International Journal of Approximate Reasoning 53 (2012) 248–272

Contents lists available at SciVerse ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

Learning Bayesian network classifiers by risk minimization

Roy Kelner [a], Boaz Lerner [b,*]

[a] Department of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel
[b] Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel

ARTICLE INFO

Article history:
Received 22 June 2011
Received in revised form 1 October 2011
Accepted 24 October 2011
Available online 29 October 2011

Keywords:
Bayesian networks
Classification
Probabilistic graphical models
Structure learning

ABSTRACT

Bayesian networks (BNs) provide a powerful graphical model for encoding the probabilistic relationships among a set of variables, and hence can naturally be used for classification. However, Bayesian network classifiers (BNCs) learned in the common way using likelihood scores usually tend to achieve only mediocre classification accuracy because these scores are less specific to classification, but rather suit a general inference problem. We propose risk minimization by cross validation (RMCV) using the 0/1 loss function, which is a classification-oriented score for unrestricted BNCs. RMCV is an extension of classification-oriented scores commonly used in learning restricted BNCs and non-BN classifiers. Using small real and synthetic problems, allowing for learning all possible graphs, we empirically demonstrate RMCV superiority to marginal and class-conditional likelihood-based scores with respect to classification accuracy. Experiments using twenty-two real-world datasets show that BNCs learned using an RMCV-based algorithm significantly outperform the naive Bayesian classifier (NBC), tree augmented NBC (TAN), and other BNCs learned using marginal or conditional likelihood scores and are on par with non-BN state of the art classifiers, such as support vector machine, neural network, and classification tree. These experiments also show that an optimized version of RMCV is faster than all unrestricted BNCs and comparable with the neural network with respect to run-time. The main conclusion from our experiments is that unrestricted BNCs, when learned properly, can be a good alternative to restricted BNCs and traditional machine-learning classifiers with respect to both accuracy and efficiency.

1. Introduction

One fundamental task of machine learning is classification, where instances (patterns) are assigned to their corresponding classes. This is a supervised learning task, where a training dataset of instances with labels representing instance classes is used to train a classifier. Since a Bayesian (belief) network (BN) [1–3] provides a graphical model for encoding relationships, such as dependencies and conditional independencies between variables, and for inferring probabilistically about variables, it is natural to use the BN for classification.

Indeed, along with the traditional classifiers based on the neural network (NN), the support vector machine (SVM), and the (decision) classification tree (CT), classifiers based on BN have recently been introduced and studied [4–11]. Learning a Bayesian network classifier (BNC) requires learning the structure (graph) of the graphical model and its parameters so that the learned BN will excel in inference of a specific variable, that is the class variable, and not necessarily of all variables. When focusing on structure learning, exhaustively searching the space of possible graphs is infeasible [2], and thus search and score (S&S) structure learning algorithms sub-optimally search the space and select the structure achieving the highest value of a score [2,3,12]. However, until very recently, all S&S structure learning algorithms used a generative score, and thereby led to learning a generative model that is not specific to classification, but to general inference.

* Corresponding author. Tel.: +972 8 6479375; fax: +972 8 6472958.
E-mail addresses: kelnerr@bgu.ac.il (R. Kelner), boaz@bgu.ac.il (B. Lerner).

0888-613X/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2011.10.006

Common approaches for learning a model are roughly partitioned into generative, discriminative, or a combination of both approaches [13–15]. Generative models (e.g., BN, density estimation) summarize data probabilistically and are more flexible, since the user can bring in conditional independence assumptions, priors, and hidden variables. Generative classifiers learn a model of the joint probability of the variables and the related class label, and use Bayes' theorem to compute the posterior probability of the class variable and make predictions. Discriminative models (e.g., NN and SVM) only learn from data to make accurate predictions by directly estimating the class posterior probability or via discriminant functions, and thus offer the user less flexibility in data representation and inference. The dilemma in the machine learning community regarding which approach of learning – generative or discriminative – is more appropriate for learning a BNC structure, has gained considerable attention in recent years. Most empirical studies demonstrate superiority of the discriminative approach with respect to the accuracy of the learned BNC [5–7,16]. For some models, however, it is shown [13] that the choice of either of the approaches depends on the sample size; for small sample sizes, the generative approach, which relatively quickly approaches its asymptotic error, is favored, whereas the discriminative method is preferred for larger sample sizes. Classifiers combining generative and discriminative modeling use generative models, yet estimate the model structure and/or parameters to reduce the classification error.

Several studies [4–7,10,17] have demonstrated that BNC structures learned using generative scores do not usually contribute to high classification accuracy since there is lack of agreement between the score used for learning and the score used for evaluation, i.e., the classification accuracy. That is, classifiers based on structures having high values of the generative scores are not necessarily highly accurate. To address this issue, we propose risk minimization by cross validation (RMCV) for a classification-oriented score and S&S algorithm for learning unrestricted BNCs. Note that other uses of classification-oriented scores in learning unrestricted BNCs [7,18] are in a somewhat different context. Moreover, RMCV is an extension to common use of classification-oriented scores in learning restricted-BNCs and non-BN classifiers. While commonly used S&S algorithms use likelihood-based scores suitable for general inference, RMCV minimizes an empirical estimation of the classification error rate, and thereby learns highly accurate BNCs. That is, RMCV performs discriminative learning of a generative (BN) model. This model does not need to estimate the true distribution, generate data from this distribution, or infer about any non-class variable. It needs to perform a discriminative classification task. RMCV learns generative models that are complicated, only to discriminate accurately among classes.

In the beginning, we suggest and compare several variants of the RMCV score and algorithm. We further show that the RMCV score is better suited for classification than any other score commonly used for learning a BNC, i.e., compared with other scores and BNCs, the accuracy of an RMCV-based classifier increases monotonically with the improvement in the value of the RMCV score. Then, we compare the classifier learned using RMCV with likelihood-based, conditional-likelihood-based, and other BNCs, as well as with non-BN classifiers, such as NN, SVM, and CT. This involves nine leading BNCs and five state of the art non-BN classifiers in a most extensive comparison of BNCs and non-BN classifiers using twenty-two real-world datasets. The comparison demonstrates that an RMCV-based classifier is faster and significantly more accurate than all BNCs and comparable (usually favorably), with respect to accuracy, with the non-BN classifiers. These are encouraging news for researchers and practitioners who appreciate the benefits of the BN model, but once they encounter a classification task, replace the BN with a traditional classifier and thereby lose the BN benefits.

We begin by reviewing BN in Section 2. In Section 3, we focus on learning a BNC using common scores. The RMCV score and algorithm are presented in Section 4, and classifiers learned using RMCV are experimentally compared to other BNCs and non-BN classifiers in Section 5. Section 6 describes recent studies in learning a BNC. Finally, we draw conclusions and summarize the study in Sections 7 and 8, respectively.

2. Bayesian networks

A BN model $\mathcal{B}$ for a set of random variables $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$, each having a finite set of mutually exclusive states, consists of two main components, $\mathcal{B} = (\mathcal{G}, \Theta)$. The structure $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is a directed acyclic graph (DAG). $\mathbf{V}$ is a finite set of nodes of $\mathcal{G}$ corresponding to $\mathbf{X}$, and $\mathbf{E}$ is a finite set of directed edges of $\mathcal{G}$ connecting $\mathbf{V}$. Θ is a set of parameters that quantify the structure. The parameters are local conditional probability distributions (or densities), $P(X_i = x_i|\mathbf{Pa}_i, \mathcal{G})$, for each $X_i \in \mathbf{X}$ conditioned on its parents in the graph, $\mathbf{Pa}_i \subset \mathbf{X}$. In this study, we are interested only in discrete variable BNs and complete data.

The joint probability distribution over $\mathbf{X}$ given $\mathcal{G}$ – assumed to encode this distribution – is the product of these local probability distributions [2,3],

$$P(\mathbf{X} = \mathbf{x}|\mathcal{G}) = \prod_{i=1}^{n} P(X_i = x_i|\mathbf{Pa}_i, \mathcal{G}), \tag{1}$$

where $\mathbf{x}$ is the assignment of states to the variables in $\mathbf{X}$ and x_i is X_i's state.

During inference, the conditional probability distribution of a subset of nodes in the graph (the 'hidden' nodes) given another subset of nodes (the 'observed' nodes) and the BN model is calculated. A common method for exact inference is the junction tree algorithm [19], but when there is only one hidden node (e.g., the class node in classification), direct inference based on (1) and Bayes' rule is more feasible. Note that the computation of conditional probability distributions for inference depends on the graph. Thus, a structure, either based on expert knowledge or learned from the data, must first be obtained.

The S&S approach to learning a structure from data [2,12] comprises a search for the structure achieving the highest score, e.g., hill-climbing (HC) [3,12], and a score, generally the Bayesian score [2],

$$P(\mathcal{G}|D) = \frac{P(D|\mathcal{G})P(\mathcal{G})}{P(D)} = \frac{P(D,\mathcal{G})}{P(D)} \tag{2}$$

for a structure $\mathcal{G}$ given a dataset $D = \{v_1, v_2, \ldots, v_N\}$, which is a random sample of N independent instances from the joint probability distribution of $\mathbf{X}$.

3. Learning a Bayesian network classifier

It is clear from (2) that a score should reflect a correspondence between the structure and the data. The minimum description length (MDL) score [20] can approximate $P(D|\mathcal{G})$ – the marginal likelihood [3] – but [4] argued that this score is not suitable for classification and recommended the class-conditional log likelihood (CLL) (as opposed to log likelihood (LL)),

$$CLL(\mathcal{G}|D) = \sum_{i=1}^{N} \log P(c_i|v'_i), \tag{3}$$

where the vector v_i for the ith instance in D consists of a feature vector v'_i and a class label c_i, so that $v_i = (c_i, v'_i)$. Notice that $CLL(\mathcal{G}|D) = \sum_{i=1}^{N} \log P(v_i) - \sum_{i=1}^{N} \log P(v'_i) = LL(\mathcal{G}|D) - \sum_{i=1}^{N} \log P(v'_i)$.

By maximizing CLL, the structure that best approximates the probability of predicting the class given feature values for every instance is learned [5]:

$$P(c^N|v'^N, \mathcal{G}) = \frac{P(c^N, v'^N|\mathcal{G})}{P(v'^N|\mathcal{G})} = \frac{P(D|\mathcal{G})}{\sum_{c'^N} P(c'^N, v'^N|\mathcal{G})}, \tag{4}$$

where v'^N consists of all feature vectors and c^N consists of all possible combinations of the r_C states of the class variable C in a random sample D of size N. The computation of this score is infeasible, since the sum in the denominator is exponential in N (r_C^N terms), let alone score maximization.

An approximation [5] considers the left-hand side of (4) and the marginal likelihood (2) as the supervised and unsupervised marginal likelihoods, respectively. The marginalization over the parameters in (4) is:

$$P(c^N|v'^N, \mathcal{G}) = \int_{\Theta} P(c^N|v'^N, \Theta, \mathcal{G}) P(\Theta|v'^N, \mathcal{G})\, d\Theta, \tag{5}$$

and its approximation [5] using a single term is

$$P(c^N|v'^N, \mathcal{G}) \approx P(c^N|v'^N, \hat{\Theta}, \mathcal{G}). \tag{6}$$

$\hat{\Theta}$ is the parameter configuration maximizing the parameter posterior probability, $P(\Theta|v'^N, c^N, \mathcal{G})$, which is a different solution than that derived when maximizing $P(c^N|v'^N, \Theta, \mathcal{G})$. However, there is no general closed-form solution to the supervised form of the score and the posterior is not decomposable in this case, hence the need for approximation [4].

Another predictive local criterion (LC) [3] for learning a BNC [5] is based on the prequential approach [21],

$$LC(D, \mathcal{G}) = \sum_{i=1}^{N} \log P(c_i|\{v_j\}_{j=1}^{i-1}, v'_i, \mathcal{G}). \tag{7}$$

Other cumulative logarithmic loss scores [5] use 10-fold cross validation (CV) or leave-one-out, which are reputable methods for model selection [22], both described here under the general term CV-K, where $K = 10$ or $K = N$ for the two cases, respectively. A score using CV-K for predicting a class is defined:

$$CV_K(D, \mathcal{G}) = \sum_{k=1}^{K} \sum_{i=1}^{N/K} \log P(c_{i+A_k}|D \backslash D_k^K, v'_{i+A_k}, \mathcal{G}), \tag{8}$$

where $A_k = (k-1)N/K$ and $D_k^K = \{v_{j+A_k}\}_{j=1}^{N/K}$ is a validation set derived from the training set D.

Using either of the supervised (conditional) marginal likelihood scores ((5), (7), or (8)) for learning a BNC is asymptotically optimal. However, for a finite sample, though a high score value may indicate correct classification, it cannot guarantee it.

A score that measures the degree of compatibility between a possible state of the class variable and the correct class is the 0/1 loss function:

$$L(c_i, \widehat{c}_i) = \begin{cases} 0, & c_i = \widehat{c}_i, \\ 1, & c_i \neq \widehat{c}_i, \end{cases} \tag{9}$$

where c_i is the true class label and $\widehat{c}_i$ is the estimated class label for the ith instance.

To demonstrate the difference between a class-conditional score and the 0/1 score, consider a two-class classification problem, two candidate classifiers A and B, and two instances v'_1 and v'_2 [17]. Classifier A predicts the correct class for instances v'_1 and v'_2 with probabilities of 0.3 and 0.51, respectively, while classifier B predicts the correct class for the same two instances with probabilities of 0.45 and 0.49. Since the sum of log probabilities (i.e., "log-loss" score) is larger for classifier B than for classifier A, the former classifier will be selected. However, if evaluating the 0/1 loss values, classifier B is inaccurate for both v'_1 and v'_2, whereas classifier A is correct for v'_2. Thus, choosing classifier A based on the 0/1 loss score is more sensible for classification than choosing classifier B based on the log-loss score. We therefore suggest using the classification-specific 0/1 loss function for learning BNCs of enhanced classification accuracies.

4. Risk minimization by cross validation

Instead of selecting a structure based on summation of supervised marginal likelihoods over the dataset ((7) or (8)), we suggest selecting a structure based on summation of false decisions about the class state over the dataset. Our score is based on risk minimization [23] using the 0/1 loss function measured on a validation set. The training set D is divided into a validation set D^K (having N/K of N instances) and an effective training set (having $N(K-1)/K$ instances). The classification error rate (0/1 loss) is measured for each candidate structure and in any iteration of the search on D^K. During learning, no use of a (third) test set is made. As part of a CV experiment, the score of a candidate structure is computed by averaging the error rates over K non-overlapping validation sets. Since the structure that minimizes the empirical risk is being searched for, we call the score *risk minimization* by *cross validation* (RMCV)[1] $\left(\text{and we deliberately do not simplify } \frac{1}{K}\frac{K}{N}\right)$:

$$RMCV_K(D,\mathcal{G}) = \frac{1}{K}\sum_{k=1}^{K}\frac{K}{N}\sum_{i=1}^{N/K} L\left(c_{ki}, \underset{c\in\{c_1,..,c_{r_C}\}}{\operatorname{argmax}} P(C=c|D\backslash D_k^K, v'_{ki}, \mathcal{G})\right), \tag{10}$$

where $v_{ki} = (c_{ki}, v'_{ki})$ is the ith instance of D_k^K and $L(,)$ is the 0/1 loss function (9). Being a CV-based score, RMCV is easy to implement and computationally feasible (see, for example, (6)) and it depends on only one parameter (K). Further, it is argued [5] that a CV-based score can be regarded as an approximation of a factorization of the supervised marginal likelihood (4). Note that the RMCV score is normalized by the dataset size N, whereas (7) and (8) are not. Although normalization has the same effect on all learned structures, it can clarify the meaning of the score (i.e., an error rate) and help comparing scores over datasets. Moreover, sharing the same range of values ($[0, 1]$), RMCV establishes its correspondence to classification accuracy.

To compute RMCV, the candidate structure has to be turned into a classifier by learning its parameters. Local probabilities are modeled using the unrestricted multinomial distribution [3] (assuming discrete variables), where the distribution parameters are obtained using maximum likelihood (ML) [2], similar to [5]. Moreover, Grossman and Domingos [6] argued based on experiments that ML parameter estimation does not deteriorate the results compared to maximum conditional likelihood estimation, which can only be obtained by computationally expensive numerical approximation. Learning a BN rather than a structure has an additional cost of parameter learning, though this cost is negligible while using ML estimation and fully observed data.

To prevent over-fitting the training set, RMCV is computed by K-fold CV. Thus, over-fitting is controlled through the score itself, and not through the search dynamics as in other algorithms discussed later. Also, note that the same measure is used for learning the BNC and for evaluating its accuracy, which makes learning oriented towards classification. Similar to CLL-based scores [5,6], RMCV is not decomposable.

A suggested S&S structure learning algorithm consists of the RMCV score and a simple HC search:

Algorithm: RMCV
Input: An initial DAG, $\mathcal{G}$; A training set that is partitioned to K mutually exclusive validation sets, $D = \{D_k^K\}_{k=1}^K$.
Output: BN model ($\mathcal{G}$,Θ).

```
compute RMCV_K(D, G)
converged := false
While converged = false
    For each G' ∈ Neighborhood(G)
        compute RMCV_K(D, G')
    G* := arg min_G' RMCV_K(D, G')
    If RMCV_K(D, G*) < RMCV_K(D, G)
        Then G := G*
        Else converged := true
Θ := LearnParameters(D, G)
Return (G, Θ)
```

[1] The proposed score and algorithm are based on [17] with changes.

