Data Mining and Knowledge Discovery (2021) 35:158–212
https://doi.org/10.1007/s10618-020-00703-x

For real: a thorough look at numeric attributes in subgroup discovery

Marvin Meeng[1] · **Arno Knobbe**[1]

Received: 28 June 2019 / Accepted: 20 June 2020 / Published online: 21 September 2020

Abstract

Subgroup discovery (SD) is an exploratory pattern mining paradigm that comes into its own when dealing with large real-world data, which typically involves many attributes, of a mixture of data types. Essential is the ability to deal with numeric attributes, whether they concern the target (a regression setting) or the description attributes (by which subgroups are identified). Various specific algorithms have been proposed in the literature for both cases, but a systematic review of the available options is missing. This paper presents a generic framework that can be instantiated in various ways in order to create different strategies for dealing with numeric data. The bulk of the work in this paper describes an experimental comparison of a considerable range of numeric strategies in SD, where these strategies are organised according to four central dimensions. These experiments are furthermore repeated for both the classification task (target is nominal) and regression task (target is numeric), and the strategies are compared based on the quality of the top subgroup, and the quality and redundancy of the top-k result set. Results of three search strategies are compared: traditional beam search, complete search, and a variant of diverse subgroup set discovery called cover-based subgroup selection. Although there are various subtleties in the outcome of the experiments, the following general conclusions can be drawn: it is often best to determine numeric thresholds dynamically (locally), in a fine-grained manner, with binary splits, while considering multiple candidate thresholds per attribute.

Keywords Subgroup discovery · Supervised local pattern mining · Numeric attributes · Discretisation

Mathematics Subject Classification H.2.8: Data mining

Responsible editor: Johannes Fürnkranz

✉ Marvin Meeng
meeng@liacs.nl

Extended author information available on the last page of the article

1 Introduction

This work considers an aspect of subgroup discovery that has been insufficiently addressed by the existing literature: numeric data. Although numeric attributes have been the subject of a number of recent papers, this work will explain and empirically demonstrate that this coverage has been incomplete, and that actually superior results can be obtained by a more thorough treatment of numeric attributes. Essentially two areas exist where the presence of numeric attributes requires attention: on the side of the target attribute(s) (in the case of a regression setting), and on the side of the description attributes (those attributes that are not targets, and are available to construct subgroups from). On the target side, several recent papers discuss the treatment of numeric target attributes (Atzmüller and Lemmerich 2009; Boley et al. 2017; Lemmerich et al. 2012, 2013, 2016), but all these papers describe methods that essentially assume nominal description attributes.

Conversely, on the description side, only few papers discuss algorithms that incorporate substantial treatment of numeric description attributes (Belfodil et al. 2018; Bosc et al. 2018; Grosskreutz and Rüping 2009), but they use or require a binary target. Consequently, none of the papers described here cover the combination of numeric description attributes and a numeric target. With the possible exception of the MIDOS system (Wrobel 1997), one of the first SD systems, this makes the approach analysed here in detail the only one that provides full numeric capability. The combined approach is central to the design of Cortana,[1] the SD system that the authors have worked on for over 10 years (Duivesteijn 2013; Duivesteijn et al. 2010, 2012; Konijn et al. 2013, 2015; Mampaey et al. 2012, 2015; Meeng and Knobbe 2011, 2020; Meeng et al. 2014; Duivesteijn and Meeng 2016)

Of course, when arguing that many SD systems do not allow numeric description attributes, the immediate counterargument is that this limitation is easily resolved with discretisation prior to the SD run. This is true indeed, but such a discretisation step might permanently remove information, leading to a loss of precision and suboptimal results. And while discretisation is an important, if not crucial, tool in SD, because an unequivocal subgroup description requires clear boundaries in the continuous domain, the question is at what stage in the analysis this switch from continuous to discrete is best made. This work argues and demonstrates that dynamic, or *local*, discretisation, in other words thresholding the data while performing the search for subgroups, is generally preferable over pre-discretisation (*global*).

Aside from loss of precision, there is another, more subtle downside to the naive discretisation performed by most SD systems. By replacing numeric attributes with nominal approximations, one not only lowers the resolution, but also destroys the order information stored in the numeric representation: it is no longer clear that 'high' is greater than 'medium' and 'low', it is just *different*. A corollary of this loss of order is that it is also no longer possible to investigate subgroups of different sizes, by combining more or fewer consecutive labels (e.g. 'low' and 'medium'). An undesirable side effect of this type of discretisation is that the size of candidate subgroups is too directly tied to the granularity of the discretisation. The alternative approach is to

[1] http://datamining.liacs.nl/cortana.html.

discretise the data while retaining the order information, for which a slightly richer representation is needed.

This work identifies a number of dimensions over which SD algorithms can vary in their treatment of numeric attributes. These include the choice when to apply discretisation, whether to do classical (nominal) or order-preserving discretisation, but also other options, such as whether to do a fine or coarse discretisation. Although one could examine the effect of choices made for each of these dimensions in isolation, such an analysis would miss interactions between them. So, besides analysing the effects of various parameter choices within individual dimensions, this work also offers a systematic analysis of the combined effects of these choices over all examined dimensions, as is relevant in real-world analyses.

Table 1 gives an overview of the different numeric strategies that are examined in this work. Each line in the table represents a separate strategy, identified by the different choices for the following dimensions: *discretisation timing*, *interval type*, *granularity*, and *selection method*. Of these dimensions, the first three are related to *candidate generation*, whereas the last relates to *candidate selection*. Strategies are referred to by a combination of a number and an acronym formed from the first character of their values for the aforementioned dimensions. As the table suggests, there are many different options to compare, and extensive experimentation is required to shed light on the best choice for each dimension and the optimal combination of settings. The usefulness of each strategy is evaluated for discrete (classification) and numeric (regression) target attributes, using multiple quality measures and search strategies, and results are compared based on both subgroup quality and redundancy. The dimensions and strategies are outlined in more detail in Sect. 3.

The contributions of this paper can be summarised as follows:

- An outline is provided of how numeric data plays a role in SD on the description side, and on the target side.
- Four dimensions relevant to dealing with numeric description attributes are identified, leading to a theoretical sixteen possible configurations.
- Extensive experimentation is discussed in order to compare the different configurations on a range of datasets.
- Experiments are performed in both a classification (nominal target) and a regression (numeric target) setting.
- For each target type, experiments are performed using three quality measures that favour different aspects of subgroups.
- The optimal choice per dimension is reported, as is an overall ranking of configurations, for both target types, considering all evaluated quality measures.
- With respect to result (set) quality, a comparison is made between complete search and traditional beam search.
- With respect to subgroup redundancy, complete search is compared to both a traditional beam search and a specialised redundancy-reducing beam search strategy.

The remainder of this work is organised as follows. The introduction is completed by listing the four central dimensions and discussing some key aspects of SD. Section 2 provides the necessary foundations, including pseudocode for the generic SD algorithm that will be applied. Section 3 covers the various numeric strategies and

Table 1 Dimensions over which subgroup discovery algorithms dealing with numeric description attributes can vary

strategy	discretisation timing	interval type	granularity	selection method	included
1-lbfa	local	binaries	fine	all	✓
2-lbfb				best	✓
3-lbca			coarse	all	✓
4-lbcb				best	✓
5		nominal	fine	all	[a]
6				best	[a]
7-lnca			coarse	all	✓
8-lncb				best	✓
9-gbfa	global	binaries	fine	all	✓
10-gbfb				best	✓
11			coarse	all	[b]
12				best	[b]
13		nominal	fine	all	[a]
14				best	[a]
15-gnca			coarse	all	✓
16				best	[c]
17-lxfb	local	–	fine	best	✓

Strategies not considered in this work are listed without acronym
[a] This would create single value intervals, these are generally uninformative.
[b] This would be the same as performing a more coarse discretisation with the fine variant
[c] No algorithm like this exists

how these can be organised according to four dimensions. Section 4 describes the relevant literature. Section 5 contains the bulk of this work, with a series of experiments investigating the different settings empirically, as well as discussions about the results. Section 6 presents general conclusions and lists future work.

2 Preliminaries

2.1 Data

Throughout this work, the following definition of a dataset is used:

Definition 1 (Dataset) A dataset $\mathcal{D}$ is a bag of N *records* $\mathbf{r}^i \in \mathcal{D}$ of the form: $\mathbf{r}^i = \left(a_1^i, \ldots, a_l^i, t^i\right)$, with l a positive integer from $\mathbb{Z}^+$.

Here, $a_1^i, \ldots, a_l^i$ are the values of $\mathbf{r}^i$ for the *description attributes* $a_1, \ldots, a_l$, and t^i is the value for the *target attribute* t. In general, both description and target attributes can be taken from an unrestricted domain $\mathbb{A}$. When a nominal attribute serves as target, a single *target value* needs to be assigned, and it comes from its domain of class labels. Targets can also be formed by a numeric attributes, taken from $\mathbb{R}$. In this case, no target value is assigned.

When considering subgroups, the central concept of SD, a distinction should be made between their intensional and extensional part. Definition 2 covers the first.

Definition 2 (Description) A description is a function: $I : \mathbb{A}^l \rightarrow \{0, 1\}$, where I *covers* a record $\mathbf{r}^i$ *iff*: $I\left(a_1^i, \ldots, a_l^i\right) = 1$.

Typically, the *description language* $\mathcal{I}$ in SD consists of (conjunctions of) conditions on description attributes of the general form 'a_i *operator value*'. Examples include '*Smokes* = *false*' and '*EyeColour* = *brown* $\wedge$ *Length* $\geq$ *1.76*'. The notation $I_\varnothing$ is used for the empty description, which imposes no restrictions. It selects the entire dataset and can be refined by adding a first conjunct. The *depth* of a subgroup is defined as the number of conjuncts in the description.

As noted above, a subgroup description could be said to *precede* a subgroup extension, in that it is through the description that a search algorithm imposes restrictions on the data to select a subset of records. Definition 3 expresses this relation.

Definition 3 (Extension) An extension $\mathcal{E}_I$ corresponding to a description I is the bag of records $\mathcal{E}_I \subseteq \mathcal{D}$ that I covers: $\mathcal{E}_I = \left\{\mathbf{r}^i \in \mathcal{D} \mid I\left(a_1^i, \ldots, a_l^i\right) = 1\right\}$.

From now on, the subscript I is omitted if no confusion can arise, and a subgroup extension is simply referred to as $\mathcal{E}$.

The explicit differentiation of the intensional and extensional facets of a subgroup is required in some SD algorithms (van Leeuwen and Knobbe 2012). However, for the remainder of this work, s denotes a subgroup, encompassing both its intension and extension. For any particular subgroup s, with extension $\mathcal{E}$, n denotes its size, that is, the number of records in that subgroup: $n = |\mathcal{E}|$.

2.2 Subgroup discovery

Subgroup Discovery is a local, supervised, descriptive, pattern mining paradigm. These three aspects set it apart from other paradigms (such as classification), and entail certain behaviour discussed in more detail here.

Descriptive Since the inception of SD (Klösgen 1992, 1996; Wrobel 1997), subgroups are taken to consist of both an intensional and extensional part. The former is the subgroup *description*, the latter is the *extension*, the subset of records that is selected through this description. Although an extension is relevant, without the accompanying concise description in terms of the available attributes it is of limited use. The paradigm would be simply *subset* discovery, rather than Subgroup Discovery.

Moreover, multiple very different descriptions that select similar subsets can all yield new insights individually, and learning about their correlation can expand knowledge as well. In this, redescription mining (Galbrun and Miettinen 2017) is related.

Local The goal of SD is to identify interesting local models, by means of subgroups, of which multiple (potentially overlapping) ones might exist. Here, local means that the quality of a subgroup is independent from any other findings, which is in contrast with the global models that dominate other paradigms such as classification or regression.

Its local nature allows both identifying small parts of the data that behave exceptionally and fully capturing overlapping patterns of which global models can only identify part of the knowledge.

Supervised As a supervised paradigm, SD optimises with respect to a predefined target. Various (multivariate) target types exist, but this work is confined to single nominal (in fact, binary) and single numeric targets. For these two types, one typically looks for subsets of the data with a substantially higher share of either positive or negative cases, and for subset with a substantially higher or lower average value than can be expected from that of the entire dataset, respectively.

Combining the supervised and local aspects, SD potentially values smaller subsets of the data with an interesting target distribution. How interesting this target distribution is, is quantified by a *quality measure*, where the typical quality measure strikes a balance between how unusual a subset is, and how large the subset is (in other words, how reliable the observed phenomenon is).

Overlap Although some degree of (extension) overlap is intentional and inherent to the SD paradigm, Section Selection Method below, where so-called *saturation effects* are discussed, will demonstrate redundancy should be suppressed in both the candidate and result set. In this respect, specialised redundancy-reducing methods (Bosc et al. 2018; Kaytoue et al. 2011; Knobbe and Ho 2006a, b; Lavrač and Gamberger 2004; Lemmerich et al. 2013; Meeng et al. 2014) can help to avoid saturation and to improve diversity among the reported subgroups. This work evaluates one variant of Diverse Subgroup Set Discovery (van Leeuwen and Knobbe 2011, 2012) called cover-based subgroup selection (CBSS).

Some of the above include (intermediate) post-processing procedures to reduce saturation and redundancy. Another approach to prevent these is to avoid the many variations in the first place. All but one of the strategies listed in Table 1 achieve this naturally.

2.2.1 Subgroup discovery algorithm

In order to perform the experiments with different numeric strategies under comparable circumstances, a generic algorithm is introduced that can be parameterised in a number of ways. As such, this SDMM algorithm, which also features in Cortana, can implement the various settings that are analysed. This section presents the various aspects of this generic algorithm, including its description language, the discretisation algorithm, its search strategies, and the quality measures.

Description Language A description language in SD determines the nature of the descriptions it will consider and report. In the majority of SD implementations, as in this paper, descriptions consist of a *conjunction of conditions* on individual attributes. Deriving more complex subgroups from simpler ones by adding conjuncts to the description one by one is known as *refinement*, and is the principal way of traversing the search space. The attractive property of conjunctions is that the size of the subgroup never grows with refinement. The algorithm presented below is slightly more strict, as

it requires that fewer records in the dataset are covered after adding a new condition to a conjunction, such that it selects a proper subset.

Quality Measure A quality measure objectively evaluates a candidate description in a given dataset. For each description I in the description language $\mathcal{I}$, a quality measure is a function that quantifies how interesting the subgroup s is.

Definition 4 (Quality Measure) A *quality measure* is a function $\varphi_{\mathcal{D}} : \mathcal{I} \rightarrow \mathbb{R}$ that assigns a unique numeric value to a description I and its associated extension $\mathcal{E}_I$, given a dataset $\mathcal{D}$.

A quality measure quantifies various aspects of a subgroup (Fürnkranz and Flach 2005), and in the choice of quality measure, the analyst indicates their preference for certain aspects of the desired subgroups. As different target types and quality measures entail very different search and solution spaces (Mampaey et al. 2012, 2015), six measures are evaluated, three for each target type. Details of these measures and their respective properties are provided in the experimental section.

Search Constraints In principle, SD algorithms aim to discover subgroups that score high on a quality measure. But, it is common practice to also impose additional constraints on subgroups that are found by SD algorithms. Usually these constraints include lower bounds on the quality of the description ($\varphi_{\mathcal{D}}(I) \geq p_1$) and the size of the induced subgroup ($n \geq p_2$). Also, an upper limit is often set on the search depth (d).

Algorithm 1: SDMM($\mathcal{D}$, $\varphi_{\mathcal{D}}$, $\mathcal{P}$)

```
   input : dataset D, quality measure φ_D, search constraints P
   output: final result set F
1  F ← ∅, S ← ∅
2  S ← S ∪ I_∅
3  while S ≠ ∅ do
4      s ← SelectSeed(P, S)
5      S ← S \ s
6      R ← GenerateRefinements(s, D, P)
7      foreach r ∈ R do
8          score ← φ_D(r)
9          AddToCandidateSet(P, S, r, score)
10         AddToResultSet(P, F, r, score)
11 return F
```

Pseudocode Algorithm 1 describes the generic SD algorithm SDMM. The set of search parameters $\mathcal{P}$ holds all required parameters, including target information, the selection method used, and search constraints. On line 2, the empty description $I_\varnothing$ is added to $\mathcal{S}$, the set of subgroups that will act as candidates for refinement. This description selects the whole dataset, as it poses no restriction on it. The `SelectSeed`

Algorithm 2: EQUALFREQUENCYDISCRETISATION($\mathbf{a}$, B, o)

input : $\mathbf{a}$, vector of ascending values; B, desired number of bins; o, refinement operator used
output: set of cut points $\mathcal{B}$

1 $\mathcal{B} \leftarrow \varnothing, n \leftarrow |\mathbf{a}|$
2 **for** $b = 1$ **to** $B - 1$ **do**
3 $\quad x \leftarrow \lfloor nb/B \rfloor$
4 $\quad$ **if** $o \neq$ '$\geq$' **and** $(nb \texttt{ modulo } B) = 0$ **then** $x \leftarrow x - 1$
5 $\quad \mathcal{B} \leftarrow \mathcal{B} \cup \mathbf{a}[x]$
6 **return** $\mathcal{B}$

function (line 4) selects the appropriate candidate. The selection is based on the quality and canonical order of the candidates, as defined by (the conditions in) its description.

Further, to accommodate different search space exploration strategies in a single generic algorithm, `GenerateRefinements` (line 6) adapts its behaviour accordingly. For level-wise searches, refinements are created only for the current search level, and relevant refinements are then added to the set of candidates $\mathcal{S}$ for the next level. For other forms of search space traversal (e.g. depth-first or breadth-first search), `GenerateRefinements` will show slightly different behaviour.

Finally, the addition of a subgroup to the candidate set $\mathcal{S}$ (line 9) and the final result set $\mathcal{F}$ (line 10) is performed by specialised functions that check against search constraints and take care of trimming, re-ordering, or other post-processing of these sets, if required. The addition rules for $\mathcal{S}$ and $\mathcal{F}$ slightly differ regarding minimum subgroup size ($mincov + 1$ for $\mathcal{S}$), and the maximum size and minimum quality constraint are irrelevant for $\mathcal{S}$. Regarding the latter, optimistic estimates (Grosskreutz and Rüping 2009; Wrobel 1997) appreciate that candidate with a score below the minimum quality might be refined into good (the best possible) subgroups.

It is good to note that the SDMM algorithm is a *local-refinement* algorithm. Local refinement means that descriptions are generated based only on the local domain of a subgroup extension, and not on the domain of the entire dataset. This for example entails that more accurate cut points will be generated, the further the search progresses to smaller subsets of the data. Local refinement also allows minimum coverage pruning and avoiding meritless conjunctions, thus reducing search complexity.

2.3 Discretisation

In the context of this work, all discretisation is performed using Algorithm 2, EQUALFREQUENCYDISCRETISATION. Also, the number of cut points is static, meaning it is the same for all attributes, throughout a search (Dougherty et al. 1995).

The algorithm takes a few inputs. A vector of values $\mathbf{a}$ forms the current input domain, which can either be the entire attribute data, or only those values covered by the subgroup under investigation. By design of the SDMM algorithm, domain values never require sorting during search and are always in ascending order. The desired number of bins B, and the operator o that will be used for (the conditions using) the returned cut points are also required.

The selected cut points define consecutive intervals that cover (approximately) the same number of values. Equal coverage is impossible when n/B does not produce an integer and in case of duplicate values around cut points. These are fundamental problems of discretisation, but, for more than a decade, this heuristic has shown to generally produce (close to) equal coverage.

Bounds in $\mathcal{B}$ are always inclusive. For operator '$\geq$', they are the left (lower) bounds of the half-bounded intervals. For other operators, they are the right (upper) bounds, as used in Mampaey et al. (2012) and Mampaey et al. (2015)). As an example, for $\mathbf{a} = [w, x, y, z]$ and $B = 2$, line 4 produces $\geq y$ and $\leq x$, for operators '$\geq$' and '$\leq$', respectively. For $\mathbf{a} = [x, y, z]$ and $B = 2$, both operators select intervals covering two values ($\geq y$ and $\leq y$). The effect of the operation on line 4 is equal to using $\mathbf{a}$ with descending values.

In general, other methods for discretisation could be used equally well. These could be supervised techniques for classification targets (Fayyad and Irani 1993), single (Kontkanen and Myllymäki 2007) or multidimensional MDL-based methods (Nguyen et al. 2014), or V-optimal discretisation (Ioannidis 2003). This work uses Algorithm 2 exclusively, as it is fast, simple, and applicable in both the classification and regression setting.

It further works well, because, as SD is supervised, good subgroups select a subset of records for which the target distribution is favourable (positives or numeric extremes). So, although the selection of the cut points from the description attribute domain is unsupervised, this information of the target is always taken into account when generating refinements. Also, SDMM performs local refinement (see Sect. 2.2.1), making the binning progressively finer and to the point.

3 Numeric strategies

3.1 Dimensions

This work revolves around the different ways in which numeric attributes can be treated in SD. Different possible strategies are identified, by means of four dimensions. Below, the dimensions are described in more detail. A worked example of the described dimensions can be found at the end of this section.

