Subgroup Discovery using Bump Hunting on Multi-Relational Histograms

Radomír Černoch and Filip Železný

Czech Technical University, Faculty of Electrical Engineering,
Department of Cybernetics, Intelligent data analysis research lab

Abstract. We propose an approach to subgroup discovery in relational databases containing numerical attributes. The approach is based on detecting bumps in histograms constructed from substitution sets resulting from matching a first-order query against the input relational database. The approach is evaluated on seven data sets, discovering interpretable subgroups. The subgroups' rate of survival from the training split to the testing split varies among the experimental data sets, but at least on three of them it is very high.

1 Introduction

Subgroup discovery (SD) [8] is a data mining technique, which has gained significant attention in recent years. The notion of SD differs slightly among researchers, but in general the task can be defined as discovering a subset of a dataset, which exhibits "interesting" statistical properties. As a motivating example, consider a bank with a database of its clients. Various properties (regularity of income, average balance on the account, number of loans, ...) divides the clients into potentially overlapping groups to be discovered by SD.

Originally the goal of *subgroup discovery* [8] was defined for the MIDOS system [20] as "given a population of individuals and a property of individuals we are interested in, find population subgroups that are statistically 'most interesting', e.g., are as large as possible and have the most unusual statistical (distributional) characteristics with respect to the property of interest." This concept became predominant in the majority of SD systems, both in the propositional [14,1] and relational [20,19] setting.

Here we explore a slightly different notion of subgroup discovery. Consider first a single data table in the banking example in which rows describe clients and one of the columns (attributes) contains the client's age, which is the property of interest. Since age is a numerical attribute, the most natural way to discover subgroups with respect to this property is to plot the histogram of age from the data table and look for possible bumps in the histogram. Analogically, bumps in multi-dimensional histograms may be considered when several numerical attributes are jointly of interest.

Here we adhere to the single-dimensional case but instead generalize the bump-hunting framework to the relational setting. For example, we would like to be able to discover in a relational banking database that clients from Prague fall apart into distinct subgroups according to the size of deposits they make. This can be accomplished by inspecting the histogram of the *Amount* variable's values in the set of substitutions making the query

$$c = \mathsf{client}(C) \wedge \mathsf{residence}(C, \mathtt{prague}) \wedge \mathsf{account}(C, A) \wedge \mathsf{deposit}(A, Amount) \quad (1)$$

true in the database. However, to preserve the interpretability of the histogram, there should be one element in the substitution set per each client. That is to say, values of *Amount* pertaining to a single client must be aggregated (e.g., averaged) in the substitution set before constructing the histogram.

Motivated by the above example, we developed an algorithm that, given a relational database with a distinguished main relation (such as client above), searches for a triple consisting of a query, a variable within the query, and a suitable aggregation function, so that the histogram of numerical values constructed from these three ingredients, in the way exemplified above, exhibits remarkable bumps. To this end, we address both the logical and statistical aspect of the algorithm: in particular, we design a refinement-based search for the target query as well as a fast, "visually inspired" histogram-inspection technique. The simplicity and speed of the latter is vital due to the generally daunting embedding of one data mining task (bump hunting) in another (query search).

The paper is organized as follows. Section 2 provides an overview of related work. Section 3 gives a necessary theoretic background and terminology and Section 4 describes the new bump hunting technique, which is employed in a multi-relational settings in Section 5. Section 6 presents results of the algorithm on a real-world dataset and Section 7 concludes the paper.

2 Related work

Bump hunting is discussed as a task of finding multiple modes in statistical distributions (while eliminating outliers). Concerning bump-hunting, we can distinguish two tasks: *bump detection* and *parameter estimation*. Despite some applications need only to count the number of modes in the dataset, the research focuses on evaluating parameters of the individual bumps (e.g. means and variances in a mixture of gaussians). This prevalent trend, which derived from the early work on finding optimal scale of kernels [17], can be traced in more recent bump-hunting algorithms [3,4,21]. To the best of our knowledge, all use extensive search in the parameter space (e.g. expectation-maximization, random restarts in genetic algorithms).

The extensive search in the parameter space works well in a single-relational dataset, but the multi-relational environment adds another level of complexity in the search space of first-order patterns. Nevertheless, as argued in [15], "often a visual examination of a dotplot or histogram is sufficient to judge whether the results [...] can be regarded as unimodal." Hence in our work we focus on detecting modes rather than estimating their parameters.

Subgroup Discovery. Secondly, there is a large area of research on SD. Apart from the pioneering systems EXPLORA [7] and MIDOS [20], current research focuses on two tasks: One group of algorithms employ or adapt existing machine-learning algorithms for SD (originally CN2-SD [14], recent ones include e.g. [6,1]). RSD [19] is the most relevant to our work, as it builds upon the idea of *propositionalization* of a multi-relational dataset (used also in machine learning algorithms such as nFOIL and kFOIL [13,12,11]). RSD builds a set of first-order features and constructs a single-relational boolean table, in which rows correspond to examples and columns correspond to the first-order features. Then the table is passed to a fast attribute-value learner, resulting in a good performance, inherent to propositionalization techniques in general. The drawbacks are a limited interpretability of the subgroups, as the attribute-value learner acts as a black-box inside RSD and a lack of support for numeric attributes.

