Instant Exceptional Model Mining using Weighted Controlled Pattern Sampling

Sandy Moens[1,2] and Mario Boley[2,3]

[1] University of Antwerp, Belgium, firstname.lastname@uantwerpen.be,
[2] University of Bonn, Germany, firstname.lastname@uni-bonn.de,
[3] Fraunhofer IAIS, Germany, firstname.lastname@iais.fgh.de

Abstract. When plugged into instant interactive data analytics processes, pattern mining algorithms are required to produce small collections of high quality patterns in short amounts of time. In the case of Exceptional Model Mining (EMM), even heuristic approaches like beam search can fail to deliver this requirement, because in EMM each search step requires a relatively expensive model induction. In this work, we extend previous work on high performance controlled pattern sampling by introducing extra weighting functionality, to give more importance to certain data records in a dataset. We use the extended framework to quickly obtain patterns that are likely to show highly deviating models. Additionally, we combine this randomized approach with a heuristic pruning procedure that optimizes the pattern quality further. Experiments show that in contrast to traditional beam search, this combined method is able to find higher quality patterns using short time budgets.

Keywords: Controlled Pattern Sampling, Subgroup Discovery, Exceptional Model Mining

1 Introduction

There is a growing body of research arguing for the integration of Local Pattern Mining techniques into instant, interactive discovery processes [3, 5, 9, 10, 14]. Their goal is to tightly integrate the user into the discovery process to facilitate finding patterns that are interesting with respect to her current subjective interest. In order to allow true interactivity, the key requirement for a mining algorithm in such processes, is that it is capable of producing high quality results within very short time budgets—only up to a few seconds.

A particularly hard task for this setting is Exceptional Model Mining (EMM) [8], i.e., the discovery of subgroups showing data models that highly deviate from the model fitted to the complete data. In the EMM setting, even fast heuristic methods that cut down the search space tremendously, e.g., beam search [8], can fail to deliver the fast response times necessary for the interactive setting. This comes from the fact that every individual search step in the subgroup description space involves an expensive model induction step.

In this paper, we extend an alternative randomized technique to pattern discovery, Controlled Direct Pattern Sampling [4], and adapt it to EMM. As opposed to many other algorithmic approaches, direct pattern sampling does not traverse any part of the pattern search space. Instead, it defines an efficient sampling process that yields patterns according to a distribution, which overweights high-quality patterns. A previously published framework [6] allows to express distributions in terms of the pattern support. Here we extend it to specify distributions in terms of the weighted pattern support. We then develop a weighting scheme based on Principal Component Analysis, which leads to efficient sampling procedures particularily suitable for EMM tasks. As we show empirically, when combined with a lightweight local search procedure as post-processing step, the resulting EMM algorithms outperform both, pure local search as well as pure sampling strategies, and deliver high-quality results for short time budgets.

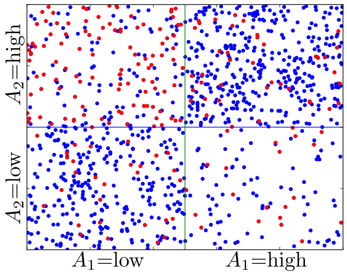
2 Exceptional Model Mining

Throughout this paper we assume that a **dataset** $D = \{d_1, \dots, d_m\}$ consists of m **data records** $d \in D$, each of which is described by n **descriptive attributes** $A = \{a_1, \dots, a_n\}$ and annotated by k **target attributes** $T = \{t_1, \dots, t_k\}$. All attributes $f \in A \cup T$ assign to each data record a value from their **attribute domain** $\mathrm{Dom}(f)$, i.e., $f : D \to \mathrm{Dom}(f)$. In this paper we assume that all attributes $f \in A \cup T$ are either **numeric**, i.e., $\mathrm{Dom}(f) \subset \mathbb{R}$ and we use $\leq$ to compare attribute values, or **categoric**, i.e., $\mathrm{Dom}(f)$ is finite and its values are conceptually incomparable. We are interested in conjunctive **patterns** of simple binary propositions about individual data records. This is the standard setting in subgroup discovery and itemset mining. That is, a **pattern descriptor** p can be formalized as a set $p = \{c_1, \dots, c_l\}$ where $c_j : \mathrm{Dom}(a_{i_j}) \to \{\mathrm{true}, \mathrm{false}\}$ is a **constraint** on the descriptive attribute a_{i_j} for $j = 1, \dots, l$ (corresponding to item literals in frequent set mining). Correspondingly, the **support set** (or **extension**) of p is the subset of data records for which all constraints hold, i.e.,

