Non-Redundant Subgroup Discovery Using a Closure System

Paper for Subgroup Discovery Seminar 2017

Magnus Halbe

Abstract

Subgroup discovery finds descriptions of subsets of data which are notable due to their behavior in a target variable. A lot of the theoretical possible descriptions are redundant, bloating search space and run time. The contribution considers each set of descriptions which describe the same subgroup an equivalence class, and all classes' highest cardinality representatives form a closure system on the dataset. Two algorithms are presented to find the minimum cardinality representatives of each such class: One is based on a greedy problem solver of **NP**-hard problem `MIN-SET-COVER`, while the other provides no additional cost during enumeration by utilizing a graph-based approach. Finally it is shown that actually utilizing the reduced description search space of the minimum representatives on real-world datasets is usually orders of magnitudes faster than a traditional method.

1 Introduction

Consider the problem of Subgroup Discovery [see Atzmueller, 2015]: Find a subset of the data (or population) which behaves notably different in a given real-valued target variable. These subset are not presented as a mere collection of individuals, but instead as human-friendly descriptions.

Each description is a conjunction of basic propositions, and only those individuals which fulfill all of them are part of the subgroup. When one imagines all possible constraints and all their potential conjunctions, one ends up with an enormous search space. For each categorical feature you have one constraint for each category (e.g. $sex = Male, sex = female$), and it is even worse for numerical data (e.g. $height \leq 180, height > 180, \dots$). Considering all of these descriptions could lead to an exponential runtime.

Boley and Grosskreutz [2009] propose an efficient alternative to the entire search space. All those descriptions which describe (or more formally: *extend* to) the same subgroup on the given data are considered members of the same *equivalence class*. This is practical as all of these descriptions then also have one equivalent result on the target variable.[1]

[1]Note that the target variable or function (in some literature [Webb, 2001] *interestingness measure*)

Each equivalence class is uniquely defined by its highest cardinality description, i.e. the *maximal representative*. Due to both the general consensus that shorter descriptions are easier to understand and a higher usefulness for generalization we instead **search for all the lowest cardinality descriptions** — accordingly the *minimum representatives*. Note that an equivalence class can have more than one of them.

After it is established that the search of such a minimum representative given any description is **NP**-hard, the logical step of applying a well-known greedy strategy for that **NP**-hard problem is applied in a first solution. The greediness leads to possibly non-optimal solutions.

An elaborate alternative is therefore presented which traverses over a graph with equivalence classes as nodes and class-dividing constraints as edges. The minimum representatives are then exactly the shortest paths to each equivalence class. To avoid wasting computation time on considering all the path's permutations, a canonical constraint order is introduced and thus the concept of *canonical minimum representatives* can be established.

Finally, Boley and Grosskreutz [2009] show that utilizing the resulting search space of only minimum representatives performs is usually an order of magnitudes faster with experimental evidence on ten real-world data sets. These results are not surprising as all possible extensions are still being considered by searching for only one member for each equivalence class.

The discarding of whole equivalence classes is also discussed, as the presented methods are also scrutinized according to the typical variants of subgroup discovery: Either only give the *top-k* quality classes or mining only classes whose representatives do not go over a given *length-limit*. While the latter is a natural addition to the path-finding solution, it does not combine well with the greediness-induced shortcomings of the **NP**-solver solution.

The given prior approaches for cutting down the description search space are interesting as most of them can be combined with this contribution. Removal of irrelevant descriptions, successive weighted covering and the application of fast, optimistic estimators can for instance be applied on the selected representative descriptions instead of the whole space. However, if each equivalence class contains only one description the presented methods focus might be unnecessary, but the real-world results show that this is not a practical concern.

2 Preliminaries

We consider the complete dataset a *population*, with each data record an *individual*.

Let log denote the binary logarithm.

While not the focus of the contribution, *descriptive variables* $\{x_1, \ldots, x_m\}$ are then such that each X_j is formally regarded as a mapping $x_j : P \to X_j$ that assign to each individual from an underlying population P some value from a domain X_j.

and its results are not considered for the task at hand. Check the aforementioned Atzmueller [2015] for a thorough overview.

#	2^4	2^3	2^1	2^0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
		⋮		

Figure 1. Binary example for $\mathcal{D}_n, n = 4$, can be extended to any n and fitting $\log(n)$ attributes.