The other group of SD research focuses on numeric attributes. A direct treatment of numeric attributes, exemplified by decision trees algorithms, is not prevalent in SD systems (see [5] for an overview). Instead, large effort is put in optimizing the *binarization* technique, which splits the real axis into a finite number of intervals and each is assigned a separate symbol. After the transformation, the dataset only contains nominal attributes, which are better supported by mainstream systems. The MergeSD system [5] seems to be the largest systematic treatment of binarization, achieving a high performance gain by dynamic merging of the potentially overlapping intervals.

Still, we argue that the binarization approach can be overcome by creating SD algorithms that process numeric values directly by examining their statistical properties.

Substitution sets in machine-learning were explored most notably in the Relaggs [9] system. Relaggs uses a predefined set of operators to aggregate certain columns in the clause-instance matching table. Such operators include *mean*, *minimum*, *count*, etc. By doing this, the system resembles RSD: Both create a single-relational table, which is passed to a propositional learner. Unlike RSD, Ralaggs uses numeric values in the table, by generating it using the aggregation operators. Nevertheless, since many propositional learners can handle high-dimensional data, Relaggs does not need nor provide an efficient way of evaluating and pruning the features. We build our algorithm on the idea of *Relaggs* and use the substitution sets, but enhance it for SD.

3 Background information

First order features. A conjunctive clause c is a set of atoms, whose arguments can be variables or constants. An example e is a set of ground atoms, whose arguments are only constants. We say that a clause c *θ-subsumes* an example e [16] iff $c\,\theta \subseteq e \cup \mathcal{B}$, where $\mathcal{B}$ is some background knowledge. This relation is denoted as $c \models_\theta e$. Given an example e, a clause c and a variable x appearing in c (the *query variable*): The multiset of all *instantiations* $\mathbf{V}$ is the multiset of values, which x can take if the pattern c is applied to e:

$$\mathbf{V}_e = \{\!\{ x\,\theta \mid c \models_\theta e \}\!\} \tag{2}$$

The *indicator function* (number of occurrences of the value v in the multiset $\mathbf{V}$) will be denoted as $|v \in \mathbf{V}|$. For all values not included in $\mathbf{V}$ the indicator function is zero: $\forall v \notin V.\ |v \in \mathbf{V}| = 0$.

Example 1. We further formalize the banking example from Section 1. Let there be a single example $e_1 = \{$ client(john), residence(john, prague) account(john, 1), account(john, 2), deposit(1, 1000), deposit(1, 1400), deposit(1, 1400), deposit(2, 10 000)$\}$. The set of instantiations for the *Amount* variable in query (1) is

$$\mathbf{V}_{e_1} = \{\!\{ 1000, 1400, 1400, 10\,000 \}\!\} \tag{3}$$

The indicator function's values are as follows: $|1400 \in \mathbf{V}_{e_1}| = 2$, $|1000 \in \mathbf{V}_{e_1}| = |10\,000 \in \mathbf{V}_{e_1}| = 1$ and $|v \in \mathbf{V}_{e_1}| = 0$ for all other v.

Aggregation functions. The *domain of a variable* x, denoted by $\mathcal{D}_x$, is the set of all values, which can be assigned to x. Most importantly the domain of *numeric variables* is the set of real numbers $I\!R$. An *aggregation function* is a function which maps a multiset of values to a real number: $f_{agg}(x) : \{\!\{ \mathcal{D}_x \}\!\} \to I\!R$.

Histogram. Given the set of *bin centres* $W = \{w_1, ..., w_n\}$ and a multiset of values $\mathbf{V}$, a *histogram* is defined as a multiset

$$\mathbf{W}_\mathbf{V} = \{\!\{ w \;\mid\; v \in \mathbf{V} \wedge w = \arg\min_{w' \in W} \|w' - v\| \}\!\} \ , \tag{4}$$

which assigns one bin w to each value v minimizing their mutual distance.[1] The histogram also defines a probability mass function over the set of bins

$$\mathrm{P}(\mathbf{W}_\mathbf{V} = w) = \frac{|w \in \mathbf{W}_\mathbf{V}|}{\sum_{w' \in \mathbf{W}_\mathbf{V}} |w' \in \mathbf{W}_\mathbf{V}|} \ . \tag{5}$$

Example 2. Applying the mean $\varnothing$ function on e_1 gives $\varnothing(\mathbf{V}_{e_1}) = 3450$. Suppose that there are 4 more examples $e_2 \dots e_5$ s.t. $\varnothing(\mathbf{V}_{e_2}) = 2000$, $\varnothing(\mathbf{V}_{e_3}) = 6500$,

[1] For clarity we assume there is only one such value. Otherwise the smallest one can be taken.

