Lior Rokach
Oded Maimon
Erez Shmueli *Editors*

Machine Learning for Data Science Handbook

Data Mining and Knowledge Discovery Handbook

Third Edition

Springer

Machine Learning for Data Science Handbook

Lior Rokach • Oded Maimon • Erez Shmueli
Editors

Machine Learning for Data Science Handbook

Data Mining and Knowledge Discovery Handbook

Third Edition

Springer

Editors

Lior Rokach
Department of Software and Information
Systems Engineering
Ben-Gurion University of the Negev
Beer-Sheva, Israel

Oded Maimon
Department of Industrial Engineering
Tel Aviv University
Ramat Aviv, Israel

Erez Shmueli
Department of Industrial Engineering
Tel Aviv University
Tel Aviv, Israel

ISBN 978-3-031-24627-2 ISBN 978-3-031-24628-9 (eBook)
https://doi.org/10.1007/978-3-031-24628-9

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Contents

Empowering Interpretable, Explainable Machine Learning Using Bayesian Network Classifiers

Boaz Lerner

1 Introduction

From the seminal works of Pearl [1], Spirtes [2], Lauritzen and Spiegelhalter [3], Cooper [4], and Microsoft Research's researchers (mainly Heckerman [5, 6], Meek [7], and Chickering [8]) and their colleagues introducing Bayesian networks (BNs), to the works admitting and demonstrating BN classifiers (BNCs) [9, 10, 11, 12, 13, 14, 15, 16, 17, 18], BNs were mostly considered an esoteric field, a neglected little brother of the more popular mainstream neural networks, support vector machines, and boosting and bagging machines and their machine learning (ML) variants. BNs and BNCs have attracted the attention of only relatively few faithful, non-mainstream scientists in the ML community who recognized, cherished, and advanced the networks' huge potential.

Whether due to the NP-hard complexity of BN structure learning [8], the BN traditional near-exclusive focus on discrete data [4, 5, 6], or the frequency of new trends in ML, few in the ML community found study of the powerful BN theory and tools attractive. Traditionally presented as a knowledge representation paradigm, until the late 90s BNs were not considered accurate classifiers and works demonstrating their superb classification capability are unjustifiably scarce even today.

While in training a classifier, we have to minimize a loss function (usually the 0/1 loss function), which is equivalent to maximizing the classification accuracy of a single target variable, when learning a BN, we usually maximize/minimize a general likelihood-driven [4, 6] or similar-fitting function (Akaike information criterion (AIC) [19], Bayesian information criterion (BIC) [20], or Kullback–

B. Lerner (✉)
Ben-Gurion University of the Negev, Be'er Sheva, Israel
e-mail: boaz@bgu.ac.il
https://www.ee.bgu.ac.il/~boaz/

L. Rokach et al. (eds.), *Machine Learning for Data Science Handbook*,
https://doi.org/10.1007/978-3-031-24628-9_7

Leibler (KL) divergence [21]) over the set of all variables. However, as the domain size increases, maximization of a likelihood-driven function over many variables used in classification—of which only one is of real interest, i.e., the class variable—can hardly lead to an accurate classifier [9].

In contrast to the belief that BNCs are less accurate than traditional ML classifiers, several studies [11, 13, 17, 22] have shown that properly learned, BNCs are comparable to traditional ML classifiers.[1] In this chapter, we will present some of these studies. Beyond accuracy however, the use of BNCs should be considered for the natural interpretability (understanding the results; the "how" question) and explainability (explaining the results; the "why" question) they provide.[2]

Let us demonstrate the difficulty in achieving classifier interpretability using three examples. The linear regression model accompanies a predictor with a coefficient measuring "importance" and direction of impact, but with credibility that depends on the predictor value size. The ensemble classifier (whether by averaging methods such as bagging and the random forest (RF), or boosting methods such as the XGBoost) provides a ranked list of "important" variables but may show only slight differences between their levels of "importance"—say by a second or third digit after the decimal point—providing negligible information about their relative contribution to the classification. With hundreds and thousands of neurons residing in many internal modules and layers, connected by millions of parameters, the deep neural network (DNN) excels in classification but due to these complexities inherently lacks interpretability and explainability [30]. The BNC, in contrast to these examples, shows a hierarchy of interrelations among domain variables along with causal paths of influence and inference mechanisms, revealing causal relations that can readily be investigated while and for interpreting and explaining the domain.

ML tools are frequently accused of being black boxes,[3] sacrificing interpretability in favor of usability and effectiveness [31, 32, 33, 34]. In times when ML is struggling to enhance its transparency, fairness (even when the bias is in the data and not in the analysis), and accountability [35], to be auditable, to increase trust and trustworthiness among ML developers and non-ML users alike [33, 34, 36],

[1] Unfortunately, most comparisons of BNCs are among themselves [18, 23, 24, 25, 26, 27] and not to non-BNCs.

[2] Conventional categorization of interpretable ML methods [28] is through analysis of model components using, for example, linear regression and decision trees, sensitivity studies of input perturbations, or analysis of local or global surrogate approximations of the ML model [29]. Although these methods show readiness and stability, many challenges, such as dealing with dependent features, causal interpretation, and uncertainty estimation remain. These challenges need to be resolved for successful application of interpretable ML methods to scientific problems [28, 30].

[3] This is not necessarily true for all ML tools, but DNNs, following their recent meteoric rise, attract this criticism on behalf of the entire ML field, as DNNs can justifiably be considered black boxes even for ML specialists.

and even to be responsible [32],[4] researchers do their utmost to develop dedicated schemes that can help explain the predictions and decisions made by ML models [30, 37]. One example of this is the SHapley Additive exPlanations (SHAP [38]) and local interpretable model-agnostic explanations (LIME [29]) for linear models and ensemble classifiers. Another example is the tremendous effort of researchers to enhance DNN transparency, visualizability, and explainability. This may be done by advancing visualization methods as a fundamental building block that, combined with additional tools, will empower humans to understand DNNs [39], by generating saliency maps that indicate the relevance of image pixels to the network output [40, 41], or building inference graphs to interpret hidden layer activity for understanding the general inference process of a class, as well as explaining decisions the network makes regarding specific images [42]. While these schemes enrich our explainability tools, they have only limited causal interpretations [43]. A causal explanation for the mechanism of a DNN gives insight about the meaning of the DNN's output, its relation to the network input, and any change in it. In highly sensitive domains involving peoples' lives, company finances, criminal justice, and autonomous driving, a causal explanation is critical to creating justification, transparency, trust, and eventually co-operation. The ultimate goal is that any such causal explanation will be accessible and comprehensible to a human, who may then challenge the explanation until it is fully understood.

The BNC is a natural means for knowledge representation. Its graphical structure [1, 2] on the one hand and causal inference mechanisms [44, 45] on the other hand readily convey interpretability and explainability. First, the BNC provides a feature-selection mechanism through its Markov blanket (MB).[5] Conditioned on its MB, a variable is independent of the rest of the nodes in the network, allowing us to focus our attention on exploring the importance of interrelations of this variable only with those in its MB. This gives a BNC an advantage over conventional feature (variable)-selection and importance ranking methods that can only analyze variables separately (or in simple interactions), and lack any ability to consider their interrelations. Moreover, the BNC demonstrates a hierarchical structure of interrelations among both MB and non-MB variables, allowing us to explore and understand the source of the feature importance (e.g., by being included in the MB) and to identify causal paths of influence[6] originating, passing, or ending in/through

[4] Interpretability, explainability, transparency, fairness, accountability, auditability, trustworthiness, and responsibility—can we wholeheartedly confirm that we have demanded all of these from ourselves as human decision-makers in the era prior to machine learning? I will leave this question open until the discussion.

[5] The MB of a network node (representing a domain variable) includes the node, its parents, children, and children's co-parents.

[6] Implicit paths that may become explicit in the absence of latent variables in the domain or following intervention [44]. Nevertheless, experimental studies exercising intervention are hard to make and follow, whereas observational approaches such as those exerted by the BN are easy to make, follow, and interpret either by assuming no latent mechanisms in the domain or by learning these mechanisms from data (see Sect. 4).

a variable [1, 2, 44]. Regarding classification, the MB of the target (class) variable fosters discriminability in a lower dimension by enabling a quick exploration of those causal relations within the blanket that contribute to accurate classification [46, 47, 48, 49, 50, 51, 52]. Simultaneously, MBs of selected non-target variables allow deeper exploration and understanding of mechanisms establishing the domain and thereby also promote interpretability and explainability. Like using the MB, a BNC may also allow investigation of why an instance (pattern) has been classified positively or negatively by identification of a minimal set of the currently active features responsible for the current classification, or a minimal set of features whose current state (active or not) is sufficient for the classification [53]. Moreover, the BN enables inference on the values of any combination of variables (related or not to the classification) conditioned on values of the remaining variables (or only those in a specific MB) [46]. Finally, studying intervention as a source of causality is natural only to using the inference tools of graphical models [44, 45]. That is, BNs, BNCs, and graphical models, in general, have tremendous potential to promote explainable AI, and the ML community is encouraged to divert some of its efforts in this direction. We will make the argument for this later in the chapter, specifically in Sects. 3.4 and 4.

However, in order to also provide interpretability and explainability in domains having non-semantic huge (in space and/or time) inputs such as those found in images, speech, and text, much progress in BN learning algorithms is needed. Today, neither BNs nor BNCs can cope with high-dimensional raw data coming from images, speech, and text, and thus the contribution of these networks, specifically to advance interpretability of DNNs, can only come when applied to already processed, projected, or embedded raw data. We will address this in the conclusion.

In Sect. 2, we describe the most popular, though restricted, BNC, the naïve Bayesian classifier, and some of its many variants. In Sect. 3, we survey general unrestricted BNCs, focusing on the risk minimization by cross-validation BNC, and in Sect. 4, we extend our study to causal–temporal BNs and their application to classification. Section 5 concludes the chapter.

2 Restricted BNCs–The Naïve Bayesian Classifier and Its Variants

Although they make assumptions about the domain that are sometimes found unjustified (variables are class-conditionally independent [54], a maximal number of node parents [2], existence of a variable topological order [4]), restricted BNCs are surprisingly accurate as well as efficient, as structure learning is avoided or dramatically shortened. In this chapter, we will focus on and review the naïve Bayesian classifier (NBC), which is the most fundamental restricted BNC, continue by demonstrating variants of the NBC, which try to relieve the class-conditional independence assumption of the NBC, and conclude by presenting a new variant of the NBC demonstrating empirical advantage.

2.1 Naive Bayesian Classifier

Although outdated, the NBC [54]—a learning-free structure, which is obtained with virtually no computational effort—is considered a simple, practical, and state-of-the-art classifier [9, 13, 18], very often selected as the default for pattern classification in industrial applications.