More relevant for us is the description language $\mathcal{L}_X \subseteq \{\bot, \top\}$. A description is of the form $\Pi_X = \bigcup_{j\in[m]}\{x(i) = v_l\}$ with $x_j : P \to \{v_1, \ldots, v_s\}$. This results in selectors of the form

$$\sigma(i) = (\pi_{i_1}(i) = v_{j_1}) \wedge \cdots \wedge (\pi_{i_j}(i) = v_{j_l}) \tag{1}$$

The extension for a description is then exactly the set of individuals which fulfill these selectors: $\text{ext}(\sigma) = \{i \in P : \sigma(i) = \top\}$. To give some intuition, this is the subgroup of the population which fits the description.

Note that adding constraints to a description is anti-monotone to the extensions, that is $\Pi_X \subseteq \Pi'_X \implies \text{ext}(\sigma) \supseteq \text{ext}(\sigma')$. To overcome the first glance confusion of notationally unconnected constraint set and selector, we abuse our notation a bit by identifying σ with the set of constraints $\pi_{i_1}, \pi_{i_2}, \ldots \pi_{i_l}$

We are now aware of the concept of descriptions and their extensions. This allows us to present the **equivalence class**: Two descriptions σ, σ' are *equivalent* on a dataset $\mathcal{D}$, denoted by $\sigma \equiv \sigma'$ if $\text{ext}(\sigma) = \text{ext}(\sigma')$, that is they refer to the same subgroup of individuals. This equivalence class is then $[\sigma]$, i.e. $\{\sigma, \sigma'\} \subseteq [\sigma] = [\sigma']$. Each description is therefore a representative of its equivalence class.

The border case descriptions are the most interesting ones for the contribution. There are actually two applied orders on the representatives. One can consider the space from *minimum* to *maximum* representative. This refers to the cardinalities of the constraint sets — all minimum representatives of an equivalence class thus have the same number of constraints. Consider instead an order from *minimal* to *maximal* representatives: A representative is minimal if removing any constraint from it would change the extension.

The maximal representative is also the maximum representative — it combines all relevant constraints and is thus unique. One can have however i minimal representatives at once and j minimum representatives at once (possibly with $i \neq j, i > 1, j > 1$).

For a quick intuition let the population $\mathcal{D}_n$ be a the bit representations of the integers up to n, with $\log(n)$ boolean attributes. See the case $n = 4$ in fig. 1. For the extension which contains only the integer 4 of $\mathcal{D}_n$ there are two minimum representatives ($\# = 4$) and ($2^3 = 1$). While these are also minimal representatives, there is an additional minimal representative ($2^4 = 0 \wedge 2^1 = 1 \wedge 2^0 = 0$). The maximum and maximal constraint is

$$(\# = 4 \wedge 2^4 = 0 \wedge 2^3 = 0 \wedge 2^1 = 1 \wedge 2^0 = 0)$$

As we have familiarized ourselves with cardinalities of border cases in the equivalence class, it is now a good opportunity to recognize how big an equivalence class itself can be. Consider any population $\mathcal{D}_n$ as in fig. 1. Now add an additional individual z and let all attributes be 2. As the rest of the population uses only binary values, z will stand out notably. In fact, the equivalence class which extension contains only z has $2^{\log(n)}$ elements — any combination of attribute constraints which check for equality with 2 will work. Note how this is an exponential size.

Consider an operator[2] $\mathrm{cl}(\sigma)$ which results precisely in the unique maximal element of an equivalence class $[\sigma]$.

Lemma 1. ***[Pasquier et al., 1999]*** *The map* cl *is a closure operator which satisfies for all descriptions* σ, σ'

extensivity $\sigma \subseteq \mathrm{cl}(\sigma)$

monotonicity $\sigma \subseteq \sigma' \implies \mathrm{cl}(\sigma) \subseteq \mathrm{cl}(\sigma')$

idempotence $\mathrm{cl}(\sigma) = \mathrm{cl}(\mathrm{cl}(\sigma))$

Thus the population with cl-operator forms a closure system. There exist reasonably efficient algorithms which find all closed sets given a closure system.