$\varnothing(\mathbf{V}_{e_4}) = 8000$ and $\varnothing(\mathbf{V}_{e_5}) = 16000$. Furthermore take the multiset of all the means $\mathbf{V}_{\mathcal{E}} = \{\!\!\{\varnothing(\mathbf{V}_{e_1}), \ldots, \varnothing(\mathbf{V}_{e_4})\}\!\!\}$ and the set of bin centres

$$W = \{2500, 5000, 7500, 10000, 12500\} \ . \tag{6}$$

The $\mathbf{W}_{\mathbf{V}_{\mathcal{E}}}$ is the histogram containing values from W, with frequencies being the number of nearest values from $\mathbf{V}_{\mathcal{E}}$:

$$\mathbf{W}_{\mathbf{V}_{\mathcal{E}}} = \{\!\!\{2500, 2500, 7500, 7500, 12500\}\!\!\} \ . \tag{7}$$

The probability mass function $\mathrm{P}(\mathbf{W})$ is then defined as

$$\mathrm{P}(\mathbf{W}_{\mathbf{V}_{\mathcal{E}}} = w) = \begin{cases} {}^{2}\!/_{5} : w = 2500 \\ \ 0 \ : w = 5000 \\ {}^{2}\!/_{5} : w = 7500 \\ \ 0 \ : w = 10000 \\ {}^{1}\!/_{5} : w = 12500 \end{cases} \tag{8}$$

Now we can proceed to formally defining the task of subgroup discovery:

- **Given** a set of examples $\mathcal{E} = \{e_1, \ldots, e_n\}$ (possibly with some background knowledge), each of which is a set of ground atoms.
- **Find** a first-order clause c, a query-variable x and an aggregation function f_{agg} such that the histogram of values $\{\!\!\{f_{agg}(\mathbf{V}_{e_1}), f_{agg}(\mathbf{V}_{e_2}), \ldots, f_{agg}(\mathbf{V}_{e_n})\}\!\!\}$ shows unusual characteristics for a subset of examples $S \subset \mathcal{E}$ (the subgroup).

4 Bump hunting

The essential part of the entire algorithm is the way of recognizing subgroups in histograms. Similarly to bump-hunting algorithms, it must be able to cancel noise without smoothing out the sought bumps. Moreover it must be comprehensible enough to provide a definition of the subgroup. And lastly the produced subgroups must be large enough.

The basic definition of the proposed measure is visually-inspired: Two bins in the histogram can be called modes if they have a high probability and are separated by bins with low probability. Formally we start by defining local minima and maxima of the histogram.

Given bin centres $W = \{w_1, \ldots, w_n\}$ and a histogram $\mathbf{W}$, let $Min \subset W$ be the set of all bin centres with negative second derivative.

$$\begin{aligned} Min = \{w_i \in W \mid i \in \langle 1 \ldots n \rangle \,, \\ |w_i \in \mathbf{W}| < |w_{i-1} \in \mathbf{W}| \ \text{ and } \ |w_i \in \mathbf{W}| < |w_{i+1} \in \mathbf{W}|\} \end{aligned} \tag{9}$$

However we want to consider the smallest (resp. largest) element from W even though there is no w_0 (resp. w_{n+1}). Hence the definition is altered slightly by including an artificial values $w_0 = -\infty$ (resp. $w_{n+1} = +\infty$); notice that $|\pm\infty \in \mathbf{W}| = 0$.

Similarly the set $Max \subseteq W$ is the set of all bin centres with positive second derivative.

$$\begin{aligned} Max = \{ w_i \in W \mid i \in \langle 1 \dots n \rangle \,, \\ |w_i \in \mathbf{W}| > |w_{i-1} \in \mathbf{W}| \text{ and } |w_i \in \mathbf{W}| > |w_{i+1} \in \mathbf{W}| \} \end{aligned} \tag{10}$$

Without loss of generality, we can assume that no two neighbouring bins have the same frequency in $\mathbf{W}$. If they did, both bins could be merged into a single bin, whose width is double of the original ones. Under this assumption and considering that $|w \in \mathbf{W}| \geq 0$, notice that the following condition holds $|Max| = |Min| + 1$. Moreover the items $\{k_0, ..., k_m\} \in Max$ and $\{l_1, ..., l_m\} \in Min$ can be ordered as follows:

$$k_0 < l_1 < k_1 < l_2 < k_2 < \cdots < l_m < k_m \tag{11}$$

This yields the definition of the estimate of *area of low probability* (ALP). Let $k_i, k_{i+1} \in Max$ be two successive peaks. The ALP inbetween is measured as

$$\mathsf{ALP}(i, \mathbf{W}) = \frac{1}{|W|} \cdot \sum_{w \in \mathbf{W}\colon\ k_i < w < k_{i+1}} level - \mathrm{P}(\mathbf{W} = w) \ , \tag{12}$$

where $level = \min(\mathrm{P}(\mathbf{W} = k_i), \mathrm{P}(\mathbf{W} = k_{i+1}))$. If the normalizer constant $|W|$ was removed, the measure would rise with an increaing number of bins. The definitions are illustrated on the running example from Section 3:

Example 3. Recall the multiset $\mathbf{W}_{\mathbf{V}_\varepsilon}$ from (7) and the probability mass function $\mathrm{P}(\mathbf{W}_{\mathbf{V}_\varepsilon} = w)$ from (8). We start by identifying the local maxima and minima of the histogram $\mathbf{W}_{\mathbf{V}_\varepsilon}$:

$$Max = \{2500, 7500, 12500\} \text{ and } Min = \{5000, 10000\} \ ,$$

$$k_0 = 2500,\ l_1 = 5000,\ k_1 = 7500,\ l_2 = 10000 \text{ and } k_2 = 12500 \ .$$

