Non-redundant Subgroup Discovery in Large and Complex Data

Matthijs van Leeuwen[1] and Arno Knobbe[2]

[1] Dept. of Information & Computing Sciences, Universiteit Utrecht, The Netherlands
[2] LIACS, Universiteit Leiden, The Netherlands
mleeuwen@cs.uu.nl, knobbe@liacs.nl

Abstract. Large and complex data is challenging for most existing discovery algorithms, for several reasons. First of all, such data leads to enormous hypothesis spaces, making exhaustive search infeasible. Second, many variants of essentially the same pattern exist, due to (numeric) attributes of high cardinality, correlated attributes, and so on. This causes top-k mining algorithms to return highly redundant result sets, while ignoring many potentially interesting results.

These problems are particularly apparent with Subgroup Discovery and its generalisation, Exceptional Model Mining. To address this, we introduce *subgroup set mining*: one should not consider individual subgroups, but sets of subgroups. We consider three degrees of redundancy, and propose corresponding heuristic selection strategies in order to eliminate redundancy. By incorporating these strategies in a beam search, the balance between exploration and exploitation is improved.

Experiments clearly show that the proposed methods result in much more diverse subgroup sets than traditional Subgroup Discovery methods.

1 Introduction

In this paper, we assume that we are dealing with complex data. This complexity can be due to several aspects of the data, e.g. datasets may contain many rows as well as many attributes, and these attributes may be of high cardinality. Such complex data is challenging for existing discovery algorithms, primarily for reasons of computation time: all these aspects will have an impact on the time required for mining the data. Especially where numeric data is concerned, detailed analysis of the data will imply high cardinalities on such attributes, and many candidate hypotheses will need to be tested. Also, complexity may reside in the discovery task, for example when modelling non-trivial interactions between attributes. The result of these challenges is that individual candidate testing becomes very time-consuming, and hypothesis spaces become prohibitively large.

In the majority of discovery algorithms, including those for Subgroup Discovery (SD) [6,17], it is assumed that complete solutions to a particular discovery task are required, and thus some form of exhaustive search is employed. In order to obtain efficiency, these algorithms typically rely on top-down search combined

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part III, LNAI 6913, pp. 459–474, 2011.

with considerable pruning, exploiting either anti-monotonicity of the quality measure (e.g. frequency), or so-called optimistic estimates of the maximally attainable quality at every point in the search space [3]. With small datasets and simple tasks, these tricks work well and give complete solutions in reasonable time. However, on the complex datasets that we assume, exhaustive approaches simply become infeasible, even when considerable pruning can be achieved. Additionally, we consider Exceptional Model Mining (EMM) [11,10], which allows multiple target attributes and complex models to be used for measuring quality. With EMM in particular, we are often dealing with quality measures that are not monotonic, and for which no optimistic estimates are available.

Apart from the computational concerns with discovery in large datasets, one also needs to consider the practicality of complete solutions in terms of the size of the output. Even when using condensed representations [13,14] or some form of pattern set selection [1,7,15] as a post-processing step, the end result may still be unrealistically large, and represent tiny details of the data overly specifically. The experienced user of discovery algorithms will recognise the large level of redundancy that is common in the final pattern set. This redundancy is often the result of dependencies between the (non-target) attributes, which lead to large numbers of variations of a particular finding. Note that large result sets are problematic even in top-k approaches. Large result sets are obviously not a problem in top-1 approaches, but they are when $k \geq 2$, as the mentioned dependencies will lead to the top of the pattern list being populated with different variations on the same theme, and alternative patterns dropping out of the top-k. This problem is aptly illustrated by Figure 1, which shows that the top-100 subgroups obtained on *Credit-G* cover almost exactly the same tuples.

Approach and contributions. The obvious alternative to exhaustive search, and the one we consider in this paper, is of course *heuristic search*: employ educated guesses to consider only that fraction of the search space that is likely to contain the patterns of interest. When performing heuristic search, it is essential to achieve a good balance between *exploitation* and *exploration*. In other words, to focus and extend on promising areas in the search space, while leaving room for several alternative lines of search. In this work, we will implement this balance by means of *beam search*, which provides a good mixture between parallel search (exploration) and hill-climbing (exploitation). Within the beam search framework, we will experiment with different variations of achieving diversity in the *beam*, that is, the current list of candidates to be extended. Due to the above-mentioned risk of redundancy with top-k selection, the level of exploration

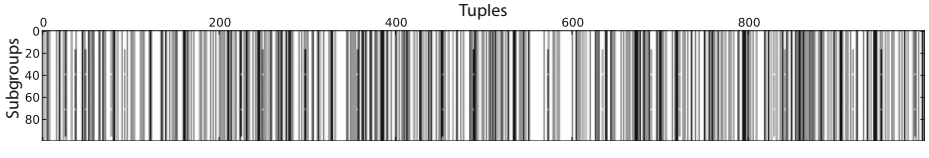

Fig. 1. Redundancy in top-k Subgroup Discovery. Shown are the covers (in *black*) of the top-100 subgroups obtained on *Credit-G* with weighted relative accuracy.

within a beam can become limited, which will adversely affect the quality of the end result. Inspiration for selecting a diverse collection of patterns for the beam at each search level will come from pattern set selection techniques, which were originally designed for post-processing the end-result of discovery algorithms.