3 Contribution

The contribution focuses on *the search of minimum representatives*, as the smallest cardinality descriptions are believed to be easier to understand for humans and result in better building blocks for generalizations [Boley and Grosskreutz, 2009]. In order to achieve this, it would be useful to output an equivalent minimum representative given any description.

For which descriptions do we request the minimum representative? When have we found them all?

Recall the closure operator cl and that there exist reasonably efficient algorithms which find all closed sets given a closure system. As each closed set refers to one equivalence class, all that is left to be done is running an algorithm repeatedly to retrieve all minimum representatives.

Theorem 1. *Given a subgroup description* σ*, it is* ***NP****-hard to compute an equivalent minimum description* σ'*, i.e.* $\sigma' \in [\sigma]$*.*

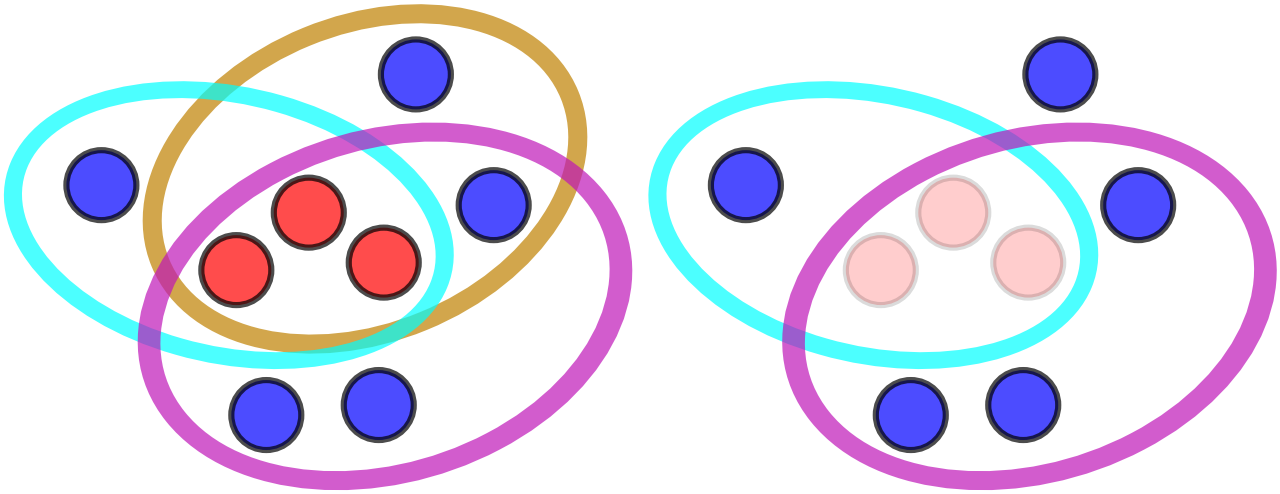
Theorem 1 is proven by reducing from `MIN-SET-COVER`. The required reduction is of great importance for us. Reducing from it to our problem proofs the hardness. Reducing in the other direction allows us to apply well-known greedy approximation algorithm.

Recall that `MIN-SET-COVER` receives an universe of values U and a set of subsets of that universe S and is tasked to find the smallest subset of S which fully covers U.

[2] In Pasquier et al. [1999] this operator is symbolized through σ which would overload that letter in our notation. We denote it as Latin letter acronym, analogue to $\mathrm{ext}(\sigma)$ which also behaves consistently for each member of an equivalence class.

Given a description σ we ask for a covering not on individual in the inverse of $\mathrm{ext}(\sigma)$, with the smallest amount of constraints from $\mathrm{cl}(\sigma)$. This is what algorithm 1 does: The problem is reduced to `MIN-SET-COVER` and solved greedily — note that due to shortcomings of the greedy approach these may not be minimum.

For an intuition consider fig. 2(a). The dots are the population, while only red dots are in the extension of a given σ. The ellipses present the single constraints/their extensions. Now finding the smallest subset of $\mathrm{cl}(\sigma)$ which extension does not include any blue dots, while ignoring the red dots, is the same as finding the minimum representative of $[\sigma]$. See a solution in fig. 2(b).



(a) A population (dots) with three constraints and their extensions (ellipses).

(b) Combining two red-covering constraints to cover none of the blue dots.

