Artificial Intelligence 239 (2016) 54–69

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Extracting qualitative relations from categorical data

Jure Žabkar [*], Ivan Bratko, Janez Demšar

Faculty of Computer and Information Science, University of Ljubljana, Večna pot 113, 1000 Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 4 September 2014
Received in revised form 23 June 2016
Accepted 28 June 2016
Available online 5 July 2016

Keywords:
Qualitative modeling
Machine learning
Ceteris paribus

ABSTRACT

Qualitative modeling is traditionally concerned with the abstraction of numerical data. In numerical domains, partial derivatives describe the relation between the independent and dependent variable; qualitatively, they tell us the trend of the dependent variable. In this paper, we address the problem of extracting qualitative relations in categorical domains. We generalize the notion of partial derivative by defining the probabilistic discrete qualitative partial derivative (PDQ PD). PDQ PD is a qualitative relation between the target class c and the discrete attribute; the derivative corresponds to ordering the attribute's values, a_i, by $P(c|a_i)$ in a local neighborhood of the reference point, respecting the ceteris paribus principle. We present an algorithm for computation of PDQ PD from labeled attribute-based training data. Machine learning algorithms can then be used to induce models that explain the influence of the attribute's values on the target class in different subspaces of the attribute space.

1. Introduction

A data set collected at the Institute of Oncology, Ljubljana, Slovenia, contains data on 1220 patients with breast cancer. Each patient was given one of the two possible chemotherapy treatments: CMF[1] or anthracycline-based chemotherapy. No treatment is generally superior to another; the physician's role is to determine the optimal treatment for each particular patient [1]. The data contains demographic and clinical data, the chosen treatment and whether the cancer recurred in a certain period. How do we induce a useful model from such data? In a more general case, the physician may need to make several (dependent or independent) decisions, and there may be multiple objectives – like the patient's survival, comfort and side-effects – to optimize. Which algorithms are suitable for analyzing the data collected in such studies?

The problem resembles a number of different tasks. Model induction belongs to the field of classical machine learning; however, here we deal with one or more categorical or numeric target variables, and with attributes at and beyond our control, which need to be treated differently. From another perspective, the problem is superficially similar to preference learning, yet the preferences are not explicitly given but need to be extracted from the data. The resulting model may be represented as a CP-net, but the task here is to derive the preferences rather than the structure of the network. We will explore the relation with other methods at appropriate points later in the paper. The methods we will present are also not limited to induction of (clinical or general) decision support systems like the one described above.

Our venture point is qualitative modeling. Qualitative predictive models in their original form are a substitute for regression models. Instead of predicting the numerical value of the dependent variable, they only predict the direction of its

[*] Corresponding author.
E-mail addresses: jure.zabkar@fri.uni-lj.si (J. Žabkar), ivan.bratko@fri.uni-lj.si (I. Bratko), janez.demsar@fri.uni-lj.si (J. Demšar).

[1] Cyclophosphamide, methotrexate, and fluorouracil.

http://dx.doi.org/10.1016/j.artint.2016.06.007
0004-3702/© 2016 Elsevier B.V. All rights reserved.

change with respect to the change(s) of independent variable(s). They are particularly useful for complex problems in, for instance, economy [2], where exact numerical relations may be unattainable, while qualitative models may still correctly describe, for example, the conditions under which a decrease of interest rates will stimulate economic growth and how will this affect the unemployment rate. Kupiers [3] argued that qualitative models should be used for reasoning about continuous phenomena with incomplete or unreliable knowledge. Despite the popularity of qualitative models in some branches of science, most notably economy, they gained but a little traction in machine learning, resulting in only a few algorithms for their automated induction from data [4,5]. Most qualitative models are still derived manually.

In mathematical terms, qualitative models can be viewed as models that predict the sign of partial derivatives, also dubbed *qualitative partial derivatives*. This makes them attractive for the problem at hand: relations between the outcomes and the variables under our control resemble derivatives. The dependent variables in our case are discrete, which requires a different definition of derivative. We assume that the variables by which we differentiate are categorical as well, since categorical variables are indeed common and since using discrete values may increase the reliability in the spirit of the above findings and, finally, because it is easier to differentiate categorical variables with respect to another categorical and not continuous variable.

The paper's basic contributions are contained in Section 2:

a) the definition of probabilistic discrete qualitative partial derivative,
b) the algorithm for computation of such derivatives at each data point,
c) induction of models, that is, generalization via machine learning from derivatives.

In Section 2.1, we define a new type of qualitative relation, *probabilistic discrete qualitative partial derivative* (PDQ PD) which relates categorical variables. The definition is based on recasting the qualitative partial derivative in probabilistic terms: instead of predicting *the effect of change in numeric input variable on the numeric output*, we predict *the effect of a certain change of the categorical input variable on the probability of the target class*. To end up with a qualitative model, we are interested in whether the change was positive or negative (or none).

The first step in computation of derivatives requires calculating the conditional probabilities of class values with the target attribute and all other relevant attributes as conditions. The set of relevant attributes at each data point are determined using a greedy approach (Section 2.2). Probabilities are then clustered to obtain partial ordering (Section 2.3). Both procedures, which constitute the new algorithm Qube, are heuristic and also depend on the availability of suitable data in the vicinity of the point where the derivative is computed, therefore some smoothing or generalization is necessary.

The paper puts less emphasis on the third point (c). By using the computed derivatives as labels, an arbitrary machine learning method can be used for induction of qualitative models (Section 2.4). We experiment with decision trees because of their simplicity and interpretability. When aiming for accuracy, one can choose support vector machines, neural networks or other modern methods. Alternatively, the data produced by Qube can be visualized or used for construction of CP-nets. Exploring these options is beyond the scope of the paper.

We demonstrate different uses of the method and observe its properties on several examples in Section 3. We start with an artificial data set resembling the one described in the beginning but with known ground truth, so the results can be evaluated (Section 3.1). This example also shows why direct use of machine learning algorithms on such data does not yield useful results. We continue by systematically observing the behavior of the algorithm with respect to the data size and noise, again using a purely synthetic domain. The third case is typical for qualitative modeling: the data describes a complex physical phenomenon and the modeling must find a simple qualitative description, in this case in terms of PDQ PDs. The final experiment is run on medical data in which we do not have control over any of the attributes. This shows how qualitative partial derivatives are also useful for analysis of standard machine learning data.

2. Methods

We start by a definition of probabilistic discrete qualitative partial derivatives based on conditional probabilities of the target class given the attribute values. The computation of these probabilities from data requires selecting proper subsets of examples, which we describe next. Finally, we show how to combine the computed probabilities into partial derivatives and use them to induce qualitative models.

2.1. Probabilistic discrete qualitative partial derivative

The derivative of a function $f(x)$ at a certain point x_0, $f'(x_0)$ tells us, informally, the change of the function value corresponding to a certain (small) change in the value of the function's argument, *e.g.*

$$f(x_0 + \Delta x) - f(x_0) = f'(x_0)\Delta x. \tag{1}$$

For functions of multiple arguments, *e.g.* $f(x_1, x_2, \ldots, x_n)$, we compute derivatives by each argument x_i separately and denote them by $\partial f/\partial x_i$.

Qualitative derivatives, which we will denote by $\frac{@f}{@x}$, are similar to ordinary derivatives except that they give only the direction of change, that is, whether the function will increase or decrease when its argument increases. A qualitative derivative of a function is positive (negative, zero) if the continuous derivative is positive (negative, zero).

Now consider a multivariate distribution which assigns a probability y to each element of Cartesian product of $\mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_n$. In machine learning, $(a_1, a_2, \ldots, a_n) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_n$ can be values of discrete attributes $A_1, A_2, \ldots A_n$ describing an example; we will call this example a *reference example*. The probability that such an example belongs to some target class c is thus:

$$y = p(c|a_1, a_2, \ldots, a_n). \tag{2}$$

Recall that qualitative derivative of f with respect to x_i computed at $x_1, x_2, \ldots$ tells whether a certain change of x_i will *increase or decrease the function value* if other arguments remain constant. Let the *probabilistic discrete qualitative partial derivative* at $(a_1, a_2, \ldots, a_n)$ with respect to A_i tell whether a change of value of the attribute A_i from a_i to a_i' will *increase or decrease the probability* of the target class:

$$\frac{@f}{@A_i : a_i \to a_i'}(a_1, \ldots, a_n) = \\ \begin{cases} +, & p(c|a_1, \ldots, a_i, \ldots, a_n) < p(c|a_1, \ldots, a_i', \ldots, a_n) \\ \circ, & p(c|a_1, \ldots, a_i, \ldots, a_n) = p(c|a_1, \ldots, a_i', \ldots, a_n) \\ -, & p(c|a_1, \ldots, a_i, \ldots, a_n) > p(c|a_1, \ldots, a_i', \ldots, a_n). \end{cases} \tag{3}$$

Let us define a total order on set $\mathcal{A}_i$, with respect to fixed values of a_j for all $j \neq i$:

$$a_i \prec a_i' \Leftrightarrow p(c|a_1, \ldots, a_i, \ldots, a_n) < p(c|a_1, \ldots, a_i', \ldots, a_n) \tag{4}$$

This allows us to rewrite (3) as

$$\frac{@f}{@A_i : a_i \to a_i'}(a_1, \ldots, a_n) = \begin{cases} +, & a_i \prec a_i' \\ \circ, & a_i = a_i' \\ -, & a_i \succ a_i'. \end{cases} \tag{5}$$

The derivative $@f/@A_i : a_i \to a_i'$ for any pair a_i and a_i' can thus be described by a total ordering of attribute values $\mathcal{A}_i$.

