Neurocomputing 456 (2021) 648–665

Contents lists available at ScienceDirect

Neurocomputing

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

Bayesian networks for interpretable machine learning and optimization

Bojan Mihaljević [a,*], Concha Bielza [b], Pedro Larrañaga [b]

[a] Departamento de Matemáticas, Universidad Autónoma de Madrid, Madrid, Spain
[b] Universidad Politécnica de Madrid, Madrid, Spain

ARTICLE INFO

Article history:
Received 16 March 2020
Revised 12 December 2020
Accepted 20 January 2021
Available online 17 June 2021

Keywords:
Interpretability
Explainable machine learning
Probabilistic graphical models

ABSTRACT

As artificial intelligence is being increasingly used for high-stakes applications, it is becoming more and more important that the models used be interpretable. Bayesian networks offer a paradigm for interpretable artificial intelligence that is based on probability theory. They provide a semantics that enables a compact, declarative representation of a joint probability distribution over the variables of a domain by leveraging the conditional independencies among them. The representation consists of a directed acyclic graph that encodes the conditional independencies among the variables and a set of parameters that encodes conditional distributions. This representation has provided a basis for the development of algorithms for probabilistic reasoning (inference) and for learning probability distributions from data. Bayesian networks are used for a wide range of tasks in machine learning, including clustering, supervised classification, multi-dimensional supervised classification, anomaly detection, and temporal modeling. They also provide a basis for estimation of distribution algorithms, a class of evolutionary algorithms for heuristic optimization. We illustrate the use of Bayesian networks for interpretable machine learning and optimization by presenting applications in neuroscience, the industry, and bioinformatics, covering a wide range of machine learning and optimization tasks.

1. Introduction

Artificial intelligence is increasingly present in everyday lives of ordinary citizens. For example, machine learning is widely applied for evidence-based decision-making in domains such as healthcare, policing, and finance [1]. Many of the machine learning models used are black boxes [2] that do not explain their predictions in a way that a human can understand. This lack of transparency has had particularly severe consequences in high-stakes applications: people have been incorrectly denied parole, while, on the other hand, poor bail decisions have led to releasing of dangerous criminals [3]. Government agencies [4] and other actors [5] are now increasingly demanding for artificial intelligence to be explainable and transparent; the European Union, for example, guarantees the right to an explanation of an automated decision in domains such as medicine, law and finance. In addition, experts in many fields — ranging from medical diagnosis [6] and bioinformatics [7] to financial applications [8]— require understanding the model's decisions in order to use it [9]. In data science and scientific discovery, the analysis of interpretable models learned from data can provide novel knowledge and lead to the formulation of new theories (see e.g. [10]).

The 'explainable artificial intelligence' [11,12] approach tries to 'explain' a black box model with a second, post hoc model. For example, neural networks are often converted into decision trees or logical rules while deep neural networks for text and images are explained with saliency masks that highlight the determining aspects of a text or image [13]. These explanations, however, are often not faithful to the original model nor do they provide sufficient detail [3]. Contrary to a black-box model, an interpretable model does not require a post hoc explanation in order to be understandable by a human [14,15]. Interpretability is, however, an elusive concept [14,16,17] that may mean different things to different stakeholders [18,19]. For example, the developers of the model may primarily care about quality assurance, policy-makers about fairness, and end users about whether the output can be trusted. Arrieta et al. [16], Lipton [14] and Murdoch et al. [17], among others, consider that the general characteristics of an interpretable model include simulatability (i.e., a human is able to contemplate and reason about the entire decision-making process at once), decomposability, and algorithmic transparency while Rudin [3] considers it a domain-specific notion and emphasises models that obey domain specifics such as causality or monotonicity. Nonetheless, models such as logical rules, linear models and deci-

* Corresponding author.
E-mail address: bojan.mihaljevic@uam.es (B. Mihaljević).

https://doi.org/10.1016/j.neucom.2021.01.138
0925-2312/

sion trees are widely considered to be interpretable [3,13,16,20,21], while some authors also consider k-nearest neighbours, additive models, and Bayesian networks to be interpretable [16,21]. A small rule set, for example, is simulatable, as it can be fully contemplated by a user, while being decomposable as well as algorithmically transparent. A large rule set, on the other hand, is not simulatable, yet each of its predictions might be, as long as they are given by a few rules; the model is then locally interpretable [13]. Although many authors assume that there is a trade-off between predictive performance and interpretability (e.g., [17]), there is often no significant difference between complex and simple, often interpretable, classifiers on structured data with meaningful features [3]. In addition, the apparent advantages observed in 'laboratory' settings during model comparison may well be overwhelmed in practice by issues such as low quality class labels and sample selection bias [22]. For example, a classifier that very accurately distinguished between friendly and enemy tanks on the test set of photographs later had very poor performance in the field; subsequent analysis found that friendly photos were taken on sunny days while enemy photos on overcast days [21]. Such issues leave the simpler models as the appropriate choice by the principle of parsimony [22].