Figure 2

Input: description σ
Output: equivalent approximate minimum description σ'
$\bar{D} \leftarrow D \setminus \mathrm{ext}(\sigma)$
$\sigma' \leftarrow \emptyset$
while $\mathrm{ext}_{\bar{D}}(\sigma') \neq \emptyset$ **do**
  $\sigma' \leftarrow \sigma' \cup \{\arg\min_{c \in \mathrm{cl}(\sigma)} |\mathrm{ext}_{\bar{D}}(\sigma' \cup \{c\})\}$
end
return σ'

Algorithm 1: Greedy solver

Note that the reverse reduction similarly asks for a "non-covering". Each set s in S becomes an individual with an binary attribute for each of the elements u in U, which is 1 if and only if $u \notin s$. An additional individual whose attributes are all 1 is added. A minimum representative's constraint set of the description in which at least a single attribute is not 1 is then easily converted into a min. set cover.

Through the closure operator cl we get a representative of every equivalence class, on which we apply the greedy algorithm. To get the top-k results the descriptions are simply collected along with their result on the target variable in a k-length priority queue. Adjusting for *length limit* is not straightforward due to the greediness and its margin of error. See how the arg min choice without any backtracking in algorithm 1 might make mistakes[3].

A considerable focus is put on the constraint length in algorithm 2. The search space traverses over a graph with equivalence classes as vertices and class-dividing constraints as edges. Minimum representatives are then exactly the shortest paths to each equivalence class, which can all be found through breadth-first search.