In Section 2, we will first formalise both Subgroup Discovery and Exceptional Model Mining, after which we will recap the commonly used search techniques, including the standard beam search algorithm. We will then introduce the notion of *subgroup set mining* in Section 3, and argue that it is better to mine subgroup sets rather than individual subgroups, to ensure diversity. This leads to the *Non-Redundant Generalised Subgroup Discovery* problem statement. We will show that redundancy in subgroup sets can be formalised in (at least) three different ways, each subsequent definition being more strict than its predecessor. Each of these three degrees of redundancy is used as basic principle for a beam selection strategy in Section 5. Section 4 presents the quality measures that will be used in the experiments, which are presented in Section 6. We round up with related work and conclusions in Sections 7 and 8.

2 Preliminaries

2.1 Subgroup Discovery and Exceptional Model Mining

We assume that the tuples to be analysed are described by a set of attributes A, which consists of k *description attributes* D and l *model (or target) attributes* M ($k \geq 1$ and $l \geq 1$). In other words, we assume a supervised setting, with at least a single target attribute M_1 (in the case of classical SD), but possibly multiple attributes $M_1, \ldots, M_l$ (in the case of EMM). Each attribute D_i (resp. M_i) has a domain of possible values $\mathrm{Dom}(D_i)$ (resp. $\mathrm{Dom}(M_i)$). Our dataset $\mathcal{S}$ is now a bag of tuples t over the set of attributes $A = \{D_1, \ldots, D_k, M_1, \ldots, M_l\}$. We use x^D resp. x^M to denote the projection of x onto its description resp. model attributes, e.g. $t^D = \pi_D(t)$ in case of a tuple, or $\mathcal{S}^M = \pi_M(\mathcal{S})$ in case of a bag of tuples. Equivalently for individual attributes, e.g. $\mathcal{S}^{M_i} = \pi_{M_i}(\mathcal{S})$.

Arguably the most important concept in this paper is the *subgroup*, which consists of a *description* and corresponding *cover*. A *subgroup (cover)* is a bag of tuples $G \subseteq \mathcal{S}$ and $|G|$ denotes its size, also called *subgroup size* or *coverage*.

A *subgroup description* is an indicator function s, as a function of description attributes D. That is, it is a function $s : (\mathrm{Dom}(D_1) \times \ldots \times \mathrm{Dom}(D_k)) \mapsto \{0, 1\}$, and its corresponding subgroup cover is $G_s = \{t \in \mathcal{S} \mid s(t^D) = 1\}$. As is usual, in this paper a subgroup description is a *pattern*, consisting of a conjunction of conditions on the description attributes, e.g. $D_x = \mathit{true} \wedge D_y \leq 3.14$. Such a pattern implies an indicator function as just defined.

Given a subgroup G, we would like to know how interesting it is, looking only at its model (or target) data G^M. We quantify this with a quality measure. A *quality measure* is a function $\varphi : \mathcal{G}^M \mapsto \mathbb{R}$ that assigns a numeric value to a subgroup $G^M \subseteq \mathcal{S}^M$, with $\mathcal{G}^M$ the set of all possible subsets of $\mathcal{S}^M$.

Subgroup Discovery and Exceptional Model Mining. The above definitions allow us to define the two main variations of data mining tasks that feature

in this paper: Subgroup Discovery (SD) and Exceptional Model Mining (EMM). As mentioned, in SD we consider datasets where only a single model attribute M_1 (the target) exists. We are interested in finding the top-ranking subgroups according to a quality measure φ that determines the level of interestingness in terms of unusual distribution of the target attribute M_1:

Problem 1 (Top-k Subgroup Discovery). Suppose we are given a dataset $\mathcal{S}$ with $l = 1$, a quality measure φ and a number k. The task is to find the k top-ranking subgroups $\mathcal{G}_k$ with respect to φ.

EMM is a generalisation of the well-known SD paradigm, where the single target attribute is replaced by a collection of model attributes [11]. Just like in SD, EMM is concerned with finding subgroups that show an unusual distribution of the model attributes. However, dependencies between these attributes may occur, and it is therefore desirable to consider the joint distribution over $M_1, \ldots, M_l$. For this reason, modelling over G^M is employed to compute a value for φ. If the model induced on G^M is substantially different from the model induced on $\mathcal{S}^M$, quality is high and we call this an *exceptional model*. We can now formally state the EMM problem.

Problem 2 (Top-k Exceptional Model Mining). Suppose we are given a dataset $\mathcal{S}$, a quality measure φ and a number k. The task is to find the k top-ranking subgroups $\mathcal{G}_k$ with respect to φ.

2.2 Subgroup Search

To find high-quality subgroups, the usual choice is a top-down search strategy. The search space is traversed by starting with simple descriptions and refining these along the way, from general to specific. For this a *refinement operator* that specialises subgroup descriptions is needed. A *minimum coverage* threshold (*mincov*) is used to ensure that a subgroup covers at least a certain number of tuples. A *maximum depth* (*maxdepth*) parameter imposes a maximum on the number of conditions a description may contain.

Exhaustive search. When exhaustive search is possible, depth-first search is commonly used. This is often the case with moderately sized nominal datasets with a single target. Whenever possible, (anti-)monotone properties of the quality measure are used to prune parts of the search space. When this is not possible, so-called *optimistic estimates* can be used to restrict the search space. An optimistic estimate function computes the highest possible quality that any refinement of a subgroup could give. If this upper bound is lower than the quality of the current kth subgroup, this branch of the search space can be safely ignored.

