Interactive Discovery of Interesting Subgroup Sets

Vladimir Dzyuba and Matthijs van Leeuwen

Department of Computer Science, KU Leuven, Belgium
vladimir.dzyuba@cs.kuleuven.be, matthijs.vanleeuwen@cs.kuleuven.be

Abstract. Although subgroup discovery aims to be a practical tool for exploratory data mining, its wider adoption is hampered by redundancy and the re-discovery of common knowledge. This can be remedied by parameter tuning and manual result filtering, but this requires considerable effort from the data analyst. In this paper we argue that it is essential to involve the user in the discovery process to solve these issues. To this end, we propose an interactive algorithm that allows a user to provide feedback *during search*, so that it is steered towards more interesting subgroups. Specifically, the algorithm exploits user feedback to guide a diverse beam search. The empirical evaluation and a case study demonstrate that uninteresting subgroups can be effectively eliminated from the results, and that the overall effort required to obtain interesting and diverse subgroup sets is reduced. This confirms that within-search interactivity can be useful for data analysis.

Key words: Interactive data mining, pattern set mining

1 Introduction

Informally, subgroup discovery [12,18] is concerned with finding subsets of a dataset that have a substantial deviation in a property of interest when compared to the entire dataset. It can be regarded as an exploratory data analysis task, with a strong emphasis on obtaining comprehensible patterns in the form of subgroup descriptions. In the context of a bank providing loans, for example, we could find that 16% of all loans with *purpose* = *used car* are not repaid, whereas for the entire population the proportion is only 5%. Subgroup discovery algorithms can cope with a wide range of data types, from simple binary data to numerical attributes and structured data. Various quality measures have been proposed to quantify subgroup interestingness, for which generally both the amount of deviation and the size of the subset are taken into account.

Subgroup discovery aims to be a *practical* tool for data exploration, and many case studies on real-world applications have been performed; see Herrera et al. [11] for a recent overview. Unfortunately, obtaining interesting results is usually a time-consuming job for which expertise on subgroup discovery is required. This is due to two main reasons: 1) large amounts of subgroups are found, of which many are redundant, and 2) background knowledge of the domain expert is not taken into account. To remedy these issues, careful tuning of the algorithm parameters and manual filtering of the results is a necessity. This requires considerable effort and expertise from the data analyst, and this clearly hampers the wider adoption of subgroup discovery as a tool for data exploration.

To address the *pattern explosion* in subgroup discovery, Diverse Subgroup Set Discovery (DSSD) [14] was recently proposed in an attempt to attain diverse rather than redundant subgroup sets. The main idea is to integrate pattern set mining into a levelwise search, so that diversity is maintained throughout search. Specifically, heuristic methods for selecting diverse subgroup sets are used to select a beam on each level, resulting in a diverse beam search.

Case Study: Sports Analytics To illustrate the problems of existing methods and the potential of our proposed approach, we investigate the use of subgroup discovery in the context of sports analytics. There has recently been a significant interest in data mining in the professional sports community[1]. 'Black box' approaches that do not explain their outcomes would never be accepted, but subgroup discovery has the advantage that its results are *interpretable*.

The case study concerns a dataset containing information about games played by the Portland Trail Blazers in the 2011/12 season of the NBA[2]. Each tuple corresponds to a segment of a game played by the same group of 10 players (including 5 players on the opposing team). The attributes include 18 binary variables indicating presence of a particular player on the court, a nominal variable representing the opposing team, a numeric attribute pace[3], and 3 binary variables per team indicating whether offensive rating[4] and offensive/defensive rebound rates[5] of a team are higher than the season average.

We select *offensive rating* as the target property of interest, and the commonly used Weighted Relative Accuracy as the quality measure (see Section 5 for further details). This means that high-quality subgroups represent common situations in which the team is likely to have a high offensive rating, described in terms of players, opponents, and the course of the game.

To assess whether Diverse Subgroup Set Discovery gives satisfactory results, we ran the algorithm on the NBA data with default settings (cover-based heuristic with default quality-diversity trade-off [14]). We asked for the discovery of five subgroups, which are all shown in Table 1. The results suffer from two severe problems: 1) the results are clearly redundant, i.e. diversity could not be attained with the default parameter settings, and 2) none of the discovered subgroups is interesting to a domain expert, as the descriptions contain no surprising and/or actionable information. For example, it is a trivial fact for experts that poor defensive rebounding by an opponent ($opp\_def\_reb = F$) eases the task of scor-

[1] See for example `http://www.sloansportsconference.com/`.