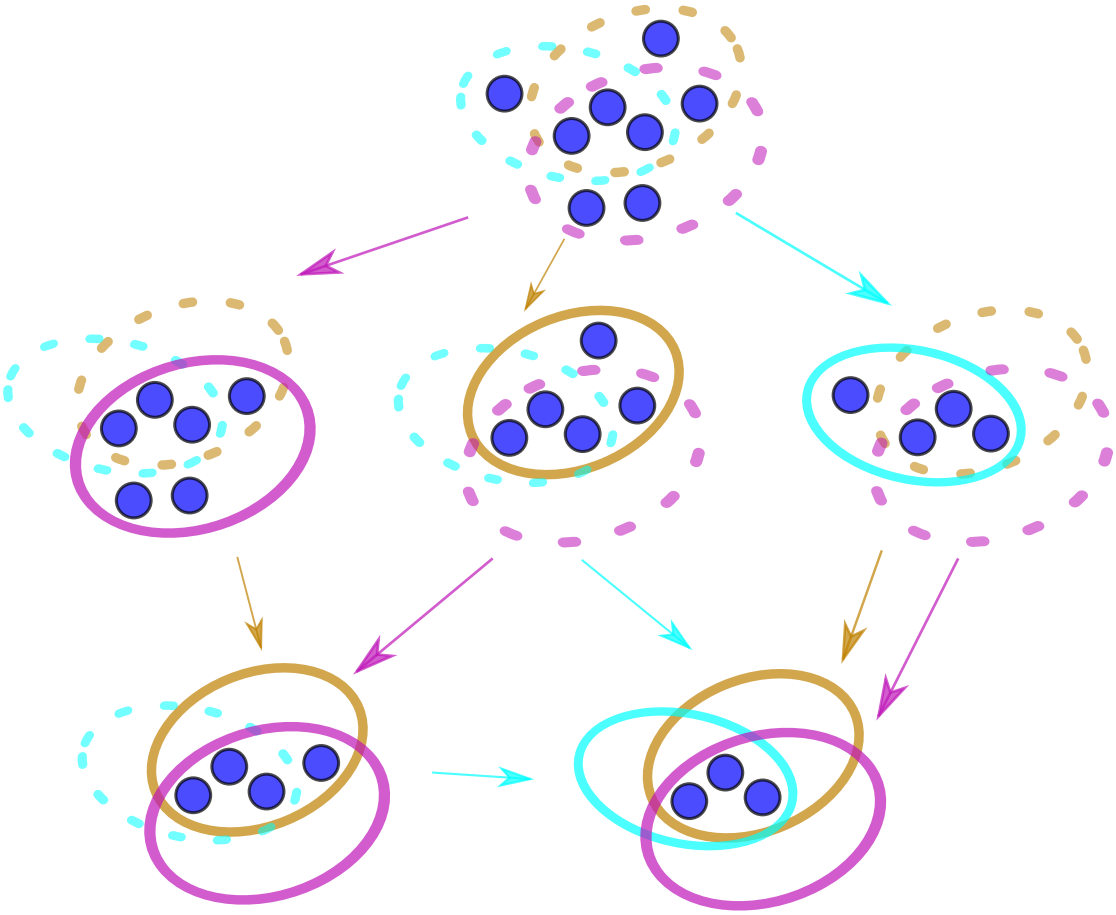


Figure 3. graph with *equivalence classes as vertices* and *class-dividing constraints as edges*. Each equivalence class is represented by its individuals (blue dots). Each constraint is represented by an ellipse, which is dotted if the constraint is not applied.

Refer to Fig. 3 for a representation. Note how the path <light blue, yellow> is shorter than <purple, yellow, light blue>, which translates to the idea that $\{light\ blue, yellow\}$

[3]Refer to Boley and Grosskreutz [2009] to learn the guaranteed error bounds of the algorithm.

is a minimum representative of the bottom-right extension which contains exactly three blue dots.

Input: ordered ground set of constraints $\mathbf{C} = c_1, \ldots, c_n$
extension closure operator cl
quality function q with optimistic estimator $\hat{q}$ and threshold q^*
Output: family $\{\mu(H) : q(H) \geq q^*\}$ in lexicographical order
init $\mathcal{Q}$ as empty queue and $\mathcal{V}$ as empty prefix tree
enqueue $(\emptyset, \mathrm{cl}(\emptyset), \mathbf{C})$ on $\mathcal{Q}$ **while** $\mathcal{Q} \neq \emptyset$ **do**
  dequeue front element $(\sigma, S, \mathbf{A})$ of $\mathcal{Q}$
  if $q(S) \geq q^*$ **then**
    print $\mathcal{Q}$
  end
  $\mathbf{A}' \leftarrow c \in \mathbf{A}\ S : c < \min \sigma, \hat{q}(\sigma \cup \{c\}) \geq q^*$
  foreach $c_i \in \boldsymbol{A}'$ *in ascending order of their index* **do**
    $\sigma' \leftarrow \sigma \cup \{c_i\}$
    $S' \leftarrow \mathrm{cl}(G')$
    if $S' \notin \mathcal{V}$ **then**
      add S' *to* $\mathcal{V}$
    end
    enqueue $(\sigma', S', \mathbf{A}')$ on $\mathcal{Q}$
  end
end
return σ'

Algorithm 2: Inductive Minimum Representative Construction

This method is afflicted by two problems. Firstly, all possible permutations are considered. To get to the bottom-left equivalence class of descriptions extending to four individuals the paths <purple, yellow> and <yellow, purple> are viable. The permutations are always equivalent for our purpose, thus one result would suffice. This can easily be achieved by introducing an arbitrary but fixed canonical edge order, and thus only finding *canonical minimum representatives*.

Secondly, the algorithm can run out of memory. Let the population be the already shown bit representations (fig. 1) of the integers up to $2m$, with m boolean attributes. Now each individual is an unique combination of attributes, and all possible combinations are present — which means that adding a constraint to σ must change its extension. Inductively, each equivalence class has exactly one member. If this is coupled with unfortunate target variable values, the breadth-first search algorithm has to branch everywhere every step and keep track of all the extensions it visited, possibly running out of memory.

The method can easily be changed to the length limit variant by restricting the breadth-first search. Adapting to a top-k variant is done analogously to the greedy method.

4 Evaluation

We have noticed that the theoretical benefits are enormous. One target variable evaluation for a minimum representative might provide the same findings as a evaluation on a (compared to the population) exponentially sized set of descriptions, given that they all are in the same equivalence class.

The theoretical downsides are negligible: as long as the algorithm converges it did not miss an equivalence class, and thus is unable to miss anything. This is provided by the theoretic foundation of the closure operator system.

How well does it work in practice?