Algorithm description: The RMCV algorithm starts with an initial graph and a training set that is divided into K mutually exclusive validation sets. For each $k \in \overline{1,K}$, the parameters are learned using the effective training set $D\backslash D_k^K$ and the error rate is evaluated using D_k^K. The average error rate over the K validation sets D_k^K, $\forall k \in \overline{1,K}$ is the RMCV score (10). The initial graph and its score are kept as the current graph and score, respectively. Next, the neighborhood of the current graph is generated by all single edge additions, deletions, and reversals. Since only DAGs are allowed, any cyclic directed graphs in the neighborhood are excluded [24]. The graph having the lowest RMCV score in the neighborhood is chosen. Its score is compared to the current score, and the search is halted if the current score is lower than or equal to the score of the chosen graph. If, however, the chosen graph has a lower score than the current score, it becomes the current graph and the procedure repeats itself.

Algorithm initialization: The algorithm is not limited to any specific DAG to start from. Here we start from either the empty graph, or the naive Bayesian classifier (NBC) [25] – a learning-free structure, which is obtained with virtually no computational effort, and is considered a state of the art classifier [4].

Algorithm termination: During structure evaluation, only the effective training sets $D\backslash D_k^K$, $\forall k \in \overline{1,K}$ are used for parameter learning. Yet, once the search for a structure completes, the role of the validation sets is finished and the entire training set D can be used for a more reliable parameter learning for this structure, rendering the structure a BN. The algorithm then returns the learned BN defined by $(\mathcal{G}, \Theta)$.

Note that when using ML parameter estimation, fully observed data, and the suggested search, there is no need to reassess all of the parameters for the different structures during each HC step. Parameters are changed only for nodes whose sets of parents have been modified. In case of an addition or deletion of an edge, only one node is affected, and in case of a reversal of an edge, only two nodes are affected. For the same reason, it is beneficial to keep the history of the probability calculations, using the factorization of (1), for the initial/current DAG.

5. Experiments, results, and analysis

Three experiments were conducted to evaluate RMCV. In Section 5.1, RMCV is compared to risk minimization holdout (RMHO) [17] – a variant of RMCV. Further motivation for using the RMCV score is given in Section 5.2, where several common scores are compared to RMCV with respect to classification accuracy. In Section 5.3, a comparison of an RMCV-based classifier to other state of the art classifiers is presented.

Twenty-two UCI natural datasets [26] were used in Sections 5.1 and 5.3. Their basic characteristics are presented in Table 1. Continuous features were quantized using the MLC++ library [27] and instances with missing values were eliminated, similarly to [4] and [6]. The BN implementation was aided by the BNT [28] and SLP [29] toolboxes. Experiments with SVM and NN were performed using LIBSVM [30] and PRTOOLS [31], respectively.

To reduce the sensitivity to CV data partitioning, ten random permutations were created for each dataset (Sections 5.1 and 5.3). For each permutation, the classifiers were evaluated using a CV5 experiment. The data had been randomly divided into five complementary subsets, and in each of the five folds of the experiment a fifth of the dataset was used for the test and the remainder of the dataset for training. During the operation of the RMCV algorithm, the training set was further divided into four mutually exclusive validation sets needed for computing the RMCV score ($K = 4$). One should not confuse

Table 1
Characteristics of the studied datasets.

	Dataset	# Classes	# Features	# Instances
1	Australian	2	14	690
2	Breast	2	9	683
3	Chess	2	36	3196
4	Cleve	2	11	296
5	Corral	2	6	128
6	Crx	2	15	653
7	Diabetes	2	8	768
8	Flare	8	10	1389
9	Glass	6	9	214
10	Glass2	2	9	163
11	Hayes	3	4	160
12	Heart	2	13	270
13	Hepatitis	2	19	80
14	Iris	3	4	150
15	Letter	26	16	20,000
16	Lymphography	4	18	148
17	Mofn	2	10	1324
18	Pima	2	8	768
19	Shuttle	6	8	5800
20	Tic-Tac-Toe	2	9	958
21	Vehicle	4	18	846
22	Voting	2	16	232

this CV4, which was internal to the RMCV algorithm, with the CV5 used for classifier training and evaluation. For each dataset, all evaluated algorithms were compared using the same permutations and CV folds of the 10xCV5 experiment.

Following [32], accuracies of all classifiers were compared using the Friedman test, which is a non-parametric test recommended for comparing several algorithms over multiple datasets. This test was followed by either the Bonferroni–Dunn post-hoc test, which compares all algorithms with a control algorithm (RMCV in our case), or the Wilcoxon signed-rank test, which compares pairs of algorithms (RMCV vs. other) over multiple datasets and, for our experimental setting, is preferred over the parametric t-test.

5.1. Risk minimization holdout

In the first experiment, the RMCV classifier was compared to a classifier based on the RMHO score [17]:

$$RMHO_K(D, \mathcal{G}) = \frac{K}{N}\sum_{i=1}^{N/K} L\left(c_i, \underset{c\in\{c_1,\ldots,c_{r_C}\}}{\operatorname{argmax}} P(C = c|D\backslash D^K, v_i', \mathcal{G})\right), \tag{11}$$

where $v_i = (c_i, v_i')$ is the ith instance of D^K and $L(,)$ is the 0/1 loss function. The RMHO algorithm [17] is similar to the RMCV algorithm, but uses (11).

There are two substantial differences between RMCV and RMHO. While the RMHO score is based on a single holdout experiment, the RMCV score is based on a CV experiment. RMCV uses the training data repeatedly and thus much more effectively, but RMHO is cheaper computationally. Second is that while the RMCV algorithm uses the same partitioning of the training data throughout the entire HC search, the RMHO algorithm selects a random D^K in each step of the search, and thereby reduces the score consistency and increases the score variability.

Experiments with both algorithms, where the search starts from either the NBC structure or the empty graph, were conducted using $K = 4$ folds in the internal CV (a holdout ratio of $1/4$, in the case of RMHO). Also tested was a variant of the RMHO algorithm, where D^K is fixed during the entire HC search (referred to as RMHO-f, 'f' for 'fixed set'). By fixing the set, this variant copes with the increased variability of RMHO. In a third variant that is proposed, the search procedure avoids those neighboring graphs that differ from the current graph by a single edge reversal (referred to as RMHO-ar, 'ar' for 'avoids reversals'). In RMHO-ar, D^K is fixed as well. By considering only edge additions and deletions and avoiding reversals, we reduce the size of the graph neighborhood and thereby also the search run-time, but this may undermine classification accuracy. Results for the comparison of RMCV and the three RMHO variants are presented in Table 2.

Table 2 shows that avoiding edge reversals does not undermine accuracy, as results are almost completely identical for RMHO-f and RMHO-ar regardless of the initial structure. Thus, we will not address the latter further. A deeper inspection of the table shows that it is generally advantageous to start the HC search from the NBC structure, but not always. It also seems that RMHO has a stochastic nature, in comparison to RMHO-f, leading to a few good results, but also a fair number of failures due to the repeated random selection of the holdout dataset during the search.

Table 2
Mean (std) of classification accuracies for RMCV and three RMHO variants. Bayesian network initial structures appear in heading brackets. Bold/italic fonts are used, respectively, for the best/worst classifiers for a dataset.

Dataset	RMCV (NBC)	RMHO (NBC)	RMHO-f (NBC)	RMHO-ar (NBC)	RMCV (empty)	RMHO (empty)	RMHO-f (empty)	RMHO-ar (empty)
Australian	**86.14** (2.3)	86.01 (2.6)	85.65 (2.4)	85.65 (2.4)	85.10 (2.5)	85.33 (2.6)	*84.86* (2.9)	*84.86* (2.9)
Breast	**96.91** (1.4)	96.85 (1.4)	96.79 (1.4)	96.79 (1.4)	96.56 (1.6)	*94.79* (2.0)	95.54 (1.8)	95.54 (1.8)
Chess	**97.43** (0.8)	*92.78* (1.8)	94.93 (1.1)	94.93 (1.1)	94.25 (0.7)	94.01 (1.0)	93.83 (1.4)	93.83 (1.4)
Cleve	81.53 (3.8)	**82.53** (3.6)	81.87 (3.4)	81.87 (3.4)	76.73 (5.7)	*73.37* (4.9)	74.80 (5.2)	74.80 (5.2)
Corral	**99.57** (2.8)	92.50 (8.4)	91.29 (7.2)	91.29 (7.2)	81.86 (12.2)	*72.14* (9.6)	74.71 (12.2)	74.71 (12.2)
Crx	**85.71** (2.0)	85.67 (2.3)	*85.31* (2.6)	*85.31* (2.6)	**85.71** (2.0)	85.67 (2.3)	*85.31* (2.6)	*85.31* (2.6)
Diabetes	74.03 (2.6)	73.73 (3.4)	73.73 (3.3)	73.73 (3.3)	**74.50** (2.5)	73.64 (2.8)	*73.29* (2.8)	*73.29* (2.8)
Flare	84.03 (1.9)	*81.20* (2.6)	82.57 (2.6)	82.57 (2.6)	**84.32** (1.7)	**84.32** (1.7)	**84.32** (1.7)	**84.32** (1.7)
Glass	57.39 (6.4)	57.13 (6.1)	**57.43** (7.0)	**57.43** (7.0)	56.87 (6.7)	*50.35* (6.7)	54.26 (7.3)	54.26 (7.3)
Glass2	75.03 (7.8)	74.97 (7.8)	**75.20** (7.7)	**75.20** (7.7)	75.03 (7.8)	*72.40* (8.2)	73.49 (8.3)	73.49 (8.3)
Hayes	**84.75** (5.1)	82.31 (6.5)	82.19 (6.4)	82.19 (6.4)	84.38 (6.1)	*64.69* (13.8)	72.56 (12.6)	72.56 (12.6)
Heart	81.52 (4.7)	**82.41** (5.1)	81.70 (5.7)	81.70 (5.7)	74.59 (6.2)	*72.37* (5.6)	74.00 (6.6)	74.00 (6.6)
Hepatitis	81.75 (9.6)	**82.25** (9.6)	81.75 (9.6)	81.75 (9.6)	*77.00* (11.2)	77.13 (10.4)	77.63 (11.2)	77.63 (11.2)
Iris	93.73 (3.9)	*93.33* (3.9)	93.67 (4.1)	93.67 (4.1)	94.60 (3.2)	**95.27** (2.8)	94.93 (3.2)	94.93 (3.2)
Letter	**83.40** (0.5)	81.58 (1.1)	82.93 (0.5)	82.93 (0.5)	83.24 (0.7)	*79.82* (1.7)	82.40 (0.8)	81.94 (0.8)
Lymphography	**79.06** (6.6)	77.81 (6.2)	76.94 (6.7)	76.94 (6.7)	76.13 (7.4)	*69.13* (6.8)	70.00 (7.0)	70.00 (7.0)
Mofn	**94.77** (5.8)	89.20 (2.7)	90.82 (3.9)	90.82 (3.9)	*77.98* (2.3)	*77.98* (2.3)	*77.98* (2.3)	*77.98* (2.3)
Pima	**74.71** (3.0)	74.44 (2.9)	74.31 (2.8)	74.31 (2.8)	73.96 (3.3)	*71.45* (3.9)	73.32 (3.1)	73.32 (3.1)
Shuttle	99.49 (0.2)	*99.41* (0.2)	99.45 (0.2)	99.45 (0.2)	**99.61** (0.1)	99.56 (0.2)	99.57 (0.1)	99.57 (0.1)
Tic-Tac-Toe	**89.98** (5.9)	73.31 (4.5)	80.75 (7.0)	80.52 (7.0)	79.13 (9.1)	*69.00* (3.5)	72.38 (6.7)	72.38 (6.7)
Vehicle	**69.51** (3.1)	66.52 (3.7)	68.39 (3.2)	68.39 (3.2)	**69.51** (3.8)	*64.73* (4.1)	67.24 (3.4)	67.24 (3.4)
Voting	94.71 (2.7)	93.83 (3.5)	*93.75* (3.2)	*93.75* (3.2)	96.88 (2.1)	96.83 (2.0)	**96.92** (1.9)	**96.92** (1.9)
Average	**84.78** (3.8)	82.72 (4.1)	83.25 (4.2)	83.24 (4.2)	81.72 (4.5)	*78.36* (4.5)	79.70 (4.8)	79.68 (4.8)

Table 3
Statistical significance tests for the accuracies of RMCV, RMHO, and RMHO-f in Table 2.

Algorithm	Average rank	Post-hoc tests significance values	
		Bonferroni–Dunn	Wilcoxon
The Friedman test for RMCV and RMHO variants			
RMCV (NBC)	2.00		
RMCV (empty)	3.11	Non-significant	0.007
RMHO-f (NBC)	3.16	Non-significant	<0.001
RMHO (NBC)	3.45	0.0496	0.002
RMHO-f (empty)	4.39	<0.001	<0.001
RMHO (empty)	4.89	<0.001	<0.001
The Friedman test statistics			
N	22	Chi-square	32.844
Degrees of freedom	5	Significance value	<0.001

Table 3 contains results of the Friedman test for the accuracies of RMCV, RMHO, and RMHO-f. The null hypothesis that all BNCs are the same had been rejected with high confidence, and post-hoc tests were applied to validate which algorithms are different. RMCV (NBC) is ranked highest according to the Friedman test. Based on the Bonferroni–Dunn post-hoc test (right part of Table 3) with RMCV (NBC) as the control classifier, RMCV (NBC) is superior to RMHO (NBC/empty) and RMHO-f (empty) with a significance level of at least $p < 0.05$. The Wilcoxon signed-rank test (right part of Table 3) is less conservative, and finds RMCV (NBC) to be superior (with $p < 0.01$) to all examined classifiers. The Friedman test ranking demonstrates how starting the HC search from the NBC structure is advantageous over starting from the empty graph. We can also see that RMCV is ranked before all RMHO variants, regardless of the search initial structure. For these reasons, the RMHO algorithm had been excluded in Section 5.3 and the experiments were continued with the superior RMCV (NBC) algorithm.

5.2. Between score and accuracy

In this section, we explore the relation between a score used for learning a BNC and that BNC accuracy. We expect from a 'good' score that an improvement in its value for a structure will lead to an improvement in the accuracy of a BNC that is based on the structure, and that the accuracy corresponding to the best score value achieved during the search will be the highest possible. To test how 'good' a score is, we examined the relation between the score values and the classification accuracies of all possible DAGs for a problem. However, the number of possible DAGs increases more than exponentially with the number of variables, n [33]

$$r(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} r(n-i) = n^{2^{O(n)}}. \tag{12}$$

According to (12), $r(2) = 3$, $r(3) = 25$, $r(5) = 29{,}281$, and this number grows rapidly to $r(6) = 3{,}781{,}503$ and $r(10) \approx 4.20 \cdot 10^{18}$. That is, it is infeasible to consider all DAGs, except in low-dimensional domains.

Therefore, we experimented with three datasets with $n = 5$ variables (including the class variable). These are the Iris dataset (Table 1), the UCI Balance dataset, which has 625 instances and three classes, and a synthetic dataset with 10,000 instances generated from the BN shown in Fig. 1. While creating the synthetic dataset, we opted for a structure in which the Markov blanket [1] of the class node consists of one parent, one child, and one co-parent of that child (Fig. 1(a)). For simplicity, all variables were chosen to be binary. Probabilities were chosen almost arbitrarily, avoiding uniform distributions (Fig. 1(b)). For the small to medium Iris and Balance datasets, CV5 experiments were conducted. For the large synthetic dataset, a single holdout experiment used half of the data for training and the other half for the test. For all three datasets, each of the MDL, K2, CMDL, CLL (Part I of Table 4), RMHO, RMHO-f (Section 5.1), and RMCV scores was evaluated for each of the 29,281 possible DAGs using the training set, and the DAG accuracy was computed using the training and test sets.

Figs. 2-6 plot the relations between training/test accuracies and score values of all DAGs for each dataset and score evaluated. Each dot in a figure represents one of the 29,281 possible DAGs. The left side plots in Fig. 2 show for the MDL score that a better (lower) score value does not necessarily mean a higher training classification accuracy. Generalization is much more important than the training set performance, and the right side plots show that the most accurate DAGs do have relatively low MDL score values. However, except for the synthetic dataset, many DAGs with low MDL score values do not classify accurately (Iris and Balance), and other DAGs having high score values classify quite well (Iris). The synthetic dataset has several advantages over the other two datasets since it is noise-free, has many instances, and its instances are i.i.d. These advantages are reflected in the dataset asymptotic behavior. While we do care about asymptotic behavior, real-world problems usually behave differently (see the Iris and Balance datasets).

Similar phenomena were exhibited with the K2 score (and hence results are not shown), supporting our previous results [37] that BNCs learned based on K2/MDL are not necessarily accurate. It should be mentioned that the more instances that exist in a dataset, the more similar are the MDL and (logarithmic) K2 score values (though their signs are opposite), since MDL is a large sample approximation to the marginal likelihood (K2 score) [3].