$$\mathrm{Ext}(D, p) = \{d \in D\colon\ c_1(a_{i_1}(d)) \wedge \dots \wedge c_l(a_{i_l}(d))\} \ ,$$

and the **frequency** of p is defined as the size of its extension relative to the total number of data records $\mathrm{frq}(D, p) = |\mathrm{Ext}(D, p)|/m$. We write $\mathrm{Ext}(p)$, resp. $\mathrm{frq}(p)$, when D is clear. For the constraints, one typically uses equality constraints if a_{i_j} is categorical, i.e., $c_j(v) \equiv v = w$ for $w \in \mathrm{Dom}(a_{i_j})$, and interval constraints if a_{i_j} is numeric, i.e., $c_j(v) \equiv v \in [l, u]$ for a few expressive choices of interval borders $l, u \in \mathrm{Dom}(a_{i_j})$ (e.g., corresponding to the quartiles of $\{a_{i_j}(d)\colon\ d \in D\}$).

Let C denote the **constraint universe** containing all the constraints that we want to use to express patterns. We are interested in searching the **pattern language** $L = \mathcal{P}(C)$ for descriptors $p \in L$ with a) a relatively high frequency and b) such that the target attributes behave differently in $\mathrm{Ext}(p)$ than in the complete data. This behavior is captured by how the target attributes are represented by a model of a certain **model class** M. That is, formally, a **model** $m(D') \in M$ can be induced for any subset of the data records $D' \subseteq D$, and there is a meaningful



(a) Data representation

	A_1=low	A_1=high
A_2=high	.20	.40
A_2=low	.30	.10

(b) Global model

	A_1=low	A_1=high
A_2=high	.55	.20
A_2=low	.15	.10

(c) Local model

Fig. 1: Exceptional contingency table models for fictitious dataset with two numerical attributes: red+blue is the global data and red is a local pattern. (b) shows the global model and (c) the local model. The model deviation equals .35.

distance measure $\delta \colon M \times M \to \mathbb{R}_+$ between models. Then the **interestingness** of a pattern descriptor $p \in L$ is given as $\mathrm{int}(p) = \mathrm{frq}(p)\, \delta(m(D), m(\mathrm{Ext}(p)))$.

In this paper we focus on non-functional models that treat all target attributes symmetric. When all target attributes are numeric, the perhaps simplest example of a model class are the **mean models** $M_{\mathrm{mn}} = \mathbb{R}^k$ defined by

$$m(D') = (\bar{t}_1(D'), \ldots, \bar{t}_k(D'))$$

with $\bar{t}_i(D') = \sum_{d \in D'} t_i(d)/|D'|$. A useful distance measure between two mean models $m, m' \in M_{\mathrm{mn}}$ is for instance given by the **normalized Euclidean distance** $\delta_{\mathrm{nl2}}(m, m') = \sqrt{(m - m')^T S^{-1} (m - m')}$ where S denotes the diagonal matrix with entries $S_{i,i}$ equal to the standard deviation of target attribute t_i on the data. For categorical targets, a simple example are the **contingency table models** $M_{\mathrm{ct}} = \mathbb{R}^{V_{t_1} \times \cdots \times V_{t_k}}$ where each $m \in M_{\mathrm{ct}}$ represents the relative counts of all target value combinations, i.e.,

$$m(D')_v = |\{d \in D' \colon\, t_1(d) = v_1 \wedge \cdots \wedge t_k(d) = v_k\}|/|D'|$$

for all $v \in V_{t_1} \times \cdots \times V_{t_k}$. A meaningful distance measure between contingency tables $m, m' \in M_{\mathrm{ct}}$ is the **total variation distance** defined by $\delta_{\mathrm{tvd}}(m, m') = \sum_{v \in V_{t_1} \times \cdots \times V_{t_k}} |m_v - m'_v|/2$. See Figure 1 for an example.