The NBC is a special case of a BN consisting of a finite set of random variables, $U = \{X_1, X_2, \ldots, X_m, C\} = \{\boldsymbol{X}, C\}$, where $X_1, \ldots, X_m$ are the observable variables that represent the problem features and C is the class variable having K states. The NBC is termed naïve since it makes use of a simplifying assumption that its observable variables are conditionally independent given the class variable. All edges of the NBC are directed from the class variable to the observable variables (Fig. 1); hence, the only parent of the observable variables, X_i, is $Pa_i = C$, and $Pa(C) = \emptyset$ for the class variable.

Given a selected network structure, the NBC assigns a test pattern $\mathbf{x}$ to the class C_k $(k = 1, \ldots, K)$ with the highest a posteriori probability

$$P(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)P(C_k)}{p(\mathbf{x})}, \tag{1}$$

where $p(\mathbf{x}|C_k)$ is the class-conditional probability density, $P(C_k)$ is the a priori probability for class C_k, and $p(\mathbf{x})$ is the unconditional density normalizing the product of the two probabilities such that $\sum_k P(C_k|\mathbf{x}) = 1$. The NBC independence assumption eliminates the "curse of dimensionality" since density estimation requires only linearly rather than exponentially increasing numbers of patterns. Omitting $p(\mathbf{x})$, which is common to all classes, the posterior probability can be written as

$$P(C_k|\mathbf{x}) \propto p(\mathbf{X} = \mathbf{x}|C_k)P(C_k) = P(C_k)\prod_{i=1}^{m} p(X_i = x_i|C_k), \tag{2}$$

where $\mathbf{X} = \mathbf{x}$ represents the event that $X_1 = x_1 \wedge X_2 = x_2 \wedge \ldots \wedge X_m = x_m$ and $\prod_{i=1}^{m} p(X_i = x_i|C_k)$ is a product of class-conditional densities for $\mathbf{x}$.

Both $P(C_k)$ and $p(X_i|C_k)$ can be estimated from the training data. $P(C_k)$ is the relative frequency of patterns belonging to C_k out of all of the patterns

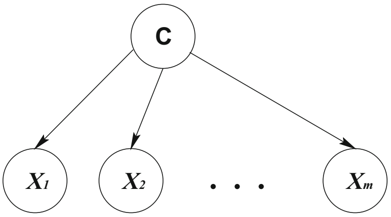


Fig. 1 The NBC depicted as a BNC in which the observable variables $(X_1,X_2, \ldots X_m)$ are conditionally independent given the class variable (C)

in the data (the class prior probabilities). To estimate $p(X_i|C_k)$ (or $p(x_i|C_k)$), the one-dimensional class-conditional probabilities (or densities) for discrete (or continuous) variables, for each class C_k and variable X_i, we employ a training set comprising of a finite number of patterns $\mathbf{x}^n$, where n gets values for each of the N_k training patterns of class C_k. For a discrete variable, the class-conditional probability is estimated using the sample (unsmoothed or smoothed) frequency of each value of the variable [6, 9, 25]. For a continuous variable, the parameters of $p(x_i|C_k)$ are usually estimated by maximum likelihood using any of the popular density estimation methods such as single Gaussian estimation, which assumes the data are generated from a single normal distribution, kernel density estimation models, representing the data using a linear combination of kernels around each of the training patterns, or the Gaussian mixture model, which estimates the data using a few Gaussians with adaptable parameters [55, 56].

The NBC is an interpretable model because of the (conditional) independence assumption; it is very clear how much each feature contributes toward a certain class prediction, since we can interpret the conditional probability. When the degree of independence between variables is high and the naïve assumption is justified and/or the database is small, appropriate for the small-scale learning problem of only classifier parameters, an NBC provides an accurate classifier as was demonstrated, for example, in diagnosing genetic abnormalities [56, 57, 58].

2.2 Variants of the NBC

The NBC is based on the assumption that all attributes (variables) are mutually independent, conditioned on the class attribute. On the one hand, this assumption ignores attribute dependencies and is thus often violated. On the other hand, learning from data, a BNC that can represent arbitrary attribute dependencies is intractable (Sect. 3). Thus, researchers have focused their attention on improving the NBC, which has led to many effective and efficient leaning algorithms [9, 26, 49, 59, 60, 61, 62, 63]. These may be divided into those of *feature selection* (learning an NBC based on a subset of the features that better satisfy the independence assumption), e.g., for cycle-time key factor identification and prediction in semiconductor manufacturing [64], *local learning* (learning an NBC based on a local training set rather than the whole set), *structure extension* (learning an NBC also representing dependencies among some features), *data expansion* (learning an NBC using a training set expanded from the original), and *multinet classifiers* (learning a classifier to each class separately).

For example, lazy learning algorithms are popular local learning methods for extending the NBC. Lazy learning delays learning until classification time by storing training data and waiting until it is given a test instance; that is, generalization is delayed until test time, generating a hypothesis for each instance instead of generating one hypothesis for all instances. Among lazy learning algorithms, we find the lazy Bayesian rule (LBR) [60] and selective neighborhood naïve Bayes

(SNNB) [61]. Another algorithm is the locally weighted naïve Bayes (LWNB) [62]. In LWNB, the *k*-nearest neighbors of a test instance are first found, and each of them is weighted in terms of its distance from the test instance. Then a local NBC is trained using the locally weighted training instances. Although it is a *k*-related algorithm, its classification performance is not particularly sensitive to the size of *k* as long as it is not too small.

Among numerous proposals to improve the accuracy of NBC by structure extension, the one-dependence estimator (ODE) is similar to the NBC except each attribute is allowed to depend on at most one other attribute, in addition to the class attribute. The ODE provides a simple, yet powerful alternative to NBC. Its most popular variant is the tree-augmented network (TAN) [9] that uses the Chow-Liu algorithm [65] to learn a maximum weighted spanning tree over all non-class variables that are connected pairwise by edges weighted by the conditional (on the class variable) mutual information between these variables. The super-parent (SP) TAN algorithm (SP-TAN) [26] greedily learns a TAN from NBC by adding in each iteration the edge achieving the highest (cross-validation) accuracy improvement. It demonstrates remarkable classification performance but at a considerable computational cost. The averaged one-dependence estimator (AODE) [59] weakens the NBC attribute independence assumption by averaging all SuperParent-one-dependence estimators (SPODEs) [26] that satisfy a minimum support constraint, where a SPODE allows each attribute to depend on a common single attribute (i.e., SP) in addition to the class. This technique achieves comparable classification accuracy to SP-TAN with a substantially improved computational efficiency at training time. In ensemble selection of SPODEs, we select only some of the SPODEs that are averaged by AODE. This improves the classification accuracy while reducing the classification runtime, albeit at a cost of additional training time. *K*-dependence Bayesian (KDB) and selective KDB (SKDB) [22] classifiers allow every variable to be conditioned on the class and, at most, *k* other attributes. SKDB classifiers showed an advantage over the NBC and TAN. Other methods may initialize a structure search procedure, such as the K2 algorithm [4], using the NBC in order to extend the naïve classifier using more meaningful connections among graph nodes that may improve its performance [66].

The Bayesian multinet classifier (BMC) is another powerful extension of the NBC [9, 67]. A BMC comprises a set of local networks, each corresponding to a value that the class node can take. While a BNC forces the relations among the attributes to be the same for all values of the class node, a BMC allows these relations to be different for different values of the class node, forming a local network for each class and thereby providing a more expressive representation than, for example, the NBC and TAN. Conventionally, each local network is learned by minimizing the KL divergence (also maximizing the log likelihood [9]) to induce a Chow–Liu (CL) tree [65]. Using the estimated class prior probabilities, the BMC classifies to the class maximizing the product of the prior and the variable joint probability for this class estimated using only the class patterns. Although a local network must be searched for each class, the BMC is generally more accurate and has a smaller computational complexity than a BNC because each local network

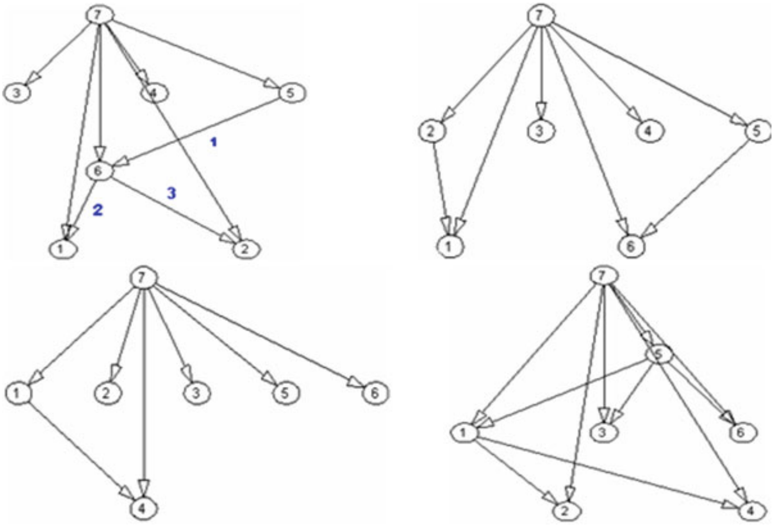


Fig. 2 Four local networks learned by the tBCM^2 algorithm for the four classes of the UCI Car database: C_1 (top-left), for which the order of edge learning is indicated, C_2 (top-right), C_3 (bottom-left), and C_4 (bottom-right). Node 7 is the class variable [69]

has a simpler problem to model, using a lower number of nodes in both a static scenario [67] and a dynamic one [68]. The BMC has two flaws [69]. The first is that constructing a CL tree using joint-probability-based scores for evaluating a structure is less specific to classification, i.e., CL multinet classifiers based on structures providing high scores are not necessarily accurate. The second flaw is that training a local network is based only on patterns of the corresponding class. Although this approach may approximate the class data effectively, information discriminating between the class and other classes may be discarded, undermining selection of the structure that is most appropriate for distinguishing this class. The TAN-based Bayesian class-matched multinet (tBCM^2) [69] utilizes a discrimination score for each local network separately, which maximizes accuracy by simultaneously detecting and rejecting patterns of the corresponding class and other classes, respectively, using both the entire data set, and the SuperParent algorithm to learn the TAN that maximizes this discrimination score. The tBCM^2 demonstrated [69] superiority over the naïve Bayesian, TAN, CL multinet, and recursive Bayesian multinet (RBMN) [70] classifiers for 32 UCI [71] databases. Figure 2 shows an example of a BMC learned using tBCM^2 for the UCI Car database.