[2] Data source: `http://basketballvalue.com/downloads.php`.

[3] *Pace* captures the speed of the game and is indicative of the team's playing style.

[4] *Offensive rating* is computed as the average number of points per shot.

[5] *Rebound rate* estimates how effective a team is at gaining possession of the ball after a missed shot, either by an opponent or by one of its own players.

Table 1. Five subgroups discovered by Diverse Subgroup Set Discovery [14]; cover-based approach with default quality-diversity trade-off. For each discovered subgroup its description, size and quality are given.

Description	Size	Quality
$opp\_def\_reb = F \wedge opponent \neq ATL \wedge thabeet = F$	219	0.0692
$opp\_def\_reb = F \wedge opponent \neq ATL$	222	0.0689
$opp\_def\_reb = F \wedge opponent \neq ATL \wedge ajohnson = F$	222	0.0689
$opp\_def\_reb = F \wedge thabeet = F \wedge opponent \neq PHI$	225	0.0685
$opp\_def\_reb = F \wedge opponent \neq PHI$	228	0.0682

ing, whereas *absence* of *reserve* players (*thabeet* and *ajohnson*) is not useful for decision making either.

Aims and Contributions We argue that it is essential to *actively involve the user in the discovery process* to ensure diverse and interesting results. Even when diversity can be obtained through a fully automated discovery process, on itself this is not sufficient to guarantee interesting results. The main reason is that the user's background knowledge and goals are completely ignored. Some existing algorithms that try to leverage expert knowledge require specifying it in advance, but this is a hard task and may therefore be barely less time-consuming than post-processing humongous result sets.

We propose an interactive subgroup discovery algorithm that allows a user to provide feedback with respect to provisional results and steer the search away from regions that she finds uninteresting, towards more promising ones. The intuition behind our approach is that the 'best' subgroups often correspond to common knowledge, which is usually uninteresting. Users expect to obtain novel, unexpected insights, and therefore our system is designed to eliminate such uninteresting subgroups already during search.

The Interactive Diverse Subgroup Discovery (IDSD) framework that we propose builds upon DSSD by re-using the diverse beam search. However, we augment it by making the beam selection interactive: on each level of the search, users are allowed to influence the beam by *liking* and *disliking* subgroups. One of two subgroup similarity measures is then used to generalise this feedback to all subgroups for a specific level, by re-weighing qualities. The adjusted quality measure affects the (diverse) beam selection and hence the search can be guided.

Since it is hard to evaluate interactive data mining methods, we perform two types of evaluations. First, we perform an extensive quantitative evaluation in which user feedback is emulated. For this we treat a set of high-quality subgroups as 'background knowledge' in which the user is not interested, based on which we emulate the user feedback. The purpose of these experiments is to show that undesired results can be effectively avoided, which in return leaves space for novel, potentially more interesting results.

Second, we turn back to the case study that we introduced in this section. We asked a domain expert to use our interactive discovery system, and he successfully found more interesting patterns than with the standard diverse approach. This confirms that human-computer interaction makes it possible to discover interesting subgroups with much less effort than using standard algorithms.

2 Related Work

Subgroup discovery can be seen as an instance of *supervised descriptive rule discovery* [13], like contrast set mining [3] and emerging pattern mining [5]. Apart from DSSD, which was inspired by pattern set mining, several local approaches to redundancy elimination have been proposed: closed sets for labeled data [9] applies only to binary targets, a recent approach uses quadratic programming to do feature selection prior to the discovery process [15].

The importance of taking user knowledge and goals into account was first emphasised by Tuzhilin [17]. More recently De Bie argued that traditional objective quality measures are of limited practical use and proposed a general framework that models background knowledge as a *Maximum Entropy distribution* [4].

Applications of subgroup discovery in various domains often involve iterative refinement of results based on feedback of experts, e.g. in medicine [7,8]. A classification of background knowledge relevant to subgroup discovery was developed [1], and some of the insights were used in the VIKAMINE tool, which enables knowledge transfer between otherwise independent search sessions [2]. SVMs were applied to learn subgroup rankings from user feedback [16], but the feedback phase was not integrated into search.

Outside subgroup discovery, ideas regarding interactive search have been explored in Redescription Mining [6], but we go much further with the influence of users on beam selection. Finally, MIME is an interactive tool that allows a user to explore itemsets using traditional interestingness measures [10].

