $\{a_1, \ldots, a_k\}$ the *descriptive attributes*, or *descriptors*, whose domain is unrestricted. The remaining two attributes, b and r, are the *targets*. The first, b, is the *binary target*; we will denote its values by 0 and 1. The second, r, is the *real-valued target*, taking values in $\mathbb{R}$.

The goal of the SCaPE model class is to find subgroups for which the soft classifier outputs, as captured by r, represent the ground truth, as captured by b. In Section V-A, we develop measures that quantify how well b is represented by r, on the entire dataset and on subsets of the dataset. In Section V-B, we use these measures to define a *quality measure* for the SCaPE model class, that gauges how exceptional the interplay between r and b is on a subgroup when compared to this interplay on the entire dataset. First, however, we need to introduce the following notation and conventions.

TABLE I. Example dataset Ω_E for the ASL illustration

Ω_E	x^1	x^2	x^3	x^4	x^5	x^6	x^7	x^8	x^9
a_1	A	B	C	A	C	B	A	C	C
b	0	1	0	0	1	0	1	1	1
r	0.0	0.1	0.3	0.4	0.5	0.6	0.8	0.9	1.0

If we need to distinguish between particular records of the dataset, we will do so by superscripted indices: x^i is the i^{th} record, b^i is its value for the binary target and a^i_j is its value for the j^{th} descriptor. For the sake of notational convenience, we assume that the records are indexed in non-descending order by their values of r: $i < j \Rightarrow r^i \leq r^j$. We call the records x^i in the dataset for which the binary target is true the *positives*, and the other records the *negatives*.

A. Average (Sub-)Ranking Loss

To explain our reasoning we refer to the toy dataset Ω_E in Table I, featuring nine records consisting of the two targets and only one descriptor. Recall that the binary target contains the ground truth, and the real-valued target contains the output of a soft classifier striving to represent this ground truth.

In Section II-B we have discussed how a soft classifier can be converted into a hard classifier by imposing a threshold at any chosen value v: the predicted label for record x^i is set to 1 if and only if $r^i > v$. One can read from the table that v could be chosen such that the hard classifier based on r lines up reasonably well with the ground truth as provided by b; by and large, high values for the real-valued target coincide with $b = 1$, and low values with $b = 0$. Notice that this capability of r is primarily sensitive not to its precise *values*, but to the *ordering* it implies on the records. Therefore, we capture the alignment of r and b on the whole dataset by the Average Ranking Loss [32]:

$$\mathrm{ARL}(\Omega) = \frac{\sum\limits_{i=1}^{N}\left(\mathbb{1}\left\{b^i = 1\right\} \cdot \sum\limits_{j=i+1}^{N} \mathbb{1}\left\{b^j = 0\right\}\right)}{\sum\limits_{i=1}^{N} \mathbb{1}\left\{b^i = 1\right\}} \tag{1}$$

Essentially, for every positive in the dataset a penalty is computed. The penalty for x^i is equal to the number of negatives x^j that have a higher value for the real-valued target: $r^i < r^j$ (here, the formula for ARL uses the fact that the dataset is ordered non-descendingly by r^i, and conveniently ignores for the moment that two consecutive r-values may be equal). This *ranking loss* is then averaged over all positives in the dataset, arriving at the ARL. Obviously, lower values of the ARL correspond to a better representation of b by r.

For the example dataset in Table I, the penalty assigned to x^7, x^8, and x^9 is 0, since no negatives have a higher value for r than they do. Since x^5 has a lower value for r than x^6, we assign penalty 1 to x^5, and x^2 gets penalty 3 since it has a lower value for r than x^3, x^4, and x^6. Averaging these penalties over all five positives, we arrive at $\mathrm{ARL}(\Omega_E) = 0.8$.

To determine the degree of representation of b by r in a given subgroup S of the dataset, we compute the ARL again, but then restricted to just those records belonging to the subgroup. We call this the *Average Subranking Loss* of S, denoted by $\mathrm{ASL}(S)$. For the example dataset in Table I, we would find $\mathrm{ASL}(a_1 = \mathrm{B}) = 1$; its only positive has a lower value for r than its only negative, so that positive gets a penalty of 1, which is averaged over 1 positive in the subgroup. Similarly, $\mathrm{ASL}(a_1 = \mathrm{A}) = 0$ and $\mathrm{ASL}(a_1 = \mathrm{C}) = 0$, since *within* each of these subgroups, all positives have a higher value for r than all negatives, so all positives get penalty 0.

1) Handling Ties: So far, we have assumed that all values for the real-valued target r in the dataset are distinct. This simplifies the formula in Equation (1), and allows for an easier intuitive explanation in that section. In practice, of course, such an assumption is not necessarily justified and hence undesirable. In this section, we discuss how to handle tied r-values in the dataset. Since we compute the Average (Sub-)Ranking Loss as an average of penalties assigned to all positives, we can focus on how to update the penalty assigned to a positive when its r-value is replicated in the dataset. Suppose that x^i is such a positive: we know that $b^i = 1$ and $r^i = r^j$ for some $j \neq i$. If x^j is also a positive, then the penalty does not need to change: the values for both the real-valued and the nominal target agree, so the relative ranking of these two records is not wrong. If, on the other hand, x^j is a negative, then we should increment the penalty by some amount; a natural choice for this amount can be motivated.

