International Journal of Approximate Reasoning 115 (2019) 69–95

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

# Learning Bayesian networks with local structure, mixed variables, and exact algorithms

Topi Talvitie [a], Ralf Eggeling [b], Mikko Koivisto [a,*]

[a] Department of Computer Science, University of Helsinki, Finland
[b] Department of Computer Science, University of Tübingen, Germany

ARTICLE INFO

*Article history:*
Received 16 March 2019
Received in revised form 11 July 2019
Accepted 8 September 2019
Available online 16 September 2019

*Keywords:*
Bayesian networks
Decision trees
Exact algorithms
Structure learning

ABSTRACT

Modern exact algorithms for structure learning in Bayesian networks first compute an exact local score of every candidate parent set, and then find a network structure by combinatorial optimization so as to maximize the global score. This approach assumes that each local score can be computed fast, which can be problematic when the scarcity of the data calls for structured local models or when there are both continuous and discrete variables, for these cases have lacked efficient-to-compute local scores. To address this challenge, we introduce a local score that is based on a class of classification and regression trees. We show that under modest restrictions on the possible branchings in the tree structure, it is feasible to find a structure that maximizes a Bayes score in a range of moderate-size problem instances. In particular, this enables global optimization of the Bayesian network structure, including the local structure. In addition, we introduce a related model class that extends ordinary conditional probability tables to continuous variables by employing an adaptive discretization approach. The two model classes are compared empirically by learning Bayesian networks from benchmark real-world and synthetic data sets. We discuss the relative strengths of the model classes in terms of their structure learning capability, predictive performance, and running time.


## 1. Introduction

Bayesian networks (BNs) compose a multivariate distribution as a product of univariate conditional probability distributions (CPDs). The potential complexity of the local CPDs, together with the global acyclicity constraint of the BN structure, make the task of learning BNs from data challenging in many ways. Even if we assume that the available data contain neither unobserved variables nor missing entries—like we will do throughout this article—there often remain the following three concerns:

(i) *Statistical efficiency:* the data may be scarce, rendering simple tabular parameterizations of the CPDs, such as the *conditional probability tables (CPTs)*, statistically inefficient.
(ii) *Heterogeneous variables:* so-called *hybrid BNs* contain both discrete and continuous variables, again ruling out the most convenient and standard parameterizations of CPDs.

* Corresponding author.
*E-mail address:* mikko.koivisto@helsinki.fi (M. Koivisto).

https://doi.org/10.1016/j.ijar.2019.09.002

(iii) *Computational efficiency:* the most natural formulations of the learning problem are computationally hard. This limits both the dimensions of the data and the complexity of the CPDs in relation to what can be solved under useful quality guarantees.

Each of these issues (i–iii) has been addressed in the literature, as we will describe in the next paragraphs. The present authors are, however, not aware of any prior work that would simultaneously address all the three concerns.

To address the issue of statistical efficiency, one can replace tabular CPDs by more sophisticated structures where the effective number of parameters adapts to the amount and complexity of the available data. The notion of context-specific independence [1] offers a particularly appealing way to achieve this: given certain configuration for a subset of the *predictor* variables, the *response* variable is independent of the remaining predictors. For structure learning in BNs, context-specific independences have been represented with decision trees [2] and with the somewhat more expressive decision graphs [3]. Both model classes are non-parametric in the sense their number of parameters can grow large, letting the models approximate any CPD (in fact, match CPTs for discrete variables over finite domains). These studies were, however, limited to discrete variables and greedy or local search algorithms, thus not addressing issues (ii) and (iii). Also exact algorithms to partition to joint state space of discrete predictors have been proposed [4], but these algorithms become computationally infeasible already with, say, three predictors that take four states each.

For hybrid BNs, there are two main approaches. One is to discretize all continuous variables. In practice, this means that separately for each continuous variable, its range is partitioned into intervals, i.e., bins, which are then treated as the states of the discretized variable. Then it is common to assign CPTs for the discretized variables, even if this ignores the natural ordering of the states. Simple preprocessing routines of this kind are computationally convenient, but they are prone to lose information in an uncontrolled way. And if, as a remedy, one increases the number of bins, then the statistical efficiency becomes a concern. There are also more involved schemes, which couple multi-variable discretization with greedy search for a high-scoring network structure [5]; these schemes potentially yield better discretizations, but they appear to require search heuristics that cannot guarantee optimality of the found solution. Another approach is to employ CPDs than can accommodate both discrete and continuous variables, such as conditional linear Gaussian models and their generalizations to discrete responses via appropriate link functions [6, Ch. 5]. These models are, however, nonsatisfactory concerning statistical efficiency: one can argue that they overparameterize by introducing separate linear model for every configuration of discrete predictors, while they also underparameterize by ignoring nonlinear interactions of continuous predictors. Mixtures of truncated exponentials (MTEs) offer a more sophisticated model, in which a decision tree partitions the space of predictor variables into regions, in each of which the response is assigned a mixture distribution [7]. Unfortunately, no fast algorithms to handle MTEs of needed sizes are known, and consequently one has to resort to procedures that do not offer performance guarantees.

The hope for computational efficiency has to be interpreted in relation to the NP-hardness of the BN structure learning problem [8]: a realistic goal is to solve typical moderate-size instances in practice without sacrificing the optimality of the solution, rather than aiming at worst-case polynomial time complexity. Nowadays, we have several exact algorithms, or solvers, that achieve this goal, especially if choosing the bests solver for each given problem instance [9]. While different solvers employ different algorithmic techniques, such as dynamic programming [10–13], A$^*$ search [14], integer linear programming [15], and constraint programming [16], they all proceed in two steps: First, the local scores are computed for all nodes and their possible parent sets, often setting an upper bound for the number of parents. Second, a globally optimal network structure is found by combinatorial search. One challenge in applying these solvers is that the number of local scores computed in the first step can grow large for larger networks. Partly for this reason, the solvers have been mostly studied in settings where the local scores are of some of the aforementioned simple forms, enabling fast evaluation. It has been observed that exact algorithms, when applicable, not only remove the uncertainty on the quality, but also tend to perform better in structure learning and density estimation [17].

In this article, we address all the three issues (i–iii) simultaneously by putting forward a local model based on classification and regression trees (CARTs) [18]. We derive a Bayes score mostly following the works on Bayesian CART [19–21]. The main challenge is that learning CART models is computationally intractable; the existing algorithms are based on greedy and local search and cannot offer useful quality guarantees. Here, two observations come to rescue. First, in the BN learning context the number of predictors can be assumed to be relatively small in practice. This reduces the number of possible CART structures, i.e., decision trees, but is alone not sufficient for making the search computationally tractable. The second observation is that the so-called *dyadic CART*, in which the recursive splits of the ranges of continuous variables are always at the midpoint, essentially inherits the statistical power of CART and yet enables significantly faster global optimization [22–25]. Building on these observations, we introduce the *partition–dyadic CART (PCART)* model; here "partition" refers to (recursive) splitting of the state spaces of categorical variables—to the best of our knowledge, categorical predictors have not been allowed in previous works on dyadic CART.

When all variables are categorical, one can compare the performance of PCART against standard CPTs in a straightforward manner. To compare PCART against analogous tabular models also in the presence of continuous variables, we introduce a *full-table (FT)* variant of PCART: it discretizes the involved continuous predictors (but not the response). While FT adapts the number of states per variable to the data at hand, it renders global optimization computationally feasible, unlike some aforementioned discretization schemes [5].

In addition to these methodological developments, we study empirically the following two questions: Does PCART boost the statistical efficiency of structure learning and density estimation in practice, that is, on benchmark data sets? What is the price to be paid in terms of computational efficiency, i.e., which kind of problem instances, in terms of model complexity and data dimensions, can be solved in a reasonable time? One could hypothesize that, thanks to a parsimonious parameterization, PCART enables more powerful detection of arcs and more robust parameter inference at the cost of an only modest increase in running time due to efficient PCART optimization algorithms. We will show how the empirical results support this hypothesis.

Parts of this work have been published in a preliminary form in conference proceedings [26]. While the conference paper only considered categorical and continuous variables, the present article also considers ordinal variables—this amounts to a major extension to both the theoretical and empirical contributions. We also revise the general exposition and substantially extend the experimental study; specifically, the studies on predictive performance and hyperparameter sensitivity are new additions. Furthermore, we now also make our implementation of the presented algorithms for computing local scores under the PCART model publicly available.[1]

The remainder of this article is organized as follows. Section 2 reviews the necessary preliminary concepts and notation regarding BNs, CARTs, and the Bayesian approach to learning. We introduce the PCART model in Section 3 and our exact algorithms for learning a PCART from data in Section 4. Section 5 is devoted to empirical studies. Last, we conclude in Section 6. For ease of reading, some detailed material is postponed to the appendix.

## 2. Preliminaries

This section reviews some basics of BNs and decision trees, including the Bayesian approach to learning them from data. We will focus on the key terminology and notation needed in later sections; for a more comprehensive discussion, the reader is referred to the literature on graphical models [6, Chs. 3, 5, and 18] and score-based learning of decision trees [19,21].

### 2.1. Bayesian networks

Consider a multivariate probability distribution $p(X_1, X_2, \ldots, X_n)$, where each variable $X_v$ takes values from some set $S_v$. We write $V$ for the index set $\{1, 2, \ldots, n\}$ and index by subsets: e.g., if $P \subseteq V$, we write $X_P$ for the tuple $(X_v)_{v \in P}$ and $S_P$ for the Cartesian product $\times_{v \in P} S_v$; here we follow the convention that the indices are taken in increasing order.

We say that the distribution $p$ and a directed acyclic graph (DAG) $G = (V, A)$ form a BN if the joint distribution factorizes into a product of the CPDs $p(X_v | X_{G_v})$, where $G_v = \{u : (u, v) \in A\}$ is the set of *parents* of $v$ in $G$. When referring to a particular CPD, we call $X_v$ the *response* variable and each $X_u$, with $u \in G_v$, a *predictor* variable.

There are various ways to parameterize a CPD $p(X_v | X_P)$, with $P \subseteq V \setminus \{v\}$. Here and henceforth we may index the predictor variables simply by $P$ when there is no need to consider the DAG on $V$. Among the simplest ones are CPTs, which assign a distinct parameter for each joint configuration of the involved variables, assuming their state spaces are finite. In what follows, we will focus on more sophisticated CART models, which generalize the tabular CPTs.

### 2.2. Classification and regression trees

In CART models, the structure is represented as a decision tree. A decision tree over a multidimensional space $S_P$ is a recursive, axis-parallel, binary partition of the space. More formally, let $T$ be a rooted binary tree where each node $R$ is a subset of $S_P$ of the form $\times_{u \in P} R_u$. We call $T$ a *decision tree* of $S_P$ if its root is $S_P$ and every inner node $R$ and its two children $R'$ and $R''$ differ in exactly one dimension $u \in P$ for which $\{R'_u, R''_u\}$ is a partition of $R_u$. The "decision" associated with node $R$ is whether the variable $X_u$ belongs to $R'_u$ or $R''_u$. Consequently, the leaves of $T$ partition $S_P$. A decision tree $T_v$ of $S_P$ and a CPD $p(X_v | X_P)$ are *compatible* with each other if, for all leaves $R$, we have that $X_v$ is independent of $X_P$ conditionally on the event $X_P \in R$. In other words, the CPD is constant over each leaf $R$ and piecewise constant over $S_P$.

In order to parameterize the distribution of the response variable $X_v$ in each leaf, we distinguish between a discrete and a continuous distribution. In this work, we focus on two standard cases:

*Categorical distribution:* If $S_v$ is finite, we let $\theta_x$ be the probability that $X_v = x$, for each $x \in S_v$.
*Normal distribution:* If $S_v$ is the set of reals, $X_v$ is normally distributed with mean $\mu$ and variance $\sigma^2$.

In the context of BNs where each variable $X_v$ appears as the response, we understand that the parameters are distinct for all $v$ and all leaves of $T_v$, and omit emphasizing this in the notation. Depending on whether the response is categorical or continuous we call the decision tree equipped with the CPD a *classification tree* or a *regression tree*, generally a *CART*.

What is maybe not immediate from the above description is that we also model ordinal responses by categorical distributions: we enforce the parameters $\theta_x$ be equal for some subsequent values $x$. We give details in Section 2.5.

[1] The software, written in the C++ language, is publicly and freely available at https://github.com/ttalvitie/pcart.

### 2.3. Bayesian structure learning

To learn the model structure from data, we take a Bayesian score-based approach. We only note in passing that the derivations of Sections 3 and 4 also apply to other scoring schemes, especially penalized maximum-likelihood scores.

Consider a set of data points $X^i = (X_1^i, X_2^i, \ldots, X_n^i)$, $i = 1, 2, \ldots, N$. We model the data points as independent draws from a distribution $p$ whose structure, namely a DAG $G = (G_v)_{v \in V}$ with a decision tree $T_v$ of $S_{G_v}$ for each $v \in V$, are unknown; the parameters of the CPDs are unknown as well. We will describe a modular prior of the structure and the parameters and obtain the score as the posterior probability of the structure, by marginalizing out the parameters.

Consider a decision tree $T_v$ of $S_P$; in our context $P = G_v$. For each leaf of $T_v$, we assign standard conjugate priors for the parameters of the categorical and normal distributions:

*Dirichlet:* Let $\theta \sim \text{Dirichlet}(\alpha)$, where $\alpha = (\alpha_x)_{x \in S_v}$ are hyperparameters.
*Normal-inverse-gamma:* Let $\sigma^2 \sim \text{IG}(\nu/2, \nu\lambda/2)$ and $\mu|\sigma^2 \sim \text{N}(\bar{\mu}, \sigma^2/a)$, where $\nu$, $\lambda$, $\bar{\mu}$, and $a$ are hyperparameters.

When an ordinal response is modeled by a categorical distribution, a Dirichlet prior is, however, only applied after the values are partitioned into some number of bins; we give details in Section 2.5.

While we could set the hyperparameters separately for each response variable $X_v$, in our experiments we will focus on the case where all are set to constant values; see Section 5.1 for the default values we used in our experiments and Section 5.6 for a sensitivity analysis.

By letting the parameters of all leaves of $T_v$ be independent, we obtain computationally convenient formulas for the marginal likelihood function. For a leaf $R$, let $I_R = \{i : X_P^i \in R\}$ be the index set of the data points that belong to $R$. If $X_v$ is categorical, we obtain the marginal likelihood of $T_v$ as

$$\ell_v(T_v) = \prod_{R \text{ leaf of } T_v} \frac{\Gamma(\sum_x \alpha_x)}{\Gamma(N_R + \sum_x \alpha_x)} \prod_x \frac{\Gamma(N_{Rx} + \alpha_x)}{\Gamma(\alpha_x)},$$

where $N_R = |I_R|$ and $N_{Rx} = |\{i \in I_R : X_v^i = x\}|$. If $X_v$ is continuous, we get

$$\ell_v(T_v) = \prod_{R \text{ leaf of } T_v} \pi^{-N_R/2} (\lambda\nu)^{\nu/2} \frac{\sqrt{a}}{\sqrt{N_R + a}} \frac{\Gamma((N_R + \nu)/2)}{\Gamma(\nu/2)} (s_R + t_R + \nu\lambda)^{-(N_R+\nu)/2},$$

where $s_R = (N_R - 1)\sigma_R^2$, $t_R = N_R\, a(\mu_R - \bar{\mu})^2/(N_R + a)$, and $\mu_R$ is the sample mean and $\sigma_R^2$ the sample variance of the $N_R$ values $X_v^i$ with $i \in I_R$ [21, Eq. 14]. For an ordinal $X_v$ we derive the likelihood function and discuss its computational complexity in Section 2.5.

Furthermore, we let the parameters be independent for each $v$, so that the marginal likelihood of the structure $(G_v, T_v)_{v \in V}$ factorizes into a product of the local marginal likelihoods $\ell_v(T_v)$. We also assign a modular structure prior

$$p\big((G_v, T_v)_{v \in V}\big) \propto \prod_{v \in V} \pi_v(G_v, T_v),$$

where each $\pi_v$ is some (unnormalized) distribution. In the literature, various priors have been proposed both for DAGs (see, e.g., Angelopoulos and Cussens [27] and references therein) and for decision trees [19,21]. Our choices for the priors depend on the restrictions we make to the considered decision trees, and will be described in Section 3.

From an algorithms standpoint, the structure learning problem now becomes that of finding a structure $(G_v, T_v)_{v \in V}$ so as to maximize the posterior probability

$$\text{constant} \times \prod_{v \in V} \pi_v(G_v, T_v)\, \ell_v(T_v). \tag{1}$$

As the unspecified (normalizing) constant is the same for all structures, it can be ignored.

### 2.4. Posterior predictive distribution

Given a set of $N$ data points $X^{(N)} := (X^i)_{i=1}^N$, the distribution for a new data point $X^{N+1}$ is called the posterior predictive distribution. We will consider a setting where we first learn a model structure $\hat{M} := (\hat{G}_v, \hat{T}_v)_{v \in V}$ from $X^{(N)}$, and then the distribution of $X^{N+1}$ is obtained conditionally on both the learned structure and the $N$ data points. The distribution of interest is thus $p(X^{N+1}|X^{(N)}, \hat{M}) = p(X^{(N+1)}|\hat{M})/p(X^{(N)}|\hat{M})$, and consequently, the mixed density–mass function of $X^{N+1}$ is obtained as the ratio of the marginal likelihoods of $\hat{M}$ for $X^{(N+1)}$ and $X^{(N)}$. This, in turn, amounts to a product variable-wise ratios over $v \in V$.