Since there are two local minima l_1 and l_2, there are two values of ALP:

$$\begin{aligned} level_1 &= \min\{\mathrm{P}(\mathbf{W}_{\mathbf{V}_\varepsilon} = 2500), \mathrm{P}(\mathbf{W}_{\mathbf{V}_\varepsilon} = 7500)\} = \min\{^2/_5, {}^2/_5\} = {}^2/_5 \\ \mathsf{ALP}(1, \mathbf{W}) &= \frac{1}{5} \cdot \sum_{w \in \{5000\}} level_1 - \mathrm{P}(\mathbf{W}_{\mathbf{V}_\varepsilon} = w) \\ &= {}^1/_5 \cdot (level_1 - 0) = {}^2/_{25} \end{aligned} \tag{13}$$

$$\begin{aligned} level_2 &= \min\{\mathrm{P}(\mathbf{W}_{\mathbf{V}_\varepsilon} = 7500), \mathrm{P}(\mathbf{W}_{\mathbf{V}_\varepsilon} = 12500)\} = \min\{^2/_5, {}^1/_5\} = {}^1/_5 \\ \mathsf{ALP}(2, \mathbf{W}) &= \cdots = {}^1/_5 \cdot (level_2 - 0) = {}^1/_{25} \end{aligned} \tag{14}$$

The values $^2/_{25}$ and $^1/_{25}$ roughly estimate the size of the bumps in the example.

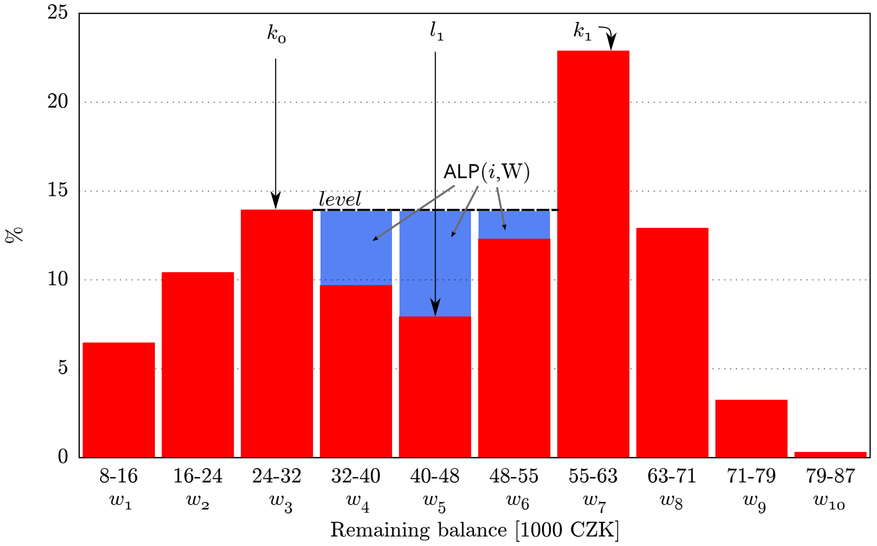

Fig. 1. Illustrative example for demonstrating the meaning of *level*, $\mathsf{ALP}(i, \mathbf{W})$, $Max = \{k_0, k_1\}$ and $Min = \{l_1\}$. The data originates from the financial dataset and show the highest balance ever to appear on client's account.

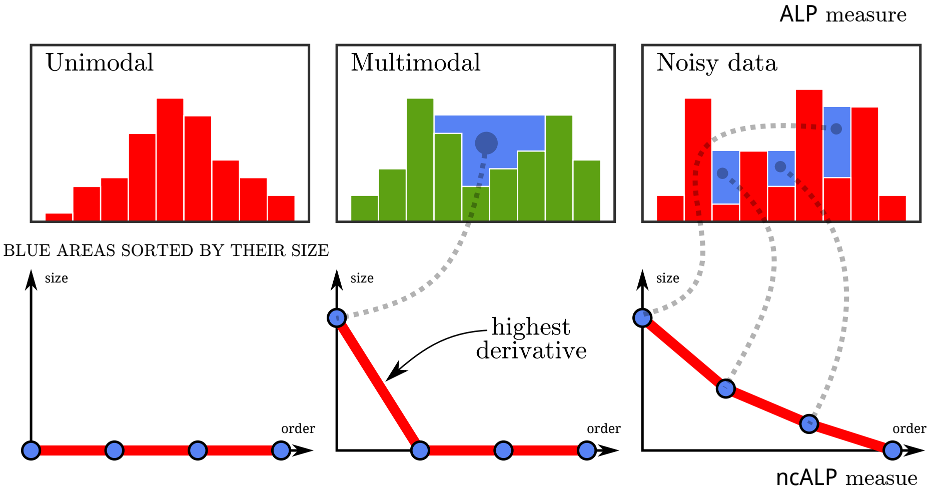

Fig. 2. Three basic types of data. Despite both multi-modal and noisy data achieve a high value of ALP, only the multi-modal should be considered as a subgroup. Hence the ncALP metric removes this deficiency by sorting the ALP values and measuring their derivative.