Beam search. When exhaustive search is not feasible, beam search is the widely accepted heuristic alternative. It also uses a levelwise top-down strategy and the same refinement operator, but it explores only part of the search space. The basic algorithm is shown as Algorithm 1. On each level, the w highest ranking subgroups with respect to quality are selected for the *beam*. Candidate subgroups for the next level are generated from individual subgroups b using the refinement

Algorithm 1. Beam Search

Input: A dataset $\mathcal{S}$, a quality measure φ and parameters k, w, *mincov* and *maxdepth*.
Output: $\mathcal{R}$, an approximation of the top-k subgroups $\mathcal{G}_k$.

```
1. R ← ∅, Beam ← {∅}, depth = 1
2. while depth ≤ maxdepth do
3.    Cands ← ∅
4.    for all b ∈ Beam do
5.       Cands ← Cands ∪ GenerateRefinements(b, mincov)
6.    for all c ∈ Cands do
7.       UpdateTopK(R, k, c, φ(c))
8.    Beam ← SelectBeam(Cands, w, φ)
9.    depth ← depth + 1
10. return R
```

operator (*GenerateRefinements*), while respecting the *mincov* parameter. The initial candidate set is generated from the empty subgroup description. *SelectBeam* selects the w highest ranking $c \in Cands$ (with respect to φ) to form the beam for the next level.

3 Non-Redundant Generalised Subgroup Discovery

The redundancy issues experienced with SD/EMM algorithms suggest that we should not only look at each individual subgroup locally, but also take the other subgroups into account. That is, we should consider *subgroup set mining*, similar to recent pattern set selection approaches [1,7,15].

Problem 3 (Non-Redundant Generalised Subgroup Discovery). Suppose we are given a dataset $\mathcal{S}$, a quality measure φ and a number k. The task is to find a non-redundant set $\mathcal{G}$ of k high-quality subgroups.

The term *Generalised Subgroup Discovery* is used to emphasise that it encompasses both SD and EMM.

Although it may be clear to the data miner whether a (small) set of patterns contains redundancy or not, formalising redundancy is no trivial task. We can consider three degrees of redundancy removal.

In a *non-redundant subgroup set* $\mathcal{G}$, all pairs $G_i, G_j \in \mathcal{G}$ (with $i \neq j$) should have substantially different:

1. *subgroup descriptions*, or
2. *subgroup covers*, or
3. *exceptional models*. (Only in the case of EMM.)

Note that each subsequent degree is more strict than its predecessor. On the first, least restrictive degree, substantially different descriptions are allowed, ignoring any potential similarity in the cover. The second degree of redundancy would also address this kind of similarity in the subgroup covers. The third degree of redundancy will consider subgroups that are different in both description and

cover, and will address their difference in terms of the associated models built on the model attributes M.

In Section 5, each of the three degrees of redundancy will be used as basic principle for a subgroup set selection method and be incorporated in the beam search. The resulting search strategies eliminate redundancy in subgroup sets.

To quantify redundancy in subgroup sets, we consider the subgroup covers because it is independent of any other choices and can be easily interpreted. By assuming a uniform distribution of all subgroup covers over all tuples in the dataset, we can compute an *expected cover count* and measure how far each individual tuple's cover count deviates from this. This results in the following.

Definition 1 (Cover Redundancy). *Suppose we are given a dataset $\mathcal{S}$ and a set of subgroups $\mathcal{G}$. Define the* cover count *of a tuple $t \in \mathcal{S}$ as $c(t, \mathcal{G}) = \sum_{G \in \mathcal{G}} s_G(t)$. The* expected cover count *$\hat{c}$ of a random tuple $t \in \mathcal{S}$ is defined as $\hat{c} = \frac{1}{|\mathcal{S}|} \sum_{t \in \mathcal{S}} c(t, \mathcal{G})$. The* Cover Redundancy *CR is now computed as:*

$$CR^{\mathcal{S}}(\mathcal{G}) = \frac{1}{|\mathcal{S}|} \sum_{t \in \mathcal{S}} \frac{|c(t, \mathcal{G}) - \hat{c}|}{\hat{c}}$$

The larger the CR, the larger the deviation from the uniform distribution. Because Generalised Subgroup Discovery aims to find only those parts of the data that stand out, this measure on itself does not tell us much. However, if we have several subgroup sets of (roughly) the same size and for the same dataset, a lower CR indicates that less tuples are covered by more subgroups than expected, and thus the subgroup set is more diverse/less redundant.

As an example, the subgroup set in Figure 1 has a CR of 1.19. Clearly, this cover distribution is highly undesirable and (much) lower values are preferred.

4 Quality Measures

Weighted Relative Accuracy. Weighted Relative Accuracy (*WRAcc*) [9] is a well-known SD quality measure for datasets with one binary target attribute. Let 1^G (resp. $1^{\mathcal{S}}$) denote the fraction of ones in the target attribute, within the subgroup (resp. entire dataset). Weighted Relative Accuracy is then defined as $\varphi_{WRAcc}(G) = \frac{|G|}{|\mathcal{S}|}(1^G - 1^{\mathcal{S}})$.

Weighted Kullback-Leibler divergence. We previously [10] introduced a measure based on the Kullback-Leibler (KL) divergence. Each attribute-value is assumed to be an independently drawn sample from an underlying random variable. The empirical probability distribution for attribute M_i is estimated by $\hat{P}$. We here present an alternative that weighs quality by subgroup size, because this works better in combination with a levelwise search (without this weight, smaller subgroups always tend to have larger qualities). This measure can be used with either a single or multiple binary model attributes, and even with nominal attributes.

Definition 2 (WKL quality). *Given a database $\mathcal{S}$ and subgroup G, define (independent)* Weighted KL quality *as*

$$\varphi_{\text{WKL}}(G^M) = \frac{|G|}{|\mathcal{S}|} \sum_{i=1}^{l} \text{KL}(\hat{P}(G^{M_i}) \parallel \hat{P}(\mathcal{S}^{M_i}))$$

Weighted Krimp Gain. In [10] we introduced a second measure that, contrary to (Weighted) KL quality, does take associations between (binary) attributes into account. It uses KRIMP code tables [16] as models, but the principle is equivalent to that of *WKL*: a subgroup is interesting if it can be compressed much better by its own compressor, than by the compressor induced on the overall database. Similar to *WKL* quality, we here introduce a weighted alternative.