A visual comparison between the results for the CMDL score (Fig. 3) and those for the MDL score (Fig. 2) reveals only a slight difference. Considering Table 4, the difference between MDL and CMDL is reflected in the difference between LL (MDL)

Table 4
Algorithms and classifiers compared to RMCV.

I. Search and score BN algorithms:

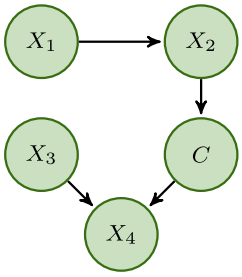
MDL [20]
Score: $MDL(D,\mathcal{G}) = \frac{1}{2}Dim(\mathcal{G})\log N - LL(\mathcal{G}|D) = \frac{1}{2}Dim(\mathcal{G})\log N - \sum_{i=1}^{N}\log P(v_i)$, where $Dim(\mathcal{G})$, N, and $\frac{1}{2}\log N$ are the number of parameters in $\mathcal{G}$, data sample size, and number of bits used to encode each parameter of $\mathcal{G}$, respectively.
Search algorithm: The algorithm is initialized using either NBC or the empty graph and uses HC search.

K2 [2]
Score: $P(D,\mathcal{G}) = P(\mathcal{G})\prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{(r_i-1)!}{(N_{ij}+r_i-1)!}\prod_{k=1}^{r_i}N_{ijk}!$, where n, q_i, r_i, and N_{ijk} are the numbers of: nodes in $\mathcal{G}$, configurations of the parents of the ith node, mutual exclusive states of the ith node, and instances in which the ith node is in its kth state and its parents are in their jth configuration in the dataset D, respectively, and $N_{ij} = \sum_{k=1}^{r_i}N_{ijk}$.
Search algorithm: The algorithm is initialized using either NBC or the empty graph and uses HC search (instead of the K2 search).

BNC-MDL [6]
Score: $CMDL(\mathcal{G}|D) = \frac{1}{2}Dim(\mathcal{G})\log N - CLL(\mathcal{G}|D)$, where CLL was defined in (3), and $Dim(\mathcal{G})$, N, and $\frac{1}{2}\log N$ were defined for the MDL score.
Search algorithm: The algorithm is initialized using the empty graph and uses HC search.

BNC-2P [6]
Score: The CLL score is used instead of the CMDL score.
Search algorithm: The algorithm is initialized using the empty graph and is limited to DAGs in which each node has a maximum of two parents.

II. Non-BN classifiers:
Naive Bayesian classifier (NBC) [25], tree augmented NBC (TAN) [4], classification tree (CT) [34] with pruning, multilayer perceptron neural network (NN) [35], and support vector machine (SVM) [36] with either a linear, polynomial, or Gaussian kernel.



(a) BN structure

$$P(X_1=0)=0.3, P(X_1=1)=0.7;$$
$$P(X_3=0)=0.7, P(X_3=1)=0.3;$$
$$P(X_2=0|X_1=0)=0.7,\ P(X_2=1|X_1=0)=0.3,$$
$$P(X_2=0|X_1=1)=0.3,\ P(X_2=1|X_1=1)=0.7;$$
$$P(C=0|X_2=0)=0.85,\ P(C=1|X_2=0)=0.15,$$
$$P(C=0|X_2=1)=0.25,\ P(C=1|X_2=1)=0.75;$$
$$P(X_4=0|X_3=0,C=0)=0.1,\ P(X_4=1|X_3=0,C=0)=0.9,$$
$$P(X_4=0|X_3=0,C=1)=0.8,\ P(X_4=1|X_3=0,C=1)=0.2,$$
$$P(X_4=0|X_3=1,C=0)=0.3,\ P(X_4=1|X_3=1,C=0)=0.7,$$
$$P(X_4=0|X_3=1,C=1)=0.2,\ P(X_4=1|X_3=1,C=1)=0.8;$$

(b) Probabilities

Fig. 1. BN used to generate the synthetic dataset.

and CLL (CMDL). Following the presentation in Section 3, LL comprises CLL plus a term that estimates the joint distribution of the features. This term dominates CLL for large problems [4]. However, for small problems such as ours ($n = 5$), this is not the case and, practically, there is almost no difference between LL and CLL, as is shown in Figs. 2 and 3 (Iris and Balance). Comparing the plots for CMDL (Fig. 3) and CLL (Fig. 4), we observe differences that are attributed (Table 4) to the penalty term in CMDL that balances CLL and depends on the number of parameters in the graph. That is, DAGs of the almost 30,000 that create Fig. 3 are penalized differently by CMDL, but not by CLL (Fig. 4). Moreover, the CLL score seems to perform very well on the training set; however, at the cost of over-fitting (clearly apparent with the Balance dataset, and to a lesser degree with the Iris dataset). That is, the test classification accuracy starts to decline, although the value of the CLL score keeps growing. There is no over-fitting of the synthetic dataset due to the previously mentioned asymptotic behavior.

Figs. 5 and 6 show the relations between test set classification accuracies and values of one minus the RMHO/RMHO-f and RMCV scores, respectively. Since each such score estimates the training classification error rate of a DAG, we may view one minus the score as a predictor of the test classification accuracy of the DAG. The advantage of RMCV over RMHO/RMHO-f is clear from the figures, as the disparity around the $y = x$ line is smaller for RMCV, meaning that RMCV predicts the test classification accuracy better than RMHO/RMHO-f.

The accuracy predicted by the RMCV/RMHO scores (or by the training set accuracy) can be easily compared to the test classification accuracy and evaluated using the mean squared error (MSE):

$$MSE = \frac{1}{r(n)}\sum_{i=1}^{r(n)} e_i^2, \tag{13}$$

where e_i is the difference between the predicted accuracy due to a score and the test set accuracy of the ith DAG, and $r(n)$ is the number of possible DAGs (12). Graphically, MSE is relative to the sum of squared distances of the $r(5)$ =29,281 dots

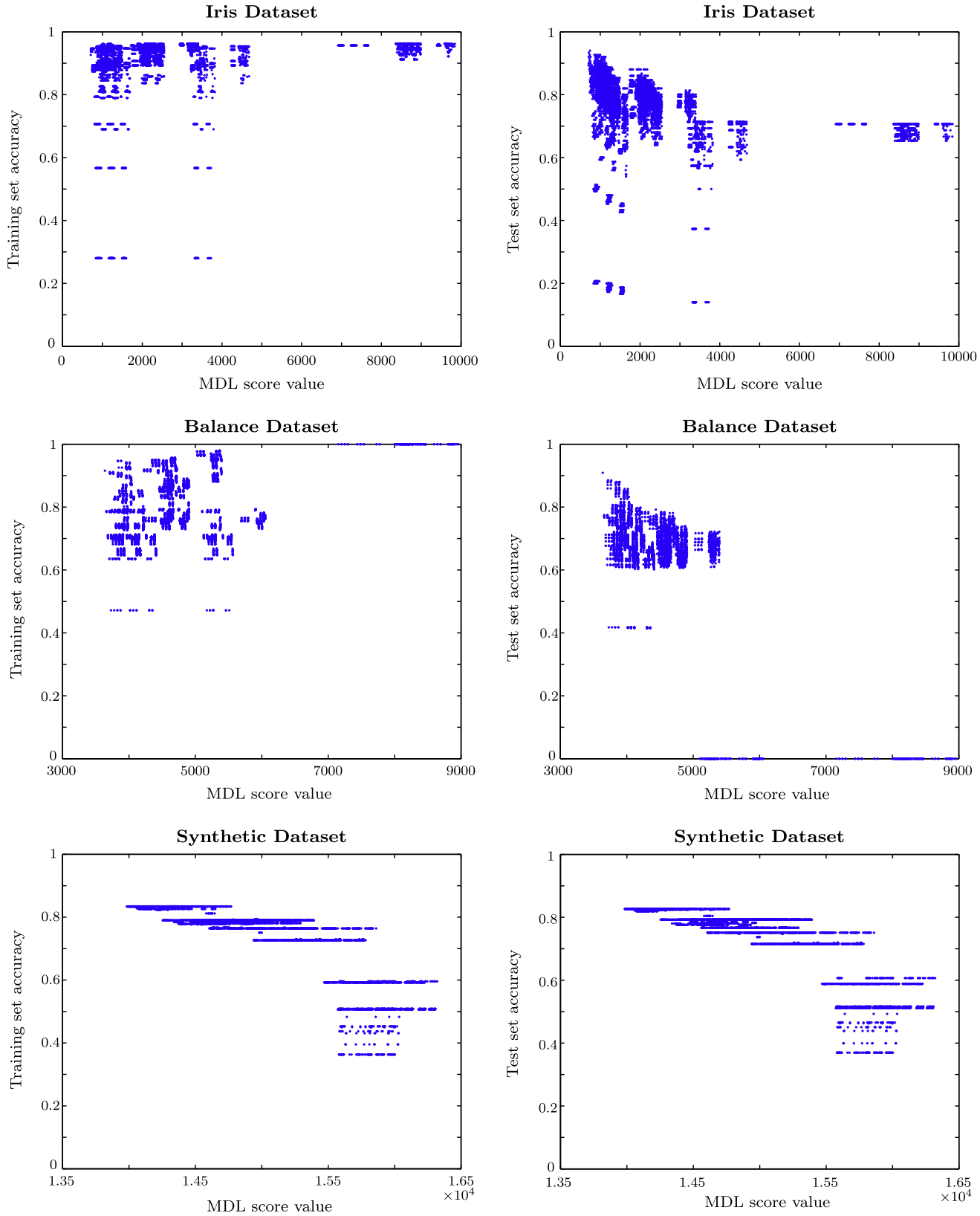


Fig. 2. Training/test set classification accuracies vs. MDL values for all possible DAGs as obtained using the Iris, Balance, and synthetic datasets. Each dot in the figure represents results for a DAG (dots may overlap).

(DAGs) in a plot from the $y = x$ line normalized by $r(5)$. Table 5 demonstrates the advantage of RMCV over RMHO and RMHO-f, with respect to MSE and hence also to prediction accuracy. As demonstrated in Table 5, when there is enough data (synthetic dataset), using the training set accuracy as well as the score values, as a test accuracy predictor, works. However, for the small Iris and Balance datasets, the training set accuracy and the values of the RMHO variants err considerably more than RMCV does, indicating over-fitting of the training set by the formers.

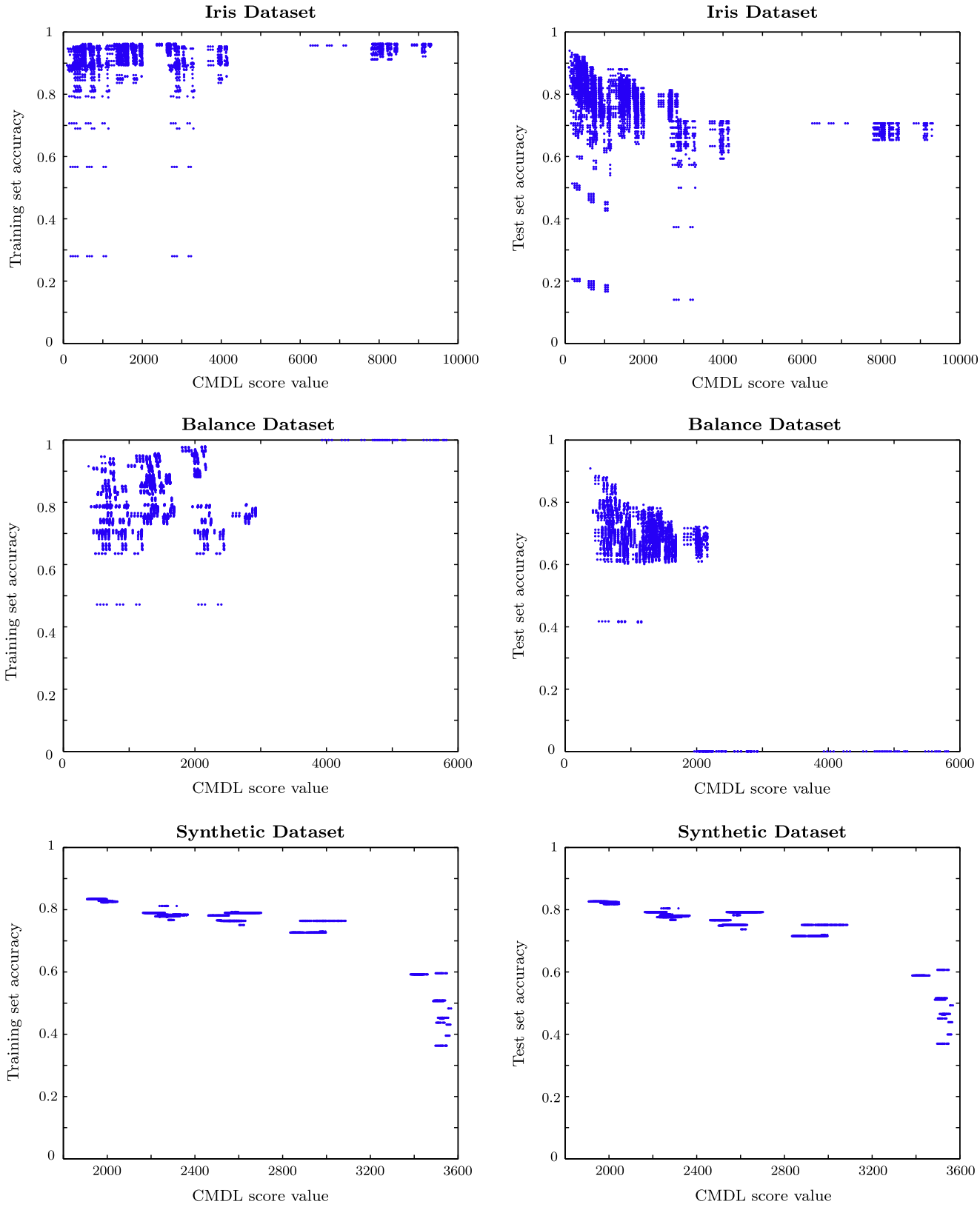


Fig. 3. Training/test set classification accuracies vs. CMDL values for all possible DAGs as obtained using the Iris, Balance, and synthetic datasets. Each dot in the figure represents results for a DAG (dots may overlap).

Generally, we would like to see correspondence between the test accuracy of a classifier, which is learned using a specific score, and the score value, because this will guarantee that every time we select a structure achieving the highest score value, this structure would yield the most accurate classifier. By processing the results for the synthetic dataset, we found that 3766 of the 29,281 legitimate DAGs achieved the highest possible test set accuracy of 82.6%, and that the generating

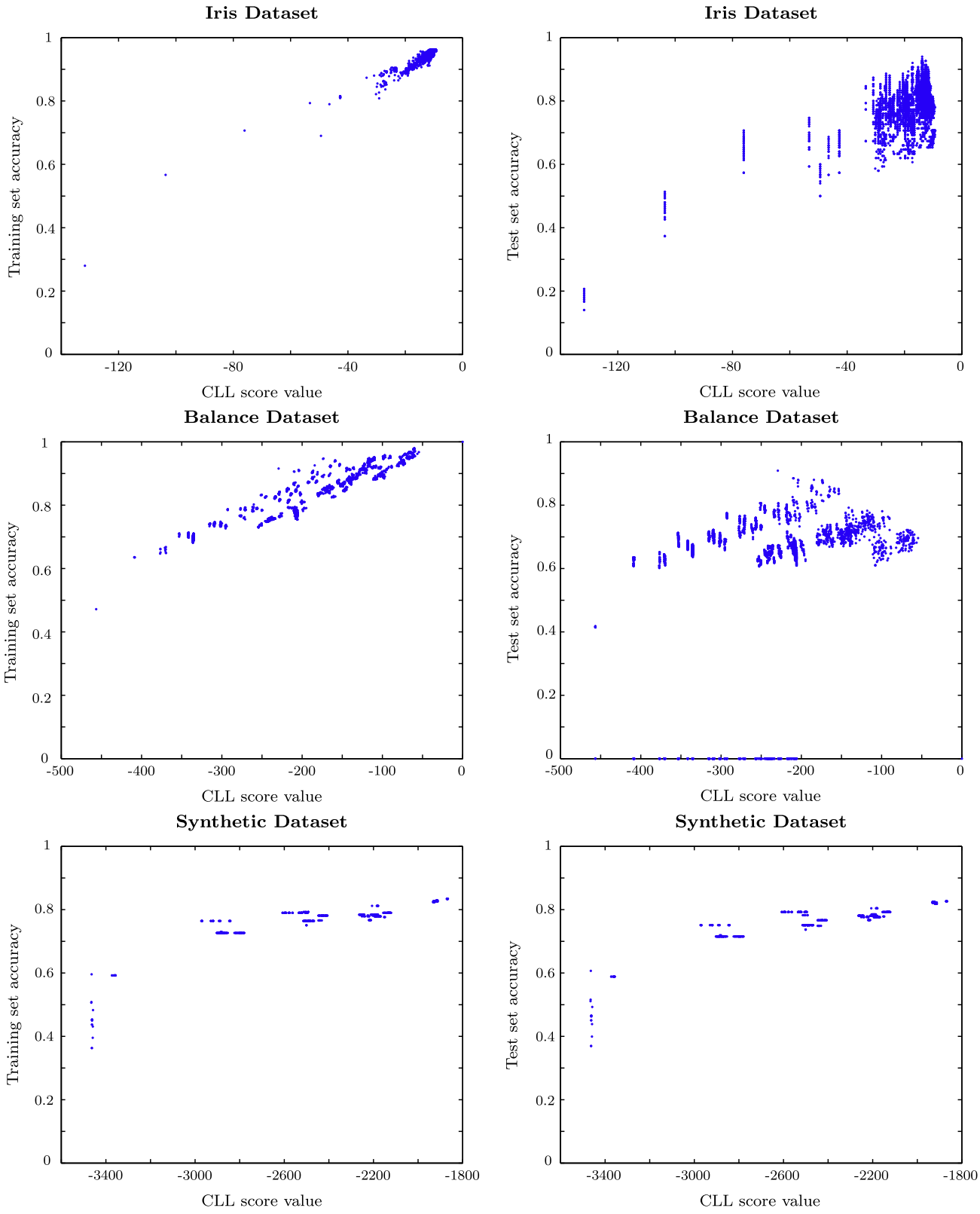


Fig. 4. Training/test set classification accuracies vs. CLL values for all possible DAGs as obtained using the Iris, Balance, and synthetic datasets. Each dot in the figure represents results for a DAG (dots may overlap).