2.3 Experimental Evaluation of NBC Variants

While [72] select the same ensemble of SPODEs (SuperParent-one-dependence estimators [26]) for all the classes, the multi-class SPODE (MSPODE) we present

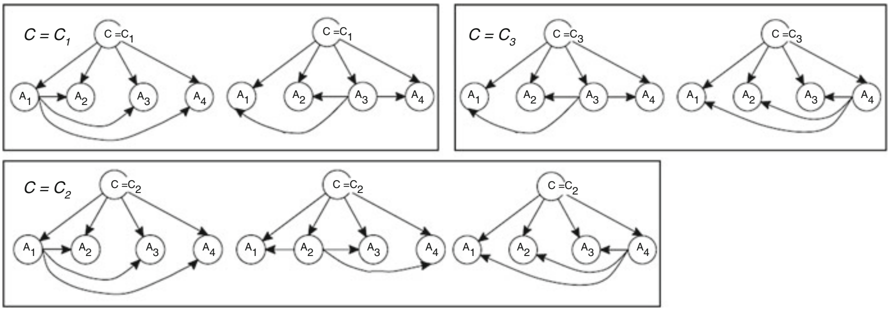


Fig. 3 An example MSPODE for a four-dimensional three-class classification problem. Note the existence of a super parent in each local network of an SPODE

here, inspired by the BMC, selects a different ensemble of SPODEs for each class (Fig. 3). An MSPODE may be learned for each class using the conditional mutual information [1] or detection–rejection measure [69]. This classifier has been analyzed theoretically and is empirically evaluated here in Table 1 in comparison to other BNCs using 32 UCI [71] data sets in terms of classification accuracy, and training and test times. Table 1 shows that although all conventional approaches are effective, accuracy of the MSPODE is superior with a similar time complexity.

3 Beyond the NBC—the Unrestricted BNC

NBC and its variants often lack not only accuracy, but also interpretability and explainability due to the naïve assumptions they make and the limited structure spaces they search. However, along with the traditional classifiers based on the neural network (NN), support vector machine (SVM), decision tree, and ensembles (e.g., the RF and XGBoost family), classifiers based on the BN (and not the NBC) have been introduced and studied for 25 years [9, 10, 11, 12, 13, 14, 15, 16, 17, 25, 27, 47, 48, 66]. To promote the use of the BNC, we first present the general BN (Sect. 3.1) on which the BNC is based. We then outline the conventional (likelihood-driven) unrestricted BNC (Sect. 3.2), before introducing the risk minimization by cross-validation (RMCV) BNC (Sect. 3.3).

3.1 The General BN

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 $\mathbf{X}$ (usually, X_i refers to both the variable

Table 1 Accuracies (%) and time complexities of our suggested multi-class SPODE (MSPODE) vs. popular variants of the NBC classifying 32 UCI databases. Besides the NBC, the variants are: Averaged one-dependence estimator (AODE) [59], lazy AODE (LAODE) [73], one ensemble selection of SuperParent-one-dependence estimators (ESPODE) [72], and tree-augmented network (TAN) [9]. k, t, n, and v are the numbers of classes, training instances, features, and feature values (average), and s is the average similarity value between a test instance and each training instance [73]

Database	NBC	AODE	LAODE	ESPODE	TAN	MSPODE
Australian	85.2	85.6	85.0	**86.4**	84.0	**86.4**
Balance	**90.9**	87.2	88.1	85.2	87.3	88.7
Bupa	64.7	66.5	67.1	66.5	**67.7**	66.5
Car	86.1	88.4	88.5	88.7	**94.3**	90.4
Corral	86.7	93.3	95.0	**98.3**	97.9	**98.3**
Crx	86.1	**89.2**	**89.2**	**89.2**	85.9	**89.2**
Cytogenetics	78.1	82.3	82.1	80.2	81.3	**83.4**
E.coli	86.4	83.3	85.9	84.9	**86.9**	85.4
Flare	82.0	**86.5**	86.2	**86.5**	85.6	**86.5**
Hayes	79.5	82.5	**83.1**	78.1	76.7	79.1
Hepatitis	70.0	72.5	72.5	73.7	72.5	**73.8**
Ionosphere	91.7	91.7	**93.1**	91.1	92.3	92.0
Iris	94.7	94.7	94.0	93.3	94.3	**96.7**
Krkp	88.4	91.9	N/A	93.5	**94.3**	93.5
Led-7	74.6	74.8	N/A	74.5	74.0	**75.5**
Lymphography	85.2	86.2	82.1	86.2	80.7	**88.0**
Monks	96.4	**98.9**	98.7	98.4	97.3	98.7
Nursery	90.2	92.7	N/A	92.4	93.4	**93.7**
Pendigit	87.3	97.6	N/A	**97.7**	95.7	97.6
Pima	76.0	75.9	75.1	75.4	75.5	**76.5**
Post-operative	67.5	68.7	70.0	70.0	**71.2**	70.0
Segment	92.1	95.4	**97.0**	95.7	94.4	96.4
Shuttle	98.7	99.8	N/A	99.8	**100**	99.8
Splice	94.8	95.5	N/A	95.5	88.8	**96.8**
Tic Tac Toe	69.4	75.8	**81.8**	74.5	74.7	74.5
Tokyo	91.3	**94.2**	93.9	93.3	91.9	93.5
Vehicle	61.5	72.5	72.9	72.7	70.0	**73.0**
Voting	91.3	**96.1**	**96.1**	94.8	94.4	95.7
Vowel	66.7	87.2	90.8	89.3	83.5	**91.1**
Waveform-21	81.7	84.5	N/A	82.8	82.0	**85.4**
Wine	**98.8**	97.7	97.1	97.1	98.2	**98.8**
Zoo	93.0	93.0	**96.0**	93.0	**96.0**	94.0
Average accuracy	84.0	86.9	86.4	86.8	86.3	**87.7**
Training complexity	$O(tn)$	$O(tn^2)$	$O(tn^2)$	$O(ktn^2)$	$O(n^2(t+kv^2))$	$O(ktn^2)$
Test complexity	$O(kn)$	$O(kn^2)$	$O(stn^2)$	$O(kn^2)$	$O(kn)$	$O(kn^2)$

[a] Bold font is for most accurate classifier for a task

and its corresponding node), and $\mathbf{E}$ is a finite set of directed edges of $\mathcal{G}$ connecting $\mathbf{V}$. Edges and missing edges encode dependencies and conditional independencies, respectively, in $\mathcal{G}$. 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, $\mathbf{Pa}_i \subset \mathbf{X}$. Most studies, including this one, focus mainly on 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 [1, 2, 4, 5],

$$P(\mathbf{X} = \mathbf{x}|\mathcal{G}) = \prod_{i=1}^{n} P(X_i = x_i|\mathbf{Pa}_i, \mathcal{G}), \tag{3}$$

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 [3], but when there is only one hidden node (e.g., the class node in classification), direct inference based on Eq. 3 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 search-and-score (S&S) approach to learning a structure from data [4, 5, 6] comprises a search for the structure achieving the highest score, e.g., hill climbing (HC), and a score, generally the Bayesian score,

$$P(\mathcal{G}|D) = \frac{P(D|\mathcal{G})P(\mathcal{G})}{P(D)} = \frac{P(D, \mathcal{G})}{P(D)} \tag{4}$$

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

3.2 The General BNC

While the BN provides a powerful graphical model for encoding the probabilistic relationships among a set of variables and can therefore naturally be used for classification, 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. Learning a 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—the class variable—and not necessarily of all variables. When focusing on structure learning, exhaustively searching the space of possible graphs is infeasible [4], and thus S&S structure learning algorithms sub-optimally search the space and select the structure achieving the highest value of a score [4, 5, 6]. 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. Researchers have demonstrated that BNC structures learned using generative scores do not usually contribute to high classification accuracy [9, 10, 11, 12, 16, 17] since there is lack of agreement between these scores used for learning and the score used for evaluation, which is the classification accuracy. That is, classifiers based on structures having high generative scores are not necessarily highly accurate.[7]

It is clear from Eq. 4 that a score should reflect a correspondence between the structure and the data. The minimum description length (MDL) score [74] can approximate $P(D|\mathcal{G})$—the marginal likelihood [5, 6]—but [9] argued that this score is not suitable for classification and instead recommended the class-conditional log likelihood (CLL) (as opposed to log likelihood (LL)),

$$CLL(\mathcal{G}|D) = \sum_{i=1}^{N} \log P(c_i|v_i'), \tag{5}$$

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 pattern is learned [10]:

$$P(c^N|v'^N, \mathcal{G}) = \frac{P(c^N, v'^N|\mathcal{G})}{P(v'^N|\mathcal{G})} = \frac{P(D|\mathcal{G})}{\sum_{c'^N} P(c'^N, v'^N|\mathcal{G})}, \tag{6}$$

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 [10] considers the left-hand side of Eq. 6 and the marginal likelihood Eq. 4 as the supervised and unsupervised marginal likelihoods, respectively. The marginalization over the parameters in Eq. 6 is

$$P(c^N|v'^N, \mathcal{G}) = \int_{\Theta} P(c^N|v'^N, \Theta, \mathcal{G}) P(\Theta|v'^N, \mathcal{G}) d\Theta, \tag{7}$$

[7] Note, however, that although constraint-based structure learning algorithms of BNs [2] are usually not considered in inducing a BNC, they may nevertheless lead to supreme BNCs [27].

and its approximation [10] using a single term is

$$P(c^N|v'^N, \mathcal{G}) \approx P(c^N|v'^N, \hat{\Theta}, \mathcal{G}). \tag{8}$$

$\hat{\Theta}$ 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 [9].

Another predictive local criterion (LC) [5] for learning a BNC [10] is based on the prequential approach [75],

$$LC(D, \mathcal{G}) = \sum_{i=1}^{N} \log P(c_i|\{v_j\}_{j=1}^{i-1}, v'_i, \mathcal{G}). \tag{9}$$

Other cumulative logarithmic loss scores [10] use 10-fold cross-validation (CV) or leave-one-out, which are reputable methods for model selection [76]. They are 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:

$$CV_K(D, \mathcal{G}) = \sum_{k=1}^{K} \sum_{i=1}^{N/K} \log P(c_{i+A_k}|D \setminus D_k^K, v'_{i+A_k}, \mathcal{G}), \tag{10}$$

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 (Eqs. 7, 9, or Eq. 10) 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:

$$L(c_i, \widehat{c_i}) = \begin{cases} 0, & c_i = \widehat{c_i} \\ 1, & c_i \neq \widehat{c_i} \end{cases}, \tag{11}$$

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.

3.3 Risk Minimization by Cross-Validation

We propose risk minimization by cross-validation (RMCV) for a classification-oriented score and an S&S algorithm for learning unrestricted BNCs. Note that other uses of classification-oriented scores in learning unrestricted BNCs [12, 13] are in a somewhat different context. 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. This model does not need to estimate the true distribution, generate data from this distribution, or infer about any non-class variable. That is, RMCV performs discriminative learning of a generative (BN) model. It learns generative models that are complicated, only to discriminate accurately among classes. The RMCV uses the 0/1 loss function, which is a classification-oriented score for unrestricted BNCs and non-BN classifiers alike. Its superiority to marginal and class-conditional likelihood-based scores with respect to classification accuracy using small real and synthetic problems, allowing for learning all possible graphs, was empirically demonstrated [17].