We have two records with the same values for the real-valued target, but opposing values for the binary target. As discussed in Section II-B, a soft classifier can be converted into a hard classifier by thresholding on the soft classifier outputs. Such thresholding cannot distinguish between x^i and x^j since $r^i = r^j$. Hence, whatever threshold value is chosen, these records will *both* be assigned either class 0 or class 1. That means, that *exactly one of these two* records will be misclassified. Since each tie between a positive and a negative will necessarily lead to the misclassification of exactly half the involved records, we will add $1/2$ to the penalty for x^i for each such tie. Incorporating this penalty leads to the following definitions of the ARL and ASL:

Definition (Average (Sub-)Ranking Loss). Given a dataset Ω satisfying the stipulations of Section V, its *Average Ranking Loss*, $\mathrm{ARL}(\Omega)$, is given by:

$$\mathrm{ARL}(\Omega) = \frac{\sum_{i=1}^{N} \mathbb{I}\left\{b^i = 1\right\} \cdot \mathrm{PEN}_i^N(\Omega)}{\sum_{i=1}^{N} \mathbb{I}\left\{b^i = 1\right\}}$$

where the *penalty* for the i^{th} record, $\mathrm{PEN}_i^N(\Omega)$, is given by:

$$\begin{aligned}\mathrm{PEN}_i^N(\Omega) = \quad & \sum_{j=i+1}^{N} \mathbb{I}\left\{b^j = 0 \ \wedge \ r^j > r^i\right\} \\ & + \frac{1}{2} \sum_{j=i+1}^{N} \mathbb{I}\left\{b^j = 0 \ \wedge \ r^j = r^i\right\}\end{aligned}$$

Given a subgroup S of Ω, its *Average Subranking Loss*, $\mathrm{ASL}(S)$, is given by:

$$\mathrm{ASL}(S) = \mathrm{ARL}(\Omega')$$

where Ω' is the dataset constructed by taking from Ω only those records belonging to S.

B. Quality Measure: Relative Average Subranking Loss

In Exceptional Model Mining, we strive to find subgroups for which the target interaction captured by the model class is exceptional. This exceptionality is evaluated by a quality measure. We define a quality measure for the SCaPE model class, whose maxima, minima, and extremities correspond to three distinct goals:

Definition (Relative Average Subranking Loss). Given a subgroup S of Ω, its *Relative Average Subranking Loss*, φ_{rasl}, is given by:

$$\varphi_{\text{rasl}}(S) = \mathrm{ASL}(S) - \mathrm{ARL}(\Omega)$$

To find subgroups for which r represents b *poorly*, i.e., subgroups for which the soft classifier *does not work*, one should *maximize* φ_{rasl}; positive values for φ_{rasl} indicate that the soft classifier performs worse than usual on this subgroup. For instance, in our example dataset from Table I, $\varphi_{\text{rasl}}(a_1 = \mathrm{B}) = 0.2$. To find subgroups for which r represents b *well*, i.e., subgroups for which the soft classifier *does work*, one should *minimize* φ_{rasl}; negative values for φ_{rasl} indicate that the soft classifier performs better than usual on this subgroup. For instance, in our example dataset from Table I, $\varphi_{\text{rasl}}(a_1 = \mathrm{A}) = -0.8$.

Alternatively, one could find a list of subgroups for which the soft classifier performs exceptionally in general, by maximizing $|\varphi_{\text{rasl}}|$. The resulting list of subgroups could be partitioned into poorly- and well-classified subgroups in a post-processing step. In this paper, however, we maintain the strict separation of bad and good subgroups by presenting results of φ_{rasl}-maximizing and -minimizing runs separately.

VI. Experimental Results

For the sake of reproducibility, we first present results of artificial experiments performed on UCI datasets, before moving to the astrophysical domain and the FACT data.

A. Artificial Experiments on UCI Datasets

We illustrate the SCaPE model class for EMM with artificial experiments on nine datasets taken from the UCI machine learning repository [33]; properties of the selected datasets can be found in Table II. As selection criterion, the dataset must have a clear binary label, for use as the binary target b in the SCaPE model class. The final dataset collection spans a representative range in terms of number of records, number of attributes, and types of attributes.

1) Generating the Real-Valued Target: The SCaPE model class for EMM also requires the presence of a real-valued target r, which strives to emulate the binary target. We generate this numeric target ourselves, employing the RapidMiner analytics platform [15]. Each of the datasets (available to us in ARFF format) is fed to RapidMiner's out-of-the-box Naive Bayes [34] classifier, with standard parameter settings. This particular algorithm is selected because it can handle all types of available attributes (binary, nominal, numeric), as well as missing values. Moreover, the algorithm is a soft classifier: output comes in the form of probabilities. The resulting probabilities are written out to a new ARFF file, containing the original dataset but with an additional column containing the real-valued target r.