A direct extension of logical rules for plausible reasoning with uncertainty is probability theory [23]. Modeling a joint probability distribution (JPD) over the random variables of a domain allows us to perform probabilistic reasoning, such as predicting the value of a particular variable given the values of other variables. A JPD over many variables, however, cannot be specified directly due to its enormous size. A Bayesian network (BN) [24–26] allows us to compactly model a JPD over many random variables by leveraging conditional independencies among them. It also provides a basis for specifying algorithms for reasoning (inference) and for learning models from data. In a BN, the variables are represented as the nodes of a directed acyclic graph (DAG); the graph is referred to as the networks' structure and its arcs have a formal interpretation in terms of probabilistic conditional independence among variables. In addition to the graph, a BN has a quantitative part, a set of parameters that specify the conditional probabilities for each node in the DAG. The JPD is then given by the product of all these conditional probabilities associated with the DAG. The structure and the conditional probabilities of a BN can be given by a domain expert or may alternatively be learned automatically from data, with the optional inclusion of expert knowledge. Once the BN is specified, it constitutes a powerful tool for reasoning with exact or approximate inference methods. For example, it allows for abductive inference, that is, finding an explanation for some observed evidence. They are thus widely used [e.g.,][27–29] for diagnosis, prognosis and prescription in healthcare, as they provide for interpretable and rational decision-making in domains with inherent uncertainty [30]. A BN is decomposable due to conditional independencies, the learning algorithm is transparent as it mainly amounts to combinatorial search and distribution fitting, and is simulatable as long as it is not excessively large [16].

Indeed, many of the early applications of BNs were to expert systems in medicine. The medical experts were unlikely to follow the given advice unless they understood how the model reached its conclusion and why it was appropriate [31,32]. In such a setting, an interpretable and simple model on its own may not be sufficient. In particular, experiments with the early medical decision support system MYCIN [33] showed that logical rules alone are not sufficient for explanations that are understandable to medical students [34]. Indeed, while the normative probabilistic reasoning, implemented with BNs, can be at odds with human reasoning under uncertainty, which is plagued with heuristics and biases [35], formal logic is not necessarily a good model of human reasoning either, as it provides domain-independent rules while human reasoning is content-dependent [36]. A number of solutions have thus been developed to help the user understand the reasoning (that is, the inference process) of a BN as well as the model itself [37]. These solutions are studied, along with abductive inference, under the term explanations of BNs. The term explanation here comes from the expert systems literature and its meaning is distinct from that of the post hoc explanations of black-box models. For example, many of the tools for 'explaining' the model are simply tools for visualizing the graph of a BN; they simplify the access to the model without altering it.

In addition to machine learning, BNs are widely used in another area of artificial intelligence, namely, that of heuristic optimization [38]. In particular, they provide a basis for estimation of distribution algorithms [39–42], evolutionary algorithms that, instead of the mutation and crossover operators of genetic algorithms, estimate a JPD over promising solutions and then sample from this JPD to produce a new generation of solutions.

The aim of this paper is to illustrate that BNs are an excellent paradigm for interpretable artificial intelligence. Besides being able to provide explanations of their predictions, the structure and the parameters learned from data provide information about the probabilistic dependencies among the variables. BNs are also a versatile framework that is used for a wide range of tasks in machine learning. We illustrate our claims by describing a number of applications of BNs for machine learning and heuristic optimization. In particular, we cover applications to different machine learning tasks —namely, clustering, supervised classification, multi-dimensional supervised classification, and anomaly detection in a temporal domain— in two different domains, neuroscience and the industry. Regarding heuristic optimization, we present an application to a bioinformatics uni-dimensional combinatorial problem and an application to a multi-dimensional problem with continuous variables, focusing on the discovery of relationships between the variables in the former and the objectives in the latter.