It is known that on summer Mondays the rain in Spain stays mainly in the plain, that is

$$p(\text{rain}|\text{plain, summer, Monday, Spain}) > \\ p(\text{rain}|\text{mountains, summer, Monday, Spain}).$$

Therefore,

$$\frac{@\text{rain}}{@\text{location: plain} \to \text{mountains}}(\text{summer, Monday, Spain}) = -$$

since going from the plain to the mountains decreases the probability of rain (on summer Mondays in Spain).

Equivalently, we can write that mountains ≺ plain (for summer Mondays in Spain). The derivative is defined by the total ordering of values. Assuming that there are only three types of locations in the Spain, with the third, the seashore, getting the least rain, the derivative may equal

$$\text{seashore} \prec \text{mountains} \prec \text{plain}.$$

The relation refers to the specific data point, that is, for summer Mondays in Spain. Weather patterns for Spanish winters or for summer Sundays in Britain may be different.

2.2. Computation of conditional probabilities

To compute the derivative with respect to A_i, we have to estimate the conditional probabilities $p(c|a_1, \ldots, a_n)$ at a single point $E = (a_1, a_2, \ldots, a_n)$ from data sample for different values of A_i. For example, to compute the derivative of rain likelihood with respect to the location on summer Mondays in Spain, we must compute $p(\text{rain}|\textit{location}, \text{summer, Monday, Spain})$ for all $\textit{location} \in \{\text{plain, mountains, seashore}\}$.

These probabilities cannot be estimated directly using a perfect Bayesian approach, for instance by relative frequencies, since there may be only a few or even no examples in the data which match the conditional part. We also cannot use a naive Bayesian method since it would reduce the PDQ PD to comparison of $p(c|a_1)$ and $p(c|a_1')$, canceling out all the terms corresponding to values of other attributes: the naive Bayesian assumption of conditional independence of attributes given the class implies that the derivative $\frac{@f}{@A_i : a_i \to a_i'}$ is constant over the entire attribute space.

The problem requires a semi-naive Bayesian approach. We replace the condition in $p(c|a_1,\ldots,a_n)$ with a relaxed condition $P(c|a_i,\mathcal{D})$, where $\mathcal{D}\subseteq\{a_1,\ldots,a_n\}$ includes only the attribute values that are conditionally dependent on a_i given the class, and the values of the attribute with respect to which we compute the derivative. Ignoring the conditionally independent values does not change the computed derivative (see the proof in the Appendix). In our running example, we may discover that day of the week plays no role in rain likelihood, so it suffices to compute probabilities $p(\text{rain}|\textit{location},\text{summer},\text{Spain})$, that is, $\mathcal{D}=\{\text{summer},\text{Spain}\}$.[2]

We construct the set of conditions $\mathcal{D}$ with a greedy approach. We start with an empty set $\mathcal{D}$.

To check whether an attribute A_j should be added to $\mathcal{D}$, we try to reject the assumption that the value of the attribute A_i is conditionally (given $\mathcal{D}$) independent of whether A_j has the value a_j or not, that is, $p(A_j=a_j,A_i|c,\mathcal{D})=p(A_j=a_j|c,\mathcal{D})p(A_i|c,\mathcal{D})$. If the assumption holds, the condition $A_j=a_j$ is unrelated to A_i and does not need to be added to $\mathcal{D}$.

Note that we do not check the total independence of A_j and A_i: we consider all values of A_i, while for A_j we consider only a_j *vs.* other values of A_j.

- All values of A_i need to be considered so that we can use the same set of conditions $\mathcal{D}$ for all derivatives with respect to A_i at a certain reference example. This ensures that probabilities $p(c|a_i,\mathcal{D})$ for all $a_i\in\mathcal{A}_i$ in (3) are comparable and thus useful for defining a total ordering of A_i.
- For A_j, we are interested only in whether a_j can be substituted by other values of A_j without affecting A_i.

The independence assumption is checked by the χ^2 test. We construct separate tables with $2\times|\mathcal{A}_i|$ cells for class c and for its complement; their rows correspond to whether $A_j=a_j$ and columns correspond to the values of A_i. From the first table, we compute the expected absolute frequencies as $n(c,\mathcal{D})p(a_j|c,\mathcal{D})p(v|c,\mathcal{D})$ and $n(c,\mathcal{D})(1-p(a_j|c,\mathcal{D}))p(v|c,\mathcal{D})$, where $v\in\mathcal{A}_i$ and $n(c,\mathcal{D})$ is the number of examples in class c that satisfy the conditions $\mathcal{D}$. Frequencies for the complement of c are computed analogously. The observed frequencies are computed from the training data by taking the conditions $\mathcal{D}$ into account.

The sum of χ^2 statistics for the two tables is distributed according to χ^2 distribution with $2(|\mathcal{A}_i|-1)$ degrees of freedom. The greedy algorithm for construction of $\mathcal{D}$ starts with an empty set. At each step, it tests the independence assumption for all attributes not within $\mathcal{D}$ and computes the corresponding p-value. We select the one with the lowest value and add it to $\mathcal{D}$. We stop the procedure when the lowest p-value is above the specified threshold or when the number of examples matching the conditions $\mathcal{D}$ falls below the given minimum. This is needed to ensure the reliability of χ^2 statistics and of estimated conditional probabilities. Our use of p-value does not require adjustments for multiple hypotheses testing since the p-value is used only as a stopping criteria and not to claim the significance of the alternative hypothesis. After completing the set of conditions $\mathcal{D}$, we compute $p(c|a_i,\mathcal{D})$ for all $a_i\in\mathcal{A}_i$ using relative frequency, Laplacean estimate or m-estimate [6] on examples matching $\mathcal{D}$. The pseudo code of the algorithm is shown in Algorithm 1 and 2.

Algorithm 1 Compute conditional probabilities.

Input: Learning data set
Output: Conditional probabilities $p(c|a_i,\mathcal{D})$ for $a_i\in\mathcal{A}_i$ for each example

```
function COMPUTE_CONDITIONAL_PROBABILITIES( )
    for each $e$ in data do
        $\mathcal{D}=\{\}$
        repeat
            $(p\_value, A)=$ SELECT_MOST_IMPORTANT_ATTRIBUTE( )
            $\mathcal{D}=\mathcal{D}\cup A$
        until $p\_value > p\_limit$
        compute $p(c|a_i,\mathcal{D})$ for $a_i\in\mathcal{A}_i$ from data
    end for
end function
```

Although the proposed greedy procedure is simplistic, it works well in practice. We must also keep in mind that the selection of attributes needs to be fast since it is recomputed for each point at which we compute the derivative, which rules out any advanced search for sets of dependent values.

[2] This can be illustrated on the more familiar derivatives of continuous functions defined by:

$$\frac{\partial f}{\partial x_1}(x_1,\ldots,x_n)=\lim_{h\to 0}\frac{f(x_1+h,x_2,\ldots,x_n)-f(x_1,x_2,\ldots,x_n)}{h}.$$

When computing the derivative with respect to x_1, we subtract the function value in two points where all arguments except x_1 are the same. If the function value at point $(x_1,\ldots,x_n)$ does not depend on, say, x_2, we may use different values of x_2 in the two terms.