structure (Fig. 1) is included in this group. These and only these DAGs were ranked the highest by RMCV and RMHO-f. This can visually be seen in well-confined points (each representing 3766 DAGs) in the upper-right corners of the graphs for the test accuracy on the synthetic dataset of RMHO-f (Fig. 5) and RMCV (Fig. 6). On the other hand, RMHO ranked the highest only 55 of the 3766 structures that yielded the highest classification accuracy (none of these 55 is the generating structure). This can be seen visually in the right end (representing 55 DAGs) of a wide line (representing the same 3766 DAGs) in the

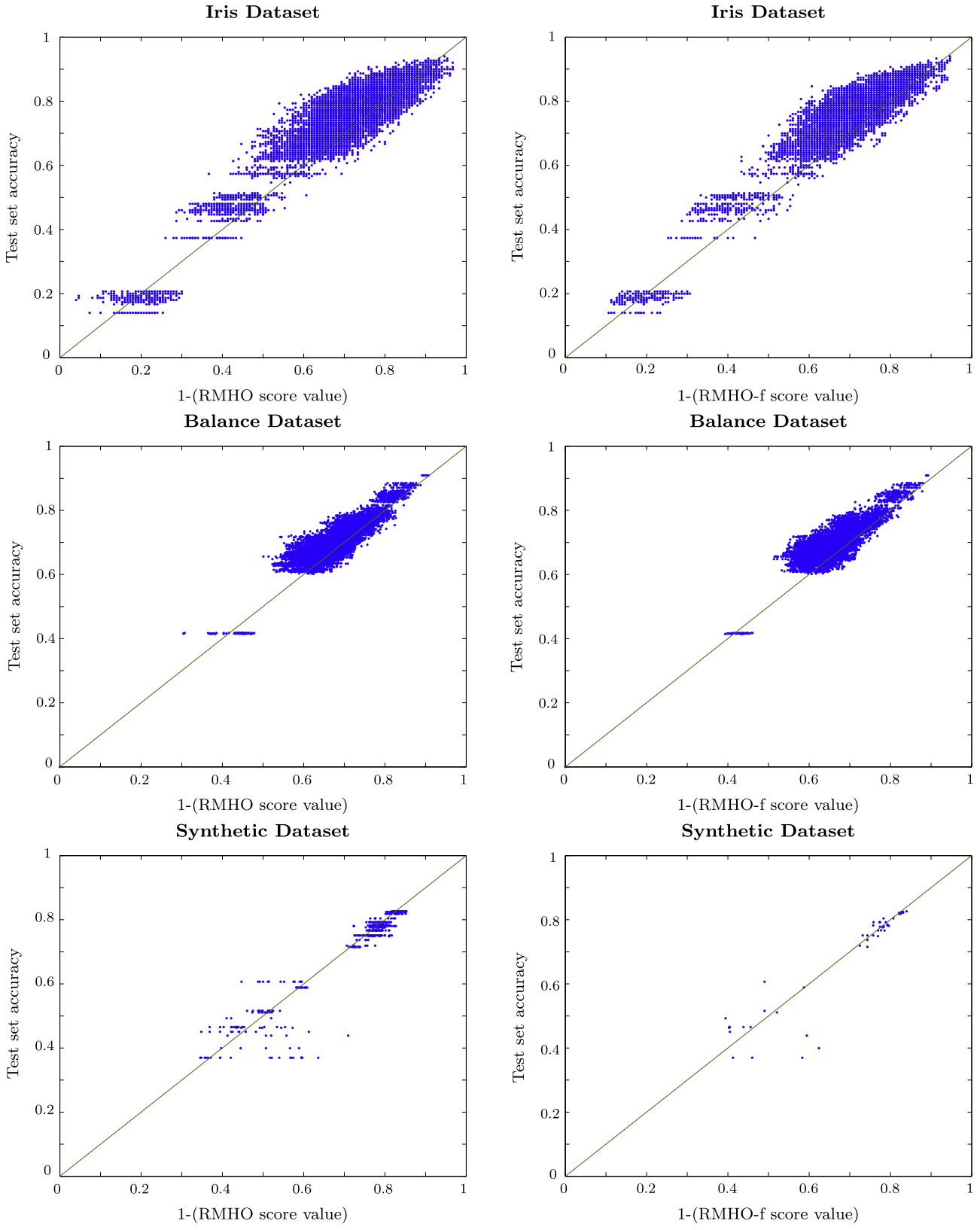


Fig. 5. Test set classification accuracies vs. RMHO/RMHO-f values for all possible DAGs as obtained using the Iris, Balance, and synthetic datasets. Each dot in the figure represents results for a DAG (dots may overlap).

upper-right corner of the graph for the test accuracy on the synthetic dataset of RMHO (Fig. 5). Therefore, it seems that RMHO is less suitable for choosing a best candidate within a group of DAGs. We attribute this inferiority of RMHO to the random selection of a validation set for each DAG. From examination of those highest ranked DAGs, values of the predicted accuracies using RMCV (83.4%), RMHO (85.3%), and RMHO-f (84.0%) exhibit a higher degree of over-fitting (compared with

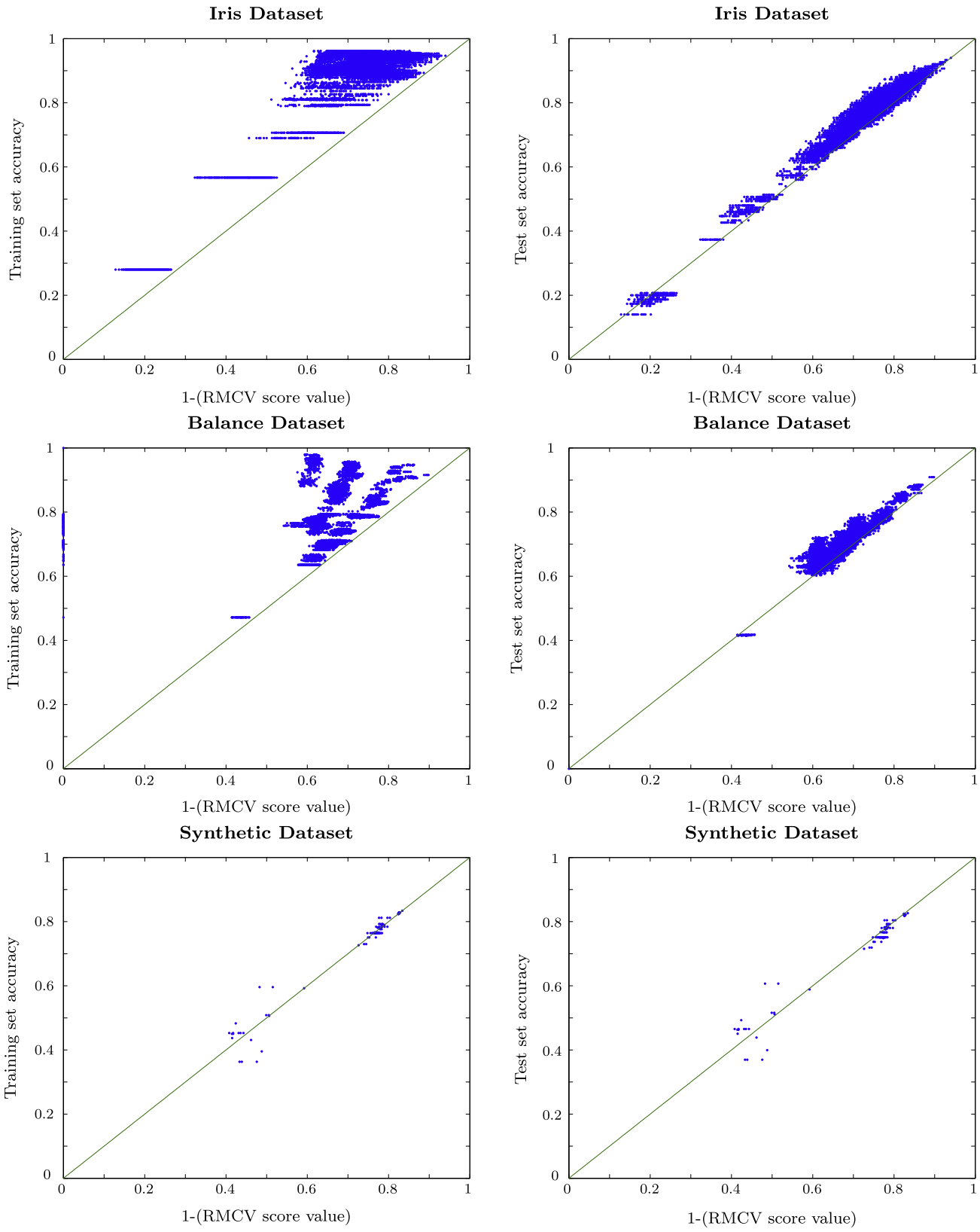


Fig. 6. Training/test set classification accuracies vs. RMCV values for all possible DAGs as obtained using the Iris, Balance, and synthetic datasets. Each dot in the figure represents results for a DAG (dots may overlap).

82.6%) of the two latter scores. Compared with RMCV and RMHO-f that scored the highest all of the 3766 DAGs with the highest test accuracy, K2, MDL, CMDL, and CLL scored the highest only 1, 1, 1, and 158, respectively, of the 3766 DAGS having the highest test accuracy. In addition, all of the DAGs that were ranked the highest by the latter scores belong to the above mentioned group of 3766 DAGs.

Table 5
MSE for test accuracy predictors.

Dataset	RMCV	RMHO	RMHO-f	Training
Iris	$0.15 \cdot 10^{-2}$	$0.37 \cdot 10^{-2}$	$0.52 \cdot 10^{-2}$	$2.46 \cdot 10^{-2}$
Balance	$8.98 \cdot 10^{-4}$	$0.12 \cdot 10^{-2}$	$0.19 \cdot 10^{-2}$	$15.1 \cdot 10^{-2}$
Synthetic	$1.56 \cdot 10^{-4}$	$3.87 \cdot 10^{-4}$	$3.88 \cdot 10^{-4}$	$0.57 \cdot 10^{-4}$

Table 6
Mean (std) of classification accuracies. Bayesian network initial structures and SVM kernel types appear in heading brackets. Bold/italic fonts are used, respectively, for the best/worst classifiers for a dataset.

Dataset	RMCV (NBC)	MDL (NBC)	MDL (empty)	K2 (NBC)	K2 (empty)	BNC-MDL	BNC-2P
Australian	86.14 (2.3)	86.26 (2.7)	85.72 (2.5)	84.33 (3.1)	83.96 (2.9)	84.77 (2.9)	*81.14* (3.3)
Breast	96.91 (1.4)	**97.08** (1.6)	96.29 (1.9)	96.71 (1.6)	95.40 (2.4)	91.83 (2.1)	*77.12* (7.3)
Chess	97.43 (0.8)	95.64 (1.2)	94.66 (1.2)	94.89 (1.2)	95.40 (1.1)	95.41 (1.2)	93.16 (1.2)
Cleve	81.53 (3.8)	80.27 (4.1)	81.20 (3.9)	79.93 (4.1)	80.47 (4.3)	80.70 (4.2)	79.63 (4.7)
Corral	99.57 (2.8)	99.71 (1.0)	99.86 (0.7)	99.43 (1.8)	**100.0** (0.0)	**100.0** (0.0)	98.57 (2.9)
Crx	85.71 (2.0)	86.21 (2.6)	86.17 (2.8)	84.21 (2.6)	84.26 (2.7)	85.62 (2.3)	81.80 (3.1)
Diabetes	74.03 (2.6)	73.41 (3.4)	73.94 (3.0)	73.54 (3.1)	73.95 (3.0)	75.60 (2.4)	71.09 (3.8)
Flare	84.03 (1.9)	84.07 (2.0)	84.09 (2.0)	83.95 (1.9)	84.00 (2.0)	**84.32** (1.9)	83.08 (1.9)
Glass	57.39 (6.4)	50.30 (6.8)	47.48 (8.3)	51.22 (6.1)	50.13 (7.3)	46.52 (7.8)	56.78 (7.9)
Glass2	75.03 (7.8)	70.97 (9.0)	69.31 (8.8)	71.26 (8.9)	72.29 (8.5)	69.31 (8.8)	70.86 (9.1)
Hayes	84.75 (5.1)	**85.69** (5.4)	**85.69** (5.4)	**85.69** (5.4)	81.56 (12.5)	**85.69** (5.4)	66.69 (8.4)
Heart	81.52 (4.7)	81.96 (5.8)	81.07 (5.5)	81.44 (5.2)	81.26 (5.5)	80.52 (4.9)	*59.56* (6.6)
Hepatitis	81.75 (9.6)	70.88 (13.2)	71.13 (14.4)	57.00 (15.7)	58.75 (14.6)	74.63 (10.5)	*46.50* (13.3)
Iris	93.73 (3.9)	*93.20* (4.4)	93.73 (3.7)	93.27 (4.1)	93.33 (3.7)	95.07 (3.4)	94.13 (4.0)
Letter	83.40 (0.5)	74.91 (0.6)	74.56 (0.6)	78.48 (1.5)	78.39 (1.5)	74.51 (0.6)	81.32 (0.5)
Lymphography	79.06 (6.6)	76.00 (7.4)	69.94 (7.6)	72.75 (7.0)	71.06 (7.7)	72.94 (6.8)	*61.50* (9.7)
Mofn	94.77 (5.8)	93.50 (2.4)	93.75 (1.9)	95.35 (5.7)	99.43 (2.0)	90.63 (4.4)	92.75 (1.8)
Pima	74.71 (3.0)	72.08 (2.9)	72.08 (2.9)	73.18 (2.9)	73.18 (2.9)	71.94 (3.2)	*53.50* (5.2)
Shuttle	99.49 (0.2)	97.99 (2.0)	*95.79* (2.1)	99.08 (0.3)	98.91 (0.3)	99.57 (0.2)	99.37 (0.3)
Tic-Tac-Toe	89.98 (5.9)	68.21 (2.8)	68.59 (3.1)	68.64 (2.8)	68.34 (3.0)	72.74 (3.6)	72.45 (3.0)
Vehicle	69.51 (3.1)	62.88 (4.3)	*60.25* (4.6)	67.31 (3.9)	64.92 (3.8)	61.92 (4.1)	70.99 (3.3)
Voting	94.71 (2.7)	92.83 (4.5)	94.67 (3.3)	*90.04* (5.5)	91.00 (4.7)	**96.83** (2.2)	96.71 (2.3)
Average	**84.78** (3.8)	81.55 (4.1)	80.91 (4.1)	80.99 (4.3)	80.91 (4.4)	81.41 (3.8)	*76.76* (4.7)