3 Preliminaries

We consider datasets that are bags of tuples. Let $A = \{A_1, \ldots, A_{l-1}, A_l\}$ denote a set of attributes, where each attribute A_j has a domain of possible values $\mathrm{Dom}(A_j)$. Then a dataset $\mathcal{D} = \{x_1, \ldots, x_n\} \subseteq \mathrm{Dom}(A_1) \times \ldots \times \mathrm{Dom}(A_l)$ is a bag of tuples over A. The attribute A_l is a *binary target attribute*, i.e. the property of interest. Attributes $D = \{A_1, \ldots, A_{l-1}\}$ are *description attributes*.

The central concept is the *subgroup*, which consists of a *description* and a corresponding *cover*. In this paper, a *subgroup description* d is a conjunction of boolean expressions over D, e.g. $D_1 = a \wedge D_2 > 0$. A *subgroup cover* G is a bag of tuples that satisfy the predicate defined by d: $G_d = \{\forall t \in \mathcal{D} : t \in G \Leftrightarrow d(t) = true\}$, the size of the cover $|G|$ is also called *subgroup coverage*.

Subgroup quality measures generally balance the degree of deviation and the size of a subgroup. We use *Weighted Relative accuracy*, given by $\varphi_{WRAcc}(G) = \frac{|G|}{|\mathcal{D}|} \times (1^G - 1^{\mathcal{D}})$, where 1^G (resp. $1^{\mathcal{D}}$) is the proportion of positive examples in G (resp. $\mathcal{D}$). The previous allows us to define *top-k subgroup discovery*:

Problem 1 (Top-k Subgroup Discovery). Given a dataset $\mathcal{D}$, a quality measure φ, and integer k, find the k top-ranking subgroups with respect to φ.

Bottom-up search is usually applied to solve this problem. The search space consists of all possible descriptions and is traversed from short to long descriptions. Common parameters to restrict the search space are a *minimum coverage* threshold (*mincov*), and a *maximum depth* (*maxdepth*). Either exhaustive search or *beam search* can be used, where the latter has the advantage that it is also feasible on larger problem instances. It explores the search space in a levelwise manner, and at each level only the w highest ranking candidates with respect to φ (the *beam*) are selected for further refinement, where *beam width* w is a user-supplied parameter. This makes it ideal for our current purposes.

Diverse Subgroup Set Discovery We recently introduced the DSSD framework [14], which uses heuristic pattern set selection to select a more diverse beam on each level of beam search. The purpose of this approach is to achieve globally less redundant and therefore more interesting results.

The diverse beam selection strategies add a candidate subgroup to the beam only if it is sufficiently different from already selected subgroups. In this paper we use *description-based beam selection* because preliminary experiments showed that it works well for our purposes; our prototype discovery system primarily presents subgroup descriptions to the user. It first sorts all candidates descending by quality and initialises $beam = \emptyset$, then iteratively considers each subgroup in order until $|beam| = w$, and selects it only if there is no subgroup in the (partial) beam that has equal quality and the same conditions *except for one*.

We use *cover redundancy* (CR) to quantify redundancy of a subgroup set, i.e. $CR(\mathcal{G}) = \frac{1}{|\mathcal{D}|} \sum_{t \in \mathcal{D}} \frac{|c(t, \mathcal{G}) - \hat{c}|}{\hat{c}}$, where $\mathcal{G}$ is a set of subgroups, $c(t, \mathcal{G})$ is the *cover count* of a transaction, i.e. the number of subgroups in $\mathcal{G}$ that cover t, and $\hat{c}$ is the average cover count over all $t \in \mathcal{D}$. Essentially, it measures the deviation of the cover counts from the uniform distribution. Although absolute values are not very meaningful, CR is useful when comparing subgroup sets of similar size for the same dataset: a lower CR indicates that fewer tuples are covered by more subgroups than expected, therefore the subgroup set must be more diverse.

4 Interactive Diverse Subgroup Discovery

We now present the Interactive Diverse Subgroup Discovery (IDSD) algorithm, which employs user feedback to guide a beam search. Main design goals are to develop 1) a simple interaction mechanism that 2) requires little user effort. We rely on two observations to achieve these goals. First, it is easier for a user to assess patterns rather than individual transactions or attributes. Second, it is possible to generalise user feedback using similarities between subgroups.