Now that both the binary target b and the real-valued target r are available, we can compute the Average Ranking Loss for each entire dataset, as defined in Section V-A1. The resulting values can be found in the last column of Table II. Note that for the Labor dataset the ARL is zero; the ordering imposed on the binary target by the real-valued target perfectly separates the zeroes from the ones. This means that there is a threshold value for which the converted Naive Bayes classifier makes no mistakes on this dataset. Particularly relevant to us is the observation that this perfect ordering is maintained when restricting the dataset to subgroups; every considered subgroup will necessarily also have an Average Subranking Loss of zero. Therefore, further experimentation on the Labor dataset makes no sense; the SCaPE model class cannot learn anything relevant about a classifier that makes no mistakes.

2) Parametrizing the EMM Algorithm: The SCaPE model class itself is implemented in Cortana [35], a toolbox featuring a plethora of Subgroup Discovery and Exceptional Model Mining settings. The model class should become publicly available in a future Cortana release; in the mean time, the authors will gladly provide interested parties with an unofficial jar file — please approach us via email. The central search algorithm at the core of Cortana is highly parametrizable, so for the sake of reproducibility we report the main parameter settings in this section.

We restrict the search to a refinement depth of 1, i.e., we allow the resulting subgroups to be defined on only one condition of one descriptor. This limits the expressive power of the resulting subgroups, but enhances their potential for interpretation by domain experts. Notice that nothing prevents the user of the SCaPE model class from seeking subgroups defined in terms of more attributes; we choose to limit ourselves to only one in order to end up with easily-interpretable subgroups, but mining deeper would not be a problem. The search space is defined conditional on the types of attributes. If an attribute a_i is binary, we consider the subgroups $a_i = 0$ and $a_i = 1$. If a_i is nominal with m different possible values $v_1, \ldots, v_m$, we consider the m subgroups of the form $a_i = v_j$. If a_i is real-valued, we consider all half-intervals with the values present in the dataset as endpoints. Only two of these subgroups are reported: the best-scoring subgroup of the form $a_i \leq v_j$, and the best-scoring subgroup of the form $a_i \geq v_j$. In the worst-case scenario, if a real-valued attribute has N distinct values, we consider $2N$ half-intervals. Finally, in an attempt to prevent overfitting, we only consider subgroups that contain at least 1% of the records in the dataset.

We run Cortana twice for each dataset: once maximizing φ_{rasl} in order to find subgroups on which the classifier performs poorly, and once minimizing φ_{rasl} in order to find subgroups on which the classifier performs well. In each run, we only report subgroups whose ASL outperforms (in a bad or a good way, depending on what we are looking for in this run) the baseline set by the ARL of the whole dataset: the maximizing run reports only subgroups with $\varphi_{\text{rasl}}(S) \geq 0$, and the minimizing run reports only subgroups with $\varphi_{\text{rasl}}(S) \leq 0$.

TABLE II. UCI DATASETS USED IN THE ARTIFICIAL EXPERIMENTS, WITH THE AVERAGE RANKING LOSSES FOR RAPIDMINER'S NAIVE BAYES CLASSIFIER.

i	Ω_i	N	# attributes		$\text{ARL}(\Omega_i)$
			discrete	numeric	
1	Adult	48842	8	6	1266.415
2	Credit-a	690	9	6	36.047
3	Haberman	306	1	2	22.436
4	Ionosphere	351	0	34	8.256
5	Labor	57	8	8	0.0
6	Mushroom	8124	22	0	0.459
7	Pima-indians	768	0	8	87.841
8	Tic-tac-toe	958	9	0	76.792
9	Wisconsin	699	0	9	8.303

3) Experimental Results: For every dataset, the best subgroup found while maximizing φ_{rasl} is reported in Table III, along with its quality. On the Tic-tac-toe dataset, Ω_8, no subgroups satisfying the constraints were found: all considered subgroups S had $\varphi_{\text{rasl}}(S) \leq 0$, meaning that their Average Subranking Loss was lower than the Average Ranking Loss on the whole dataset. In other words, there is not one particular coherent part of the search space where the soft classifier performs poorly; the awfulness is distributed fairly over the search space.

When we compare the final columns of Tables II and III, the values for two datasets stand out: Credit-a and Mushroom. For the subgroups S found on these datasets Ω_i, we see that $\varphi_{\text{rasl}}(S)$ is larger than $\text{ARL}(\Omega_i)$, which implies that the Average Subranking Loss of the top-ranked subgroups is more than twice as high as the Average Ranking Loss on the whole dataset. To explain why the soft classifier performs so extremely badly on these subspaces, we interpret the subgroups on the domains of their datasets. Unfortunately, for the Credit-a dataset, all attributes names and values have been scrambled to protect their confidential source [36]. Hence, we concentrate on the Mushroom dataset [37].