	TAN	NBC	CT	NN	SVM (linear)	SVM (polynomial)	SVM (Gaussian)
Australian	82.86 (3.1)	85.46 (2.3)	83.96 (2.9)	82.62 (3.8)	85.38 (2.8)	**86.35** (3.2)	85.51 (2.8)
Breast	89.65 (2.3)	96.71 (1.5)	95.01 (1.8)	94.75 (2.0)	96.99 (1.5)	95.68 (1.6)	97.05 (1.6)
Chess	92.55 (1.1)	*87.59* (1.1)	**99.33** (0.3)	99.31 (0.3)	94.77 (0.8)	92.56 (1.0)	93.84 (0.7)
Cleve	80.73 (4.0)	82.87 (3.6)	78.67 (4.4)	*76.67* (5.7)	83.50 (3.6)	83.07 (4.0)	**83.87** (3.7)
Corral	98.14 (2.9)	*87.00* (6.8)	91.07 (5.7)	99.93 (0.5)	87.14 (6.2)	91.79 (6.9)	96.36 (4.0)
Crx	83.89 (2.7)	85.61 (2.3)	83.97 (2.9)	*81.14* (3.8)	**86.38** (3.0)	86.12 (2.9)	**86.38** (3.0)
Diabetes	72.59 (3.3)	73.90 (2.8)	71.77 (2.8)	*66.58* (3.9)	75.59 (2.4)	74.60 (3.3)	**75.96** (2.7)
Flare	82.22 (2.0)	*77.74* (1.9)	82.37 (2.0)	84.25 (1.9)	84.29 (2.0)	**84.32** (1.9)	**84.32** (1.9)
Glass	56.65 (7.8)	57.61 (6.4)	**57.74** (8.4)	56.91 (8.0)	55.78 (6.8)	*31.04* (5.6)	50.00 (7.6)
Glass2	74.63 (8.1)	75.09 (7.7)	76.17 (8.0)	**76.40** (7.9)	76.00 (7.4)	*54.34* (10.2)	69.89 (11.9)
Hayes	72.13 (6.6)	83.50 (5.2)	82.75 (5.2)	76.38 (7.2)	51.56 (9.1)	*42.13* (5.5)	51.75 (8.8)
Heart	77.11 (5.0)	83.00 (4.6)	78.07 (5.6)	75.26 (5.6)	**84.07** (4.4)	82.48 (4.3)	84.04 (3.8)
Hepatitis	84.25 (10.2)	82.25 (9.2)	**84.63** (9.3)	82.00 (10.5)	83.38 (9.5)	83.75 (8.3)	83.75 (8.3)
Iris	94.20 (3.7)	*93.20* (4.0)	95.00 (3.5)	94.87 (3.6)	94.73 (3.3)	**95.20** (4.6)	94.60 (3.9)
Letter	**87.32** (0.5)	74.71 (0.6)	85.60 (0.6)	79.09 (1.9)	82.10 (0.5)	*53.73* (1.0)	74.00 (0.6)
Lymphography	77.19 (6.4)	78.56 (6.2)	75.38 (7.6)	80.81 (5.5)	**83.81** (5.5)	77.69 (6.7)	81.13 (7.9)
Mofn	93.51 (2.1)	*86.29* (2.8)	99.85 (0.6)	99.69 (0.6)	**100.0** (0.0)	**100.0** (0.0)	**100.0** (0.0)
Pima	71.51 (3.5)	74.35 (2.9)	72.01 (2.9)	70.13 (3.9)	**76.69** (2.5)	67.95 (4.5)	76.28 (2.7)
Shuttle	99.75 (0.1)	98.74 (0.3)	99.63 (0.1)	**99.81** (0.1)	98.54 (0.5)	97.05 (0.6)	99.24 (0.2)
Tic-Tac-Toe	76.23 (3.0)	69.96 (3.0)	85.30 (3.0)	**95.98** (3.5)	*65.42* (2.9)	65.44 (2.9)	67.31 (3.6)
Vehicle	**73.85** (3.6)	63.71 (3.5)	70.12 (3.3)	71.42 (3.6)	70.44 (2.6)	61.11 (4.4)	67.79 (3.5)
Voting	94.33 (3.8)	91.92 (3.6)	95.83 (2.8)	94.33 (3.1)	96.79 (2.3)	94.29 (3.3)	96.79 (2.2)
Average	82.51 (3.9)	81.35 (3.7)	83.83 (3.8)	83.56 (4.0)	82.43 (3.6)	77.30 (3.9)	81.81 (3.9)

5.3. Comparing RMCV to other classifiers

In the third experiment, the RMCV classifier was compared to other BNCs and to state of the art machine-learning classifiers as detailed in Table 4. Table 6 and Fig. 7 compare the RMCV classification accuracy with those of the other tested algorithms using the twenty-two UCI datasets. Table 6 shows that while RMCV is never the best performing classifier for a dataset, it is *always* reasonably good and never fails. On average, over the datasets, RMCV is the most accurate classifier. Inspection of standard deviations shows that our algorithm is also quite stable in comparison to the others. Each dot in Fig. 7 represents the average accuracies of two algorithms over the same dataset, while the cross size is determined according to

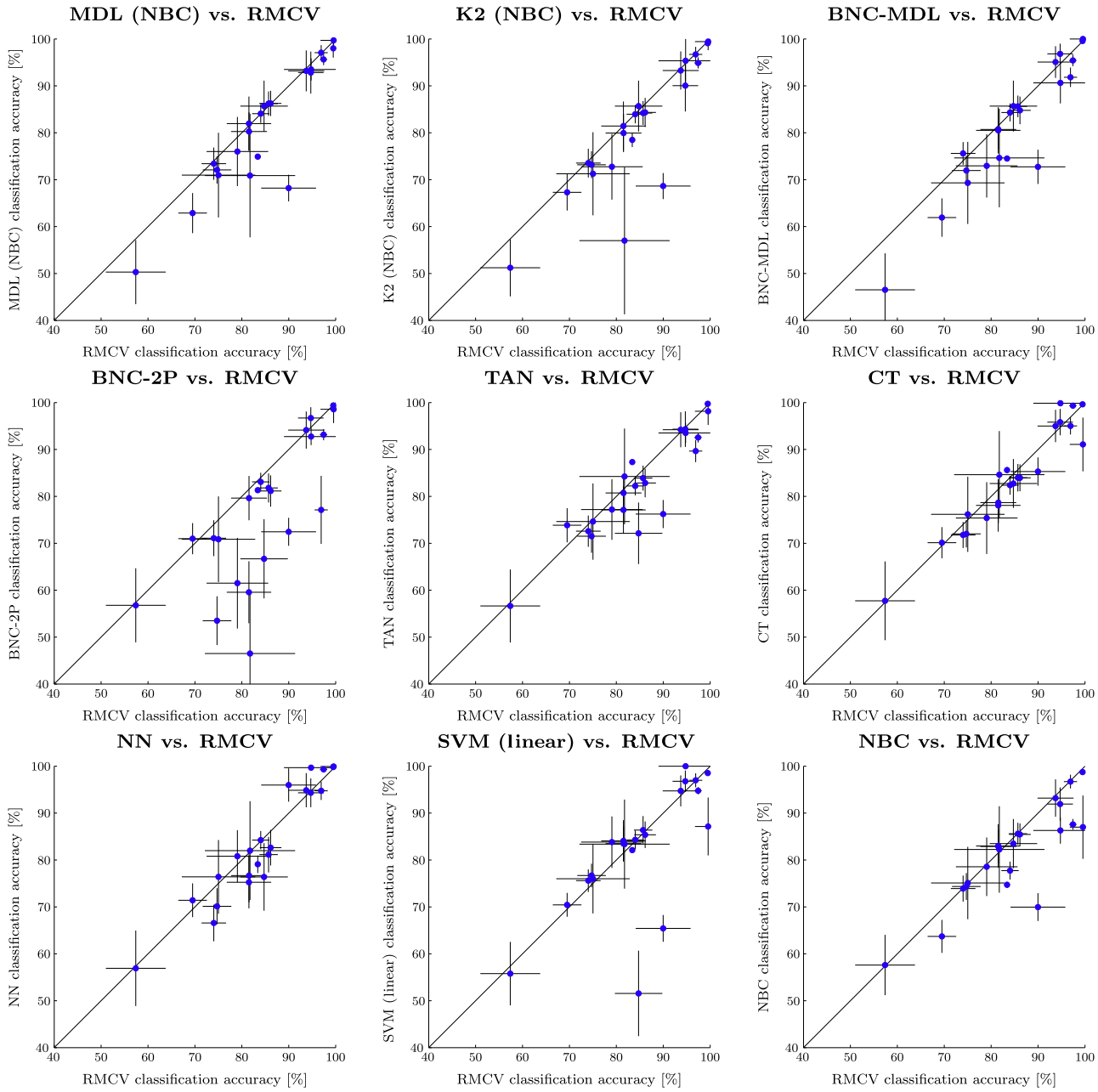


Fig. 7. Classification accuracies of RMCV compared to those of the competing classifiers. Charts are shown for the MDL and K2 (NBC) initializations and SVM (linear) kernel that were found most accurate on average in Table 6.

one standard deviation. Points below the $y = x$ line are in favor of the RMCV algorithm. Our algorithm generally outperforms every other BN algorithm. In those cases where it does not, it mostly ties with the competitor, or loses by a neck. Also, RMCV seldom falls short relative to CT, NN, or SVM.

Table 7 contains results for the corrected repeated k-fold cross validated paired t-test [38] for RMCV and competing algorithms. This test was found to have the highest replicability among common paired t-tests [38]. Each entry in the table results from a pairwise comparison between RMCV and a competing algorithm over a single dataset based on the classification accuracies measured using the 10xCV5 experiment. As can be seen in the table, RMCV significantly outperforms (with $p \leq 0.05$ or $p \leq 0.10$) the other algorithms frequently. In those cases where it does not, the differences are mostly non-significant. It should be noted, however, that final conclusions should be based on the Friedman test that compares all classifiers over all datasets simultaneously [32].

Table 8 shows results of the Friedman test for the classifier comparison in Table 6. The null hypothesis that accuracies of all BNCs are the same had been rejected with high confidence, and post-hoc tests were applied to validate which accuracies are different (Table 8(a)). RMCV is ranked highest according to the Friedman test. Based on the Bonferroni–Dunn post-hoc test with RMCV as the control classifier, RMCV is superior to all other BNCs with a significance level of $p < 0.05$, with the exception of BNC-MDL, for which $p < 0.1$.

Table 7
Significance values for RMCV classification accuracy compared with those of the competing algorithms using the corrected repeated k-fold cross validated paired t-test. Bayesian network initial structures and SVM kernel types appear in heading brackets. A minus sign (-), if present, indicates that RMCV is not favored over its competitor. Significance values of $p \leq 0.10$ appear in bold representing significant advantage of RMCV over the other classifier (unless (-) appears to indicate the opposite). Wins/losses are of RMCV compared to a competing algorithm with respect to all datasets.

RMCV (NBC) vs.	MDL (NBC)	MDL (empty)	K2 (NBC)	K2 (empty)	BNC-MDL	BNC-2P	TAN
Australian	-0.914	0.700	0.165	**0.095**	0.241	**0.002**	**0.009**
Breast	-0.577	0.523	0.581	0.216	**0.000**	**0.000**	**0.000**
Chess	**0.009**	**0.001**	**0.001**	**0.006**	**0.009**	**0.000**	**0.000**
Cleve	0.599	0.886	0.445	0.652	0.750	0.444	0.669
Corral	-0.932	-0.861	0.886	-0.787	-0.787	0.752	0.450
Crx	-0.661	-0.714	0.232	0.261	0.944	**0.012**	0.162
Diabetes	0.731	0.958	0.728	0.962	-0.201	0.183	0.356
Flare	-0.881	-0.839	0.779	0.918	-0.268	**0.031**	**0.003**
Glass	**0.044**	**0.031**	**0.037**	**0.013**	**0.014**	0.878	0.867
Glass2	0.158	**0.078**	0.252	**0.084**	**0.078**	**0.039**	0.836
Hayes	-0.453	-0.453	-0.453	0.587	-0.453	**0.000**	**0.001**
Heart	-0.841	0.831	0.975	0.903	0.626	**0.000**	0.124
Hepatitis	**0.056**	**0.070**	**0.002**	**0.003**	0.108	**0.000**	-0.698
Iris	0.689	1.000	0.702	0.739	-0.410	-0.774	-0.740
Letter	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**-0.000**
Lymphography	0.361	**0.027**	**0.071**	**0.017**	**0.096**	**0.001**	0.672
Mofn	0.673	0.733	-0.871	-0.193	0.154	0.490	0.676
Pima	**0.093**	**0.093**	0.297	0.297	0.117	**0.000**	0.106
Shuttle	0.149	**0.001**	**0.001**	**0.000**	-0.411	0.255	**-0.022**
Tic-Tac-Toe	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
Vehicle	**0.004**	**0.000**	0.260	**0.025**	**0.000**	-0.412	**-0.036**
Voting	0.263	0.975	**0.066**	**0.077**	**-0.065**	-0.118	0.784
Wins ($p \leq 0.05$)	22.73%	31.82%	27.27%	36.36%	27.27%	59.09%	27.27%
Losses ($p \leq 0.05$)	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	13.64%
Wins ($p \leq 0.10$)	31.82%	45.45%	36.36%	50.00%	36.36%	59.09%	27.27%
Losses ($p \leq 0.10$)	0.00%	0.00%	0.00%	0.00%	4.55%	0.00%	13.64%

	NBC	CT	NN	SVM (linear)	SVM (polynomial)	SVM (Gaussian)
Australian	0.409	**0.081**	**0.064**	0.562	-0.855	0.645
Breast	0.581	**0.063**	**0.036**	-0.899	0.173	-0.842
Chess	**0.000**	**-0.000**	**-0.000**	**0.000**	**0.000**	**0.000**
Cleve	-0.389	0.243	0.107	-0.284	-0.446	-0.210
Corral	**0.002**	**0.013**	-0.824	**0.000**	**0.053**	0.223
Crx	0.888	0.212	**0.009**	-0.633	-0.760	-0.633
Diabetes	0.917	0.155	**0.000**	-0.275	-0.743	-0.207
Flare	**0.000**	**0.016**	-0.433	-0.269	-0.268	-0.268
Glass	-0.938	-0.935	0.895	0.674	**0.000**	**0.083**
Glass2	-0.877	-0.458	-0.490	-0.626	**0.000**	0.454
Hayes	0.450	0.387	**0.020**	**0.000**	**0.000**	**0.000**
Heart	-0.490	0.244	**0.045**	-0.282	-0.652	-0.273
Hepatitis	-0.853	-0.601	-0.969	-0.790	-0.714	-0.714
Iris	0.689	-0.432	-0.489	-0.528	-0.526	-0.508
Letter	**0.000**	**-0.000**	**0.000**	**0.000**	**0.000**	**0.000**
Lymphography	0.855	0.439	-0.653	-0.246	0.729	-0.679
Mofn	**0.002**	-0.132	-0.137	-0.119	-0.119	-0.119
Pima	0.708	0.146	**0.013**	-0.114	**0.011**	-0.267
Shuttle	**0.000**	-0.245	**-0.018**	**0.000**	**0.000**	**0.030**
Tic-Tac-Toe	**0.000**	0.154	-0.103	**0.000**	**0.000**	**0.000**
Vehicle	**0.001**	-0.711	-0.376	-0.601	**0.000**	0.303
Voting	**0.068**	-0.338	0.809	**-0.025**	0.765	**-0.053**
Wins ($p \leq 0.05$)	36.36%	9.09%	31.82%	27.27%	40.91%	22.73%
Losses ($p \leq 0.05$)	0.00%	9.09%	9.09%	4.55%	0.00%	0.00%
Wins ($p \leq 0.10$)	40.91%	18.18%	36.36%	27.27%	45.45%	27.27%
Losses ($p \leq 0.10$)	0.00%	9.09%	9.09%	4.55%	0.00%	4.55%

As expected [4–6], MDL (Empty) is ranked relatively low. However, MDL (NBC) is ranked relatively high, demonstrating the influence of the initial structure on accuracy. BNC-2P is ranked the lowest. We suspect that this is because BNC-2P, which circumvents over-fitting by limiting the number of possible parents of each node to two, misses the true (data faithful) underlying network in some of the datasets. Some of our results for BNC-2P and most of our results for TAN and NBC are consistent with [6]. Factors to the disparity between the studies may be the different test methodologies and dataset

Table 8
Statistical significance tests for the accuracies in Table 6 for (a) the BNCs and (b) all classifiers.

Algorithm	Average rank	Post-hoc tests significance values Bonferroni–Dunn	
(a) The Friedman test for BN classifiers			
RMCV	2.591		
BNC-MDL	4.659	0.0981	
MDL (NBC)	4.932	0.0367	
NBC	5.000	0.0282	
TAN	5.091	0.0197	
MDL (empty)	5.341	0.0069	
K2 (empty)	5.364	0.0063	
K2 (NBC)	5.386	0.0057	
BNC-2P	6.636	<0.0001	
The Friedman test statistics			
N	22	Chi-square	26.424
Degrees of freedom	8	Significance value	<0.001

		Post-hoc tests significance values	
		Bonferroni–Dunn	Wilcoxon
(b) The Friedman test for all classifiers			
RMCV	5.136		
SVM (linear)	5.205	Non-significant	0.610
SVM (Gaussian)	5.727	Non-significant	0.570
CT	6.682	Non-significant	0.147
NN	6.932	Non-significant	0.276
BNC-MDL	7.341	Non-significant	0.006
MDL (NBC)	7.841	Non-significant	0.001
TAN	8.068	Non-significant	0.013
NBC	8.182	Non-significant	0.004
SVM (polynomial)	8.205	Non-significant	0.033
MDL (empty)	8.523	0.0944	0.001
K2 (NBC)	8.523	0.0944	<0.001
K2 (empty)	8.545	0.0894	<0.001
BNC-2P	10.091	0.0011	<0.001
The Friedman test statistics			
N	22	Chi-square	33.081
Degrees of freedom	13	Significance value	0.002

quantizations. Nevertheless, in most cases where CLL values in both studies were similar the corresponding classification accuracies were similar too.