Algorithm 2 Select the most relevant attribute.

```
Input: $i$ the index of differentiated attribute
Output: most relevant attribute $X$ and the corresponding $p$-value
  function SELECT_MOST_RELEVANT_ATTRIBUTE
    $p_{min} = 0$
    for each attribute $A_j \neq A_i$ do
      $\chi^2 = 0$
      for $v$ in $A_i$ do
        // $E$ ... expected frequencies
        // $O$ ... observed frequencies
        // for target class $c$
        $E = n(c, \mathcal{D}) p(a_j|c, \mathcal{D}) p(v|c, \mathcal{D})$
        $\chi^2 = \chi^2 + (O(a_i, a_j) - E)^2/E$
        $E = n(c, \mathcal{D})(1 - p(a_j|c, \mathcal{D})) p(v|c, \mathcal{D})$
        $\chi^2 = \chi^2 + (O(a_i, \overline{a_j}) - E)^2/E$
        // for non-target class $\overline{c}$
        $E = n(\overline{c}, \mathcal{D}) p(a_j|\overline{c}, \mathcal{D}) p(v|\overline{c}, \mathcal{D})$
        $\chi^2 = \chi^2 + (O(a_i, a_j) - E)^2/E$
        $E = n(\overline{c}, \mathcal{D})(1 - p(a_j|\overline{c}, \mathcal{D})) p(v|\overline{c}, \mathcal{D})$
        $\chi^2 = \chi^2 + (O(a_i, \overline{a_j}) - E)^2/E$
      end for
      $p = $ get_ p_value($\chi^2$)
      if $p < p_{min}$ then
        $p_{min} = p$
        $X = A_j$
      end if
    end for
  end function
```

2.3. Computation of derivatives

The total order of $\mathcal{A}_i$ is determined by the order of the corresponding probabilities as defined in (3). To handle noisy data, we will however treat two probabilities (and the corresponding values of A_i) as equal if they differ by less than a user-provided threshold.

For this, we use hierarchical clustering of values with average linkage [7] using the difference of probabilities as distances. The clustering is stopped when the distance between the closest clusters is greater than some predefined threshold (in our experiments set to 0.2).

For example, let A_i be a five-valued attribute with values v_1 to v_5. Probabilities $p(c|v_i, \mathcal{D})$ for these values equal 0.1, 0.2, 0.3, 0.5 and 0.6, respectively. Let the merging threshold be 0.2. We recognize v_1 and v_2 as equivalent and assign them the average probability of $(0.1 + 0.2)/2 = 0.15$. Next we merge v_4 and v_5, the average probability is $0.5 + 0.6 = 0.55$. Finally, we merge v_1 and v_2 with v_3; the average probability is $(0.1 + 0.2 + 0.3)/3 = 0.2$. We then stop since the difference between $p = 0.2$ (v_1 to v_3) and $p = 0.55$ (v_4 to v_5) exceeds the threshold of 0.2. The resulting total ordering of A_i is $v_1 = v_2 = v_3 \prec v_4 = v_5$. In case of ties, clustering chooses the pair to merge at random, potentially resulting in different possible results.

2.4. Induction of qualitative models

To induce a qualitative model with respect to a certain attribute A_i, we first compute the PDQ PD for the entire learning set: for each example, we compute the set of dependent values $\mathcal{D}$ and find the total ordering of attribute values $\mathcal{A}_i$ as explained in the previous two sections. We replace the original class labels with partial derivatives (that is, the total ordering) and induce a model for predicting the ordering. In principle, any learning algorithm can be used for this task.

3. Evaluation

The described approach is new and we are not aware of any method that can be used on similar data in a similar fashion. The aim of this section is rather to demonstrate the different scenarios under which Qube can be used, to qualitatively compare its results with the direct use of machine learning methods that can be run on this type of data, and to observe its behavior under different conditions.

In all below experiments we use Qube to compute partial derivatives, replace the original target variable with the derivatives and then induce a model from the modified data. This approach is more practical than analyzing the derivatives directly, and it is also the anticipated use of Qube. For modeling, we use classification trees to be able to compare the models to the ground truth relations. In practice, symbolic methods would be used when we are interested in interpretability, while sub-symbolic methods may be preferred when we aim for higher accuracy.

We start with data that resembles the motivation from the beginning of the paper. We constructed a similar data set with three different treatments and a control group, but with a known ground-truth concept so that we can, unlike in

Table 1
The hidden model for Rat data set.

Treatment	*p(survival)*	*Under condition*
1	100%	$A_1 = 1 \wedge A_3 = 2 \wedge A_4 = 0$
2	65%	$A_1 = 1 \wedge A_3 = 2 \wedge A_4 = 0$
3	85%	$A_1 = 0 \wedge A_2 \in \{1, 2\}$
0, 1, 2 or 3	100%	$A_3 = 1$
0	20%	$A_3 \neq 1$

Table 2
A small sample of the rat data set illustrating the input data for Qube algorithm. The entire data set consists of $N = 1000$ learning examples.

Treatment	A_1	A_2	A_3	A_4	*Survived*
0	1	2	0	0	0
2	1	0	0	1	0
1	1	0	1	1	1
0	1	2	0	0	0
3	0	0	1	0	1
0	0	1	0	0	0
0	0	0	0	1	0
2	0	2	2	0	0

the data from the motivation, observe the correctness of the model. We also show why using Qube for this data is more appropriate than inducing the classification trees directly from the data.

We will then use a similar data set, but with a simpler target concept to observe the accuracy of the method on different data sizes and noise levels.

The third scenario is a typical use case from qualitative modeling. The data is generated using a rather complex physical model and the standard task of qualitative modeling is to subsume it with a simple but qualitatively correct model. With Qube, the model will take the form of qualitative partial derivatives.

The last use case will show that PDQ PD are not limited to decision support or preference induction, but can also be useful in general labeled attribute-based data. For this case, we used actual medical data, again generalized the compute derivatives in the form of rules and then asked two medical doctors evaluate interpretability and correctness of the rules.

We ran all experiments with parameters that were fixed in advance. The threshold for stopping the clustering was always 0.2, as described in Section 2.3. The m in the m-estimate [6] of probabilities was set to the common setting $m = 2$. Classification trees were induced using our reimplementation of the C4.5 algorithm [8] with the following arguments: max_depth=5, mForPruning=2, sameMajorityPruning=True, binarization=False, max_majority = 0.95.

3.1. Rat data set

3.1.1. Data

One thousand sick rats with different genetic predispositions are given one of the three treatments (1, 2, or 3) or no treatment (0). The outcome describes the rat's survival (1) or death (0). Four attributes describing the important genetic markers, $A_1 = \{0, 1\}$, $A_2 = \{0, 1, 2\}$, $A_3 = \{0, 1, 2\}$, $A_4 = \{0, 1\}$, constitute the following hidden ground truth model:

The probability of survival is 100% for the rats with genetic marker $A_3 = 1$ disregarding the treatment type.

Treatment 1 works perfectly for the combination of markers $A_1 = 1 \wedge A_3 = 2 \wedge A_4 = 0$, while the rats with the same combination of genetic markers that receive treatment 2 have only 65% chance of survival. Treatment 3 works well for rats with $A_1 = 0 \wedge A_2 \in \{1, 2\}$; the probability of survival in this group is 85%. If the rat is given some treatment but does not match the necessary condition, it dies.

The survival rate for rats that do not receive the treatment (treatment $= 0$) is 20%. Table 2 presents a small sample of the data set.

In a real world scenario, the researchers studying the effect of treatments do not know the model and thus work in the reverse direction: they measure the genetic parameters, carry out the experiment, and compile the data (Table 2) with a goal of learning the relation between the treatment and the genetic markers (Table 1).

3.1.2. Experiment

The desired output is *survived* $= 1$ and selected variable in which the clinicians are interested is *treatment*. In the formalism described above, we are interested in a PDQ PD $\frac{@survived=1}{@treatment}$.

We used Qube to calculate $\frac{@survived=1}{@treatment}$ for each learning example in the original data set. The result is an ordering of treatment types by increasing probability of survival for each rat. We replaced the original outcome (survival *vs.* death) with those orderings and induced a decision tree in Fig. 1.

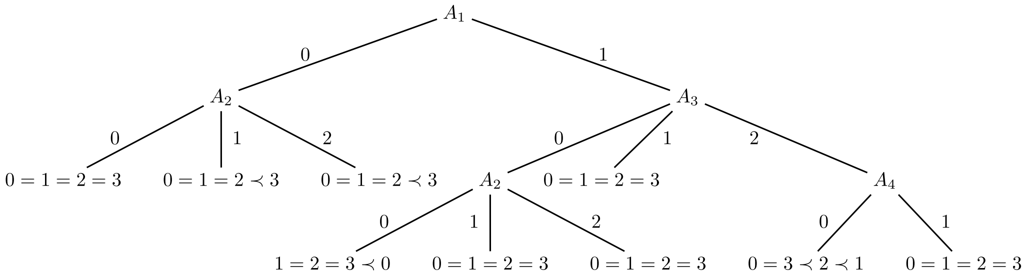


Fig. 1. Tree-like preference model for $\frac{@survived}{@treatment}$ in rats data set.

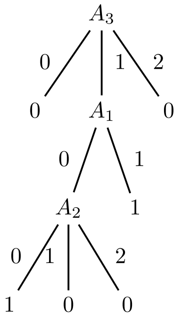


Fig. 2. Decision tree for prediction of survival in rats data set.