The Mushroom dataset details 23 species of gilled mushrooms in the Agaricus and Lepiota family [37, pp. 500–525]. The task is to classify the mushrooms as edible or poisonous; according to [37], there is no simple rule to make this separation. The SCaPE model class for EMM teaches us that the soft classifier has particular problems with the subgroup of mushrooms without odor. This is congruent with benchmark rules found in previous work [38]. Two particular rules from that paper are relevant to us. On the one hand, the authors report odor $=$ almond $\vee$ anise $\vee$ none $\Rightarrow$ edible [38, p. 5, negation of rule R_1]; this benchmark rule associates odorless mushrooms with the edible class. On the other hand, the authors report odor $=$ none $\wedge$ stalk-surface-below-ring $=$ scaly $\wedge$ stalk-color-above-ring $= \neg$brown $\Rightarrow$ poisonous [38, p. 5, rule R_3]; this benchmark rule associates odorless mushrooms with the poisonous class. Since this subspace of the dataset is associated with conflicting classes, it makes sense that the soft classifier finds this subspace tough to perform on, and it makes sense that the SCaPE model class singles out this subgroup as a part of the dataset that deserves more attention.

Table IV contains the best subgroup found while minimizing φ_{rasl} and associated quality for every dataset. Unlike the run maximizing φ_{rasl}, in these experiments we find subgroups satisfying the constraints on all datasets.

Comparing the final columns of Tables II and IV, we see

TABLE III. SUBGROUP PER UCI DATASET MAXIMIZING φ_{RASL}

Ω_i	Worst-classified subgroup S	$\varphi_{\text{rasl}}(S)$
Ω_1	Marital status = Married-civ-spouse	803.323
Ω_2	A9 = 0	69.561
Ω_3	Age_of_patient $\geq$ 33.0	0.142
Ω_4	a09 $\leq$ 0.66938	2.179
Ω_6	odor = n	14.508
Ω_7	plas $\leq$ 154.0	16.147
Ω_8	-	-
Ω_9	Cell shape uniformity $\leq$ 1.0	0.447

TABLE IV. SUBGROUP PER UCI DATASET MINIMIZING φ_{RASL}

Ω_i	Best-classified subgroup S	$\varphi_{\text{rasl}}(S)$
Ω_1	Capital gain $\geq$ 15020.0	-1266.415
Ω_2	A15 $\geq$ 5777.0	-36.047
Ω_3	Patients_year_of_operation = 68	-22.436
Ω_4	a26 $\geq$ 0.35696	-8.256
Ω_6	bruises? = f	-0.459
Ω_7	skin $\geq$ 52.0	-87.841
Ω_8	middle-middle-square = b	-74.792
Ω_9	Clump thickness $\geq$ 9.0	-8.303

that for almost all datasets we find a subgroup with $\varphi_{\text{rasl}}(S) = -\,\text{ARL}(\Omega_i)$. This implies that the Average Subranking Loss of the subgroup is zero; it is typically possible to find subspaces in the dataset on which the soft classifier performs perfectly.

On the Mushroom dataset, the best subgroup identified while minimizing φ_{rasl} is the group of mushrooms without bruises; classification is an easy task on this subspace of the data. This is backed by the fact that this subgroup appears in all kinds of previously found rules identifying edible mushrooms [39].

B. Real-World Experiments on FACT Data

The SCaPE model class for EMM also requires a binary and a real-valued target for real-world experiments. For this purpose we use the FACT Monte Carlo Simulation for gamma- and proton-induced air showers, as the binary target is already present by the information of the primary particle. The real-valued target is generated in RapidMiner by the WEKA random forest (RF) classifier, as it can produce probabilities of being a gamma shower expressed by the Signalness (as defined in Section II-A). The RF algorithm is implemented and used as a separation method in other IACT experiments such as MAGIC [12] and H.E.S.S. [13], where it has proven to be a stable and robust method performing comparatively superior to classical methods [12].

Disjoint Monte Carlo datasets were generated for training and testing the RF. The training sets for the individual trees containing gamma and proton showers were sampled in such a way that they have the same size. The dataset contains simulated reconstructed image parameters such as the area of the shower ellipse, and source-dependent parameters which allow to estimate a statistical signal of the astrophysical source at which the telescope is pointing. Finally, the binary and the real-valued target are added to the dataset.

On this FACT dataset, we again run Cortana twice (once maximizing and once minimizing φ_{rasl}), using the same parametrization as used in the artificial experiments of Section VI-A2. The Average Ranking Loss on the whole dataset is 1,446.761. The dataset will be made available upon request — please approach the authors via email.

1) Experimental Results — Maximizing φ_{rasl}: When maximizing φ_{rasl}, we strive to find subgroups on which the classifier performs poorly. The top-eight found subgroups are listed in Table V. As the last column shows, the first three subgroups have a substantially worse Average Subranking Loss than the rest, so they warrant further investigation. These three subsets are described by two distinct attributes. Both are source-dependent parameters, and between them they are strongly correlated.