In the empirical evaluation provided by Boley and Grosskreutz [2009] the greedy and the graph approach are compared against a traditional algorithm, that is Dsubgroup [Grosskreutz et al., 2008]. That one is often taking over twelve hours — and is thus beaten through reduced search space benefits. The graph approach seems to be done either extremely fast — often terminating in seconds — or never due to running out of memory. Given a quality threshold however, Dsubgroup beats the graph approach on smaller search spaces. Greedy is more reliable than the graph approach as it usually is within time and memory limits, while rarely performs best. It is much closer in needed time to the other closed approach than Dsubgroup.

Another comparison is on classification results, evaluated through the found AUC. They compared top-20 subgroup descriptions against top-20 minimum representatives against a rule learner RIPPER [Cohen, 1995]. While the latter occasionally beats both of them, it was always more successful to apply the top minimum representatives instead of the top subgroup descriptions.

Grosskreutz and Paurat [2011] use the contribution (graph-based, dubbed "CloSD") in their experimental evaluation section. Their approach is notably different — subgroups dominated by other subgroups are discarded, while we instead do not discard subgroups, but only equivalent descriptions. The third presented algorithm is again Dsubgroup. They measure how many nodes are considered before finding the actual top-k subgroups. Our approach scores closely to theirs, either before or behind it. Dsubgroup behaves far differently, both in good and in bad ways.

From the experiments we conclude that indeed the contribution is rarely worse and often orders of magnitudes faster. This fits theoretical considerations. Running out of memory is a real concern for the graph approach, effectively excluding what might otherwise be the better method from too big databases.

5 Conclusion

Boley and Grosskreutz [2009] present the concept of equivalence classes over descriptions. It is shown that each of these are uniquely defined by one highest cardinality description, i.e. the *maximal representative*. Due to both the general consensus that shorter descriptions are easier to understand and a higher usefulness for generalization we instead search for all the lowest cardinality descriptions — accordingly the *minimum*

representatives.

After it is proven that the search of such a minimum representative given any description is **NP**-hard, the logical step of applying a well-known greedy strategy for that very **NP**-hard problem is applied in the first solution. An elaborate alternative is presented which traverses over a graph with equivalence classes as nodes and class-dividing constraints as edges. The minimum representatives are then exactly the shortest paths to each equivalence class. To avoid wasting computation time on considering all the path's permutations, a canonical constraint order is introduced and thus the concept of *canonical minimum representatives* can be established.

The experimental results prove that the run-time for cutting down the amount of descriptions is well-worth the effort. This is not contradicted by independent findings. However, the graph-based implementation runs out of memory on occasion, rendering one method occasionally useless in practice.

In theory the closure system guarantees that each equivalence class is considered — which means by extension every possibly describable subgroup is examined. Ergo the resulting search space can not miss a subgroup.

A theoretical shortcoming is the worst-case population. Let the population be the already shown bit representations D_{2^n} (fig. 1) which means that adding a constraint to σ changes its extension. A set of minimum representatives is therefore as useful as the set of regular descriptions. Of course, this exact case is unlikely with that few attributes, yet on modern day big data with an extremely high number of attributes and individuals it could well be that the descriptions are not as redundant. Keep in mind that in theory the contribution's pessimistic amount of extra work is much smaller than the optimistic amount of saved work.

Boley and Grosskreutz remark that future work should be directed at adapting the approach for ordinal constraints. Not only would this be practical as ordinal real-world data sets are common, the equivalence classes should prove even more useful as there are more redundant descriptions.

References

Martin Atzmueller. Subgroup discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(1):35–49, 2015.

Mario Boley and Henrik Grosskreutz. Non-redundant subgroup discovery using a closure system. *Machine Learning and Knowledge Discovery in Databases*, pages 179–194, 2009.

William W Cohen. Fast effective rule induction. In *Proceedings of the twelfth international conference on machine learning*, pages 115–123, 1995.

Henrik Grosskreutz and Daniel Paurat. Fast and memory-efficient discovery of the top-k relevant subgroups in a reduced candidate space. *Machine Learning and Knowledge Discovery in Databases*, pages 533–548, 2011.

Henrik Grosskreutz, Stefan Rüping, and Stefan Wrobel. Tight optimistic estimates for fast subgroup discovery. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 440–456. Springer, 2008.

Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Efficient mining of association rules using closed itemset lattices. *Information systems*, 24(1):25–46, 1999.

Geoffrey I Webb. Discovering associations with numeric variables. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 383–388. ACM, 2001.