Consider the ratio for a single $v \in V$. Write $y$ for the value of $X_v^{N+1}$ and $R$ for the leaf of $\hat{T}_v$ that matches the values of $X_{G_v}^{N+1}$. If the variable $X_v$ is categorical, we get the expression

$$\frac{\Gamma(N_R + \sum_x \alpha_x)}{\Gamma(N_R + 1 + \sum_x \alpha_x)} \frac{\Gamma(N_{Ry} + 1 + \alpha_y)}{\Gamma(N_{Ry} + \alpha_y)} = \frac{N_{Ry} + \alpha_y}{N_R + \sum_x \alpha_x}.$$

If the variable $X_v$ is continuous, we get the density function of a non-standardized Student's $t$-distribution[2] with $\nu'$ degrees of freedom, location parameter $\bar{\mu}'$, and scale parameter $\lambda'(a'+1)/a'$, where $\nu'$, $\bar{\mu}'$, $\lambda'$, and $a'$ are the posterior counterparts of the hyperparameters $\nu$, $\bar{\mu}$, $\lambda$, and $a$, obtained by

$$a' := a + N_R\,, \quad \bar{\mu}' := (a\bar{\mu} + N_R\mu_R)/a'\,, \quad \nu' := \nu + N_R\,, \quad \lambda' := \big(s_R + t_R + \nu\lambda\big)/(a'\nu')\,,$$

with $s_R$ and $t_R$ as defined before; for a derivation (up to some differences in the parameterization), see, e.g., Murphy's notes [28, Sections 5.5 and 6.5].

We consider the case of an ordinal $X_v$ in the next subsection.

### 2.5. Extension to ordinal response variables

The previous sections focused on the cases of categorical and continuous response variables. We now extend the model to also accommodate ordinal response variables. To this end, we introduce a Bayesian histogram where the bin locations and widths are treated as auxilliary parameters and "integrated away" in the marginal likelihood for each leaf $R$ in a given decision tree $T_v$; by multiplying these leaf scores we get the likelihood $\ell_v(T_v)$ as before.

Consider a response variable $X_v$. We assume that $X_v$ takes a finite number of different values, $S_v = \{1, 2, \ldots, k\}$, with some integer $k$. Similarly to a categorical response, we let $\theta_x$ denote the probability that $X_v$ takes value $x$. However, instead of assigning the vector $\theta$ a Dirichlet distribution, we assign a prior that constrains the parameters $\theta_x$ be equal for some consecutive values $x$. More precisely, given a *segmentation* $B = \{B_1, B_2, \ldots, B_b\}$ of $S_v$, that is, a partition of $S_v$ into sets of consecutive numbers, called *bins*, we let $\theta_x := \beta_j/|B_j|$ for all $x \in B_j$. Now, we let $\beta \sim \text{Dirichlet}(\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(b)})$ with some $\alpha^{(j)}$. Observe that the parameters $\theta_x$ sum up to 1, as required.

To complete the model, we let the prior probability of a segmentation $B$ with $b$ bins be proportional to $\kappa^b$, with some $\kappa > 0$, and let $\alpha^{(j)} := \sum_{x \in B_j} \alpha_x$. Thus the model has $k+1$ hyperparameters: $\alpha_1, \alpha_2, \ldots, \alpha_k$, and $\kappa$.

**Proposition 1.** *The marginal likelihood of a decision tree $T_v$ is given by*

$$\ell_v(T_v) = \prod_{R \text{ leaf of } T_v} w_v(R)\,, \quad \text{with } w_v(R) := \sum_{B \text{ segm. of } S_v} \frac{\kappa^{b-1}}{(1+\kappa)^{k-1}} \frac{\Gamma(\sum_j \alpha^{(j)})}{\Gamma(N_R + \sum_j \alpha^{(j)})} \times \prod_{j=1}^{b} \frac{\Gamma(N_R^{(j)} + \alpha^{(j)})}{\Gamma(\alpha^{(j)})} \binom{N_R^{(j)}}{(N_{Rx})_{x \in B_j}} |B_j|^{-N_R^{(j)}}\,,$$

*where $N_R^{(j)} := \sum_{x \in B_j} N_{Rx}$ is the total number of data points in $B_j$.*

**Proof.** The probability of the data $(N_{R1}, N_{R2}, \ldots, N_{Rk})$ given a segmentation $B = \{B_1, B_2, \ldots, B_b\}$ can be written as

$$\binom{N_R}{N_{R1}, N_{R2}, \ldots, N_{Rk}} \int p(\beta \mid B) \prod_{j=1}^{b} \left(\frac{\beta_j}{|B_j|}\right)^{N_R^{(j)}} d\beta\,,$$

where $p(\beta \mid B)$ is the density function of the Dirichlet distribution. By writing

$$\binom{N_R}{N_{R1}, N_{R2}, \ldots, N_{Rk}} = \binom{N_R}{N_R^{(1)}, N_R^{(2)}, \ldots, N_R^{(b)}} \prod_{j=1}^{b} \binom{N_R^{(j)}}{(N_{Rx})_{x \in B_j}},$$

taking the terms $|B_j|^{-N_R^{(j)}}$ out of the integral, and using the known closed-form expression for the remaining integral, we get the summand in $w_v(R)$, without the factor $\kappa^{b-1}/(1+\kappa)^{k-1}$, which is the prior probability of $B$; the unnormalized prior probabilities $\kappa^b$ sum up to $\kappa(1+\kappa)^{k-1}$, since there are exactly $\binom{k-1}{b-1}$ segmentations with $b$ nonempty bins. □

**Proposition 2.** *Given the counts $N_{R1}, N_{R2}, \ldots, N_{Rk}$, the term $w_v(R)$ can be computed in time $O(k^2 \log N_R)$.*

[2] The density function $f(y)$ of the non-standardized Student's $t$-distribution with $d$ degrees of freedom, location parameter $m$, and scale parameter $s$, is given by $f(y) = \frac{\Gamma\left((d+1)/2\right)}{\Gamma(d/2)} (\pi d s)^{-1/2} \left(1 + \frac{(y-m)^2}{ds}\right)^{-(d+1)/2}$. Note: in the literature the scale parameter is sometimes defined as the square root of $s$.

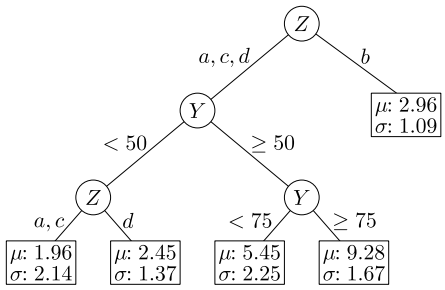



**Fig. 1.** An example of a PCART model. It represents the conditional probability density $p(X|Y, Z)$, where $X \in (0, 10)$, $Y \in (0, 100)$, and $Z \in \{a, b, c, d\}$.

**Proof.** Since $\sum_j \alpha^{(j)}$ equals $\alpha_+ := \sum_x \alpha_x$ for all segmentations $B$, the term $\kappa^{-1}(1+\kappa)^{1-k}\Gamma(\alpha_+)/\Gamma(N_R + \alpha_+)$ in the summand does not depend on $B$ and can be factored out. The remainder of the summand factorizes into a product of $b$ terms, one for each $B_j$. In total, there are $\binom{k+1}{2}$ different terms, one for each bin $[s, t] := \{s, s+1, \ldots, t\}$ with $1 \le s, t \le k$. To show how these terms can be precomputed within the claimed time, denote by $N_R^{(s,t)}$ and $\alpha^{(s,t)}$ the corresponding numbers $N_R^{(j)}$ and $\alpha^{(j)}$, i.e., when $B_j = [s, t]$. Observe first that the numbers $N_R^{(s,t)}$ and $\alpha^{(s,t)}$ are simple sums that can be computed incrementally, i.e., by adding $N_{Rt}$ and $\alpha_t$ to the already computed $N_R^{(s,t-1)}$ and $\alpha^{(s,t-1)}$. Thus, assuming the values of the gamma function have been precomputed, their ratio can be computed in constant time per bin. Similarly, the multinomial coefficient is updated by multiplying it by the binomial coefficient $\binom{N_R^{(s,t)}}{N_{Rt}}$, which we assume to be accessible in constant time (e.g., via three evaluations of the gamma function). Finally, the bin length $t - s + 1$ is raised to the exponent $N_R^{(s,t)}$ by repeated squaring in time $O\big(\log N_R^{(s,t)}\big)$. Let $f(s, t)$ denote the precomputed factors, $1 \le s, t \le k$. Consider the function $g(t)$ defined by $g(0) := 1$ and $g(t) := \sum_{s=1}^{t} g(s-1) f(s, t)$ for $1 \le t \le k$. We have that $w_v(R) = g(k)\,\Gamma(\alpha_+)\big/\Gamma(N_R + \alpha_+)$. It remains to observe that $g(k)$ can be evaluated in time $O(k^2)$. □

We can reduce the running time to $O(k^2)$ by precomputing the powers $l^m$ for all $1 \le l \le k$ and $1 \le m \le N$. Even thought the precomputation takes time and space or order $kN$, this can be advantageous when $w_v(R)$ is computed for many $v$ and $R$.

For the probability mass function of the posterior predictive distribution, we are not aware any "closed-form formula," but the following result is immediate and sufficient for the computational purposes:

**Proposition 3.** *Given a decision tree $T_v$ of $S_v$ and $N$ data points, the response equals $y$ in a new data point with probability $w'_v(R)/w_v(R)$, where $R$ is the unique leaf of $T_v$ that matches the new data point, $w_v(R)$ is as defined in Proposition 1, and $w'_v(R)$ is the same when $N_{Ry}$ is replaced by $N_{Ry} + 1$.*

## 3. Partition–dyadic CART

*Partition–dyadic CART (PCART)* is a subclass of the CART model, which we obtain by restricting the way in which the range of a continuous predictor variable is recursively partitioned by the decision tree. Specifically, we assume the range is an interval and only allow halving it in its midpoint, whence the name dyadic splits. For the number of recursive dyadic splits per variable we set an upper bound, *the maximum number of splits*, denoted by $s$. In contrast, for a categorical predictor we allow arbitrary (binary) partitioning. Fig. 1 shows an example of a PCART model.

PCART generalizes *dyadic decision trees* [23], which assume all predictors be continuous (or binary) and the response be categorical. The motivation for dyadic splits stems from their ability to guarantee both statistical and computational efficiency: dyadic decision trees enable computationally feasible global optimization on data sets with up to around a dozen of predictor variables, and they achieve (in a minimax sense) nearly optimal rates of convergence for a range of classification problems [23,24].

### 3.1. Structure prior

Having fixed the family of decision trees, we proceed to specify a prior distribution over the family. To conform to the modular structure of the likelihood function, we construct a prior where the probability $p(T)$ of a tree $T$ factorizes over the leaves of $T$. This yields a decomposable scoring function, which property is crucial for the optimization algorithm we give in the next section.

We consider priors that assign $p(T) \propto \gamma^{l(T)}$, where $l(T)$ is the number of leaves of $T$ and $0 < \gamma \le 1$ is a constant independent of $T$. Importantly, we do not set $\gamma$ to any fixed value, but we instead let $\gamma$ depend on the number of splits possible in each inner node. Without such dependency, the prior could distribute nearly all probability to very large decision trees, as their number overwhelms the number of small trees.

More precisely, we set $\gamma = 1/(4C)$, where $C$ is the total number of possible splits at an inner node (ignoring splits made in other inner nodes). We have that $C$ is a sum where each continuous predictor contributes 1, each categorical predictor with $k$ possible values contributes $2^{k-1} - 1$, and each ordinal predictor with $k$ possible values contributes $k - 1$. The factor 4 comes from the number of ordered full binary trees with $i$ inner nodes, which is given by the $i$th Catalan number and equalling to $4^i$ up to a factor polynomial in $i$. We can interpret the prior as first drawing the number of leaves from a nearly uniform distribution, then drawing a random full binary tree with the given number of leaves, drawing a random split independently for each inner node of the tree, and accepting the tree if it is a valid decision tree.

For an illustration, consider the decision tree shown in Fig. 1. In this case we have one categorical predictor with 4 possible values and one continuous predictor. Consequently, the constant $C$ equals $2^{4-1} - 1 + 1 = 8$, whence $\gamma = 1/32$. Thus, the prior probability of the shown tree with 5 leaves is $32^4 = 2^{20}$ times smaller than the prior probability of the independence model, i.e., the tree where the root is the only leaf.

### 3.2. PCART as a local model

To incorporate PCART as local models in BNs, we need to specify the components $\pi_v(G_v, T_v)$ of the structure prior. Let $P = G_v \subseteq V \setminus \{v\}$ with $|P| \leq D$; the *maximum indegree* parameter $D$ controls the computational and statistical complexity of the model. For brevity, write $T$ for $T_v$. We use the factorization

$$\pi_v(P, T) = \pi_v(P)\, \pi_v(T|P)\, c(P)\,,$$

where $c(P)$ normalizes $\pi_v(T|P)$ into a probability distribution for $T$. It remains to specify $\pi_v(P)$ and $\pi_v(T|P)$. We consider two different choices for both $\pi_v(T|P)$.

Our first choice is to set $\pi_v(P) = 1$, i.e., the prior is *nearly uniform over parent sets* of size at most $D$.[3] This is the simplest and most "ignorant" prior, which treats all DAGs equally a priori; for a deeper discussion of graph priors we refer to Angelopoulos and Cussens [27] and references therein. We get the basic variant of our model:

*PCART (partition–dyadic CART):* Let $\pi_v(T|P) = (4C)^{-l(T)}$ for all decision trees $T$ of $S_P$ with at most $s$ dyadic splits per continuous variable.

Our second choice is given by $\pi_v(P) = 1\big/\binom{n-1}{|P|}$, i.e., the prior is *nearly uniform over the sizes of the parent sets*.[4] This prior avoids concentrating almost all prior probability on the densest graphs; see, e.g., Friedman and Koller [29] for usage and discussion of this prior. We will call this the *penalized* variant and refer to it as *PCARTp*.

Through the likelihood function, the model defines a joint local score $\pi_v(G_v, T_v)\, \ell_v(T_v)$. This, in turn, induces a local score for the parent set, $g_v(G_v)$, obtained by maximizing the joint local score over $T_v$. The roles of these two kinds of local scores in structure learning will be clarified in the next section (Proposition 4). We note that the resulting global scoring function for DAGs, obtained by taking a product of the parent-wise local scores, is not score equivalent, that is, two DAGs that encode the same conditional independence relations may get different scores. It is an open question, whether there are any nontrivial constraints under which we would achieve score equivalence.

### 3.3. Full table variants and adaptive discretization

Recall that the standard CPD model in BNs is the full table model, where each configuration the parent variable's states is assigned a dedicated categorical (multinomial) distribution. For comparing PCART to this standard model, we next extend the full table model also to cases where the response or some predictor variables are not categorical.

If all predictor variables are categorical, the PCART and PCARTp models we just defined have the following tabular counterparts:

*FT (full table):* Supposing all predictor variables are categorical, let $F$ be a maximal decision tree of $S_P$ where each leaf is a singleton $\{x_P\}$ with $x_P \in S_P$. We let $\pi_v(T|P) = 1$ if $T = F$ and $\pi_v(T|P) = 0$ otherwise. In addition, $\pi_v(P) = 1$.
*FTp (penalized full table):* Like FT but with $\pi_v(P) = 1\big/\binom{n-1}{|P|}$.

In other words, these models introduce, for each joint configuration of the predictor variables' states, a dedicated categorical or normal distribution. In the former case this coincides with a standard CPT; in the latter case, it coincides with a standard conditional linear Gaussian model, albeit of a somewhat degenerate form.

To make the tabular variants applicable to continuous and ordinal predictors as well, we introduce the following discretization scheme. For each continuous or ordinal predictor $u \in P$, discretize its range of values into $k_u$ bins matching

[3] The prior is uniform over DAGs but not exactly uniform over the possible parents for a fixed node, because a smaller set of parents is compatible with a larger number of DAGs.

[4] The prior is not exactly uniform because sparser directed graphs are acyclic with higher probability.

the $k_u$ quantiles of the empirical distribution. The numbers $k_u$ are free structural parameters of the model, taking values between 2 and some maximum $M$ (we used $M = 7$ in our experiments). We assign uniform prior over the joint values $(k_u)_{u \in P}$.

This discretization scheme has two important features. One is that it considers joint discretizations of the predictors in relation to the response, as opposed to some simpler schemes that consider the variables separately or in predictor-response pairs. The other feature is that the scheme does not fix any single discretization of a continuous variable in the BN, but the discretization may vary between the parent sets to which the variable belongs. Both features enhance the power of the discretization scheme as compared to the simple schemes. The prize we have to pay is that the computational cost of searching for optimal discretizations grows, as we need to consider all possible configurations $(k_u)_{u \in P}$ for each $P$ we encounter; we will address this issue in Section 4.