The parameter ThetaSq describes the distance of the reconstructed source position, deduced from the orientation of the shower, to the real source position, known by source coordinates written to files during data taking. Thus, near-zero values express that the corresponding shower points to the real astrophysical source. We see the same behavior for the parameter dca, which describes the distance of the closest approach of the shower to the source position with respect to the x-axis. Again, showers with near-zero values have a higher probability of coming directly from the real source.

In the Monte Carlo simulations, gamma showers are assumed and simulated as if they were coming directly from the source, since this is the case in the real world we are interested in. In real data we also have a minor fraction of diffuse gamma showers, coming from sources other than the observed astrophysical source; these are not taken into account in the simulations. By contrast, proton-induced showers are assumed to be isotropically distributed in the sky.

Taking this information into account we can easily explain why the classifier performs particularly poorly on the first three subgroups in Table V. In both involved parameters, the gamma showers are accumulated around low values, while proton showers are equally distributed over the full parameter value range. Thus, the gamma showers decrease in frequency for higher values. For instance, the two subgroups for the dca parameter encompass just $\sim 10^{-5}\,\%$ of the gamma events in the whole dataset. While training the RF, one source-dependent parameter was used. This means that the classifier learned that the probability of being a gamma shower is high with low values in ThetaSq and dca. Conversely, the classification gets tougher if we have only a small number of gamma showers with high values in ThetaSq and dca.

This observation is corroborated by Figure 3. Each subfigure displays the distributions of the positives and negatives (normalized independently of each other; the figures give no direct information on the relative occurrence of positives and negatives!) related to the confidence level of the random forest that said record is a positive. Figure 3a displays these distributions over the entire dataset, while Figures 3b and 3d depict these distributions over just the subgroups defined in terms of dca. Here, the classifier has problems to distinguish between some gamma and proton showers, indicated by the confidence spikes around 0.5. For very small confidences the classification gets worse, as the overall probability of being a gamma shower is low in this subgroup and thus the confidence for gamma showers decreases. On the other hand, the proton shower classification is very good in these subgroups, for the aforementioned reasons. Comparing the distributions over the subgroup defined in terms of ThetaSq, in Figure 3c, with the overall distributions, we see that the

TABLE V. SUBGROUPS ON THE FACT DATASET MAXIMIZING φ_{RASL}

Rank	Worst-classified subgroups S	$\varphi_{\text{rasl}}(S)$
1.	dca $\geq$ 79.2745	1294.939
2.	ThetaSq $\geq$ 0.136131	1116.781
3.	dca $\leq$ -68.3173	1114.739
4.	SizeArea $\leq$ 0.5564718	100.786
5.	MCMomentumZ $\leq$ -1618.63	59.373
6.	cut1 = 0	46.957
7.	MCEnergy $\geq$ 1641.69	39.205
8.	Conc1Size $\leq$ 39.874977	28.153

proton distributions are roughly equivalent, but the subgroup encompasses substantially fewer high-confidence gamma rays.

The subgroups in Table V with less extreme values for φ_{rasl}, such as the ones with rank 4 and 8, are less straightforward to explain. The parameter SizeArea describes the compactness of the deposited light of the showers and the parameter Conc1Size describes the deposited light in the brightest pixel of a shower. The higher these values are, the more likely it is that we are dealing with a gamma shower. On first look, the poor classification on these particular subgroups is surprising, because the parameter distributions are clearly separated for lower values of gamma and proton showers as well. However, this result could be explained by internal cuts in the RF, which affects the distributions and tends to misclassify events with a lower probability of being a gamma.

2) Experimental Results — Minimizing φ_{rasl}: When minimizing φ_{rasl}, we strive to find subgroups on which the classifier performs well. The top-eight such subgroups are listed in Table VI.

The first and eighth-ranked subgroup are described by the same parameter cosdeltaalpha, which is again source-dependent and roughly expresses the cosine of the angle between the shower main axis and the source position. Thus, values of cosdeltaalpha around 1 or -1 indicate that the shower axis is pointing to the source, which also means a higher probability for the shower to come directly from the source and thus a higher probability of being a gamma shower. Contrary to dca, which appears high-ranked in the poorly-classified subgroups, these well-classified subgroups contain a big fraction of gamma showers compared to the fraction of proton showers. This means that the classifier learns that showers which are contained in these subgroups are very likely gamma showers and are better classified than in other ranges.

The third-ranked subgroup is the known source-dependent parameter ThetaSq. It appears in the well-classified subgroups with very low values as well as in the poorly-classified subgroups with higher values. This behavior is perfectly explainable, as very low values indicate a higher probability of being a gamma shower, and the probability decreases slowly with higher ThetaSq values, until a value is reached where gamma showers cannot be distinguished well from the proton showers if only ThetaSq is taken into account.

We see the same effect with the seventh-ranked subgroup described by SizeArea. The classifier performs well on higher values but worse on lower values. Again, this result could be explained by internal cuts in the RF.