Discretisation Timing SD deals with numeric data by setting conditions on the included values, typically by requiring values to be above or below a certain threshold. In realistic, non-trivial data, the continuous domain is extensive, and one needs to select a subset of reasonable values in order for the SD process to be remain tractable (discretisation).

One option, *global* discretisation, performs this by replacing the original values prior to analysis, based on all the data. The *local* alternative determines suitable cut points dynamically whenever a numeric attribute is encountered while mining. Consequently, it has the option of choosing cut points appropriate for the subset of data under investigation at any point in the search space.

The dimension referred to as *discretisation timing* distinguishes the two options.

Interval Type The term *interval type* refers to the way in which a given set of cut points is treated to produce candidate subgroups. In the context of discretisation, it is customary to take $B-1$ cut points, and create a single nominal feature to represent in which of the B intervals (bins) the numeric value falls. Subgroups are then formed by setting the derived feature to one of these values.

Although this approach is very popular, due to its straightforward discretisation and ease of interpretation, it also has fundamental limitations. Because of its consecutive nature, subgroups can never cover more than roughly $1/B$ of the data, making this approach rather inflexible, and too directly dependent on the setting of B. A more flexible alternative is to (conceptually) translate the $B-1$ cut points into $B-1$ binary features, each corresponding to a binary split on the respective cut point, thus allowing a wide range of sizes. The resulting overlapping features might better capture the subsets in the data with favourable target values (positives or numeric extremes).

The two values for the interval type, *nominal* and *binaries*, now correspond to the two approaches described here.

Granularity The term *granularity* is used to describe how (many) candidates are generated given a numeric input domain, and the possible choices are *fine* and *coarse*.

In case of *fine*, every value from the input domain is used to generate candidates. For *coarse*, only a selected number of values from the input domain are used to generate candidates. For the latter process, discretisation techniques can be used. In granularity, there is a trade-off between the computation cost and the chances of finding the optimal subgroups. A *fine* strategy may produce good subgroups, but the subgroups produced by a similar *coarse* approach might be as good or only marginally worse, at a fraction of the computation time.

Selection Method The dimensions above all relate to candidate generation, influencing which candidate subgroups are generated and evaluated by the description generator. Besides these *candidate generation* dimensions, there is also a *candidate selection* dimension to an SD algorithm. Candidate selection refers to the process used to include generated candidates into the final result set and/or use them for the remainder of the search. On the set of all valid generated candidates, two selection methods can be applied, *all* and *best*.

The *all* method does not filter out any of the generated candidates, meaning that all valid candidates will be included in the result set and/or will be available for the remainder of the search process. In contrast, the *best* method allows only the single best of all valid candidates for a given numeric attribute to continue.

When refining a single attribute, the number of generated candidates is equal for these options, as determining the best candidate requires evaluating all. Still, depending on other settings, there might be huge differences, both in terms of computation and redundancy. In case of exhaustive search, *all* is computationally more expensive, but, as it makes the search less greedy (more exploration), it might yield better results. In combination with beam search, there is no essential difference, as the beam width limits the size of the search space.

Redundancy is a common issue with *all*, due to saturation effects. Especially in combination with *fine*, a single attribute might produce many similar high-valued

Table 2 Example showing discretisation of column 'height' using the *nominal* (height_n) and *binaries* (height_l, height_{lm}) strategy for all data, and for the subgroup '$gender = female$' only (v_n and v_l, v_{lm})

gender	height	height_n	height_l	height_{lm}	v_n	v_l	v_{lm}
female	153	a	1	1	a	1	1
male	166	a	1	1			
female	169	a	1	1	a	1	1
female	170	a	1	1	b	0	1
female	171	b	0	1	b	0	1
female	174	b	0	1	c	0	0
male	178	b	0	1			
male	179	b	0	1			
male	180	c	0	0			
male	180	c	0	0			
female	182	c	0	0	c	0	0
male	187	c	0	0			
male	190	c	0	0			

subgroups. Result sets then show little variation, and additionally, having many such candidates in a beam undermines exploration, as greediness *increases*.

This demonstrates that an optimal choice for this dimension is not obvious and relies on other choices. Therefore, thorough experimental analysis is required that gauges combined effects.

Example Table 2 shows an example that demonstrates the effects of various options described for the dimensions above. The table contains the height of 13 women and men. A straightforward discretisation in three bins (over the entire dataset) would produce the following cut points: ≤ 170 and ≤ 179. The three conceptual bins can be represented in two alternative ways: a single nominal attribute to represent three bins of size roughly 13/3 (column height_n), or two binary columns, one representing the *low* bin and one representing the *low/medium* bin (columns height_l and height_{lm}). Column height_n is the result of a *nominal* strategy, columns height_l and height_{lm} of the *binaries* strategy.

Note that negations (complements) and conjunctions of these two features suffice to generate all other possibilities. Also, the example demonstrates a *coarse* granularity, since only a modest number of cut points is produced. A *fine* discretisation would look similar, but just with more (12) nominal values, and more binary features.

The cut points mentioned here are relevant for a *global* discretisation (or for subgroups at depth 1). However, at greater depths, these cut points are suboptimal, since a subset with a different distribution of *height* might exist. For example, if search would reach subgroup '$gender = female$', cut points ≤ 169 and ≤ 171 would be more balanced, providing a 2/2/2 discretisation, rather than the 3/2/1 discretisation produced by the global cut points. This is demonstrated in the last three columns (the column names v (for *virtual*) indicate that these features are typically not materialised). The local cut points differ in two important aspects from the global ones. First, the cut points are placed at smaller values, since on average, the women in the dataset are

less tall. And second, the cut points are placed closer together, demonstrating a higher resolution at greater search depths. A *local* strategy would produce such cut points dynamically during the search process.

3.2 Dimensions table

Table 1 in the Introduction already presented an overview of the different strategies that are examined in this work. Since the choice for each of the four dimensions described above is binary, this leaves a combined total of sixteen strategies. To these sixteen, one extra strategy is added (see below), as it does not properly fit within the framework of dimensions described above. However, a number of strategies are considered to be not useful, and thus not included in our experiments.

First, in a *nominal* setting, consecutive bounded intervals are created, and when this is done for each unique value in the input domain, as per the *fine* setting, this would result in single value intervals. This behaves the same as using '=' on numeric values and, in general, it leads to uninformative descriptions and tiny subgroups. As this is unaffected by the parameter settings for both the dimensions *discretisation timing* and *selection method*, all four strategies combining *nominal* and *fine* are omitted from the experiments below. They are 5-lnfa, 6-lnfb, 13-gnfa, and 14-gnfb.

Also not included are strategies that involve the combination *global*, *binaries* and *coarse* (11-gbca and 12-gbcb). The reasoning here is that *global* yields a fixed discretisation that reduces the cardinality of the data (and therefore the number of possible cut points) before the search process commences. If the cardinality is then further reduced in a *coarse* setting, still prior to analysis, the result is a reduction that could have been established by a more coarse discretisation to begin with.

The final omission is the strategy combining *global*, *nominal*, *coarse* and *best*. While this theoretically would produce a valid combination, it is highly restrictive in the candidate space considered, and (probably for that reason) is not present in the literature. The combination *global*, *nominal* produces nominal attributes from numeric ones (that in itself already involve a considerable loss of information), and additionally only continuing with the best candidates per discretised attribute (something that is typically not done to normal nominal attributes either) does not seem like a reasonable and promising approach.

Finally, an extra strategy is added: the BestInterval algorithm introduced by Mampaey et al. (2012, 2015). In linear time, it creates bounded intervals that maximise the quality for the target by setting an upper and a lower bound simultaneously. It is given the systematic name 17-lxfb, but it is only considered in the classification target setting and deviates from the other strategies, so it is mentioned separately.

3.3 Search space exploration strategies

This work compares three different search space exploration strategies: traditional beam search, 'complete' search, and CBSS beam search.

In SD, the local patterns are typically searched for by means of a top-down traversal of the pattern space, up to a certain specified depth. Small descriptions, selecting

large subsets of de data, are *refined* by adding conditions on individual attributes to form more extensive descriptions, selecting a subset of the 'parent' subgroup, or seed. Various different modes of search have been proposed, from exhaustive to very heuristic (e.g. by sampling from the pattern space). Although different search method are considered, the majority of this work focusses on a heuristic method (beam search) that nicely balances exploration and exploitation of the search space.

Traditional beam search (Lowerre 1976) conducts a level-wise search, and at each level it maintains a ranked list (by quality) of subgroups considered so far. At the end of a level, only the top W (known as the *beam width*) subgroups are allowed to produce candidates for the next level by means of refinement.

CBSS beam search is similar, but instead of maintaining a list of W candidates, it initially collects many more solutions per level. At the end of each search level, and also at the end of the entire search process, a subset of size W of the solutions in the temporary collection is selected which should encourage diversity of the candidate or result set. (The work considers 'attaining diversity […] equivalent to removing redundancy.')

Conversely, 'complete' search does not set a limit on the number of candidates, and thus is not heuristic in this sense. Here, the term *complete* is used, instead of the more common term *exhaustive*, since the exhaustive alternative to beam search is often combined with heuristic numeric strategies, that include (*local*) discretisation or the selection method *best*. Note that this particularly holds for a setting that is often referred to as exhaustive in the literature, that is, global discretisation followed by a complete traversal of the resulting, much reduced, pattern space.

Section 5.6 of the experiments compares results of the traditional and CBSS beam search, both in terms of quality and diversity. Section 5.5 evaluates to what extent result quality is influenced by the traditional beam search heuristic, compared to the complete setting.

4 Related work

The strategies presented in this work relate to an extensive range of topics. Therefore, only a selection of relevant work is discussed.

First, many papers make a comparison between SD algorithms. This is done in overview papers like (Atzmüller 2015; Herrera et al. 2011), as well as in papers that introduce new algorithms (Atzmüller and Lemmerich 2009; Atzmüller and Puppe 2006; Boley et al. 2017; Grosskreutz and Rüping 2009; Grosskreutz et al. 2008; Klösgen 1999; van Leeuwen and Knobbe 2012; Lemmerich et al. 2016; Mampaey et al. 2015; Meeng et al. 2014; Wrobel 1997). However, these papers only include a subset of the strategies presented in this work, and then only a very specific implementation of this limited set. The exclusive aim of this work is to provide a systematic and comprehensive experimental evaluation and comparison of all presented strategies, and this sets it apart from earlier work.

Historically, algorithms that were unable to deal with numeric description attributes resorted to a *nominal* strategy, usually in combination with *global* discretisation. Algorithms based on the FP-growth method (Han et al. 2000), like (Atzmüller and

Lemmerich 2009; Atzmüller and Puppe 2006; Grosskreutz et al. 2008; Lemmerich et al. 2016), are examples of this. The use of this combination has limitations, but facilitates (the design of) fast and efficient SD algorithms. Another class of such algorithms uses optimistic estimates (Wrobel 1997), and requires neither option. However, other drawbacks exist. For example, Boley et al. (2017) requires ordinal targets, and, like Lemmerich et al. (2016), found that computing optimistic estimates can be more costly than a much simpler approach, whereas Grosskreutz and Rüping (2009) requires depth-first search, a binary target, a huge data structure, and two separate mining runs to create nominal and numeric descriptions.

The *local* variations of *nominal* strategies presented in this work appear to be novel, at least in the context of SD. These are a logical consequence of completing the matrix of SD strategies in Table 1.

Strategy 17-lxfb results from the work of Mampaey et al. (2012, 2015). It exploits properties of convex and additive quality measures in order to compute the bounded interval that maximises the quality for the target in linear time. Conceptually, it could be seen as a strategy that uses a *Cartesian* alternative for dimension *interval type*. It considers all ordered interval pairs based on the cut points in a domain, like Grosskreutz and Rüping (2009), which combines *global*, *Cartesian*, *fine*, and *all*. The Cartesian option is not included in Table 1, as many of the additional possible combinations have never been considered in literature.

Even though Grosskreutz and Rüping (2009) is not considered here, a number of its observations are relevant in the current context. First, it explains why entropy-based discretisation, with either overlapping or non-overlapping intervals, typically leads to suboptimal results. Moreover, it found that for entropy-based discretisation, the *local* option never resulted in an optimal result where the *global* discretisation did not, corroborating the observations of Dougherty et al. (1995) and Frank and Witten (1999). In fact, overall, equal-frequency discretisation outperformed entropy-based discretisation. The experiments below use only equal-frequency discretisation, but evaluate it in both *global* and *local* discretisation contexts.

Section Overlap in the Introduction listed some literature concerning redundancy reduction, and indicated that a Diverse Subgroup Set Discovery (DSSD) variant, the CBSS covering approach (van Leeuwen and Knobbe 2011, 2012), is included in the experiments. With (Diverse) Subgroup *Set* Discovery, one moves away from Subgroup Discovery and its local nature, as subgroups are no longer 'judged purely on their own merit' (Duivesteijn 2013, p. 2), but 'should always be judged also on their joint merit' (van Leeuwen and Knobbe 2012, p. 219). Still, redundancy reduction has become somewhat of a staple in pattern mining, such that the analysis below is justified. The experiments will gauge whether a pure SD approach, using a traditional beam search and the strategies listed in Table 1, can compete with this technique in terms of redundancy and quality (Sect. 5.6), and run time (Sect. 5.7).

Aspects of DSSD are used in the work of Bosc et al. (2018), which is an original approach to SD, that poses very little restrictions on the numeric description attributes. The technique uses Monte Carlo Tree Search to sample subgroups from the search lattice, and is even compared to the Cortana tool used for the experiments in the current work. Some aspects of the analysis presented below could be investigated for this method also, but not all dimensions are relevant in a sampling context. Additionally,

the method only addresses classification, such that a full analysis would not be possible. For the same reason, Lavrač and Gamberger (2004) is not considered.

An elegant method that tries to bridge the gap between heuristic and exhaustive search is RefineAndMine (Belfodil et al. 2018). This method is an anytime algorithm that, given enough time, enumerates the pattern space exhaustively. It can be interrupted at any time, while offering guarantees about how close the intermediate result is to the (theoretic) optimum. Unfortunately, its search and numeric strategy, and interpretation of search depth, do not fit the framework of this work, and, as the method is strictly confined to binary targets, it is not considered here.

5 Experiments

The experiments described below analyse the benefits and drawbacks of the strategies listed in Table 1. Before the individual experiments are discussed, an overview is presented of the experimental conditions, parameters, and the datasets that feature in the subsequent sections. The 17,020 experiments[2] were all performed using the SD tool Cortana (Meeng and Knobbe 2011), using the SDMM algorithm.

The primary quality measures considered, motivated by their popularity in the literature, are *WRAcc* (Lavrač et al. 1999) for nominal targets, and |*z-score*| (Pieters et al. 2010) for numeric targets. Additionally, two alternative measures for each target type are considered, for a total of six measures. These are *lift* (Brin et al. 1997) and *binomial* (Klösgen 1992) for nominal targets (tested in a 'target value versus rest' setting), and |*deviation*| and |*t-statistic*| (Pieters et al. 2010) for numeric targets. Table 3 provides definitions for these measures. Note that for the numeric measures, absolute versions of the quality measures are used, simply because the interest is in both the high and the low deviations of the target distribution.

The three quality measures per type mainly differ in how they treat subgroup size (this is deliberate). On the low end, both *lift* and |*deviation*| do not have a weighting factor, and thus somewhat favour smaller subgroups (where larger deviations are more easily observed). On the high end, *WRAcc* and |*z-score*| favour larger subgroups, and *binomial* and |*t-statistic*| fall in between. While |*t-statistic*| and |*z-score*| use the same subgroup-size scaling factor $\sqrt{n}$, the former divides by the standard deviation of the subgroup σ_s, instead of that of the dataset ($\sigma_{\mathcal{D}}$). As it is easier to achieve a smaller dispersion using small subgroups, this measure generally favours such subgroups.

Some of the detailed analyses focus mainly on *WRAcc* and |*z-score*|, as these are good representatives for the two target types. Still, Sect. 5.3 combines the findings for all measures, and Sect. 5.4 offers a direct comparison of them.

Years of observation showed that the type of SD algorithm employed here rarely produces better results beyond the first three levels, but, to be safe, experiments are performed using search depth settings between 1 and 4. A cursory analysis of average top-10 scores for different datasets and strategies is presented in Fig. 1, which confirms this observation. Additionally, one could argue that too complex subgroup descriptions

[2] Results and methods for replication are found at: http://datamining.liacs.nl/for-real.zip.

Table 3 Quality measures used in subsequent experiments

Target type	Measure	Definition
Classification	*lift*	$(TP/n)/(T/N)$
	binomial	$\sqrt{n/N} \cdot (TP/n - T/N)$
	WRAcc	$TP/N - (T/N \cdot n/N)$
Regression	|*deviation*|	$\lvert\mu_s - \mu_{\mathcal{D}}\rvert$
	|*t-statistic*|	$\lvert\left(\sqrt{n} \cdot (\mu_s - \mu_{\mathcal{D}})\right)/\sigma_s\rvert$
	|*z-score*|	$\lvert\left(\sqrt{n} \cdot (\mu_s - \mu_{\mathcal{D}})\right)/\sigma_{\mathcal{D}}\rvert$

For the classification target, T is the number of occurrences of the designated target value in the whole dataset, and TP indicates how many of those are covered by the subgroup. For the regression target, μ_s and $\mu_{\mathcal{D}}$ refer to the target mean of the subgroup and of the dataset, respectively. The standard deviation $\sigma_{\mathcal{D}}$ is computed using the sum of squared deviations divided by N, for σ_s the division is by $n - 1$

are in disagreement with the easy-to-interpret, exploratory, and descriptive nature of the paradigm.

Respectively, a minimum and maximum subgroup size of $0.1N$ and $0.9N$ is enforced for all subgroups, to avoid overly small subgroups attaining unrealistically high scores. Beam search is performed using a beam of size 100. Although the SD process can be stopped at any time, all experiments are run until completion (the search space is exhausted). The parallelisation option of Cortana is not used, so all experiments were performed using a single cpu-thread.

Tables 4 and 5, respectively, list the datasets used in the experiments for the classification and regression setting. These datasets are taken from the UCI repository (Dua and Graff 2017), and the collection is chosen such that it gives a good mix with respect to the various statistics. It represents a range of sizes (N), number of numeric description attributes (|numeric|), (positive) target share (for classification datasets), and target cardinality (C) (for regression datasets). The *adult* and *pima-indians* datasets are customarily used with a classification target, but here they are also used in a regression setting, using the *age* attribute as numeric target.

5.1 Best number of bins

This section is dedicated to the 'number of bins' parameter B. This parameter controls the number of cut points that is eventually used by the SD algorithm. Setting this parameter such that results are optimal is a non-trivial task, as it is not immediately clear what the effect of this parameter is within the context of the various strategies. Furthermore, the possibility that effects differ amongst target type settings, and datasets, further hinders a straightforward selection of the parameter value. Therefore, this section presents the results of experiments performed to obtain insights into the intrinsic complexities stemming from these compound effects.

The 9828 experiments concern all strategies that use the *coarse* alternative for candidate generation, and the two strategies that combine *fine* with *global* and *binaries*. As described in Sect. 3.2, the latter can (should) be used in a *coarse* setting.