A visual illustration is shown in Fig. 1. The reader should avoid confusion by relating the running example to this graph. For illustrative purposes, the values were taken from the real-world dataset.

It may be tempting to say that a histogram with a high value of $\mathsf{ALP}(i, \mathbf{W})$ for some i defines a subgroup. In fact it satisfies two of the desired conditions: Firstly an area of high $\mathsf{ALP}(\cdot, \cdot)$ signalizes a low probability of a large number of successive bins and therefore k_i and k_{i+1} are good candidates for modes. Secondly the number of examples both *left of* l_{i+1} and *right of* l_{i+1} in the histogram cannot be small. If it were, the *level* would be a small probability, decreasing the value of ALP. In other words, the *coverage*, usually added into the metrics of subgroup discovery systems, emerges from our definition.

Nevertheless to include the third property of noise-cancellation, consider a histogram with a large number of local maxima and with roughly equal size of ALP. Such a large number of modes can arise either from periodically repeating data or noise; neither of which is an example of a subgroup.

Noise cancellation. The following add-on effectively limits the noise in the data by prioritizing histograms with a large value of ALP only if one of the areas of low probability is dominant. Suppose that the set $Max = \{k_i \mid 0 \leq i \leq m\}$ is reordered using a new index j s.t. $\mathsf{ALP}(j, \mathbf{W}) > \mathsf{ALP}(j+1, \mathbf{W})$ (maxima with a high ALP have lower value of j). Then the noise cancelling ALP is defined as

$$\mathsf{ncALP}(\mathbf{W}) = \max_{1 \leq j \leq m} \frac{\mathsf{ALP}(j, \mathbf{W}) - \mathsf{ALP}(j+1, \mathbf{W})}{j} \tag{15}$$

where $\mathsf{ALP}(m+1, \mathbf{W}) := 0$ is added artificially to include the smallest ALP without enforcing a piecewise definition.

The numerator of the fraction is aimed at noise-filtering, which has been commented above. The denominator disqualifies periodic data by preferring histograms with small quantity of ALP over large quantity of equally sized ALP. The effect of ncALP is illustrated in the Figure 2.

Now we can show how the function $\mathsf{ncALP}(\mathbf{W})$ is used to discover subgroups in the main algorithm.

5 Beam search strategy

We follow a beam-search strategy with refinement steps (adding an atom with variables not occurring in any other atom, unifying or instantiating a variable) according to a pre-defined language bias as usual in inductive logic programming. We start with an empty query. Unlike in standard ILP algorithms, each query in the search agenda is in exactly one of three successive phases.

1. **Germination:** Before an atom with a numeric variable is added into a clause, it can not generate any histogram. Such clauses are not evaluated and they are merely refined to create “off-springs”.

2. **Production:** Clause has a numeric variable and it generates histograms, whose handling is described below.
3. **Retirement:** When a clause becomes overly specific, the multiset of instantiations for a variable becomes empty. Given the top-down search, there is no hope for the clause to return more data after successive refinements and it can be discarded.

Clause evaluation. For each learning example and each variable v in the clause c, a multiset of instantiations is obtained from the database. The instantiations are converted into a single value using an aggregation operator f_{agg}. Given the number of bins n, these aggregated values constitute a histogram.

With all combinations of v, n and f_{agg} values, multiple histograms are constructed given a single clause c. Out of these histograms, the one which maximizes the ncALP criterion is selected and is associated with the clause. The evaluation pseudocode is shown in Program 1.

Beam search. The three phases for a clause are reflected in the Program 2. The search procedure keeps two lists of open states: $open^G$ for clauses in the germination phase and $open^P$ for clauses in the production phase. In each iteration the beam search trims $open^P$ to reduce the search space. The procedure stops refining the clause after the clause with highest score has not been updated for `MAX_NON_IMPROVEMENTS` consecutive iterations.

In the outer loop, the beam search is called repeatedly and its results are stored in the SGs set. To prevent the beam search from finding identical subgroups in each iteration, the evaluation procedure takes SGs into account. The ncALP$(\mathbf{W})$ metric is multiplied by the distance between the evaluated histogram $\mathbf{W}$ and each histogram from SGs. As equal histograms have 0 distance, the score of previously found histograms is nullified.

To conclude with technical details, the equidistant(min, max, n) returns a set of n equidistant values spread uniformly across the interval $\langle min, max\rangle$. Other functions should be self-explanatory.

6 Evaluation

The experimental evaluation of the proposed algorithm is presented in this section. Since the proposed algorithm is an example of unsupervised learning, its results can not be directly compared with an expected outcome. Instead we provide examples of discovered histograms and estimate overfitting.

Methodology. The overfitting was estimated using $k \times 2$ cross-validation in which the *training* and *testing* sets are equally large:[2] The total of 12 folds ($k = 6$) were evaluated according to the scheme:

[2] More reliable methods such as *leave-one-out* are not applicable. Histograms are created with one datapoint per example, and hence both sets needs to be large enough.