TABLE VI. SUBGROUPS ON THE FACT DATASET MINIMIZING φ_{RASL}

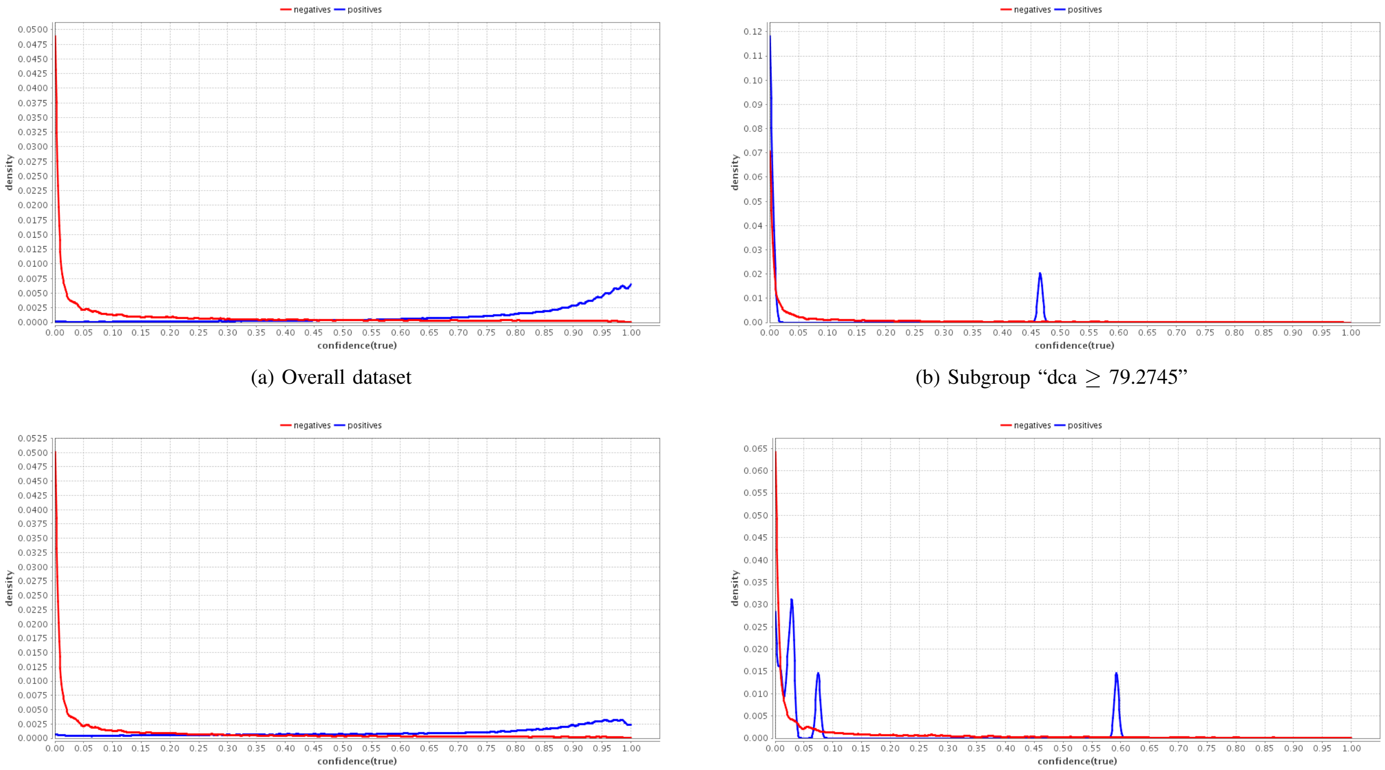
Rank	Best-classified subgroups S	$\varphi_{\text{rasl}}(S)$
1.	cosdeltaalpha $\geq$ 0.999994	-1446.259
2.	SizeSinglePixels $\geq$ 372.953	-1445.761
3.	ThetaSq $\leq$ 6.57561E-4	-1445.753
4.	Length $\leq$ 9.70734	-1445.336
5.	logLength $\leq$ 0.98710024	-1445.336
6.	NumberSinglePixels $\geq$ 73.0	-1444.539
7.	SizeArea $\geq$ 1.8111843	-1444.535
8.	cosdeltaalpha $\leq$ -0.999995	-1444.275

VII. CONCLUSIONS

Motivated by a real-life astrophysics data scenario, we introduce the SCaPE (Soft Classifier Performance Evaluation) model class for Exceptional Model Mining (EMM). SCaPE strives to find coherent subgroups displaying exceptional interaction between the probabilities provided by a soft classifier and a binary ground truth. This interaction is evaluated by the Average (Sub-)Ranking Loss, a quantity expressing how well the soft classifier probabilities can represent the binary ground truth. The quality measure φ_{rasl} is designed to find coherent subspaces of the dataset where the soft classifier performs poorly (when maximizing φ_{rasl}), well (when minimizing φ_{rasl}), or exceptionally (when maximizing $|\varphi_{\text{rasl}}|$). The focus of EMM lies on finding easily interpretable subgroups. Hence, as opposed to a meta-learning framework, which is focused on letting the machine improve the machine, the primary goal in the SCaPE model class for EMM is to provide a better *understanding* to the domain expert. We want the expert to be able to understand where his/her classifier does or does not work well, by reporting the problem and success areas in familiar terms.