To involve the user already during the discovery process, the central idea is to *alternate between mining and user interaction*: the algorithm mines a set of patterns, a user is given the opportunity to provide feedback, the feedback is used to steer the search, and the algorithm mines a new set of patterns.

Algorithm 1 Interactive Diverse Subgroup Discovery (IDSD)

Input: Dataset $\mathcal{D}$; beam selection S; subgroup similarity σ; $mincov$, w, $maxdepth$
Output: Set of k subgroups R

1: $beam \leftarrow \{\emptyset\}$, $I \leftarrow \emptyset$, $R \leftarrow \emptyset$, $depth \leftarrow 1$
2: **repeat** ▷ Generate all candidates for this level
3: $\quad cands = \{c \in Extensions(beam) \mid Coverage(c, \mathcal{D}) \geq mincov \wedge$
$\qquad \neg\exists n \in I_{\text{neg}} : IsExtension(c, n)\}$
4: $\quad beam \leftarrow \emptyset$
5: $\quad$ **repeat** ▷ Selection and interaction loop
6: $\qquad beam \leftarrow SelectBeam(S, cands, \varphi', w, beam)$
7: $\qquad I \leftarrow I \cup GetFeedback(beam)$
8: $\qquad R \leftarrow R \cup I_{\text{pos}}$
9: $\qquad beam \leftarrow beam \setminus I_{\text{neg}}$, $cands \leftarrow cands \setminus I_{neg}$
10: $\quad$ **until** $|beam| = w$ ▷ No patterns were disliked
11: $\quad$ **for all** $c \in cands$ **do**
12: $\qquad UpdateTopK(R, k \times 100, c, \varphi'(c, I, \sigma))$
13: $\quad depth \leftarrow depth + 1$
14: **until** $depth > maxdepth$
15: **return** $SelectBeam(S, R, \varphi', k, \emptyset) \cup I_{\text{pos}}$ ▷ Selection from large overall top-k

As a levelwise search procedure that takes only a limited number of intermediate solutions to the next level, beam search provides an excellent framework to implement this high-level procedure. That is, on each level we let the user influence the beam by *liking* and *disliking* subgroups. Patterns that are disliked are immediately removed from the beam and replaced by others, effectively guiding search away from those apparently uninteresting branches of the search space.

This approach has the advantage that it is relatively easy to evaluate subgroups with short descriptions at early levels, while this has a strong influence on search. Providing feedback at later levels allows fine-tuning, and search parameters such as $maxdepth$ and w allow a user to manage her efforts.

Algorithm Details Algorithm 1 presents the method that we propose. In the following we focus on how the DSSD diverse beam search, as briefly explained in the Preliminaries, is modified to incorporate user feedback. The main difference with respect to DSSD is that user feedback is used to re-weigh the qualities of all possible patterns, effectively re-ranking patterns according to the user's interest.

Feedback elicitation – Feedback elicitation is performed on line 7, after a beam has been selected (line 6, see also below). All w selected subgroups are presented to the user in a GUI, and she can provide feedback before continuing.

As feedback, the user can mark each subgroup in a beam as *interesting* ('like') or *uninteresting* ('dislike'). Let I_{pos} (resp. I_{neg}) denote the set of all positively (resp. negatively) evaluated subgroups. Additionally, let $I = I_{pos} \cup I_{neg}$ be the set of all evaluated subgroups. Note that a user is not obliged to provide any feedback, hence I might not include all subgroups that are in the current beam and it might even be empty. In the latter case the resulting search is equal to that of (non-interactive) DSSD.

If any subgroups are disliked in this phase, they are removed from the beam (line 9) and lines 5-9 are repeated until a complete beam consisting of w subgroups is obtained.

Candidate generation – On each level, initially all *direct extensions* of all subgroups in the current beam are generated as candidates (line 3). Here, a direct extension is a subgroup description augmented with one additional condition. Subgroups with too small coverage and all direct extensions of negatively evaluated subgroups in I_{neg} are removed. Note that this does not necessarily result in the complete pruning of the corresponding branch in the search tree. Consider the following example: $A \wedge B \wedge C$ may be generated as extension of $B \wedge C$, even if A was disliked and thus added to I_{neg} at *depth*-1. This preserves the capability to discover high-quality subgroups via other branches.

Feedback-driven subgroup selection – Since feedback only concerns individual subgroups, we need to generalise it to the complete candidate set. We achieve this through modification of the qualities of all subgroups in *cands*: starting from the 'prior' given by φ, the qualities are updated according to how similar they are to the evaluated subgroups. This way, we obtain a quality measure φ' that takes user feedback into account and effectively re-ranks all possible subgroups. Subsequently, we use the regular diverse beam selection strategy on line 6, with the only difference that the modified qualities are used.