Since EMM is a computationally hard problem and no efficient way is known to find a pattern descriptor $p \in L$ for a given dataset that maximizes the EMM interestingness, the standard algorithmic approach to EMM is heuristic **beam search**. This strategy is an extension of a greedy search, where on each search level (corresponding to a number of constraints in a pattern descriptor) instead of extending only one partial solution by a constraint that locally optimizes the interestingness, one considers $b \in \mathbb{N}$ best partial solutions. This parameter b is referred to as the **beam-width**. Formally, starting from **search level** $L'_0 = \{\emptyset\}$, level L'_{i+1} is defined as

$$L'_{i+1} = \bigcup_{i=1}^{b} \{p_i \wedge c \colon\, c \in C \setminus p_i, \mathrm{frq}(p_i \wedge c) \geq \tau\}$$

where $\{p_1, \ldots, p_b, \ldots, p_z\} = L'_i$ in some order consistent with decreasing interestingness, i.e., $\text{int}(p_i) \geq \text{int}(p_j)$ for $i < j$, and $\tau \in [0, 1]$ is a **frequency threshold** used to reduce the search space (possibly alongside other anti-monotone hard constraints). This algorithm has to construct the models for $\Theta(bl|C|)$ elements of the pattern language where l denotes the average length of descriptors that satisfy the constraints.

3 Sampling Exceptional Models

In this section we develop an alternative approach to EMM using Controlled Direct Pattern Sampling (CDPS). The key idea of this approach is that we create random patterns by a fast procedure following a controlled distribution that is useful for EMM, i.e., that favors patterns with a high frequency and a large model deviation. In contrast to beam search, sampling only requires to perform a model induction after a full descriptor is found. As we will argue later, it is most efficient to combine this sampling approach with a very lightweight local search procedure as post-processing step.

3.1 Weighted Controlled Direct Pattern Sampling

Boley et al. [6] gives a fast algorithm for CDPS that draws samples from a user-defined distribution over the pattern space using a simple two-step random experiment. Distributions that can be simulated with this approach are those that can be expressed as the product of frequency functions wrt to different parts of the data. Here we extend this idea by allowing to specify **utility weights** $w(d) \in \mathbb{R}^+$ for each data record $d \in D$.

With this we define the **weighted frequency** as the relative total weight of a pattern's extension, i.e., $\text{wfrq}(D, p) = \sum_{d \in \text{Ext}(p)} w(d) / \sum_{d \in D} w(d)$, and the **negative weighted frequency** equals $\overline{\text{wfrq}}(D, p) = 1 - \text{wfrq}(D, p)$. Let $D_i^+, D_j^- \subseteq D$ be subsets of the data for $i \in \{1, \ldots, a\}$ and $j \in \{1, \ldots, b\}$. Now we can define a random variable over the pattern space $\mathbf{p} \in L$ by

$$\mathbb{P}[\mathbf{p} = p] = \prod_{i=1}^{a} \text{wfrq}(D_i^+, p) \prod_{j=1}^{b} \overline{\text{wfrq}}(D_j^-, p)/Z, \tag{1}$$

with a normalization constant Z such that $\sum_{p \in L} \mathbb{P}[\mathbf{p} = p] = 1$. This distribution gives a high probability to patterns that have a high weighted frequency in D_i^+, further referred to as the **positive data portions**, and a low weighted frequency in D_j^-, further referred to as the **negative data portions**. As an example, when designing an algorithm for subgroup discovery in data with binary labels, data records with a positive label could be assigned to a positive data portion and data records with a negative label to a negative data portion. This results in a pattern distribution favoring patterns for which data records in their extension are assigned mainly a positive label. In subsequent sections we will use distributions from this family (Eq. 1) to construct effective EMM algorithms.

However, we first show that realizations of $\mathbf{p}$ can be computed with a two-step framework similar to the one given in Boley et al. [6][4].