## 4. Algorithms

The state-of-the-art algorithms for finding a globally optimal BN structure proceed in two steps:

**Step 1 (Candidate parent set scoring and pruning)** For each $v \in V$ and $P \subseteq V \setminus \{v\}$ with $|P| \leq D$ compute the local score $g_v(P)$, defined in terms of the local model and the data (e.g., prior and likelihood). Prune every set that has a subset with at least as large score. Return the remaining parent sets and their scores.

**Step 2 (DAG search)** Return a DAG $G$ that maximizes the total score $\prod_{v \in V} g_v(G_v)$.

For Step 2 we may use existing complete solvers, e.g., ones based on integer linear programming [15] or A* [14]. In our experiments we used the integer linear programming based solver *GOBNILP* [15], since in our preliminary study it appeared to scale well. It is worth noting that we cannot use the constraint programming based solver of van Beek and Hoffmann [16] because it assumes score equivalence, i.e., that the scoring function gives the same score for all Markov-equivalent DAGs.[5]

Our concern is Step 1. To agree with the definition of the scoring function (1), we set the local scores by

$$g_v(P) = \max_T \left\{ \pi_v(P, T)\, \ell_v(T) \right\},$$

where $T$ runs through all local structures on the predictors $P$. Indeed, it is easy to see that this assignment guarantees that the algorithm will find a globally optimal structure:

**Proposition 4** *(Optimality). Let $G$ be a DAG that maximizes $\prod_{v \in V} g_v(G_v)$. Then, for each $v \in V$, there exists a $T_v$ such that the structure $(G_v, T_v)_{v \in V}$ maximizes the scoring function (1).*

**Proof.** For any fixed DAG $G$ the choices for the local trees $T_v$ are independent. Therefore, we have that

$$\prod_{v \in V} \max_{T_v} \left\{ \pi_v(G_v, T_v)\, \ell_v(T_v) \right\} = \max_{(T_v)_{v \in V}} \left\{ \prod_{v \in V} \pi_v(G_v, T_v)\, \ell_v(T_v) \right\}.$$

Thus, choosing for each $G_v$ a tree $T_v$ that maximizes $\pi_v(G_v, T_v)\, \ell_v(T_v)$ yields a structure $(G_v, T_v)_{v \in V}$ that maximizes the scoring function (1) □

In the case of PCART each local score requires optimization over a large number of decision trees. Moreover, the number of required local scores grows rapidly with the total number of variables $n$, being $n \sum_{d=0}^{D} \binom{n-1}{d}$ for maximum indegree $D$. In the remainder of this section we detail a dynamic programming algorithm to solve this optimization problem. We have developed a similar algorithm for computing the required normalizing term $c(P)$, which is a *sum* over decision trees; however, as this term is independent of the data and faster to compute, we will omit a more detailed description.

For FT (maximal decision tree), it suffices to enumerate the possible joint discretizations of the continuous predictors. For fixed $v$ and $P$, this requires at most $(M-1)^{|P|}$ evaluations of a closed-form expression; for moderate values of the maximum number of bins $M$, the computational cost is small compared to that of PCART structure optimization.

### 4.1. PCART structure optimization

We give a dynamic programming algorithm to find an optimal PCART structure for a response $v$ and a set of predictors $P \subseteq V \setminus \{v\}$. The algorithm applies to any objective function that factorizes into a product of local scores $f(R)$ over the leaves $R$ of the tree. Our Bayes score is clearly of this form as both the prior and the likelihood factorize over the leaves. For every possible decision tree node $R = \times_{u \in P} R_u$, the algorithm computes, in a top-down fashion, the best score $f(R)$ for the subtree below $R$. Thus the score of the root, $f(S_P)$, is the maximum score over all PCART structures.

[5] We hereby correct the mistake of using *CPBayes* in the preliminary version of this work [26].

**Algorithm 1** PCART optimization.

**Input:** Response $v$, predictors $P \subseteq V \setminus \{v\}$ and data $X = (X^1, X^2, \ldots, X^N)$

$\gamma \leftarrow 1/4 \left[ \sum_{u \in P \text{ continuous}} 1 + \sum_{u \in P \text{ categorical}} (2^{|S_u|-1} - 1) + \sum_{u \in P \text{ ordinal}} (|S_u| - 1) \right]^{-1}$ ▷ *Compute the structure prior factor*

**function** LEAF-SCORE($y^1, y^2, \ldots, y^{N'}$)
  **return** $\gamma \times$ marginal likelihood of response values $y$ computed using the formulas in Sections 2.3 and 2.5

Let $\mathcal{M}$ be an associative array
**function** SUBTREE-SCORE($R$, $Y = (Y^1, \ldots, Y^{N'})$) ▷ *Computes $f(R)$ given that $Y$ is the set of data points in node $R$*
  **for** each categorical or ordinal predictor $u \in P$ **do** ▷ *Remove missing values from discrete variables*
    $R_u \leftarrow \{Y_u^1, Y_u^2, \ldots, Y_u^{N'}\}$
  **if** $\mathcal{M}$ does not contain element $R$ **then**
    $\mathcal{M}_R \leftarrow$ LEAF-SCORE($Y_v^1, Y_v^2, \ldots, Y_v^{N'}$) ▷ *Consider the case where $R$ is a leaf node*
    **if** $N' > 0$ **then**
      **for** each continuous predictor $u \in P$ **do** ▷ *Consider all splits in continuous variables*
        Let $R_u = [x, y)$, $S_u = [x', y')$
        **if** $(x - y)/(x' - y') > 2^{-s}$ **then**
          Let $R'$, $R''$ be such that $R'_u = [x, (x+y)/2)$, $R''_u = [(x+y)/2, y)$ and $R'_{P\setminus\{u\}} = R''_{P\setminus\{u\}} = R_{P\setminus\{u\}}$
          Partition $Y$ into $(Y', Y'')$ such that $Y'$ contains the data points $Y^i$ satisfying $Y_u^i < (x+y)/2$
          $\mathcal{M}_R \leftarrow \max\{\mathcal{M}_R, \text{SUBTREE-SCORE}(R', Y') \times \text{SUBTREE-SCORE}(R'', Y'')\}$
      **for** each categorical predictor $u \in P$ **do** ▷ *Consider all splits in categorical variables*
        **for** each partition $\{C', C''\}$ of $R_p$ into two nonempty sets **do**
          Let $R'$, $R''$ be such that $R'_u = C'$, $R''_u = C''$ and $R'_{P\setminus\{u\}} = R''_{P\setminus\{u\}} = R_{P\setminus\{u\}}$
          Partition $Y$ into $(Y', Y'')$ such that $Y'$ contains the data points $Y^i$ satisfying $Y_u^i \in C'$
          $\mathcal{M}_R \leftarrow \max\{\mathcal{M}_R, \text{SUBTREE-SCORE}(R', Y') \times \text{SUBTREE-SCORE}(R'', Y'')\}$
      **for** each ordinal predictor $u \in P$ **do** ▷ *Consider all splits in ordinal variables*
        **for** each partition $\{C', C''\}$ of $R_p$ into two nonempty sets such that $\max C' < \min C''$ **do**
          Let $R'$, $R''$ be such that $R'_u = C'$, $R''_u = C''$ and $R'_{P\setminus\{u\}} = R''_{P\setminus\{u\}} = R_{P\setminus\{u\}}$
          Partition $Y$ into $(Y', Y'')$ such that $Y'$ contains the data points $Y^i$ satisfying $Y_u^i \in C'$
          $\mathcal{M}_R \leftarrow \max\{\mathcal{M}_R, \text{SUBTREE-SCORE}(R', Y') \times \text{SUBTREE-SCORE}(R'', Y'')\}$
  **return** $\mathcal{M}_R$
**Output:** SUBTREE-SCORE($S_P$, $X$)

The best subtree score for node $R$ is obtained by considering all possibilities for the node: the node may be a leaf node, or it is an inner node with two children $R'$ and $R''$. The score of a leaf is computed from the training data points contained in $R$ as specified in Sections 2 and 3. For an inner node the score is computed recursively as the product $f(R')f(R'')$ of the scores of the child subtrees. The recursion caches the already computed subtree scores to avoid computing the same score multiple times. After computing the scores, an optimal decision tree is extracted by tracing the splits that gave the optimal scores.

With our choice of scores, it happens that a node $R$ that contains no training data points has to always be a leaf node, and the recursion does not need to advance further. Furthermore, if for some categorical or ordinal variable $u$ there are some values in $R_u$ that do not appear in the training data that is contained in $R$, we can simply consider a smaller node with the unused values removed. This may cause $R_u$ to consist of nonadjacent values for some ordinal variable $u$, but it does not affect the result since we only consider splits respective to the ordering of the variable, that is, all the values in $R'_u$ are smaller than the values in $R''_u$. These optimizations significantly reduce the number of nodes the algorithm has to consider. The pseudocode for the algorithm is shown in Algorithm 1.

We next give an asymptotic upper bound for the number of nodes $K$ the algorithm considers and for the running time of the algorithm. The bounds will be rather loose: they fail to account for all the optimizations and special properties of the data sets.

**Proposition 5** *(Time complexity). Algorithm 1 considers $K \leq \min\left\{N2^{A(k-1)-2C}(s+1)^B, 2^{Ak+B(s+1)-C}\right\}(r+1)^{2C}$ tree nodes, and the total time complexity is $O\left(n(2^k A + B + rC)K \log K\right)$, where $A$, $B$, and $C$ are the numbers of categorical, continuous, and ordinal predictor variables, respectively, $k$ is the maximum number of categories, $s$ the maximum number of recursive splits per continuous variable, and $r$ the maximum number of values of any ordinal variable.*

**Proof.** Counting the possible nodes simply as the product of at most $2^k$ possible value subsets $R_u \subset S_u$ per categorical variable $u \in P$, at most $2^{s+1}$ subsets per continuous variable, and at most $\binom{r+1}{2} \leq (r+1)^2/2$ subsets per ordinal variable, gives an upper bound $K \leq 2^{Ak+B(s+1)-C}(r+1)^{2C}$. In the case when the number of training data points $N$ is low, we get a better bound similarly to Blanchard et al. [24] (who assume all variables be continuous): Since the algorithm only considers nodes that contain at least one training data point, we can bound the number of nodes by the total number of containing nodes for each data point. The number of value subsets that contain a given value is at most $2^{k-1}$ for each categorical variable, $s+1$ for each continuous variable, and $(r+1)^2/4$ for each ordinal variable, yielding an upper bound $K \leq N2^{A(k-1)-2C}(s+1)^B(r+1)^{2C}$.

Now, for every node the algorithm considers all possible splits—at most $2^k$ for each categorical variable, one for each continuous variable, and at most $r-1$ for each ordinal variable—and finds the score from an associative array data structure in $O(\log K)$ time. Thus the total time complexity is $O\left(n(2^k A + B + rC)K \log K\right)$. □

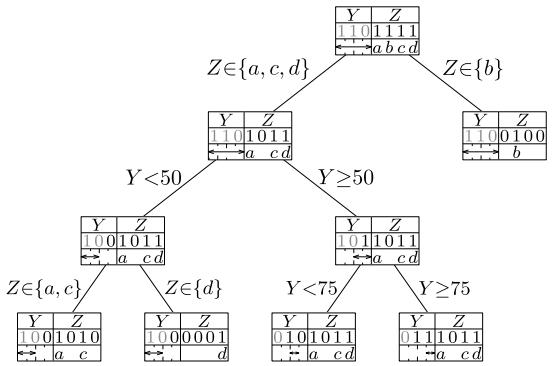



**Fig. 2.** An illustration of the binary node representation. The example corresponds to the PCART in Fig. 1, and consists of two parts: 3 bits for the continuous variable $Y$ (here the maximum number of splits is 2) and 4 bits for the categorical variable $Z$. For the continuous variable, the length of the leading 1-bits and the following 0-bit (in gray) indicates the length of the interval, and the remaining bits encode the position of the interval.

For an illustration of the complexity bound, consider the decision tree shown in Fig. 1. In this case we have one categorical predictor with 4 possible values and one continuous predictor. By Proposition 5, the time complexity is $O(nK \log K)$, where the number of considered tree nodes $K$ is at most $\min\{64 \cdot N(s+1), 256 \cdot 2^{s+1}\}$.

### 4.2. Implementation tricks

The dynamic programming algorithm for finding an optimal PCART structure is easily implemented using recursion, as described in Section 4.1. However, by carefully tuning the memory representation of nodes, we are able to optimize our C++ implementation to run an order of magnitude faster.

Our implementation stores node $R$ as a bit vector that is a concatenation of fixed-length bitvectors, each of which stores the value subset $R_u$ for a variable $u \in P$. For each continuous variable with maximum number of splits $s$, the $2^{s+1} - 1$ possible intervals are stored as $(s+1)$-bit binary numbers. For each categorical variable $u$, the subset of values $R_u \subseteq S_u$ is stored using $|S_u|$ bits. An example of the representation is shown in Fig. 2. The predictor values of the data points are also encoded as the minimal nodes containing them. In practice, 64 bits suffice for storing the node representations, which means that the operations performed by the algorithm on the nodes translate into efficient bit operations on 64-bit machine words. The data can be partitioned in-place using simple bit operations and because of the compact representation, the data will stay present in CPU caches, making the innermost loops of the algorithm very efficient.

## 5. Empirical studies

We empirically compared the different local models in relation to (a) structure recovery, (b) predictive performance, and (c) running time. To this end, we used both synthetic data generated from known benchmark networks and real data from various domains.

### 5.1. Data sets and hyperparameters

As ground-truth benchmark networks, we used Alarm, Insurance, and Water, all extracted from the *bnlearn* [30] repository.[6] The networks contain 37, 27, and 32 variables respectively, and are thus suitable for finding the globally optimal DAG with modern solvers. However, as these benchmark networks contain only categorical variables, we did not use them for evaluating the performance on mixed variables.

For real data, we extracted all suitable data sets from the UCI machine learning repository [31]; more exactly, we considered all the classification and regression data sets that have at least 100 data points, at least 8 but fewer than 36 variables, and that consist of a single table that is freely available. We preprocessed the data sets by normalizing all the continuous variables to have mean zero and variance one and limiting the number of categories in each categorical variable to 5 by combining the rarest categories into one. We also removed identifier variables and data points that contain missing values, rejecting the data sets where we need to remove a majority of the data points. We also rejected the data sets Mushroom and Parkinsons Telemonitoring, since the network optimization phase using *GOBNILP* took too much time for them. See Table B.5 (Appendix B) for a complete list of the 52 data sets.

We set the user parameters of the models as follows. The hyperparameters of the prior distributions were set as $\alpha = 1/2$ (Jeffreys prior), $\bar{\mu} = 0$, and $a = \nu = \lambda = 1$, except for the sensitivity study (Section 5.6) where we varied the latter three values; for the segmentation prior of ordinal responses we set $\kappa = 1/4$ to approximately balance the prior mass on smaller

[6] The web address is http://www.bnlearn.com/bnrepository.

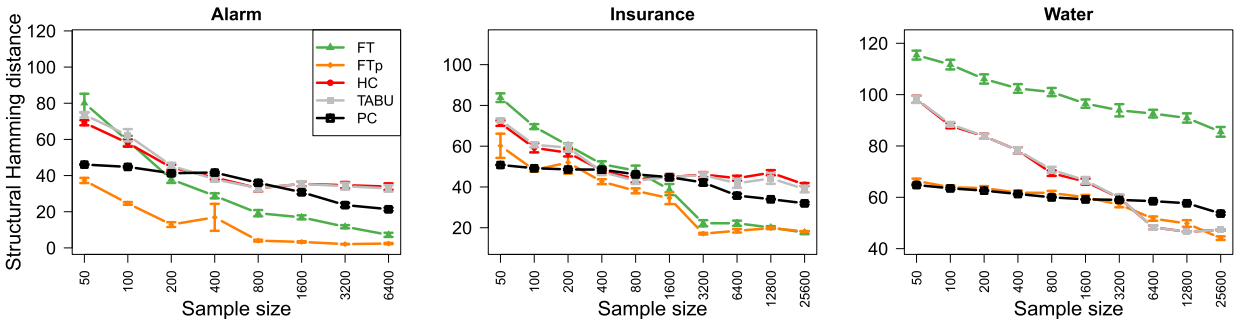



**Fig. 3.** Comparison of globally optimal DAG search (FT, FTp) with heuristic approaches (HC, TABU) and a constraint-based algorithm (PC) for the task of recovering a ground-truth network structure based on data samples of varying size. The evaluation metric is an average structural Hamming distance over ten independent data samples. Error bars indicate (here and in all following plots) the standard error in both directions.

and larger numbers of bins. We set the maximum number of splits per continuous variable to $s = 5$, the maximum number of bins in the full table variant to $M = 7$, and the maximum indegree of a DAG to $D = 4$.

### 5.2. Comparison of exact and heuristic algorithms for Bayesian network structure learning

As a preliminary small-scale study, we investigated the advantage of finding a globally optimal DAG as opposed to a merely high-scoring DAG. Here we restrict ourselves to standard CPT based local scores. To be precise, we compared our implementation of the FT local score (with *GOBNILP* for searching for the globally optimal DAG) with Scutari's *bnlearn* implementation [30] of hill-climbing (HC), TABU search, and the constraint-based PC algorithm. Except for the PC algorithm, all these implementations use the same scoring function (the BD score with hyperparameters set to $1/2$, combined with a uniform structure prior), thus enabling fair comparison to FT. For reference, we also compared to global optimization under the penalized FTp score. We used the PC-implementation with its default parameters: an asymptotic chi-squared test for the mutual information test statistic with a significance level of 0.05.