Program 1 Subroutine *evaluate*(c, SGs) to evaluate a single clause.

```
Input: The clause to evaluate $c$, the set of previously found results $SGs$
Output: Returns TOO_SPECIFIC if the clause is too specific, or NO_NUM_VAR if the
    clause has no numeric variable or the histogram with highest ncALP achievable.
    $w^* \leftarrow -1$; $\mathbf{W}^* \leftarrow$ nil
    for all $v \leftarrow$ allNumericVariables$(c)$ do
        query $\mathbf{V}_e$ from the database using $c$ and $v$
        for all $n \leftarrow \langle 3, |\mathcal{E}|\rangle$ and $f_{agg} \in \{\varnothing, \text{stdDev}, \text{median}, \min, \max\}$ do
            {Phase 1: Aggregate values for each example $e$.}
            $\mathbf{V}_\mathcal{E} \leftarrow \{\!\{\}\!\}$
            for all $e \in \mathcal{E}$ do
                query $\mathbf{V}_e$ from the database
                if $\mathbf{V}_e = \{\!\{\}\!\}$ then
                    return TOO_SPECIFIC
                end if
                $\mathbf{V}_\mathcal{E} \leftarrow \mathbf{V}_\mathcal{E} \cup f_{agg}(\mathbf{V}_e)$
            end for
            {Phase 2: Binarize the values into a histogram $\mathbf{W}$}
            $W = \{w_1, ..., w_n\} \leftarrow \text{equidistant}(\min(\mathbf{V}_\mathcal{E}), \max(\mathbf{V}_\mathcal{E}), n)$
            binarize $\mathbf{V}_\mathcal{E}$ using $W$ to produce $\mathbf{W}$
            {Update the best clause}
            $w \leftarrow \mathsf{ncALP}(\mathbf{W}) \times \prod_{\mathbf{V} \in SGs} |\mathbf{W} - \mathbf{V}|$
            if $w^* < w$ then
                $w^* \leftarrow w$; $\mathbf{W}^* \leftarrow \mathbf{W}$
            end if
        end for
        if $w = -1$ then
            return NO_NUM_VAR
        else
            return $(w^*, \mathbf{W}^*)$
        end if
    end for
```

Program 2 Beam-search for subgroup discovery algorithm

```
Input: Set of examples $\mathcal{E}$ and the language bias for the $refine$ operator.
Output: The set $SGs$ contains the subgroup-defining clauses
  $SGs = \{\}$
  loop
    $open^G_0 \leftarrow \{\top\}$; $open^P_0 \leftarrow \{\}$; $closed \leftarrow \{\}$
    $(c^*, w^*, \mathbf{W}^*) \leftarrow (\mathsf{nil}, \mathsf{nil}, \mathsf{nil})$
    $i \leftarrow 0$; $lastImprovement \leftarrow 0$
    while $(i - lastImprovement) <$ MAX_NON_IMPROVEMENTS do
      $i \leftarrow i + 1$
      $open^G_i \leftarrow \{\}$; $open^P_i \leftarrow \{\}$
      for all $c \in refine(open^G_{i-1} \cup$ clauses from $open^P_{i-1})$ do
        if $c \notin closed$ then
          $r \leftarrow evaluate(c, SGs)$
          if $r =$ TOO_SPECIFIC then
            {Retiring clauses are silently discarded.}
          else if $r =$ NO_NUM_VAR then
            $open^G_i \leftarrow open^G_i \cup \{c\}$
          else
            $(w, \mathbf{W}) \leftarrow r$
            $open^P_i \leftarrow open^P_i \cup \{(c, w, \mathbf{W})\}$
          end if
        end if
      end for
      {Update the best clause if necessary.}
      $(c, w, \mathbf{W}) \leftarrow$ the best clause from $open^P_i$
      if $w^* = \mathsf{nil}$ or $w^* < w$ then
        $(c^*, w^*, \mathbf{W}^*) \leftarrow (c, w, \mathbf{W})$
        $lastImprovement \leftarrow i$
      end if
      {Narrow the beam for performance}
      $closed \leftarrow closed \cup open^G_i \cup$ clauses from $open^P_i$
      $open^P_i \leftarrow$ take BEAM_WIDTH best results from $open^P_i$
    end while
    if $\mathbf{W}^* \notin SGs$ and $\mathbf{W}^* \neq \mathsf{nil}$ then
      $SGs \leftarrow SGs \cup \{\mathbf{W}^*\}$
    else
      return $SGs$
    end if
  end loop
```

1. Induce 5 first-order clauses from the *training* set using the algorithm as described in Program 2. Remember the clause c, the query variable v and aggregation function f_{agg} which served for obtaining the histogram $\mathbf{W}_{train}$.
2. Using the same c, v and f_{agg} applied to the testing set, obtain $\mathbf{W}_{test}$.
3. Compute the ratio $r = \mathsf{ncALP}(\mathbf{W}_{test})/\mathsf{ncALP}(W_{train})$.

This method ends up with 60 values of r ($12 \cdot 5 = 60$). A single representative value is then computed as the geometric mean $R = \sqrt[60]{r_1 \cdot r_2 \cdots r_{60}}$. Such value decreases with overfitting: Ideally $R = 1$, but if bumps found in the training set do not exists in the testing set, then R goes to 0.