Instead of selecting a structure based on summation of supervised marginal likelihoods over the data set (Eq. 9 or Eq. 10), we suggest selecting a structure based on summation of false decisions about the class state over the data set. Our score is based on risk minimization [77] 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, the score is called risk minimization by cross-validation (and we deliberately do not simplify $\frac{1}{K}\frac{K}{N}$) [17]:

$$RMCV_K(D,\mathcal{G}) = \frac{1}{K}\sum_{k=1}^{K}\frac{K}{N}\sum_{i=1}^{N/K} L(c_{ki}, \arg\max_{c\in\{c_1,\dots,c_{r_C}\}} P(C=c|D\setminus D_k^K, v'_{ki}, \mathcal{G})), \tag{12}$$

where $v_{ki}=(c_{ki}, v'_{ki})$ is the ith instance of D_k^K and $L(\,,\,)$ is the 0/1 loss function (Eq. 11). Being a CV-based score, RMCV is easy to implement and computationally feasible (see, for example, Eq. 8), and it depends on only one parameter (K). Further, it is argued [10] that a CV-based score can be regarded as an approximation of a factorization of the supervised marginal likelihood (Eq. 6). Note that the RMCV score is normalized by the data set size N, whereas Eqs. 9 and 10 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 when comparing scores over data sets. 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 [5] (assuming discrete variables), where the distribution parameters are obtained using maximum likelihood [4], similar to [10, 25]. Moreover, [11] argued, based on experiments, that maximum likelihood 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 when using maximum likelihood 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 here. Also, note that the same measure is used for learning the BNC and for evaluating its accuracy, which makes learning oriented toward classification. Similar to CLL-based scores [10, 11], RMCV is not decomposable.

A suggested S&S structure learning algorithm consists of the RMCV score and a simple HC search [17]:

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 $RMCV_K(D, \mathcal{G})$
converged:= false
While converged = false
    For each $\mathcal{G}' \in$ Neighborhood($\mathcal{G}$)
        compute $RMCV_K(D, \mathcal{G}')$
    $\mathcal{G}^* := \arg\min_{\mathcal{G}'} RMCV_K(D, \mathcal{G}')$
    If $RMCV_K(D, \mathcal{G}^*) < RMCV_K(D, \mathcal{G})$
        Then $\mathcal{G} := \mathcal{G}^*$
        Else converged:= true
$\Theta$:=LearnParameters($D, \mathcal{G}$)
Return ($\mathcal{G}$, $\Theta$)
```

The RMCV algorithm starts with any initial graph (e.g., the empty graph or the NBC) 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 $D \setminus D_k^K$, 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 (Eq. 12). 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 [78]. 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. 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 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 maximum likelihood 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 set of parents has 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 Eq. 3, for the initial/current DAG.

3.4 Experimental Evaluation of Unrestricted BNCs

Our first evaluation (Sect. 3.4.1) follows previous research that conventionally uses synthetic and traditional data sets. The empirical investigation includes several unrestricted BNCs. The RMCV algorithm is generally initialized by either the NBC or an empty structure to induce the RMCV (NBC) or RMCV (Empty) BNCs, respectively [17]. Along with the NBC, these RMCV BNCs are compared here with three other types: BNCs learned using the K2 score and algorithm [4] initialized by either the NBC or empty structures [66], i.e., K2 (NBC) and K2 (Empty); BNCs learned using HC search initialized by either the NBC or empty structures to minimize the MDL score [74] (maximize the marginal likelihood), i.e., MDL (NBC) and MDL (Empty) [17]; and BNCs learned to maximize the class-conditional log likelihood, initialized by the empty graph, which use either HC search or a two-parent limitation on a node, i.e., BNC-MDL and BNC-2P [11].

Our second evaluation (Sect. 3.4.2) presents original studies using the RMCV algorithm and own authentic data sets from five real-life domains in genetic abnormality inspection, semiconductor manufacturing, Parkinson's disease diagnosis, amyotrophic lateral sclerosis (ALS) prediction and explanation, and young driver motorcycle accidents analysis.

3.4.1 Evaluation of BNCs Using Traditional Data Sets

BNCs were found comparable and even superior to non-BN classifiers in different reports. Grossman and Domingos [11] showed that when a BNC's structure is learned to optimize the conditional likelihood of the class variable (Eq. 5), it is advantageous to the classification and regression tree (CART) classifier [79]. Pernkopf [13] compared variants of the NBC and BNCs to the selective k-nearest neighbor classifier (skNN), selecting by sequential feature selection a subset of features that maximizes the classification performance. He showed using several UCI [71] databases that the BNCs are usually more accurate than the skNN and are superior with respect to memory requirements and computational demands during classification. The selective k-dependence Bayesian (SKDB) classifier [22] showed an advantage over the NBC, TAN [9], and AODE [59] (see Sect. 2.2 for details about the two later classifiers), and comparable accuracy to the RF [80], with no significant difference between them based on the Friedman test followed by the Nemenyi post-hoc test [81].

Kelner and Lerner [17] reported classification performance using 22 UCI [71] databases with various characteristics, e.g., the numbers of classes, features, and patterns between 2 and 26, 4 and 36, and 80 and 20,000, respectively. Table 2, extracted from [17], shows dominance of the RMCV over all other BNCs, and also over non-BNCs, such as the CART, three-layer-perceptron NN [82], and SVM [83] with its three conventional kernels—all non-BNCs were optimized for each task separately. According to a Friedman test followed by a Bonferroni–Dunn post-hoc test with RMCV as the control classifier vs. all other BNCs, RMCV was superior to all of them with a significance level of $p < 0.05$, with the exception of BNC-MDL, for which $p < 0.1$. According to a Friedman test for all of the classifiers, RMCV was ranked the highest, and the null hypothesis that all algorithms are the same had been rejected with high confidence. However, due to the large number of models compared (fourteen), a relatively large difference of average ranks due to the Friedman test was required by the Bonferroni–Dunn test to indicate a significant difference, and RMCV was found to be significantly superior (with $p < 0.1$) to

Table 2 Average and std of classifiers' accuracy over 22 UCI [71] databases. Bayesian network initial structures and SVM kernel types appear in brackets. Bold/italic fonts are used, respectively, for the best/worst classifiers

	RMCV (NBC)	MDL (NBC)	MDL (Empty)	K2 (NBC)	K2 (Empty)	BNC-MDL	BNC-2P
Average	**84.8**	81.5	80.9	81.0	80.9	81.4	*76.7*
Std	(3.8)	(4.1)	(4.1)	(4.3)	(4.4)	(3.8)	(4.7)
	TAN	NBC	CART	NN	SVM (Linear)	SVM (Polynomial)	SVM (Gaussian)
Average	82.5	81.3	83.8	83.6	82.4	77.3	81.8
Std	(3.9)	(3.7)	(3.8)	(4.0)	(3.6)	(3.9)	(3.9)

only four other classifiers [MDL (empty), K2 (NBC), K2 (empty), and BNC-2P]. 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 CART, NN, and SVM (Linear/Gaussian), RMCV was not significantly superior with $p \in [0.147, 0.610]$. This means that RMCV, CART, NN, and SVM (Linear/Gaussian) were comparable in terms of classification accuracy. That study also showed that an optimized version of RMCV is faster than all unrestricted BNCs and comparable to the neural network with respect to runtime.

These comparative studies, like most studies cited here and elsewhere, usually use ready-to-use databases of what are called "real-world" problems, taken mainly from the UCI [71] and similar repositories. However, the selection of the sample databases used in each comparative study is neither complete nor standardized nor, needless to say, is it representative of real problems[8] [84]. To demonstrate BNC performance on a wider range of complexities presented in actual real-world data sets, we report in Sect. 3.4.2 on six studies with five such data sets.

3.4.2 Evaluation of BNCs Using Authentic Data Sets

The data set in the first study contains data extracted from fluorescence in situ hybridization (FISH) microscope images used in genetic abnormality inspection [55, 56, 85]. The various instances represent red and green signals, corresponding to Down and Patau syndromes, respectively. Each of the two signals can be either "real," where the syndrome can be observed in the image, or an "artifact" due to refraction and scattering of the fluorescence signal in the microscope optics, where the syndrome is not actually present. Each of 3,144 instances of signal images is represented by twelve features of the signal that are characterized by five types: size (area), shape (eccentricity, measuring the signal's similarity to an ellipse), intensity (different features measuring total, average, and standard deviation in the red-green-blue (RGB) channels), hue (maximum, average, standard deviation, difference between the maximum and average normalized by the average), and eigenfeatures, corresponding to the red and green intensity components of the signal. The class label takes one of the following values: Real Red (RR) (551 signals), Artificial Red (AR) (1,224), Real Green (RG) (594), or Artificial Green (AG) (775), forming a four-class classification problem.

Table 3 shows that the relatively high accuracy of the NBC in classifying FISH signals is attributed to its accuracy in classifying the Real Red signals, for which the variables are mostly independent given the class value, as assumed by the NBC. Therefore, we also attribute the high accuracies of the other NBC-based BNCs (K2, MDL) to the NBC initialization. While the K2-based classifiers are

[8] Check [84] that shows that the nine most popular UCI databases have between 100,000 and 400,000 hits, but the comparative studies using them provide no explanation to why these databases have been selected.

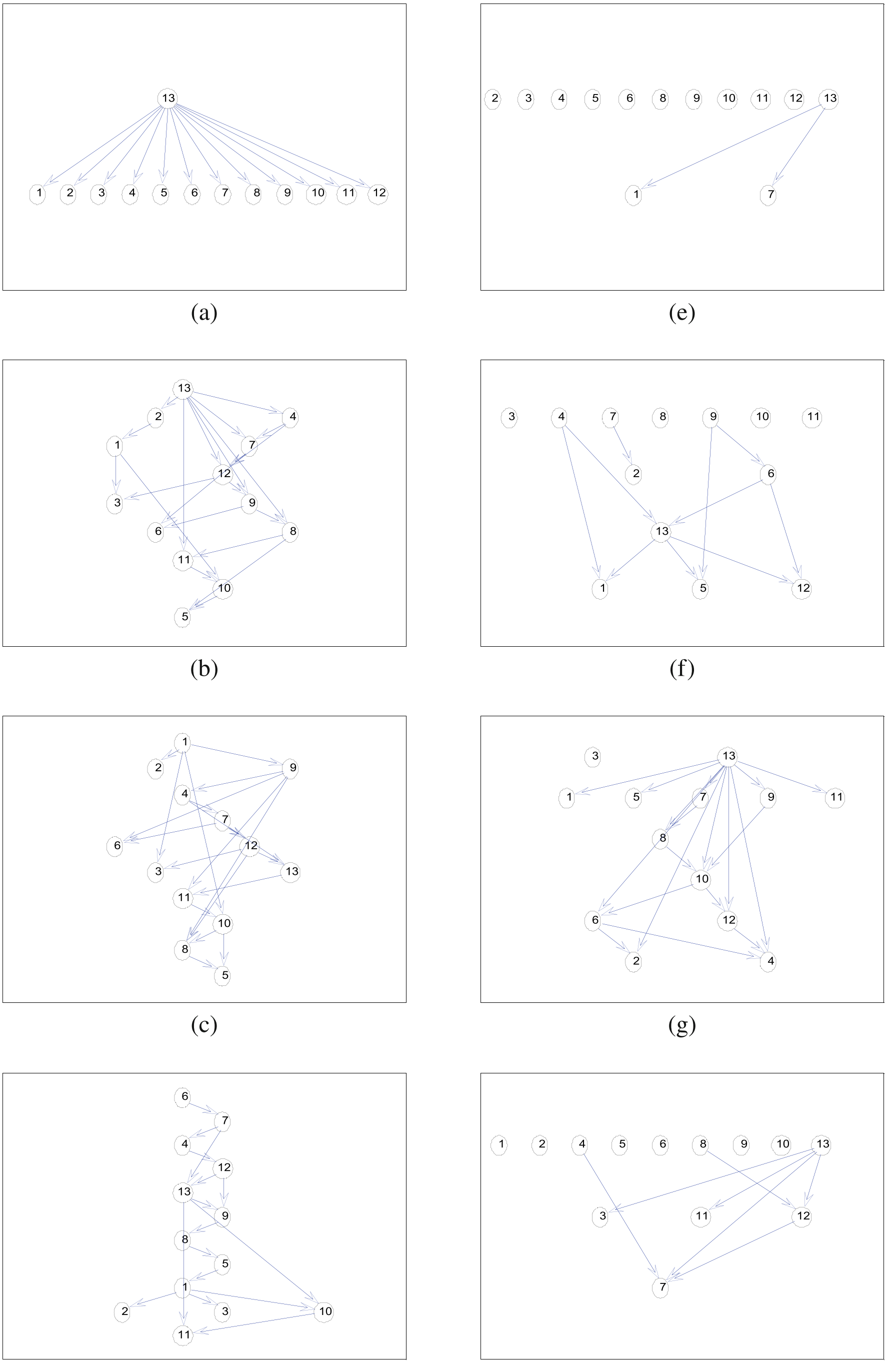
Table 3 Accuracies (%) of BNCs for four signal classification tasks for the FISH data set

Algorithm	RR	AR	RG	AG	Total
NBC [54]	*88.0*	76.9	81.1	74.4	79.0
K2 (NBC) [4]	85.8	77.6	80.1	*75.7*	79.2
K2 (Empty) [4]	75.7	72.1	81.7	71.6	74.4
MDL (NBC) [74]	85.3	80.7	**85.0**	65.8	78.7
MDL (Empty) [74]	80.2	81.1	82.0	74.3	79.4
BNC-MDL [11]	80.1	79.5	*84.0*	67.0	77.5
BNC-2P [11]	82.9	78.9	74.2	72.4	77.1
RMCV (NBC) [17]	**88.2**	*81.5*	83.7	75.1	*81.5*
RMCV (Empty) [17]	83.7	**84.2**	82.8	**76.9**	**82.1**

[a] Bold and italic fonts are for most and second-most accurate classifiers for a task, respectively

highly dependent on the K2 algorithm initialization, the MDL BNCs and BNC-MDL/2P are not. In all but one of the four classification tasks, at least one of the RMCV BNCs is the most or second-most accurate classifier, which makes these two BNCs the most accurate on average. Figure 4 shows (for one arbitrary fold of the CV5 experiment) that while RMCV (NBC) reveals a similar structure to that of the NBC, RMCV (Empty) shows a different structure, as six of the twelve attributes are disconnected. This aggressive but educated feature selection of RMCV (Empty) contributes not only to the best classification results (Table 3) but also to an interpretable model relying only on three types of signal features (intensity, hue, and eigenfeatures), compared to the other too dense or too sparse less informative models (Fig. 4).

The second original data set, a flash memory semiconductor manufacturing data set, consists of 362 instances of wafer lots represented by 35 tool variables describing an ion-implementation process that is part of wafer manufacturing. The data are highly imbalanced, where 30 of the lots are faulty and 332 are normal. Table 4 shows that most BNC algorithms have been fooled by the imbalance in the data, wrongly classifying most or all of the faulty lots as normal. This is the case with K2 (but also with MDL and BNC), which are almost perfect in classifying normal lots but fail completely with faulty lots. The only two algorithms that have a relatively reasonable accuracy in classifying faulty lots in spite of the imbalance are the NBC and RMCV. The 36.7% accuracy of the NBC for faulty lots came at the expense of its accuracy in normal lots (93.1%), which is the lowest of all algorithms, positioning this classifier as the poorest in total. The RMCV, and particularly the “weighted” version, which penalizes errors according to the classes’ prior probability ratio, balances its performance between the classes, positioning this classifier as the best in total and most even. Of particular note in Fig. 5 are the graphs of the NBC (best for faulty lots), K2 (NBC) (best for normal lots), and RMCV (Empty, weighted) (best in total). We attribute the high and balanced performance of the RMCV to the drastic dimensionality reduction it performed, identifying only a few significant variables in its Markov blanket—enough to accurately classify both classes but without over-fitting any of them at the expense of the other.



(a)

(e)

(b)

(f)

(c)

(g)

(d)

(h)

Fig. 4 BNC structures learned for one CV5 fold for the FISH data set. 13 is the class node. (**a**) NBC. (**b**) K2 (NBC). (**c**) K2 (Empty). (**d**) MDL (Empty). (**e**) BNC-MDL. (**f**) BNC-2P. (**g**) RMCV (NBC). (**h**) RMCV (Empty)

Table 4 Accuracies (%) of BNCs in detecting either normal or defective lots of wafers of an imbalanced semiconductor manufacturing data set

Algorithm	Normal	Faulty	Total
NBC [54]	93.1	**36.7**	88.4
K2 (NBC) [4]	**99.7**	0.0	**91.4**
K2 (Empty) [4]	*99.4*	0.0	*91.2*
MDL (NBC/Empty) [74]	98.2	10.0	90.9
BNC-MDL [11]	99.1	0.0	90.9
BNC-2P [11]	98.2	13.3	*91.2*
RMCV (NBC) [17]	96.7	20.0	90.3
RMCV (Empty) [17]	97.0	13.3	90.1
RMCV (NBC, weighted) [17]	97.9	13.3	90.9
RMCV (Empty, weighted) [17]	97.0	*30.0*	**91.4**

[a] Bold and italic fonts are for most and second-most accurate classifiers for a task, respectively

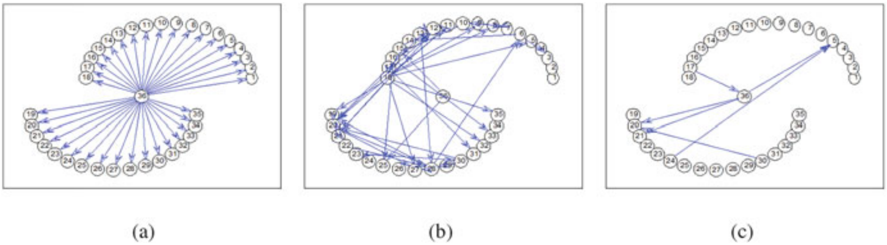


Fig. 5 Three BNCs learned for one CV5 fold for the semiconductor manufacturing data set. 36 is the class node. (**a**) NBC. (**b**) K2 (NBC). (**c**) RMCV