Table 8(b) contains results of the Friedman test for all of the classifiers. RMCV is ranked highest according to the Friedman test. Again, the null hypothesis that all algorithms are the same had been rejected with high confidence and post-hoc tests were applied to validate which algorithms are different. Due to the large number of models compared (fourteen), a relatively large difference of average ranks is required by the Bonferroni–Dunn test to indicate a significant difference, and RMCV is found to be significantly superior (with $p < 0.1$) only to four other classifiers. Two pairs of classifiers (i.e., NBC and TAN as well as K2 (Empty) and K2 (NBC)) swap places between Table 8(a) and (b). This is simply because each of the tables represents a somewhat independent Friedman test using different candidates, except of the two classifiers in the pair. Such a setting may yield different rankings. Besides, there is almost no difference between the rankings of the classifiers in a pair between the two tables. The less conservative Wilcoxon signed-rank test finds RMCV to be superior (with $p < 0.05$) to all of the evaluated BNCs and also to SVM (polynomial). For CT, NN, and SVM (Gaussian/linear), RMCV is not significantly superior with $p \in [0.147, 0.610]$. This means that RMCV, CT, NN, and SVM (Gaussian/linear) are comparable in terms of classification accuracy, which may be considered an achievement for RMCV.

Analyses of the receiver operating characteristic (ROC) curve and the area under the ROC curve (AUC) are commonly used to evaluate binary classifiers. Table 9 presents mean AUC values for all BNCs evaluated using all two-class datasets. On average, RMCV outperforms all BNCs. The Friedman test ranked RMCV highest. However, the null hypothesis had not been rejected, and so the Bonferroni–Dunn post-hoc test was not performed. The Wilcoxon signed-rank test found RMCV in some favor over all other BNCs, but not significantly (with the exception of BNC-2P, over which RMCV was found significantly better with $p < 0.05$). Together with the results of the corrected repeated k-fold cross validated paired t-test (Table 10), we can conclude that in terms of AUC, RMCV seldom falls short of other BNCs and is frequently better.

5.4. RMCV complexity and run-time

Following the evaluation of RMCV classification accuracy, the complexity and run-time of RMCV are examined. The maximal number of iterations of the HC search is needed when the initial structure performs the most poorly, i.e., errs for all N instances, and then is improved by classifying correctly an additional single instance at each iteration, until a perfect

Table 9
Mean AUC values for all BNCs and all two-class datasets from Table 1. Bayesian network initial structures appear in heading brackets. Bold/italic fonts are used for the best/worst classifiers for a dataset.

Dataset	RMCV (NBC)	MDL (NBC)	MDL (empty)	K2 (NBC)	K2 (empty)	BNC-MDL	BNC-2P	TAN	NBC
Australian	0.9203	0.9238	**0.9253**	0.9201	0.9223	0.9085	*0.8692*	0.8835	0.9168
Breast	**0.9807**	0.9797	0.9793	0.9773	0.9763	0.9747	0.9669	*0.9642*	0.9773
Chess	**0.9952**	0.9907	0.9882	0.9906	0.9916	0.9903	0.9781	0.9819	*0.9518*
Cleve	0.8954	0.8881	0.8869	0.8866	0.8801	0.8738	*0.8669*	0.8848	**0.9118**
Corral	0.9981	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	*0.9381*
Crx	0.9208	**0.9338**	0.9312	0.9313	0.9322	0.9197	*0.8814*	0.9005	0.9163
Diabetes	0.7981	0.7891	0.7859	0.7877	0.7917	**0.8056**	0.7862	*0.7790*	0.8048
Glass2	0.7943	0.7280	0.7111	0.7777	0.7934	*0.6993*	0.7892	0.7811	**0.8006**
Heart	0.8727	0.9005	0.8916	0.8901	0.8877	0.8887	*0.8018*	0.8467	**0.9042**
Hepatitis	0.8163	0.6911	0.7118	0.7033	*0.6859*	0.7452	0.7314	**0.8731**	0.8283
Mofn	*0.9840*	0.9995	0.9988	0.9997	**1.0000**	0.9997	0.9993	0.9998	**1.0000**
Pima	**0.8209**	0.7821	0.7821	0.7977	0.7979	0.7781	*0.6993*	0.7330	0.8092
Tic-Tac-Toe	**0.9580**	0.7078	0.7016	0.7054	*0.7011*	0.7859	0.7695	0.8202	0.7469
Voting	0.9671	0.9685	0.9804	0.9707	0.9749	**0.9905**	0.9869	0.9877	*0.9658*
Average	**0.9087**	0.8773	0.8767	0.8813	0.8811	0.8828	*0.8662*	0.8883	0.8908

Table 10
Significance values for RMCV AUC values compared with those of the competing algorithms using the corrected repeated k-fold cross validated paired t-test. A minus sign (-), if present, indicates that RMCV is not favored over its competitor. Significance values of $p \leq 0.10$ appear in bold representing significant advantage of RMCV over the other classifier (unless (-) appears to indicate the opposite). Wins/Losses are of RMCV compared to a competing algorithm with respect to all datasets.

RMCV (NBC) vs.	MDL (NBC)	MDL (empty)	K2 (NBC)	K2 (empty)	BNC-MDL	BNC-2P	TAN	NBC
Australian	-0.696	-0.594	0.985	-0.840	0.334	**0.002**	**0.010**	0.556
Breast	0.745	0.645	0.290	0.213	0.410	0.118	**0.009**	0.290
Chess	**0.083**	**0.015**	**0.076**	**0.073**	**0.032**	**0.016**	**0.000**	**0.000**
Cleve	0.716	0.601	0.646	0.425	0.183	0.144	0.518	-0.290
Corral	-0.787	-0.787	-0.787	-0.787	-0.787	-0.787	-0.787	**0.010**
Crx	-0.178	-0.278	-0.227	-0.200	0.926	**0.029**	0.139	0.496
Diabetes	0.578	0.451	0.538	0.675	-0.593	0.494	0.208	-0.525
Glass2	0.107	**0.028**	0.466	0.951	**0.076**	0.755	0.447	-0.469
Heart	-0.252	-0.460	-0.466	-0.536	-0.533	**0.041**	0.360	-0.189
Hepatitis	0.372	0.388	0.355	0.250	0.444	0.372	-0.554	-0.798
Mofn	-0.215	-0.226	-0.208	-0.196	-0.207	-0.215	-0.201	-0.196
Pima	**0.012**	**0.012**	0.120	0.121	**0.010**	**0.000**	**0.000**	0.183
Tic-Tac-Toe	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
Voting	-0.922	-0.366	-0.822	-0.584	**-0.100**	-0.129	-0.115	0.923
Wins ($p \leq 0.05$)	14.29%	28.57%	7.14%	7.14%	21.43%	42.86%	35.71%	21.43%
Losses ($p \leq 0.05$)	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Wins ($p \leq 0.10$)	21.43%	28.57%	14.29%	14.29%	35.71%	42.86%	35.71%	21.43%
Losses ($p \leq 0.10$)	0.00%	0.00%	0.00%	0.00%	7.14%	0.00%	0.00%	0.00%

RMCV score of 0. In the worst case, the HC search needs to consider $n(n-1)$ edge additions in each iteration (if starting from the empty graph) or $\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$ edge deletions or reversals. All are of complexity of $O(n^2)$. Therefore, this worst case scenario imposes an upper bound on RMCV complexity of $O(N \cdot n^2)$.

However, when using a discriminative score such as RMCV, changes to the current DAG, which do not affect the Markov blanket of the class variable, have no effect on the score value and therefore can be avoided. For a general graph, it is difficult to quantize the reduction in complexity due to this factor, but it is prominent when starting the search from the empty graph and only moderate in the first steps when starting the search from NBC. Our experiments (see below) show that avoiding changes that do not affect the Markov blanket of the class variable contributes to reduction in the run-time.

As previously mentioned (Section 4), the RMCV score (as other discriminative scores) is not decomposable. However, the combination of ML parameter estimation together with the suggested HC search enables a speed-up by a factor of approximately n in the run-time. Parameter learning of a neighboring DAG to the current DAG does not require a complete re-assessment of all parameters. When a neighbor DAG differs from the current DAG by a single edge following deletion or addition, only one of the DAG nodes has a different set of parameters; whereas when it differs by a single edge reversal, only two of the DAG nodes have different sets. Following the factorization of the joint probability distribution (1), we can store in memory and use for each instance and for each possible class only the few parameters needed for updating the calculation of the RMCV score based on the relevant node(s) and their parents. Since the evaluations apply to only one or two of the n nodes, a speed-up of the run-time by a factor of approximately n (for edge addition/deletion) or $n/2$ (for edge reversal) is obtained by the suggested optimization. Since parameter learning is not in the scope of this work, we refrained from further optimization of this stage.

Table 11
Mean (std) of run-times (in seconds) for RMCV and the three RMHO variants. Bayesian network initial structures appear in heading brackets. Note that two averages are presented: with and without the Letter dataset (see text).

Dataset	RMCV (NBC)	RMHO (NBC)	RMHO-f (NBC)	RMHO-ar (NBC)
Australian	9946.50 (2712.83)	1027.34 (389.96)	2136.03 (669.91)	1676.68 (609.69)
Breast	3142.34 (788.27)	510.21 (204.24)	539.31 (227.95)	496.24 (216.11)
Chess	5444149.77 (541897.27)	164791.69 (97797.66)	763328.45 (286418.61)	712681.75 (294780.70)
Cleve	1351.55 (537.61)	206.68 (62.13)	249.26 (103.73)	230.58 (95.52)
Corral	83.74 (21.66)	18.39 (5.80)	19.12 (5.70)	18.78 (5.96)
Crx	25249.94 (6533.89)	3322.63 (1044.42)	8833.84 (2943.41)	7364.66 (2570.07)
Diabetes	1512.45 (556.71)	221.67 (54.87)	350.10 (115.56)	316.63 (102.80)
Flare	18637.47 (2344.38)	2275.29 (972.84)	4041.15 (1019.66)	3766.32 (939.60)
Glass	361.25 (133.70)	74.29 (33.11)	79.27 (30.74)	72.65 (27.83)
Glass2	187.70 (61.59)	47.30 (25.64)	38.44 (13.19)	35.09 (11.85)
Hayes	20.04 (6.12)	7.04 (3.52)	6.68 (2.46)	6.33 (2.16)
Heart	2079.64 (736.39)	293.42 (89.93)	392.93 (140.38)	367.48 (129.01)
Hepatitis	4795.11 (1768.47)	612.37 (287.52)	587.18 (274.67)	564.68 (274.11)
Iris	33.34 (7.77)	6.21 (2.58)	6.78 (2.71)	6.04 (2.36)
Letter	805094.28 (87776.07)	97872.33 (27552.36)	197105.51 (43405.19)	186340.48 (40561.34)
Lymphography	8821.40 (2164.97)	853.61 (208.21)	1476.11 (531.27)	1103.27 (498.97)
Mofn	16589.88 (9063.84)	1789.13 (591.24)	2658.96 (1449.57)	2396.50 (1338.59)
Pima	1123.98 (338.79)	218.79 (75.88)	261.60 (93.96)	240.13 (88.57)
Shuttle	11349.08 (3650.13)	1557.95 (355.56)	1931.30 (650.92)	1760.68 (604.87)
Tic-Tac-Toe	7050.81 (2132.08)	511.33 (248.36)	1246.93 (579.40)	1131.68 (456.79)
Vehicle	111961.08 (23528.03)	7681.05 (2769.41)	21378.49 (4439.96)	20142.26 (4205.08)
Voting	7615.42 (2208.35)	1099.31 (325.97)	1066.50 (359.34)	1002.31 (331.59)
Average	294598.04 (31316.77)	12954.45 (6050.05)	45806.09 (15612.65)	42805.51 (15811.53)
Excluding Letter	270288.69 (28628.23)	8910.75 (5026.14)	38601.35 (14289.20)	35970.51 (14632.96)
	RMCV (empty)	RMHO (empty)	RMHO-f (empty)	RMHO-ar (empty)
Australian	4895.87 (4382.60)	796.11 (336.42)	1027.80 (794.76)	855.83 (602.36)
Breast	3683.64 (707.21)	566.28 (172.33)	651.87 (126.98)	647.28 (126.87)
Chess	408920.01 (110659.53)	91445.65 (17168.82)	102272.67 (17067.51)	103254.38 (21616.94)
Cleve	1266.94 (650.58)	142.25 (48.34)	192.78 (99.35)	190.92 (99.07)
Corral	71.01 (45.61)	14.67 (6.86)	18.49 (10.87)	17.80 (9.31)
Crx	11016.94 (6112.15)	1785.69 (772.13)	2255.61 (1800.92)	2112.53 (1633.03)
Diabetes	1294.88 (349.32)	208.44 (92.89)	329.51 (120.99)	320.35 (115.84)
Flare	2175.73 (140.09)	556.49 (27.91)	573.60 (31.34)	579.24 (38.24)
Glass	542.21 (84.61)	77.52 (25.49)	107.48 (25.90)	106.00 (25.50)
Glass2	233.02 (31.18)	59.01 (16.07)	59.10 (7.90)	58.78 (7.61)
Hayes	42.16 (9.53)	8.29 (3.41)	12.44 (3.86)	11.98 (2.90)
Heart	1672.84 (830.14)	206.94 (79.34)	331.57 (137.72)	328.79 (132.84)
Hepatitis	2659.63 (832.38)	455.62 (189.27)	504.08 (215.38)	507.25 (207.02)
Iris	18.49 (3.20)	5.68 (1.38)	5.96 (1.07)	5.92 (0.96)
Letter	922519.85 (89425.91)	129311.40 (28266.37)	215100.29 (39077.66)	207979.07 (35388.85)
Lymphography	5144.18 (1410.12)	688.84 (263.46)	940.22 (245.92)	965.24 (262.35)
Mofn	2037.54 (105.37)	528.01 (28.48)	549.99 (24.16)	550.66 (24.20)
Pima	1306.10 (325.23)	207.51 (75.37)	325.64 (110.92)	317.40 (104.20)
Shuttle	6373.11 (1072.70)	1820.50 (416.51)	1666.66 (304.19)	1680.92 (296.05)
Tic-Tac-Toe	3521.21 (2827.62)	276.23 (125.56)	639.12 (599.37)	625.99 (602.88)
Vehicle	68690.84 (15797.94)	5933.07 (1866.99)	14382.61 (8226.40)	14392.50 (8300.24)
Voting	2803.47 (262.56)	889.74 (204.90)	732.24 (56.97)	724.98 (88.34)
Average	65949.53 (10730.25)	10726.54 (2281.29)	15576.35 (3140.46)	15283.36 (3167.53)
Excluding Letter	25160.47 (6982.84)	5079.65 (1043.90)	6075.21 (1429.16)	6107.37 (1633.18)

Table 11 shows the run-times (in seconds) for the experiments of Section 5.1. The run-times were obtained using a straightforward implementation of the algorithms, as presented in Sections 4 and 5.1, without the above special optimizations. It is clear from the table that all RMHO variants have comparable run-times. RMHO is usually faster than RMCV, though this virtue is diminished considering the inferior accuracy of the former (Table 2). RMHO-f and RMHO-ar have quite similar run-times when starting from the empty graph. This is clear because RMHO-ar is RMHO-f that avoids edge reversal, and the initial empty graph and its neighbor graphs have few edges, if any, to reverse. When starting from NBC, RMHO-ar has an average advantage of seven percent over RMHO-f. Since RMHO-ar and RMHO-f have equal classification accuracies (Table 2), RMHO-ar may be considered a viable alternative to RMHO-f, although the gain is minimal. RMCV usually takes about four and six times longer to complete for the empty graph and NBC, respectively, compared to RMHO-f. This is expected, since the RMCV score with $K = 4$ should take four (K) times longer to compute than the RMHO score. While this is the main contributing factor for the differences, we should remember that the search paths are likely to be different, varying the run-times even further.

Tables 12 and 13 present, respectively, the run-times (in seconds) for the experiments comparing RMCV to BN and non-BN classifiers in Section 5.3. We excluded the (structure) learning-free NBC. RMCV (NBC), BNC-MDL, and BNC-2P all

Table 12
Mean (std) of run-times (in seconds) for learning BNCs. Bayesian network initial structures appear in heading brackets. Note that two averages are presented: with and without the Letter dataset (see text).