3.1.3. Discussion

Analyzing the tree branch by branch, we see that the model almost perfectly matches the underlying hidden model.

- The leaf of the branch corresponding to $A_1 = 1 \wedge A_3 = 2 \wedge A_4 = 0$ states that $0 = 3 \prec 2 \prec 1$. This is correct because for this combination of genetic markers, treatment 1 ($p(survived) = 100\%$) is better than treatment 2 ($p(survived) = 65\%$), while treatment 3 has no effect, which makes it equally non-desirable as no treatment (0).
- Similarly, branches $A_1 = 0 \wedge A_2 = 1$ and $A_1 = 0 \wedge A_2 = 2$ end in preference relation $0 = 1 = 2 \prec 3$, which is in agreement with the fact that rats with $A_1 = 0 \wedge A_2 \in \{1, 2\}$ should be treated with *treatment* = 3 ($p(survived) = 85\%$).
- Branch $A_1 = 1 \wedge A_3 = 0 \wedge A_2 = 0$ states that no treatment (0) is better than treatments 1, 2 or 3: $1 = 2 = 3 \prec 0$. This rule complies with the default rule of the ground truth model; 20% of the rats receiving treatment 0 also survive, unlike the mistreated rats, which die.
- The remaining leaves state that the treatment type has no effect on the survival – there are no preferred treatments ($0 = 1 = 2 = 3$) for some combinations of genetic markers.

In some branches, no treatment is equivalent to some treatments since the leaf covers some rats that are susceptible to those treatments and some that are not.

We contrasted this model with a classical machine learning approach of inducing a model (in this case classification tree) from the original data, without substituting the outcomes with partial derivatives. The resulting tree is shown in Fig. 2. The tree may correctly predict the survival, but this is not what the researcher is interested in. The attribute of interest, *treatment*, does not even emerge in the final model. If we disable the tree pruning, the *treatment* attribute may appear in the tree, but not necessarily in the leaves, which makes the direct interpretation and use in decision making difficult.

The tree induced from the data prepared by Qube expresses the preferences in the leaves, which is a natural way for a clinician interested in the most appropriate treatment for the specific rat: following the attribute values from the root of the tree, the preference in the leaf tells the optimal treatment.

3.2. Equality data set

3.2.1. Data

We study the behavior of the algorithm under varying the number of learning examples (N), the number of attributes (*#atts*), and level of noise (μ) with a slightly simpler target concept that allows us to automatically assess the correctness of the model.

Table 3
Ground-truth models for Equality data set.

PDQ PD	Correct model

$\frac{@f=1}{@A_1}$	if $A_2 = 1$ then $2 \prec 1$
	if $A_2 = 2$ then $1 \prec 2$
	if $A_2 = 3$ then $1 = 2$

$\frac{@f=1}{@A_2}$	if $A_1 = 1$ then $2 = 3 \prec 1$
	if $A_1 = 2$ then $1 = 3 \prec 2$

$\frac{@f=1}{@A_3}$	$1 = 2 = 3 = 4 = 5$

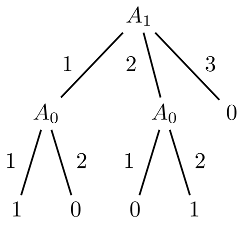


Fig. 3. Classification model for the original data set describing the equality of attributes A_1 and A_2.

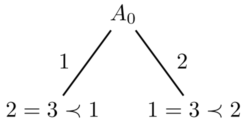


Fig. 4. Preference model for $\frac{@f=1}{@A_2}$.

The underlying concept f in this data set is a binary variable representing the equality of the attributes $A_1 \in \{1, 2\}$ and $A_2 \in \{1, 2, 3\}$:

$$f = \begin{cases} 1, & A_1 = A_2 \\ 0, & A_1 \neq A_2. \end{cases} \tag{6}$$

Other attributes are unrelated to the output variable. Attributes A_3, A_4 and A_5 have five, four and six values, respectively, and the attributes A_i for $i > 5$ have a random number of values in the interval $[2, 5]$.

We compute PDQ PDs of the target output value $f = 1$ with respect to A_1, A_2 and A_3. Each attribute has a different number of values, so we can examine the influence of the attribute cardinality.

The correct models (Table 3) are less intuitive than in the rat data.

For $\frac{@f=1}{@A_1}$, when A_2 equals 1 the value $A_1 = 1$ is “preferred” over $A_1 = 2$ since $f = 1$ requires $A_1 = A_2$. The situation is inverted for $A_2 = 2$ where $A_1 = 2$ is “preferred” over $A_1 = 1$. Since A_1 and A_2 have different number of values, an interesting case appears when $A_2 = 3$. There is no value A_1 could take to increase the likelihood of $f = 1$, i.e. $A_1 = A_2$; the effects of $A_1 = 1$ and $A_1 = 2$ are the same, qualitatively $1 = 2$.

The case of $\frac{@f=1}{@A_2}$ is similar: when $A_1 = 1$, $A_2 = 1$ increases the likelihood of $f = 1$ in comparison with the other two values. Analogously, when $A_1 = 2$, $A_2 = 2$ has higher likelihood for $f = 1$ than 1 and 3.

The third model seems the simplest of the three: $\frac{@f=1}{@A_3}$ as neither value of A_3 increases the likelihood of $f = 1$.

3.2.2. Experiment

We randomly generated data sets for different number of learning examples $N = \{100, 500, 1000\}$, number of attributes $\#atts = \{5, 10, 30, 50\}$, and level of noise $\mu = \{0\%, 5\%, 10\%, 30\%\}$. Level of noise refers to the fraction of learning examples with corrupted values of the output variable f.

For each set of parameters (N, $\#atts$, μ) we sampled ten data sets, computed the PDQ PDs $\frac{@f=1}{@A_i}$, for $i \in \{1, 2, 3\}$, and induced the qualitative models using the C4.5 algorithm. We compared the obtained models with theoretically correct models as described in Table 3. In the following results, we treat the obtained models as correct only if they perfectly match their theoretical counterparts. The models that are only partially correct (*e.g.* one branch of the tree is wrong) are treated as wrong. (See Fig. 3 and Fig. 4.)

Table 4 reports the results for different sizes of data sets: the table contains fractions of correctly reconstructed models over 10 random runs for $\frac{@f=1}{@A_1}$, $\frac{@f=1}{@A_2}$, $\frac{@f=1}{@A_3}$; the symbol ✓ means that the models were correct in all 10 runs and ×

Table 4
The summary of the results for randomly generated data sets with the underlying concept $A_1 = A_2$. We varied the number of learning examples N, the number of attributes *#atts* and the level of noise μ. For each combination, a fraction of correct models learnt from the data set was computed over 10 runs. The models that were always correct are marked with ✓ and the models that were never correct are marked with ×.

						$N = 100$						
#attributes	$\mu = 0\%$			$\mu = 5\%$			$\mu = 10\%$			$\mu = 30\%$		
5	✓	✓	✓	✓	✓	✓	✓	✓	✓	0.6	0.4	0.9
10	✓	✓	✓	0.9	✓	✓	0.9	✓	✓	0.7	0.4	0.9
30	✓	✓	✓	✓	0.8	0.9	0.9	0.7	0.9	0.1	0.2	0.7
50	✓	0.8	✓	0.9	✓	0.9	0.9	0.7	0.5	0.1	0.2	0.5
						$N = 500$						
#attributes	$\mu = 0\%$			$\mu = 5\%$			$\mu = 10\%$			$\mu = 30\%$		
5	✓	✓	✓	✓	✓	✓	✓	0.8	✓	0.2	0.1	✓
10	✓	✓	✓	✓	✓	✓	0.8	0.7	✓	0.1	×	✓
30	✓	✓	✓	0.9	✓	✓	0.9	0.7	✓	×	×	✓
50	✓	✓	✓	0.8	✓	✓	0.5	0.5	✓	×	×	✓
						$N = 1000$						
#attributes	$\mu = 0\%$			$\mu = 5\%$			$\mu = 10\%$			$\mu = 30\%$		
5	✓	✓	✓	✓	✓	✓	✓	✓	✓	0.5	0.2	✓
10	✓	✓	✓	✓	✓	✓	0.9	0.7	✓	0.2	×	✓
30	✓	✓	✓	✓	✓	✓	0.5	0.9	✓	×	×	✓
50	✓	✓	✓	0.9	✓	✓	0.7	0.7	✓	0.1	×	✓

Table 5
Time complexity for calculation of one PDQ PD for different number of learning examples N and the number of attributes *#atts*. All times are in seconds.