The rest of the paper is organized as follows. Section 2 introduces BNs by describing their semantics and the concept of conditional independence, explaining exact and approximate inference, giving a brief review of explanation in BNs, describing algorithms for learning a network structure and the conditional probabilities from data, and presenting their adaptations to supervised classification and temporal modeling. Section 3 then presents applications of BNs for machine learning in neuroscience and the industry. In neuroscience, we present applications in neuroanatomy, neurophysiology and in a neurodegenerative disease. Regarding industry, we present the use of dynamic BNs for anomaly detection in laser surface heat treatment in manufacturing. Section 4 presents the application of BNs in estimation of distribution algorithms in a combinatorial uni-objective problem as well as in a multi-objective problem with continuous variables. Section 5 rounds the paper off with conclusions.

2. Bayesian networks

BNs are widely used models of uncertain knowledge. They are useful because they can provide a compact representation of a JPD across many random variables, $p(X_1, \ldots, X_n)$. The JPD over the variables of a domain is of great interest since, when known, it lets us answer any probabilistic question about the domain and thus solve tasks of interest such as, for example, predicting the value of a particular variable given the values of other variables.

In general, however, a JPD is intractable with a medium or large n (number of variables) because specifying it requires an enormous number of parameters (e.g., $2^n - 1$ parameters if all variables are binary). This is intractable computationally, as we cannot store nor process that many parameters; cognitively, as an expert cannot

understand such a model; and statistically, as we cannot obtain sufficient data to estimate it reliably. A JPD can be made tractable by leveraging the notion of conditional independence between variables in order to reduce the number of parameters. Random variables X and Y are *conditionally independent* (c.i.) given another random variable Z if

$$p(x|y,z) = p(x|z) \quad \forall x,y,z \text{ values of } X,Y,Z.$$

That is, X and Y are c.i. given Z if, for any $Z = z$, knowing $Y = y$ does not affect the probability of x (note that X, Y, Z may also be disjoint random *vectors*). Thus, after decomposing $p(x,y,z)$ according to the chain rule, $p(x,y,z) = p(z)p(y|z)p(x|y,z)$, we can equivalently write it as $p(z)p(y|z)p(x|z)$, thus reducing the number of parameters in the last factor.

A BN consists of a DAG $\mathcal{G}$ and a set of parameters $\boldsymbol{\theta}$ (see Fig. 1). The vertices (i.e., nodes) of $\mathcal{G}$ correspond to the variables $\mathbf{X} = (\mathbf{X_1}, \ldots, \mathbf{X_n})$ while its directed edges (i.e., arcs) encode the conditional independencies among the variables $\mathbf{X}$. The *parents* of a node X_i, $\mathbf{Pa}(X_i)$, are all the nodes with arcs pointing to X_i, while the children of X_i are all nodes towards which X_i has outgoing arcs. The *descendants* of X_i are all the nodes reachable from X_i by following the arcs, while its complement in $\mathbf{X} \setminus \{\mathbf{X_i}\}$ is $\mathbf{ND}(X_i)$, the set of *non-descendants* of X_i. The basic set of conditional independences encoded by a BN is

$$X_i \text{ is c.i. of } \mathbf{ND}(X_i) \text{ given } \mathbf{Pa}(X_i), \ i = 1, \ldots, n,$$

that is, each node is c.i. of its non-descendants given its parents. This set of independencies is referred to as the *local Markov independencies*.

If the local Markov independencies of $\mathcal{G}$ hold in a JPD $p(\mathbf{X})$ then we can factorize $p(\mathbf{X})$ according to $\mathcal{G}$ and vice versa. Namely, the fact that $\mathcal{G}$ is acyclic ensures there is at least one topological ordering of the variables $X_1, \ldots, X_n$ such that $\{X_1, \ldots, X_{i-1}\}$ only contains non-descendants of X_i. Thus, after applying the chain rule,

$$p(X_1, \ldots, X_n) = p(X_1)p(X_2|X_1)p(X_3|X_1, X_2) \cdots p(X_n|X_1, \ldots, X_{n-1}),$$

we can remove all non-descendants other than parents from the conditioning sides,

$$p(X_1, \ldots, X_n) = p(X_1|\mathbf{Pa}(X_1)) \cdots p(X_n|\mathbf{Pa}(X_n)), \tag{1}$$

thus obtaining the factorization of $p(\mathbf{X})$ according to $\mathcal{G}$. This factorization is known as the chain rule for BNs.