Definition 3 (Weighted Krimp Gain). *Let $\mathcal{D}$ be a binary database, $G \subseteq \mathcal{D}$ a subgroup, and $CT_{\mathcal{D}}$ and CT_G their respective optimal code tables. We define the* Weighted Krimp Gain *of group G from $\mathcal{D}$, denoted by* $\text{WKG}(G \parallel \mathcal{D})$*, as*

$$\text{WKG}(G \parallel \mathcal{D}) = L(G \mid CT_{\mathcal{D}}) - L(G \mid CT_G),$$

with $L(G \mid CT)$ the size of G, in bits, encoded with code table CT.

Given this, defining the quality measure is straightforward.

Definition 4 (WKG quality). *Let $\mathcal{S}$ be a database and $G \subseteq \mathcal{S}$ a subgroup. Define* Weighted KG quality *as* $\varphi_{\text{WKG}}(G^M) = \text{WKG}(G^M \parallel \mathcal{S}^M)$.

5 Non-Redundant Beam Selection

In this section we show how selection strategies based on the three degrees of redundancy from Section 3 can be incorporated in the basic beam search algorithm (see Algorithm 1). Instead of simply choosing the –potentially highly redundant– top-k subgroups for the beam, we will modify the algorithm to select diverse *subgroup sets* at each level. In other words, we strive to achieve high-quality yet non-redundant beam selection.

A *beam selection strategy* is a selection scheme that decides which candidates are included in the beam, and is invoked by *SelectBeam* in Algorithm 1. We will refer to regular top-k beam selection as the *Standard* strategy.

Most pattern set selection criteria require all possible pattern sets to be taken into consideration to ensure that the global optimum is found. However, large numbers of subgroups may be evaluated at each search level and such exhaustive strategies are therefore infeasible. Hence, we have to resort to greedy and heuristic methods, as is usual in pattern set selection [1,7,15]. The following three selection strategies correspond to the three degrees of redundancy.

Description-based beam selection. Order all candidates descending by quality and consider them one by one until beam width w is reached. For each considered subgroup $G \in Cands$, discard it if its quality and all but 1 conditions

are equal to that of any $b \in Beam$, otherwise include it in the beam. Time complexity for selecting the beam of a single level is $\mathcal{O}(|Cands| \cdot log(|Cands|) + |Cands| \cdot depth)$ (the current search *depth* influences how long a comparison of descriptions takes).

Cover-based beam selection, This strategy focuses on the subgroup covers and how they overlap. A score based on multiplicative weighted sequential covering [9] is used to weigh the quality of each subgroup, aiming to minimise the overlap between the selected subgroups. This score is defined as

$$\Omega(G, Beam) = \frac{1}{|G|} \sum_{t \in G} \alpha^{c(t, Beam)},$$

where $\alpha \in \langle 0, 1]$ is the weight parameter. The less often tuples in subgroup G are already covered by subgroups in the beam, the larger the score. If the cover contains only previously uncovered tuples, $\Omega(G, Beam) = 1$.

In w iterations, w subgroups are selected for inclusion in the beam. In each iteration, the subgroup that maximises $\Omega(G, Beam) \cdot \varphi(G)$ is selected. The first selected subgroup is always the one with the highest quality, since the beam is empty and $\Omega(G, Beam) = 1$ for all G. After that, the Ω-scores for the remaining $Cands$ are updated each iteration. Complexity per level is $\mathcal{O}(w \cdot |Cands| \cdot |\mathcal{S}|)$.

Compression-based beam selection. To be able to do model-based beam selection, a (dis)similarity measure on models is required. For this purpose, we focus on the models used by the *WKL* and *WKG* quality measures. These measures have in common that they rely on *compression*; they assume a coding scheme and the induced models can therefore be regarded as *compressors*.

In case of *WKG*, the compressor is the code table induced by Krimp. In case of *WKL*, the compressor replaces each attribute-value x by a code of optimal length $L(x)$ based on its marginal probability, i.e. $L(x) = -\log_2(\hat{P}(M_i = x))$.

Adopting the MDL philosophy [4], we say that the best set of compressors is that set that together compresses the dataset best. Selecting a set of compressors is equivalent to selecting a set of subgroups, since each subgroup has exactly one corresponding compressor. Since exhaustive search is infeasible, we propose the following heuristic.

1. We start with the 'base' compressor that is induced on the entire dataset, denoted C^S. Each $t \in S$ is compressed with this compressor, resulting in encoded size $L(S \mid C^S)$.
2. Next, we iteratively search for the subgroup that improves overall compression most, relative to the compression provided by the subgroups already selected. That is, the first selected subgroup is always the top-ranked one, since its compressor C^1 gives the largest gain with respect to $L(S \mid C^S)$.
3. Each transaction is compressed by the last picked subgroup that covers it, and by C^S if it is not yet covered by any. So, after the first round, part of the transactions are encoded by C^S, others by C^1.

4. Assuming this encoding scheme, select that subgroup $G \in Cands \setminus \{C^1, \dots\}$ that maximises $L(S \mid C^S, C^1, \dots) - L(S \mid C^S, C^1, \dots, G)$ in each subsequent step. Stop when the beam has attained its desired width w.