#atts	N		
	100	500	1000
5	0.04	0.4	1.6
10	0.06	0.7	2.4
30	0.1	1.9	6.7
50	0.2	3.1	11.5

means that the model was never correct. Throughout all the experiments, the p-value was set to 0.05 and the threshold was set to 20.

3.2.3. Discussion

In data sets without the added noise, Qube almost always produces correct models for all values of N and *#atts*; it fails 2 out of 10 times on the smallest data set and with 50 dummy attributes. In the presence of noise, the number of induced incorrect models increases with higher values of added noise and increasing number of dummy attributes. However, the effect of noise and dummy attributes diminishes with increasing size of the data set N.

We also evaluated the complexity of the algorithm measuring the time it needed for the calculation of a PDQ PD, including the learning time of the classification tree algorithm. Table 5 summarizes the average times (in seconds) over three runs per data set. It shows quadratic growth in N and linear growth in the number of attributes.

3.3. Billiards

3.3.1. Data

Billiards is a common name for table games played with a stick and a set of balls, such as snooker or pool and their variants. The goals of the games vary, but the main idea is common to all: the player uses the stick to stroke the cue ball aiming at another ball to achieve the desired effect. The friction between the table and the balls, the spin of the cue ball and the collision of the balls combine into a very complex physical system [9]. However, despite the complexity, an amateur player can still learn the basic principles of how to stroke the cue ball without knowing much about the physics behind it. In this case study we will use our method to induce a simple model, which could be used by a human player for ranking different shot types in different ball settings.

Our goal is to learn shot preferences in different circumstances from simulated data. Although it is possible to pocket the black ball with several different type of shots, some shots are more appropriate than others thus increasing the probability of a successful shot. For example, if a hole, a black ball and a cue ball are collinear, a direct shot is preferred over a rail-first shot because it is much more difficult to correctly estimate the reflection angles so that the black ball would pocket (not to say that it makes no sense to do so). However, in the presence of other balls which may present an obstacle for a direct shot, a rail-first shot may be preferred.

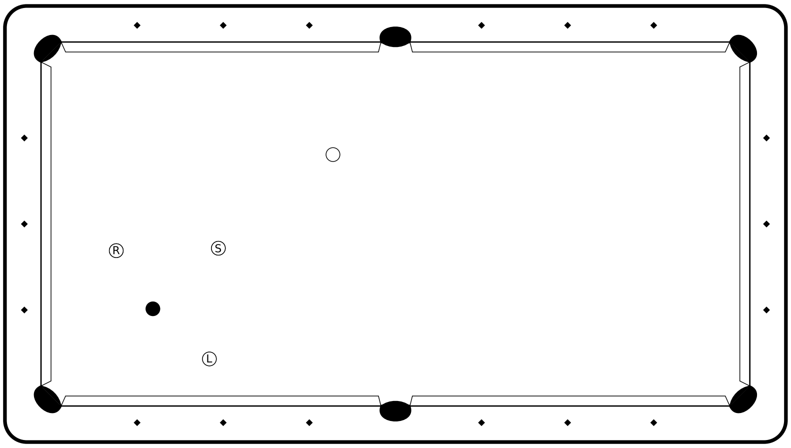


Fig. 5. Blocking the strokes. Ball S blocks a direct shot while balls L and R block left and right shots respectively.

Table 6
Definitions of different types of shots.

Action	Stick elevation	Stroke angle	Stroke velocity	Stroke follow
Strong direct shot	$[0, 5]$	$[\varphi - 0.2, \varphi + 0.2]$	$[3, 4]$	$[-0.1, 0.1]$
Weak direct shot	$[0, 5]$	$[\varphi - 0.2, \varphi + 0.2]$	$[.1, 2]$	$[-0.1, 0.1]$
Left rail-first shot	$[0, 5]$	$[\varphi - 0.2, \varphi + 0.2]$	$[3, 4]$	$[-0.1, 0.1]$
Right rail-first shot	$[0, 5]$	$[\varphi - 0.2, \varphi + 0.2]$	$[3, 4]$	$[-0.1, 0.1]$

We consider the problem of stroking a cue ball in order to pocket the black ball in the presence of other balls that may present an obstacle in the player's attempt to make the desired stroke. The balls have a fixed position as shown in Fig. 5. Usually, a player has several different options how to hit the black ball [9], however, in this case study, we will only consider direct shots and rail-first shots. In the former, the white ball directly hits the black ball, while in the latter, the white ball first hits the rail (also called cushion) and only then the black ball. For the sake of our case study, we derived 4 possible actions a player can make: *strong direct shot, weak direct shot, left-rail-first shot* and *right-rail-first shot*. Strong and weak direct shots (Fig. 6a) differ in the force applied by the player, which is visible in different initial velocities of the white ball. The difference between left and right rail-first shots are shown in Figs. 6b and 6c. In the left-rail-first shot, the white ball will pass the black ball on the left, hit the rail, and then hit the black ball. Similarly, in the right-rail-first shot, the white ball will first pass the black ball on the right, hit the rail, and then hit the black ball.

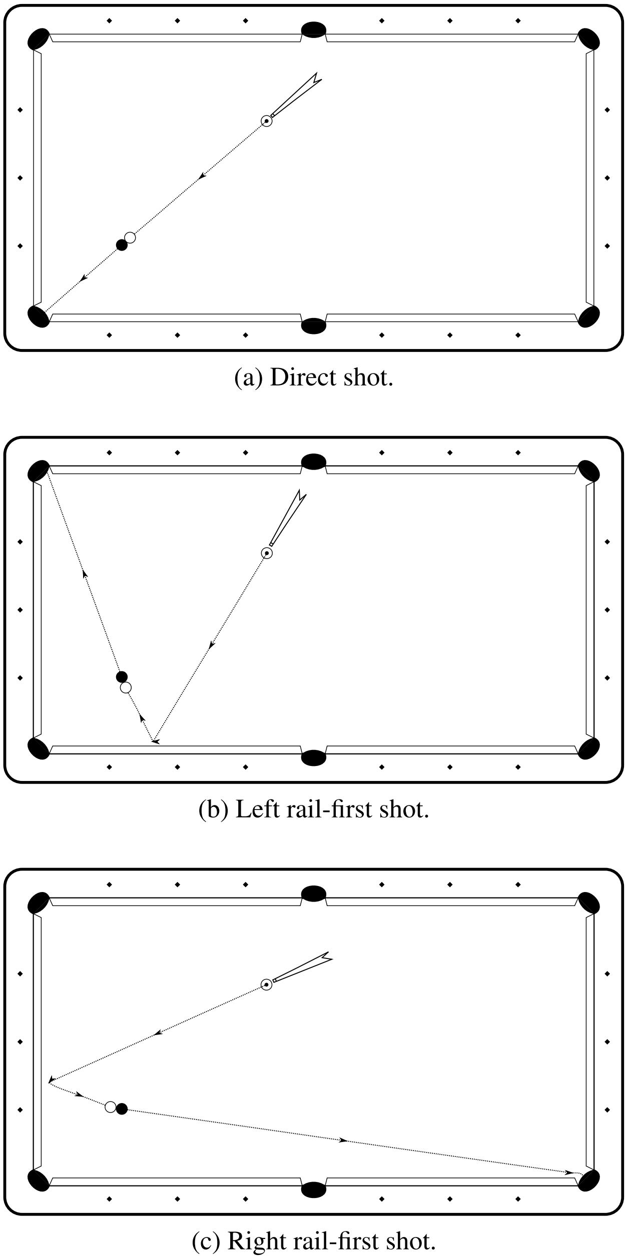
We created a data set of 1000 shots (i.e. learning examples), where each example is described by the following four attributes:

- S – whether a ball blocking a straight shot was present (values:yes/no),
- L – whether a ball blocking a left shot was present (values:yes/no),
- R – whether a ball blocking a right shot was present (values:yes/no),
- *ShotType* – shot type (values:StrDir, WeakDir, LeftRail, RightRail),

and the class variable *black in pocket* with values *yes* and *no*. The goal (target class) in our method is thus '*black in pocket* = *yes*', namely, whether the shot forced the black ball into a pocket. The attribute values of each single example were randomly selected, while the value of the class was determined using the billiards simulator [10]. Each shot in the simulator is defined by: shot direction, stick elevation, shot velocity and shot follow. The most important property of a shot is its direction, as it has the dominant effect on the direction of the black ball after it is hit by the white ball. First, we computed the optimal stroke direction φ that results in pocketing the black ball. However, to account for human imprecision, we added uniform noise in the interval $[-0.2\,\text{deg}, 0.2\,\text{deg}]$. Therefore, the value of stroke direction in the simulator is set to a random value selected from the interval $[\varphi - 0.2, \varphi + 0.2]$. We would expect that more difficult shots are more prone to changes in the optimal stroke setting. Similarly, other properties of the shot were randomly chosen from specified intervals, however these did not require computation of their optimal values. The exact specifications of shot properties are given in Table 6.