From the three benchmark networks we sampled data sets of different sizes, ten repetitions each, and learned a network structure for each data set and algorithm. We then computed the structural Hamming distance (SHD) [32], which measures differences up to the equivalence class, between each learned network and the ground truth. We averaged the SHDs for each algorithm and sample size over the ten independent repetitions. The resulting learning curves are shown in Fig. 3.

We observe that FT and FTp outperform the other three approaches for Alarm and Insurance once the sample size exceeds a few hundred data points. For Water, however, FT performs substantially worse than HC and TABU. This can be explained by the observation that the BD score alone (without a nonuniform structure prior) can be a very poor scoring function due to a preference for overly complex models at relatively small sample sizes. In such cases, the inability of the heuristic algorithms to find the rather dense globally optimal DAG becomes a fortunate advantage, as they stop the search at a local optimum that is a relatively sparse subgraph. The poor performance of FT justifies the inclusion of the FTp variant, which uses a structure prior that is known to penalize complexity appropriately [33]. Unfortunately, *bnlearn* does not allow calling HC or TABU with an arbitrary scoring function, so it is not possible to directly compare the globally optimal FTp solution to the corresponding heuristic results.

The constraint-based algorithm (PC) has advantages at very small sample sizes, but it does not learn much from an increased amount of data and eventually recovers the generating network worse than the score-based learning methods. (One might improve the performance of PC by letting its significance level parameter depend on the data size in some appropriate way, but the *blearn* implementation does not do that by default.)

Aside from the rather pathological case of the Water network, the observations from the preliminary study add to previously published evidence for the need of global optimization [17] and justifies our restriction to study learning with local structures solely within the global optimization framework.

### 5.3. The effect of local structure on benchmark network structure recovery

Consider then the same experimental setting as in the previous section, but now for comparing the four local models, PCART(p) and FT(p), and only exact algorithms. We observe (Fig. 4) that both PCART variants yield smaller SHDs than FT for all generating networks and sample sizes. With the sparsity-favoring DAG prior added, FTp becomes competitive for Alarm. A possible explanation is that Alarm has the smallest edge/node ratio of only 1.24, whereas Insurance and Water have 1.93 and 2.06, respectively. A small average indegree leaves little potential for structured CPDs as the majority of CPTs can be accurately represented with a (small) probability table. For Insurance and Water the PCART variants yield faster convergence than FTp.

In summary, PCART(p) appears to have a slight advantage over the full-table variants in recovering the network structure, but not in each and every case. This is not surprising, since there is no reason to believe that every conditional probability

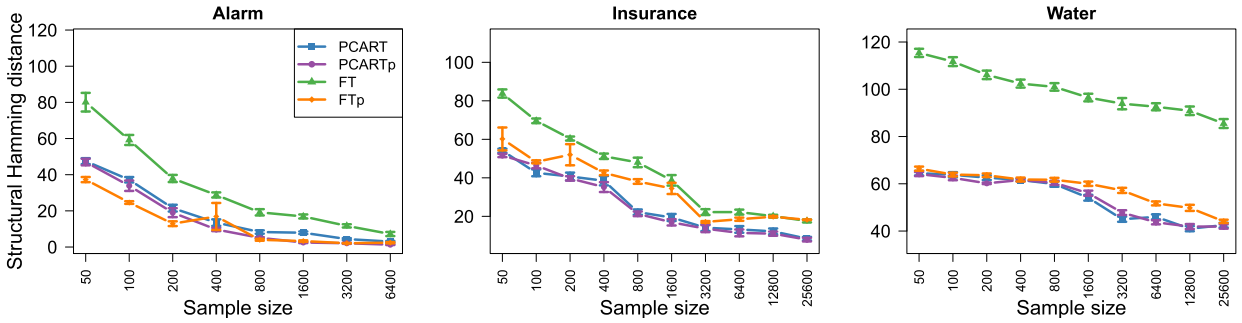



**Fig. 4.** Accuracy of PCART(p) with FT(p) in structure recovery from data samples generated from three benchmark networks.

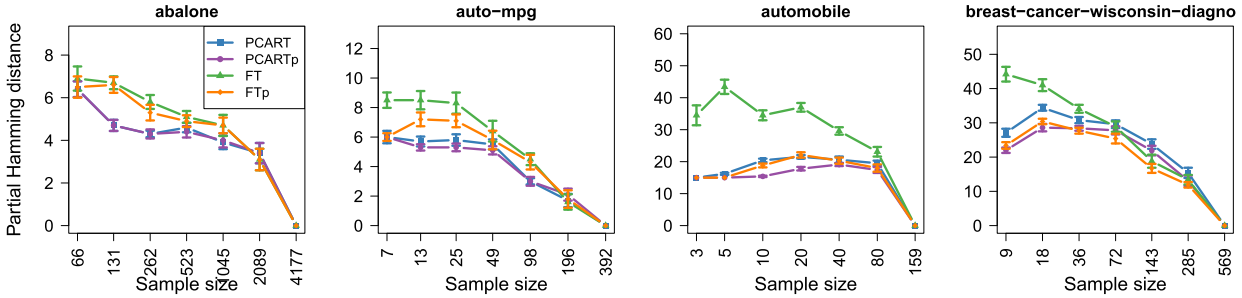



**Fig. 5.** Learning curves produced by the Intersection-Validation method for four UCI data sets. The largest sample size in each plot corresponds to the original data size, for which all methods yield the same partial Hamming distance (of value zero) by definition.

table in the ground-truth networks admits an economical tree-structured representation. Moreover, in terms of SHD it is often beneficial to learn a conservative (sparse) model when the sample size is small [33]. For very small data sets, even attempting to learn a structured local model may yield small structural overfitting effects that outweigh the possibly higher expressiveness of PCART.

### 5.4. Structure recovery on real data

For evaluating the accuracy of structure learning from real data, we apply a very recent method called Intersection-Validation [34]. The method allows us to draw learning curves similar to the examples in Fig. 4 even when no ground-truth DAG is available. The method checks which structural features all learning algorithms (e.g., scoring functions) that are to be compared agree upon when learning from the entire data set, and treats them as surrogate ground truth. Structures learned at smaller subsamples of the entire data set are then compared against this partial network to compute an approximation of SHD, called *partial Hamming distance*.

For each of the 52 UCI data sets under consideration and for each of the four scoring functions, we learn networks from subsamples of $1/2$, $1/4$, $1/8$, $1/16$, $1/32$, and $1/64$ of the original data size and repeat the subsampling process ten times. The resulting averaged learning curves for the four scoring functions in comparison are shown in Fig. 5 for the first four data sets according to an alphabetical ranking of their names, and in Appendix A for the remaining 48 data sets. By visually inspecting the learning curves, we draw similar conclusions as in the benchmark network study from the previous section: FT is widely inferior to the other methods, likely due to learning overly complex models at small sample sizes. PCART local models perform often, though not always, slightly better than FTp.

In order to summarize the results from the 52 data sets in a systematic fashion, we aggregated them by the following procedure. We considered each fraction of the original data size ($1/2$, $1/4$, etc.) as a separate group, irrespective of the absolute sample size. We then made a pairwise comparison of the four local models for each of the six groups. To this end, we used the following assessment: method A performs better/worse than method B (win/loss) if it yields a smaller/larger average partial Hamming distance and if the errors bars in the learning curves do not overlap. If they do overlap, we considered both models to perform equally well (tie).

We show these aggregated statistics in Table 1 for the two most important model comparisons and in Table A.3 for the remaining four cases. We observe that the PCART variants perform consistently better than the full-table models, especially at small sample sizes. This does not only pertain to the comparison to FT, but also to FTp, which demonstrates that structured CPDs help to recover correct structural features with fewer samples, in comparison to full-table models. This is an interesting result as the effect was not that obvious in Fig. 4, which indicates that real data sets tend to have different properties than synthetic data sampled from benchmark networks. An alternative to declaring a tie between methods in the event of overlapping error bars is to use a nonparametric statistical test (Table A.4). The number of ties does here in-

**Table 1**
Summarized method comparison based on the Intersection-Validation method. Sample size refers to the fraction of the size of the original data set. A tie occurs if the error bars of two methods overlap. Otherwise the mean partial Hamming distance decides upon win or loss.

(a) PCARTp vs FTp

| Sample size | Win | Tie | Loss |
|---|---|---|---|
| 1/64 | 21 | 30 | 1 |
| 1/32 | 26 | 24 | 2 |
| 1/16 | 33 | 15 | 4 |
| 1/8 | 29 | 18 | 5 |
| 1/4 | 23 | 22 | 7 |
| 1/2 | 18 | 30 | 4 |

(b) PCART vs FT

| Sample size | Win | Tie | Loss |
|---|---|---|---|
| 1/64 | 50 | 2 | 0 |
| 1/32 | 49 | 3 | 0 |
| 1/16 | 47 | 4 | 1 |
| 1/8 | 40 | 12 | 0 |
| 1/4 | 39 | 10 | 3 |
| 1/2 | 31 | 16 | 5 |

**Table 2**
Summary of prediction results on UCI data. A tie occurs if the error bars of two methods overlap.

| | | | Win | Tie | Loss |
|---|---|---|---|---|---|
| PCART | vs | PCARTp | 1 | 51 | 0 |
| PCART | vs | FT | 20 | 23 | 9 |
| PCART | vs | FTp | 19 | 24 | 9 |
| PCARTp | vs | FT | 21 | 20 | 11 |
| PCARTp | vs | FTp | 18 | 23 | 11 |
| FT | vs | FTp | 0 | 51 | 1 |

crease due to the small sample size (ten repetitions), but among instances with a statistical significant difference, the PCART methods outperform their full table counterparts.

These results on real data further imply that the PCART variants are effective not only for categorical variables, but also in the presence of mixed, or completely continuous data.

### 5.5. Predictive performance

In all previous studies, we viewed the graph structure as the learning target. Another commonly considered target for learning is the joint probability distribution. One can evaluate a learned distribution, e.g., by computing the cross-entropy to the ground truth distribution or, in absence of ground truth, by approximating the cross-entropy by the average log loss in a cross-validation experiment.

We followed this approach for all the 52 UCI data sets and four local models. For each combination we computed the average log loss, called henceforth *log loss* for short, by first finding a globally optimal model structure, including the local structure, in relation to the training data, and then taking the loss with respect to the Bayesian posterior predictive density of each test data point. All log loss values, along with standard errors, are shown in Appendix B. Table 2 shows a win–tie–loss summary based on overlapping error bars and Table B.6 shows an alternative based on a two-sided Wilcoxon-signed rank test [35]. In contrast to the summary statistics for the intersection-validation study, were here observe the significance test to produce less ties than the overlapping error bar criterion, which can be explained by the relatively large sample size of 50 repetitions.

In both cases we see that for prediction it makes virtually no difference whether a sparsity-favoring structure prior is used or not: the penalized and unpenalized versions of PCART and FT perform nearly identically. Models that are overly complex according to SHD can still host the true probability distribution, and estimating it from data is effective unless their CPDs are so large that massive parameter overfitting occurs, which is unlikely even for models learned with FT.

Aside from this aspect, we again observe a slight advantage of PCART compared to full tables, albeit in many cases both types of local models perform equally well. This is not surprising as it has been observed that measuring the learned distribution shows less differences among structure learning methods than measuring the structural features [34].

### 5.6. Hyperparameter sensitivity analysis

All of the previous studies used a fixed set of hyperparameters for the Bayesian priors. While $\alpha = 1/2$ (Jeffreys prior) and $\bar{\mu} = 0$ (mean of a standard normal) can be conceptually motivated, the choice of $a = \nu = \lambda = 1$ was based on simplicity and decent performance in preliminary tests. We now study how varying each of these three parameters (while keeping the two others fixed at their default value) affects the predictive performance under the four local models, both in absolute terms and relative to each other.

The results for the UCI data set concrete-compressive-strength are shown in Fig. 6, whereas the results for 14 additional randomly chosen UCI data sets are shown in Appendix C. We find that varying the hyperparameter may indeed have an effect. Our default values are generally a good choice, albeit they could be improved in some cases.

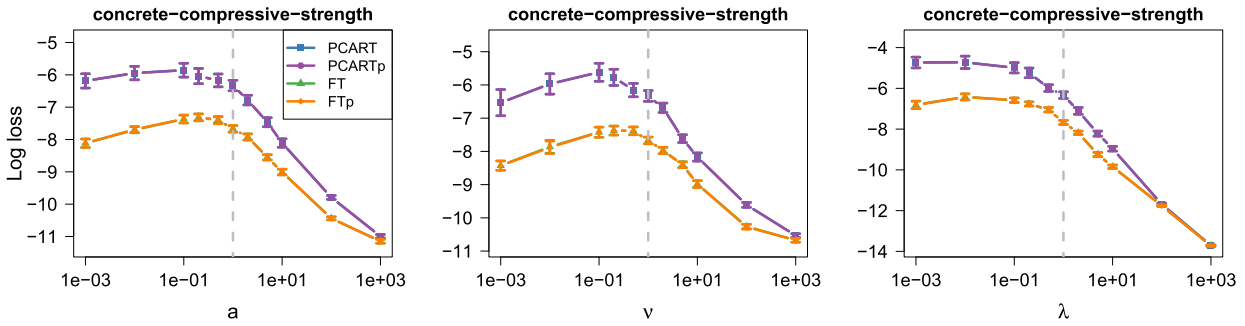



**Fig. 6.** Hyperparameter sensitivity analysis for one UCI data set. Each plot varies one hyperparameter (x-axis) while keeping the other two fixed, showing the effect to predictive performance as measured by log loss (y-axis). Dashed vertical line indicates the default value. Note that for this data set the difference between the penalized and unpenalized algorithm variants is extremely small.

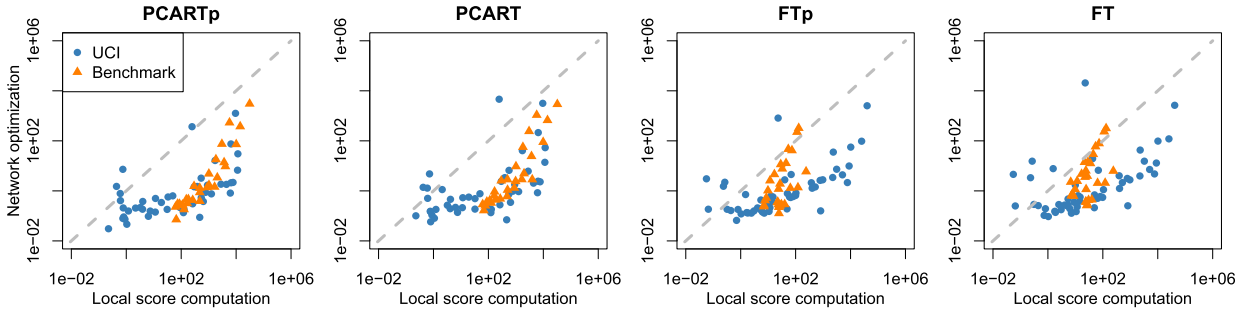



**Fig. 7.** Running time in seconds of Step 1 vs Step 2 under the four local models. See text for a description of the included data sets.

The more important observation, however, is that the decision whether PCART(p) or FT(p) is superior for a particular data set does not heavily depend on the particular choice of hyperparameter values, except for the case of very large values (such as $\lambda = 10^3$ in Fig. 6) where all models perform poorly.

### 5.7. Running times

Last, we investigated how the choice of the local model affects the time requirements for the local score computations (Step 1) and for the DAG optimization (Step 2, by *GOBNILP*). Fig. 7 shows the measured running times for the data sets we sampled from the three benchmark networks (Section 5.3) and for the real UCI data sets we included in the structure recovery study (Section 5.4).

We see that under all four models, Step 1 tends to consume more time than Step 2. This observation highlights the fact that *GOBNILP* solves the DAG optimization step fast on typical problem instances, and thus the local score computations can be a bottleneck, even under the standard CPT model (e.g., FT and FTp on categorical data generated from benchmark networks). Nevertheless, there are also data sets for which Step 1 is faster than Step 2 or almost equally fast; importantly, this holds also under PCART and PCARTp. This observation, in turn, highlights the fact that DAG optimization is hard in general, and therefore one can afford computationally demanding local scores. It is worth noting that the time complexity of the local score computations grows only polynomially in the number of nodes $n$, assuming a constant maximum indegree, whereas the complexity of the DAG optimization step is suspected to grow superpolynomially, the current best wort-case bound being exponential in $n$.

### 5.8. Studies with ordinal variables

In order to benchmark the algorithm variants that can handle ordinal data, we searched for ordinal variables in the 52 UCI data sets from the previous sections. In order to avoid additional discretization, we only considered discrete variables with less than 16 possible values. As a consequence, we treat an attribute such as “Age” (in years), which is ordinal in principle, as continuous variable nevertheless. From all data sets, we chose six data sets that have at least three ordinal variables; see Appendix D.1 for details.