Let us denote by $\mathbb{D} = D_1^+ \times \cdots \times D_a^+ \times D_1^- \times \cdots \times D_b^-$ the Cartesian product of all data portions involved in the definition of $\mathbf{p}$ containing one representative record for each positive and each negative data portion. For a tuple of data records $r \in \mathbb{D}$ let

$$L_r = \{p \in L\colon\ r(i) \in \operatorname{Ext}(D_i^+, p), 1 \le i \le a \wedge r(j) \notin \operatorname{Ext}(D_{j-a}^-, p), a < j \le a+b\}$$

denote the set of pattern descriptors having in their extensions all positive representatives $r(1), \ldots, r(a)$ and none of the negatives $r(a+1), \ldots, r(a+b)$. Then consider the random variable $\mathbf{r} \in \mathbb{D}$ defined by

$$\mathbb{P}[\mathbf{r} = r] = |L_r| \prod_{i=1}^{a+b} w(r(i))$$

In the following proposition we note that in order to simulate our desired distribution $\mathbf{p}$ it is sufficient to first draw a realization r of $\mathbf{r}$ and then to uniformly draw a pattern from L_r.

Proposition 1. *For a finite set X denote by $\mathbf{u}(X)$ a uniform sample from X. Then $\mathbf{p} = \mathbf{u}(L_{\mathbf{r}})$.*

Proof. Denote by $\mathbb{D}^p = \{r \in \mathbb{D}\colon\ p \in L_r\}$. Noting that $\mathbb{D}^p$ is equal to

$$\operatorname{Ext}(D_1^+, p) \times \cdots \times \operatorname{Ext}(D_a^+, p) \times \big(D_1^- \setminus \operatorname{Ext}(D_1^-, p)\big) \times \cdots \times \big(D_b^- \setminus \operatorname{Ext}(D_b^-, p)\big)$$

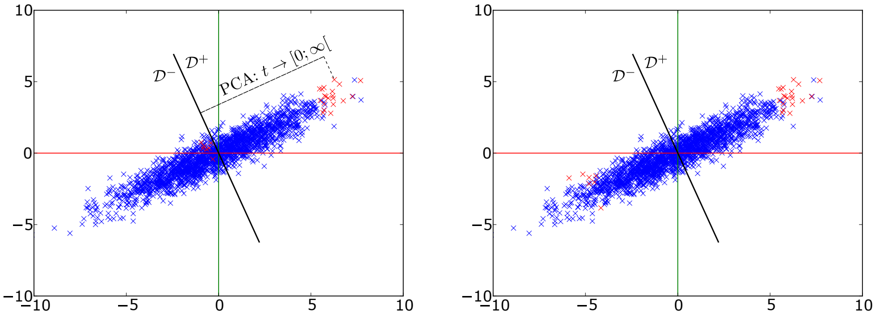
it follows that

$$\begin{aligned}
\mathbb{P}[\mathbf{u}(L_{\mathbf{r}}) = p] &= \sum_{r \in \mathbb{D}^p} \mathbb{P}[\mathbf{u}(L_r) = p | \mathbf{r} = r] \mathbb{P}[\mathbf{r} = r] \\
&= \sum_{r \in \mathbb{D}^p} \frac{1}{|L_r|} \frac{|L_r| \prod_{i=1}^{a+b} w(r(i))}{Z} = \frac{1}{Z} \sum_{r \in \mathbb{D}^p} \prod_{i=1}^{a+b} w(r(i)) \\
&= \frac{1}{Z} \prod_{i=1}^{a} \sum_{d \in \operatorname{Ext}(D_i^+, p)} w(d) \prod_{j=1}^{b} \sum_{d \in D_j^- \setminus \operatorname{Ext}(D_j^-, p)} w(d) \\
&= \frac{1}{Z} \prod_{i=1}^{a} \operatorname{wfrq}(D_i^+, p) \prod_{j=1}^{b} \overline{\operatorname{wfrq}}(D_j^-, p) = \mathbb{P}[\mathbf{p} = p]
\end{aligned} \tag{2}$$

□

Efficient implementations of $\mathbf{r}$ and $\mathbf{u}(L_r)$ for $r \in \mathbb{D}$ can be performed by using coupling from the past and sequential constraint sampling, respectively, for which we refer to Boley et al. [6]. In the remainder of this paper, we focus on utilizing the resulting pattern sampler for EMM.

[4] Note that the algorithm given in Boley et al. [6] also allows to specify modular prior preferences for pattern descriptors as well as to avoid descriptors of length 1 and 0. We omit both additions here for the sake of simplicity and note that they could be included in exactly the same way as in the original algorithm.



(a) PCA based weight computation and pattern describing points with high mean shift deviation.