The data set in the third study contains medical diagnosis data extracted from visuomotor measurements of people who have been diagnosed with Parkinson's disease (PD) or essential tremor (ET) versus healthy controls. PD and ET have very similar symptoms, but ET, unlike PD, is related to long life expectancy. The data set used to predict PD has 164 instances, relatively balanced among the three classes (55 PD, 51 ET, 58 controls). Each is represented using the person's age, their worst affected hand, and fourteen visuomotor features measured for the persons' two hands. The MDL (NBC or Empty) BNC provided the highest accuracy on PD patients, and the K2 (NBC or Empty) BNC supplied the best accuracy on the healthy controls, but both classifiers failed to classify any ET patient. This failure was due to learned structures where either the class node was not connected to the graph (making classification based on the a priori probabilities) or the Markov blanket of the class node was relatively empty of nodes with which to make accurate predictions. The BNC-2P provided reasonable performances on all three classes (where the BNC-MDL was similar to the MDL and K2 classifiers). However, its scores were lower than those of the RMCV (NBC or Empty), which was equally accurate for all three classes and the most accurate classifier. The NBC, dispensing with structure learning, was the classifier least affected by the small data set; it was thus the second-most accurate classifier after the RMCV, although it manifested a non-informative graph. The RMCV's structure was not as dense and uninformative as that of the NBC but was also not as empty as those of the MDL, K2, and BNC-MDL/2P. Such a structure conveys knowledge representation by interrelating visuomotor measurements with clinical characteristics of the patients, which also improves the accuracy in classifying PD patients, as distinct from ET patients and from healthy people.

For the fourth data set, the RMCV was applied [46] to identify the most influential variables in predicting and explaining functional deterioration (e.g., walking, writing, climbing stairs, and speaking) of five levels from a large clinical-trial database of ALS patients [86]. For each variable representing patient functionality, the RMCV selected to include in the variable's MB only those variables from the tens available for the algorithm that contribute to accurate prediction of this functionality. For example, the MB of the climbing stairs variable shows connections between the ability to climb stairs and lab test results that are related to the body muscle metabolism (e.g., glucose, creatinine, phosphorus, and alkaline phosphatase), forced vital capacity (FVC), the total amount of air a person can exhale during a forced breath, which is also related to the person's physical capability, and the disease onset site, whether in the bulbar or limbs (and then the patients are limited in climbing stairs sooner in their progression). It could also be possible to distinguish mild from severe ALS patients by the different combinations of values these MB variables take for the two groups of patients. For example, severe swallowing functionality in patients who their onset was in the bulbar and their FVC capability is low is between 8 and 35 times more frequent than in patients who their onset was in the limbs and their FVC capability is between moderate and high, respectively [46].

In the fifth study [87], when predicting young driver (YD) fatalities in motorcycle accidents, the RMCV classifier identified key factors in the class variable's MB that distinguish between minor, severe, and fatal accidents. Some of the main factors were the accident type (inexperienced YDs are more likely to lose control over their motorcycle and crash into inanimate objects, skid, or turn over with usually deadly results), road speed limit (accidents on roads where the speed limit is high tend to be fatal or severe), gender (in all fatal accidents in the data, men were the drivers, and in general, the victims of severe and fatal YD accidents are three times more likely to be men), age at time of accident (most accidents of older YDs ($\geq$22) are fatal, since they drive heavier motorcycles than younger drivers), and motorcycle type (a YD accident that involves a heavy, $\geq 400cc$, motorcycle is eight times more likely to be deadly than for a lighter one).

In the sixth and last study we report here [88], the RMCV was modified to deal with class-imbalance ordinal classification problems and to provide information about the distribution of misclassifications and about the sensitivity to error severity (distinguishing between misclassification of class X as class Y or as class Z). The modified RMCV achieved superior average accuracy over the CART, NN, and SVM using 23 synthetic and 17 UCI databases, and superior average accuracy over the RF using the synthetic databases but inferior average accuracy using the UCI databases.[9] In addition, using data of three real problems (the above ALS [46] and YD motorcycle accidents [87], as well as missed due date—no delay, 3–5 days of delay, and more than 5 days of delay—of a product in Teleco orders), the modified RMCV classifier showed confusion matrices with errors that are the most balanced over the classes compared with the NN and SVM competitors, contributing to its superior accuracy.

Still, a further step is needed to allow examples such as those in this section—in genetic abnormality inspection, semiconductor manufacturing, Parkinson's disease diagnosis, ALS prediction and explanation, and young driver motorcycle accidents analysis—convincingly demonstrate domain experts interpretability and explainability. This step involves human–machine interaction, and we will return to this issue in the conclusion section.

4 Beyond the BNC–Causal–Temporal Classifiers

Let us consider progression of a neurodegenerative disease such as ALS or Alzheimer's or a chronic disease such as type 2 diabetes, for which we wish to predict a future state for patients. In other words, to predict a diagnosis using symptoms such as clinical markers and lab test results collected routinely over time. We may further consider a left-to-right model with a state index that either decreases or stays the same and thereby represents progression of such diseases. Figure 6a shows a two-slice graph that represents a medical domain in which there is a cause to a disease. We wish to predict the current disease state based on that cause and the previous disease state, where both the cause and disease state are unknown latent

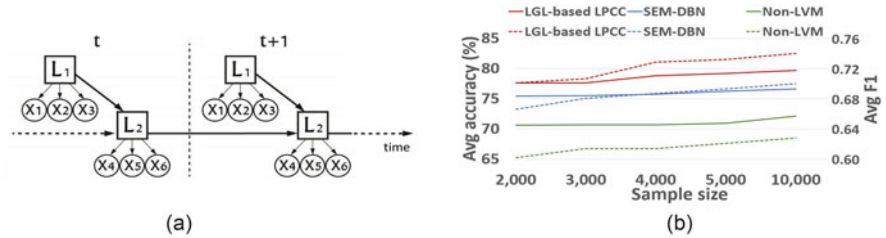


Fig. 6 (**a**) Artificial temporal graph with three OVs per each of two LVs also making a collider per slice. (**b**) Classification accuracy (solid line) and F1-measure (dashed line) for increasing sample sizes of the data sets sampled from the graph in (**a**)

[9] The modified RMCV score balances the 0/1 loss (accuracy) function with the mutual information between predictions and true labels and with the severity of misclassifications [88, 89]. Synthetic databases had combinations of different numbers of classes and instances and degrees of imbalance.

variables (LVs) L_1 and L_2, respectively, each of which has three observed variables (OVs), which are proxies for disease symptoms, together $X_1 - X_6$.

Since a future patient disease state is unknown and thus cannot be modeled and predicted, we can instead predict the value of a symptom that hints at the disease state using the values of other symptoms. However, if there is no a priori information that a specific symptom is the most predictive of the disease and we want to eliminate uncertainty, we might repeat and predict each symptom in turn and average the prediction performance over all predicted symptoms. While this strategy may sound very practical, it can neither discover cause–disease relations nor their evolution over time in the domain and thereby cannot contribute to understanding the disease or its progression, assist in drug development, or enable a better cure for the disease.

Therefore, for causal discovery and cause–disease relations monitoring, we may wish to use graphical models such as dynamic BNs and, especially, latent variable models (LVMs) [2, 44, 90, 91, 92]. To meet this challenge, we propose learning a causal latent model in each time slot locally. Then, we suggest local-to-global learning over time slices, based on probabilistic scoring and temporal reasoning to transfer the local graphs into a non-stationary latent dynamic BN (DBN) with intra- and inter-slice edges showing causal interrelationships among latent variables and between latent variables and observed variables. This is performed based on the learning pairwise cluster comparison (LPCC) algorithm [91, 92] using the LPCC-based local-to-global (LGL) algorithm [93] to learn a temporal LVM.

The LPCC-based LGL algorithm was evaluated on data sets that were sampled from the artificial temporal BN in Fig. 6a for varying sequence sizes, $4 \leq T \leq 15$, i.e., a record is $(|\mathbf{O}| \times T)$-dimensional, and $\mathbf{O}$ is the set of observed variables. The sample size was $D = \{2{,}000, 3{,}000, 4{,}000, 5{,}000, 10{,}000\}$. The cardinality of all variables was set to four, where the probability that an observed variable takes the same value as does its parent latent variable is 0.8, and the probability that it takes any other value is equally distributed (i.e., a 0.2 "noise" level was evenly distributed among the other values). These probabilities are the same for all T values to guarantee stationarity. Reported results are averaged over ten data permutations for each value combination of T and D.

We empirically compared the LPCC-based LGL algorithm with the state-of-the-art structural expectation maximization (SEM) algorithm [94, 95] that learns a latent DBN, i.e., the SEM-DBN algorithm. This algorithm uses an S&S procedure to find the best fitted model from data, although not necessarily a causal one. It also requires the user to specify the number of LVs and their cardinalities beforehand. For fairness, we limited it to: (1) search over the (smaller) space of pure measurement models (PMMs),[10] and even initialized it with a random PMM, and (2) not to direct an edge from t to $t - 1$.

[10] A DAG over sets of observed variables, latent variables, and edges is a measurement model if a latent variable is a parent of at least one observed variable, an observed variable is a child of at least one latent variable, and none of the observed variables is a parent of any latent variable. A measurement model is called a pure measurement model (PMM) if each observed variable has a single parent and that parent is a latent variable [90].

For comparison, assuming the absence of an LVM, we also compared the classification accuracy of the LPCC-based LGL algorithm and that of the SEM-DBN with the average accuracy of six RF classifiers, each classifying each of the six OVs of the graph in Fig. 6a in the Tth (last) slice (acting as the classification node, where all the remaining OVs in all slices are predictors). By taking the average over the six classifiers, we avoid any preference in the classification (as above). That is, we compared this straightforward classification approach (denoted as a non-LVM) to that based on identifying latent variables and their values using either the LPCC-based LGL or SEM-DBN algorithms, and using the values of the learned latent variables to perform the classification. Thereby, we plan to demonstrate the contribution of learning a temporal LVM in a classification task compared with ignoring the existence of latent variables. We repeated this comparison for each combination of sequence size, sample size, and data permutation.

In total, we trained $5 \times 8 \times 10 \times 6 \times 3 = 7{,}200$ classifiers for five sample sizes, eight sequence sizes, ten data permutations, six classifiers (classification nodes), and three models (LGL/SEM-DBN/no-LVM). We used 80% of the instances of each data set for training and 20% for testing. Figure 6b shows the accuracy and F1-measure averaged over all data permutations, sequence sizes, and observed variables for different sample sizes of the three models. It shows that the LPCC-based LGL algorithm achieves the best classification performance (significantly superior to the others). It reaches the highest possible accuracy of 80% since the noise was originally set at 20%, i.e., even if the algorithm learns the true classification rule, in 20% of cases it will be wrong. This experiment not only allows us to appreciate the importance of learning an LVM in general but also specifically in classification tasks, since latent mechanisms always exist.