To perform this selection strategy, all compressors belonging to the subgroups of a certain level are required. If these can be kept in memory, the complexity of the selection scheme is $\mathcal{O}(w \cdot |Cands| \cdot |\mathcal{S}| \cdot |M|)$, where $|M|$ is the number of model attributes. However, keeping all compressors in memory may not be possible. They could then be either cached on disk or reconstructed on demand, but both approaches would severely impact runtimes.

Each subsequent beam selection strategy is more strict than its predecessor, but also computationally more demanding. This offers the data miner the opportunity to trade-off diversity with computation time.

5.1 Improving Individual Subgroups

Despite all efforts to prevent and eliminate redundancy in the result set, some of the found subgroups may be overly specific. This may be caused by a large search depth, but also by heuristic choices in e.g. the refinement operator. For example, the subgroup corresponding to $A = true \wedge B = true$ might have the highest possible quality, but never be found since neither $A = true$ nor $B = true$ has high quality. However, $C = false \wedge A = true \wedge B = true$ could be found. Now, *pruning* the first condition would give the best possible subgroup.

We propose to improve individual subgroups by pruning the subgroup descriptions as a post-processing step, based on the concept of *dominance*. A subgroup G_i *dominates* a subgroup G_j iff

1. the conditions of the description of G_i are a strict subset of those of G_j, and
2. the quality of G_i is higher than or equal to that of G_j, i.e. $\varphi(G_i) \geq \varphi(G_j)$.

Observe that although dominance is clearly inspired by relevancy [2], it is not the same. The former is more generic, making it also suitable for e.g. EMM.

The heuristic method we propose for *dominance-based pruning* is to consider each of the conditions in a subgroup description one by one, in the order in which they were added. If removing a condition does not decrease the subgroup's quality, then permanently remove it, otherwise keep it.

5.2 Non-Redundant Beam Search

The overall subgroup set mining process we propose consists of three steps. First, a beam search (Algorithm 1) is performed to mine N subgroups, with any of the proposed beam selection strategies (plugged in as *SelectBeam*). Next, each of the N resulting subgroups is individually pruned based on dominance, and syntactically equivalent subgroups are removed. As the final result set potentially also suffers from the redundancy problems of top-k-selection, a selection strategy is used to select S subgroups ($S \ll N$) from the remaining subgroups ('post-selection'). For this, the same strategy as during the beam search is used.

Refinements. We distinguish three types of description attributes, each with its own associated condition types: $\{=\}$ for binary, $\{=, \neq\}$ for nominal and $\{<, >\}$ for numeric attributes. For binary and nominal attributes, the refinement operator always generates all possible refinements, i.e. each combination of condition type and attribute-value. To prevent the search space from exploding, the values of a numeric attribute are locally binned into 6 equal-sized bins and $\{<, >\}$-conditions are generated for the 5 split points obtained this way. This 'on-the-fly' discretisation, performed upon subgroup refinement, results in a much more fine-grained binning than 'a priori' discretisation of numeric attributes.

Except for refinements that lead to a contradiction, all refinements for all description attributes are always considered. (Adding $D_x = true$ to a description that already contains $D_x = false$ would be meaningless, for example.) Consequently, multiple conditions on the same attribute can be imposed; especially with nominal attributes, slowly peeling off tuples with $\neq$ can be helpful.

6 Experiments

Datasets. To evaluate the proposed methods, we perform experiments on the datasets listed in Table 1. The upper three datasets, taken from the UCI repository[1], contain a single target (SD), the lower four datasets have multiple model attributes (EMM). Two variants of the UCI *Adult* dataset are used: *Adult-SD* is the commonly used variant, with the binary class label as single target, in *Adult-EMM* all numeric attributes are considered as description attributes, and all binary attributes as model attributes (except for class, which is not used). Furthermore, we take the *Emotions* and *Yeast* datasets from the 'Mulan' repository[2], and we use the *Mammals* dataset [5] (each of these has numeric description attributes and binary model attributes).

Table 1. Datasets. For each dataset the number of tuples, the number of description and model attributes, and the *minsup* used for *WKG* are given.

Dataset	*Properties*			*WKG*
	$\lvert\mathcal{S}\rvert$	$\lvert D\rvert$	$\lvert M\rvert$	minsup
Adult-SD	48842	105	1	-
Credit-G	1000	20	1	-
Mushroom	8124	22	1	-
Adult-EMM	48842	6	99	10%
Emotions	593	72	6	1%
Mammals	2221	67	124	-
Yeast	2417	103	14	1%

Methods for comparison. Depth-first search (DFS) is used, with *WRAcc* in combination with tight optimistic estimate [3]. With DFS, only a single condition per attribute is allowed and all attributes are considered in a fixed order. This is necessary to limit the size of the search space and thus computation time, but also means that beam search can potentially reach better solutions.

[1] `http://archive.ics.uci.edu/ml/`
[2] `http://mulan.sourceforge.net/datasets.html`

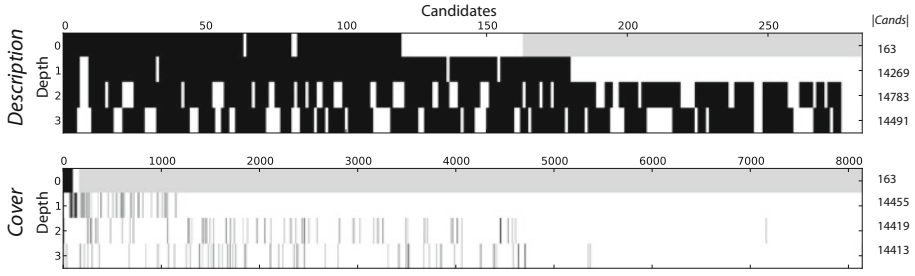

Fig. 2. Two beam selection strategies in action: description-based and cover-based. For each level in the beam search, it is shown which candidate subgroups are selected for inclusion in the beam (*black*) and which are ignored (*white*). Candidates are ordered descending on quality. On the right, the total number of candidate subgroups for each level is shown (candidates not shown are not selected). *Credit-G* with *WRAcc*.