(b) Pattern describing points with low mean shift deviation.

Fig. 2: Example weighting scheme for mean shift models for fictitious dataset.

3.2 Application to Exceptional Model Mining

We start with the case of the contingency table models M_{ct}. Let $\mathbb{V} = V_{t_1} \times \cdots \times V_{t_k}$ be the set of all cells of a contingency table $m \in M_{\mathrm{ct}}$. We give an instantiation of Eq. 1 using exactly one positive and one negative frequency factors, i.e., $a = b = 1$ with disjoint data portions D^+, D^- that partition D. The idea is that we try to oversample patterns with an extension lying mostly in contingency table cells with small counts. For that we can sample a random subset of the table cells $\mathbf{W} \subseteq \mathbb{V}$ with $|W| = |\mathbb{V}|/2$ such that $\mathbb{P}[v \in \mathbf{W}] = m(D)_v^{-1} Z^{-1}$ and assign each $d \in D$ to D^+ if $v(d) \in \mathbf{W}$ and to D^- otherwise (where $v(d)$ denotes the contingency table cell of d). Note that more focused versions of this distribution can be achieved by simply replicating the two frequency factors described here. Also, for this simple instantiation we did not use utility weights for the data records (i.e., they are chosen uniform).

Now turning to the case of high mean shift deviation M_{mn}, we will give another instantiation of Eq. 1 that also uses weights in addition to defining suitable positive and negative data portions. Since we only have one weight vector for the data records, we are interested in the direction in which the largest target deviation from the mean can be achieved. By applying a centralized Principal Component Analysis (PCA) we can find a linear transformation of the target data vectors that maximizes the variance among the data points. The first component then gives the direction of interest. Let us denote by $\mathrm{PCA}_1(d) \in \mathbb{R}$ the first component of $(t_1(d), \ldots, t_k(d))$, i.e., the length of the target vector of a data record in the direction of highest variance. We define $d \in D^+$ if $\mathrm{PCA}_1(d) \geq 0$ and $d \in D^-$ otherwise. This idea is shown in Figure 2a: the black line shows the first component, the points to the right are assigned to D^+ and the points to the left are assigned to D^-. Note that in practice we can randomly choose which side is D^+ or D^-. For the computation of weights, recall that our task is finding descriptors with a high mean shift (see again Figure 2a). As such, data points in the extension of a pattern that are in D^+, should be far away from the mean. While data point in the extension of a pattern that are in D^-, should be very close to the mean, such that the mean gets minimally shifted towards the center. Hence, it is sensible to use $w(d) = |\mathrm{PCA}_1(d)|$ as weights for data records. This means that in the positive part, data records far away from the mean will contribute a lot to $\mathrm{wfrq}(D^+, p)$ and in the negative part, points close to the center—having small absolute weights—will contribute a lot to $\overline{\mathrm{wfrq}}(D^-, p)$.

Finally, we propose to combine the EMM pattern sampler with a pruning routine, in order to further optimize the quality of sampled patterns. Our method then becomes a two-step framework: (1) optimizes the model deviation through direct sampling and (2) optimizes the interestingness via pruning. We employ heuristic optimization on patterns to optimize wrt the interestingness. First, we generate a random permutation of constraints. Then we remove each constraint one by one. If the quality increases, we replace the pattern. For a pattern p, the pruning step constructs models for $\Theta(k)$ patterns where $k = |p|$. The total cost of our sampling procedure is $\Theta(l + 1)$, where l is the average length of descriptors that satisfy the constraints. This is a theoretical advantage over beam search when model induction is expensive (e.g., when there are a lot of data points for which contingency tables have to be computed).

4 Experiments

dataset	#attributes	#data records	time budget (ms)
Adult	15	30,163	300
Bank Marketing	17	45,211	300
Twitter	34	100,000	2,000
Cover Type	10	581,012	2,000