Finally, the LPCC-based LGL algorithm was applied to the ALS open-source *PROACT ALS data set* [86] that consists of 3,171 patients with 22,089 clinic visits, from which we derived a subset of 2,590 patients who had at least four visits, each consecutive two are up to six months apart. A visit consists of lab test results and clinical variables describing patient physical functioning, e.g., in walking, writing, and speech. The LGL-based LPCC learned a graph using four latent variables (Fig. 7) demonstrating patient functioning: bulbar functionality ($L1$) indicated mainly by speech (Sp), salivation (Sv), and swallowing (Sw); gross-motor functionality ($L2$) indicated by walking (Wa) and climbing stairs (Cs); fine-motor functionality ($L4$) indicated by dressing (Dr), writing (Wr), and cutting food (Cu); and full body functionality ($L3$) indicated by turning (Tr) in bed, respiratory (Re) ability, FVC, and two lab tests: CK and chloride (Ch) (also found correlated in [46, 96]). The three intra-slice edges represent the natural connections between the bulbar and gross-motor, gross-motor and full body, and fine-motor and full body functionalities. The inter-slice edges between bulbar and full body to themselves complete the temporal–causal reasoning that resembles medical categorization and convention.

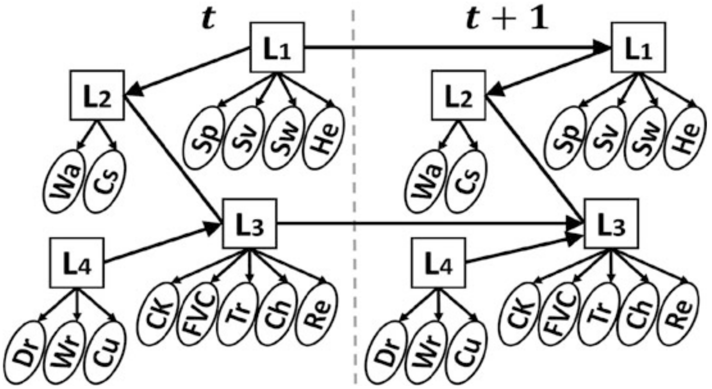


Fig. 7 A temporal LVM learned by the LPCC-based LGL for the ALS data set

5 Conclusion and Discussion

Machine learning models are usually not used in an isolated way but are embedded in some process or product and interact with people. Thus, a more flexible yet holistic view of the entire process, from data collection to the final consumption of the explained prediction, is needed. This includes considering both how to explain predictions to individuals with diverse knowledge and backgrounds and the need for interpretability on the level of an institution or society in general [28, 32]. This is especially required when moving from sandbox studies of benchmark data sets to actual real-world problems [97, 98].

Researchers and practitioners seek to make their algorithms more understandable by focusing, for example, on explicit explanation of decisions and actions to a human observer. However, this focus should range beyond the ML researchers' intuition of what constitutes a "good" explanation and build on existing research from philosophy, cognitive psychology/science, and social psychology, disciplines that grapple with these topics, and study how people define, generate, select, evaluate, and present explanations [37].

This chapter encourages more exploration and exploitation of human-understandable causal models, such as the BNCs, of the operation of ML and especially DNN paradigms. BNC models will allow better introduction and use of causal discovery, interventions, and queries as effective tools to promote explainability and interpretability in developing and applying ML. They have natural interpretability (ability to understand the results) and explainability (ability to explain the results). Unlike ensemble classifiers, the BNC does not provide a list of "important" variables ranked by a statistical, discriminability, or information measure (where the variable "importance" value can differ even by a fraction of a percent and/or due to calibration issues), but also a feature-selection mechanism through variables' Markov blankets and hierarchies of connections along with paths of influence and inference mechanisms that together expose causal relations in the domain.

A main challenge is that the BN/BNC is not suitable for processing high-dimensional and/or non-semantic data that exist in domains based, for example, on images, text, and speech. Current BN learning algorithms are limited to small to medium domains usually of discrete variables. While it is hard to believe that this will be changed in the near future, it is more reasonable to expect BNs to be applied to processed, projected, or embedded DNN representations of high-dimensional domain input. DNN projections and embeddings of high-dimensional data already reflect semantic evidence that can more easily be extracted, analyzed, and exploited by the BN compared, for example, to the same data re-processed or re-embedded by more layers of the DNN. Effective BN-based tools to promote explainability and interpretability can, for example, allow the user to understand the chain of causal effects from DNN input, to low-level features of the domain, to high-level human-understandable concepts, to DNN outputs [99]. Such a capability is a powerful tool for debugging, understanding bias, and ensuring the safe operation of AI systems. Additionally, these tools may extract low-dimensional concepts from DNNs to generate a human-understandable "vocabulary" and learn a BN that relates the DNN's inputs to the concepts, and the concepts to the DNN's outputs [43]. Other probabilistic models, such as the hidden Markov model (for multilayer perceptron networks) and Gaussian mixture model (for convolutional neural networks), may extract activity patterns of the network hidden layers, where transition probabilities between clusters (mixture components) in consecutive modeled layers may be estimated from the data [42]. Then nodes and paths relevant for network prediction can be chosen, connected, and visualized as an inference graph. This graph is useful for understanding the general inference process for a class, as well as explaining decisions the network makes regarding specific images. Also, a scalable graphical-model framework [100] was shown to aid human understanding and reasoning by providing criticism to explain what is not captured by the examples and improving the interpretability of complex data distributions. In addition, it was demonstrated [101] that when casting the problem of learning the connectivity of a DNN as a BN structure learning problem according to the recursive autonomy identification notion [27], the resulting DNN structure encodes independencies in the input distribution hierarchically, where lower-order independencies are encoded in deeper layers. Tools and mechanisms such as these may also facilitate the struggle of DNN researchers to interpret hidden layer activity in order to understand the general inference process of a classifier, explain decisions the network makes, and indicate the relevance of specific inputs to the network output.

To conclude, returning to Footnote 4, we believe that transparency and accountability are hard to achieve by BOTH human and machine decision-making, and thus we call for more human involvement and intervention for confirmation and validation in AI-driven systems. This could be accomplished by the development of graphical user interface tools soliciting, fostering, and supporting human–machine interaction and bi-directional communication by which, on the one hand, users' inquiries will manipulate and extend the learned BNC model to better address these and further inquiries, and, on the other hand, the tools will inspire users' curiosity to further interrogate and explore the model to enrich their understanding of the domain beyond what they expected to achieve when running the tool. Bayesian network classifiers can confer transparency on DNNs and other ML tools, enable interpretability and explainability, and empower humans—ML experts and non-ML users alike—to understand but also to affect the results of these tools, all of which will increase trust and trustworthiness in ML to deliver true "explainable AI." To maximize the tremendous impact that ML is having on all aspects of our lives, and enhance trustworthiness in AI, let us embrace the opportunities BNCs are bringing us.

Acknowledgements The author thanks Nuaman Asbeh, Idit Bernstein, Jonathan Gordon, Yaniv Gurwicz, Dan Halbersberg, Roy Kelner, Yael Konforti, Lev Koushnir, Roy Malka, Yair Meidan, Alon Shpigler, Maydan Wienreb, and Raanan Yehezkel (Rohekar) for the use of their dissertations' results in this chapter, and Aviad Raz and Aharon Bar-Hillel for comments and feedback that helped improve this chapter.

References

1. Pearl, J.: Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan Kaufmann, San Francisco (1988)
2. Spirtes, P., Glymour, C., Scheines, R.: Causality, prediction and search (2nd edition). MIT Press, Cambridge, MA (2000)
3. Lauritzen, S.L., Spiegelhalter, D.J.: Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society, Series B, **50**, 157–224 (1988)
4. Cooper, G.F., Herskovits, E.: A Bayesian method for the induction of probabilistic networks from data. Machine Learning, **9**, 309–347 (1992)
5. Heckerman,D.: A tutorial on learning with Bayesian networks. Microsoft Research Technical Report MSR-TR-95-06 March 1995 (revised November 1996)
6. Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian networks: the combination of knowledge and statistical data. Machine Learning, **20**, 197–243 (1995)
7. Meek, C.: Strong completeness and faithfulness in Bayesian networks. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, 411–418 (1995)
8. Chickering, D.M.: Optimal structure identification with greedy search. Journal of Machine Learning Research, **3**, 507–554 (2002)
9. Friedman, N., Geiger, D., Goldszmidt, M: Bayesian network classifiers. Machine Learning, **29**, 131–163 (1997)
10. Kontkanen, P., Myllymaki, P., Sliander, T., Tirri, H.: On supervised selection of Bayesian networks. Proceedings of the Fifteenth International Conference on Uncertainty in Artificial Intelligence, 334–342 (1999)
11. Grossman, D., Domingos, P.: Learning Bayesian network classifiers by maximizing conditional likelihood. Proceedings of the 21st International Conference on Machine Learning, 361–368 (2004)
12. Guo, Y., Greiner, R.: Discriminative model selection for belief net structures. Proceedings of the AAAI, 770–776 (2005)
13. Pernkopf, F.: Bayesian network classifiers versus selective *k*-NN classifier. Pattern Recognition, **38**, 1–10 (2005)

14. Roos, T., Wettig, H., Grünwald, P., Myllymäki, P., Tirri, H.: On discriminative Bayesian network classifiers and logistic regression. Machine Learning, **59**, 267–296 (2005)
15. Acid, S., Campos, L., Castellano, J.: Learning Bayesian network classifiers: Searching in a space of partially directed acyclic graphs. Machine Learning, **59**, 213–235 (2005)
16. Pernkopf, F., Bilmes, J.A.: Efficient heuristics for discriminative structure learning of Bayesian network classifiers. Journal of Machine Learning Research, **11**, 2323–2360 (2010)
17. Kelner, R., Lerner, B.: Learning Bayesian network classifiers by risk minimization. International Journal of Approximate Reasoning **53**, 248–272 (2012)
18. Cheng, J., Greiner, R.: Comparing Bayesian network classifiers. Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, 101–108 (1999)
19. Akaike, H.: Information theory and an extension of the maximum likelihood principle. In: Parzen E., Tanabe K., Kitagawa G. (eds) Selected Papers of Hirotugu Akaike. Springer Series in Statistics (Perspectives in Statistics). Springer, New York, NY (1998)
20. Schwarz, G.: Estimating the dimension of a model. The Annals of Statistics, **6**, 461–464 (1978)
21. Kullback, S., Leibler, R.: On information and sufficiency. The Annals of Mathematical Statistics, **22**, 79–86 (1951)
22. Martínez, A.M., Webb, G.I., Chen, S., Zaidi, N.A.: Scalable learning of Bayesian network classifiers. Journal of Machine Learning Research, **17**, 1–35 (2016)
23. Jing, Y., Pavlović, V. Rehg, J.M.: Boosted Bayesian network classifiers. Machine Learning, **73**, 155–184 (2008)
24. Carvalho A.M., Oliveira A.L., Sagot MF.: Efficient learning of Bayesian network classifiers. In: Orgun M.A., Thornton J. (eds) AI 2007: Advances in Artificial Intelligence. AI 2007. Lecture Notes in Computer Science, vol 4830. Springer, Berlin, Heidelberg (2007)
25. Madden, M.G.: On the classification performance of TAN and general Bayesian networks. Knowledge-Based Systems, **22**, 489–495 (2009)
26. Keogh, E., Pazzani, M.: Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches. Proceedings of the International Workshop on Artificial Intelligence and Statistics, pp. 225–230 (1999)
27. Yehezkel, R., Lerner, B.: Bayesian network structure learning by recursive autonomy identification. Journal of Machine Learning Research, **10**, 1527–1570 (2009)
28. Molnar, C., Casalicchio, G., Bischl, B.: Interpretable machine learning–A brief history, state-of-the-art and challenges. arXiv:2010.09337 (2020)
29. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should I trust you?": Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1135–1144 (2016)