3.3.2. Experiment

The induced preference model is shown in Fig. 7.



(a) Direct shot.

(b) Left rail-first shot.

(c) Right rail-first shot.

Fig. 6. Examples of different type of shots used in our case study.

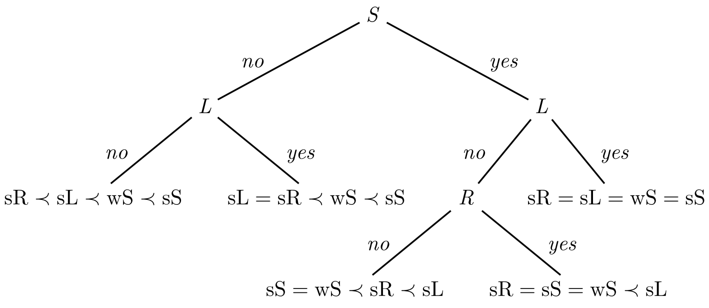


Fig. 7. The induced preference model suggesting the best action regarding the situation of the balls on the table.

3.3.3. Discussion

Induced relations are correct.

- $S = \text{no} \wedge L = \text{no}$: Balls S and L are not on the table. In this case, strong direct shot is preferred over weak direct shot since the velocity of the black ball may not be sufficient for the black ball to reach the pocket using the weak shot. A weak shot is preferred over left rail-first shot which is preferred over right rail-first shot. In general, rail-first shots are less successful than direct shots due to distortions caused by cue ball hitting the rail. However, left rail-first shot is usually better since the path of the ball is shorter than the path in the right rail-first shot. It is irrelevant whether ball R is present or not because it can only block the right rail-first shot which has the lowest preference in any case.
- $S = \text{no} \wedge L = \text{yes}$: Ball L is blocking the left rail-first shot which makes both direct shots preferred over the rail-first shots. The preference of direct shots is the same as above while both rail-first shots are equally preferred.
- $S = \text{yes} \wedge L = \text{no} \wedge R = \text{no}$: Ball S is blocking direct shots which makes both rail-first shots preferable and both direct shots equally bad. Left rail-first shot is preferred over the right one for the same reason as above.
- $S = \text{yes} \wedge L = \text{no} \wedge R = \text{yes}$: Balls S and R are blocking the shots. The left rail-first shot is preferred over all other shots since it is the only open shot.
- $S = \text{yes} \wedge L = \text{yes}$: Balls S and L are blocking both direct and left rail-first shots. There are no shot preferences since all shots are equally hopeless. It seems that the right rail-first shot should be preferred from the others in a similar way that the left rail-first shot is preferred in the previous situation above. Fig. 6c indicates that the black ball should travel much longer distance compared to the case in Fig. 6b meaning that the shot should be stronger. A strong rail-shot is a very difficult one and rarely succeeds, which explains why the preferences are all equal in this case.

3.4. Bacterial infections in elderly

In this last experiment we run the algorithm on a real problem and, at the same time, show its usefulness in a domain where we are interested in partial derivatives without being able to control any of the variables.

3.4.1. Data

The aim of the realistic study is to qualitatively asses the influence of individual risk factors for mortality due to bacterial infection in elderly population. The proportion of elderly people in developed world is rapidly growing [11]. It is estimated that by 2020, the elderly will constitute more than 16% of the population in the USA [12]. Bacterial infection is a common cause of mortality in the aged: nearly 14% of hospital admissions in elderly patients are due to bacterial infection [13] and they account for one third of all deaths in older population [14]. Compared with the younger population, bacterial infection in elderly usually presents with different clinical symptoms. The signs of infection in older patients may be less apparent or even absent, which presents unique diagnostic challenges to clinicians [11].

The data was collected at the Department of Infectious Diseases of the University Medical Center Ljubljana, Slovenia. Patients were enrolled in the study in the period from June 1st, 2004 to December 31st, 2005 using the following inclusion criteria: (i) age ≥ 65 years, (ii) hospitalization due to bacterial infection, (iii) routine laboratory tests performed.

Data included 602 patients having C-reactive protein value above 60 mg/l upon the admission to the hospital, which indicated a bacterial infection. Patients were identified prospectively and data were collected prospectively over the time of the study by assistants that were blinded to the study purpose. An infectious diseases specialist who was unaware of the final outcome reviewed the charts and excluded the patients with nonbacterial infections. We observed the mortality; the outcome (class) was a binary variable *DEATH* with the following distribution: $DEATH = Yes$: $77/602 = 12.8\%$ and $DEATH = No$: $525/602 = 87.2\%$.

Data consists of 31 categorical attributes (Table 7), defined by clinicians who carried out the study.

Demographic data included: sex, age, nursing home residence, immobility, presence of permanent urine catheter, presence of pressure ulcer, presence of prosthetic medical device (artificial heart valve or joint prosthesis). Comorbidities included: diabetes mellitus, coronary artery disease, congestive heart disease, chronic obstructive pulmonary disease, renal impairment, liver disease, cerebrovascular disease, immunosupression. Immunosupression was defined as prolonged therapy (6 months or more) with corticosteroids, treatment with citostatic agents or other immunomodulatory agents, patients with transplanted organs or tissues, patients with malignancy. Vital signs were collected from nursing data and included: body temperature, respiratory rate, heart rate, systolic blood pressure, oxygen saturation. We recorded whether patients experienced frailty from the onset of the disease. Frailty was defined as inability to perform daily tasks (*e.g.* feeding, bathing, *etc.*) that patients were able to perform themselves before the onset of the illness. We recorded whether mental status changed from the onset of the current disease. Mental status was recorded as normal/oriented, responsive but disoriented, unconscious. Laboratory data included leukocyte count, percentage of band forms, platelets, blood creatinine and urea value, glucose value, serum sodium concentration and presence of abnormal liver function tests. We recorded all microbiological specimens obtained and final diagnosis at discharge. The primary outcome for the study was functional decline 21–28 days after hospital discharge. Functional decline was defined as discharge disposition to a nursing home care facility, any significant decline in functional or cognitive ability, or overall quality of life observed by patients themselves, their relatives or nursing staff.

We set $DEATH = Yes$ as a target class, therefore predicting the risk posed by different factors.

Table 7
The observed attributes and their values.

Attribute name	Attribute values
SEX	M, F
AGE	$[65, 74]$, $[75, 84]$, ≥ 85
NH RESIDENT	Yes, No
COMORBIDITIES	0, 1, MANY
DIABETES	Yes, No
HEART D.	Yes, No
KIDNEY D.	Yes, No
LIVER D.	Yes, No
LUNG D.	Yes, No
IMMUNOSUPPRESSION	Yes, No
NEUROLOGICAL D.	Yes, No
MOBILITY	Yes, No
CONTINENCE	Yes, No
PRESSURE ULCER	Yes, No
URINE CATHETER	Yes, No
BODY TEMP.	≤ 37.80, > 37.80
RESPIRATORY RATE	≤ 10.00, $(10.00, 20.00]$, > 20.00
SATURATION	≤ 90.00, > 90.00
HEART RATE	≤ 60.00, $(60.00, 100.00]$, > 100.00
BLOOD PRESSURE	≤ 90.00, > 90.00
MENTAL CHANGE	Yes, No
UNCONSCIOUS	Yes, No
FRAILTY	Yes, No
LEUKOCYTE	≤ 4.00, $(4.00, 10.00]$, > 10.00
BAND FORMS	≤ 10.00, > 10.00
THROMBOCYTES	≤ 100.00, > 100.00
CREATININE	≤ 90.00, > 90.00
UREA VALUE	≤ 6.00, > 6.00
GLUCOSE	≤ 4.00, $(4.00, 7.50]$, > 7.50
Na	≤ 135.00, $(135.00, 145.00]$, > 145.00
SITE OF INFECTION	Respiratory, Other, Gastrointerocolitis, SoftTissue, Urinary

3.4.2. Experiment

We again built a qualitative tree from the entire data set, taking the PDQ PD as the class variable. The resulting model explains a qualitative relation between mortality and the attribute A_i, assuming ceteris paribus: how does the risk change if we change A_i but keep all other attributes' values fixed. Table 8 lists the models for all attributes.

3.4.3. Discussion

Two clinical doctors evaluated the above models regarding their use in clinical practise. They found the models easy to understand and in accordance with the domain knowledge, except for the following two models: $@DEATH = Yes/@$ KIDNEY DISEASE and $@DEATH = Yes/@$ CATHETER. While additional medical tests should be carried out to explain the model for kidney disease, further analysis of the models for CATHETER revealed that the algorithm discovered a subgroup of immobile patients that usually also have the urine catheter: COMORBIDITIES $\neq 0$ and THROMBOCYTES >100 and PRESSURE ULCER $=$ Yes. This results in the ordering No $\prec$ Yes in this leaf of the tree. The model still correctly captures the patterns in the data but does not imply causality.