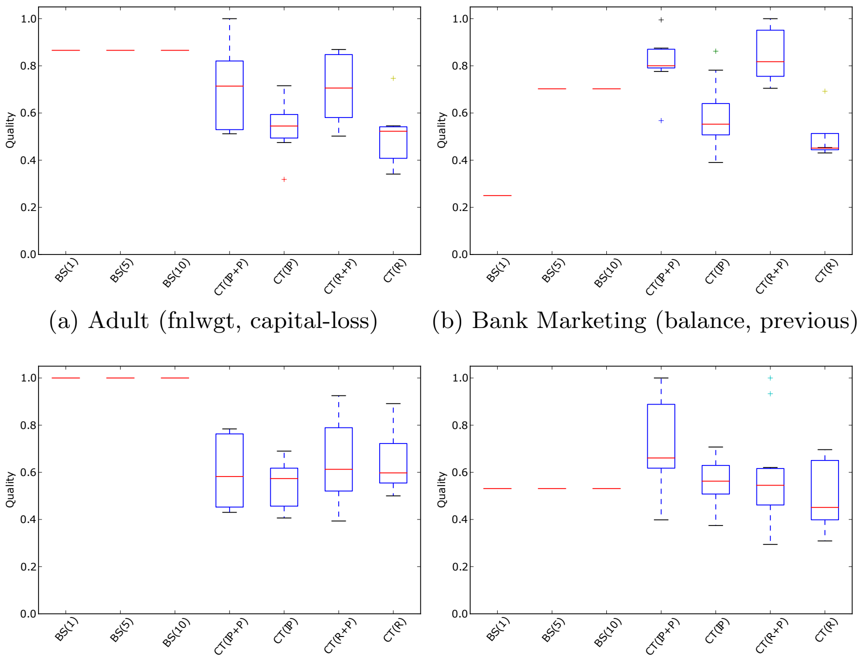
Table 1: Overview of dataset characteristics

In the previous section we introduced a method for sampling exceptional models. We show now that our method is able to outperform beam search when given short time budgets. Throughout the experiments we used datasets available from the UCI Machine Learning Repository [2]. Their main characteristics together with the individual mining times are summarized in Table 1. The mining times do not taking into account loading the data, since in interactive systems the data is already loaded. For each dataset we removed lines with missing values. For Twitter we used only the first two measurements and for Cover Type we used the first 10 attributes. At last, both techniques run on Java 7.

4.1 Contingency Table Quality

In this experiment we analyze contingency table models found by our sampler and compare them to models found by beam search. Throughout the experiments we used short time budgets found in Table 1. For the quality assessment we used the interestingness from Section 2.

We ran the algorithms to find exceptional contingency tables with 2 predefined targets. For beam search (BS) we only reported runs with beam widths 1, 5 and 10 since the others behave similar. For the sampling process Equation 1 with 2 positive and 1 negative factor and no weighting strategy. We fixed



(a) Adult (fnlwgt, capital-loss)

(b) Bank Marketing (balance, previous)

(c) Twitter 34 (NSD_0, CS_0)

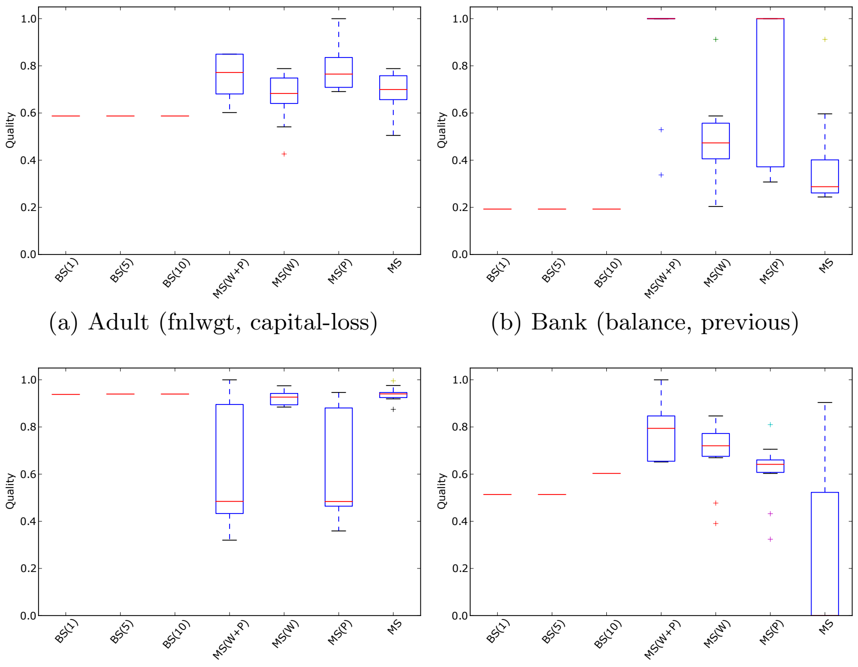
(d) Cover Type (elevation, slope)