For this to work we need a notion of subgroup similarity: subgroups that are similar to interesting subgroups get a higher quality, whereas subgroups that are similar to uninteresting subgroups get a lower quality. Let $c \in cands$ and $i \in I$, and let d_x resp. G_x denote the description resp. cover of a subgroup x. We use the following two subgroup similarity measures:

$$\sigma_{description}(c, i) = \frac{|d_c \cap d_i|}{|d_c \cup d_i| - 1}, \; \sigma_{cover}(c, i) = \frac{|G_c \cap G_i|}{|G_c|} \tag{1}$$

Description similarity is almost equal to Jaccard similarity; -1 is added to the denominator so that a subgroup and any of its direct extensions have maximal similarity of 1. Cover similarity is based on the overlap coefficient and has the same property for direct extensions.

Finally, given a subgroup similarity measure σ, the modified subgroup quality measure φ' that takes user feedback into account is defined as:

$$\varphi'(G) = \frac{1 + \sum_{i \in I_{pos}} \sigma(G, i)}{1 + \sum_{i \in I_{neg}} \sigma(G, i)} \times \varphi(G) \tag{2}$$

It re-weighs the 'base quality' φ with a factor based on similarity to evaluated subgroups in I. Note that φ' is equivalent to φ when $I = \emptyset$. Also, values of φ' change immediately after each round of feedback elicitation, hence feedback has an immediate effect on (incremental) beam selection.

Overall results – During search a large overall 'top-k' is maintained using the re-weighed quality measure φ' (line 12). At the end of the algorithm (line 15), a set of k subgroups is selected from this overall large top-k using the subgroup selection procedure just described. Note that all positively evaluated subgroups are added to this final result set R regardless of their qualities.

5 Experiments

5.1 Quantitative Evaluation

In order to be able to perform a large series of experiments, we emulate user feedback. We select a set of high-quality subgroups BK that serves as *background knowledge*; subgroups in BK are already known and should therefore be avoided as much as possible. The intuition behind this approach echoes the example in Section 1: top subgroups usually correspond to common knowledge and are therefore uninteresting.

BK is selected from the output of a standard subgroup discovery algorithm. Selection depends on the subgroup similarity measure used: when using description similarity, we iteratively select the highest quality subgroup having a description with ≤ 3 conditions that does not overlap with the description of any previously selected subgroup; for cover similarity, we only select the highest quality subgroup. During search, BK is used to emulate evaluations: any subgroup s in the beam for which $\exists b \in BK : \sigma(s,b) > \beta$ is automatically 'disliked' by adding it to I_{neg}. Parameter β allows varying the amount of evaluated subgroups: larger values result in fewer negative judgements. Note that no positive feedback is emulated.

To evaluate the effectiveness of the algorithm in eliminating undesired conditions or tuples from the results, we compute the overlap of the discovered subgroups with elements of BK. Depending on the subgroup similarity measure, this is either $overlap_{desc}(s, BK) = \max_{b \in BK} |d_s \cap d_b|$ or $overlap_{cov}(s, BK = \{b\}) = |G_s \cap G_b|$. We report the average overlap for all subgroups included in result set R, i.e. $overlap(R, BK) = \frac{1}{|R|} \times \sum_{s \in R} overlap(s, BK)$.

Dataset properties are listed in Table 2, together with size, average description length, and coverage of the generated background knowledge. Except for *nba*, which was introduced in Section 1, all were taken from the UCI repository[6]. The datasets were preprocessed as follows: transactions with missing values were removed, and all numeric attributes were discretised into 6 bins using equal-width binning.