Dataset	RMCV (NBC)	RMCV (NBC) optimized	RMCV (NBC) speedup	MDL (NBC)	MDL (empty)
Australian	9946.50 (2712.83)	63.01 (13.36)	x157.84	389.91 (47.32)	330.09 (71.98)
Breast	3142.34 (788.27)	25.86 (7.74)	x121.49	24.99 (4.29)	52.28 (6.92)
Chess	5444150 (541897)	41152 (7548)	x132.29	245292 (32997)	157168 (15679)
Cleve	1351.55 (537.61)	9.63 (3.37)	x140.28	89.03 (17.26)	87.34 (9.84)
Corral	83.74 (21.66)	1.09 (0.20)	x76.75	7.60 (1.26)	8.59 (1.37)
Crx	25249.94 (6533.89)	70.54 (23.14)	x357.94	493.81 (57.61)	386.42 (78.53)
Diabetes	1512.45 (556.71)	18.96 (6.04)	x79.78	47.04 (6.14)	35.63 (4.91)
Flare	18637.47 (2344.38)	886.89 (182.15)	x21.01	302.03 (31.31)	149.78 (25.86)
Glass	361.25 (133.70)	10.49 (3.11)	x34.43	37.21 (5.66)	25.69 (4.80)
Glass2	187.70 (61.59)	2.08 (0.56)	x90.12	43.80 (5.60)	18.87 (3.68)
Hayes	20.04 (6.12)	0.61 (0.23)	x33.09	1.93 (0.25)	2.19 (0.21)
Heart	2079.64 (736.39)	13.58 (4.89)	x153.19	103.36 (12.89)	112.88 (20.50)
Hepatitis	4795.11 (1768.47)	13.87 (2.72)	x345.61	722.92 (118.08)	402.04 (73.05)
Iris	33.34 (7.77)	0.65 (0.11)	x51.63	1.30 (0.10)	2.13 (0.19)
Letter	805094 (87776)	1601337 (142266)	x0.50	11580 (2129)	16048 (3459)
Lymphography	8821.40 (2164.97)	71.61 (16.52)	x123.19	678.62 (106.24)	423.85 (59.26)
Mofn	16589.88 (9063.84)	238.27 (153.79)	x69.63	117.31 (19.30)	107.60 (22.66)
Pima	1123.98 (338.79)	13.05 (3.43)	x86.13	37.93 (5.93)	14.73 (2.91)
Shuttle	11349.08 (3650.13)	3643.50 (874.28)	x3.11	459.42 (77.70)	265.45 (42.00)
Tic-Tac-Toe	7050.81 (2132.08)	219.32 (50.79)	x32.15	129.70 (18.89)	73.28 (16.33)
Vehicle	111961.08 (23528.03)	1506.07 (286.77)	x74.34	1576.41 (199.83)	1066.57 (147.96)
Voting	7615.42 (2208.35)	36.46 (8.02)	x208.89	525.80 (76.20)	459.66 (70.22)
Average	294598.04 (31316.77)	74969.74 (6884.30)	x108.79	11939.18 (1633.53)	8056.42 (900.08)
Excluding Letter	270288.69 (28628.23)	2285.60 (437.56)	x113.95	11956.27 (1609.93)	7675.85 (778.21)

	K2 (NBC)	K2 (empty)	BNC-MDL	BNC-2P	TAN
Australian	303.31 (45.64)	237.19 (51.47)	10668.87 (2003.44)	29140.34 (3597.51)	0.283 (0.066)
Breast	15.58 (3.13)	44.87 (10.79)	1485.20 (89.11)	7196.08 (2115.00)	0.289 (0.054)
Chess	236275 (43496)	159317 (34397)	1521928 (84560)	2601383 (186435)	2.472 (1.003)
Cleve	67.74 (12.49)	73.28 (13.16)	812.91 (126.17)	2070.97 (279.92)	0.307 (0.062)
Corral	6.13 (0.86)	6.95 (1.38)	408.24 (49.37)	409.64 (44.59)	0.073 (0.022)
Crx	377.43 (55.03)	306.94 (70.27)	25861.71 (6118.18)	57251.00 (10724.86)	0.462 (0.171)
Diabetes	31.96 (6.93)	27.44 (9.00)	753.95 (95.47)	1615.35 (436.58)	0.122 (0.030)
Flare	195.79 (24.75)	98.98 (21.99)	3485.26 (993.18)	20304.80 (4036.85)	0.735 (0.361)
Glass	38.47 (5.94)	29.23 (6.66)	190.46 (154.20)	777.05 (105.18)	0.203 (0.009)
Glass2	48.85 (5.64)	30.64 (7.05)	333.53 (215.07)	428.71 (123.25)	0.078 (0.005)
Hayes	1.64 (0.18)	1.95 (0.28)	83.39 (7.03)	129.56 (15.21)	0.031 (0.000)
Heart	95.68 (12.81)	84.38 (20.21)	1109.96 (218.01)	3436.49 (391.74)	0.252 (0.089)
Hepatitis	571.39 (167.86)	424.22 (123.69)	2425.09 (872.23)	17343.70 (2217.24)	0.355 (0.009)
Iris	1.82 (0.55)	2.35 (0.48)	29.92 (0.71)	79.40 (16.79)	0.020 (0.007)
Letter	11473 (2920)	10031 (3442)	757483 (23920)	980736 (22116)	17.602 (7.591)
Lymphography	436.14 (135.75)	425.01 (105.11)	3267.80 (1083.41)	18197.64 (1669.18)	0.834 (0.582)
Mofn	84.76 (24.71)	69.41 (17.92)	19276.07 (6522.23)	31542.38 (10030.43)	0.183 (0.081)
Pima	27.48 (3.58)	13.28 (7.67)	358.82 (20.32)	1767.86 (280.48)	0.336 (0.101)
Shuttle	242.61 (47.86)	243.69 (39.19)	10977.94 (820.25)	40287.55 (4037.44)	0.386 (0.143)
Tic-Tac-Toe	64.82 (20.88)	41.15 (10.73)	7111.94 (746.97)	11472.25 (1937.70)	0.144 (0.049)
Vehicle	977.40 (261.28)	923.64 (211.46)	29935.43 (3521.87)	85936.72 (11026.17)	1.145 (0.471)
Voting	428.39 (113.26)	380.87 (94.11)	4245.16 (730.24)	11049.81 (2739.72)	0.220 (0.006)
Average	11443.88 (2152.96)	7855.17 (1757.32)	109192.40 (6039.44)	178298.02 (12017.10)	1.206 (0.496)
Excluding Letter	11442.51 (2116.41)	7751.54 (1677.11)	78321.41 (5187.97)	140086.70 (11536.21)	0.425 (0.158)

use discriminative scores and have run-times of the same orders, which are longer than those of the classifiers that use generative scores or TAN (Table 12). While RMCV (NBC) may train $O(K \cdot n^2)$ BNCs within each HC step, BNC-MDL may train $O(n^2)$ BNCs. Yet, each RMCV classifier uses only $(K-1)/K$ of the instances for training, whereas BNC-MDL needs them all. Note however that both scores are evaluated using the same number of instances (N). Table 12 also shows that although BNC-2P defines a smaller neighborhood than BNC-MDL, BNC-2P run-time is longer than that of BNC-MDL. This is because BNC-2P lacks a regularization term in the CLL score that can penalize complex graphs.

The generative score-based BNCs in Table 12, MDL and K2, have similar run-times depending on the initial structure. This is reasonable, as they all use the same HC search together with a decomposable score. Decomposition gives the generative models a clear advantage over the discriminative models with respect to run-time. The optimized version of RMCV (second column of Table 12) was implemented according to the speed-up presented above. This speed-up contributed to run-time reduction of approximately n. Avoidance of changes not affecting the Markov blanket of the class variable and further improvements to the implementation added another reduction in the run-time in an order of magnitude, as reflected in

Table 13
Mean (std) of run-times (in seconds) for RMCV and non-BN classifiers. Bayesian network initial structures and SVM kernel types appear in heading brackets. Note that two averages are presented: with and without the Letter dataset (see text).

Dataset	RMCV (NBC) optimized	CT	NN	SVM (linear)	SVM (polynomial)	SVM (Gaussian)
Australian	63.01 (13.36)	0.133 (0.0065)	78.81 (15.14)	0.0362 (0.0087)	0.0213 (0.0074)	0.0200 (0.0072)
Breast	25.86 (7.74)	0.054 (0.0485)	12.70 (2.83)	0.0025 (0.0057)	0.0040 (0.0069)	0.0040 (0.0069)
Chess	41152 (7548)	0.282 (0.0141)	4523.96 (3316.11)	0.2508 (0.1181)	0.8164 (0.3556)	0.4824 (0.0113)
Cleve	9.63 (3.37)	0.058 (0.0041)	16.10 (3.82)	0.0093 (0.0077)	0.0075 (0.0085)	0.0044 (0.0071)
Corral	1.09 (0.20)	0.018 (0.0024)	2.88 (0.74)	0.0016 (0.0047)	0.0013 (0.0044)	0.0012 (0.0043)
Crx	70.54 (23.14)	0.128 (0.0070)	62.18 (6.66)	0.0165 (0.0048)	0.0156 (0.0045)	0.0202 (0.0072)
Diabetes	18.96 (6.04)	0.143 (0.0060)	28.73 (11.33)	0.0175 (0.0051)	0.0179 (0.0054)	0.0375 (0.0317)
Flare	886.89 (182.15)	0.180 (0.0071)	240.24 (46.22)	0.0293 (0.0061)	0.1002 (0.0446)	0.0880 (0.0088)
Glass	10.49 (3.11)	0.029 (0.0025)	30.30 (21.49)	0.0018 (0.0050)	0.0019 (0.0051)	0.0031 (0.0063)
Glass2	2.08 (0.56)	0.018 (0.0017)	5.14 (1.54)	0.0006 (0.0031)	0.0006 (0.0031)	0.0012 (0.0043)
Hayes	0.61 (0.23)	0.022 (0.0019)	17.79 (5.35)	0.0012 (0.0042)	0.0015 (0.0047)	0.0019 (0.0053)
Heart	13.58 (4.89)	0.065 (0.0085)	10.98 (2.56)	0.0022 (0.0055)	0.0022 (0.0055)	0.0040 (0.0069)
Hepatitis	13.87 (2.72)	0.026 (0.0038)	6.64 (3.20)	0.0003 (0.0023)	0.0006 (0.0030)	0.0006 (0.0031)
Iris	0.65 (0.11)	0.012 (0.0013)	10.80 (2.99)	0.0003 (0.0023)	0.0013 (0.0044)	0.0003 (0.0023)
Letter	1601337 (142266)	6.238 (1.0785)	5126.70 (408.53)	15.624 (7.9495)	53.072 (25.049)	30.442 (11.972)
Lymphography	71.61 (16.52)	0.057 (0.0488)	87.32 (22.38)	0.0035 (0.0073)	0.0012 (0.0043)	0.0050 (0.0107)
Mofn	238.27 (153.79)	0.087 (0.0013)	11.63 (1.72)	0.0090 (0.0077)	0.0371 (0.0076)	0.1176 (0.0084)
Pima	13.05 (3.43)	0.169 (0.0186)	39.29 (5.07)	0.0137 (0.0051)	0.0166 (0.0039)	0.0281 (0.0063)
Shuttle	3643.50 (874.28)	0.094 (0.0087)	542.81 (81.49)	0.2477 (0.1146)	0.2602 (0.0531)	0.3383 (0.0271)
Tic-Tac-Toe	219.32 (50.79)	0.125 (0.0098)	56.64 (4.40)	0.1030 (0.0126)	0.0502 (0.0067)	0.0999 (0.0164)
Vehicle	1506.07 (286.77)	0.256 (0.0120)	183.35 (35.04)	0.0468 (0.0113)	0.0582 (0.0084)	0.0639 (0.0048)
Voting	36.46 (8.02)	0.026 (0.0031)	5.13 (2.45)	0.0053 (0.0074)	0.0024 (0.0057)	0.0031 (0.0063)
Average	74969.74 (6884.30)	0.374 (0.0589)	504.55 (181.87)	0.7465 (0.3772)	2.4768 (1.1637)	1.4440 (0.5529)
Excluding Letter	2285.60 (437.56)	0.094 (0.0104)	284.45 (171.07)	0.0381 (0.0166)	0.0675 (0.0263)	0.0631 (0.0092)

Table 12 (third column). Overall, the results indicate that the optimized RMCV competes well with all BNCs expect TAN, but TAN is a restricted BNC yielding accuracy that is inferior to that of RMCV (Table 8(a)).

It is very encouraging to see that RMCV can challenge NN (Table 13) (at least when considering run-times excluding the Letter dataset). Yet, RMCV loses to CT and SVM. Note, however, that the SVM implementation uses the C++ based LIBSVM package [30], rendering the comparison with the other MATLAB (interpreter based) implementations unfair.

Note that Tables 11–13 present two averages: with and without the Letter dataset. The Letter dataset, having 26 classes, 16 features, and 20,000 instances, poses difficulties to all classifiers and especially to the BNCs that compute a discriminative score (BNC-MDL, BNC-2P, and RMCV/RMHO). The run-time was affected not only from passing over the large number of instances, but also from keeping several very large matrices in memory, leading to inefficient and thus time-consuming use of memory. This significantly affected the optimized version of RMCV, which is more sensitive to memory-intensive tasks than its unoptimized counterpart and the other BNCs, as is reflected in Table 12. However, we also note that the average run-time of all classifiers in Table 13, except the neural network, is increased by about an order or two of magnitude when taking into consideration the Letter dataset.

6. Related work

In recent years, learning a BNC has been studied intensively (for up-to-date reviews see [7–10]). Some studies focus on learning restricted DAGs, usually NBC-based (e.g., TAN), that implement classification primarily through the positioning of the class variable as a parent of all DAG nodes, regardless of the learned structure and learning algorithm [4,39,40]. Keogh and Pazzani [39] extend NBC to TAN by learning edges between non-class nodes that increase the classification accuracy, instead of the conditional mutual information as in the original TAN [4]. Cheng and Greiner [40] compare NBC and TAN with the unrestricted general BN, for which the structure is learned using node ordering and based on a spanning tree. Sierra and Larranaga [41] search structures within a space created using a genetic algorithm allowing, in the first case, a crossover between BNs and a mutation of a BN or that, in the second case, is restricted to structures based only on Markov blankets (MBs) of the class variable. The selected structure, in the first case, is the one maximizing the K2 metric [2], and, in the second case, the one that maximizes the classification error (empirical risk under the 0/1 loss). Classifiers are learned for predicting the survival time of individuals diagnosed with malignant skin melanoma. Ling and Zhang [42] suggest learning a TAN that maximizes AUC rather than the classification accuracy. They consecutively add to NBC edges between a SuperParent [39] and its favorite child – each (separately) increases AUC the most – and keep adding such edges, as long as the AUC value of the derived BNC is improved. Ling and Zhang have found that BNCs learned according to this scheme provide both high AUC values and high classification accuracies. Recently, Pernkopf and Bilmes [10] considered the classification error as the ideal criterion for discriminative structure learning, assuming sufficient training data. Kontkanen et al. [5], Grossman and Domingos [6], and Guo and Greiner [7] drew a similar conclusion. However, for computational reasons, Pernkopf and Bilmes limit the use of this measure to learning only k-tree structures. These tree-restricted structures are also required to have edges pointing from the class node to each of the structure nodes and not vice versa (similar, e.g., to TAN). To further reduce

computational complexity, Pernkopf and Bilmes suggest several heuristic methods based on conditional mutual information and/or the classification error to order the variables (parents first) and select node parents according to this order. Results show an advantage of discriminative structure learning over generative learning.

While TAN-based classifiers relax some of the strong conditional independency assumptions of NBC, they are still restricted BNCs. Other restrictions to NBC that facilitate possible curse-of-dimensionality and improve the classifier accuracy are imposed through mechanisms of feature selection that let only variables that increase accuracy to be connected to the class variable (e.g., [11,43–46]). For example, in [44] a subset of the features is selected using topological orders based on conditional independence (CI) tests of increasing orders to maximize the amount of information provided by the structure.

Some studies that concern unrestricted DAGs concentrate on developing discriminative methods for parameter learning [9,47], comparing methods of discriminative and generative parameter learning [7,16], or comparing unrestricted, generatively-learned BNCs with restricted BNCs [48]. For example, Madden [48] finds that unrestricted BNCs learned using likelihood-based (MDL and BDeu) scores and careful parameter learning are comparable to TAN, and thus wonders why use unrestricted BNCs at all. He provides three reasons for that. First is the advantage of using an unrestricted BNC that models the whole distribution, demonstrates all node connections, and isolates variables that do not contribute to classification (as they are outside the class node MB). Second is the greater representational power of unrestricted BNCs over restricted classifiers (such as TAN). Third is the complexity of unrestricted BNCs that frequently is less than that of BNCs that are restricted to a fixed number of edges (e.g., between the class node and any non-class node of TAN), where some of these edges are quite often unnecessary.

Studies that are more relevant to our research are in discriminative (supervised) structure learning of the unrestricted BNC, where the class variable is in the focus of the criterion for learning the structure. Kontkanen et al. [5] – one of the pioneer studies in the field – demonstrates experimentally the advantage of supervised learning methods over unsupervised learning methods in learning a BNC. The authors make a distinction between the logarithmic loss function, commonly used in structure learning, and the 0/1 loss function that is more appropriate for optimizing a BNC. They also suggest using CV to estimate the criterion employed during learning and point to the connection between CV and an approximation of the factorization of the supervised (marginal) likelihood. Bilmes [49] introduces the explaining away residual discriminative measure for learning a dynamic BN for speech recognition. Grossman and Domingos [6] start from the empty structure, use an HC search, and learn a BNC using a CLL-based criterion that is either penalized using the MDL principle or balanced by a restriction of no more than two parents for a node (Table 4). Pernkopf and Bilmes [16] empirically compare both discriminative and generative parameter learning on both discriminatively and generatively structured BNCs. They found that discriminative structure learning of NBC, TAN, and the Bayesian multinet by optimizing the classification error produces the most accurate classifiers on almost half of the 25 datasets they examined.