For comparing the ordinal PCART variants (OPCART and OPCARTp) with the ordinal full table models (OFT and OFTp), we repeated the intersection-validation study from Section 5.4 and the prediction study from Section 5.5 and display the results in Appendix D.2 and Appendix D.3 (Table D.7). We observe that the PCART variants continue to perform better than the full table variants on average. Hence, the treatment of ordinal variables as such, instead of being modeled as either continuous or categorical, does not change the relative performance of PCART-based models in relation to their full table counterparts.

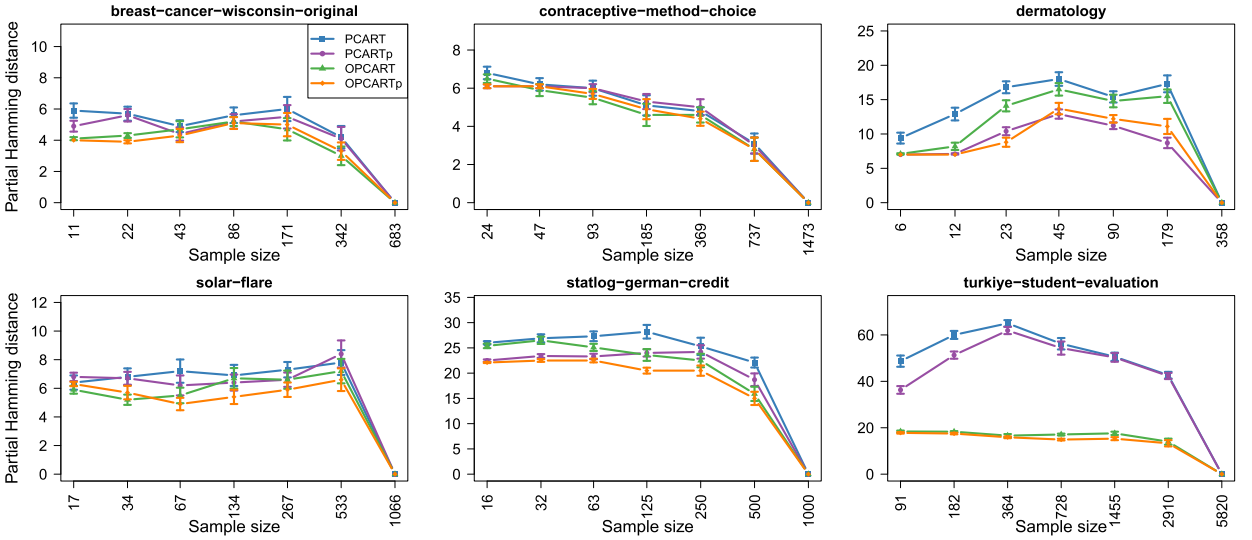



**Fig. 8.** The effect of including ordinal variables on structure learning measured by the intersection-validation method.

Nevertheless, from these results it is yet unclear whether the extension to ordinal variables is practically useful. Hence, we investigated how well the ordinal variants of our algorithm perform in relation to the non-ordinal variants studied in the previous sections. While the predictive probabilities and densities are not comparable any longer as soon as a variable changes its type, the interpretation of the learned DAGs remains the same, allowing a fair comparison among the methods. We thus applied the intersection-validation method similar to Section 5.4 and compared PCART, PCARTp, OPCART, and OPCARTp (Fig. 8). We observe that the ordinal variants yield almost never a substantially worse partial Hamming distance compared to their non-ordinal counterparts, and often there is even a slight improvement. This demonstrates that the extension of our algorithms to ordinal variables pays off indeed.

## 6. Discussion

This work revisited the idea of incorporating decision trees as local models in Bayesian networks. Specifically, we introduced a new model class, PCART, which can handle both discrete and continuous predictor and response variables, takes a Bayesian approach, and admits computationally feasible global optimization in a range of problem sizes. We also introduced an analogous full-table variant, FT, which extends the standard CPT of categorical variables to continuous and ordinal response and predictor variables.

Empirical comparison of the tree-structured PCART and the tabular FT showed a clear advantage of PCART in structure learning and a slight advantage in density estimation. While the results were obtained mainly under a fixed, global choice for the values of the hyperparameters, the results were observed to be robust to changes of these values. That being said, the sensitivity analysis also suggests that the performance of both PCART and FT could be further enhanced by choosing the hyperparameter values separately for each given data set and each variable, e.g., by means of empirical Bayes.

In our experiments we set the parameters that control the computational complexity of the model (maximum indegree, number of splits) so that the computations of the local scores (Step 1) could be completed within a few hours for each data set. This was motivated by the fact that for some of the data sets, the state-of-the-art solvers require hours to find an optimal DAG (Step 2). There are two simple ways to expedite Step 1, if needed: by setting the complexity parameters to lower values, and by running the computations for different parent sets in parallel. An open question is whether one could gain further reductions in the time requirement for PCART optimization by pruning the search space based on appropriate score upper bounds; this method has been successful in a related problem of learning parsimonious context trees [36], which can be viewed as a special case of PCART that is restricted to categorical variables and a sequential ordering of the predictors.

As the presented methods concern structure learning, with PCART as the local model, they do not give any particular way to fix the parameters of the joint distribution. Indeed, we obtained a posterior predictive distribution by “integrating out” the parameters. If one wishes to employ the learned model in probabilistic inference in the standard framework, where each variable is assigned a fixed finite domain, one can use the original domains of categorical and ordinal variables in a straightforward manner, just interpreting the learned PCART models as succinct representations of the full CPT. Continuous variables are more problematic, however, and we are not aware of any representation of the joint distribution that would allow for efficient probabilistic inference. The challenge of representing mixed distributions is by no means specific to PCART models, but notorious more generally [6, Chapter 14]. If one is willing to discretize continuous variables, PCART offers one approach: namely, one may replace continuous variables by ordinal variables, with some moderate number of possible values.

There are also other directions for future work. First, PCART and FT warrant further investigation as alternatives to other approaches to hybrid Bayesian networks, such as hybrid copulas [37], iterative discretization schemes [5], and simpler discretization routines that only consider the marginal or pairwise distributions. Second, the present implementation of PCART assigns the response variables categorical or normal distributions, which is an obvious limitation, particularly for continuous and ordinal responses. The main ideas and the algorithms are, however, not tied to these distributions, and it is straightforward to accommodate other distributions, as long as they are coupled with appropriate priors so as to yield efficient-to-compute leaf scores. Third, the approach we took is not "fully Bayesian" in the sense that we based both structure recovery and density estimation on a single maximum-a-posteriori model structure. It is natural to ask to what extent the presented exact algorithmic approach and techniques can be transferred to the computations needed for Bayesian model averaging or sampling from the posterior.

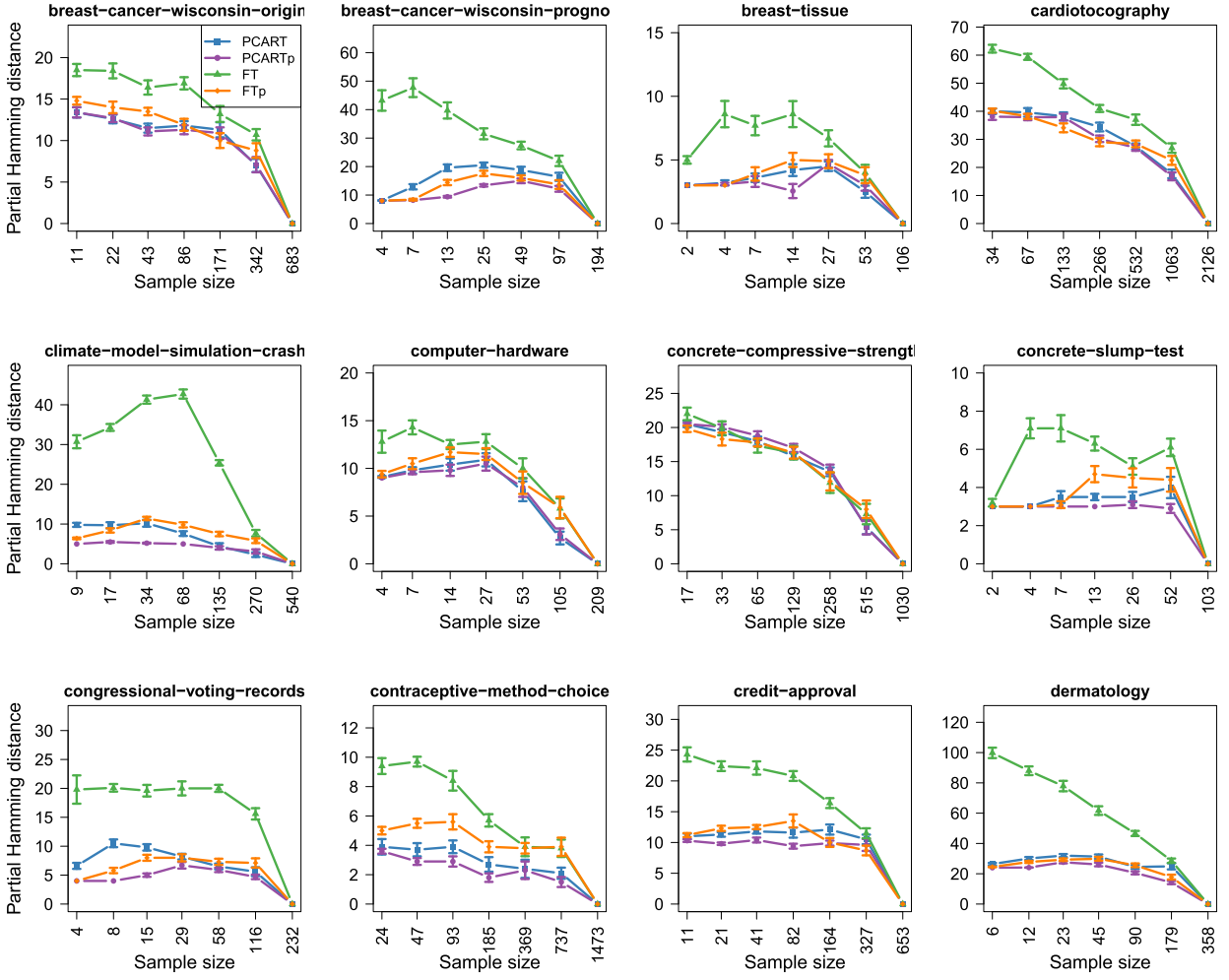
## Declaration of competing interest

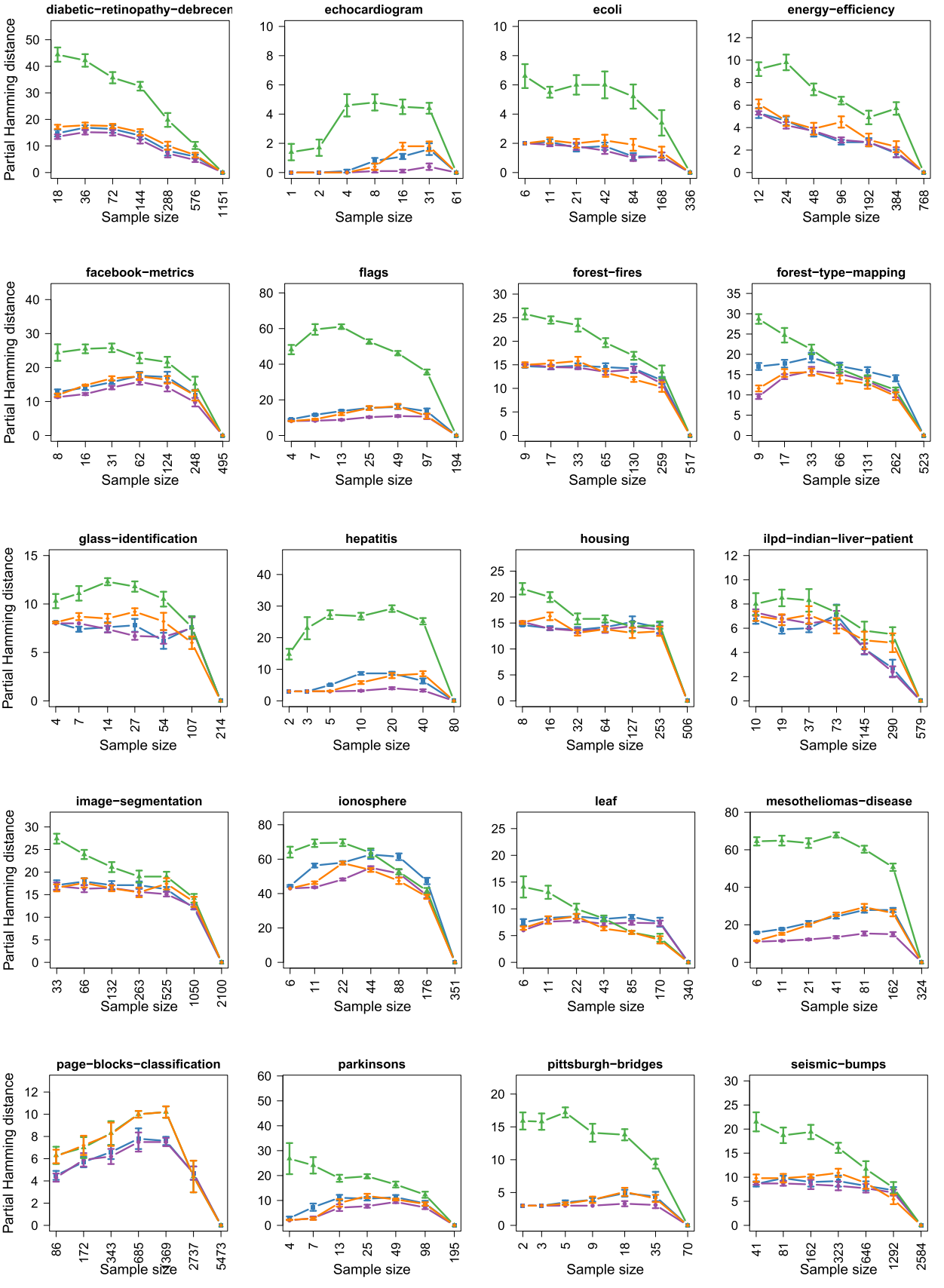
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

The authors are grateful to Antti Hyttinen for pointing out that *CPBayes* assumes score equivalence. The work was supported in part by the Academy of Finland, Grants 276864 and 316771. For the access to the Ukko 2.0 computer cluster, the authors acknowledge the Finnish Grid and Cloud Infrastructure (urn:nbn:fi:research-infras-2016072533).

## Appendix A. Full Intersection-Validation results




diabetic−retinopathy−debrecen
echocardiogram
ecoli
energy−efficiency
facebook−metrics
flags
forest−fires
forest−type−mapping
glass−identification
hepatitis
housing
ilpd−indian−liver−patient
image−segmentation
ionosphere
leaf
mesotheliomas−disease
page−blocks−classification
parkinsons
pittsburgh−bridges
seismic−bumps
Partial Hamming distance
Sample size

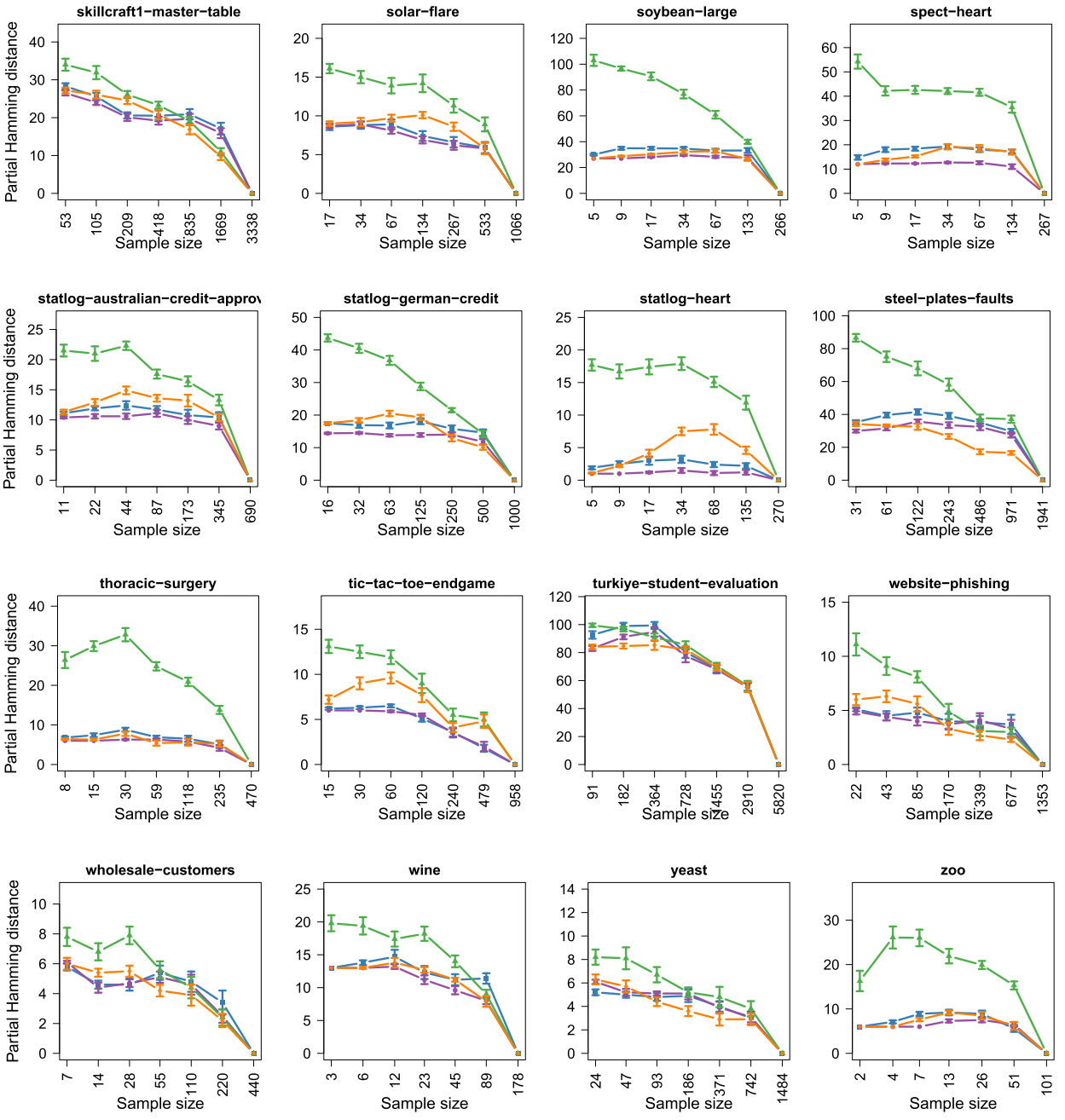



**Table A.3**
Method comparison for Intersection-Validation (continues Table 1).

(a) PCART vs FTp

| Sample size | Win | Tie | Loss |
|---|---|---|---|
| 1/64 | 8 | 31 | 13 |
| 1/32 | 14 | 21 | 17 |
| 1/16 | 11 | 26 | 15 |
| 1/8 | 11 | 29 | 12 |
| 1/4 | 12 | 27 | 13 |
| 1/2 | 11 | 28 | 13 |