We illustrate the findings one could expect from the SCaPE model class by artificial experiments on nine UCI datasets. On seven of those, subgroups are found proving troublesome for our classifier, and on eight UCI datasets, subgroups are found where our classifier has barely any problems (on the ninth dataset, the classifier already performed perfectly, so neither troublesome nor particularly benign areas could be highlighted). The found subgroups on the Mushroom dataset, whose learning task is well known to be nontrivial, are coherent with previously reported benchmark rules. The SCaPE model class highlights as a particularly troublesome area a subgroup whose characterizing feature is known to be associated with both classes in the dataset, and as a particularly benign area a subgroup that is associated with one particular class. Overall, when minimizing φ_{rasl} one easily finds small subgroups on which the soft classifier performs perfectly; the subgroups on which the soft classifier performs badly are typically less trivial, hence they demand further attention.

We also perform real-world experiments with the SCaPE model class, on an astrophysics dataset concerned with the classification of air showers induced by high-energetic cosmic particles. The subgroups with the most deviating Average Subranking Losses — both the poorly-classified ones and the well-classified ones — have an astrophysical interpretation corroborating their appearance as a particularly (un-)problematic subspace of the search space. Subgroups with less extreme but still high/low values for the quality measure are non-trivial to explain and deserve a closer look. The results show that the random forest classifier performs better when the incidence of



(a) Overall dataset

(b) Subgroup “dca $\geq$ 79.2745”

(c) Subgroup “ThetaSq $\geq$ 0.136131”

(d) Subgroup “dca $\leq$ -68.3173”

Fig. 3. Distribution of positives and negatives in the FACT dataset, for the whole dataset and for the three subgroups with highest φ_{rasl}-values. In each plot, both distributions are normalized, independently of each other.

gamma showers is higher.

In gamma ray astronomy, the separation of gamma and proton showers marks an important step in the analysis of astrophysical sources. Better classifier performance leads to less dilution of the interesting physics results and improves the statement of results of the astrophysical source. The result set will more frequently contain the infrequently appearing gamma showers, which should increase the effective observation time. Due to the importance of the separation in this field, understanding why the classifier does not perform as desired is extremely valuable. The SCaPE model class for EMM helps to understand the classification, which leads to ideas on how to improve the overall classifier performance. The performance of this particular RF could be improved by building separate clusters in the training dataset and training a random forest for each cluster; as future work we intend to investigate whether the experimental results of this paper lead to a good starting point for this clustering.

Acknowledgments

We thank Marvin Meeng for his helpful comments on a previous version of this paper.

References

[1] W. Duivesteijn, J. Thaele, Understanding Where Your Classifier Does (Not) Work — the SCaPE Model Class for EMM, Proc. ICDM 2014, to appear.

[2] H. Anderhub, M. Backes, A. Biland et al., Design and Operation of FACT – the First G-APD Cherenkov Telescope, arXiv:1304.1710 [astro-ph.IM], Instrumentation and Methods for Astrophysics 8, pp. P06008, 2013.

[3] T. Bretz, H. Anderhub et al., FACT — The First G-APD Cherenkov Telescope: Status and Results, arXiv:1308.1512, Instrumentation and Methods for Astrophysics (astro-ph.IM), 2013.

[4] D. Leman, A. Feelders, A.J. Knobbe, Exceptional Model Mining, Proc. ECML/PKDD (2), pp. 1–16, 2008.

[5] W. Duivesteijn, Exceptional Model Mining, PhD thesis, Leiden University, 2013.

[6] C. Grupen, Astroteilchenphysik: Das Universum im Licht der kosmischen Strahlung, Vieweg, 2000.

[7] T.C. Weekes, M.F. Cawley, D.J. Fegan, K.G. Gibbs, A.M. Hillas, P.W. Kowk, R.C. Lamb, D.A. Lewis, D. Macomb, N.A. Porter, P.T. Reynolds, G. Vacanti, Observation of TeV gamma rays from the Crab nebula using the atmospheric Cerenkov imaging technique, The Astrophysical Journal 342, pp. 379–395, 1989.

[8] M. Meyer, D. Horns, H.-S. Zechlin, The Crab Nebula as a standard candle in very high-energy astrophysics, Astronomy and Astrophysics 523 (A2), 11 pp., 2010.

[9] S.F. Taylor, T. Abu-Zayyad, K. Belov et al., The Highest Energy Cosmic Rays and Gamma Rays, American Astronomical Society, 192nd AAS Meeting, # 09.03; Bulletin of the American Astronomical Society 30, p. 827, 05/1998.