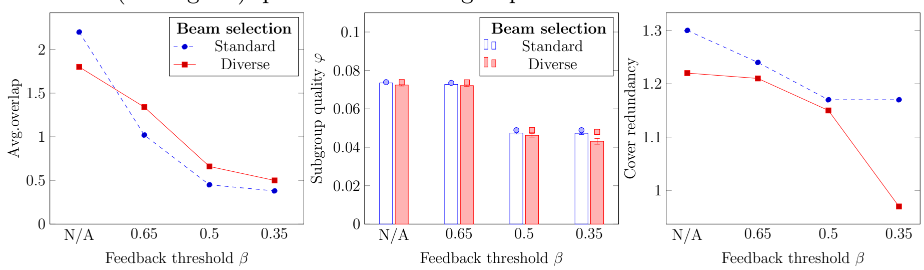
Table 2. Datasets and background knowledge

	$\|\mathcal{D}\|$	$\|A\|$	*Desc.BK* $\|BK\|$	*Desc.BK* $\overline{\|d_{BK}\|}$	*Cov.BK* $\|G_{BK}\|$
breast-w [bw]	683	11	2	2.0	137
credit-a [ca]	653	17	3	3.0	325
credit-g [cg]	1000	22	2	3.0	394
diabetes [db]	768	10	2	3.0	82
liver [lv]	345	8	2	3.0	107
nba	923	26	6	2.8	228

Search parameters were set to the following values in all experiments: $mincov = 0.1 \times |\mathcal{D}|$, $maxdepth = 5$, $w = 20$, $k = 100$. Note that small values of w and k are chosen in order to match the limited processing capabilities of a human user. We first focus on a single dataset, *credit-g*, and then discuss the results of experiments with multiple datasets.

[6] `http://archive.ics.uci.edu/ml/datasets.html`

Fig. 1. IDSD results for *credit-g*, with description subgroup similarity and emulated user feedback. A lower feedback threshold β results in more negative feedback; the *N/A*-setting corresponds to the non-interactive algorithm (i.e. no feedback emulated). The subgroup quality plot depicts averages, standard deviations and maxima of the individual (unweighed) qualities in the subgroup sets found.



A Characteristic Experiment in Detail Figure 1 shows the results that we obtained on *credit-g* for various values of β, which controls the amount of emulated negative feedback. Also, we experimented with both standard and (description-based) diverse beam search. The left plot shows that the average overlap with BK decreases considerably when β decreases and thus more subgroups are evaluated negatively. This demonstrates that re-weighing subgroup quality using description-based similarity is effective at eliminating undesired conditions from the beam and final results. Only modest numbers of negative evaluations (8 to 12) were required to achieve this.

The middle plot shows that both maximum and average qualities of the subgroup sets decrease. This is as expected though: the user emulation scheme was designed to prune high-quality subgroups. Nevertheless, the algorithm succeeds in discovering (other) subgroups of quite high quality. Finally, the right plot shows that redundancy slightly decreases as I grows. Although the differences between the standard and diverse beam search appear to be small, the diverse results are clearly less redundant: in particular for lower β, cover redundancy is lower and standard deviation of the subgroup qualities is higher.

Overall Results For the experiments in Table 3 we use diverse beam selection, with either description ($\beta = 0.35$) or cover ($\beta = 0.5$) subgroup similarity.

In general, both approaches adequately eliminate undesired subgroups from the result set, whether it is through negatively marked conditions or tuples. This is demonstrated by the consistently decreasing overlap with the background knowledge. Importantly, the number of evaluations required to achieve this is generally modest, i.e. $|I| \leq 25$ for all cases except two. This makes the approach practically useful and usable for a domain expert. As expected, average and maximum subgroup qualities decreases. Only the effect on cover redundancy varies depending on the dataset, but the difference is often small. We conclude that interaction and quality re-weighing work well together with the diverse beam selection.

Table 3. Overall results using feedback emulation and description or cover subgroup similarity measure, with and without interaction (after/before the $\rightarrow$). $|I|$ is the number of emulated dislikes. For each obtained subgroup set are shown: overlap with background knowledge, average and maximum quality, and cover redundancy.