Note that if there is a single value of $r = 0$, the value of R becomes 0 destroying any information in other r values. To overcome this issue, the amount of zero values among r was saved as D and then R was computed only from positive values of r.[3]

Datasets. We tested the proposed algorithm on seven real-world datasets to see if it is able to discover non-trivial yet interpretable subgroups, and to quantify any possible tendency towards overfitting. We used the financial dataset [2], two parts of the mutagenesis dataset [18] (regression-friendly, -unfriendly), and four genomic datasets [10], each describing the relational structure of a biochemical pathway along with numerical values of expressions of the involved genes in a single phenotype (glioma samples).

Dataset		**Disappeared bumps (D)**	**ncALP ratio on testing:training set (R)**
mutagenesis	*easy*	0%	90%
	hard	0%	64%
financial		4%	48%
gene expr.	*100_pw_hsa00190*	30%	41%
	105_pw_hsa04360	26%	37%
	108_pw_hsa01430	20%	32%
	118_pw_hsa04514	37%	15%

Table 1. Results for different datasets.

Comments on the results. The results for different datasets are presented in the Table 1. Although the values do not offer comparison with different algorithms, a pattern seems to emerge. The R value tend to be high for easily classifiable datasets. The *mutagenesis easy* achieves the highest score, followed by its *hard* counterpart and the *financial* dataset.

We ascribe the low performance on the gene expression data to two factors. Firstly the gene expression data from microarray is rather small and contain a

[3] E.g. $D = 20\%, R = 75\%$ means that one fifth of histograms in the testing set contain no bump and the remaining ones are 1/4 less significant in average.

large amount of noise, which might get captured by the histograms. Secondly, the gene dataset contains a limited amount of background knowledge, resulting in the largest clause having only two atoms. Nevertheless this evidence together with a good performance on other datasets show that the algorithm does not provide sufficient measure of confidence on the subgroup stability, which will need to be incorporated.

Histogram interpretation.

- Fig. 3 shows a large bump in urbanization of client's region of residence. This histogram is of particular interest for two reasons. Firstly it is easily interpretable given the structure of Czech Republic. Unlike all other parts of the country, Prague constitutes a single region with urbanization reaching 100%, which explains the bump at the right side of the plot. Secondly this histogram shows the relational nature of the data. 3 relations must be linked in the dataset (loans, accounts and districts) in order to create this plot.
- Fig. 4 displays the highest balance appearing on client's account. We are convinced that this is a very good example of a multimodal distribution.

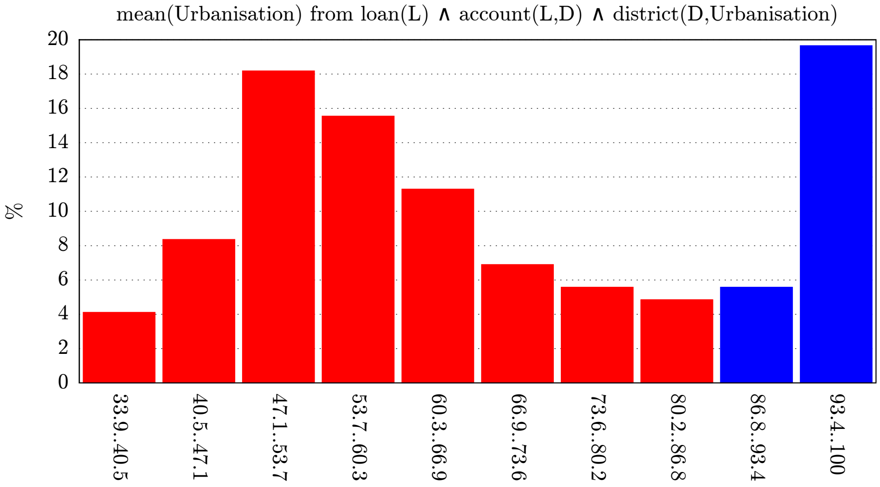

Fig. 3. Example of a discovered subgroup: Mean of relative urbanization (percentage of population living in cities) of regions, where clients live. The values of the analysed variable in the substitution set are aggregated with respect to the main relation loan so that the vertical axis corresponds to the frequency of loans.

7 Conclusions and future work

We proposed a new concept of relational subgroup discovery based on searching bumps in histograms constructed from substitution sets derived from matching

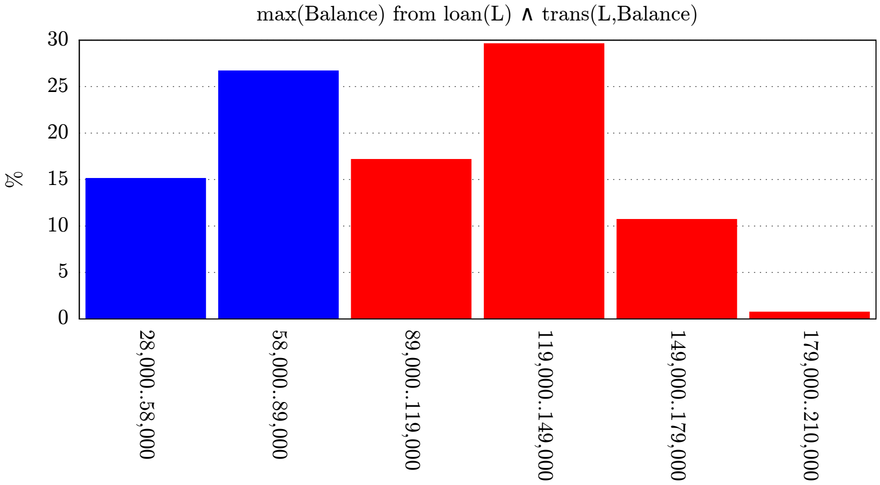

Fig. 4. Example of a discovered subgroup: Max balance after executing a transaction. The values of the analysed variable in the substitution set are aggregated with respect to the main relation loan so that the vertical axis corresponds to the frequency of loans.