30. Molnar, C., König, G., Herbinger, J., Freiesleben, T., Dandl, S., Scholbeck, C.A., Casalicchio, G., Grosse-Wentrup, M., Bischl, B.: Pitfalls to avoid when interpreting machine learning models. ICML 2020 Workshop XXAI: Extending Explainable AI Beyond Deep Models and Classifiers (2020)
31. Lo Piano, S.: Ethical principles in machine learning and artificial intelligence: cases from the field and possible ways forward. Humanities and Social Sciences Communications, **7**, 9 (2020)
32. Spiegelhalter, D.: Should we trust algorithms? Harvard Data Science Review, 2(1) (2020)
33. Brundage, M., Avin, S., Wang, J., Belfield, H., Krueger, G., Hadfield, G., et al.: Toward trustworthy AI development: Mechanisms for supporting verifiable claims. arXiv e-print arXiv:2004.07213 (2020)
34. Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., Zhong, C.: Interpretable machine learning: Fundamental principles and 10 grand challenges, **CoRR**, abs/2103.11251, https://arxiv.org/abs/2103.11251 (2021)
35. Kroll, J.A., Huey, J., Barocas, S., Felten, E.W., Reidenberg, J.R., Robinson, D.G., Yu, H.: Accountable algorithms, 165 University of Pennsylvania Law Review 633 (2017)
36. Ashoori, M., Weisz, J.D.: In AI we trust? Factors that influence trustworthiness of AI-infused decision-making processes. arXiv e-print arXiv:1912.02675 (2019)

37. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence, **267**, 1–38 (2019)
38. Lundberg, S.M., Lee, S-I.: A unified approach to interpreting model predictions. In: Guyon, I., et al. (eds) Proceedings of the Advances in Neural Information Processing Systems 30, 4765–4774 (2017)
39. Olah, C., Mordvintsev, A., Schubert, L.: Feature visualization. Distill, **2** (2017)
40. Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., Batra, D.: Grad-CAM: Why did you say that? Visual explanations from deep networks via gradient-based localization, CoRR, **abs/1610.02391**, http://arxiv.org/abs/1610.02391 (2016)
41. Binder A., Bach S., Montavon G., Müller KR., Samek W.: Layer-wise relevance propagation for deep neural network architectures. In: Kim K., Joukov N. (eds) Information Science and Applications (ICISA). Lecture Notes in Electrical Engineering, 376. Springer, Singapore (2016)
42. Konforti, Y., Shpigler, A., Lerner, B., Bar Hillel, A.: SIGN: Statistical inference graphs based on probabilistic network activity interpretation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **45**(3), 3783–3797 (2023)
43. Harradon, M., Druce, J., Ruttenberg, B.: Causal learning and explanation of deep neural networks via autoencoded activations, arXiv:1802.00541 (2018)
44. Pearl, J.: Causality: Models, reasoning, and inference (2nd edition), Cambridge University Press, New York, NY (2009)
45. Peters, J., Janzing, D., Schölkopf, B.: Elements of causal inference - Foundation and learning algorithms. The MIT Press (2017)
46. Gordon, J., Lerner, B.: Insights into ALS from a machine learning perspective. Journal of Clinical Medicine, **8**, 1578 (2019)
47. Drugan, M.M., Wiering, M.A.: Feature selection for Bayesian network classifiers using the MDL-FS score. International Journal of Approximate Reasoning, **51**, 695–717 (2010)
48. dos Santos, E.B., Hruschka Jr., E.R., Hruschka, E.R., Ebecken, N.F.F.: Bayesian network classifiers: Beyond classification accuracy. Intelligent Data Analysis, **15**, 279–298 (2011)
49. Bielza, C., Larrañaga, P.: Discrete Bayesian network classifiers: A survey. ACM Computing Surveys, **47** 1–43 (2014)
50. Aliferis, C.F., Tsamardinos, I., Statnikov, A.: HITON: a novel Markov blanket algorithm for optimal variable selection. AMIA Annual Symposium Proceedings, 21–25 (2003)
51. Tan, Y., Liu, Z.: Feature selection and prediction with a Markov blanket structure learning algorithm. BMC Bioinformatics, **14** (Suppl 17), A3 (2013)
52. Antal, P., Millinghoffer, A., Hullám, G., Szalai, C,. Falus, A.: A Bayesian view of challenges in feature selection: Feature aggregation, multiple targets, redundancy and interaction. In: Saeys, Y., Liu, H., Inza, I., Wehenkel, L., Pee, Y. (eds) Proceedings of the Workshop on New Challenges for Feature Selection in Data Mining and Knowledge Discovery at ECML/PKDD 2008, Proceedings of Machine Learning Research, 74–89 (2008)
53. Shih, A., Choi, A., Darwiche, A.: A Symbolic approach to explaining Bayesian network classifiers. Proceedings of the 27th International Joint Conference on Artificial Intelligence, 5103–5111 (2018)
54. Langley, P., Iba, W., Thompson, K.: An analysis of Bayesian classifiers. Proceedings of the Tenth National Conference on Artificial Intelligence, 223–228 (1992)
55. Lerner, B.: Bayesian fluorescence in situ hybridisation signal classification. Artificial Intelligence in Medicine, **30**, 301–316 (2004)
56. Lerner, B., Lawrence N. D.: A comparison of state-of-the-art classification techniques with application to cytogenetics. Neural Computing & Applications, **10**, 39–47 (2001).
57. Lerner, B., Yeshaya, J., Koushnir, L.: On the classification of a small imbalanced cytogenetic image database. IEEE-ACM Transactions on Computational Biology and Bioinformatics, **4**, 204–215 (2007)
58. Lerner, B., Koushnir, L., Yeshaya, J.: Segmentation and classification of dot and non-dot-like fluorescence in-situ hybridization signals for automated detection of cytogenetic numerical abnormalities. IEEE Transactions on Information Technology in Biomedicine, **11**, 443–449 (2007)

59. Webb, G.I., Boughton, J.R., Wang, Z.: Not so naive Bayes: Aggregating one dependence estimators. Machine Learning, **58**, 5–24 (2005)
60. Zheng, Z., Webb, G.I.: Lazy learning of Bayesian rules. Machine Learning, **41**, 53–84 (2000)
61. Xie, Z., Hsu, W., Liu, Z., Lee, M.: A selective neighborhood based naive Bayes for lazy learning. In: Chen, M.-S., Yu, P.S., Liu, B. (eds.) PAKDD 2002. LNCS (LNAI), **2336**, 104–114. Springer, Heidelberg (2002)
62. Frank, E., Hall, M., Pfahringer, B.: Locally weighted naive Bayes. Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), 249–256. Morgan Kaufmann Publishers, Seattle (2003)
63. Chan, H., Darwiche, A.: Reasoning about Bayesian network classifiers, **CoRR**, abs/1212.2470, http://arxiv.org/abs/1212.2470 (2012)
64. Meidan, Y., Lerner, B., Rabinowitz, G., Hassoun, M.: Cycle-time key factor identification and prediction in semiconductor manufacturing using machine learning and data mining. IEEE Transactions on Semiconductor Manufacturing, **24**, 237–248 (2011)
65. Chow, C., Liu, C.: Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory, **14**, 462–467 (1968)
66. Lerner, B., Malka, R.: Investigation of the K2 algorithm in learning Bayesian network classifiers. Applied Artificial Intelligence, **25**, 74–96 (2011)
67. Geiger, D., Heckerman, D.: Knowledge representation and inference in similarity networks and Bayesian multinets. Artificial Intelligence, **82**, 45–74 (1996)
68. Bilmes, J.: Dynamic Bayesian multinets. Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI), Morgan Kaufmann Publishers (2000)
69. Gurwicz, Y., Lerner, B.: Bayesian class-matched multinet classifier. SSPR/SPR, ser. Lecture Notes in Computer Science, D. Y. Yeung, J. T. Kwok, A. L. N. Fred, F. Roli, and D. de Ridder, Eds., Springer, Vol. 4109, 145–153 (2006)
70. Pena, J.M., Lozano, J.A., Larranaga, P.: Learning recursive Bayesian multinets for data clustering by means of constructive induction. Machine Learning, **47**, 63–89 (2002)
71. Dheeru, D., Casey, G.: UCI machine learning repository, http://archive.ics.uci.edu/ml, University of California, Irvine, School of Information and Computer Sciences (2017)
72. Yang, Y., Korb, K., Ting, K.M., Webb, G.I.: Ensemble selection for SuperParent-One-Dependence estimators. In Lecture Notes in Computer Science: Proceedings of the 18th Australian Conference on AI (AI 05), volume LNCS 3809, pages 102–111. Berlin: Springer (2005)
73. Jiang, L., Zhang, H.: Lazy averaged one-dependence estimators. In: Lamontagne, L., Marchand, M. (eds.) Advances in Artificial Intelligence, Canadian AI 2006. Lecture Notes in Computer Science, **4013**. Springer, Berlin, Heidelberg (2006)
74. Lam, W., Bacchus, F.: Learning Bayesian belief networks: an approach based on the MDL principle. Computational Intelligence, **10**, 269–293 (1994)
75. Dawid, A.P.: Present position and potential developments: Some personal views. Statistical theory. The prequential approach. Journal of Royal Statistical Society, Series A, **147**, 278–292 (1984)
76. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. International Joint Conference on Artificial Intelligence, 1137–1143 (1995)
77. Vapnik, V.N.: Statistical learning theory. John Wiley & Sons, New York (1998)
78. Cowell, R.: Introduction to inference for Bayesian networks. In: M. I. Jordan (ed.) Learning Graphical Models, 9–26. MIT Press, Cambridge, Massachusetts (1999)
79. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.H.: Classification and Regression Trees, Wadsworth, Belmont, CA (1984)
80. Breiman, L.: Random forests. Machine Learning, **45**, 5–32 (2001)
81. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research, **7**, 1–30 (2006)
82. Bishop, C.M.: Neural networks for pattern recognition. Oxford University Press, UK (1995)
83. Cortes, C., Vapnik, V.: Support vector networks. Machine Learning, **20**, 273–297 (1995)

84. Macià, N., Bernadó-Mansilla, E.: Towards UCI+: A mindful repository design. Information Sciences, **261**, 237–262 (2014)
85. Malka, R., Lerner, B.: Classification of fluorescence in-situ hybridization images using belief networks. Pattern Recognition Letters, **25**, 1777–1785 (2004)
86. Atassi, N., Berry, J., Shui, A., Zach, N., Sherman, A., Sinani, E., et al.: The PRO-ACT database design, initial analyses, and predictive features. Neurology, **83**, 1719–1725 (2014)
87. Halbersberg, D., Lerner, B.: Young driver fatal motorcycle accident analysis by jointly maximizing accuracy and information. Accident Analysis and Prevention, **129**, 350–361 (2019)
88. Halbersberg, D., Wienreb, M., Lerner, B.: Joint maximization of accuracy and information for learning the structure of a Bayesian network classifier. Machine Learning, **109**, 1039–1099 (2020)
89. Halbersberg, D., Lerner, B.: Learning a Bayesian network classifier by jointly maximizing accuracy and information, Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI), The Hague, Holland, 1638–1639 (2016)
90. Silva, R., Scheines, R., Clark, G., Spirtes, P.: Learning the structure of linear latent variable models. Journal of Machine Learning Research, **7**, 191–246 (2006)
91. Asbeh, N., Lerner, B.: Learning latent variable models by pairwise cluster comparison. Part I - Theory and overview. Journal of Machine Learning Research, **17** (224), 1–52 (2016)
92. Asbeh, N., Lerner, B.: Learning latent variable models by pairwise cluster comparison. Part II - Algorithm and evaluation. Journal of Machine Learning Research, **17** (233), 1–45 (2016)
93. Halbersberg, D., Lerner, B.: Local to global learning of a latent dynamic Bayesian network. In: De Giacomo, G., Catalá, A., Dilkina, B., Milano, M., Barro, S., Bugarín, A., Lang. J. (eds.) ECAI 2020. Proceedings of the 24th European Conference on Artificial Intelligence, Santiago de Compostela, Spain, Frontiers in Artificial Intelligence and Applications, **325**, 2600–2607, IOS Press (2020)
94. Friedman, N.: The Bayesian structural EM algorithm. Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, 129–138, Morgan Kaufmann Publishers Inc. (1998)
95. Murphy, K.P.: Machine learning: A probabilistic perspective. MIT Press (2014)
96. Ikeda, K., Hirayama, T., Takazawa, T., Kawabe, K., Iwasaki, Y.: Relationships between disease progression and serum levels of lipid, urate, creatinine and ferritin in Japanese patients with amyotrophic lateral sclerosis: a cross-sectional study. Internal Medicine, **51**, 1501–1508 (2012)
97. Wagstaff, K.L.: Machine learning that matters. In Proceedings of International Conference on Machine Learning (ICML), 529–536 (2012)
98. Rudin, C. Wagstaff, K.L.: Machine learning for science and society. Machine Learning, **95**, 1–9 (2014)
99. Woodward, J.: Making things happen: A theory of causal explanation. Oxford University Press (2005)
100. Kim, B., Khanna, R., Koyejo, O.: Examples are not enough, learn to criticize! Criticism for interpretability. In: D. Lee and M. Sugiyama and U. Luxburg and I. Guyon and R. Garnett (eds.), Advances in Neural Information Processing Systems, **29**, Curran Associates, Inc. (2016)
101. Rohekar, R.Y., Nisimov, S., Gurwicz, Y., Koren, G., Novik, G.: Constructing deep neural networks by Bayesian network structure learning. Proceedings of the 32nd International Conference on Neural Information Processing Systems, 3051–3062, Montréal, Canada (2018)