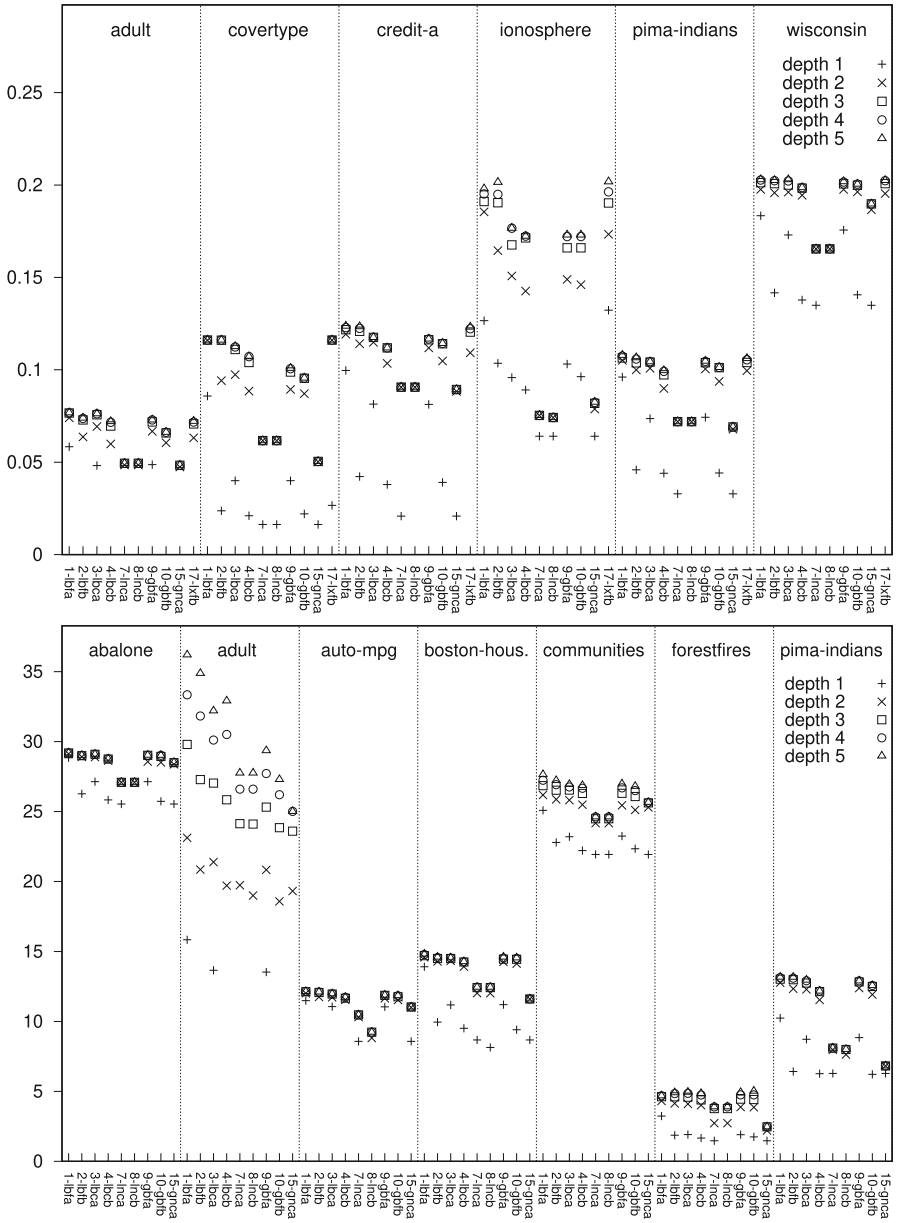

Fig. 1 Plots showing the development of average subgroup quality (*WRAcc* and |*z-score*|) for the 10 best subgroups, for different datasets and strategies. Only the *adult* dataset (regression), show a noticeable increase beyond depth 3. For ionosphere (classification) there is a small increase, for some strategies

For a single strategy, for each dataset and search depth, result sets of experiments using different parameter settings for B are collected. For each result set, the average score for the top-k subgroups is computed, and by ranking these average scores it is

Table 4 Datasets with a classification target. Listed are the dataset and target name, target value, positive target share (*T*/*N*, proportion of records having the target value), dataset size *N*, and |numeric|, the number of numeric description attributes

Dataset	Target	Target value	Positive (%)	*N*	|numeric|
adult	Class	≥ 50 K	23.93	48,842	6
covertype	Cover_Type	Lodgepole pine	48.76	581,012	10
credit-a	class	+	44.49	690	6
ionosphere	class	g	64.10	35	34
pima-indians	class	tested_positive	34.90	768	8
wisconsin	Class	1	34.48	699	9

Table 5 Datasets with a regression target

Dataset	Target	Target μ	*C*	*N*	|numeric|
abalone	Class_Rings	9.934	28	4177	7
adult[a]	Age	38.644	74	48,842	5
auto-mpg	class	23.515	129	398	4
boston-housing	MEDV	22.533	229	506	12
communities	ViolentCrimesPerPop	0.238	98	1994	126
forestfires	area	12.847	251	517	10
pima-indians[a]	age	33.241	52	768	7

Listed are the dataset and target name, target average μ, target cardinality *C*, dataset size *N*, and |numeric|, the number of numeric description attributes (so excluding the target column)
[a] Same datasets as in the classification setting, but using a different (target) setting

determined what value of B, ranging from 2 to 10, results in the highest average score. This process is performed separately for each quality measure.

Table 6 presents a summary of the results, and lists, for each strategy and quality measure, the optimal number of bins. These are also the numbers used in the subsequent experiments that compare different strategies. The various subtables in Table 12 in Appendix A provide more detail, and list results for depths 1, 2, 3, and 4, and a top-k of 1. Table 6 lists the overall $\mu(B)$ values, rounded to the nearest integer.

Experiments involving *binaries* strategies often led to multiple settings of B yielding the highest score. In part, this results from the nature of the algorithm. That is, the complete set of cut points obtained when creating B half-bounded intervals will occur in the set of cut points obtained when creating $2B$ half-bounded intervals, or more generally, any integer multiple of B. In such cases, only the lowest value of B is reported. Furthermore, within each table, the value for B at depth 1 is equal for the *all* and *best* alternative of an otherwise similar strategy, this is true by design.

The results show that there is no universal rule that guarantees a good number of bins. Not only does the best number differ per strategy, for a single strategy it can differ over quality measures, even for the same target type. Nonetheless, a number of general observations that can serve as guideline are listed and discussed below.

Table 6 The number of bins used in subsequent experiments, for each strategy and quality measure

Target type	Measure	3-lbca	4-lbcb	7-lnca	8-lncb	9-gbfa	10-gbfb	15-gnca
Classification	*lift*	6	6	4	4	6	5	5
	binomial	7	8	4	4	8	8	3
	WRAcc	7	6	2	2	7	7	2
Regression	|*deviation*|	7	7	6	6	9	9	6
	|*t-statistic*|	6	6	5	5	7	6	4
	|*z-score*|	7	8	4	4	7	7	4

These numbers are based on $\mu(B)$ in Table 12

The best number of bins for …

1. *binaries* is higher than for *nominal*, irrespective of target type, *nominal* is really low for classification targets, when used with *WRAcc*,
2. *binaries* greatly decreases over depths for *lift* and \|*deviation*\|, is stable for *binomial* and \|*t-statistic*\|, and increases somewhat for *WRAcc* and \|*z-score*\|, *nominal* decreases over depths, regardless of the target type and quality measure,
3. most strategies vary greatly over datasets, irrespective of target type.

A first general conclusion concerns the clear difference between *binaries* and *nominal* strategies. Consistently, the former lists higher numbers. This is not surprising, as for the two strategies the effects of higher values of B are quite distinct, although the impact differs per target type.

For *nominal*, when B increases, the size of the subgroups decreases, something that is especially problematic in the classification target setting, as the subgroups might no longer be able to cover all positives. In terms of ROC-space analysis (Fawcett 2006), these subgroups fall in the lower left-hand corner, such that the upper left-hand corner can never be reached by further refinement. In a regression setting, small subgroups are less problematic, as they can still cover numeric extremes.

For *binaries*, it is tempting to think that a higher B would also result in smaller subgroups. But, remember that for *binaries*, $B-1$ overlapping conjuncts are created, covering both small and large subsets of the data. And especially in conjunctions at greater search depths, including larger conjuncts might be more useful than having only smaller ones, as combinations of the latter often become too small to meet the minimum subgroup size constraint, and allow little possibilities for optimisation. It also explains why all strategies generally list a higher number of bins in the regression target setting.

Noteworthy also is the behaviour of *lift* and \|*deviation*\|. At greater depths, these list substantially lower number of bins. This prevents the candidates from becoming too small, through refinement, before reaching this depth. So, this compensates for the fact that these measures favour small subgroups.

Obviously, the last item is the most troubling. Some general trends are observed, but the key problem of choosing a good setting of B for all situations remains illusory, and the dataset characteristics listed in Tables 4 and 5 provide no guidance here.

Conclusion There is no universal rule that guarantees a good number of bins, but nominal strategies prefer a lower number than *binaries* strategies.

5.2 Comparing subgroup discovery strategies

The experimental sections further down (from Sects. 5.2.3–5.2.7) each focuses on a different dimension, but all follow a similar setup. First, it lists the strategies that are compared. Then, separately for each target type, results are discussed. Finally, a conclusion closes off each section, stating which option should be preferred.

Results are presented in two different forms, tables with qualities of the best (top-1) subgroup and tables with Mann-Whitney U-scores (Mann and Whitney 1947) for the top-10 subgroups. Each table lists the results for all strategies in Table 1, but 17-lxfb is only included for classification targets. Individual sections then contrast different pairs of strategies, depending on the dimension being discussed. For strategies that involve a B parameter, a superscript behind the name indicates what setting of B was used to produce the result. Most tables can be found in Appendix A.

5.2.1 Performance by best subgroup

The analyses of the different strategies start by considering the best subgroup found. Table 7 presents a summary for all quality measures, whereas Tables 13 and 14, provided in Appendix A, offer the detailed scores of the different strategies on the various datasets for *WRAcc* and |*z-score*|, respectively. Per depth and dataset, strategies are ranked based on their quality score, and given are both the average rank over all datasets per depth, and the average of these averages ($\mu(r_j, (1, 2, 3, 4))$). The tables also list Friedman F values, which are relevant for the critical difference diagrams in Fig. 2. These figures plot the different strategies on a horizontal scale representing their average rank as computed from the best subgroup. Low numbers (in other words good ranks) indicate that on average, a strategy performs well.

The figures furthermore provide information about the significance of differences in average rank, by means of critical difference (CD) indicator bars. If two strategies in any diagram are separated by less than the length of the CD bar at the top, they cannot be said to differ significantly (significance level $\alpha = 0.05$). Black bars across the CD plot help making this call for different pairs of strategies. The procedure for computing the critical distance is outlined by Demšar (2006), and only its relevant details are provided here. The Friedman critical value CV_f equals 2.096 for the classification experiments, and 2.138 for regression. The value F_F in Table 7 should be above the CV_f for the strategies not to be equal. If this is the case, post-hoc Nemenyi tests are permitted and the critical distance now becomes $\mathrm{CD} = 5.531$ for classification, and $\mathrm{CD} = 4.541$ for regression, as indicated in the respective figures.

The first observation is that the general order over all depths does not differ much. In the classification setting, the extensive strategies 1-lbfa and 2-lbfb, and the special strategy 17-lxfb, always rank best. Then comes a group of *binaries* strategies, first those combined with *coarse* and *all*, than those with *coarse* and *best*. The *nominal* strategies always rank last.

Table 7 Per depth and dataset, strategies are ranked based on their quality score

Measure	Depth	F_F	1-lbfa	2-lbfb	3-lbca	4-lbcb	7-lnca	8-lncb	9-gbfa	10-gbfb	15-gnca	17-lxfb
lift	1	2.639	3.75	3.75	6.25	6.25	6.58	6.58	6.17	7.83	5.92	1.92
	2	*1.157*	4.08	4.50	3.42	5.83	6.17	6.17	6.00	6.75	7.58	4.50
	3	*1.931*	4.00	4.17	3.25	4.58	7.00	7.00	5.00	5.67	8.33	6.00
	4	*2.073*	4.17	4.33	3.25	4.33	7.17	7.17	4.67	5.50	8.33	6.08
	$\mu(r_j, (1, 2, 3, 4))$		4.00	4.19	4.04	5.25	6.73	6.73	5.46	6.44	7.54	4.63
binomial	1	*1.738*	3.92	3.92	6.33	6.25	5.75	5.75	6.42	6.42	7.67	2.58
	2	16.214	1.75	3.08	4.67	4.92	8.83	8.83	5.08	5.67	9.17	3.00
	3	27.108	1.75	2.83	3.58	5.42	8.83	8.83	5.75	6.25	9.17	2.58
	4	21.975	1.75	2.83	3.92	5.92	8.83	8.83	5.25	5.75	9.17	2.75
	$\mu(r_j, (1, 2, 3, 4))$		2.29	3.17	4.63	5.63	8.06	8.06	5.63	6.02	8.79	2.73
WRAcc	1	2.914	3.67	3.67	5.75	7.17	7.58	7.58	4.42	4.42	7.58	3.17
	2	14.840	1.92	3.50	4.25	5.58	9.08	9.25	4.25	5.17	8.67	3.33
	3	17.281	2.00	2.92	4.67	5.25	9.08	9.25	4.67	5.58	8.67	2.92
	4	18.553	2.17	3.00	4.17	5.42	9.08	9.25	4.92	5.75	8.67	2.58
	$\mu(r_j, (1, 2, 3, 4))$		2.44	3.27	4.71	5.85	8.71	8.83	4.56	5.23	8.40	3.00

Table 7 continued

Measure	Depth	F_F	1-lbfa	2-lbfb	3-lbca	4-lbcb	7-lnca	8-lncb	9-gbfa	10-gbfb	15-gnca	17-lxfb
|*deviation*|	1	2.741	2.36	2.36	5.36	5.36	5.93	5.93	5.64	5.64	6.43	
	2	7.067	1.93	3.50	3.43	4.36	5.79	6.43	5.14	5.57	8.86	
	3	11.909	1.29	3.86	2.71	4.57	6.07	6.64	5.21	5.79	8.86	
	4	10.921	1.64	4.07	2.43	4.57	6.21	6.79	4.93	5.50	8.86	
	$\mu(r_j, (1, 2, 3, 4))$		1.80	3.45	3.48	4.71	6.00	6.45	5.23	5.63	8.25	
|*t-statistic*|	1	3.321	2.36	2.36	5.29	5.29	6.79	6.79	4.93	5.36	5.86	
	2	2.207	3.43	3.50	3.86	4.21	5.93	6.36	4.79	5.36	7.57	
	3	3.597	2.71	3.43	3.00	4.79	6.50	6.93	5.07	5.43	7.14	
	4	3.265	3.43	2.93	3.14	4.14	6.36	6.79	5.50	5.57	7.14	
	$\mu(r_j, (1, 2, 3, 4))$		2.98	3.05	3.82	4.61	6.39	6.71	5.07	5.43	6.93	
|*z-score*|	1	3.484	2.43	2.43	6.07	3.71	6.36	6.36	5.86	5.86	5.93	
	2	5.503	1.93	3.21	3.36	5.64	7.29	7.29	4.79	4.93	6.57	
	3	13.820	1.57	2.57	3.14	4.29	7.71	7.71	5.29	5.43	7.29	
	4	13.039	2.00	2.43	3.29	4.57	7.86	7.86	4.93	4.64	7.43	
	$\mu(r_j, (1, 2, 3, 4))$		1.98	2.66	3.96	4.55	7.30	7.30	5.21	5.21	6.80	

For each strategy, the average rank over all datasets, for depths 1, 2, 3, and 4, is given here, as is the average of these averages ($\mu(r_j, (1, 2, 3, 4))$). For *WRAcc* and |*z-score*| more detailed information can be found in Tables 13 and 14, respectively. The Friedman F values are listed in those tables, per depth, on the $\mu(r_j, \text{depth})(F_F)$ lines, and are based on the (average) ranks in column r. Friedman F scores in italic are below the critical value, using $\alpha = 0.05$. As such, the post-hoc Nemenyi test is not permitted, and no valid critical difference diagrams can be created. The sections discussing experimental results for the various dimensions, Sect. 5.2.3 and onwards, and Table 9 use the average rank information from this table

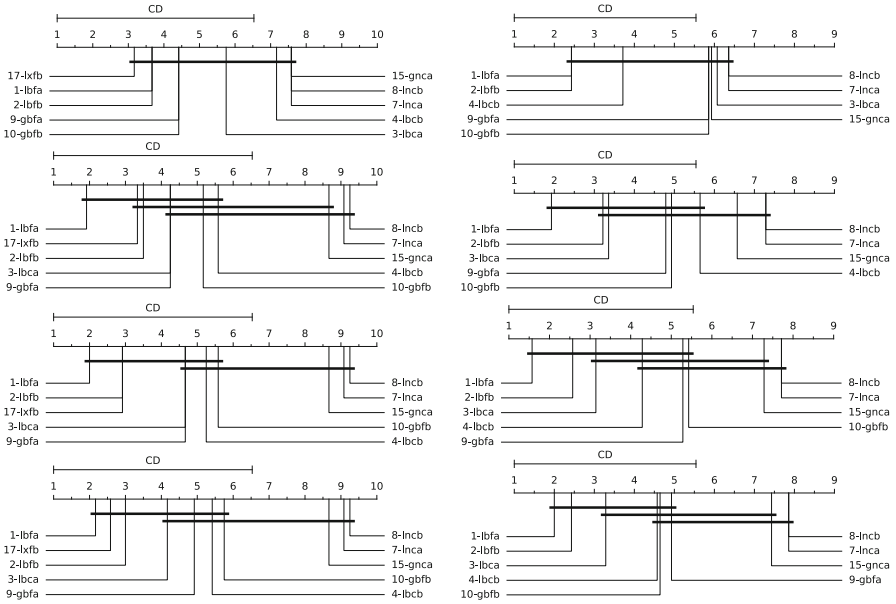

Fig. 2 Critical difference diagrams for the classification targets (left), using *WRAcc*, and regression targets (right), using $|z$-score$|$, for depth 1, 2, 3, and 4 (top to bottom). Within a plot, the strategies are ordered counter-clockwise, from best to worst. Strategies connected by a thick horizontal line, differ by less than the critical difference (CD), and cannot be said to differ significantly ($\alpha = 0.05$). The general order over all depths does not differ much, and is similar for the two target settings. In the regression setting, strategy 17-lxfb is not used, and the average rank of 3-lbca is better than that of the related *binaries* strategies.

Generally, the order in the regression setting is the same. But here the *local* variants of the *binaries* strategies perform better than the *global* variants. Also, 3-lbca performs better than the rest of this group.

Conclusion The *binaries* strategies rank before the *nominal* ones, and the more extensive variants come first, regardless of target type.

5.2.2 Performance by top-10 ranking

The analysis described above considers only the best subgroup, whereas SD is generally expected to produce a ranked list of alternative subgroups. To give an insight into the distribution of scores in the involved rankings, the Mann-Whitney U test is used, and two tables listing these U-scores are presented (Tables 15, 16).

U-scores compare two distributions to determine if one is stochastically greater than the other, in which case the probability of an observation from the first distribution exceeding that of the second is different from the reverse probability (of an observation from the second exceeding that of the first). In the extreme case of $U = 0$, all values from one distribution come before all values of the other distribution. Such an insight can not be obtained by comparing the means, or medians, of two rankings.

When comparing two strategies, all scores of their result sets $\mathcal{F}_1$ and $\mathcal{F}_2$, of size F_1 and F_2 respectively, are put together, sorted, and assigned combined ranks. Then, U_1 is computed for set $\mathcal{F}_1$ as follows:

$$U_1 = \Sigma_1 - \frac{F_1 (F_1 + 1)}{2}, \tag{1}$$

where Σ_1 is the sum of ranks of result set $\mathcal{F}_1$. The same is done for $\mathcal{F}_2$, and the smaller of U_1 and U_2 is used as U for significance testing.

The tables in Appendix A compare the top-10 rankings of different strategies, so $F_1 = F_2 = 10$. For a one-sided test, and a significance level of 5%, the critical value is 27. So, when $U \leq 27$ the null hypothesis "the distributions are equal" is rejected.

However, the tables do not list U, but U_1, as this shows which of the two strategies is better. Using the fact that $U_2 = F_1 \cdot F_2 - U_1$, scores below 50 indicate that the first (left) strategy is better, scores above 50 mean the second (right) strategy is better.

The final columns of the table, under '≤ 27 / ≥ 73 / valid', indicate, per depth, how often the U-score is significant for the left and right strategy, respectively, and for how many datasets a top-10 is available. When 'valid' is not equal to the number of datasets, it indicates that, in some experimental settings, not enough subgroups are found to create a top-10 ranking.

The columns under 'Wins' use a number of symbols to summarise which of the strategies is better over all tested datasets, for a given depth. Triangles point in the direction of the strategy that has a better ranking more often than the other, $=$ means there is no 'winner'. Respectively, ◀, ◁, and $<$, indicate that the left strategy is: better for all datasets, and all results are significant; better for all datasets, but not all results are significant; better overall, but not better for all datasets. Right-pointing triangles have equivalent meanings for the right strategy.

Unlike the number of (valid) results, the number of symbols is equal for all strategies, which allows for a straightforward comparison. In the classification setting, each strategy is compared to nine others (Table 15). There are eight such comparisons in the regression setting (Table 16). When interpreting these tables, readers are suggested to favour the triangles over individual U-scores.

5.2.3 Interval type: binaries versus nominal

The next four sections discuss the four different dimensions separately, and determine the optimal choice for each dimension. Sometimes, there is a clear winner (as is the case in this first section), but the best choice might depend on the context, that is, on the setting of the three remaining dimensions. For the two strategies within a context, only the setting for the dimension under investigation differs, settings for the other dimensions are the same. For that reason, these sections list the appropriate contexts and discuss context-dependent choices where relevant. The four dimensions are discussed not in the order that they were introduced, but in order of complexity of the analysis. Therefore, the discussion starts with the simplest, the dimension *interval type*, with the two possible values *binaries* and *nominal*.

The choice between these two settings is relevant in the following contexts:

- *local* with *coarse* and *all*. In this context, 3-lbca is pitted against 7-lnca.
- *local* with *coarse* and *best*. This means comparing 4-lbcb with 8-lncb.
- *global* with *coarse* and *all*. Pitting 9-gbfa against 15-gnca, where the former is transformed into a *coarse* strategy by using a low number of values, as described in Sect. 3.2.

Classification Target As Tables 13 and 15 demonstrate, the *binaries* setting consistently outperforms the *nominal* setting in all relevant contexts. The average ranks are always better for *binaries*, and the difference between these ranks gets quite large at greater depths. Furthermore, *binaries* lists a better quality for 65 out of 72 comparisons. The 2 times *nominal* lists a better score, and the 5 ties, all occur at depth 1. Also note that for the *ionosphere* dataset, *binaries* qualities are sometimes twice as high as those of *nominal*. In terms of Mann-Whitney U-scores, *binaries* lists a better U-score for each of the 66 results, of which 63 are significant. For the *lift* and *binomial* quality measures, observations are similar.