a first-order query against a database. The fact that a relatively simple implementation of the concept was able to discover non-trivial interpretable subgroups that generally survive from training data to testing data is reassuring and calls for further advancement of the algorithm. This should mainly focus on improving the search strategy. Currently the search for the query, the optimization of the number of bins, and the evaluation of the histogram-heuristic are rather loosely coupled. A tighter integration would likely improve the overall performance, and might allow to find more complex patterns.

Acknowledgement

We acknowledge the financial support through the grant 103/10/1875 of the Czech Science Foundation.

References

1. Atzmueller, M., Lemmerich, F.: Fast subgroup discovery for continuous target concepts. In: Foundations of Intelligent Systems. Springer (2009)
2. Berka, P., Sochorová, M.: Guide to the financial data set (1999), `http://lisp.vse.cz/pkdd99/berka.htm`

3. Escobar, M.D., M.West: Bayesian density estimation and inference using mixtures. Journal of the American Statistical Association 90, 577–588 (1994)
4. Friedman, J.H., Fisher, N.I.: Bump hunting in high-dimensional data. Statistics and Computing 9, 123–143 (1999)
5. Grosskreutz, H., Rüping, S.: On subgroup discovery in numerical domains. Data Mining and Knowledge Discovery 19, 210–226 (2009), 10.1007/s10618-009-0136-3
6. Kavšek, B., Lavrač, N.: APRIORI-SD: adapting association rule learning to subgroup discovery. Applied Artificial Intelligence 20(7), 543–583 (Sep 2006), http://www.tandfonline.com/doi/abs/10.1080/08839510600779688
7. Klösgen, W.: Explora: a multipattern and multistrategy discovery assistant. In: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.) Advances in knowledge discovery and data mining, p. 249–271. American Association for Artificial Intelligence, Menlo Park, CA, USA (1996), http://dl.acm.org/citation.cfm?id=257938.257965
8. Kralj-Novak, P., Lavrač, N., Webb, G.I.: Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. Journal of Machine Learning Research 10, 377–403 (2009)
9. Krogel, M.A., Wrobel, S.: Transformation-based learning using multirelational aggregation. In: Inductive Logic Programming, pp. 142–155. Springer (2001)
10. Kuželka, O., Szabóová, A., Holec, M., Železný, F.: Gaussian logic for predictive classification. In: ECML/PKDD'2011: Eur. Conf. on Machine Learning / Principles and Practice of Knowledge Discovery in Databases. Springer (2011)
11. Landwehr, N., Passerini, A., De Raedt, L., Frasconi, P.: Fast learning of relational kernels. Machine Learning 78(3), 305–42 (2010)
12. Landwehr, N., Kersting, K., De Raedt, L.: nFOIL: integrating naive bayes and FOIL. In: Proceedings of the 20th national conference on Artificial intelligence - Volume 2. p. 795–800. AAAI Press (2005), http://dl.acm.org/citation.cfm?id=1619410.1619460
13. Landwehr, N., Passerini, A., De Raedt, L., Frasconi, P.: kFOIL: learning simple relational kernels. In: Proceedings of the 21st national conference on Artificial intelligence - Volume 1. p. 389–394. AAAI Press (2006), http://dl.acm.org/citation.cfm?id=1597538.1597601
14. Lavrač, N., Kavšek, B., Flach, P., Todorovski, L.: Subgroup discovery with CN2-SD. Journal of Machine Learning Research 5, 153–188 (2004)
15. Lowthian, P., Thompson, M.: Bump-hunting for the proficiency tester – searching for multimodality. The Analyst 127(10), 1359–1364 (2002)
16. de Raedt, L.: Logical and relational learning. Springer (Oct 2008)
17. Silverman, B.W.: Using kernel density estimates to investigate multimodality. Journal of the Royal Statistical Society. 43(1), 97–99 (1981)
18. Srinivasan, A., Muggleton, S.H., Sternberg, M.J.E., King, R.D.: Theories for mutagenicity: A study in first-order and feature-based induction. Artificial Intelligence 85, 277–299 (1996)
19. Železný, F., Lavrač, N.: Propositionalization-based relational subgroup discovery with RSD. Machine Learning 62(1-2), 33–63 (2006)
20. Wrobel, S.: An algorithm for multi-relational discovery of subgroups. In: Principles of Data Mining and Knowledge Discovery, pp. 78–87. Springer (1997)
21. Yukizane, T., Ohi, S.Y., Miyano, E., Hirose, H.: The bump hunting method using the genetic algorithm with the extreme-value statistics. IEICE - Trans. Inf. Syst. E89-D, 2332–2339 (2006)