Guo and Greiner [7] suggest different criteria for structure learning of unresticted BNCs and experimentally compare them for different complexities of the class variable MB. Using mainly synthetic data, Guo and Greiner compare (see Table 4 for a reference) likelihood-based criteria, such as MDL and a variant of the K2 metric called BDeu [12], class-conditional likelihood-based criteria, such as CLL and CMDL, a bias-variance (BV) criterion that is the expected mean-square-error of the classifier, and the classification error (CE). Although CE was found to be one of the best discriminative learning criteria, the authors recommend using BV for discriminative structure learning due to its superior performance. We note that no search algorithm is proposed to accompany CE and thereby to turn it into a structure learning concept/algorithm. Second, the CE score is evaluated in some synthetic scenarios (e.g., after modifying the true graph) that may represent a naive search procedure but not a real scenario or a real structure-learning task. Third, no classifier based on the CE score is either evaluated in real classification problems (but only on synthetic networks or modifications of the ALARM network) or compared to other BN and non-BN classifiers. Finally, the correspondence between CE and the true classification accuracy (error) – and therefore the contribution of CE to finding accurate BNCs – is not checked.

Pernkopf [18] compares methods of learning restricted or unrestricted BNCs to the k-nearest neighbor classifier following a preliminary feature selection. He found that both paradigms are comparable. One of Pernkopf's suggested methods – the selective unrestricted BN using classical floating search (CFS-SUN) – is the algorithm most similar to our (independently-developed) RMCV algorithm. CFS-SUN starts from the empty initial graph, and iteratively includes and excludes candidate variables to be connected to the structure based on a feature selection stage, as long as the BNC accuracy increases. Instead of focusing on a classification-based criterion for searching for a structure in a space of augmented DAGs (usually augmented NBC) or in the space of unrestricted DAGs, Acid et al. [8] focus on searching a structure in the space of partially directed acyclic graphs, even using the standard generative criteria (e.g., BDeu). The results in [8] demonstrate that attempts to improve the BNC accuracy should not necessarily concentrate only on finding discriminative scores, but should also explore more focused search spaces for the classifier using dedicated search procedures.

Other BNCs that have been investigated recently include multi-dimensional BNCs (MBCs) and Bayesian multinets. An MBC structure is partitioned to a class sub-graph (having a node representing each class), variable sub-graph, and a bridge (between class nodes and variables) sub-graph [50]. Borchani et al. [51] suggest an algorithm for learning each of the sub-graphs based on a wrapper greedy forward selection approach. A Bayesian multinet classifier is composed of a set of local BNs. Each local network represents a different class using a different set of independences [52]. By computing the joint probabilities of the local networks and the a priori probability of the class variable, and by using Bayes' theorem, classification of a test instance is made to the class that maximizes the class posterior probability. Learning a local network is based on

maximization of the network log-likelihood [4,53], minimization of the corresponding Kullback–Leibler (KL) divergence [54], maximization of the network (class) posterior probability [49], or maximization of the classification error [55].

7. Conclusions and discussion

We have proposed several scores that minimize risk based on the 0/1 loss function for learning BNCs; the RMCV score uses cross validation, whereas the RMHO score uses holdout. Evaluation using a holdout of $1/K$ of the data provides an inferior estimation of the generalization error relative to the K-fold (CV) estimation for small K values [56]. Thus, we are not surprised to find that RMCV generalizes better than RMHO and achieves greater classification accuracies, though RMHO cuts the run-time. RMHO-f improves accuracy estimation compared with RMHO, and RMHO-ar improves run-time compared with RMHO-f. Both variants are inferior to RMCV, with respect to accuracy, and comparable to RMCV, with respect to run-time, after the optimization of the implementation.

The RMCV classifier outperformed many other BNCs. The advantage of RMCV, when initialized using NBC, over NBC, can be explained by the fact that the RMCV algorithm starts from NBC and tries to modify it as long as its accuracy is improved. BNCs learned using the MDL or K2 scores exhibited inferior results to an RMCV-based BNC, since a structure achieving a better value of these scores is not necessarily a more accurate classifier. Even the classification-oriented scores, CMDL and CLL, were not as accurate as the RMCV score in predicting the most accurate classifier. Our experiments also showed that the generalization ability of these scores is limited when only a small number of instances is available. This is in contrast to the RMCV score, which generalized very well, even when the dataset was small. RMCV is also advantageous to BNC-MDL/2P in terms of classification accuracy, but in terms of run-time and complexity the three are roughly the same (although BNC-2P defines a smaller neighborhood).

A method to improve the run-time of BNC-2P by roughly two orders of magnitude [57] can be applied (at least partially) to RMCV. One can further improve RMCV run-time by reducing the searched neighborhood to the class variable MB, as changes that do not affect MB have no effect on the RMCV score (and classification accuracy) when learning with fully observed data. If, however, the data is incomplete as it may happen in real-world applications, the suggested improvement is inappropriate. Alternatively, we can search the space of partially directed acyclic graphs, which is smaller than that of DAGs [8].

Accuracies of the RMCV classifier and non-BN classifiers, such as CT, NN, and SVM, were roughly the same. This is encouraging, since the last three are state of the art classifiers and BN is not a priori considered an accurate classifier. While the run-time of training an RMCV classifier is longer than those of the non-BN classifiers, RMCV naturally provides in the Markov blanket of the class variable a simple mechanism of feature selection for finding the variables that explain the class variable the best. In addition, the learned BN structure can provide qualitative information about dependence, conditional independence, and causality relations in the domain for the benefit of the human user, beyond what the other non-BN models can provide. Whether or not all relations exhibited in a BN learned using a classification-oriented score are justified may be a subject of future research, as there is a reason to believe that the BN structure should exhibit correctness, at least in the vicinity of the class node (i.e., its MB). Other directions of future research are modifications of the RMCV score to account for classification errors using different losses or using discriminative parameter learning, and the use of more advanced search methods.

8. Summary

Likelihood-based (unsupervised) scores, such as MDL and K2, and structure learning algorithms that model the relations between variables without giving precedence to the class variable, often produce BNCs that are less accurate than optimal. Other algorithms that do consider the class variable attempt to estimate the class-conditional likelihood (supervised) score, for which exact computation is infeasible. Moreover, this attempt does not necessarily comply with expert predictions that are available in supervised learning.

In this work, a score called RMCV and an algorithm based on this score were suggested for learning accurate unrestricted BNCs. The score is based on risk minimization of the 0/1 loss function using cross validation, and thus is classification-oriented. Discriminative scores for learning a BNC, including the classification accuracy, increase the computational cost of learning, and thus the implementation of RMCV structure and parameter learning has been optimized. Our empirical evaluation using twenty-two classification problems usually demonstrated superiority of the proposed score and algorithm, with respect to accuracy and run-time, over other scores and algorithms. Thus, we believe that RMCV is a viable tool for learning accurate unrestricted BNCs and worthy of a place among other state of the art machine-learning classification algorithms.

Acknowledgments

This work was supported, in part, by the Paul Ivanier Center for Robotics & Production Management, Ben-Gurion University of the Negev, Beer-Sheva, Israel.

References

[1] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers. Inc., San Francisco, CA, USA, 1988.
[2] G.F. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, Mach. Learn. 9 (4) (1992) 309–347.
[3] D. Heckerman, A tutorial on learning with Bayesian networks, Tech. Rep. MSR-TR-95-06, Microsoft Research, 1995.
[4] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Mach. Learn. 29 (2–3) (1997) 131–163.
[5] P. Kontkanen, P. Myllymaki, T. Sliander, H. Tirri, On supervised selection of Bayesian networks, in: Proceedings of the 15th International Conference on Uncertainty in Artificial Intelligence (UAI-1999), Morgan Kaufmann Publishers. Inc., San Francisco, CA, USA, 1999, pp. 334–342.
[6] D. Grossman, P. Domingos, Learning Bayesian network classifiers by maximizing conditional likelihood, in: Proceedings of the 21st International Conference on Machine Learning (ICML-2004), 2004, pp. 361–368.
[7] Y. Guo, R. Greiner, Discriminative model selection for belief net structures, in: Proceeding of the 20th National Conference on Artificial Intelligence (AAAI-2005), 2005, pp. 770–776.
[8] S. Acid, L. Campos, J. Castellano, Learning Bayesian network classifiers: Searching in a space of partially directed acyclic graphs, Mach. Learn. 59 (2005) 213–235.
[9] T. Roos, H. Wettig, P. Grünwald, P. Myllymäki, H. Tirri, On discriminative Bayesian network classifiers and logistic regression, Mach. Learn. 59 (3) (2005) 267–296.
[10] F. Pernkopf, J.A. Bilmes, Efficient heuristics for discriminative structure learning of Bayesian network classifiers, J. Mach. Learn. Res. 11 (2010) 2323–2360.
[11] M.M. Drugan, M.A. Wiering, Feature selection for Bayesian network classifiers using the MDL-FS score, Int. J. Approx. Reas. 51 (6) (2010) 695–717.
[12] D. Heckerman, D. Geiger, D.M. Chickering, Learning Bayesian networks: The combination of knowledge and statistical data, Mach. Learn. 20 (1995) 197–243.
[13] A.Y. Ng, M.I. Jordan, On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes, in: Advances in Neural Information Processing Systems 14 (NIPS-2001), MIT, 1998, vol. 2, 2001, pp. 841–848.
[14] T. Jebara, Machine Learning: Discriminative and Generative, Kluwer Academic Publishers, Norwell, MA, USA, 2004.
[15] C.M. Bishop, J. Lasserre, Generative or discriminative? Getting the best of both worlds, in: J.M. Bernardo, M.J. Bayarri, J.O. Berger, A.P. Dawid, D. Heckerman, A.F.M. Smith, M. West (Eds.), Bayesian Statistics, vol. 8, Oxford University Press, 2007, pp. 3–24.
[16] F. Pernkopf, J. Bilmes, Discriminative versus generative parameter and structure learning of Bayesian network classifiers, in: Proceedings of the 22nd International Conference on Machine learning (ICML-2005), ACM, New York, NY, USA, 2005, pp. 657–664.
[17] R. Malka, Bayesian Network Classifiers, Master's thesis, Ben-Gurion University, 2005.
[18] F. Pernkopf, Bayesian network classifiers versus selective *k*-NN classifier, Pattern Recognit. 38 (1) (2005) 1–10.
[19] S.L. Lauritzen, D.J. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems, J. Roy. Statist. Soc., Ser. B 50 (1988) 157–224.
[20] W. Lam, F. Bacchus, Learning Bayesian belief networks: An approach based on the MDL principle, Comput. Intell. 10 (3) (1994) 269–293.
[21] A.P. Dawid, Present position and potential developments: Some personal views. Statistical theory. The prequential approach, J. Roy. Statist. Soc., Ser. A 147 (2) (1984) 278–292.
[22] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-1995), 1995, pp. 1137–1143.
[23] V.N. Vapnik, Statistical Learning Theory, John Wiley & Sons, New York, 1998.
[24] R. Cowell, Introduction to inference for Bayesian networks, in: M.I. Jordan (Ed.), Learning Graphical Models, MIT Press, Cambridge, Massachusetts, 1999, pp. 9–26.
[25] P. Langley, W. Iba, K. Thompson, An analysis of Bayesian classifiers, in: Proceeding of the 10th National Conference on Artificial Intelligence (AAAI-1992), 1992, pp. 223–228.
[26] A. Frank, A. Asuncion, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, <http://archive.ics.uci.edu/ml>, 2010.
[27] R. Kohavi, G. John, R. Long, D. Manley, K. Pfleger, MLC++: A machine learning library in C++, in: Proceedings of the 6th International Conference on Tools with Artificial Intelligence (TAI-1994), IEEE Comput. Soc. Press, 1994, pp. 740–743.
[28] K. Murphy, The Bayes net toolbox for Matlab, Comput. Sci. Stat. 33 (2001) 331–350.
[29] P. Leray, O. Francois, BNT structure learning package: Documentation and experiments, Tech. Rep. FRE CNRS 2645, Laboratoire PSI, 2004.
[30] C. Chang, C. Lin, LIBSVM: a library for support vector machines, <http://www.csie.ntu.edu.tw/∼cjlin/libsvm>, 2001.
[31] R.P.W. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D.M.J. Tax, S. Verzakov, PRTools4.1: a Matlab toolbox for pattern recognition, Delft University of Technology, 2007.
[32] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[33] R.W. Robinson, Counting unlabeled acyclic digraphs, in: C.H.C. Little (Ed.), Combinatorial Mathematics V, Lecture Notes in Mathematics, vol. 622, Springer, Berlin, Heidelberg, 1997, pp. 28–43.
[34] L. Breiman, J. Friedman, C.J. Stone, R.H. Olshen, Classification and Regression Trees, Wadsworth, Belmont, CA, 1984.
[35] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, UK, 1995.
[36] C. Cortes, V. Vapnik, Support vector networks, Mach. Learn. 20 (1995) 273–297.
[37] B. Lerner, R. Malka, Investigation of the K2 algorithm in learning Bayesian network classifiers, Appl. Artif. Intell. 25 (2011) 74–96.
[38] R.R. Bouckaert, E. Frank, Evaluating the replicability of significance tests for comparing learning algorithms, in: Proceedings of the 8th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD-2004), Springer, 2004, pp. 3–12.
[39] E.J. Keogh, M.J. Pazzani, Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches, in: Proceedings of the 7th International Workshop on Artificial Intelligence and Statistics (Uncertainty-1999), Ft. Lauderdale, Florida, 1999, pp. 225–230.
[40] J. Cheng, R. Greiner, Comparing Bayesian network classifiers, in: Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI-1999), Morgan Kaufmann Publishers. Inc., San Francisco, CA, USA, 1999, pp. 101–108.
[41] B. Sierra, P. Larrañaga, Predicting the survival in malignant skin Melanoma using Bayesian networks automatically induced by genetic algorithms – an empirical comparison between different approaches, Artif. Intell. Med. 14 (1998) 215–230.
[42] C.X. Ling, H. Zhang, Toward Bayesian classifiers with accurate probabilities, in: Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD-2002), Springer-Verlag, 2002, pp. 123–134.
[43] P. Langley, S. Sage, Induction of selective Bayesian classifiers, in: Proceedings of the 10th Annual Conference on Uncertainty in Artificial Intelligence (UAI-1994), Morgan Kaufmann Publishers. Inc., San Francisco, CA, USA, 1994, pp. 399–406.
[44] M. Singh, G.M. Provan, Efficient learning of selective Bayesian network classifiers, in: Proceedings of the 13th International Conference on Machine Learning (ICML-1996), Morgan Kaufmann Publishers. Inc., San Francisco, CA, USA, 1996, pp. 453–461.
[45] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1–2) (1997) 273–324.
[46] C. Aliferis, I. Tsamardinos, A. Statnikov, HITON: A novel Markov blanket algorithm for optimal variable selection, in: Proceedings of the American Medical Informatics Association Annual Symposium (AMIA-2003), 2003, pp. 21–25.
[47] R. Greiner, W. Zhou, Structural extension to logistic regression: discriminative parameter learning of belief net classifiers, Mach. Learn. 59 (3) (2005) 297–322.
[48] M.G. Madden, On the classification performance of TAN and general Bayesian networks, Knowl.-Based Syst. 22 (7) (2009) 489–495.
[49] J. Bilmes, Dynamic Bayesian multinets, in: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI-2000), Morgan Kaufmann Publishers. Inc., San Francisco, CA, USA, 2000, pp. 38–45.
[50] L.C. van der Gaag, P.R. de Waal, Multi-dimensional Bayesian network classifiers, in: Proceedings of the 3rd European Workshop on Probabilistic Graphical Models (PGM-2006), 2006, pp. 107–114.

[51] H. Borchani, C. Bielza, P. Larrañaga, Learning CB-decomposable multi-dimensional Bayesian network classifiers, in: Proceedings of the 5th European Workshop on Probabilistic Graphical Models (PGM-2010), 2010, pp. 25–32.
[52] D. Geiger, D. Heckerman, Knowledge representation and inference in similarity networks and Bayesian multinets, Artif. Intell. 82 (1–2) (1996) 45–74.
[53] M. Meila, M.I. Jordan, Learning with mixtures of trees, J. Mach. Learn. Res. 1 (2000) 1–48.
[54] K. Huang, I. King, M.R. Lyu, Discriminative training of Bayesian Chow-Liu tree multinet classifiers, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN-2003), vol. 1, 2003, pp. 484–488.
[55] Y. Gurwicz, B. Lerner, Bayesian class-matched multinet classifier, in: D.Y. Yeung, J.T. Kwok, A.L.N. Fred, F. Roli, D. de Ridder (Eds.), SSPR/SPR, Lecture Notes in Computer Science, vol. 4109, Springer, 2006, pp. 145–153.
[56] A. Blum, A. Kalai, J. Langford, Beating the hold-out: bounds for *K*-fold and progressive cross-validation, in: Proceedings of the 12th Annual Conference on Computational Learning Theory (COLT-1999), ACM Press, New York, USA, 1999, pp. 203–208.
[57] Y. Jing, V. Pavlović, J.M. Rehg, Boosted Bayesian network classifiers, Mach. Learn. 73 (2) (2008) 155–184.