Regression Target Again, *binaries* is the clear preferred choice in all experiments (Tables 14 and 16). Here also, the average ranks are always better for *binaries*, and this option lists a better quality for 69 out of 84 comparisons, *nominal* achieves a better score 14 times. As in the classification setting, the difference of the average ranks gets quite large at greater depths. Between strategies, the $|z\text{-score}|$ generally differs by some 5–10%, but some are up to 40 percent. With respect to the top-10 rankings, *binaries* is better in all of the 77 experiment, with 63 results being significant. For $|deviation|$ and $|t\text{-}statistic|$ these statistics are almost identical.

Conclusion Always prefer *binaries*. It clearly outperforms *nominal* in all contexts for both target types.

5.2.4 Granularity: fine versus coarse

This section contrasts options *fine* and *coarse* of dimension *granularity*. The relevant contexts are:

- *local* with *binaries* and *all*. This concerns strategies 1-lbfa and 3-lbca.
- *local* with *binaries* and *best*. The relevant strategies are 2-lbfb and 4-lbcb.

Classification Target Without exception, *fine* is the better option. All of its average ranks are better, and of the 48 comparisons, *fine* wins 38, *coarse* wins 4, and there are 6 ties. The wins of *coarse* are interesting, as one might expect that the search space of *coarse* is a subset of that of *fine*. However, this is true only for complete search, not for a heuristic (beam) search. Table 13 shows that the wins occur at depths greater than 1, indicating that the beam of the *coarse* variant of the strategy contained a candidate that was both not included in the beam of the *fine* counterpart, and proved to be a better seed for refinement than any of the candidates the latter beam contained. Closer inspection of the scores shows that for the first context, the *coarse* scores are on average 97.45% of that of *fine*, and 16 out of 24 times the difference is smaller than 2%. For the second context, *coarse* scores are on average 96.85% of that of *fine*, and 13 times out of 24

the quality it is not worse by more than 2%, or even better. Concerning the top-10 rankings in Table 15, 22 out of 24, and 15 out of 23, results are significant for the first and second context, respectively, and all results are in favour of *fine*. The trend for *lift* and *binomial* is the same, though for the former there are more ties and a few wins for *coarse*.

Regression Target Again, without exception, *fine* is the better option, as can be observed in Tables 14 and 16. All average ranks are better, and of the 56 quality comparisons *fine* wins 44, *coarse* wins 3, and there are 9 ties. For the first and second context, respectively, the *coarse* scores are on average 98.65% and 98.43% of that of *fine*, and 41 times out of 56 the difference is smaller than 2%. For the top-10 rankings, 25 out of 28, and 20 out of 26, results are significant, for the two contexts respectively. Here, results are similar for |*deviation*| and |*t-statistic*|.

Conclusion Invariably, *fine* is better. Considering that *coarse* is a heuristic numeric strategy, this might not seem remarkable, although, as mentioned, in a beam setting *fine* is not guaranteed to perform better at greater search depths. Nevertheless, the quality of the top subgroups *coarse* produces is within a few percent of those of *fine*, and sometimes even better, though the latter can only happen in a beam search.

5.2.5 Selection method: all versus best

Next, options *all* and *best* of dimension *selection method* are compared. The relevant contexts are:

- *local* with *binaries* and *fine*. Comparing 1-lbfa with 2-lbfb.
- *local* with *binaries* and *coarse*. Selecting 3-lbca and 4-lbcb.
- *local* with *nominal* and *coarse*. Pitting 7-lnca against 8-lncb.
- *global* with *binaries* and *fine*. Using both 9-gbfa and 10-gbfb as *coarse*.

Classification Target Clearly, *all* is the better option. First, note that *all* and *best* strategies using the same number of bins would attain identical average ranks, and qualities, at depth 1. Still, in every context, *all* outperforms *best* with respect to the qualities in Table 13. At depth 2, 3, and 4, *all* always has a better average rank. With respect to the qualities, it is interesting to see that in the first, third, and fourth context, there are a large number of ties, 59 of 72 results, and *all* does not often win, 12 times. Whereas, in the second context, *all* wins 18 of the 24 comparisons. Of the 96 quality comparisons, only 5 differ by more than 5%, with a maximum of 8.1%. Note that for the first context, the quality for *best* is better than *all*, at depth 4, for the *ionosphere* dataset (this can be seen by the rank, due to rounding the quality scores appear to be the same). This is caused by the same beam effect as described above. Because the variant in the second context uses a different number of bins, the fact that qualities for the *ionosphere* dataset are better for *best* than for *all* at depth 3 can not be ascribed to the beam search per se.

Table 15 show that in the *binaries* contexts, *all* is better for 66 out of 69 results, of which 50 are significant. In the only *nominal* context, results for *all* and *best* are basically identical, and thus never significant. Again, overall trends are similar for *lift*

and *binomial*, but for the former half the quality scores tie for the second context, and *all* wins half the comparisons in the fourth.

Regression Target Overall, results for *all* are better for this target type. When considering the quality scores in Table 14, the high number of ties is notable. Here, the first two contexts should be considered separately from the latter two. For the first two, the average rank of *all* is better in all settings (above depth 1), and *all* wins about half of the comparisons. For the third context, all results, and thus ranks, are identical, for the fourth, just 4 results differ. Here, a beam effect can be observed for dataset *forestfires* at depth 3 for the first context, and at depth 4 for the first and third context. With respect to the 112 equivalent qualities, only 3 differ by more than 2%. Considering the results for the top-10 rankings in Table 16, all *binaries* contexts behave similar, resulting in wins for *all* in 76 of the 78 comparisons, with 61 significant results. For the only *nominal* context, 10 out of 23 results are wins for *all*, the remaining 13 are ties, and only 3 wins are significant. Results for |*deviation*| and |*t-statistic*| are similar for all contexts.

Conclusion Option *all* performs better than *best*, which is no surprise. The more interesting observations relate to the performance of *best*. Out of the 208 results for the two target types combined, the score for the top subgroup is within 1% of the *all* score, or better, 169 times, and only 8 differences of more than 5% were observed. This suggests that the very heuristic *best* selection method is a very capable alternative to *all* when considering result quality, often coming within 1% of the *all* result.

5.2.6 Discretisation timing: local versus global discretisation

This section compares options *local* and *global* of dimension *discretisation timing*. The contexts relevant to determine the best choice among these two alternatives are:

- *binaries* with *coarse* and *all*. The relevant strategies in this context are 3-lbca and 9-gbfa, where the latter is transformed into a *coarse* strategy by using a low number of values, as described in Sect. 3.2.
- *binaries* with *coarse* and *best*. This compares 4-lbcb with 10-gbfb, here, using few values for 10-gbfb.
- *nominal* with *coarse* and *all*. It pits 7-lnca against 15-gnca.

Of all experimental settings, results for *discretisation timing* are the least unequivocal, and the most dependent on the context, search depth, and quality measure. In part, this is due to an implementation detail. Both *global* and *local* use Algorithm 2. But, for a single attribute, *local* potentially yields the same bin boundaries in case of '$\leq$' and '$\geq$', whereas *global* creates discretised data once, always using '$\leq$'. For the example in Sect. 2.3, $\mathbf{a} = [x, y, z]$ and $B = 2$, this yields $[y, y, z]$, and results on depth 1 are not guaranteed to be identical. An implementation where *global* considers the original, undiscretised, data and selects static cut points once, separately for '$\leq$' and '$\geq$', would be identical. However, most emphasis should be on the results of greater depths, where the difference between the options is most prominent.

Classification Target The results in Tables 13 and 15 show very mixed and convoluted results. In the first two contexts, alternative *local* is better 19 times, there are 7 ties, and *global* wins 22 times. For the third context, there are 18 ties, and 6 wins for *global*, all for the *ionosphere* and *wisconsin* datasets at greater depths. The average ranks follow these patterns.

Considering the Mann-Whitney U-scores for the top-10 rankings, most results at depth 1 are close or identical, and never significant. At depth 2, 3, and 4, *local* outperforms its non-dynamic counterpart 15 out of 18 times (7 significant) in the first context. Of the 3 non-significant wins for *global*, 2 occur for the *wisconsin* dataset. For the second context, there are 7 wins for *local* (4 significant), *global* wins 11 times (4 significant). For the third context, there are 12 wins for *local* (3 significant), and the 6 times *global* wins, involve the aforementioned datasets again, but all results are now significant.

This time, results for *lift* and *binomial* are quite different. Here, *all* has a better average rank 20 out of 24 times, and a better quality for 83 out of 144 results. Qualities are equal 26 times, and better for *best* 35 times. Regarding U-scores, *lift* has a better result 42 times (34 significant), there are 12 ties, and 17 wins for *global* (2 significant). For *binomial*, these results are similar to those of *WRAcc*.

Regression Target In the first context, *local* is clearly better, with better average ranks for greater depths, and a better quality 19 out of 28 times. Table 14 shows there are 4 ties, and 5 times *global* is better. Also, of the 26 U-score comparisons, *local* wins 22 (15 significant), there is 1 tie, and *global* wins 5 times (1 significant). Concerning qualities for the second and third context, there are 14 wins for *local*, 2 ties, and 12 wins for *global*, and 7 wins for *local*, 6 ties, and 15 wins for *global*, respectively. Regarding U-scores in Table 16, there are 26 wins for *local* (13 significant), 1 tie, 26 wins for *global* (14 significant).

As with the classification setting, most statistics are skewed more in favour of *local*, when using |*deviation*| and |*t-statistic*|.

Conclusion With respect to classification targets, *local* should be preferred when considering all contexts, depths, and quality measures overall. Regarding the top result, there is not much difference between the options when used with *WRAcc*, but with *lift* and *binomial*, *local* clearly performs better. The *local* alternative also performs better when considering the top-10 rankings.

For regression targets, *local* performed clearly better at greater depth in the first *binaries* context, and should be the preferred choice. Although in the second and third context there is not much difference between the two alternatives when using *WRAcc*, preference shifts towards *local* when considering |*deviation*| and |*t-statistic*|.

A more general finding is that *global* might be better than *local* at depth 1, but that the latter is, and gets, better at greater search depths, proving that its flexibility is useful. Although, this result is not all that convincing in the classification setting, it is for the more complex regression setting. While Dougherty et al. (1995), Frank and Witten (1999) and Grosskreutz and Rüping (2009) found that for entropy-based histograms there was no big difference between *local* and *global*, the difference is relevant for the

equal-frequency binning employed here. Especially the *binaries* strategy 3-lbca greatly benefits from it.

5.2.7 Mampaey et al. (17-lxfb) versus all other strategies

So far, strategy 17-lxfb was left out of the analyses, but it is the exclusive focus of this section. Unlike the previous sections, this one does not revolve around contexts. Still, a separation along dimensions is instrumental when analysing the results. Most important is the differentiation between *binaries* and *nominal* strategies. The latter is treated as a single group, the former is sometimes divided into subgroups, when this provides additional insights.

The original implementation of 17-lxfb was extended to take into account the minimum and maximum subgroup size constraints. Without the former, the algorithm selects perfect (scoring) intervals, but the subgroups might be too small to be included in the result set. Also, the linear version of this algorithm was not used.[3] Instead, the quadratic number of intervals is evaluated, as in Grosskreutz and Rüping (2009). This does not influence the results, but does affect the run times.

Strategy 17-lxfb was included in the experiments because it produces 'optimal' results at depth 1 (of which, at least, one is also a global optimum). Table 13 confirms this, as there is not a single strategy that attains a better quality for any of the datasets. And although some of the other strategies are able to attain the same quality score, 17-lxfb has the best average rank.

A more interesting behaviour occurs at greater depths. First, consider the four strategies that combine *binaries* with *local*. For the two that combine with *fine*, 1-lbfa lists a better quality than 17-lxfb 8 times out of 18, there are 6 ties, and 17-lxfb wins 4 times, and for 2-lbfb there are 5 wins, 6 ties, and 7 loses. However, these strategies are better by margins of less than 1% for most results, except those involving the *adult* dataset. The average rank of 1-lbfa is now better than that of 17-lxfb, and the average rank of 2-lbfb is close to it. For 3-lbca, qualities are now within 2% for 15 out of 18 results. This includes the 4 times this strategies score better, but the average ranks are still always worse. For 4-lbcb, 15 out of 18 qualities are now within 2%, but the average ranks are still always worse. The 3 times this strategy is better occur for the *adult* dataset.

Of the strategies combining *binaries* with *global*, 9-gbfa has a better score 4 times (3 for *adult*, 1 for *pima-indians*), and of the 18 comparisons, 10 are better or within 2%. For 10-gbfb only 8, including the 1 better score for *pima-indians*, are within 2%, 12 are within 4%. The average ranks for these strategies are also always worse.

Interestingly, all binaries strategies score better than 17-lxfb on the *adult* dataset, some score also better on *ionosphere* or *pima-indians*. Based on the characteristics of these dataset, listed in Table 4, there is no common theme that binds these datasets. As such, no reason can be given for why other strategies surpass 17-lxfb for some datasets, and not others.

For the *nominal* strategies 7-lnca, 8-lncb, and 15-gnca, every average rank and quality is better for 17-lxfb. In fact, for the *covertype* and *ionosphere* dataset, which

[3] Code analysis revealed an exceptional case that would yield a suboptimal interval. None of the results presented in this work would have been affected, but the tool has wider use.

include high cardinality description attributes, these strategies are only able to attain scores that are more than 35%, or even 58%, lower than the 17-lxfb score.

With respect to the Mann-Whitney U results in Table 15, the distinction along dimensions is informative again. Collectively, there are 180 results for 17-lxfb, these include 136 wins (76%), of which 116 are significant, for an 85% significant-to-win ratio. Against strategies combining *binaries* with *local*, there are 80 results, 43 wins (54%), and 33 significant results (77%), indicating that these strategies compare more favourably to 17-lxfb than others. Against *binaries* with *global*, there are 40 results, 33 wins (83%), and 23 significant results (70%). These numbers are even skewed in favour of 17-lxfb, as 4-lbcb has 0 wins, and 10-gbfb has 1. Most strikingly though, there are 60 *nominal* results, all of which 17-lxfb wins significantly.

Conclusion The fact that no other strategy performs better than 17-lxfb at depth 1 is expected. However, for a number of datasets, the same quality is achieved by many other strategies, some of which could be considered light heuristics. How the results of various (broad groups of) strategies evolve over increasing depths is also noteworthy. Strategies involving *binaries* fare much better than those using a *nominal* approach. Nonetheless, only the two computationally most demanding strategies outperform or tie with 17-lxfb for 3 of the 6 datasets, and just one heuristic comes close. As such, this strategy should be the method of choice when seeking high quality subgroups in a classification setting. As mentioned, 17-lxfb was not applicable to regression datasets.

5.3 Ranking subgroup discovery strategies

Table 8 presents the final ranking of strategies, combining information in Tables 7, 13, 14, 15, and 16, and similar tables for *lift*, *binomial*, |*deviation*|, and |*t-statistic*| not presented here. This ranking should be considered nothing more than a convenient summary, and no statistical claims are made about this result. This is because an aggregation is performed over the same, not different, datasets in the quality tables, and the Mann-Whitney table is based only on pairwise tests, and does not correct for multiple hypothesis testing. For more on these issues, refer to Demšar (2006).

The ranking under μ_1 is based on the qualities in Tables 7, 13 and 14, that are aggregated in Table 9. Under 'Classification' and 'Regression' of this table, the strategies are assigned a rank based on the average of their average rank per depth (also listed at the average rank line $\mu(r_j, (1, 2, 3, 4))$ in the score tables). This shows how the strategies perform for each target type. To create an overall ranking, combining the results of the two target types, strategy 17-lxfb needs to be omitted from the analysis. For example, under 'Classification excl. 17-lxfb' the average ranks listed for *WRAcc* are based on an analysis of Table 13, where the scores for 17-lxfb are excluded. Note that the agreement between rankings for classification and regression is rather high, with a Spearman's rank correlation $\rho = 0.983$. Then, listed under 'Combined', the average over these overall average ranks is computed, and the strategies are assigned a rank based on this average, where a lower average is better.

The ranking under U_{10} is based on Tables 15 and 16, and aggregate Table 10. Results are based on the symbols in the 'Wins' columns, and count the number of

Table 8 The final ranking of strategies

Rank	Ranking based on	
	μ_1	U_{10}
1	1-lbfa	1-lbfa
2	2-lbfb	3-lbca
3	3-lbca	2-lbfb
4	4-lbcb	9-gbfa
5	9-gbfa	4-lbcb
6	10-gbfb	10-gbfb
7	7-lnca	7-lnca
8	8-lncb	8-lncb
9	15-gnca	15-gnca

These are based on μ_1 and U_{10}, and are computed in Tables 9 and 10, respectively. There are two permutations in the list, where a pair of strategies is swapped. So, the rankings strongly agree with each other ($\rho = 0.967$), and rank most of the strategies combining *local* and *binaries* before those combining *global* and *binaries*, and list all *nominal* strategies at the bottom

local often performs better than *global*,
binaries is superior to *nominal*,
fine triumphs over *coarse*,
all beats *best*

left-pointing triangles (<,◁,◀) when a strategy is on the left, and the number of right-pointing triangles (>,▷,▶) when a strategy is on the right, of strategy pairs in the Mann-Whitney U tables. The sum of these wins is computed, and strategies are ordered based on this sum, where a higher sum is better. Again, the agreement between classification and regression is rather high: $\rho = 0.967$.

Although the rankings are based on different analyses (top-1 versus top-10), they strongly agree with each other ($\rho = 0.967$). The *nominal* strategies always rank last, the *local binaries* strategies usually rank before the *global* strategies, and the *all* alternatives outperform their *best* counterpart.

The most remarkable finding would probably be the fact that the heuristic 3-lbca performs so well. Certainly, 2-lbfb is a heuristic also, but its computational complexity is much higher. Another non-trivial result is the scale at which the *nominal* strategies perform worse than *binaries* strategies. For both target types, *nominal* strategies rank at the bottom of the list. Generally, the table reaffirms some of the general trends that were observed before.

Conclusion The relative order and performance of strategies is very consistent over both all quality measures and when determined considering the best individual subgroup and ranking of the set of top subgroups.

5.4 Comparison of quality measures

The detailed analyses so far mainly revolved around the popular and well-established quality measures *WRAcc* and |*z-score*|. These measures are used because they address

Table 9 A ranking of strategies based on the average ranks in the top-1 score tables for each quality measure

Strategy	Classification incl. 17-lxfb				Classification excl. 17-lxfb				Regression				Combined	
	φ_l	φ_b	φ_w	r_C^{+17}	φ_l	φ_b	φ_w	r_C^{-17}	φ_d	φ_t	φ_z	r_R	μ_{C+R}^{-17}	Rank
1-lbfa	4.00	2.29	2.44	1	3.52	1.77	1.98	1	1.80	2.98	1.98	1	2.34	1
2-lbfb	4.19	3.17	3.27	3	3.81	2.56	2.69	2	3.45	3.05	2.66	2	3.04	2
3-lbca	4.04	4.63	4.71	4	3.65	3.96	3.96	3	3.48	3.82	3.96	3	3.81	3
4-lbcb	5.25	5.63	5.85	6	4.69	4.71	5.02	5	4.71	4.61	4.55	4	4.72	4
7-lnca	6.73	8.06	8.71	8	6.00	7.08	7.75	7	6.00	6.39	7.30	7	6.75	7
8-lncb	6.73	8.06	8.83	9	6.00	7.08	7.88	8	6.45	6.71	7.30	8	6.90	8
9-gbfa	5.46	5.63	4.56	5	4.92	4.88	3.88	4	5.23	5.07	5.21	5	4.86	5
10-gbfb	6.44	6.02	5.23	7	5.75	5.15	4.42	6	5.63	5.43	5.21	6	5.26	6
15-gnca	7.54	8.79	8.40	10	6.67	7.81	7.44	9	8.25	6.93	6.80	9	7.32	9
17-lxfb	4.63	2.73	3.00	2										

Listed under 'Classification' and 'Regression' are the average rank for each strategy, presented as $\mu(r_j, (1, 2, 3, 4))$ in the relevant score tables (like 7, 13 and 14). This shows how the strategies perform for each target type. To make the ranks between two target settings comparable, r_C^{-17} lists the ranks, computed excluding 17-lxfb. Under 'combined', the two rankings are averaged, leading to the final 'rank'. For reasons of presentation, the names of the quality measures *lift*, *binomial*, *WRAcc*, |*deviation*|, |*t-statistic*|, and |*z-score*| are replaced by symbols φ_l, φ_b, φ_w, φ_d, φ_t, and φ_z, respectively. For r_C^{-17} and r_R, $\rho = 0.983$

Table 10 A ranking based on the Mann-Whitney U tables for all strategies

Strategy	Classification						Regression					Combined	
	φ_l	φ_b	φ_w	Σ_C^{+17}	Σ_C^{-17}	r_C^{-17}	φ_d	φ_t	φ_z	Σ_R	r_R	Σ_{C+R}^{-17}	Rank
1-lbfa	34	35	35	104	95	1	32	32	32	96	1	191	1
2-lbfb	21	26	27	74	65	3	25	25	26	76	2	141	3
3-lbca	29	23	26	78	74	2	25	25	24	74	3	148	2
4-lbcb	17	13	12	42	41	5	16	16	14	46	5	87	5
7-lnca	9	5	6	20	20	7	7	8	7	22	7	42	7
8-lncb	6	0	3	9	9	9	3	3	2	8	8	17	8
9-gbfa	19	20	22	61	60	4	21	22	21	64	4	124	4
10-gbfb	9	12	14	35	35	6	13	12	14	39	6	74	6
15-gnca	5	6	0	11	11	8	2	1	3	6	9	17	9
17-lxfb	18	25	25	68									

The table lists the number of wins for each strategy (left-pointing triangles ($<,\lhd,\blacktriangleleft$) when a strategy is on the left of a strategy pair, right-pointing triangles otherwise). For classification targets, the counts in the columns under φ and Σ_C^{+17}, include comparisons to strategy 17-lxfb. To make the two target settings comparable, Σ_C^{-17} lists the counts excluding all 17-lxfb comparisons. Columns do not sum to 180 (144) in case of ties (=). The rankings for the classification and regression settings, r_C^{-17} and r_R, respectively, strongly agree with each other ($\rho = 0.967$). Here, strategy 8-lncb is ranked before 15-gnca, as it wins more of their direct comparisons

different target types, that entail different search characteristics. Further experiments were performed for additional measures per target type, as there are no guarantees that findings for these measures generalise to other types of quality measures.