An often adopted approach to mining pattern sets is the 2-step approach, where 1) all patterns are mined and 2) a subset of these patterns is selected as post-processing step. We test this approach by first using DFS or standard beam search to mine the top-N subgroups, and then use cover-based selection to select S subgroups from this (denoted '+PS', for post-selection).

Search parameters. In all experiments, $N = 10,000$ subgroups are mined, from which $S = 100$ are selected for the final subgroup set. A maximum depth $maxdepth = 5$, minimum coverage $mincov = 10$, and beam width $w = 100$ are used. Preliminary experiments showed that changing these parameters has the same effect on all search strategies, keeping their differences intact. Since our aim is to compare the different strategies, we keep these fixed. Weight parameter α for cover-based beam selection is set to 0.9, since preliminary experiments indicated that this gives a good balance between quality and cover diversity.

6.1 A Characteristic Experiment in Detail

To study the effects of the proposed beam selection strategies and dominance-based pruning in detail, we focus on a single dataset. For ease of presentation, we choose the (relatively small) *Credit-G* dataset, and we use *WRAcc* as quality measure. We choose a classical Subgroup Discovery setting because it is studied and used by so many people, but this means that we cannot apply compression-based selection. In Figure 1 we have already seen that redundancy is a tremendous problem with DFS top-k subgroup discovery. Hence, we will now apply the proposed beam selection strategies to see if this improves diversity.

Figure 2 shows which subgroups are selected for refinement on each level in the beam search. Clearly, the description-based and cover-based strategies select subgroups from a much wider range than the standard top-100, which is likely to result in a more diverse beam. As expected, a higher degree of redundancy elimination results in more (high-quality but similar) candidates being skipped.

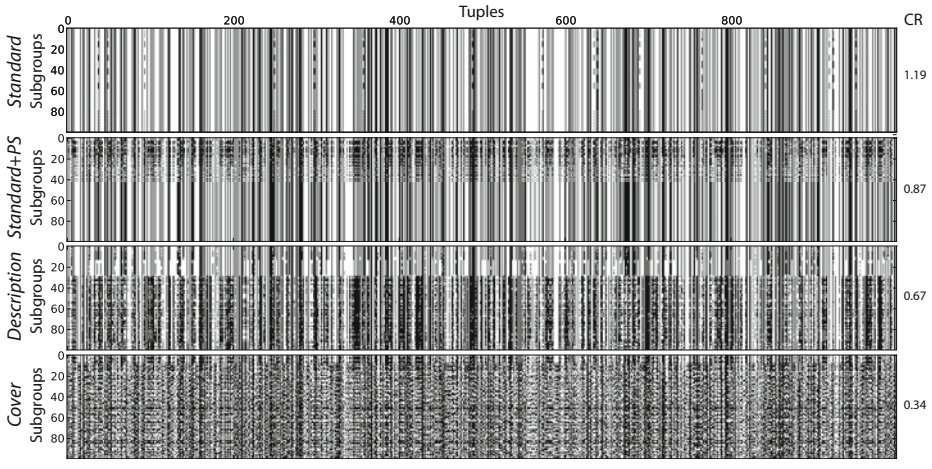

Fig. 3. Subgroup covers obtained with 4 beam search strategies: standard, standard with cover-based post-selection, description-based, and cover-based. Shown are the covers (in *black*) of the top-100 subgroups obtained on *Credit-G* with *WRAcc*. Cover Redundancies (CR) computed from the subgroup sets are shown on the right.

Our hypothesis, of course, is that this more diverse beam selection also results in a more diverse set of results. To assess this, consider the subgroup covers of the 100 subgroups that are obtained after post-selection, in Figure 3. The plots confirm that diversity increases as higher degree redundancy is eliminated: subgroup covers become more and more scattered over all tuples, and CR decreases with each new strategy (from top to bottom). Post-selection seems to perform well at first with *Standard+PS*, but after choosing about 40 subgroups there are no diverse and high-quality candidates left in the remaining 9,960 subgroups, and homogeneity is the end result.

The goal we stated in Section 3 is to find a non-redundant set of high-quality subgroups. It is therefore important that the maximum quality of a subgroup set, the highest quality obtained by any subgroup, does not decrease when using our beam selection strategies.

To assess this, consider the qualities of the 100 subgroups that are obtained after post-selection, in Figure 4. The maximum obtained quality is (almost) the same for all settings, indicating that exploitation does not suffer from beam diversity; a good result. The lower average qualities and larger standard deviations are natural consequences of the diversity enforced by subgroup set selection.

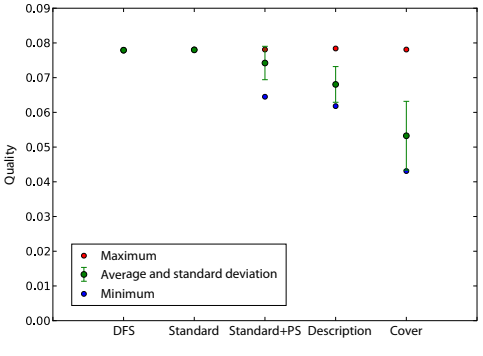

Fig. 4. Qualities of 100 subgroups obtained with different search strategies

6.2 Quantitative Results

We now present results obtained on a large set of experiments, to show that the proposed beam selection strategies have a positive effect in the large majority of cases. That is, resulting subgroup sets are more diverse (and thus less redundant), while not giving in on maximum quality.