The parameters $\boldsymbol{\theta}$ specify the local conditional distributions of each variable given its parents' values. For a discrete variable X_i, each parameter θ_{ijk} encodes the probability $p(X_i = k|\mathbf{Pa}(X_i) = j)$, and all the θ_{ijk} are usually tabulated in a *conditional probability table* (CPT). When $\mathbf{X}$ contains only continuous variables, it is straightforward to model a multivariate normal density over $\mathbf{X}$ with a BN. In this case, the local conditional density for X_i is $p(X_i|PaX_i) = \mathcal{N}(X_i; \beta_{i0} + \beta_i^T PaX_i, \sigma_i^2)$, where PaX_i is an assignment to $\mathbf{Pa}(X_i)$. The parameters set, $\boldsymbol{\theta}$, thus contains a vector of coefficients $(\beta_{i0}, \beta_i^T, \sigma_i^2)$ for each X_i.

2.1. Conditional independence

In addition to the local Markov independencies, a DAG $\mathcal{G}$ may encode additional independencies that hold in a JPD p that factorizes over $\mathcal{G}$. All such independencies can be identified by verifying the graphical *d-separation* property. Thus, if node X is d-separated from node Y given node Z, then X and Y are c.i. given Z. The set of all independencies verified by d-separation is called the set of *global Markov independencies* [25]. While all independencies implied by d-separation hold in p, the reverse might not be true: a conditional independence that holds in p need not be verified with the d-separation property. If the reverse does hold, meaning that the set of global Markov independencies and those that hold in p are equivalent, then p is said to be *faithful* to $\mathcal{G}$ and $\mathcal{G}$ to be a *perfect map* of p.

For any node X_i, a set of variables of particular interest is its *Markov blanket*, composed of its parents, its children and the par-

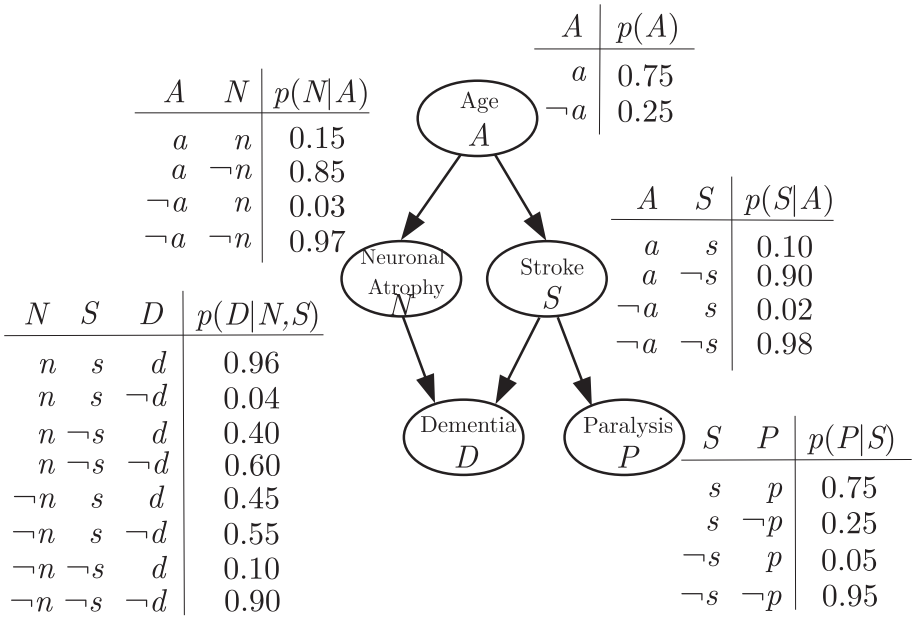


Fig. 1. A hypothetical BN modeling the risk of dementia. All variables are binary, with x denoting 'presence' and $\neq gx$ denoting 'absence', for `Dementia` D, `Neuronal Atrophy` N, `Stroke` S and `Paralysis` P; for `Age` A, a means 'aged 65+' and otherwise the state is $\neq ga$. Note that both `Stroke` and `Neuronal Atrophy` are influenced by `Age`, their parents in the DAG. These two conditions influence `Dementia`, their child in the DAG. `Paralysis` is directly associated with having a stroke. Since `Age` is a non-descendant of `Dementia`, it is independent of it given `Neuronal Atrophy` and `Stroke`, the parents of `Dementia`. The JPD factorizes as $p(A,N,S,D,P) = p(A)p(N|A)p(S|A)p(D|N,S)p(P|S)$. The CPTs are depicted as tables and they contain the parameters, encoding the conditional probabilities attached to each node. For instance, if someone has neuronal atrophy and has had a stroke, there is a 0.96 probability that the person will have dementia: $p(d|n,s) = 0.96$; in the absence of neuronal atrophy and stroke, this probability is 0.10, i.e., $p(d| \neq gn, \neq gs) = 0.10$. Figure from [43].

ents of its children (spouses) in $\mathcal{G}$. Namely, each X_i is c.i. of all other nodes in the network given its Markov blanket:

$$p(X_i|\mathbf{X} \setminus \{X_i\}) = p(X_i|\mathbf{MB}(X_i)).$$

Therefore, for example, the only knowledge useful for predicting X_i is that of the variables in $\mathbf{MB}(X_i)$.