Fig. 3: Max qualities for 2 target exceptional contingency table models.

four settings: $CT(IP+P)$ – inverse probability for sampling D^+ and pruning, $CT(IP)$ – inverse probability without pruning, $CT(R+P)$ – uniform selection of D^+ in combination with pruning $CT(R)$ – uniform selection without pruning. Moreover, we ran each algorithm 10 times and extracted the highest quality patterns found for each run. We then normalized the results by the best pattern found over all algorithms. Aggregated results are shown in Figures 3.

Generally, sampling is able to find higher quality patterns using short time budgets. The problem with beam search is that it has to start from singleton patterns every time and evaluate them individually. The sampling process, in contrast, immediately samples larger seeds with high deviation. It then locally optimizes the interestingness by pruning. Therefore, it can quickly find high deviation patterns with more than 1 descriptor, while beam search often is still enumerating patterns with 1 descriptor. Surprisingly, especially for Twitter 34, singleton patterns show already high interestingness, because of their frequency.

As expected, the unpruned versions performs slightly worse, because the frequency for sampled patterns is lacking. Comparing uniform (R) to pseudo-randomized (IP), we see that neither of the two is really able to outperform the other. One could argue that the pseudo-randomized version is a bit better at providing qualitative patterns more consistently.



(a) Adult (fnlwgt, capital-loss)

(b) Bank (balance, previous)

(c) Twitter 34 (NSD_0, CS_0)

(d) Cover Type (elevation, slope)

Fig. 4: Max qualities for 2 target exceptional mean shift deviation models.

4.2 Mean Model Quality

Here we used the same setup as before: i.e., we assume an interactive system with limited time budgets (see Table 1) and ran algorithms with 2 pre-defined targets. We ran beam search (BS) with beam sizes of 1, 5 and 10. For sampling we used 2 weighted positive and 1 weighted negative factor using the PCA method (implementation provided by WEKA [11]). We used 4 settings: $MS(W + P)$ – weighting and pruning, $MS(W)$ – weighting and no pruning, $MS(P)$ – no weights and pruning and, at last, MS – no weights and no pruning. Aggregated results of maximum quality patterns over 10 runs are summarized in Figure 4.

The results are similar to Section 4.1: beam search is often not able to get past the singleton pattern phase, and all runs provide similar quality. Also the Twitter dataset shows again that singleton patterns score very high due to their frequency. For the sampling methods, we see that the weighted variants are a bit better. A reason for the sometimes marginal difference is that the weights may be counterproductive. Comparing pruning and non-pruning, we often find a larger gap in favor of pruning, except for Twitter. The main reason is that our distribution is not optimizing enough the frequency of patterns, and only a few large seeds, with high deviation, are sampled and then pruned. In contrast, the non-pruned method more often samples small seeds with high frequency.

5 Related Work

The discovery of interesting subgroups is a topic for several years already and our main objective is not to give an elaborate study on different techniques for finding subgroups. We kindly refer the reader to the overview work by Herrera et al. [12] for more information. Exceptional model mining, on the other hand, is relatively new and can be seen as an extension to subgroup discovery, where models are induced over more than one target. For more information regarding exceptional model mining we refer the reader to the Duivesteijn's thesis [8].

Our main focus are techniques enabling the instant discovery of patterns. Sampling from the output space is an area that has attracted attention only recently. Chaoji et al. [7] use a randomized MCMC process for finding maximal subgraphs in graph databases. Their method is biased towards larger subgraphs, but they use heuristics to overcome this bias. Also on graphs, Al Hasan and Zaki [1] use Metropolis-Hastings to enable uniform sampling of maximal graphs. Moens and Goethals [13] proposed a method similar to the one by Chaoji et al. for sampling the border of maximal itemsets.