σ	$\mathcal{D}$	$\lvert I\rvert$	Standard diverse $\rightarrow$ Interactive			
			Avg.overlap(BK)	φ_{avg}	φ_{max}	CR
Desc	bw	8	1.37 $\rightarrow$ 0.80	.187 $\rightarrow$ .171	.200 $\rightarrow$ .184	1.20 $\rightarrow$ 1.26
	ca	24	1.65 $\rightarrow$ 0.89	.178 $\rightarrow$ .088	.181 $\rightarrow$ .131	1.00 $\rightarrow$ 0.90
	cg	8	1.80 $\rightarrow$ 0.66	.072 $\rightarrow$ .046	.074 $\rightarrow$ .049	1.22 $\rightarrow$ 1.15
	db	12	1.99 $\rightarrow$ 0.25	.077 $\rightarrow$ .048	.084 $\rightarrow$ .056	0.66 $\rightarrow$ 1.17
	lv	12	1.20 $\rightarrow$ 0.38	.042 $\rightarrow$ .041	.047 $\rightarrow$ .045	0.83 $\rightarrow$ 0.97
	nba	78	1.57 $\rightarrow$ 1.09	.067 $\rightarrow$ .058	.071 $\rightarrow$ .064	1.50 $\rightarrow$ 1.57
Cover	bw	17	125.37 $\rightarrow$ 120.09	.187 $\rightarrow$ .176	.200 $\rightarrow$ .200	1.20 $\rightarrow$ 1.12
	ca	157	318.95 $\rightarrow$ 232.68	.178 $\rightarrow$ .013	.181 $\rightarrow$ .025	1.00 $\rightarrow$ 0.18
	cg	7	353.49 $\rightarrow$ 168.50	.072 $\rightarrow$ .046	.074 $\rightarrow$ .049	1.22 $\rightarrow$ 1.15
	db	1	61.22 $\rightarrow$ 62.13	.077 $\rightarrow$ .077	.084 $\rightarrow$ .084	0.66 $\rightarrow$ 0.64
	lv	2	57.57 $\rightarrow$ 27.66	.042 $\rightarrow$ .041	.047 $\rightarrow$ .047	0.83 $\rightarrow$ 1.26
	nba	1	218.28 $\rightarrow$ 117.88	.067 $\rightarrow$ .036	.071 $\rightarrow$ .058	1.50 $\rightarrow$ 0.75

5.2 Case Study: Sports Analytics

As we have seen in Section 1, the subgroups discovered by DSSD were unsatisfactory. To demonstrate that the proposed interactive approach can be used to improve on this, we asked a basketball journalist to use IDSD and evaluate the results. In the following we set the search parameters to $mincov = 50$, $w = 10$, $maxdepth = 3$, and $k = 5$, and we use description similarity.

The domain expert evaluated 18 subgroups during an interactive search session, 13 of length-1 and 5 of length-2. Examples of *liked* subgroups include $crawford = F$, $pace < 88.977$, and $matthews = T \wedge hickson = T$ (7 subgroups in total). Examples of *disliked* subgroups are $opp\_def\_reb = F$, $thabeet = F$, and $pace < 88.977 \wedge opponent \neq MIA$ (11 subgroups in total). The discovered subgroups are presented in Table 4. Although the objective qualities are lower than the maximum, the results were considered more interesting as they provided novel insights about relevant players.

Table 4. Top five subgroups discovered by IDSD with description-based similarity. For each discovered subgroup its description, size and quality are given.

Description	Size	Quality
$crawford = F \wedge matthews = T$	96	0.0382
$hickson = T$	186	0.0219
$crawford = F \wedge hickson = T$	328	0.0211
$matthews = T \wedge hickson = T$	290	0.0163
$matthews = T \wedge pace < 88.518$	303	0.0221

A user needs to consider subgroups one by one when processing results and providing feedback. Hence, we can estimate user effort E by counting the subgroups she had to consider. The effort induced by non-interactive diverse subgroup discovery is then equal to the lowest rank of an interesting subgroup in the result set sorted by quality. The effort induced by the interactive approach also includes the number of subgroups presented during the search: $E_{IDSD} = maxdepth \times w + |I_{neg}|$. In this case we have $E_{DSSD} = 1049$ and $E_{IDSD} = 5 + (3 \times 10 + 11) = 46$, which confirms that within-search interaction substantially reduces the effort required to discover interesting results.

Discussion Although this is a good example of a successful interactive session, in some other sessions the domain expert deemed the results unsatisfactory. In some cases the search space was pruned too eagerly, or positive and negative evaluations were not properly balanced. Also, the expert did not find the approach using cover similarity useful, as this resulted in descriptions that were were not interpretable. Training of the domain expert might solve this and results obviously also depend on the data, but this also shows that it is worth investigating more elaborate similarity measures to generalize user feedback.

Another crucial drawback is an unintuitive effect on beam selection, e.g. *disliking* a large subgroup based on its description (e.g. $reserve\_player = F$) steers the search away from promising regions. Another concern is the capacity to discover novel subgroups (as opposed to simply replicating the feedback). Multiple sessions might be required to explore unrelated regions. However, given the significantly lower effort, the cumulative effort is still reduced.