[10] CORSIKA - An Air Shower Simulation Program, https://web.ikp.kit.edu/corsika/

[11] T. Bretz, D. Dorner, MARS - CheObs ed. A flexible Software Framework for future Cherenkov Telescopes, In C. Leroy, P.-G. Rancoita, M. Barone, A. Gaddi, L. Price, & R. Ruchti , editor, Astroparticle, Particle and Space Physics, Detectors and Medical Physics Applications, pages 681 –687, April 2010, doi: 10.1142/9789814307529_0111

[12] J. Albert et al., Implementation of the Random Forest Method for the

Imaging Atmospheric Cherenkov Telescope MAGIC, arXiv:0709.3719 [astro-ph], 2007. [http://arxiv.org/pdf/0709.3719v2.pdf]

[13] K. Egberts, C. van Eldik, J. Hinton for the H.E.S.S. Collaboration, Measurement of Cosmic Ray Electrons with H.E.S.S., Proceedings of the 30th International Cosmic Ray Conference, Vol. 2 (OG part 1), pp. 35–38, 2008.

[14] L. Breiman, Random Forests, Machine Learning 45, pp. 5–32, 2001.

[15] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, T. Euler, YALE: Rapid Prototyping for Complex Data Mining Tasks, Proc. KDD, pp. 935–940, 2006.

[16] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA Data Mining Software: An Update, SIGKDD Explorations 11 (1), pp. 10–18, 2009.

[17] D. Hand, N. Adams, R. Bolton (eds), Pattern Detection and Discovery, Springer, New York, 2002.

[18] K. Morik, J.F. Boulicaut, A. Siebes (eds), Local Pattern Detection, Springer, New York, 2005.

[19] H. Mannila, H. Toivonen, Levelwise Search and Borders of Theories in Knowledge Discovery, Data Mining and Knowledge Discovery 1 (3), pp. 241–258, 1997.

[20] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A.I. Verkamo, Fast Discovery of Association Rules, Advances in Knowledge Discovery and Data Mining, pp. 307–328, 1996.

[21] F. Herrera, C.J. Carmona, P. González, M.J. del Jesus, An Overview on Subgroup Discovery: Foundations and Applications, Knowledge and Information Systems 29 (3), pp. 495–525, 2011.

[22] 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.

[23] S.D. Bay, M.J. Pazzani, Detecting Group Differences: Mining Contrast Sets, Data Mining and Knowledge Discovery 5 (3), pp. 213–246, 2001.

[24] G. Dong, J. Li, Efficient Mining of Emerging Patterns: Discovering Trends and Differences, Proc. KDD, pp. 43–52, 1999.

[25] A.M. Jorge, P.J. Azevedo, F. Pereira, Distribution Rules with Numeric Attributes of Interest, Proc. PKDD, pp. 247–258, 2006.

[26] L. Umek, B. Zupan, Subgroup Discovery in Data Sets with Multi-Dimensional Responses, Intelligent Data Analysis 15 (4), pp. 533–549, 2011.

[27] E. Galbrun, P. Miettinen, From Black and White to Full Color: Extending Redescription Mining Outside the Boolean World, Statistical Analysis and Data Mining 5 (4), pp. 284–303, 2012.

[28] R. Vilalta, Y. Drissi, A Perspective View and Survey of Meta-Learning, Artificial Intelligence Review 18 (2), pp. 77–95, 2002.

[29] R.J. Henery, Methods for Comparison, in: D. Michie, D.J. Spiegelhalter, C.C. Taylor (eds.), Machine Learning, Neural and Statistical Classification, Ellis Horwood, 1994.

[30] J. Vanschoren, H. Blockeel, Towards Understanding Learning Behavior, Proc. BENELEARN, pp. 89–96, 2006.

[31] 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), pp. 1503–1529, 2014.

[32] G. Tsoumakas, I. Katakis, I.P. Vlahavas, Mining Multi-Label Data, Data Mining and Knowledge Discovery Handbook, Springer, pp. 667–685, 2010.

[33] K. Bache, M. Lichman, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml], Irvine, CA: University of California, School of Information and Computer Science, 2013.

[34] G.H. John, P. Langley, Estimating Continuous Distributions in Bayesian Classifiers, Proc. UAI, pp. 338–345, 1995.

[35] M. Meeng, A.J. Knobbe, Flexible Enrichment with Cortana – Software Demo. Proc. Benelearn, pp. 117–119, 2011.

[36] J.R. Quinlan, Simplifying Decision Trees, International Journal of Man-Machine Studies 27 (3), pp. 221–234, 1987.

[37] G.H. Lincoff (Pres.) et al., The Audubon Society Field Guide to North American Mushrooms, New York, Alfred A. Knopf, 1981.

[38] W. Duch, R. Adamczak, K. Grabczewski, M. Ishikawa, H. Ueda, Extraction of Crisp Logical Rules Using Constrained Backpropagation Networks — Comparison of Two New Approaches, Proc. ESANN, pp. 109–114, 1997.

[39] M.R.W. Dawson, D.A. Medler, Of Mushrooms and Machine Learning: Identifying Algorithms in a PDP Network, Canadian Artificial Intelligence 38, pp. 14–17, 1996.