(b) PCARTp vs FT

| Sample size | Win | Tie | Loss |
|---|---|---|---|
| 1/64 | 49 | 3 | 0 |
| 1/32 | 50 | 2 | 0 |
| 1/16 | 50 | 2 | 0 |
| 1/8 | 46 | 6 | 0 |
| 1/4 | 39 | 12 | 1 |
| 1/2 | 34 | 16 | 2 |

(c) PCART vs PCARTp

| Sample size | Win | Tie | Loss |
|---|---|---|---|
| 1/64 | 3 | 27 | 22 |
| 1/32 | 2 | 22 | 28 |
| 1/16 | 0 | 24 | 28 |
| 1/8 | 1 | 25 | 26 |
| 1/4 | 0 | 31 | 21 |
| 1/2 | 0 | 36 | 16 |

(d) FT vs FTp

| Sample size | Win | Tie | Loss |
|---|---|---|---|
| 1/64 | 0 | 4 | 48 |
| 1/32 | 0 | 3 | 49 |
| 1/16 | 0 | 8 | 44 |
| 1/8 | 0 | 8 | 44 |
| 1/4 | 0 | 17 | 35 |
| 1/2 | 0 | 21 | 31 |

**Table A.4**
Alternative to Table 1 that uses a statistical test instead of an errors bars for declaring upon ties: Here, a tie occurs if a two-sided Wilcoxon signed rank test yields a $p$-value above 0.05. Otherwise the mean partial Hamming distance decides upon win or loss.

(a) PCARTp vs FTp

| Sample size | Win | Tie | Loss |
|---|---|---|---|
| 1/64 | 7 | 45 | 0 |
| 1/32 | 20 | 30 | 2 |
| 1/16 | 25 | 26 | 1 |
| 1/8 | 24 | 25 | 3 |
| 1/4 | 13 | 35 | 4 |
| 1/2 | 10 | 39 | 3 |

(b) PCART vs FT

| Sample size | Win | Tie | Loss |
|---|---|---|---|
| 1/64 | 35 | 17 | 0 |
| 1/32 | 20 | 30 | 2 |
| 1/16 | 25 | 26 | 1 |
| 1/8 | 24 | 25 | 3 |
| 1/4 | 13 | 35 | 4 |
| 1/2 | 10 | 39 | 3 |

## Appendix B. Overview of used UCI data sets including full prediction results

**Table B.5**
Detailed prediction results.

| Data set | # Cont. vars. | # Cat. vars. | Sample size | Local model | Log loss | Standard error |
|---|---|---|---|---|---|---|
| abalone | 8 | 1 | 4177 | FT | -4.13322 | 0.0523931 |
| abalone | –"– | –"– | –"– | FTp | -4.13313 | 0.0524229 |
| abalone | –"– | –"– | –"– | PCART | -3.09613 | 0.0597079 |
| abalone | –"– | –"– | –"– | PCARTp | -3.0962 | 0.0597261 |
| auto-mpg | 7 | 1 | 392 | FT | -5.22265 | 0.0747865 |
| auto-mpg | –"– | –"– | –"– | FTp | -5.20905 | 0.0685599 |
| auto-mpg | –"– | –"– | –"– | PCART | -5.14424 | 0.0852387 |
| auto-mpg | –"– | –"– | –"– | PCARTp | -5.15986 | 0.0859177 |
| automobile | 16 | 9 | 159 | FT | -15.4299 | 0.307373 |
| automobile | –"– | –"– | –"– | FTp | -15.4845 | 0.329342 |
| automobile | –"– | –"– | –"– | PCART | -16.0532 | 0.373536 |
| automobile | –"– | –"– | –"– | PCARTp | -16.2046 | 0.347326 |
| breast-cancer-wisconsin-diagnostic | 30 | 1 | 569 | FT | -17.1384 | 0.269709 |
| breast-cancer-wisconsin-diagnostic | –"– | –"– | –"– | FTp | -17.1950 | 0.268222 |
| breast-cancer-wisconsin-diagnostic | –"– | –"– | –"– | PCART | -16.1698 | 0.246484 |
| breast-cancer-wisconsin-diagnostic | –"– | –"– | –"– | PCARTp | -16.1012 | 0.241510 |
| breast-cancer-wisconsin-original | 9 | 1 | 683 | FT | -5.92682 | 0.195526 |
| breast-cancer-wisconsin-original | –"– | –"– | –"– | FTp | -5.93282 | 0.197091 |
| breast-cancer-wisconsin-original | –"– | –"– | –"– | PCART | -5.98511 | 0.211834 |
| breast-cancer-wisconsin-original | –"– | –"– | –"– | PCARTp | -5.97639 | 0.214541 |
| breast-cancer-wisconsin-prognostic | 33 | 1 | 194 | FT | -30.3548 | 0.479644 |
| breast-cancer-wisconsin-prognostic | –"– | –"– | –"– | FTp | -30.3136 | 0.482371 |
| breast-cancer-wisconsin-prognostic | –"– | –"– | –"– | PCART | -31.0561 | 0.463161 |
| breast-cancer-wisconsin-prognostic | –"– | –"– | –"– | PCARTp | -30.6996 | 0.426048 |
| breast-tissue | 9 | 1 | 106 | FT | -7.44081 | 0.556503 |
| breast-tissue | –"– | –"– | –"– | FTp | -7.31847 | 0.510621 |
| breast-tissue | –"– | –"– | –"– | PCART | -5.85862 | 0.303178 |
| breast-tissue | –"– | –"– | –"– | PCARTp | -5.87273 | 0.301029 |
| cardiotocography | 21 | 2 | 2126 | FT | -13.8607 | 0.905536 |
| cardiotocography | –"– | –"– | –"– | FTp | -13.8566 | 0.904972 |
| cardiotocography | –"– | –"– | –"– | PCART | -8.30253 | 0.150645 |
| cardiotocography | –"– | –"– | –"– | PCARTp | -8.23426 | 0.142736 |
| climate-model-simulation-crashes | 18 | 2 | 540 | FT | -26.9087 | 0.0281824 |
| climate-model-simulation-crashes | –"– | –"– | –"– | FTp | -26.8998 | 0.028701 |
| climate-model-simulation-crashes | –"– | –"– | –"– | PCART | -26.8631 | 0.0284472 |
| climate-model-simulation-crashes | –"– | –"– | –"– | PCARTp | -26.8597 | 0.0288655 |
| computer-hardware | 8 | 1 | 209 | FT | -5.01143 | 0.194185 |
| computer-hardware | –"– | –"– | –"– | FTp | -5.07002 | 0.193026 |
| computer-hardware | –"– | –"– | –"– | PCART | -4.88479 | 0.173801 |
| computer-hardware | –"– | –"– | –"– | PCARTp | -4.90330 | 0.176823 |
| concrete-compressive-strength | 9 | 0 | 1030 | FT | -7.57096 | 0.0472805 |
| concrete-compressive-strength | –"– | –"– | –"– | FTp | -7.57260 | 0.0471688 |
| concrete-compressive-strength | –"– | –"– | –"– | PCART | -6.29343 | 0.0639186 |
| concrete-compressive-strength | –"– | –"– | –"– | PCARTp | -6.29343 | 0.0639186 |
| concrete-slump-test | 10 | 0 | 103 | FT | -12.4691 | 0.131444 |
| concrete-slump-test | –"– | –"– | –"– | FTp | -12.6375 | 0.127958 |
| concrete-slump-test | –"– | –"– | –"– | PCART | -12.3053 | 0.143675 |
| concrete-slump-test | –"– | –"– | –"– | PCARTp | -12.3814 | 0.140171 |
| congressional-voting-records | 0 | 17 | 232 | FT | -7.57205 | 0.0982112 |
| congressional-voting-records | –"– | –"– | –"– | FTp | -7.68824 | 0.0966163 |

(*continued on next page*)

**Table B.5** (*continued*)

| Data set | # Cont. vars. | # Cat. vars. | Sample size | Local model | Log loss | Standard error |
|---|---|---|---|---|---|---|
| congressional-voting-records | –"– | –"– | –"– | PCART | -7.67604 | 0.104446 |
| congressional-voting-records | –"– | –"– | –"– | PCARTp | -7.65147 | 0.0963997 |
| contraceptive-method-choice | 2 | 8 | 1473 | FT | -8.83841 | 0.0251841 |
| contraceptive-method-choice | –"– | –"– | –"– | FTp | -8.84223 | 0.0250688 |
| contraceptive-method-choice | –"– | –"– | –"– | PCART | -8.82863 | 0.025129 |
| contraceptive-method-choice | –"– | –"– | –"– | PCARTp | -8.82542 | 0.0250515 |
| credit-approval | 6 | 10 | 653 | FT | -11.3644 | 0.135358 |
| credit-approval | –"– | –"– | –"– | FTp | -11.3758 | 0.135212 |
| credit-approval | –"– | –"– | –"– | PCART | -11.2250 | 0.12561 |
| credit-approval | –"– | –"– | –"– | PCARTp | -11.2029 | 0.125398 |
| dermatology | 1 | 34 | 358 | FT | -21.5277 | 0.167899 |
| dermatology | –"– | –"– | –"– | FTp | -21.3569 | 0.170317 |
| dermatology | –"– | –"– | –"– | PCART | -21.5545 | 0.171296 |
| dermatology | –"– | –"– | –"– | PCARTp | -21.4447 | 0.17278 |
| diabetic-retinopathy-debrecen | 16 | 4 | 1151 | FT | -1.16755 | 0.122955 |
| diabetic-retinopathy-debrecen | –"– | –"– | –"– | FTp | -1.16245 | 0.122946 |
| diabetic-retinopathy-debrecen | –"– | –"– | –"– | PCART | -1.31978 | 0.100323 |
| diabetic-retinopathy-debrecen | –"– | –"– | –"– | PCARTp | -1.32611 | 0.101476 |
| echocardiogram | 6 | 3 | 61 | FT | -10.4335 | 0.180215 |
| echocardiogram | –"– | –"– | –"– | FTp | -10.1109 | 0.177413 |
| echocardiogram | –"– | –"– | –"– | PCART | -9.84427 | 0.188266 |
| echocardiogram | –"– | –"– | –"– | PCARTp | -9.78419 | 0.175198 |
| ecoli | 5 | 3 | 336 | FT | -6.15148 | 0.0693145 |
| ecoli | –"– | –"– | –"– | FTp | -6.16784 | 0.0725994 |
| ecoli | –"– | –"– | –"– | PCART | -6.07504 | 0.0825972 |
| ecoli | –"– | –"– | –"– | PCARTp | -6.07331 | 0.0824237 |
| energy-efficiency | –"– | –"– | –"– | FTp | -0.506756 | 0.0344042 |
| energy-efficiency | –"– | –"– | –"– | PCART | 0.330032 | 0.0282223 |
| energy-efficiency | –"– | –"– | –"– | PCARTp | 0.321051 | 0.0278874 |
| facebook-metrics | 17 | 2 | 495 | FT | -7.92066 | 0.51538 |
| facebook-metrics | –"– | –"– | –"– | FTp | -7.90208 | 0.519498 |
| facebook-metrics | –"– | –"– | –"– | PCART | -5.80853 | 0.23913 |
| facebook-metrics | –"– | –"– | –"– | PCARTp | -5.7852 | 0.237138 |
| flags | 6 | 23 | 194 | FT | -19.7518 | 0.403958 |
| flags | –"– | –"– | –"– | FTp | -19.7994 | 0.38416 |
| flags | –"– | –"– | –"– | PCART | -22.9287 | 0.720113 |
| flags | –"– | –"– | –"– | PCARTp | -22.6902 | 0.642977 |
| forest-fires | 11 | 2 | 517 | FT | -11.3889 | 0.235356 |
| forest-fires | –"– | –"– | –"– | FTp | -11.3803 | 0.223158 |
| forest-fires | –"– | –"– | –"– | PCART | -10.7881 | 0.247706 |
| forest-fires | –"– | –"– | –"– | PCARTp | -10.7746 | 0.242741 |
| forest-type-mapping | 27 | 1 | 523 | FT | -13.9835 | 0.257444 |
| forest-type-mapping | –"– | –"– | –"– | FTp | -14.015 | 0.257229 |
| forest-type-mapping | –"– | –"– | –"– | PCART | -12.4947 | 0.196846 |
| forest-type-mapping | –"– | –"– | –"– | PCARTp | -12.5661 | 0.197387 |
| glass-identification | 9 | 1 | 214 | FT | -9.62004 | 0.416359 |
| glass-identification | –"– | –"– | –"– | FTp | -9.58764 | 0.416512 |
| glass-identification | –"– | –"– | –"– | PCART | -10.4654 | 0.724344 |
| glass-identification | –"– | –"– | –"– | PCARTp | -10.4489 | 0.723731 |
| hepatitis | 6 | 14 | 80 | FT | -16.625 | 0.26554 |
| hepatitis | –"– | –"– | –"– | FTp | -16.1782 | 0.276981 |
| hepatitis | –"– | –"– | –"– | PCART | -15.6056 | 0.270122 |
| hepatitis | –"– | –"– | –"– | PCARTp | -15.4822 | 0.279482 |
| housing | 13 | 1 | 506 | FT | -5.58115 | 0.08945 |
| housing | –"– | –"– | –"– | FTp | -5.62168 | 0.0911751 |
| housing | –"– | –"– | –"– | PCART | -4.5876 | 0.164486 |
| housing | –"– | –"– | –"– | PCARTp | -4.58776 | 0.164376 |
| ilpd-indian-liver-patient | 9 | 1 | 579 | FT | -6.84129 | 0.121699 |
| ilpd-indian-liver-patient | –"– | –"– | –"– | FTp | -6.84757 | 0.11935 |
| ilpd-indian-liver-patient | –"– | –"– | –"– | PCART | -7.36577 | 0.206202 |
| ilpd-indian-liver-patient | –"– | –"– | –"– | PCARTp | -7.37624 | 0.206622 |
| image-segmentation | 18 | 1 | 2100 | FT | -0.0237947 | 0.191382 |
| image-segmentation | –"– | –"– | –"– | FTp | -0.0464886 | 0.191152 |
| image-segmentation | –"– | –"– | –"– | PCART | 2.87447 | 0.354167 |
| image-segmentation | –"– | –"– | –"– | PCARTp | 2.87656 | 0.35393 |
| ionosphere | 33 | 1 | 351 | FT | -27.3869 | 0.593631 |
| ionosphere | –"– | –"– | –"– | FTp | -27.7515 | 0.62048 |
| ionosphere | –"– | –"– | –"– | PCART | -26.0673 | 0.715903 |
| ionosphere | –"– | –"– | –"– | PCARTp | -26.0474 | 0.680705 |