4. Related work

The motivation for our work comes from the field of qualitative reasoning where Qube can be used for learning qualitative models from categorical data. Qualitative reasoning has been mostly concerned with qualitative physics [15–19]. In these works the model was provided by an expert and then used in qualitative simulations. There are only a few algorithms for automated induction of such models [20] and even these are limited to learning from numerical data [4,5]. There are, to the best of our knowledge, no algorithms for learning qualitative models from categorical data.

An important part of our method deals with relaxing the strong independence assumption of naive Bayesian approach. There exist a number of methods for this purpose, yet none fits our context. Kononenko introduced *semi-naive Bayes* [21] and Langley and Sage [22] proposed *Selective Bayesian Classiffier*, a variant of the naive method that uses only a subset of the attributes in making predictions. Since their algorithm is only searching for subset of attributes that yields highest classification accuracy, it can not reveal attribute dependencies. Kohavi [23] proposed NBTree, an algorithm for induction of a hybrid of decision-tree classifiers and naive Bayes classifiers. Friedman and Goldszmidt introduced *tree augmented naive Bayes (TAN)* [24] which allows for attributes having another attribute as a parent in the Bayesian network representation. SuperParent TAN, proposed by Keogh and Pazzani [25], improves on classification accuracy of TAN by introducing a new heuristic for exploring dependencies among the attributes. In Bayesian network representation TAN allows attributes to

Table 8
Qualitative models for the risk factors for mortality in elderly patients with bacterial infections.

Attribute name	Qualitative model
SEX	M = F

AGE	IMMUNOSUPPRESSION = Yes: $[75, 84] = [\geq 85] \prec [65, 74]$
	IMMUNOSUPPRESSION = No: $[65, 74] = [75, 84] = [\geq 85]$

NH RESIDENT	No $\prec$ Yes, **except if:** COMORBIDITIES $\neq 0$ $\wedge$ SITE OF INFECTION $\neq$ Other: Yes = No
COMORBIDITIES	$0 = 1 \prec$ MANY
DIABETES	Yes < No, **except if:** IMMUNOSUPPRESSION = No $\wedge$ BLOOD PR. > 90: Yes = No
HEART D.	Yes = No
KIDNEY D.	No $\prec$ Yes, **except if:** RESP. RATE $\in [10, 20]$ $\wedge$ SATURATION > 90: Yes = No
LIVER	No $\prec$ Yes
LUNG D.	Yes $\prec$ No, **except if:** SATURATION >90 $\wedge$ MENTAL CHANGE = No: Yes = No
IMMUNOSUPPRESSION	No < Yes
NEUROLOGICAL D.	No $\prec$ Yes, **except if:** SITE OF INFECTION $\neq$ Respiratory $\wedge$ HEART RATE ≤ 100 $\wedge$ IMMUNOSUPPRESSION = No: Yes = No
MOBILITY	Yes $\prec$ No, **except if:** BAND FORMS ≤ 10 $\wedge$ Na ≤ 145 $\wedge$ BLOOD PRESSURE > 90: Yes = No
CONTINENCE	Yes $\prec$ No
DECUBITUS	No $\prec$ Yes, **except if:** BAND FORMS ≤ 10 $\wedge$ SITE OF INFECTION $\neq$ Other: Yes = No

CATHETER	COMORBIDITIES = 0: No $\prec$ Yes
	COMORBIDITIES $\neq 0$:
	THROMBOCYTES > 100:
	DECUBITUS = Yes: No $\prec$ Yes
	DECUBITUS = No: Yes = No
	THROMBOCYTES ≤ 100: Yes $\prec$ No

BODY TEMP.	$[> 37.8] \prec [\leq 37.8]$, **except if:** MENTAL CHANGE = No $\wedge$ SITE OF INFECTION $\neq$ Other: $[\leq 37.8] \prec [> 37.8]$
RESP. RATE	$[(10, 20] = [> 20] \prec [\leq 10]$
SATURATION	$[> 90.00] \prec [\leq 90.00]$
HEART RATE	BAND FORMS > 10: $(60, 100] \prec [> 100] \prec [\leq 60]$ BAND FORMS ≤ 10: SATURATION > 90: $(60, 100] \prec [\leq 60] = [> 100]$ SATURATION ≤ 90: $(60, 100] = [> 100] \prec [\leq 60]$
BLOOD PRESSURE	$[> 90.00] \prec [\leq 90.00]$
MENTAL CHANGE	No $\prec$ Yes
UNCONSCIOUS	No $\prec$ Yes
FRAILTY	Yes $\prec$ No
LEUKOCYTE	Na > 145: $(4, 10] = [> 10] \prec [\leq 4]$
BAND FORMS	$[\leq 10] \prec [> 10]$
THROMBOCYTES	$[> 100] \prec [\leq 100]$

CREATININE	CATHETER = Yes: $[\leq 90] \prec [> 90]$
	CATHETER = No: $[\leq 90] = [> 90]$

UREA VALUE	$[\leq 6] \prec [> 6]$, **except if:** DECUBITUS = No $\wedge$ IMMUNOSUPPRESSION = No: $[\leq 6] = [> 6]$
GLUCOSE	$(4, 7.5] = [> 7.5] \prec [\leq 4]$, **except if:** UREA VALUE > 6 $\wedge$ SITE OF INFECTION $\neq$ Other: $(4, 7.5] = [> 7.5] = [\leq 4]$
Na	$[\leq 135.00] = (135, 145] \prec [> 145]$
SITE OF INFECTION	RESP. RATE $\notin (10, 20]$: Respiratory=Other=Gastrointerocolitis=Urinary $\prec$ SoftTissue RESP. RATE $\in (10, 20]$: MENTAL CHANGE = Yes: Urinary $\prec$ Respiratory=Other=Gastrointerocolitis=SoftTissue MENTAL CHANGE = No: Respiratory=Other=Gastrointerocolitis=SoftTissue=Urinary

have one other attribute as a parent. Keogh and Pazzani show that approximating the underlying probability distribution is not the best way to improve classification accuracy. A different approach is described in [26], where an Apriori frequent pattern mining algorithm is employed to discover frequent itemsets of arbitrary size together with their class supports.

A lazy algorithm, named Locally Weighted Naive Bayes (LWNB) is proposed in [27]. LWNB relaxes the independence assumption by learning local models at prediction time. The models are learned on weighted set of training instances in the neighborhood of the test instance. In LWNB, the test example neighborhood is chosen using the k-nearest neighbors algorithm. A step further is the Lazy Bayesian Rules (LBR) algorithm [28]. LBR search of the local neighborhood is not based on a global metric. Instead, for each test example, LBR uses a greedy search to generate a Bayesian rule with an antecedent that matches the test example. The basic difference between these approaches and ours is that these methods are concerned with optimizing the accuracy of predictions and not with estimations of the chosen attribute's influence on the target class probability.

Although Qube can be used for learning preferences, it is fundamentally different from the existing preference learning approaches [29–31]. Preference learning usually starts with the data that already describes the preferences, and the task of the learning algorithms is limited to their generalization [32–35]. For a contrast, Qube calculates the PDQ PDs for each learning example, which can then be used for modeling preferences. While one could continue by using the standard preference learning approaches [33,34], we use simple machine learning algorithms and treat preferences as values of a new class variable. Theoretically, it is possible that the number of class values exceeds a reasonable amount but it can be practically very well controlled by setting the threshold parameter (for joining the values) and the size of the neighborhood of the reference example (a kind of smoothing the data). The most relevant preference learning method to apply preference learning to the result of Qube would be label ranking. Similarly, we could continue by learning CP-nets [36], which represent this type of preferences. The main difference to our approach is that it represents the preferences on a single attribute A_i conditioned on the other attributes while CP-nets simultaneously represent preferences on all attribute combinations. Our representation is suitable when we have control over one attribute but not over the others as shown in Section 3.1.

5. Conclusion

Qualitative models are an established tool in many areas of science, most notably in economics. They have however stirred surprisingly little interest in the machine learning community. We have presented, to our knowledge, the first machine learning method for induction of qualitative models from categorical data. The method has a solid theoretical background and works well in practice. In the Rat domain we have shown that Qube correctly captures the qualitative relations even when the underlying concept is relatively complex. Experiments on another synthetic domain show that it performs robustly in the presence of noise and irrelevant attributes. The presented case studies suggest that it also scales well to more complex domains. In the billiards case study Qube successfully modeled the preferences regarding the shot type in different settings of the balls on the table. Finally, the medical case study also shows its more usefulness on general labeled data beyond preference learning or decision making.