6 Conclusions and Future Work

We argued that it is essential to actively involve the user in the discovery process to obtain results that she finds interesting. To this end, we proposed the Interactive Diverse Subgroup Discovery (IDSD) algorithm that allows a user to provide feedback to provisional results already *during search*. It augments a diverse beam search by letting the user 'like' and 'dislike' subgroups in the beam. Although this interaction mechanism is conceptually simple and easy to use, it allows a user to guide the search effectively.

In the quantitative evaluation, we emulated the feedback of a user that wants to avoid the re-discovery of common knowledge. Experiments show that undesired results can be eliminated, whereas other, potentially more interesting subgroups are found. Furthermore, we conducted a case study in which a domain expert was able to find more interesting patterns when compared to the results of standard algorithms. This confirms that within-search human-computer interaction can contribute to a substantial reduction in the effort needed to discover interesting subgroups.

Future Work This paper presents only a first step towards user-driven pattern discovery, but since the user is too often still neglected we believe it is an important step. In the future, one obvious line of research is to investigate what features other than inclusion/exclusion of individual conditions and tuples are relevant to the user, and are therefore useful to infer subjective interestingness.

A second direction that will be essential to research is pattern visualisation. In our prototype, we mainly focused on presenting subgroup descriptions, but in the future it will be important to visualise the different aspects of the patterns. Not only descriptions and covers should be visualised, but also other relevant features such as traditional interestingness and surprisingness measures. We deem this particularly important for larger datasets and/or beam widths. Only then will it be possible for the user to interactively explore the data in an intuitive way.

Acknowledgements. Matthijs van Leeuwen is supported by a Rubicon grant of the Netherlands Organisation for Scientific Research (NWO).

References

1. M. Atzmüller. Exploiting background knowledge for knowledge-intensive subgroup discovery. In *Proceedings of IJCAI'05*, pages 647–652, 2005.
2. M. Atzmüller and F. Puppe. Semi-automatic visual subgroup mining using vikamine. *Journal of Universal Computer Science*, 11(11):1752–1765, 2005.
3. J. Bailey and G. Dong. Contrast data mining: Methods and applications. Tutorial at ICDM'07, 2007.
4. T. De Bie. An information theoretic framework for data mining. In *Proceedings of KDD'11*, pages 564–572, 2011.
5. G. Dong, X. Zhang, L. Wong, and J. Li. CAEP: Classification by aggregating emerging patterns. In *Proceedings of DS'99*, pages 30–42, 1999.
6. E. Galbrun and P. Miettinen. A Case of Visual and Interactive Data Analysis: Geospatial Redescription Mining. In *Instant Interactive Data Mining Workshop at ECML-PKDD'12*, 2012.
7. D. Gamberger and N. Lavrac. Expert-guided subgroup discovery: Methodology and application. *Journal of Artificial Intelligence Research*, 17:501–527, 2002.
8. D. Gamberger, N. Lavrac, and G. Krstacic. Active subgroup mining: a case study in coronary heart disease risk group detection. *Artificial Intelligence in Medicine*, 28(1):27–57, May 2003.
9. G.C. Garriga, P. Kralj, and N. Lavrac. Closed sets for labeled data. *Journal of Machine Learning Research*, 9:559–580, 2008.
10. B. Goethals, S. Moens, and J. Vreeken. MIME: a framework for interactive visual pattern mining. In *Proceedings of KDD'11*, pages 757–760, 2011.
11. F. Herrera, C.J. Carmona, P. González, and M.J. Jesus. An overview on subgroup discovery: foundations and applications. *Knowledge and Information Systems*, 29(3):495–525, 2011.
12. W. Klösgen. *Advances in Knowledge Discovery and Data Mining*, chapter Explora: A Multipattern and Multistrategy Discovery Assistant, pages 249–271. 1996.
13. P. Kralj Novak, N. Lavrač, and G.I. Webb. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research*, 10:377–403, 2009.
14. M. van Leeuwen and A. Knobbe. Diverse subgroup set discovery. *Data Mining and Knowledge Discovery*, 25:208–242, 2012.
15. R. Li and S. Kramer. Efficient redundancy reduced subgroup discovery via quadratic programming. In *Proceedings of DS'12*, pages 125–138, 2012.
16. S. Rüping. Ranking interesting subgroups. In *Proceedings of ICML'09*, pages 913–920, 2009.
17. A. Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *Proceedings of KDD'95*, pages 275–281, 1995.
18. S. Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proceedings of PKDD'97*, pages 78–87. Springer, Heidelberg, 1997.