For the SD setting, we performed experiments with 3 datasets (*Adult-SD*, *Credit-G* and *Mushroom*), quality measures *WRAcc* and *WKL* and 6 search strategies. These were depth-first search with cover-based post-selection, beam search with a standard beam with and without cover-based post-selection, and beam search with the three proposed selection strategies. The compression-based strategy does not work with *WRAcc*, and DFS with *Adult-SD* and *WKL* was excluded due to a very long runtime (> 2 weeks). Taking this into account, the setup resulted in a total of 26 experiments.

Aggregated results obtained for these experiments are shown in Table 2. A search strategy is better than others if it more often achieves 1) a higher maximum quality, and 2) a lower cover redundancy. This is represented by the average rank results. For each combination of dataset and quality measure, experiments with all search strategies were performed and ranked with respect to 1) maximum quality obtained (φ^{max}, descending), and 2) cover redundancy of the attained subgroup set (CR, ascending). Tied ranks are assigned the average of the ranks for the range they cover. Finally, all ranks for a specific search strategy are averaged.

The results in Table 2 show that DFS with cover-based post-selection needs many candidates and considerable computation time to obtain subgroup sets that are hardly diverse and do not attain the highest maximum quality. The latter is partly due to the restrictions we had to impose on the hypothesis space; multiple conditions on a single attribute (often beneficial) were banned.

The slightly higher average rankings (with respect to maximum quality) of the description-based and cover-based strategies show that diverse beam selection has a modest positive impact on beam search's capability of finding high-quality solutions. A standard beam search with cover-based post-selection gives more diverse results than the description-based strategy, but the latter is faster and it is evidently more diverse than beam search without any post-processing.

Table 2. Subgroup Discovery results, aggregated over 3 datasets and 2 quality measures. Shown are the average number of candidates, time per experiment, subgroup description sizes (#conditions), subgroup sizes and cover redundancies. On the right, average ranks are given as obtained by ranking experiments stratified by strategy.

Search strategy	*Experiment avg*		*Subgroup set avg*			*Rank avg*	
	$\|Cands\|$	time (min)	descr.	size	CR	φ^{max}	CR
DFS + PS	403801872	1553	3.5	5712	1.10	3.4	3.8
Standard	88641	0.3	4.7	6535	1.23	3.2	4.0
Standard + PS	88641	4.2	3.6	7494	0.80	3.2	2.5
Description	88508	1.0	4.3	6591	0.98	2.8	3.7
Cover	89116	49	4.4	8758	0.37	2.8	1.0
Compression	87304	16	2.7	3296	1.12	3.3	3.0

Table 3. Exceptional Model Mining results, aggregated over 4 datasets and 2 quality measures. Shown are the average number of candidates, time per experiment, subgroup description sizes (#conditions), subgroup sizes and cover redundancies. On the right, average ranks are given as obtained by ranking experiments stratified by strategy.

Search strategy	*Experiment avg*		*Subgroup set avg*			*Rank avg*	
	$\|Cands\|$	time (min)	descr.	size	CR	φ^{max}	CR
Standard	244830	8	4.8	4840	1.53	3.1	4.6
Standard + PS	244830	52	3.4	5397	1.07	2.6	2.5
Description	244659	49	3.8	5163	1.36	1.9	3.5
Cover	244830	62	3.4	5493	0.48	3.2	1.2
Compression	255992	143	2.1	653	1.07	3.8	2.4

When the cover-based strategy is incorporated *within* the search, however, the results stand out with respect to cover diversity. The downside is that it needs more time, but it is still very fast when compared to DFS. Compression-based selection does not seem to work well in the SD setting, which is not unexpected since only a very limited number of distributions can be distinguished with a single binary model attribute.

We performed EMM experiments on 4 datasets (*Adult-EMM*, *Emotions*, *Mammals*, and *Yeast*), with quality measures *WKL* and *WKG* and 5 beam search strategies. *WKG* was not used in combination with *Mammals*, since the induction of Krimp code tables takes too long on this dataset; *WKL* is a good and fast alternative. We chose to apply the combination of *WKG* and compression-based selection only to *Emotions*, as all models can be cached in memory for this dataset. The results of the 26 experiments are presented in Table 3.

The results for EMM are slightly different from those for SD. Description-based selection finds better overall solutions than the other strategies. It performs better than *Standard* in terms of cover redundancy, but not better than the 2-step *Standard+PS* approach. For fast mining of high-quality results, the description-based strategy seems a good choice. Dominance-based pruning is not applied with *Standard*, resulting in lower maximum qualities than with *Standard+PS*.

As expected from its basic principle, cover-based selection is again the clear winner with respect to cover diversity: it achieves the lowest cover redundancies. The compression-based scheme gives slightly lower maximum qualities, but the subgroups are quite diverse, smaller and have shorter descriptions.

We performed a Friedman test on 8 rankings obtained with the compression-based quality measures, to be able to include the compression-based strategy in the comparison. For each of the 7 datasets, a ranking was obtained with *WKL*, 1 ranking came from *Emotions* with *WKG*. Between the φ^{max} rankings, no significant differences were found; the 5 strategies exhibit no significant differences with respect to exploitation. In the CR rankings, significant differences were found (p-value = 0.00004), and we did a post-hoc Wilcoxon-Nemenyi-McDonald-Thompson test. *Standard+PS*, *Cover* and *Compression* have significantly better rankings than *Standard*, and *Cover* is also significantly better than *Description*.