Acknowledgement

We are grateful to Prof. Dr. Eyke Hüllermeier for a very kind and valuable discussion that helped us position our work with respect to the field of preference learning. We would also like to thank Prof. Dr. Franc Strle and Dr. Jerneja Videčnik Zorman for evaluating the qualitative models in the medical domain, and Dr. Tadej Janež for his help with the billiards simulator.

Appendix

In a semi-naive Bayesian approach we replace the condition in $p(c|a_1,\dots,a_n)$ with a relaxed condition $P(c|a_i,\mathcal{D})$, where $\mathcal{D} \subseteq \{a_1,\dots,a_n\}$ includes only the attribute values that are conditionally dependent on a_i given the class. Here we prove that the ordering of probabilities $p(c|a_1,\dots,a_i,\dots,a_n)$ does not change if we omit from the conditional part the values that are conditionally independent from a_i given the class c. Let us first redefine the PDQ PD using conditional log odds ratios:

$$\frac{@f}{@A_i : a_i \to a_i'}(a_1,\dots,a_n) = \operatorname{sgn}\ln\frac{p(c|a_1,\dots,a_i,\dots,a_n)/p(\bar{c}|a_1,\dots,a_i,\dots,a_n)}{p(c|a_1,\dots,a_i',\dots,a_n)/p(\bar{c}|a_1,\dots,a_i',\dots,a_n)}, \tag{7}$$

where $\bar{c}$ is the complement of the target class c. It is easy to see that (7) is equivalent to (3).

Let us without loss of generality assume that values a_1 to a_k, $k < i$ are conditionally independent of values a_{k+1} to a_n, given the class. Applying Bayesian rule, using the independence assumption, canceling the identical terms and reapplying the Bayesian rule turns (7) into

$$\frac{@f}{@A_i : a_i \to a_i'}(a_1,\dots,a_n) = \operatorname{sgn}\ln\frac{p(c|a_{k+1},\dots,a_i,\dots,a_n)/p(\bar{c}|a_{k+1},\dots,a_i,\dots,a_n)}{p(c|a_{k+1},\dots,a_i',\dots,a_n)/p(\bar{c}|a_{k+1},\dots,a_i',\dots,a_n)}. \tag{8}$$

This is equivalent to (3) without values a_1 to a_k. Therefore, $p(c|a_1,\dots,a_i,\dots,a_n) \le p(c|a_1,\dots,a_i',\dots,a_n) \Longleftrightarrow p(c|a_{k+1},\dots,a_i,\dots,a_n) \le p(c|a_{k+1},\dots,a_i',\dots,a_n)$.

References

[1] S. Borštnar, A. Sadikov, B. Možina, T. Čufer, High levels of uPA and PAI-1 predict a good response to anthracyclines, Breast Cancer Res. Treat. 121 (2010) 615–624.
[2] P.A. Samuelson, Foundations of Economic Analysis, enlarged edition, Harvard University Press, 1983.
[3] B. Kuipers, Using qualitative reasoning, IEEE Expert 12 (3) (1997) 94–97, http://dx.doi.org/10.1109/MEX.1997.590090.
[4] J. Žabkar, M. Možina, I. Bratko, J. Demšar, Learning qualitative models from numerical data, Artif. Intell. 175 (9–10) (2011) 1604–1619.
[5] I. Bratko, D. Šuc, Learning qualitative models, AI Mag. 24 (4) (2003) 107–119.
[6] B. Cestnik, Estimating probabilities: a crucial task in machine learning, in: ECAI, 1990, pp. 147–149.
[7] R.R. Sokal, C.D. Michener, A statistical method for evaluating systematic relationships, Univ. Kans. Sci. Bull. 28 (1958) 1409–1438.
[8] J. Demšar, T. Curk, A. Erjavec, Črt Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, M. Štajdohar, L. Umek, L. Žagar, J. Žbontar, M. Žitnik, B. Zupan, Orange: data mining toolbox in Python, J. Mach. Learn. Res. 14 (2013) 2349–2353.
[9] D.G. Alciatore, The Illustrated Principles of Pool and Billiards, 1st edition, Sterling, 2004.
[10] D. Papavasiliou, Billiards manual, Tech. rep., 2009, http://www.nongnu.org/billiards/.
[11] T.T. Yoshikawa, Epidemiology and unique aspects of aging and infectious diseases, Clin. Infect. Dis. 30 (6) (2000) 931–933.
[12] K.P. High, Why should the infectious diseases community focus on aging and care of the older adult?, Clin. Infect. Dis. 37 (2) (2003) 196–200.
[13] A.T. Curns, R.C. Holman, J.J. Sejvar, M.F. Owings, L.B. Schonberger, Infectious disease hospitalizations among older adults in the united states from 1990 through 2002, Arch. Intern. Med. 165 (21) (2005) 2514–2520.
[14] C.P. Mouton, O.V. Bazaldua, B. Pierce, D.V. Espino, Common infections in older adults, Am. Fam. Phys. 63 (2) (2001) 257–269.
[15] J. de Kleer, J.S. Brown, A qualitative physics based on confluences, Artif. Intell. 24 (1984) 7–83.
[16] B. Kuipers, Qualitative simulation, Artif. Intell. 29 (1986) 289–338.
[17] B. Kuipers, Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge, MIT Press, Massachusetts, 1994.
[18] K. Forbus, Qualitative Reasoning, CRC Press, 1997.
[19] K. Forbus, Qualitative process theory, Artif. Intell. 24 (1984) 85–168.
[20] M. Klenk, K. Forbus, Analogical model formulation for transfer learning in AP physics, Artif. Intell. 173 (18) (2009) 1615–1638.
[21] I. Kononenko, Semi-naive Bayesian classifier, in: EWSL-91: Proceedings of the European Working Session on Learning on Machine Learning, Springer-Verlag New York, Inc., New York, NY, USA, 1991, pp. 206–219.
[22] P. Langley, S. Sage, Induction of selective Bayesian classifiers, in: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1994, pp. 399–406.
[23] R. Kohavi, Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid, in: Proceedings of the Second International Conference on Knoledge Discovery and Data Mining, 1996.
[24] N. Friedman, M. Goldszmidt, Building classifiers using Bayesian networks, in: Proceedings of the Thirteenth National Conference on Artificial Intelligence, AAAI Press, 1996, pp. 1277–1284.
[25] E.J. Keogh, M.J. Pazzani, Learning augmented Bayesian classifiers: a comparison of distribution-based and classification-based approaches, in: Proceedings of the International Workshop on Artificial Intelligence and Statistics, 1999, pp. 225–230.
[26] D. Meretakis, B. Wuthrich, Extending naive Bayes classifiers using long itemsets, in: KDD '99: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, 1999, pp. 165–174.
[27] E. Frank, M. Hall, B. Pfahringer, Locally weighted naive Bayes, in: Proceedings of the Conference on Uncertainty in Artificial Intelligence, UAI 2003, 2003.
[28] Z. Zheng, G.I. Webb, K.M. Ting, Lazy Bayesian rules: a lazy semi-naive Bayesian learning technique competitive to boosting decision trees, in: Proc. 16th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, 1999, pp. 493–502.
[29] J. Fürnkranz, E. Hüllermeier, Preference Learning, Springer-Verlag, 2010.
[30] F. Aiolli, A. Sperduti, A preference optimization based framework for supervised learning problems, in: J. Fürnkranz, E. Hüllermeier (Eds.), Preference Learning, Springer-Verlag, 2010, pp. 19–42.
[31] F. Rossi, K.B. Venable, T. Walsh, A Short Introduction to Preferences: Between Artificial Intelligence and Social Choice, Morgan and Claypool Publishers, 2011.
[32] C. Boutilier, R.I. Brafman, H.H. Hoos, D. Poole, CP-nets: a tool for representing and reasoning with conditional *ceteris paribus* preference statements, J. Artif. Intell. Res. 21 (2003) 2004.
[33] W. Chu, Z. Ghahramani, Preference learning with Gaussian processes, in: Proceedings of the 22nd International Conference on Machine Learning, ICML '05, ACM, New York, NY, USA, 2005, pp. 137–144.
[34] E. Brochu, N. de Freitas, A. Ghosh, Active preference learning with discrete choice data, in: Advances in Neural Information Processing Systems, 2007.
[35] W. Cheng, J.C. Huhn, E. Hüllermeier, Decision tree and instance-based learning for label ranking, in: Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14–18, 2009, pp. 161–168.
[36] Y. Chevaleyre, F. Koriche, J. Lang, J. Mengin, B. Zanuttini, Learning ordinal preferences on multiattribute domains: the case of CP-nets, in: J. Fürnkranz, E. Hüllermeier (Eds.), Preference Learning, Springer-Verlag, 2009, pp. 273–296.