Quality measures differ in the aspects of subgroups they favour, and the size of subgroups is often an important factor. As *WRAcc* and |*z-score*| favour rather large subgroups, four measures more prone to select smaller subgroups were evaluated.

Although the experiments demonstrate that the different runs vary a lot in their search and reported subgroups, the resulting ranking of numeric strategies is surprisingly similar across measures. This degree of correlation can already be gleaned from Tables 9 and 10, where the triplets of columns associated with the three measures per setting show a surprising consistency. Concerning the μ_1 and U_{10} rankings in the classification setting, the Spearman's rank correlations with *WRAcc* are $\rho = 0.921$ and $\rho = 0.875$ for *binomial*, and $\rho = 0.827$ and $\rho = 0.954$ for *lift*, respectively. For the regression setting, the Spearman's rank correlations with |*z-score*| are $\rho = 0.946$ and $\rho = 0.975$ for |*t-statistic*|, and $\rho = 0.946$ and $\rho = 0.975$ for |*deviation*|. Here, |*deviation*| and |*t-statistic*| have a correlation of $\rho = 1$ for both the μ_1 and U_{10} rankings. Concerning the rankings, the different settings benefit in very similar ways from the various strategies considered, despite searching for alternative types of subgroups.

Concerning the search, or pattern, space, a different picture emerges, that is highly related to the size aspect of the measures. Crudely, the maximum number of candidates at each search level in a beam search is given by the beam width times the number of candidates at depth 1. For the same depth, the amount of candidates produced by the measures that favour large subgroups is closest to this number, whereas for the

'unweighed' *lift* and |*deviation*| measures it deviates a lot, *binomial* and |*t-statistic*| are in between. For a single measure, with increasing depths, the amount of candidates produced by the measures that favour large subgroups remains relatively high and closest to the maximum, whereas for unweighed measures the amount rapidly decreases. These observations hold over the various strategies and datasets. The interpretation is that for measures that favour smaller subgroups the candidates in the beam are generally smaller, and such subgroups generate fewer valid refinements.

5.5 Complete search versus beam search

The experiments use a beam search to efficiently traverse the potentially large search spaces associated with numeric attributes. When considering the findings presented so far, one might object that they only hold in the context of such heuristic search. In fact, it might be that some poorly performing strategies, for example those using a *nominal* setting, are better suited for complete search space exploration. In order to test any potential differences under complete search strategies, the experiments were executed also with a complete version of the SD algorithm. Results are compared for depth 2 and 3 for *WRAcc* and |*z-score*|. Depth 1 results are always the same.

Interestingly, there is hardly any difference between the results of the two search strategies. The best number of bins for each strategy remains the same, as does the ranking of strategies in Table 8. The few differences that do occur are small, such that they do not change the overall findings. A general observation is that, basically, only strategy 1-lbfa and 3-lbca benefit from complete search, though marginally.

In the classification setting, only 7 top-1 results are different, on average by less than 0.7%, and all differ less than 1.7%. For the regression setting, there were 21 differences, that on average differed by 1.3%, with a maximum of 4.9%. The latter occurs along with 6 other differences above 1% for the *forestfires* dataset, without these the average would be 0.4%. Interestingly, out of the 28 differences, 24 involved 3-lbca, and another 3 involved the equivalent 4-lbcb experiment using the same number of bins. So, complete search only offered a benefit for these strategies.

For the classification setting, not a single triangle in the MW-U table changed, though 22 individual U-scores differ slightly, all except one for 1-lbfa and 3-lbca. In the regression setting, only 4 triangles changed, 2 times from $\triangleleft$ to $\blacktriangleleft$ for 1-lbfa, 2 from $<$ to $\triangleleft$ for 3-lbca, and one changed in favour of 3-lbca from $<$ to $>$. Here, 33 U-scores changed, again mostly for 1-lbfa and 3-lbca.

The negligible difference between the results of complete and beam search is actually not that surprising. For depth 1, and quite often for depth 2 results also, the extent of the beam search is such that it in fact considers the same candidates as the complete search. Furthermore, Sect. 3.3 argued that using the term 'exhaustive' to describe complete search is problematic. Section 5.1 suggested that some strategies, the *nominal* ones especially, require a low number of bins to perform well, as else, through conjunctions, the subgroups these select become (too) small. In contrast, the *binaries* strategies work better using a higher setting of B. As a result, a heuristic beam search, using a high setting of B, might actually be more *extensive*, and produce better results, than a complete search using fewer bins.

Conclusion The very minimal difference of results between complete search and beam search indicates that the relative order of the numeric strategies is hardly influenced by the search strategy and that beam search is preferable on any but the smallest datasets, for reasons of efficiency and scaleability.

5.6 Diverse subgroups from numeric data

So far, experiments focussed on result quality. This section also considers redundancy in the form of subgroup-extension overlap. Redundancy might cause saturation, which is problematic when domain experts prefer a result set containing a limited but diverse set of subgroups, and when it limits search space exploration, and result quality, during a beam search.

Of the specialised redundancy-reduction techniques listed in Sect. 2.2, an attractive member of the popular DSSD family (van Leeuwen and Knobbe 2012) is used. This cover-based subgroup selection variant was evaluated using a weight parameter of 0.9, and a beam width and result set size of 100, all as in the original work.

For each of the strategies listed in Table 1, results were obtained using a traditional and CBSS beam search. All but one of the strategies are heuristics that naturally reduce redundancy, and these experiments gauge to what extent the (memory-wise and computationally far more demanding) CBSS technique has added benefit over a pure and straightforward SD approach. Performance is analysed both from the perspective of attained quality and joint entropy H (Knobbe and Ho 2006a, b; van Leeuwen and Knobbe 2012) of the top-10 subgroups in the final result set.

The CBSS procedure indeed produces higher entropy scores than the presented beam-search setting. In the classification setting, it achieves a higher entropy in 36% of the experiments, but for 60% it was equal, and for the remainder beam search actually obtains a higher entropy. In the regression setting, the CBSS entropy was higher for 44% of the experiments, and equal for 53%. On average, the CBSS entropy was higher by 0.214 bits (classification setting) and 0.373 bits (regression setting).

As argued in the original work, redundancy-reduction might not only produce more diversity, it might also boost the quality of the final subgroups, since the diversity in the intermediate search levels potentially allows for more exploration, leading to better results. However, in 99% of the 1584 *WRAcc* experiments, and 95% of the 1820 using $|z\text{-}score|$, no difference in top-1 quality was observed. In fact, for $|z\text{-}score|$ 2% of CBSS results were indeed better, but 3% were actually worse than that of the beam search. Therefore, one has to conclude that in terms of quality, CBSS does not produce better results than the presented beam search, under these settings.

In order to judge the relative merit of the various numeric strategies presented in an objective sense, a single metric by which to compare strategies is required. In earlier experiments, the quality of the top 10 subgroups was used, but since in this context, diversity is the prime focus, the joint entropy of the top-10 is assessed. This metric will be used both for picking the optimal number of bins per dataset (a prerequisite for the discretisation-based methods, which is not discussed in detail) and for producing the final rankings of strategies. A number of related findings are presented first, followed by an overall discussion.

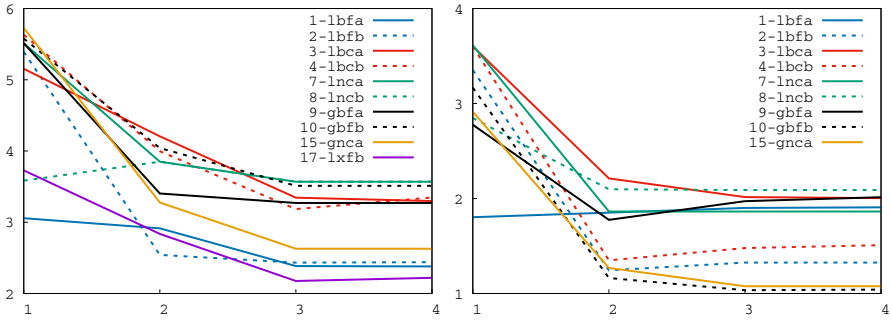

Fig. 3 Demonstration of the development of joint entropy, shown on the y-axis, at increasing depth, for datasets *adult* and *abalone*, for classification and regression settings, respectively

A first observation to make is that the obtained joint entropy is significantly affected by the search depth. Specifically, the largest values for H are typically obtained at depth 1. At depth 2, a large drop in joint entropy is observed, while at subsequent depths, there are no major changes. These phenomena are illustrated for two experiments in Fig. 3. Additionally, the ranking of strategies at depth 1 compared to greater depths are only moderately correlated, such that findings for both depths 1 and 2 are reported.

Figure 4 shows the final ranking of the strategies for the classification setting on the left, and the regression setting on the right. For the latter, post-hoc Nemenyi tests are not permitted, so the diagrams show only the rank and order of strategies. Most notably, the strategies that rank high in the earlier quality-based beam search results, 1-lbfa, 2-lbfb (and 17-lxfb for *WRAcc*), are amongst the worst-scoring strategies when it comes to joint entropy, for all depths. The poorly performing *nominal* strategies now rank high. Strategy 7-lnca is notable, as it ranks well for both quality and redundancy-based results.

Overall, it seems that the advice for optimising quality using beam search should be reversed for reducing redundancy using CBSS: strategies that work well for the former tend to not work well for the latter, and vice versa. Looking at the correlation between the average ranks for the two search strategies, this reverse trend is indeed confirmed. For *WRAcc*, a negative correlation is observed of -0.357 for depth 1, and -0.722 for depth 2. For $|z\text{-}score|$, these are -0.605 (depth 1), and -0.680 (depth 2).

Discussion The facts that CBSS does not produce better quality results, and that reverse rankings of strategies are observed, are not surprising, and can all be explained by the same observations. First, DSSD is not SD, and the measures presented in van Leeuwen and Knobbe (2011, 2012), cover redundancy CR, entropy H, and cover-based selection Ω, are all global, they value uniform coverage of the records. In the context of SD, this is irrelevant, as only records with favourable target values (positives and numeric extremes) are of interest, and their distribution is often unbalanced.

So, strategies that allow many top subgroups to accurately capture the relevant target values attain high quality, but low uniformity. High uniformity is achieved using the imprecise, disjunct, consecutive *nominal* strategies, but quality suffers. The drop in

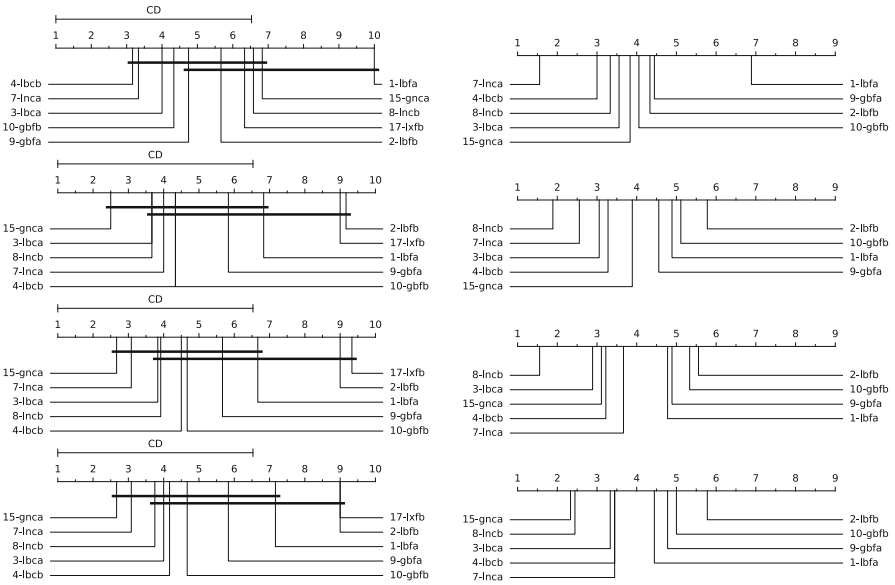

Fig. 4 Diagrams showing the rank and order of strategies based on the entropy of the top-10, using the CBSS technique, for depth 1, 2, 3, and 4 (top to bottom). Results for the classification setting, using *WRAcc*, are on the left, and on the right are the results for the regression setting, using |*z-score*|. For the latter, post-hoc Nemenyi tests are not permitted, so the diagrams show only the rank and order of strategies. The strategies that rank high in the earlier quality-based beam search results, 1-lbfa, 2-lbfb (and 17-lxfb for *WRAcc*), are amongst the worst-scoring strategies when it comes to joint entropy, for all depths. The *nominal* strategies that perform poorly when ranked on quality, now rank high.

entropy score above depth 1 occurs because, regardless of the employed strategy, the top (quality score) subgroups better capture the (skewed) target at greater depths.

Entropy and cover-based selection might be more useful when used in other ways. First, Knobbe and Ho (2006a, b) also use H, but only to select a diverse set from a limited top-k result set of high quality subgroups, say 10 out of 30. Also, the covering approaches used in APRIORI-SD (Kavšek and Lavrač 2006) and CN2-SD (Lavrač et al. 2004) weight only positive records, which prevents that at some point covering negatives becomes preferable over covering positives in terms of selection score.

Conclusion CBSS does not yield better top-1 results than traditional beam search, but it does yield higher entropy scores, especially using *local*, *nominal*, *coarse* strategies.

5.7 Notes on run times

While the primary focus of this work is on the quality of results of various strategies, run times are also relevant when choosing a practical strategy for a given dataset. This section provides details on the run times incurred by the above experiments. Experiments were performed sequentially in single-thread mode using a 2013 dual-octacore Xeon E5-2650V2 server processor (2.60 GHz). Run times should be considered indicative

Table 11 For the experiments performed using the different search strategies, the main statistics concerning run times are given

Statistic	Classification			Regression		
	Beam	CBSS	Complete	Beam	CBSS	Complete
Experiments	4752	1584	1188	5460	1820	1365
$< 1\ s$	4223	1065	1023	5445	1248	1241
$< 10\ s$	4745	1324	1137	5460	1564	1299
$< 60\ s$	4751	1424	1166	5460	1722	1338
$\mu\ (s)$	0.518	678.5	126.9	0.130	138.1	540.6
Median (s)	0.095	0.266	0.082	0.086	0.228	0.068
Maximum (s)	66.78	511,008	139,825	3.136	102,040	647,334

Per depth and quality measure, 396 and 455 experiments were performed for the classification and regression setting, respectively, but only for the beam search setting results using addition quality measures were obtained

only, as experiments were timed only once, and at any moment 15 other (unrelated) experiments would share the cpu and memory resources (256 GigaByte).

Table 11 shows the run time statistics for the various settings. Per depth and quality measure, 396 and 455 experiments were performed for the classification and regression setting, respectively. The maximum search depth was 4 for the beam and CBSS search strategy, and 3 for complete search. For all search strategies, results are available for *WRAcc* and $|z\text{-}score|$, for the beam search all measures in Table 3 were used. Six results are missing (three for CBSS, three for complete) as they either use too much memory, or take too long.

All beam experiments terminate within 10 seconds, except for 7 17-lxfb experiments in the classification setting at depth 3 or 4, of which the longest took 66.78 seconds. This is due to its quadratic, instead of linear, implementation, as referred to in Sect. 5.2.7. Although the issue was identified, experiments use this implementation to gauge whether a *Cartesian* alternative for the *interval type*, as described in Sect. 4, is feasible. At least for this target type and search strategy, results are promising, as depth 4 experiments for the largest dataset, *covertype*, took 15, 58, and 67 seconds for *lift*, *binomial*, and *WRAcc*, respectively.

The 1-lbfa strategy requires most time, but for the depth 4 *covertype* experiments, there is less than 1.7 seconds separating 1-lbfa and 3-lbca, the most involved beam setting. Also, strategies favouring larger subgroups show the longest run times, but for the regression setting, with a maximum of 3.136 seconds, this does not say much.

Run times for the CBSS setting are much worse, with more than 10% of the experiments not completing within a minute, and some taking much longer. In this respect, CBSS compares very unfavourably to the equivalent beam search results, and is often unusable due to its memory requirement and excessive run times. It should be noted here that the only difference between the two implementations is the cover-based selection procedure for the candidate beam and result set, which is thus solely responsible for the time differences.

The complete search experiments were only performed up to depth 3, as some 1-lbfa runs already took a long time to finish. Still, most experiments complete within a second, and for the classification setting the longest non-1-lbfa experiment, using 3-lbca, took about 7 min. For the regression setting, the longest non-1-lbfa experiment took 45 min.

Conclusion The various numeric strategies behave as expected, with the extensive 1-lbfa strategy yielding the longest run times. Although there is some explainable variation in run time between the strategies, on the whole, beam search is very fast (the vast majority finishes in several seconds), suggesting that run time should not be a determining factor in the choice of strategy. To a lesser degree, this is also the case for complete search, which on average is (much) slower, but this is caused by a small fraction of very long runs (specifically involving 1-lbfa). CBSS is simply quite slow.

6 Conclusions and future work

This work systematically examined a host of SD strategies. These strategies differ along a number of dimensions, and experiments were performed to gain insights into the effects of different options within these dimensions. Choices were not evaluated in isolation, but always in the context of other parameter settings, as this is required to gauge real-world performance.

Most of the findings are not unexpected, for reasons pointed out in the sections introducing each dimension. However, the fact that a single parameter choice would show markedly different behaviour in the classification and regression target type settings was unforeseen. Furthermore, it is especially the scale at which some strategies perform worse than others that is remarkable.

As a whole, this systematic evaluation both affirms some intuitions that, to the best of our knowledge, have never been rigorously tested, and garnered new insights into both existing strategies, and into how to improve future algorithm design. As such, it is of value for those seeking information guiding an informed choice regarding the analysis of real-world data. Additionally, its findings can be of benefit to researchers and algorithm designers alike.

The experiments have shown that better performance is achieved by those settings that finely tune the placement of the threshold (for example, *fine* and *local*). This means that *local* discretisation (choosing a threshold in the context of the subset currently under investigation) is often preferred over *global* discretisation (discretising the data prior to the discovery process), and *fine* granularity typically produces better results than *coarse* discretisation. Additionally, the *binaries* representation consistently outperforms the *nominal* one, and multiple candidates should be considered per numeric attribute, rather than only the single *best*. There is hardly a difference in computation time, and especially in the regression setting, 3-lbca performs really well. Experiments with alternative quality measures, ranging in how they treat the size of the subgroups found, indicate that the findings are mostly stable to the choice of measure. While the findings are comparable between beam and heuristic search, much different conclusions can be drawn from the CBSS experiments. As CBSS values diversity above accuracy of the subgroups, *local*, *nominal* and *coarse* strategies tend to be preferred.

Future work Dealing with (multivariate) numeric datasets is about choosing thresholds. In this regard, a number of improvements are possible.

First, run times indicate that the quadratic implementation of the BestInterval algorithm (17-lxfb) is not a limiting factor, at least not for this target type, where the model computation is cheap. Therefore, strategies evaluating the quadratic number of intervals, both using all cut points or a discretised subset thereof, will be explored.

Also, *more* should be done within a search level, not less, the run times indicate there is room to do so. A richer description language, and more extensive evaluation within a level, yield better subgroups at lower depths. In turn, the search can be less deep, and much of the combinatorial explosion of the search space is avoided.

This can often be achieved without increasing complexity. For example, the exact same cut points need to be established for *binaries*, *nominal*, and a quadratic interval implementation. Thereafter, the difference lies in how these are used to create descriptions. In combination with beam search, the size of search space is kept under control through the beam width.

Algorithms might also benefit from a better separation between which subgroups are good for a result set, and which are useful candidates for refinement. Therefore, the BestInterval algorithm will be extended to behave like the *best* strategies. Even when the algorithm is unable to find a single valid subgroup for the result set, it could still yield a useful candidate. Also, the technique uses a convex hull to select candidates, like Meeng et al. (2014), but the latter allows multiple, instead of just one, which offers additional directions to explore. More sophisticated variants of the *binaries* strategies, that select both small and large subgroups, are another direction to explore.