At last, we give a short overview of recent exploratory data mining tools, that have high demands wrt responsiveness. MIME [10] allows a user to interact with data directly by letting her create patterns and pattern collections that are evaluated on-the-fly. Moreover, different data mining algorithms can be applied and as their results become input to the user, she can adapt the results at will. Boley et al. [5] propose a framework combining multiple data mining algorithms in a black box environment, alleviating the user from the process of choosing pattern mining methods to apply. They employ a user preference model, based on user interactions, which influence running times for the black boxes. Dzuyba et al. [9] use beam search as their underlying method for finding interesting subgroups. Users then provide feedback on generated patterns, to give more/less importance to specific branches in the search tree.

6 Conclusion and Future Work

Existing methods for finding exceptional models, fail to produce instant results required for interactive discovery processes. In this work, we extended Controlled Direct Pattern Sampling with weights for individual data records and used the framework to directly sample exceptional models using short time budgets.

We showed in our experiments that sampling is able to find better quality patterns in settings that where previously out of reach for beam search. We also showed that by optimizing sampled patterns locally, the quality of patterns can be improved even more. Moreover, we showed that our new weighting scheme can push sampled models into higher quality parts of the search space. However, the weighting can also have a negative effect, when instantiated improperly.

At last we point out future research directions for this research. An important step is extending the mean shift model to more than 3 attributes. The current framework uses the first component by PCA to obtain the highest variance direction, and next samples patterns that lie on the poles of this direction. However, when increasing the number of attributes, using only the first component is not enough and using more components is not optimizing the deviation enough in practice. Different strategies for partitioning the data in positive and negative parts with proper weight assignments is an important issue.

References

1. Mohammad Al Hasan and Mohammed J. Zaki. Output space sampling for graph patterns. *Proc. VLDB Endow.*, pages 730–741, 2009.
2. K. Bache and M. Lichman. UCI machine learning repository, 2013.
3. Axel Blumenstock, Jochen Hipp, Steffen Kempe, Carsten Lanquillon, and Rüdiger Wirth. Interactivity closes the gap. In *Proc. ACM SIGKDD'06 Workshop on Data Mining for Business Applications*, 2006.
4. Mario Boley, Claudio Lucchese, Daniel Paurat, and Thomas Gärtner. Direct local pattern sampling by efficient two–step random procedures. In *Proc. ACM SIGKDD'11*, 2011.
5. Mario Boley, Michael Mampaey, Bo Kang, Pavel Tokmakov, and Stefan Wrobel. One click mining: Interactive local pattern discovery through implicit preference and performance learning. In *Proc. ACM SIGKDD'13 Workshop IDEA*, pages 27–35. ACM, 2013.
6. Mario Boley, Sandy Moens, and Thomas Gärtner. Linear space direct pattern sampling using coupling from the past. In *Proc. ACM SIGKDD'12*, pages 69–77. ACM, 2012.
7. Vineet Chaoji, Mohammad Al Hasan, Saeed Salem, Jeremy Besson, and Mohammed J. Zaki. Origami: A novel and effective approach for mining representative orthogonal graph patterns. *Stat. Anal. Data Min.*, pages 67–84, 2008.
8. Wouter Duivesteijn. *Exceptional model mining*. PhD thesis, Leiden Institute of Advanced Computer Science (LIACS), Faculty of Science, Leiden University, 2013.
9. Vladimir Dzyuba and Matthijs van Leeuwen. Interactive discovery of interesting subgroup sets. In *Advances in Intelligent Data Analysis XII*, pages 150–161. Springer, 2013.
10. Bart Goethals, Sandy Moens, and Jilles Vreeken. Mime: a framework for interactive visual pattern mining. In *Proc. ACM SIGKDD'11*, pages 757–760. ACM, 2011.
11. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.
12. Francisco Herrera, Cristóbal José; Carmona, Pedro González, and María José del Jesus; del Jesus. An overview on subgroup discovery: Foundations and applications. *Knowl. Inf. Syst.*, pages 495–525, 2011.
13. Sandy Moens and Bart Goethals. Randomly sampling maximal itemsets. In *Proc. ACM SIGKDD'13 Workshop IDEA*, pages 79–86, 2013.
14. Radek Škrabal, Milan Šimůnek, Stanislav Vojíř, Andrej Hazucha, Tomáš Marek, David Chudán, and Tomáš Kliegr. Association rule mining following the web search paradigm. In *Proc ECML-PKDD'12*, pages 808–811. Springer-Verlag, 2012.