**Table B.5** (*continued*)

| Data set | # Cont. vars. | # Cat. vars. | Sample size | Local model | Log loss | Standard error |
|---|---|---|---|---|---|---|
| leaf | 14 | 1 | 340 | FT | -4.12281 | 0.1412 |
| leaf | –"– | –"– | –"– | FTp | -4.12235 | 0.139118 |
| leaf | –"– | –"– | –"– | PCART | -3.67209 | 0.115732 |
| leaf | –"– | –"– | –"– | PCARTp | -3.63947 | 0.122703 |
| mesotheliomas-disease | 21 | 14 | 324 | FT | -36.5685 | 0.749786 |
| mesotheliomas-disease | –"– | –"– | –"– | FTp | -35.8341 | 0.737257 |
| mesotheliomas-disease | –"– | –"– | –"– | PCART | -36.6784 | 0.756998 |
| mesotheliomas-disease | –"– | –"– | –"– | PCARTp | -36.5297 | 0.770939 |
| page-blocks-classification | 10 | 1 | 5473 | FT | 4.01837 | 0.0736515 |
| page-blocks-classification | –"– | –"– | –"– | FTp | 4.01837 | 0.0736515 |
| page-blocks-classification | –"– | –"– | –"– | PCART | 4.08716 | 0.0698548 |
| page-blocks-classification | –"– | –"– | –"– | PCARTp | 4.08716 | 0.0698548 |
| parkinsons | 22 | 1 | 195 | FT | -12.6838 | 0.412472 |
| parkinsons | –"– | –"– | –"– | FTp | -12.7884 | 0.41484 |
| parkinsons | –"– | –"– | –"– | PCART | -11.5353 | 0.306426 |
| parkinsons | –"– | –"– | –"– | PCARTp | -11.4615 | 0.301666 |
| pittsburgh-bridges | 4 | 8 | 70 | FT | -10.4202 | 0.191279 |
| pittsburgh-bridges | –"– | –"– | –"– | FTp | -10.0553 | 0.156742 |
| pittsburgh-bridges | –"– | –"– | –"– | PCART | -9.94399 | 0.189965 |
| pittsburgh-bridges | –"– | –"– | –"– | PCARTp | -9.95035 | 0.188734 |
| seismic-bumps | 11 | 5 | 2584 | FT | 2.81868 | 0.140903 |
| seismic-bumps | –"– | –"– | –"– | FTp | 2.81758 | 0.140939 |
| seismic-bumps | –"– | –"– | –"– | PCART | -6.07423 | 3.55519 |
| seismic-bumps | –"– | –"– | –"– | PCARTp | -6.07256 | 3.55431 |
| skillcraft1-master-table | 19 | 0 | 3338 | FT | -19.7602 | 1.12933 |
| skillcraft1-master-table | –"– | –"– | –"– | FTp | -19.7588 | 1.13031 |
| skillcraft1-master-table | –"– | –"– | –"– | PCART | -17.1991 | 0.0852489 |
| skillcraft1-master-table | –"– | –"– | –"– | PCARTp | -17.1931 | 0.0847659 |
| solar-flare | 3 | 9 | 1066 | FT | -6.72262 | 0.822185 |
| solar-flare | –"– | –"– | –"– | FTp | -6.72473 | 0.822731 |
| solar-flare | –"– | –"– | –"– | PCART | -9.22578 | 1.21448 |
| solar-flare | –"– | –"– | –"– | PCARTp | -9.23772 | 1.21471 |
| soybean-large | 1 | 35 | 266 | FT | -13.588 | 0.123201 |
| soybean-large | –"– | –"– | –"– | FTp | -13.7601 | 0.118514 |
| soybean-large | –"– | –"– | –"– | PCART | -13.6173 | 0.128789 |
| soybean-large | –"– | –"– | –"– | PCARTp | -14.0551 | 0.12202 |
| spect-heart | 0 | 23 | 267 | FT | -11.121 | 0.116073 |
| spect-heart | –"– | –"– | –"– | FTp | -10.9539 | 0.107264 |
| spect-heart | –"– | –"– | –"– | PCART | -10.9515 | 0.0983607 |
| spect-heart | –"– | –"– | –"– | PCARTp | -10.8748 | 0.0981561 |
| statlog-australian-credit-approval | 6 | 9 | 690 | FT | -11.3948 | 0.134925 |
| statlog-australian-credit-approval | –"– | –"– | –"– | FTp | -11.3811 | 0.135013 |
| statlog-australian-credit-approval | –"– | –"– | –"– | PCART | -11.1695 | 0.130344 |
| statlog-australian-credit-approval | –"– | –"– | –"– | PCARTp | -11.1793 | 0.128346 |
| statlog-german-credit | 7 | 14 | 1000 | FT | -21.2715 | 0.103431 |
| statlog-german-credit | –"– | –"– | –"– | FTp | -21.221 | 0.103087 |
| statlog-german-credit | –"– | –"– | –"– | PCART | -22.566 | 0.200828 |
| statlog-german-credit | –"– | –"– | –"– | PCARTp | -22.5922 | 0.203403 |
| statlog-heart | 5 | 9 | 270 | FT | -13.1001 | 0.092923 |
| statlog-heart | –"– | –"– | –"– | FTp | -13.1018 | 0.0815172 |
| statlog-heart | –"– | –"– | –"– | PCART | -13.075 | 0.0835523 |
| statlog-heart | –"– | –"– | –"– | PCARTp | -13.0835 | 0.0830395 |
| steel-plates-faults | 34 | 0 | 1941 | FT | 5.4919 | 0.361659 |
| steel-plates-faults | –"– | –"– | –"– | FTp | 5.48751 | 0.360997 |
| steel-plates-faults | –"– | –"– | –"– | PCART | 11.9394 | 0.428835 |
| steel-plates-faults | –"– | –"– | –"– | PCARTp | 11.9415 | 0.425938 |
| thoracic-surgery | 3 | 14 | 470 | FT | -7.96896 | 0.135233 |
| thoracic-surgery | –"– | –"– | –"– | FTp | -7.86737 | 0.133709 |
| thoracic-surgery | –"– | –"– | –"– | PCART | -8.6541 | 0.218177 |
| thoracic-surgery | –"– | –"– | –"– | PCARTp | -8.62991 | 0.218839 |
| tic-tac-toe-endgame | 0 | 10 | 958 | FT | -9.4217 | 0.0103552 |
| tic-tac-toe-endgame | –"– | –"– | –"– | FTp | -9.42878 | 0.0111993 |
| tic-tac-toe-endgame | –"– | –"– | –"– | PCART | -9.1907 | 0.00815003 |
| tic-tac-toe-endgame | –"– | –"– | –"– | PCARTp | -9.18708 | 0.00800115 |
| turkiye-student-evaluation | 31 | 2 | 5820 | FT | -7.17341 | 0.216049 |
| turkiye-student-evaluation | –"– | –"– | –"– | FTp | -7.17477 | 0.216316 |
| turkiye-student-evaluation | –"– | –"– | –"– | PCART | -21.0203 | 0.172557 |
| turkiye-student-evaluation | –"– | –"– | –"– | PCARTp | -21.0023 | 0.170864 |
| website-phishing | 0 | 10 | 1353 | FT | -7.30799 | 0.020902 |

(*continued on next page*)

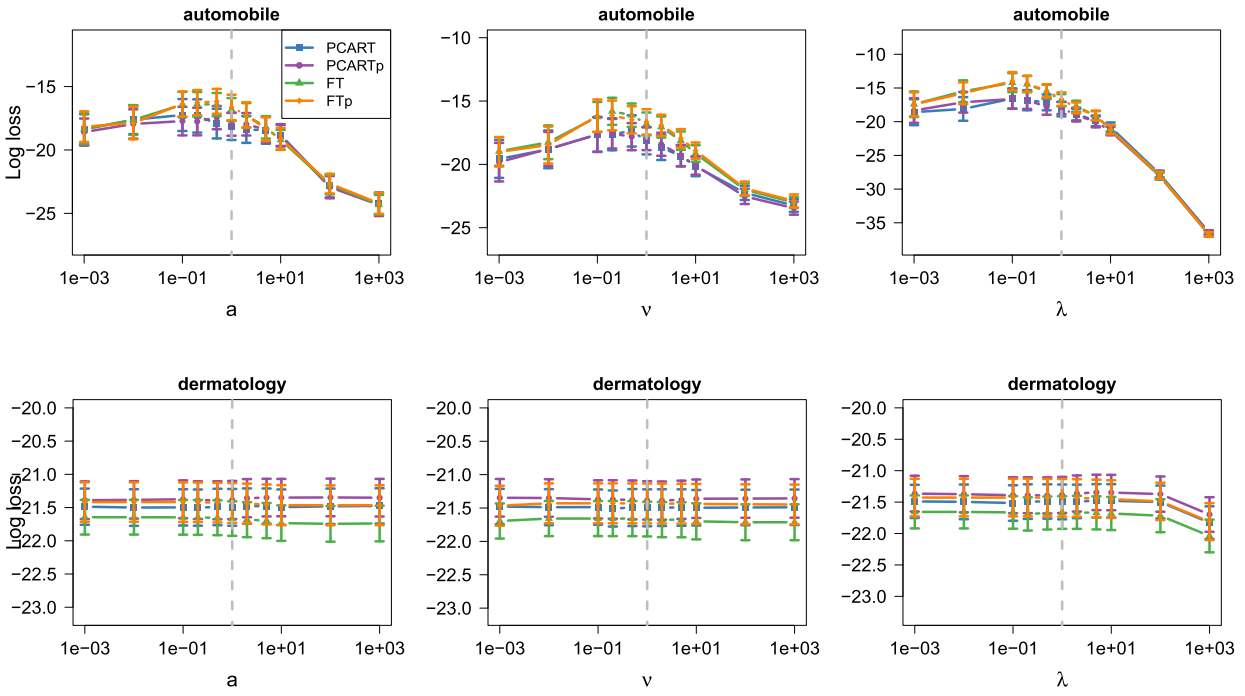
**Table B.5** (*continued*)

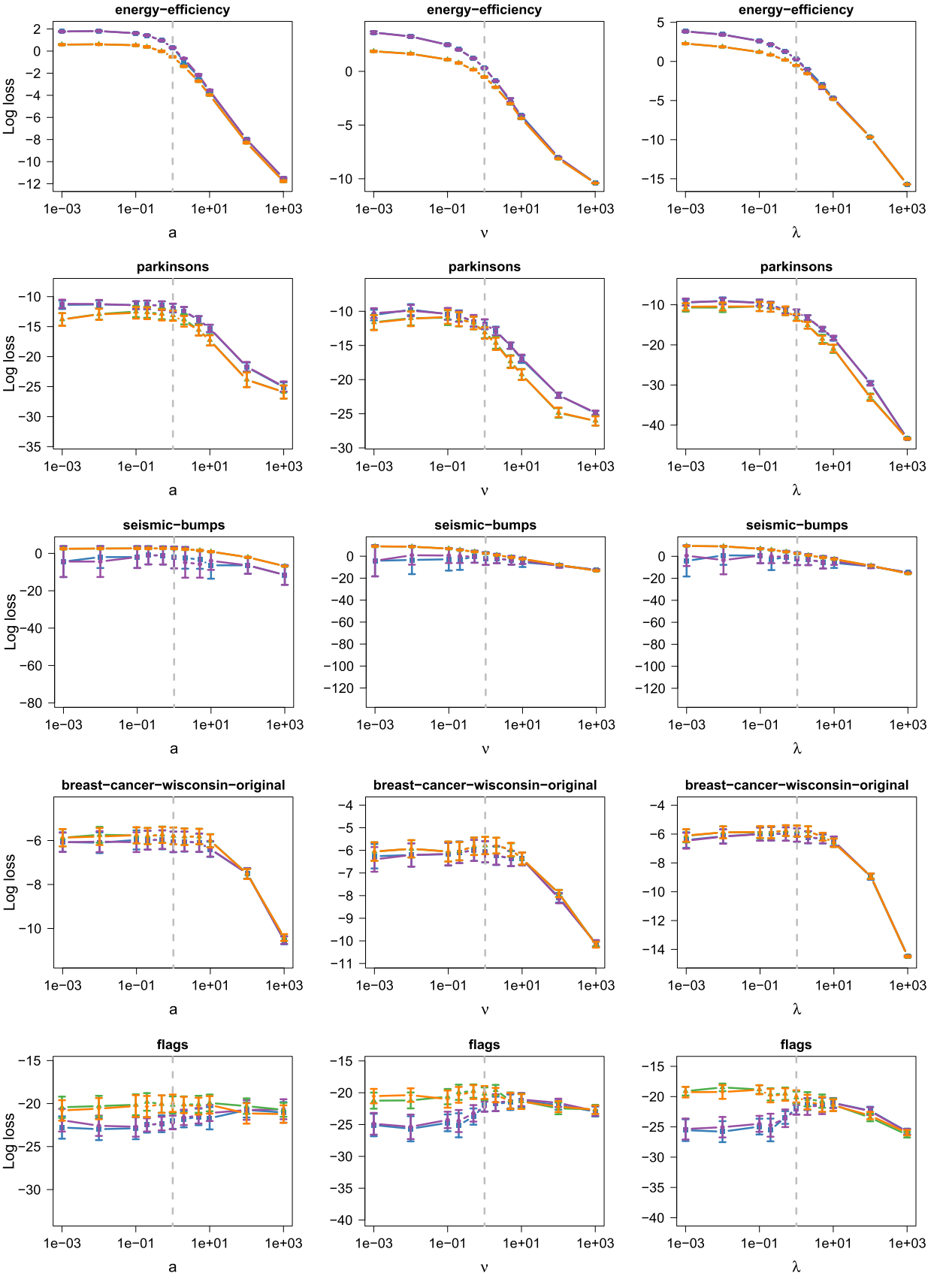
| Data set | # Cont. vars. | # Cat. vars. | Sample size | Local model | Log loss | Standard error |
|---|---|---|---|---|---|---|
| website-phishing | –"– | –"– | –"– | FTp | -7.3136 | 0.0211977 |
| website-phishing | –"– | –"– | –"– | PCART | -7.31822 | 0.0217394 |
| website-phishing | –"– | –"– | –"– | PCARTp | -7.3308 | 0.0216457 |
| wholesale-customers | 6 | 2 | 440 | FT | -6.39482 | 0.136206 |
| wholesale-customers | –"– | –"– | –"– | FTp | -6.36885 | 0.134723 |
| wholesale-customers | –"– | –"– | –"– | PCART | -6.19533 | 0.146827 |
| wholesale-customers | –"– | –"– | –"– | PCARTp | -6.19086 | 0.145875 |
| wine | 13 | 1 | 178 | FT | -14.5214 | 0.152794 |
| wine | –"– | –"– | –"– | FTp | -14.5022 | 0.164498 |
| wine | –"– | –"– | –"– | PCART | -14.7872 | 0.173345 |
| wine | –"– | –"– | –"– | PCARTp | -14.8227 | 0.17205 |
| yeast | 8 | 1 | 1484 | FT | -9.80739 | 0.269772 |
| yeast | –"– | –"– | –"– | FTp | -9.80699 | 0.269798 |
| yeast | –"– | –"– | –"– | PCART | -14.0574 | 1.00005 |
| yeast | –"– | –"– | –"– | PCARTp | -14.0525 | 1.00019 |
| zoo | 1 | 16 | 101 | FT | -6.5594 | 0.150966 |
| zoo | –"– | –"– | –"– | FTp | -6.51337 | 0.141782 |
| zoo | –"– | –"– | –"– | PCART | -7.14786 | 0.156924 |
| zoo | –"– | –"– | –"– | PCARTp | -7.26475 | 0.15188 |

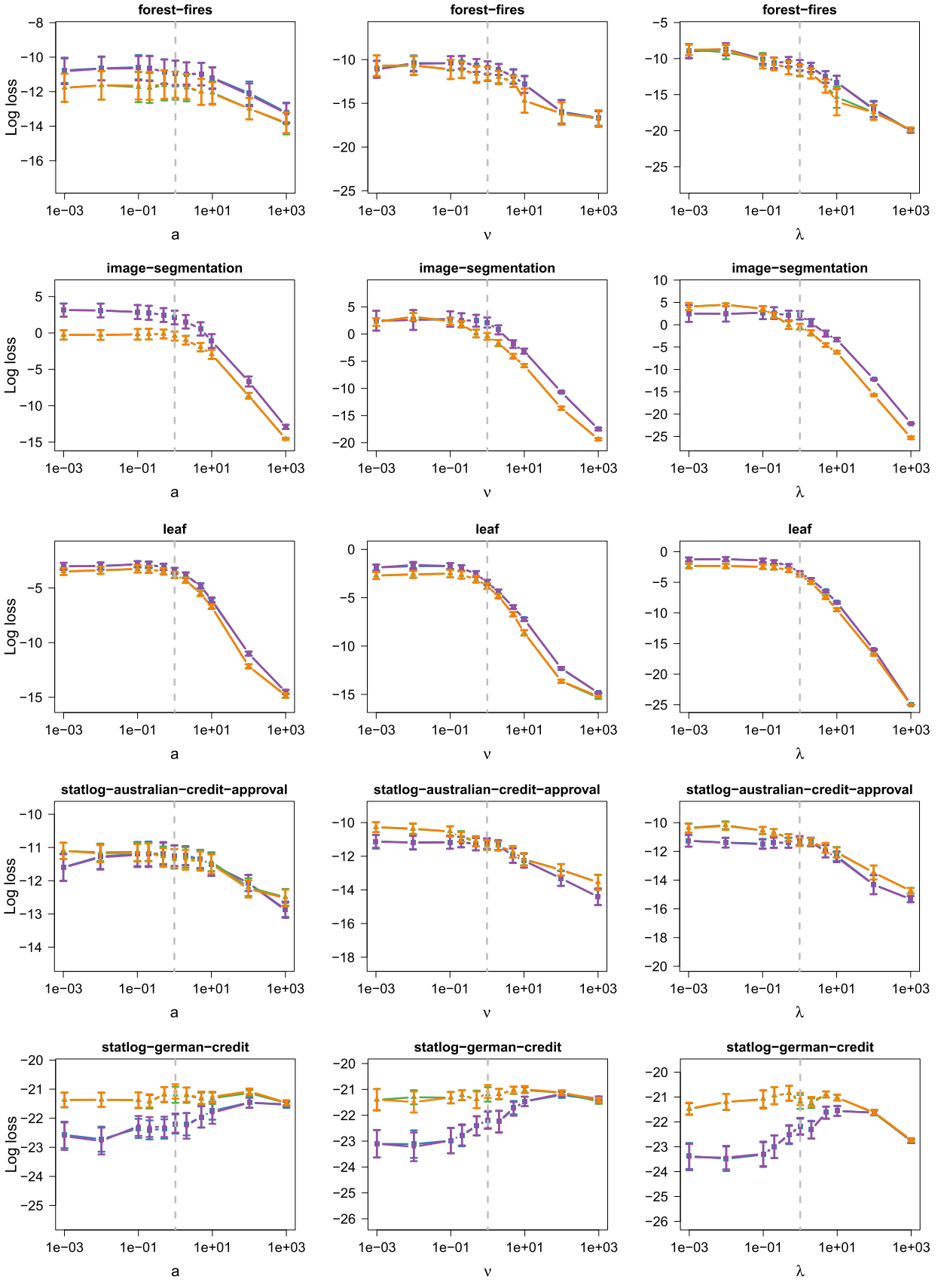
**Table B.6**
Alternative to Table 2. Here, a tie occurs if two-sided a Wilcoxon signed rank test yields a $p$-value above 0.05.