Another factor that limits the full exploitation of the richness of numeric datasets is related to the greedy nature of many (heuristic) SD algorithms. Most do not consider the joint distribution of numeric attributes when selecting thresholds. But, there is no guarantee that an 'optimal' threshold at a particular depth is still ideal when further refined.

In future work, the *simultaneous optimisation* of thresholds on numeric attributes will receive attention. The work of Mampaey et al. (2015) already investigated combined efficient optimisation of two thresholds on a single attribute, but considering combinations of *multiple* attributes might be even more fruitful (Nguyen et al. 2014).

A Appendix

See Tables 12, 13, 14, 15 and 16.

Table 12 Tables showing, for each strategy, the (minimal) number of bins resulting in highest score for top-k = 1

($\varphi = WRAcc$)	3-lbca				4-lbcb				7-lnca				8-lncb				9-gbfa				10-gbfb				15-gnca			
Dataset	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
adult	7	10	10	9	7	3	3	3	2	2	2	2	2	2	2	2	2	7	7	7	2	4	4	4	2	2	2	2
covertype	5	8	9	9	5	8	8	8	2	2	2	2	2	2	2	2	5	10	10	10	5	10	10	10	2	2	2	2
credit-a	2	6	10	10	2	6	6	6	2	2	2	2	2	2	2	2	2	4	4	4	2	4	4	4	2	2	2	2
ionosphere	9	5	8	10	9	5	9	10	3	3	3	3	3	3	3	3	9	10	10	10	9	10	10	10	3	3	3	3
pima-indians	7	9	9	10	7	3	4	4	2	2	2	2	2	2	2	2	7	7	8	8	7	7	7	7	2	2	2	2
wisconsin	3	4	5	5	3	4	8	8	3	3	3	3	3	3	3	3	5	7	7	7	5	7	7	7	3	2	2	2
$\mu(B)$ per depth	$5\frac{3}{6}$	7	$8\frac{3}{6}$	$8\frac{5}{6}$	$5\frac{3}{6}$	$4\frac{5}{6}$	$6\frac{2}{6}$	$6\frac{3}{6}$	$2\frac{2}{6}$	$2\frac{2}{6}$	$2\frac{2}{6}$	$2\frac{2}{6}$	$2\frac{2}{6}$	$2\frac{2}{6}$	$2\frac{2}{6}$	$2\frac{2}{6}$	5	$7\frac{3}{6}$	$7\frac{4}{6}$	$7\frac{4}{6}$	5	7	7	7	$2\frac{2}{6}$	$2\frac{1}{6}$	$2\frac{1}{6}$	$2\frac{1}{6}$
$\mu(B)$ overall	7.46				5.79				2.33				2.33				6.96				6.50				2.21			

($\varphi = \lvert z\text{-}score \rvert$)	3-lbca				4-lbcb				7-lnca				8-lncb				9-gbfa				10-gbfb				15-gnca			
Dataset	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
abalone	9	7	9	9	9	7	7	7	5	5	5	5	5	5	5	5	9	10	10	10	9	7	7	7	5	4	4	4
adult	5	6	8	8	5	6	8	8	3	3	3	3	3	3	3	3	4	7	3	3	4	7	3	3	3	3	3	3
auto-mpg	8	6	6	6	8	6	6	6	3	2	2	2	3	2	2	2	5	8	8	8	5	8	8	8	3	3	3	3
boston-housing	8	7	8	10	8	7	10	10	8	9	2	2	8	9	2	2	8	6	5	5	8	6	5	5	8	8	8	8

Table 12 continued

($\varphi = \lvert z\text{-}score \rvert$)	3-lbca				4-lbcb				7-lnca				8-lncb				9-gbfa				10-gbfb				15-gnca			
Dataset	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
communities	5	9	9	9	5	9	9	9	5	2	2	2	5	2	2	2	9	9	4	10	9	9	4	8	5	4	4	4
forestfires	8	7	7	9	8	7	7	9	7	7	7	2	7	7	7	2	7	9	9	9	7	9	9	7	7	7	2	2
pima-indians	3	5	10	8	3	9	9	10	3	2	2	2	3	2	2	2	5	9	9	9	5	9	9	9	3	3	3	3
$\mu(B)$ per depth	$6\frac{4}{7}$	$6\frac{5}{7}$	$8\frac{1}{7}$	$8\frac{3}{7}$	$6\frac{4}{7}$	$7\frac{2}{7}$	8	$8\frac{3}{7}$	$4\frac{6}{7}$	$4\frac{2}{7}$	$3\frac{2}{7}$	$2\frac{4}{7}$	$4\frac{6}{7}$	$4\frac{2}{7}$	$3\frac{2}{7}$	$2\frac{4}{7}$	$6\frac{5}{7}$	$8\frac{2}{7}$	$6\frac{6}{7}$	$7\frac{5}{7}$	$6\frac{5}{7}$	$7\frac{6}{7}$	$6\frac{3}{7}$	$6\frac{5}{7}$	$4\frac{6}{7}$	$4\frac{4}{7}$	$3\frac{6}{7}$	$3\frac{6}{7}$
$\mu(B)$ overall	7.46				7.57				3.75				3.75				7.39				6.93				4.29			

$\mu(B)$ per depth	3-lbca				4-lbcb				7-lnca				8-lncb				9-gbfa				10-gbfb				15-gnca			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
lift	$8\frac{3}{6}$	$5\frac{1}{6}$	$5\frac{2}{6}$	5	$8\frac{3}{6}$	$4\frac{2}{6}$	$5\frac{3}{6}$	$6\frac{3}{6}$	$6\frac{4}{6}$	4	$2\frac{2}{6}$	$2\frac{1}{6}$	$6\frac{4}{6}$	4	$2\frac{2}{6}$	$2\frac{1}{6}$	$5\frac{1}{6}$	$7\frac{1}{6}$	$5\frac{5}{6}$	$5\frac{4}{6}$	$5\frac{1}{6}$	$6\frac{1}{6}$	5	$4\frac{2}{6}$	$6\frac{4}{6}$	$4\frac{5}{6}$	$3\frac{5}{6}$	$3\frac{5}{6}$
binomial	$6\frac{3}{6}$	$7\frac{3}{6}$	$7\frac{4}{6}$	$7\frac{1}{6}$	$6\frac{3}{6}$	$7\frac{5}{6}$	$7\frac{5}{6}$	$7\frac{5}{6}$	4	$3\frac{2}{6}$	$3\frac{2}{6}$	$3\frac{2}{6}$	4	$3\frac{2}{6}$	$3\frac{2}{6}$	$3\frac{2}{6}$	$6\frac{3}{6}$	$8\frac{1}{6}$	$8\frac{4}{6}$	$8\frac{4}{6}$	$6\frac{3}{6}$	$7\frac{5}{6}$	$8\frac{2}{6}$	$8\frac{2}{6}$	4	$3\frac{2}{6}$	$3\frac{1}{6}$	$3\frac{1}{6}$
WRAcc	$5\frac{3}{6}$	7	$8\frac{3}{6}$	$8\frac{5}{6}$	$5\frac{3}{6}$	$4\frac{5}{6}$	$6\frac{2}{6}$	$6\frac{3}{6}$	$2\frac{2}{6}$	$2\frac{2}{6}$	$2\frac{2}{6}$	$2\frac{2}{6}$	$2\frac{2}{6}$	$2\frac{2}{6}$	$2\frac{2}{6}$	$2\frac{2}{6}$	5	$7\frac{3}{6}$	$7\frac{4}{6}$	$7\frac{4}{6}$	5	7	7	7	$2\frac{2}{6}$	$2\frac{1}{6}$	$2\frac{1}{6}$	$2\frac{1}{6}$
$\lvert deviation \rvert$	$8\frac{6}{7}$	$7\frac{3}{7}$	7	$5\frac{4}{7}$	$8\frac{6}{7}$	$7\frac{6}{7}$	$7\frac{3}{7}$	$5\frac{2}{7}$	$7\frac{5}{7}$	$6\frac{6}{7}$	$6\frac{3}{7}$	$4\frac{1}{7}$	$7\frac{5}{7}$	$6\frac{6}{7}$	$6\frac{3}{7}$	$4\frac{1}{7}$	$8\frac{2}{7}$	$8\frac{3}{7}$	$8\frac{3}{7}$	$8\frac{6}{7}$	$8\frac{2}{7}$	$8\frac{4}{7}$	$8\frac{4}{7}$	$8\frac{6}{7}$	$7\frac{5}{7}$	$6\frac{1}{7}$	$5\frac{3}{7}$	$5\frac{3}{7}$
$\lvert t\text{-}statistic \rvert$	$6\frac{3}{7}$	$5\frac{6}{7}$	$6\frac{1}{7}$	$7\frac{1}{7}$	$6\frac{3}{7}$	$5\frac{4}{7}$	$5\frac{2}{7}$	$6\frac{6}{7}$	$4\frac{4}{7}$	$5\frac{3}{7}$	$4\frac{6}{7}$	$3\frac{6}{7}$	$4\frac{4}{7}$	$5\frac{2}{7}$	$4\frac{4}{7}$	$3\frac{5}{7}$	$6\frac{4}{7}$	6	$6\frac{3}{7}$	$7\frac{2}{7}$	$6\frac{4}{7}$	$6\frac{1}{7}$	$5\frac{6}{7}$	$6\frac{3}{7}$	$4\frac{4}{7}$	$4\frac{2}{7}$	$4\frac{2}{7}$	$4\frac{2}{7}$
$\lvert z\text{-}score \rvert$	$6\frac{4}{7}$	$6\frac{5}{7}$	$7\frac{5}{7}$	$8\frac{3}{7}$	$6\frac{4}{7}$	$7\frac{2}{7}$	$7\frac{3}{7}$	$8\frac{3}{7}$	$4\frac{6}{7}$	$4\frac{2}{7}$	$3\frac{2}{7}$	$2\frac{4}{7}$	$4\frac{6}{7}$	$4\frac{3}{7}$	$4\frac{3}{7}$	$3\frac{5}{7}$	$6\frac{5}{7}$	$8\frac{2}{7}$	$6\frac{6}{7}$	$7\frac{5}{7}$	$6\frac{5}{7}$	$7\frac{6}{7}$	$6\frac{3}{7}$	$6\frac{5}{7}$	$4\frac{6}{7}$	$4\frac{4}{7}$	$3\frac{6}{7}$	$3\frac{6}{7}$

The first and second table show the complete results for *WRAcc*, and $\lvert z\text{-}score \rvert$, respectively. The third table is a summary for all quality measures. The line '$\mu(B)$ per depth' lists the average, over all datasets, of the number of bins resulting in the highest score. The numbers listed as '$\mu(B)$ overall' average over depths. These numbers are rounded to the nearest integer and used in the subsequent analyses. Table 6 lists these settings

Table 13 Quality scores for the classification setting, using quality measure *WRAcc*

Dataset	Depth	1-lbfa		2-lbfb		3-lbca^7		4-lbcb^6		7-lnca^2		8-lncb^2		9-gbfa^7		10-gbfb^7		15-gnca^2		17-lxfb	
		$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r
adult	1	0.061	5	0.061	5	0.061	5	0.060	10	0.061	5	0.061	5	0.061	5	0.061	5	0.061	5	0.061	5
covertype	1	0.086	2.5	0.086	2.5	0.085	5	0.083	7	0.075	9	0.075	9	0.085	5	0.085	5	0.075	9	0.116	1
credit-a	1	0.113	5.5	0.113	5.5	0.113	5.5	0.113	5.5	0.113	5.5	0.113	5.5	0.113	5.5	0.113	5.5	0.113	5.5	0.113	5.5
ionosphere	1	0.129	2.5	0.129	2.5	0.114	6	0.106	7	0.076	9	0.076	9	0.119	4.5	0.119	4.5	0.076	9	0.145	1
pima-indians	1	0.098	3	0.098	3	0.097	6	0.094	10	0.094	8	0.094	8	0.098	3	0.098	3	0.094	8	0.098	3
wisconsin	1	0.194	3.5	0.194	3.5	0.185	7	0.194	3.5	0.184	9	0.184	9	0.194	3.5	0.194	3.5	0.184	9	0.194	3.5
$\mu(r_j, 1)(F_F = 2.914)$			3.67		3.67		5.75		7.17		7.58		7.58		4.42		4.42		7.58		3.17
adult	2	0.076	1.5	0.073	5	0.076	3	0.076	4	0.061	9	0.061	9	0.076	1.5	0.070	7	0.061	9	0.071	6
covertype	2	0.116	2.5	0.116	2.5	0.115	4	0.109	5	0.075	9	0.075	9	0.104	6.5	0.104	6.5	0.075	9	0.116	1
credit-a	2	0.121	2	0.121	2	0.120	4.5	0.120	4.5	0.113	9	0.113	9	0.120	6.5	0.120	6.5	0.113	9	0.121	2
ionosphere	2	0.187	1.5	0.187	1.5	0.170	4	0.159	7	0.081	9	0.079	10	0.166	5.5	0.166	5.5	0.091	8	0.186	3
pima-indians	2	0.105	1	0.103	7	0.105	4	0.104	6	0.094	9	0.094	9	0.105	2.5	0.105	2.5	0.094	9	0.104	5
wisconsin	2	0.202	3	0.202	3	0.199	6	0.199	7	0.184	9.5	0.184	9.5	0.202	3	0.202	3	0.196	8	0.202	3
$\mu(r_j, 2)(F_F = 14.840)$			1.92		3.50		4.25		5.58		9.08		9.25		4.25		5.17		8.67		3.33

Table 13 continued

Dataset	Depth	1-lbfa		2-lbfb		3-lbca^7		4-lbcb^6		7-lnca^2		8-lncb^2		9-gbfa^7		10-gbfb^7		15-gnca^2		17-lxfb	
		$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r
adult	3	0.077	1.5	0.074	5	0.077	3	0.076	4	0.061	9	0.061	9	0.077	1.5	0.070	7	0.061	9	0.072	6
covertype	3	0.116	2.5	0.116	2.5	0.115	4	0.112	5	0.075	9	0.075	9	0.104	6.5	0.104	6.5	0.075	9	0.116	1
credit-a	3	0.123	2	0.123	2	0.120	4.5	0.120	4.5	0.113	9	0.113	9	0.120	6.5	0.120	6.5	0.113	9	0.123	2
ionosphere	3	0.192	1.5	0.192	1.5	0.172	7	0.178	4	0.081	9	0.079	10	0.173	5.5	0.173	5.5	0.091	8	0.191	3
pima-indians	3	0.107	1	0.106	3	0.105	6	0.104	7	0.094	9	0.094	9	0.105	4.5	0.105	4.5	0.094	9	0.106	2
wisconsin	3	0.203	3.5	0.203	3.5	0.203	3.5	0.200	7	0.184	9.5	0.184	9.5	0.203	3.5	0.203	3.5	0.196	8	0.203	3.5
$\mu(r_j, 3)(F_F = 17.281)$			2.00		2.92		4.67		5.25		9.08		9.25		4.67		5.58		8.67		2.92
adult	4	0.077	1	0.074	5	0.077	3	0.076	4	0.061	9	0.061	9	0.077	2	0.070	7	0.061	9	0.072	6
covertype	4	0.116	2.5	0.116	2.5	0.115	4	0.112	5	0.075	9	0.075	9	0.104	6.5	0.104	6.5	0.075	9	0.116	1
credit-a	4	0.124	2	0.124	2	0.120	4.5	0.120	4.5	0.113	9	0.113	9	0.120	6.5	0.120	6.5	0.113	9	0.124	2
ionosphere	4	0.196	3	0.196	2	0.181	4	0.178	5	0.081	9	0.079	10	0.176	6.5	0.176	6.5	0.091	8	0.199	1
pima-indians	4	0.108	1	0.107	3	0.105	6	0.104	7	0.094	9	0.094	9	0.105	4.5	0.105	4.5	0.094	9	0.107	2
wisconsin	4	0.203	3.5	0.203	3.5	0.203	3.5	0.200	7	0.184	9.5	0.184	9.5	0.203	3.5	0.203	3.5	0.196	8	0.203	3.5
$\mu(r_j, 4)(F_F = 18.553)$			2.17		3.00		4.17		5.42		9.08		9.25		4.92		5.75		8.67		2.58
$\mu(r_j, (1, 2, 3, 4))$			2.44		3.27		4.71		5.85		8.71		8.83		4.56		5.23		8.40		3.00

For each strategy, a row lists, for each dataset and depth, the score $\varphi_{\mathcal{D}}$, and rank r based on this score. Per strategy, the average rank over all datasets for a certain depth is given on the line $\mu(r_j, \text{depth})$. By summing the squares of these values, χ^2_F and F_F can be computed, the latter is given, and is used for Fig. 2. Table 9 uses $\mu(r_j, (1, 2, 3, 4))$. Due to rounding, results for which qualities appear equal might list different ranks

Table 14 Quality scores for the regression setting, using the |*z-score*| measure

Dataset	Depth	1-lbfa		2-lbfb		3-lbca^{7}		4-lbcb^{8}		7-lnca^{4}		8-lncb^{4}		9-gbfa^{7}		10-gbfb^{7}		15-gnca^{4}	
		$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r
abalone	1	28.896	1.5	28.896	1.5	28.475	8	28.522	4.5	28.522	4.5	28.522	4.5	28.475	8	28.475	8	28.522	4.5
adult	1	17.885	5.5	17.885	5.5	17.885	5.5	17.885	5.5	22.567	2	22.567	2	16.049	8.5	16.049	8.5	22.567	2
auto-mpg	1	11.553	2	11.553	2	11.441	6	11.553	2	11.301	8.5	11.301	8.5	11.487	4.5	11.487	4.5	11.301	7
boston-housing	1	13.973	1.5	13.973	1.5	13.760	6	13.813	3	12.412	8.5	12.412	8.5	13.765	4.5	13.765	4.5	12.412	7
communities	1	25.280	1.5	25.280	1.5	24.430	8	24.838	3	24.802	5.5	24.802	5.5	24.430	8	24.430	8	24.802	4
forestfires	1	3.573	2	3.573	2	3.174	6	3.573	2	2.368	8	2.368	8	3.315	4.5	3.315	4.5	2.368	8
pima-indians	1	12.233	3	12.233	3	12.233	3	11.834	6	11.476	7.5	11.476	7.5	12.233	3	12.233	3	11.476	9
$\mu(r_j, 1)(F_F = 3.484)$			2.43		2.43		6.07		3.71		6.36		6.36		5.86		5.86		5.93
abalone	2	29.131	1	29.006	5	29.123	2	28.849	6	28.522	8.5	28.522	8.5	29.023	3.5	29.023	3.5	28.756	7
adult	2	23.689	6.5	23.689	6.5	23.689	6.5	23.689	6.5	27.208	2	27.208	2	23.689	6.5	23.689	6.5	27.208	2
auto-mpg	2	12.142	1.5	12.142	1.5	11.933	3	11.873	5	11.301	8.5	11.301	8.5	11.873	5	11.873	5	11.301	7
boston-housing	2	14.656	1	14.484	3	14.492	2	14.328	6	13.110	8.5	13.110	8.5	14.416	4.5	14.416	4.5	13.757	7
communities	2	26.246	1	26.203	2	26.157	3	26.139	4	24.802	8.5	24.802	8.5	25.610	6	25.553	7	26.138	5
forestfires	2	4.471	1	4.292	3	4.360	2	4.146	6	3.720	7.5	3.720	7.5	4.157	4.5	4.157	4.5	2.789	9
pima-indians	2	12.954	1.5	12.954	1.5	12.679	5	12.442	6	11.476	7.5	11.476	7.5	12.906	3.5	12.906	3.5	11.476	9
$\mu(r_j, 2)(F_F = 5.503)$			1.93		3.21		3.36		5.64		7.29		7.29		4.79		4.93		6.57