All in all, incorporating subgroup selection *within* beam search yields clearly better results than applying it as post-processing step. Employing the description-based selection scheme comes at little computational cost, but does give higher-quality and more diverse results than without using any subgroup selection techniques. At the expense of some more computation time, cover-based selection eliminates more redundancy and results in a much more diverse subgroup set. The compression-based method does not always work well, but should be employed for datasets where many underlying distributions are present in the model data, such as it is the case for e.g. *Mammals*.

Finally, we consider the effect of dominance-based pruning on the subgroup sets. In the SD experiments, the average number of conditions per subgroup description decreases from 4.5 to 3.4 and average subgroup quality increases with 4% on average. For EMM, the effect is even larger and the average number of conditions decreases from 4.9 to 3.0, an average decrease of 1.9 conditions per description! Meanwhile, average subgroup quality increases with 20.3% on average. Note that these changes are due to both the pruning of individual descriptions and the removal of syntactically identical subgroups.

7 Related Work

To the best of our knowledge, we are the first to combine pattern selection techniques and beam search to achieve non-redundant Generalised Subgroup Discovery. Kocev et al. [8] previously proposed to incorporate similarity constraints in a beam search to improve the induction of predictive clustering trees.

Several methods have been proposed to address the redundancy problem in SD/EMM. Garriga et al. [2] proposed closed sets for labeled data, but similar to closed frequent itemsets, this only eliminates a limited part of redundancy as only individual patterns are considered. An advantage is that 'relevant' subgroups can be efficiently mined [12]. A downside is that it does not apply to the EMM setting. We previously proposed the EMDM algorithm [10], but this method does not apply to the SD setting and for the EMM setting, it is dependent on the initial candidates and it finds more complex subgroup descriptions.

The beam selection strategies we propose are clearly inspired by pattern set selection methods such as those proposed by Bringmann & Zimmerman [1] and Peng et al. [15]. The key difference is that we perform pattern selection *within* a discovery algorithm to improve the end result.

8 Conclusions

Effective and efficient heuristics are crucial for performing discovery tasks in large and complex data. In addition to that, the incredible amount of redundancy in hypothesis spaces renders straightforward top-k mining useless. We address these problems by incorporating heuristic pattern set selection methods *within* a beam search, thereby improving the balance between exploration and exploitation.

We described three degrees of redundancy and introduced a subgroup set selection strategy for each degree. Experiments with both Subgroup Discovery

and Exceptional Model Mining show that the proposed methods for *subgroup set mining* return high-quality yet diverse results. The three methods offer the data miner a trade-off between redundancy elimination and computation time.

Acknowledgments. This research is financially supported by the Netherlands Organisation for Scientific Research (NWO) under project number 612.065.822.

References

1. Bringmann, B., Zimmermann, A.: The chosen few: On identifying valuable patterns. In: Proceedings of the ICDM 2007, pp. 63–72 (2007)
2. Garriga, G.C., Kralj, P., Lavrac, N.: Closed sets for labeled data. Journal of Machine Learning Research 9, 559–580 (2008)
3. Grosskreutz, H., Rüping, S., Wrobel, S.: Tight optimistic estimates for fast subgroup discovery. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part I. LNCS (LNAI), vol. 5211, pp. 440–456. Springer, Heidelberg (2008)
4. Grünwald, P.D.: The Minimum Description Length Principle. MIT Press, Cambridge (2007)
5. Heikinheimo, H., Fortelius, M., Eronen, J., Mannila, H.: Biogeography of european land mammals shows environmentally distinct and spatially coherent clusters. J. Biogeography 34(6), 1053–1064 (2007)
6. Klösgen, W.: Explora: A Multipattern and Multistrategy Discovery Assistant. In: Advances in Knowledge Discovery and Data Mining, pp. 249–271 (1996)
7. Knobbe, A., Ho, E.K.Y.: Pattern teams. In: Proceedings of the ECML PKDD 2006, pp. 577–584 (2006)
8. Kocev, D., Struyf, J., Džeroski, S.: Beam search induction and similarity constraints for predictive clustering trees. In: Džeroski, S., Struyf, J. (eds.) KDID 2006. LNCS, vol. 4747, pp. 134–151. Springer, Heidelberg (2007)
9. Lavrač, N., Kavšek, B., Flach, P., Todorovski, L.: Subgroup discovery with cn2-sd. J. Mach. Learn. Res. 5, 153–188 (2004)
10. van Leeuwen, M.: Maximal exceptions with minimal descriptions. Data Min. Knowl. Discov. 21(2), 259–276 (2010)
11. Leman, D., Feelders, A., Knobbe, A.: Exceptional model mining. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 1–16. Springer, Heidelberg (2008)
12. Lemmerich, F., Rohlfs, M., Atzmüller, M.: Fast discovery of relevant subgroup patterns. In: Proceedings of FLAIRS (2010)
13. Mannila, H., Toivonen, H.: Multiple uses of frequent sets and condensed representations. In: Proceedings of the KDD 1996, pp. 189–194 (1996)
14. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Discovering frequent closed itemsets for association rules. In: Beeri, C., Bruneman, P. (eds.) ICDT 1999. LNCS, vol. 1540, pp. 398–416. Springer, Heidelberg (1998)
15. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(8), 1226–1238 (2005)
16. Vreeken, J., van Leeuwen, M., Siebes, A.: Krimp: mining itemsets that compress. Data Mining and Knowledge Discovery 23(1), 169–214 (2011)
17. Wrobel, S.: An algorithm for multi-relational discovery of subgroups. In: Komorowski, J., Żytkow, J.M. (eds.) PKDD 1997. LNCS, vol. 1263, pp. 78–87. Springer, Heidelberg (1997)