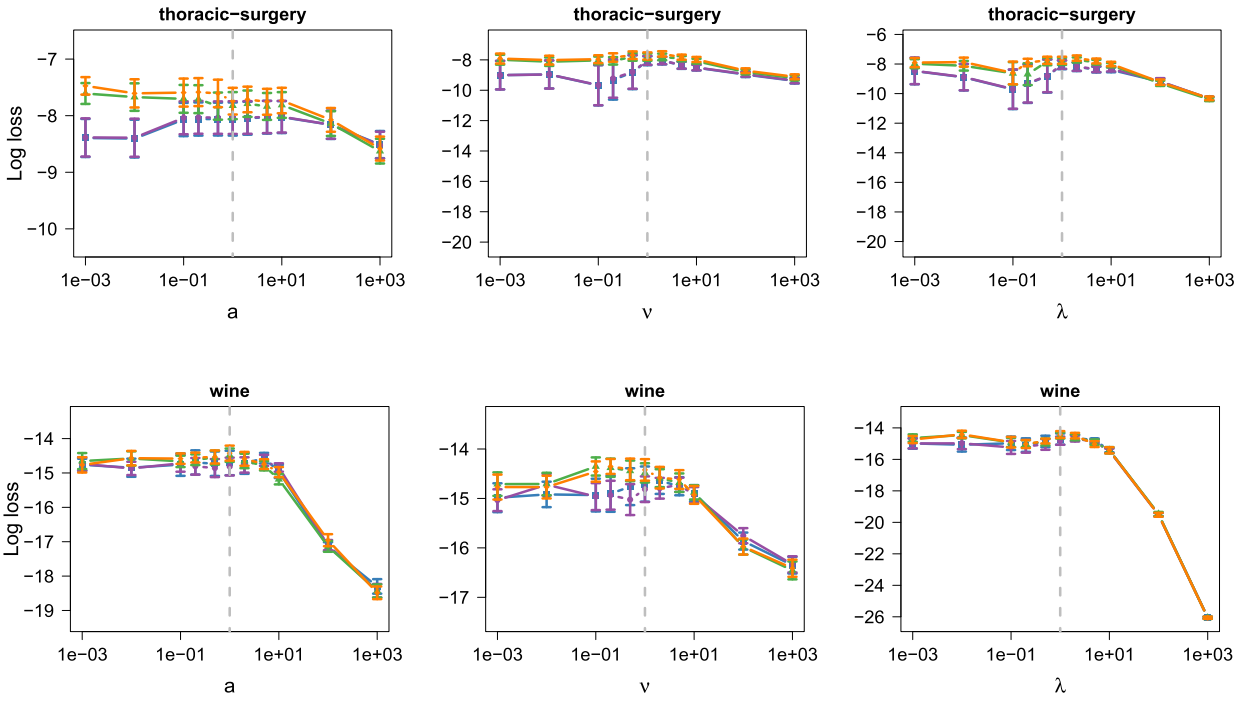
| | | | Win | Tie | Loss |
|---|---|---|---|---|---|
| PCART | vs | PCARTp | 10 | 35 | 7 |
| PCART | vs | FT | 25 | 14 | 13 |
| PCART | vs | FTp | 25 | 14 | 13 |
| PCARTp | vs | FT | 25 | 13 | 14 |
| PCARTp | vs | FTp | 24 | 13 | 15 |
| FT | vs | FTp | 11 | 33 | 8 |

## Appendix C. Remaining hyperparameter sensitivity plots




energy−efficiency
parkinsons
seismic−bumps
breast−cancer−wisconsin−original
flags
Log loss
a
ν
λ
1e−03
1e−01
1e+01
1e+03


forest−fires
image−segmentation
leaf
statlog−australian−credit−approval
statlog−german−credit
Log loss
a
ν
λ
1e−03
1e−01
1e+01
1e+03

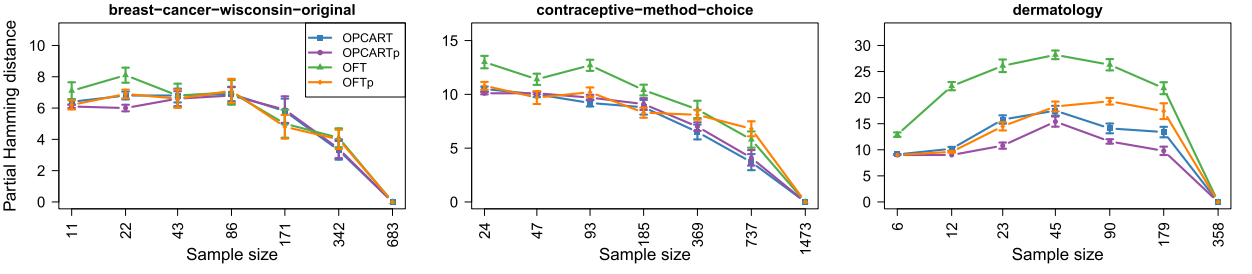


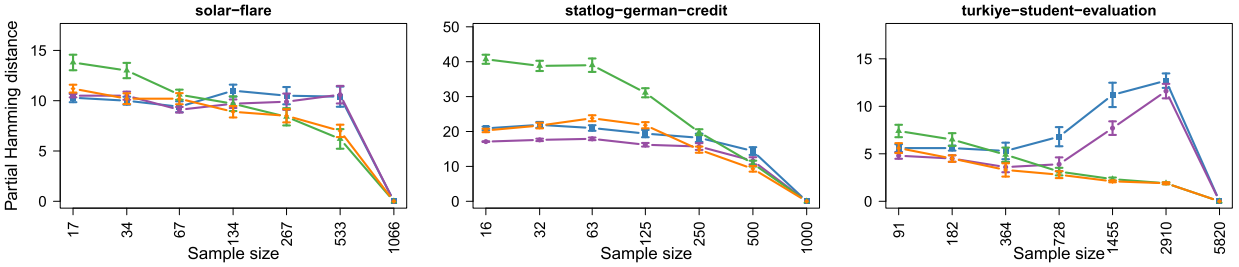

## Appendix D. Additional results for studies with ordinal variables

### *D.1. Data selection: ordinal variables*

The following paragraph describes which variables in the six chosen UCI data sets are treated as ordinal. In breast-cancer-wisconsin-original all variables (except for *class*) are pre-discretized quantitative measurements in the integer range 1-10. In contraceptive-method-choice the variables *Wife's education*, *Husband's education*, *Standard-of-living index* are quantitative and assume integer values 1-4. contraceptive-method-choice: *Wife's education*, *Husband's education*, *Standard-of-living index* In case of dermatology all attributes except for *Age* and *Class* have integer values in the range 0-3. Note that *Age* is ordinal in principle, but due to the large number of possible values additional discretization would be required. Hence we treated it as real-valued variable. solar-flare has two variables (*evolution* and *previous 24 hour flare activity*) that assume values in the range 1-3. In addition *common flares*, *moderate flares*, and *severe flares* are integer count variables with a low maximal frequency (9, 6, and 3, respectively). In statlog-german-credit the variables *Installment rate in percentage of disposable income*, *Present residence since*, *Number of existing credits at this bank*, *Number of people being liable to provide maintenance for* are pre-binned variables, assuming values in the range 1-4. Finally, turkiye-student-evaluation is poll data set that contains (except for *instr* and *class*) question outcomes assuming values in the range 1–5.

### *D.2. Intersection-validation results*






### D.3. Prediction results

**Table D.7**
Prediction results for ordinal data studies.

| Data set | # Cont. vars. | # Cat. vars. | #Ord. vars. | Local model | Log loss | Standard error |
|---|---|---|---|---|---|---|
| breast-cancer-wisconsin-original | 0 | 1 | 9 | OFT | -3.22823 | 0.0434679 |
| breast-cancer-wisconsin-original | –"– | –"– | –"– | OFTp | -3.23009 | 0.0428471 |
| breast-cancer-wisconsin-original | –"– | –"– | –"– | OPCART | -3.03550 | 0.0382332 |
| breast-cancer-wisconsin-original | –"– | –"– | –"– | OPCARTp | -3.04021 | 0.0399682 |
| contraceptive-method-choice | 2 | 5 | 3 | OFT | -6.05649 | 0.0216102 |
| contraceptive-method-choice | –"– | –"– | –"– | OFTp | -6.05589 | 0.0209412 |
| contraceptive-method-choice | –"– | –"– | –"– | OPCART | -6.07919 | 0.0205113 |
| contraceptive-method-choice | –"– | –"– | –"– | OPCARTp | -6.07733 | 0.0204595 |
| dermatology | 1 | 1 | 33 | OFT | -7.07661 | 0.1038180 |
| dermatology | –"– | –"– | –"– | OFTp | -7.08338 | 0.1029680 |
| dermatology | –"– | –"– | –"– | OPCART | -6.56488 | 0.1174400 |
| dermatology | –"– | –"– | –"– | OPCARTp | -6.65792 | 0.1155140 |
| solar-flare | 0 | 7 | 5 | OFT | -4.95808 | 0.0281127 |
| solar-flare | –"– | –"– | –"– | OFTp | -4.96509 | 0.0279616 |
| solar-flare | –"– | –"– | –"– | OPCART | -4.94454 | 0.0300316 |
| solar-flare | –"– | –"– | –"– | OPCARTp | -4.96034 | 0.0304578 |
| statlog-german-credit | 2 | 13 | 4 | OFT | -16.2500 | 0.04751570 |
| statlog-german-credit | –"– | –"– | –"– | OFTp | -16.2339 | 0.0477208 |
| statlog-german-credit | –"– | –"– | –"– | OPCART | -16.5446 | 0.0543279 |
| statlog-german-credit | –"– | –"– | –"– | OPCARTp | -16.5238 | 0.053052 |
| turkiye-student-evaluation | 0 | 2 | 31 | OFT | -4.07527 | 0.0454955 |
| turkiye-student-evaluation | –"– | –"– | –"– | OFTp | -4.07949 | 0.0455848 |
| turkiye-student-evaluation | –"– | –"– | –"– | OPCART | -3.81664 | 0.0369689 |
| turkiye-student-evaluation | –"– | –"– | –"– | OPCARTp | -3.81850 | 0.0372704 |

## References

[1] C. Boutilier, N. Friedman, M. Goldszmidt, D. Koller, Context-specific independence in Bayesian networks, in: Proc. UAI, 1996.
[2] N. Friedman, M. Goldszmidt, Learning Bayesian networks with local structure, in: Proc. UAI, 1996.
[3] D. Chickering, D. Heckerman, C. Meek, A Bayesian approach to learning Bayesian networks with local structure, in: Proc. UAI, 1997.
[4] M. Koivisto, K. Sood, Computational aspects of Bayesian partition models, in: Proc. ICML, 2005, pp. 433–440.
[5] Y. Chen, T. Wheeler, M. Kochenderfer, Learning discrete Bayesian networks from continuous data, J. Artif. Intell. Res. 59 (2017) 103–132.
[6] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, MIT Press, 2009.
[7] V. Romero, R. Rumí, A. Salmerón, Learning hybrid Bayesian networks using mixtures of truncated exponentials, Int. J. Approx. Reason. 42 (2006) 54–68.
[8] D.M. Chickering, Learning Bayesian Networks Is NP-Complete, Springer, New York, NY, 1996, pp. 121–130.
[9] B. Malone, K. Kangas, M. Järvisalo, M. Koivisto, P. Myllymäki, Empirical hardness of finding optimal Bayesian network structures: algorithm selection and runtime prediction, Mach. Learn. 107 (2018) 247–283.
[10] S. Ott, S. Miyano, Finding optimal gene networks using biological constraints, Genome Inform. 14 (2003) 124–133.
[11] M. Koivisto, K. Sood, Exact Bayesian structure discovery in Bayesian networks, J. Mach. Learn. Res. 5 (2004) 549–573.
[12] A.P. Singh, A.W. Moore, Finding Optimal Bayesian Networks by Dynamic Programming, Technical Report CMU-CALD-05-106, Carnegie Mellon University, 2005.
[13] T. Silander, P. Myllymäki, A simple approach for finding the globally optimal Bayesian network structure, in: Proc. UAI, 2006.
[14] C. Yuan, B. Malone, Learning optimal Bayesian networks: a shortest path perspective, J. Artif. Intell. Res. 48 (2013) 23–65.
[15] M. Bartlett, J. Cussens, Integer linear programming for the Bayesian network structure learning problem, Artif. Intell. 244 (2017) 258–271.
[16] P. van Beek, H. Hoffmann, Machine learning of Bayesian networks using constraint programming, in: Proc. CP, in: Lecture Notes in Computer Science, vol. 9255, Springer, 2015, pp. 429–445.
[17] B. Malone, M. Järvisalo, P. Myllymäki, Impact of learning strategies on the quality of Bayesian networks: an empirical evaluation, in: Proc. UAI, 2015, pp. 562–571.
[18] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, Wadsworth, 1984.
[19] W. Buntine, Learning classification trees, Stat. Comput. 2 (1992) 63–73.
[20] D. Denison, B. Mallick, A. Smith, A Bayesian CART algorithm, Biometrika 85 (1998) 363–377.
[21] H. Chipman, E. George, R. McCulloch, Bayesian CART model search, J. Am. Stat. Assoc. 93 (1998) 935–948.

[22] D. Donoho, CART and best-ortho-basis: a connection, Ann. Stat. 25 (1997) 1870–1911.
[23] C. Scott, R. Nowak, Minimax optimal classification with dyadic decision trees, IEEE Trans. Inf. Theory 52 (2006) 1335–1353.
[24] G. Blanchard, C. Schäfer, Y. Rozenholc, K. Müller, Optimal dyadic decision trees, Mach. Learn. 66 (2007) 209–241.
[25] S. van der Pas, V. Rockova, Bayesian dyadic trees and histograms for regression, in: Advances in Neural Information Processing Systems (NIPS), vol. 30, 2017, pp. 2086–2096.
[26] T. Talvitie, R. Eggeling, M. Koivisto, Finding optimal Bayesian networks with local structure, in: Proc. PGM, vol. 72, PMLR, 2018, pp. 451–462.
[27] N. Angelopoulos, J. Cussens, Bayesian learning of Bayesian networks with informative priors, Ann. Math. Artif. Intell. 54 (2008) 53–98.
[28] K. Murphy, Conjugate Bayesian Analysis of the Gaussian Distribution, Technical Report, University of British, Columbia, 2007.
[29] N. Friedman, D. Koller, Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks, Mach. Learn. 50 (2003) 95–125.
[30] M. Scutari, Learning Bayesian networks with the bnlearn R package, J. Stat. Softw. 35 (2010) 1–22.
[31] M. Lichman, UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA, 2013, http://archive.ics.uci.edu/ml.
[32] I. Tsamardinos, L. Brown, C. Aliferis, The max-min hill-climbing Bayesian network structure learning algorithm, Mach. Learn. 65 (2006) 31–78.
[33] R. Eggeling, J. Viinikka, A. Vuoksenmaa, M. Koivisto, On structure priors for learning Bayesian networks, in: Proc. AISTATS, vol. 89, PMLR, 2019, pp. 1687–1695.
[34] J. Viinikka, R. Eggeling, M. Koivisto, Intersection-validation: a method for evaluating structure learning without ground truth, in: Proc. AISTATS, vol. 84, PMLR, 2018, pp. 1570–1578.
[35] F. Wilcoxon, Individual comparisons by ranking methods, Biom. Bull. 1 (1945) 80–83.
[36] R. Eggeling, M. Koivisto, Pruning Rules for Learning Parsimonious Context Trees, Proc. UAI, vol. 32, AUAI Press, 2016, pp. 152–161.
[37] K. Karra, L. Mili, Hybrid copula Bayesian networks, in: Proc. PGM, vol. 52, PMLR, 2016, pp. 240–251.