Table 14 continued

Dataset	Depth	1-lbfa		2-lbfb		3-lbca[7]		4-lbcb[8]		7-lnca[4]		8-lncb[4]		9-gbfa[7]		10-gbfb[7]		15-gnca[4]	
		$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r	$\varphi_{\mathcal{D}}$	r
abalone	3	29.194	1	29.022	5	29.130	2	28.849	6	28.522	8.5	28.522	8.5	29.023	3.5	29.023	3.5	28.756	7
adult	3	30.445	2	30.445	2	28.795	7	30.445	2	29.284	5	29.284	5	27.208	8.5	27.208	8.5	29.284	5
auto-mpg	3	12.142	1.5	12.142	1.5	12.026	3	11.917	4	11.301	8.5	11.301	8.5	11.873	5.5	11.873	5.5	11.301	7
boston-housing	3	14.741	1	14.572	2	14.543	3	14.328	6	13.110	8.5	13.110	8.5	14.502	4.5	14.502	4.5	13.757	7
communities	3	26.903	1	26.616	4	26.654	3	26.683	2	24.802	8.5	24.802	8.5	26.478	5	26.345	6	26.147	7
forestfires	3	4.678	3	4.746	2	4.864	1	4.558	4	3.836	7.5	3.836	7.5	4.494	5.5	4.494	5.5	2.789	9
pima-indians	3	13.049	1.5	13.049	1.5	12.954	3	12.442	6	11.476	7.5	11.476	7.5	12.906	4.5	12.906	4.5	11.476	9
$\mu(r_j, 3)(F_F = 13.820)$			1.57		2.57		3.14		4.29		7.71		7.71		5.29		5.43		7.29
abalone	4	29.194	1	29.024	5	29.130	2	28.849	6	28.522	8.5	28.522	8.5	29.027	3.5	29.027	3.5	28.756	7
adult	4	33.732	2	33.732	2	32.012	4	33.732	2	29.924	6	29.924	6	29.284	8.5	29.284	8.5	29.924	6
auto-mpg	4	12.142	1.5	12.142	1.5	12.026	3	11.917	4	11.301	8.5	11.301	8.5	11.873	5.5	11.873	5.5	11.301	7
boston-housing	4	14.809	1	14.624	2	14.561	5	14.328	6	13.110	8.5	13.110	8.5	14.596	3.5	14.596	3.5	13.757	7
communities	4	27.278	1	27.024	2	26.887	4	26.991	3	24.802	8.5	24.802	8.5	26.727	5	26.682	6	26.147	7
forestfires	4	4.687	6	4.906	3	4.925	2	4.784	5	3.897	7.5	3.897	7.5	4.817	4	4.969	1	2.789	9
pima-indians	4	13.166	1.5	13.166	1.5	12.954	3	12.442	6	11.476	7.5	11.476	7.5	12.906	4.5	12.906	4.5	11.476	9
$\mu(r_j, 4)(F_F = 13.039)$			2.00		2.43		3.29		4.57		7.86		7.86		4.93		4.64		7.43
$\mu(r_j, (1, 2, 3, 4))$			1.98		2.66		3.96		4.55		7.30		7.30		5.21		5.21		6.80

Figure 2 uses information from this table. For its interpretation, refer to Table 13

Table 15 Mann-Whitney U scores for the classification setting, and quality measure *WRAcc*

strategies	adult				covertype				credit-a				ionosphere				pima-indians				wisconsin				wins				≤ 27/≥ 73/valid			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1-lbfa vs. 2-lbfb	–	2.5	0	0	9.5	9.5	9.5	9.5	15	11	17	9.5	9.5	9.5	28.5	38	9.5	0	0	0	26.5	30	39	34.5	◀	◁	◁	◁	5/0/5	5/0/6	4/0/6	4/0/6
1-lbfa vs. 3-lbca^{7}	23.5	11.5	1.5	0	0	0	0	0	17	9	0	0	0	0	0	0	7.5	0	0	0	21.5	32	24.5	39	◀	◁	◀	◁	6/0/6	5/0/6	6/0/6	5/0/6
1-lbfa vs. 4-lbcb^{6}	–	7.5	0	0	0	0	0	0	9.5	8.5	0	0	0	0	0	0	0	0	0	0	26	22.5	2.5	0	◀	◀	◀	◀	5/0/5	6/0/6	6/0/6	6/0/6
1-lbfa vs. 7-lnca^{2}	–	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0.5	0	0	0	11	0	0	0	◀	◀	◀	◀	5/0/5	6/0/6	6/0/6	6/0/6
1-lbfa vs. 8-lncb^{2}	–	0	0	0	0	0	0	0	–	0	0	0	0	0	0	0	–	0	0	0	–	0	0	0	◀	◀	◀	◀	2/0/2	6/0/6	6/0/6	6/0/6
1-lbfa vs. 9-gbfa^{7}	22	10	9.5	0	0	0	0	0	18	5.5	0	0	0	0	0	0	9.5	3	0	0	29	50	36	21.5	◁	<	◁	◀	5/0/6	5/0/6	5/0/6	6/0/6
1-lbfa vs. 10-gbfb^{7}	–	0.5	0	0	0	0	0	0	9.5	5.5	0	0	0	0	0	0	9.5	3	0	0	24.5	31	25	11	◀	◁	◀	◀	5/0/5	5/0/6	6/0/6	6/0/6
1-lbfa vs. 15-gnca^{2}	–	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0.5	0	0	0	11	1	0	0	◀	◀	◀	◀	5/0/5	6/0/6	6/0/6	6/0/6
1-lbfa vs. 17-lxfb	–	1	0	0	10	90	48.5	31	–	11.5	19.5	18.5	65	8	10	92	–	0	4	0	–	30	39	34.5	=	<	◁	<	1/0/2	4/1/6	4/0/6	3/1/6
2-lbfb vs. 3-lbca^{7}	–	84.5	100	100	78	72	0	0	74.5	53.5	2.5	0	32	9	0	0	78	73.5	77	0	56	60.5	41.5	47.5	>	>	<	<	0/3/5	1/2/6	3/2/6	4/1/6
2-lbfb vs. 4-lbcb^{6}	–	25	30	40	35	23	0	0	43.5	12.5	2.5	0	30	3	0	0	44	10	8	0	45.5	40.5	6.5	0	◁	◁	◁	◁	0/0/5	5/0/6	5/0/6	5/0/6
2-lbfb vs. 7-lnca^{2}	–	2.5	0	0	27	0	0	0	38	3.5	0	0	0	0	0	0	35	0	0	0	42.5	0	0	0	◁	◀	◀	◀	2/0/5	6/0/6	6/0/6	6/0/6
2-lbfb vs. 8-lncb^{2}	–	2.5	0	0	27	0	0	0	–	3.5	0	0	0	0	0	0	–	0	0	0	–	0	0	0	◀	◀	◀	◀	2/0/2	6/0/6	6/0/6	6/0/6
2-lbfb vs. 9-gbfa^{7}	–	77.5	30	40	78	31	0	0	75.5	36.5	0	0	49	10	0	0	78.5	58	46	0	61	70	51	31.5	>	=	<	◁	0/3/5	1/1/6	3/0/6	4/0/6
2-lbfb vs. 10-gbfb^{7}	–	33	0	0	40.5	23	0	0	44.5	28.5	0	0	35	7	0	0	44.5	29	13	0	48.5	57	35	17.5	◁	<	◁	◀	0/0/5	2/0/6	5/0/6	6/0/6
2-lbfb vs. 15-gnca^{2}	–	2.5	0	0	27	0	0	0	38	3.5	0	0	0	0	0	0	35	0	0	0	42.5	17	0	0	◁	◀	◀	◀	2/0/5	6/0/6	6/0/6	6/0/6
2-lbfb vs. 17-lxfb	–	43.5	0	0	52.5	94.5	93.5	87	–	35.5	37.5	31	94	84	41	93.5	–	40.5	58.5	31.5	–	45	50	50	▷	<	<	<	0/1/2	0/2/6	1/1/6	1/2/6
3-lbca^{7} vs. 4-lbcb^{6}	–	9	5	0	15	16	16	11	22.5	14	10.5	10	38	22	75	15	21	9	3	1	36.5	34	7.5	0	◁	◁	<	◀	3/0/5	5/0/6	5/1/6	6/0/6
3-lbca^{7} vs. 7-lnca^{2}	–	0	0	0	12	0	0	0	19.5	3.5	0	0	0	0	0	0	15	0	0	0	33	0	0	0	◁	◀	◀	◀	4/0/5	6/0/6	6/0/6	6/0/6
3-lbca^{7} vs. 8-lncb^{2}	–	0	0	0	12	0	0	0	–	3.5	0	0	0	0	0	0	–	0	0	0	–	0	0	0	◀	◀	◀	◀	2/0/2	6/0/6	6/0/6	6/0/6
3-lbca^{7} vs. 9-gbfa^{7}	49	25	26	26	49.5	15	0	0	49.5	29.5	34.5	38	69.5	41	31	2.5	52	43	45	70	55.5	68	63.5	36	=	<	<	<	0/0/6	2/0/6	2/0/6	3/0/6

Table 15 continued

strategies	adult				covertype				credit-a				ionosphere				pima-indians				wisconsin				wins				≤ 27/≥ 73/valid			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
3-lbca^{7} vs. 10-gbfb^{7}	–	13.5	0	0	17.5	10	0	0	22.5	24.5	20	18.5	50.5	29	31	2.5	19	27	12	10	41	43	41.5	19.5	<	◁	◁	◀	3/0/5	4/0/6	4/0/6	6/0/6
3-lbca^{7} vs. 15-gnca^{2}	–	0	0	0	12	0	0	0	19.5	3.5	0	0	0	0	0	0	15	0	0	0	33	7	0	0	◁	◀	◀	◀	4/0/5	6/0/6	6/0/6	6/0/6
3-lbca^{7} vs. 17-lxfb	–	15	0	0	19	100	100	100	–	31.5	90	100	100	97	100	100	–	26	36	94	–	38.5	58.5	52.5	=	<	>	>	1/1/2	2/2/6	1/3/6	1/4/6
4-lbcb^{6} vs. 7-lnca^{2}	–	12	0	0	32	0	0	0	40.5	18.5	7.5	7.5	4	0	0	0	38	10	3	0	47.5	0	0	0	◁	◀	◀	◀	1/0/5	6/0/6	6/0/6	6/0/6
4-lbcb^{6} vs. 8-lncb^{2}	–	12	0	0	32	0	0	0	–	18.5	7.5	7.5	4	0	0	0	–	10	3	0	–	0	0	0	◁	◀	◀	◀	1/0/2	6/0/6	6/0/6	6/0/6
4-lbcb^{6} vs. 9-gbfa^{7}	–	81	73.5	60	85	65	20	0	78.5	75.5	85.5	87	76	76.5	14	49	79	91	93	97	67	77.5	95.5	99.5	▷	▷	>	>	0/4/5	0/5/6	2/4/6	1/3/6
4-lbcb^{6} vs. 10-gbfb^{7}	–	56.5	27.5	10.5	58	47	10	0	50.5	53.5	71.5	75.5	62	66.5	14	49	50	64.5	77	72	53	62.5	81	93	>	>	=	=	0/0/5	0/0/6	2/2/6	2/2/6
4-lbcb^{6} vs. 15-gnca^{2}	–	12	0	0	32	0	0	0	40.5	19.5	9.5	9.5	4	0	0	0	38	8	3	0	47.5	15.5	0	0	◁	◀	◀	◀	1/0/5	6/0/6	6/0/6	6/0/6
4-lbcb^{6} vs. 17-lxfb	–	74	67	55	69	100	100	100	–	71.5	95	100	100	100	100	100	–	90	94	100	–	55	93.5	100	▷	▷	▷	▷	0/1/2	0/4/6	0/5/6	0/5/6
7-lnca^{2} vs. 8-lncb^{2}	–	50	50	50	50	50	50	50	–	50	50	50	50	40.5	40.5	40.5	–	50	50	50	–	50	50	50	=	<	<	<	0/0/2	0/0/6	0/0/6	0/0/6
7-lnca^{2} vs. 9-gbfa^{7}	–	100	100	100	88	100	100	100	81.5	94.5	99.5	100	100	100	100	100	86	100	100	100	71	100	100	100	▷	▶	▶	▶	0/4/5	0/6/6	0/6/6	0/6/6
7-lnca^{2} vs. 10-gbfb^{7}	–	89.5	100	100	70	100	100	100	60.5	83.5	95.5	95.5	100	100	100	100	63	95	100	100	55	100	100	100	▷	▶	▶	▶	0/1/5	0/6/6	0/6/6	0/6/6
7-lnca^{2} vs. 15-gnca^{2}	–	44.5	43	43	50	17.5	17.5	17.5	50	38	41	41	50	77.5	93.5	95.5	50	35	39	39	50	90.5	97	97	=	<	<	<	0/0/5	1/2/6	1/2/6	1/2/6
7-lnca^{2} vs. 17-lxfb	–	96.5	100	100	76	100	100	100	–	94.5	100	100	100	100	100	100	–	100	100	100	–	100	100	100	▶	▶	▶	▶	0/2/2	0/6/6	0/6/6	0/6/6
8-lncb^{2} vs. 9-gbfa^{7}	–	100	100	100	88	100	100	100	–	94.5	99.5	100	100	100	100	100	–	100	100	100	–	100	100	100	▶	▶	▶	▶	0/2/2	0/6/6	0/6/6	0/6/6
8-lncb^{2} vs. 10-gbfb^{7}	–	89.5	100	100	70	100	100	100	–	83.5	95.5	95.5	100	100	100	100	–	95	100	100	–	100	100	100	▷	▶	▶	▶	0/1/2	0/6/6	0/6/6	0/6/6
8-lncb^{2} vs. 15-gnca^{2}	–	44.5	43	43	50	17.5	17.5	17.5	–	38	41	41	50	85.5	100	100	–	35	39	39	–	90.5	97	97	=	<	<	<	0/0/2	1/2/6	1/2/6	1/2/6
8-lncb^{2} vs. 17-lxfb	–	96.5	100	100	76	100	100	100	–	94.5	100	100	100	100	100	100	–	100	100	100	–	100	100	100	▶	▶	▶	▶	0/2/2	0/6/6	0/6/6	0/6/6
9-gbfa^{7} vs. 10-gbfb^{7}	–	20	10	5	17.5	36	26.5	14	22.5	35.5	28.5	23.5	33.5	37	50	50	18.5	25.5	15.5	9.5	36	31	33	26.5	◁	◁	<	<	3/0/5	2/0/6	3/0/6	5/0/6

Table 15 continued

strategies	adult				covertype				credit-a				ionosphere				pima-indians				wisconsin				wins				≤ 27/≥ 73/valid			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
9-gbfa^{7} vs. 15-gnca^{2}	–	0	0	0	12	0	0	0	18.5	8.5	0.5	0	0	0	0	0	14	0	0	0	29	1	0	0	◁	◀	◀	◀	4/0/5	6/0/6	6/0/6	6/0/6
9-gbfa^{7} vs. 17-lxfb	–	22.5	55	60	19	100	100	100	–	44.5	94	100	100	100	100	100	–	36	54	65	–	30	49	68.5	=	<	>	▷	1/1/2	1/2/6	0/3/6	0/3/6
10-gbfb^{7} vs. 15-gnca^{2}	–	10.5	0	0	30	0	0	0	39.5	15.5	4.5	4.5	0	0	0	0	37	5	0	0	45	14	0	0	◁	◀	◀	◀	1/0/5	6/0/6	6/0/6	6/0/6
10-gbfb^{7} vs. 17-lxfb	–	63.5	94	100	63	100	100	100	–	62.5	97	100	100	100	100	100	–	70.5	87	95	–	40.5	65	82.5	▷	>	▷	▶	0/1/2	0/2/6	0/5/6	0/6/6
15-gnca^{2} vs. 17-lxfb	–	96.5	100	100	76	100	100	100	–	91	100	100	100	100	100	100	–	100	100	100	–	83	100	100	▶	▶	▶	▶	0/2/2	0/6/6	0/6/6	0/6/6

See Table 16 for a description of the content and used symbols

Table 16 Mann-Whitney U scores for the regression setting, using $|z\text{-}score|$

strategies	abelone				adult				auto-mpg				boston-housing				communities				forestfires				pima-indians				wins				≤27/≥73/valid			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1-lbfa vs. 2-lbfb	9.5	0	0	0	–	12	9.5	15	–	21.5	20.5	18	9.5	0	0	0	14	5.5	0	0	9.5	4.5	38	100	14	9.5	22.5	21	◀	◀	◁	<	5/0/5	7/0/7	6/0/7	6/1/7
1-lbfa vs. 3-lbca[7]	0	4.5	0	0	18	9.5	0	0	0	0	0	0	0	0	0	0	0	4.5	0	0	3.5	15	28	100	27.5	0	0	0	◁	◀	◁	<	6/0/7	7/0/7	6/0/7	6/1/7
1-lbfa vs. 4-lbcb[8]	0	0	0	0	–	9.5	9.5	9.5	–	0	0	0	0	0	0	0	0	2.5	0	0	9.5	0	0	100	13	0	0	0	◀	◀	◀	<	5/0/5	7/0/7	7/0/7	6/1/7
1-lbfa vs. 7-lnca[4]	0	0	0	0	–	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.5	0	0	0	◀	◁	◀	◀	6/0/6	6/0/7	7/0/7	7/0/7
1-lbfa vs. 8-lncb[4]	–	0	0	0	–	28	0	0	–	0	0	0	0	0	0	0	0	0	0	0	–	0	0	0	–	0	0	0	◀	◁	◀	◀	2/0/2	6/0/7	7/0/7	7/0/7
1-lbfa vs. 9-gbfa[7]	0	0	0	0	15	9.5	0	0	4.5	0	0	0	0	0	0	0	0	0	0	0	6.5	0	0	90	26	11	0	0	◀	◀	◀	<	7/0/7	7/0/7	7/0/7	6/1/7
1-lbfa vs. 10-gbfb[7]	0	0	0	0	–	9.5	0	0	–	0	0	0	0	0	0	0	0	0	0	0	6.5	0	0	80	13	8.5	0	0	◀	◀	◀	<	5/0/5	7/0/7	7/0/7	6/1/7
1-lbfa vs. 15-gnca[4]	0	0	0	0	–	28	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	7	0	0	0	◀	◁	◀	◀	6/0/6	6/0/7	7/0/7	7/0/7
2-lbfb vs. 3-lbca[7]	45	23	90	89	–	57	48	12	–	51	4	0	71	55	49	24	51	45.5	54	5	58.5	33.5	36	68	78.5	39.5	7	0	>	<	<	<	0/1/5	1/0/7	2/1/7	5/1/7
2-lbfb vs. 4-lbcb[8]	39	0	0	0	–	29	26.5	22.5	–	8	0	0	37.5	12	0	0	38.5	9	12	8	32	15.5	1	0	47	16.5	0	0	◁	◁	◀	◀	0/0/5	6/0/7	7/0/7	7/0/7
2-lbfb vs. 7-lnca[4]	39	0	0	0	–	36.5	25.5	0	–	0	0	0	26	0	0	0	35.5	0	0	0	31.5	0	0	0	52	0	0	0	<	◁	◀	◀	1/0/5	6/0/7	7/0/7	7/0/7
2-lbfb vs. 8-lncb[4]	–	0	0	0	–	36.5	25.5	0	–	0	0	0	19	0	0	0	35.5	0	0	0	–	0	0	0	–	0	0	0	◁	◁	◀	◀	1/0/2	6/0/7	7/0/7	7/0/7
2-lbfb vs. 9-gbfa[7]	45	10	100	91	–	48	13.5	0	–	32	0	0	72	48	22	25	54	0	3	0	53	16	0	0	80.5	61.5	0	0	>	<	<	<	0/1/5	3/0/7	6/1/7	6/1/7
2-lbfb vs. 10-gbfb[7]	35	10	71	91	–	23	8.5	0	–	20	0	0	34	34	15	17	39	0	0	0	36	16	0	10	47	25	0	0	◁	◁	<	<	0/0/5	6/0/7	6/0/7	6/1/7
2-lbfb vs. 15-gnca[4]	39	0	0	0	–	36.5	25.5	0	–	0	0	0	26	0	0	0	36	9	0	0	32	0	0	0	52	0	0	0	<	◁	◀	◀	1/0/5	6/0/7	7/0/7	7/0/7
3-lbca[7] vs. 4-lbcb[8]	45	6	0	0	–	29	26.5	57	–	9	0	0	22	7	0	0	34.5	16	12	16	26	28	6	0	20	16	0	0	◁	◁	◀	<	3/0/5	5/0/7	7/0/7	6/0/7
3-lbca[7] vs. 7-lnca[4]	45	0	0	0	–	36.5	25.5	7.5	28	0	0	0	19	0	0	0	32	0	0	0	23.5	0	0	0	13.5	0	0	0	◁	◁	◀	◀	3/0/6	6/0/7	7/0/7	7/0/7
3-lbca[7] vs. 8-lncb[4]	–	0	0	0	–	36.5	25.5	7.5	–	0	0	0	15	0	0	0	32	0	0	0	–	0	0	0	–	0	0	0	◁	◁	◀	◀	1/0/2	6/0/7	7/0/7	7/0/7
3-lbca[7] vs. 9-gbfa[7]	50	8.5	15	15.5	49.5	31.5	12.5	4	47	31	0	0	52	45	20	30	55.5	3.5	3	13	45	21	2	2	52.5	68	76	24	=	<	<	◁	0/0/7	3/0/7	6/1/7	6/0/7
3-lbca[7] vs. 10-gbfb[7]	34	8.5	9	12.5	–	17	8	2.5	–	19	0	0	23.5	37	14	20	38.5	0	0	0	31	21	2	10	19.5	27.5	16	4.5	◁	◁	◀	◀	2/0/5	5/0/7	7/0/7	7/0/7
3-lbca[7] vs. 15-gnca[4]	45	0	0	0	–	36.5	25.5	7.5	28	0	0	0	19	0	0	0	32	12	0	0	24	0	0	0	13.5	0	0	0	◁	◁	◀	◀	3/0/6	6/0/7	7/0/7	7/0/7

Table 16 continued

strategies	abelone				adult				auto-mpg				boston-housing				communities				forestfires				pima-indians				wins				≤27/≥73/valid			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
4-lbcb^8 vs. 7-lnca^4	48.5	2.5	0	0	–	39	35	5.5	–	0	0	0	32	0	0	0	45.5	0	0	0	49	0	0	0	56.5	6.5	0	0	<	◁	◁	◀	0/0/5	6/0/7	6/0/7	7/0/7
4-lbcb^8 vs. 8-lncb^4	–	2.5	0	0	–	37.5	35	5.5	–	0	0	0	25	0	0	0	45.5	0	0	0	–	0	0	0	–	6.5	0	0	◁	◁	◁	◀	1/0/2	6/0/7	6/0/7	7/0/7
4-lbcb^8 vs. 9-gbfa^7	55	19	100	100	–	69	43.5	4.5	–	79.5	82.5	84.5	78	87	100	100	67	49	61	68	69.5	28	70	59	81.5	85.5	100	100	▷	>	>	>	0/2/5	1/3/7	0/4/7	1/4/7
4-lbcb^8 vs. 10-gbfb^7	44.5	12	82	100	–	30	19	2.5	–	44.5	69.5	73.5	47.5	71	100	100	52	21	9	10	58.5	28	63	48	49.5	72	91	95	<	<	>	>	0/0/5	2/0/7	2/3/7	2/4/7
4-lbcb^8 vs. 15-gnca^4	48.5	15	4.5	2.5	–	39	35	5.5	–	0	0	0	32	3	0	0	46	31	0	0	50	0	0	0	56	6	0	0	<	◁	◁	◀	0/0/5	5/0/7	6/0/7	7/0/7
7-lnca^4 vs. 8-lncb^4	–	50	50	50	–	42.5	49.5	50	–	27	26	26	40.5	49.5	50	50	50	50	50	50	–	50	50	50	–	37.5	40	40	<	<	<	<	0/0/2	1/0/7	1/0/7	1/0/7
7-lnca^4 vs. 9-gbfa^7	55	92	100	100	–	63.5	58	63	68	100	100	100	81	100	100	100	69.5	100	100	100	73	99	100	100	88.5	100	100	100	▷	▷	▷	▷	0/3/6	0/6/7	0/6/7	0/6/7
7-lnca^4 vs. 10-gbfb^7	45	91	100	100	–	44.5	46	43.5	–	99	100	100	69	100	100	100	54.5	100	100	100	60	95	100	100	41.5	99	100	100	>	>	>	>	0/0/5	0/6/7	0/6/7	0/6/7
7-lnca^4 vs. 15-gnca^4	50	92.5	93.5	93.5	–	44	45.5	38	53	69.5	69.5	69.5	52.5	35	22.5	22.5	51	100	100	100	51	20.5	0	0	48.5	23	23	23	>	<	<	<	0/0/6	2/2/7	3/2/7	3/2/7
8-lncb^4 vs. 9-gbfa^7	–	92	100	100	–	63.5	58	63	–	100	100	100	85	100	100	100	69.5	100	100	100	–	99	100	100	–	100	100	100	▷	▷	▷	▷	0/1/2	0/6/7	0/6/7	0/6/7
8-lncb^4 vs. 10-gbfb^7	–	91	100	100	–	53.5	46	43.5	–	99	100	100	76	100	100	100	54.5	100	100	100	–	95	100	100	–	99	100	100	▷	▷	>	>	0/1/2	0/6/7	0/6/7	0/6/7
8-lncb^4 vs. 15-gnca^4	–	92.5	93.5	93.5	–	52	45.5	38	–	78	78	78	61	35	22.5	22.5	51	100	100	100	–	20.5	0	0	–	24.5	23.5	23.5	▷	>	<	<	0/0/2	2/3/7	3/3/7	3/3/7
9-gbfa^7 vs. 10-gbfb^7	34	29	10	23	–	25	23.5	19	–	27	18	6	22.5	36	38	27.5	34.5	17	6.5	3.5	34	45	38	42.5	18.5	20.5	12	7	◁	◁	◁	◁	2/0/5	4/0/7	5/0/7	5/0/7
9-gbfa^7 vs. 15-gnca^4	45	25	0	0	–	36.5	42	29	32	0	0	0	19	0	0	0	31	34	0	0	27	0	0	0	11.5	0	0	0	◁	◁	◁	◁	3/0/6	5/0/7	6/0/7	6/0/7
10-gbfb^7 vs. 15-gnca^4	55	31	4	0	–	49.5	54	42.5	–	1	0	0	31	0	0	0	46	62	16	0	40	0	0	0	58.5	1	0	0	<	<	<	◁	0/0/5	4/0/7	6/0/7	6/0/7

Scores below (above) 50 indicate that the left (right) strategy of a pair is better, 50 means the two are equal. Respectively, ◀, ◁, and <, under 'Wins', indicate that the left strategy is: better for all datasets, and all results are significant; better for all datasets, but not all results are significant; better overall, but not better for all datasets. Right-pointing triangles have equivalent meanings for the right strategy, and = means there is no 'winner'. Column '≤ 27 / ≥ 73 / valid' shows, for the left and right strategy respectively, how many U-scores are significant, and for how many datasets a top-10 is available

References

Atzmüller M (2015) Subgroup discovery. Wiley Interdiscip Rev Data Min Knowl Discov 5(1):35–49. https://doi.org/10.1002/widm.1144

Atzmüller M, Lemmerich F (2009) Fast subgroup discovery for continuous target concepts. In: Rauch J, Raś ZW, Berka P, Elomaa T (eds) ISMIS 2009, International symposium on methodologies for intelligent systems, Prague, Czech Republic, 14–17 September, 2009, Proceedings, LNCS, vol 5722. Springer, Berlin, pp 35–44. https://doi.org/10.1007/978-3-642-04125-9_7

Atzmüller M, Puppe F (2006) SD-map—a fast algorithm for exhaustive subgroup discovery. In: Fürnkranz J, Scheffer T, Spiliopoulou M (eds) PKDD 2006, European conference on principles and practice of knowledge discovery in databases, 18–22 Sept 2006, Proceedings, LNCS, vol 4213. Springer, Berlin, pp 6–17. https://doi.org/10.1007/11871637_6

Belfodil A [Aimene], Belfodil A, Kaytoue M (2018) Anytime subgroup discovery in numerical domains with guarantees. In: Berlingerio M, Bonchi F, Gärtner T, Hurley N, Ifrim G (eds) ECML PKDD 2018, European conference on machine learning and principles and practice of knowledge discovery in databases, Dublin, Ireland, 10–14 Sept 2018, proceedings, part II, LNCS, vol 11052. Springer, Cham, pp 500–516. https://doi.org/10.1007/978-3-030-10928-8_30

Boley M, Goldsmith BR, Ghiringhelli LM, Vreeken J (2017) Identifying consistent statements about numerical data with dispersion-corrected subgroup discovery. Data Min Knowl Discov 31(5):1391–1418. https://doi.org/10.1007/s10618-017-0520-3

Bosc G, Boulicaut J, Raïssi C, Kaytoue M (2018) Anytime discovery of a diverse set of patterns with Monte Carlo tree search. Data Min Knowl Discov 32(3):604–650. https://doi.org/10.1007/s10618-017-0547-5

Brin S, Motwani R, Ullman JD, Tsur S (1997) Dynamic itemset counting and implication rules for market basket data. In: Peckham JM, Ram S, Franklin M (eds) SIGMOD 1997, International conference on management of data, Tucson, Arizona, USA, 13–15 May 1997, Proceedings, ACM, New York, NY, pp 255–264. https://doi.org/10.1145/253260.253325

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30

Dougherty J, Kohavi R, Sahami M (1995) Supervised and unsupervised discretization of continuous features. In: Prieditis A, Russell SJ (eds) ICML 1995, International conference on machine learning, Tahoe City, CA, 9–12 July, 1995, Proceedings. Morgan Kaufmann, San Francisco, CA, pp 194–202. https://doi.org/10.1016/B978-1-55860-377-6.50032-3

Dua D, Graff C (2017) UCI machine learning repository. http://archive.ics.uci.edu/ml

Duivesteijn W (2013) Exceptional model mining. Ph.D. thesis, Leiden University, Leiden Institute of Advanced Computer Science. http://hdl.handle.net/1887/21760

Duivesteijn W, Meeng M (2016) SCHEP—a geometric quality measure for regression rule sets, gauging ranking consistency throughout the real-valued target space. In: Michaelis S, Piatkowski N, Stolpe M (eds) Solving large scale learning tasks. Challenges and algorithms—essays dedicated to Katharina Morik on the occasion of her 60th birthday, LNCS, vol 9580. Springer, pp 272–285. https://doi.org/10.1007/978-3-319-41706-6_14

Duivesteijn W, Knobbe A, Feelders A, van Leeuwen M (2010) Subgroup discovery meets Bayesian networks—an exceptional model mining approach. In: Webb GI, Liu B, Zhang C, Gunopulos D, Wu X (eds) ICDM 2010, IEEE international conference on data mining, Sydney, Australia, 14–17 Dec 2010, Proceedings, IEEE Computer Society, Los Alamitos, CA, pp 158–167. https://doi.org/10.1109/ICDM.2010.53

Duivesteijn W, Feelders A, Knobbe A (2012) Different slopes for different folks—mining for exceptional regression models with Cook's distance. In: Yang Q, Agarwal D, Pei J (eds) KDD 2012, ACM SIGKDD international conference on knowledge discovery and data mining, Beijing, China, 12–16 Aug 2012, Proceedings, ACM, New York, NY, pp 868–876. https://doi.org/10.1145/2339530.2339668

Fawcett T (2006) An introduction to ROC analysis. Pattern Recogn Lett 27(8):861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Fayyad UM, Irani KB (1993) Multi-interval discretization of continuous-valued attributes for classification learning. In: Bajcsy R (ed) IJCAI 1993, international joint conference on artificial intelligence, Chambéry, France, 28 Aug–3 Sept 1993, proceedings, part II. Morgan Kaufmann, San Francisco, CA, pp 1022–1029. http://ijcai.org/Proceedings/93-2/Papers/022.pdf

Frank E, Witten IH (1999) Making better use of global discretization. In: Bratko I, Džeroski S (eds) ICML 1999, International Conference on Machine Learning, Bled, Slovenia, 27–30 June, 1999, Proceedings, Morgan Kaufmann, San Francisco, CA, USA, pp 115–123, https://hdl.handle.net/10289/1507

Fürnkranz J, Flach PA (2005) ROC 'n' Rule learning—towards a better understanding of covering algorithms. Mach Learn 58(1):39–77. https://doi.org/10.1007/s10994-005-5011-x

Galbrun E, Miettinen P (2017) Redescription mining. Springer briefs in computer science. Springer, Berlin. https://doi.org/10.1007/978-3-319-72889-6

Grosskreutz H, Rüping S (2009) On subgroup discovery in numerical domains. Data Min Knowl Disco 19(2):210–226. https://doi.org/10.1007/s10618-009-0136-3

Grosskreutz H, Rüping S, Wrobel S (2008) Tight optimistic estimates for fast subgroup discovery. In: Daelemans W, Goethals B, Morik K (eds) ECML PKDD 2008, European conference on machine learning and principles and practice of knowledge discovery in databases, Antwerp, Belgium, 15–19 Sept 2008, proceedings, part I, LNCS, vol 5211. Springer, Berlin, pp 440–456. https://doi.org/10.1007/978-3-540-87479-9_47

Han J, Pei J, Yin Y (2000) Mining frequent patterns without candidate generation. In: Chen W, Naughton JF, Bernstein PA (eds) SIGMOD 2000, international conference on management of data, Dallas, TX, 16–18 May 2000, proceedings, ACM, New York, NY, pp 1–12. https://doi.org/10.1145/342009.335372

Herrera F, Carmona CJ, González P, del Jesús MJ (2011) An overview on subgroup discovery: foundations and applications. Knowl Inf Syst 29(3):495–525. https://doi.org/10.1007/s10115-010-0356-2

Ioannidis YE (2003) The history of histograms (abridged). In: Freytag JC, Lockemann PC, Abiteboul S, Carey MJ, Selinger PG, Heuer A (eds) VLDB 2003, international conference on very large data bases, Berlin, Germany, 9–12 Sept 2003, proceedings. Morgan Kaufmann, San Francisco, CA. http://www.vldb.org/conf/2003/papers/S02P01.pdf

Kavšek B, Lavrač N (2006) APRIORI-SD: adapting association rule learning to subgroup discovery. Appl Artif Intell 20(7):543–583. https://doi.org/10.1080/08839510600779688

Kaytoue M, Kuznetsov SO, Napoli A (2011) Revisiting numerical pattern mining with formal concept analysis. In: Walsh T (ed) IJCAI 2011, international joint conference on artificial intelligence, Barcelona, Catalonia, Spain, 16–22 July 2011, proceedings, IJCAI/AAAI, pp 1342–1347. https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-227

Klösgen W (1992) Problems for knowledge discovery in databases and their treatment in the statistics interpreter EXPLORA. Int J Intell Syst 7(7):649–673. https://doi.org/10.1002/int.4550070707

Klösgen W (1996) EXPLORA: a multipattern and multistrategy discovery assistant. In: Fayyad UM, Piatetsky-Shapiro G, Smyth P, Uthurusamy R (eds) Advances in knowledge discovery and data mining. American Association for Artificial Intelligence (AAAI/MIT Press), Menlo Park, pp 249–271

Klösgen W (1999) Applications and research problems of subgroup mining. In: Raś ZW, Skowron A (eds) ISMIS 1999, international symposium on methodologies for intelligent systems, Warsaw, Poland, 8–11 June 1999, proceedings, LNCS, vol 1609. Springer, Berlin, pp 1–15. https://doi.org/10.1007/BFb0095086

Knobbe A, Ho EKY (2006a) Maximally informative *k*-itemsets and their efficient discovery. In: Eliassi-Rad T, Ungar LH, Craven M, Gunopulos D (eds) KDD 2006, ACM SIGKDD international conference on knowledge discovery and data mining, Philadelphia, PA, USA, 20–23 Aug 2006, proceedings. ACM, New York, NY, pp 237–244. https://doi.org/10.1145/1150402.1150431

Knobbe A, Ho EKY (2006b) Pattern teams. In: Fürnkranz J, Scheffer T, Spiliopoulou M (eds) PKDD 2006, european conference on principles and practice of knowledge discovery in databases, Berlin, Germany, 18–22 Sept 2006, proceedings, LNCS, vol 4213. Springer, pp 577–584. https://doi.org/10.1007/11871637_58

Konijn RM, Duivesteijn W, Kowalczyk W, Knobbe A (2013) Discovering local subgroups, with an application to fraud detection. In: Pei J, Tseng VS, Cao L, Motoda H, Xu G (eds) PAKDD 2013, Pacific-Asia conference on knowledge discovery and data mining, Gold Coast, Australia, 14–17 Apr 2013, proceedings, part I, LNCS, vol 7818. Springer, Berlin, pp 1–12. https://doi.org/10.1007/978-3-642-37453-1_1

Konijn RM, Duivesteijn W, Meeng M, Knobbe A (2015) Cost-based quality measures in subgroup discovery. J Intell Inf Syst 45(3):337–355. https://doi.org/10.1007/s10844-014-0313-8

Kontkanen P, Myllymäki P (2007) MDL histogram density estimation. In: Meila M, Shen X (eds) AISTATS 2007, international conference on artificial intelligence and statistics, San Juan, Puerto Rico, 21–24

March 2007, Proceedings, Part II, PMLR. Proceedings of Machine Learning Research, pp 219–226. http://proceedings.mlr.press/v2/kontkanen07a/kontkanen07a.pdf

Lavrač N, Gamberger D (2004) Relevancy in constraint-based subgroup discovery. In: Boulicaut J, Raedt LD, Mannila H (eds) Constraint-based mining and inductive databases, European workshop on inductive databases and constraint based mining, Hinterzarten, Germany, 11–13 March 2004, Revised Selected Papers, LNCS, vol 3848. Springer, Berlin, pp 243–266. https://doi.org/10.1007/11615576_12

Lavrač N, Flach PA, Zupan B (1999) Rule evaluation measures: a unifying view. In: Džeroski S, Flach PA (eds) ILP-99, inductive logic programming, Bled, Slovenia, 24–27 June 1999, Proceedings, LNCS, vol 1634. Springer, Berlin, pp 174–185. https://doi.org/10.1007/3-540-48751-4_17

Lavrač N, Kavšek B, Flach PA, Todorovski L (2004) Subgroup discovery with CN2-SD. J Mach Learn Res 5:153–188

Lemmerich F, Becker M, Atzmüller M (2012) Generic pattern trees for exhaustive exceptional model mining. In: Flach PA, De Bie T, Cristianini N (eds) ECML PKDD 2012, European conference on machine learning and principles and practice of knowledge discovery in databases, Bristol, UK, 24–28 Sept 2012, proceedings, part II, LNCS, vol 7524. Springer, Berlin, pp 277–292. https://doi.org/10.1007/978-3-642-33486-3_18

Lemmerich F, Becker M, Puppe F (2013) Difference-based estimates for generalization-aware subgroup discovery. In: Blockeel H, Kersting K, Nijssen S, Železný F (eds) ECML PKDD 2013, European conference on machine learning and principles and practice of knowledge discovery in databases, Prague, Czech Republic, 23–27 Sept 2013, proceedings, part III, LNCS, vol 8190. Springer, Berlin, pp 288–303. https://doi.org/10.1007/978-3-642-40994-3_19

Lemmerich F, Atzmüller M, Puppe F (2016) Fast exhaustive subgroup discovery with numerical target concepts. Data Min Knowl Discov 30(3):711–762. https://doi.org/10.1007/s10618-015-0436-8

Lowerre BT (1976) The Harpy speech recognition system. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA

Mampaey M, Nijssen S, Feelders A, Knobbe A (2012) Efficient algorithms for finding richer subgroup descriptions in numeric and nominal data. In: Zaki MJ, Siebes A, Yu JX, Goethals B, Webb GI, Wu X (eds) ICDM 2012, IEEE international conference on data mining, Brussels, Belgium, 10–13 Dec 2012, proceedings. IEEE Computer Society, Los Alamitos, CA, USA, pp 499–508. https://doi.org/10.1109/ICDM.2012.117

Mampaey M, Nijssen S, Feelders A, Konijn RM, Knobbe A (2015) Efficient algorithms for finding optimal binary features in numeric and nominal labeled data. Knowl Inf Syst 42(2):465–492. https://doi.org/10.1007/s10115-013-0714-y

Mann HB, Whitney DR (1947) On a test of whether one of two random variables is stochastically larger than the other. Ann Math Stat 18(1):50–60. https://doi.org/10.1214/aoms/1177730491

Meeng M, Knobbe A (2011) Flexible enrichment with Cortana—software demo. In: van der Putten P, Veenman C, Vanschoren J, Israel M, Blockeel H (eds) Benelearn 2011, Belgian Dutch conference on machine learning, The Hague, the Netherlands, 20 May 2011, proceedings, pp 117–119

Meeng M, Knobbe A (2020) Uni- and multivariate probability density models for numeric subgroup discovery. Intell Data Anal 24(6)

Meeng M, Duivesteijn W, Knobbe A (2014) ROCsearch—an ROC-guided search strategy for subgroup discovery. In: Zaki MJ, Obradovic Z, Tan P, Banerjee A, Kamath C, Parthasarathy S (eds) SDM 2014, international conference on data mining, Philadelphia, PA, USA, 24–26 April 2014, proceedings. SIAM, pp 704–712. https://doi.org/10.1137/1.9781611973440.81

Nguyen H, Müller E, Vreeken J, Böhm K (2014) Unsupervised interaction-preserving discretization of multivariate data. Data Min Knowl Discov 28(5–6):1366–1397. https://doi.org/10.1007/s10618-014-0350-5

Pieters BFI, Knobbe A, Džeroski S (2010) Subgroup discovery in ranked data, with an application to gene set enrichment. In: PL-10, preference learning workshop at ECML PKDD 2010, European conference on machine learning and principles and practice of knowledge discovery in databases, Barcelona, Spain, 20–24 Sept 2010. http://www.ke.tu-darmstadt.de/events/PL-10/papers/7-Pieters.pdf

van Leeuwen M, Knobbe A (2011) Non-redundant subgroup discovery in large and complex data. In: Gunopulos D, Hofmann T, Malerba D, Vazirgiannis M (eds) ECML PKDD 2011, European conference on machine learning and principles and practice of knowledge discovery in databases, Athens, Greece, 5–9 Sept 2011, proceedings, part III, LNCS, vol 6913. Springer, Berlin, pp 459–474. https://doi.org/10.1007/978-3-642-23808-6_30

van Leeuwen M, Knobbe A (2012) Diverse subgroup set discovery. Data Min Knowl Disco 25(2):208–242. https://doi.org/10.1007/s10618-012-0273-y

Wrobel S (1997) An algorithm for multi-relational discovery of subgroups. In: Komorowski HJ, Zytkow JM (eds) PKDD 1997, principles of data mining and knowledge discovery, first European symposium, Trondheim, Norway, 24–27 June 1997, proceedings, LNCS, vol 1263. Springer, pp 78–87. https://doi.org/10.1007/3-540-63223-9_108

Affiliations

Marvin Meeng[1] · **Arno Knobbe**[1]

Arno Knobbe
knobbe@liacs.nl

[1] LIACS, Leiden University, Leiden, The Netherlands