2.2. Exact and approximate inference

Besides visualizing the relationships between variables and verifying conditional independencies, a BN allows for any type of probabilistic reasoning over a domain, including causal (predictive), diagnostic, and abductive reasoning. Such reasoning is performed by means of probabilistic queries. The two most common queries are conditional probability (CPQ) and maximum a posteriori (MAP) queries.

A CPQ refers to finding $p(x_i|\mathbf{e})$, the probability of a query variable X_i conditioned on $\mathbf{e}$, the values of the observed variables $\mathbf{E}$, called the *evidence*. Note that, in addition to X_i and $\mathbf{E}$, we may also have unobserved non-query variables $\mathbf{Y}$. In Fig. 2 we see that, for example, the probability of a patient having a paralysis goes up from 11% to 75% after learning that the patient has had a stroke.

A MAP query refers to finding the values of a set of variables that best explain the observed evidence, allowing for abductive reasoning. That is, we are interested in $\arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{e})$, where the solution is referred to as the *most probable explanation* (MPE) when $\mathbf{Y}$ corresponds to all variables in $\mathbf{X}$ other than $\mathbf{E}$. In the BN from Fig. 1, the MPE for a patient with paralysis is he or she is aged 65 +, has had a stroke and has not had neither neuronal atrophy nor dementia.

Computing these probabilities is conceptually simple. For example, with discrete variables

$$p(x_i|\mathbf{e}) = \frac{p(x_i, \mathbf{e})}{p(\mathbf{e})} \propto \sum_{\mathbf{y}} p(x_i, \mathbf{e}, \mathbf{y}).$$

The limitation, however, is that the summation over $\mathbf{y}$, needed in order to marginalize $\mathbf{Y}$, grows exponentially with the number of variables in $\mathbf{Y}$. Thus, many algorithms exist for tackling the computation of exact and approximate inference.

2.3. Exact inference methods

Exact inference is NP-hard [44,45] in general BNs, meaning that a polynomial time algorithm (in n) is most likely not to exist. A number of algorithms, however, work well in many practical cases. The *variable elimination* algorithm leverages the network structure in order to look for an efficient marginalization ordering. While finding the optimal marginalization ordering is an NP-hard [46] problem on its own, greedy algorithms tend to work well in practice. The algorithm for message passing on junction trees [47] allows for answering a series of queries in only twice the runtime of a single query. Also, a BN can be represented as a polynomial [48], thus allowing for efficient inference by evaluating and differentiating the polynomial.

2.4. Approximate inference methods

For complex networks and non-standard local distributions, we may need to resort to approximate inference. Approximate inference in general BNs is also NP-hard [49]. A common and broadly applicable approach is that of *particle-based* inference or Monte Carlo simulation. Namely, we use the network to sample a large number of *particles* (cases) from the JPD, and then estimate the probability of interest from the generated sample (e.g., by counting observed relative frequencies if the variables of interest are discrete). The simplest approach is *probabilistic logic sampling* [50] where, given a topological ordering of the nodes, we sample from a node once we have sampled from, and thus fixed the values of, its parents. However, if the evidence $\mathbf{e}$ is very unlikely, many sampled particles will be discarded since they will not match the evidence, and we would thus need an intractable number of samples. *Likelihood weighting* [51,52] mitigates this by fixing the values of, rather than sampling from, the evidence nodes and weighing each particle with the likelihood of the evidence given its parents' values in the particle. Other techniques are *Gibbs sampling* and more general *Markov chain Monte Carlo* (MCMC) methods.

2.5. Explanation in Bayesian networks

The explanations of BNs can be focused on the model, on the reasoning process, or on the evidence [37]. While explaining the model and the reasoning process aim to aid the user, explaining the evidence studies the use of BNs as a tool for explaining observed phenomena.

Explaining the model means, in its most basic sense, displaying it to the user either graphically or verbally [37]. When the network's graph is too large to fit onto a screen, software tools can collapse subgraphs into special nodes that can be expanded when needed. Also, arcs can be colored to denote features such as, for example, the sign of the correlation between a parent and a child [37,53,165]. The graph semantics, however, are not trivial and,

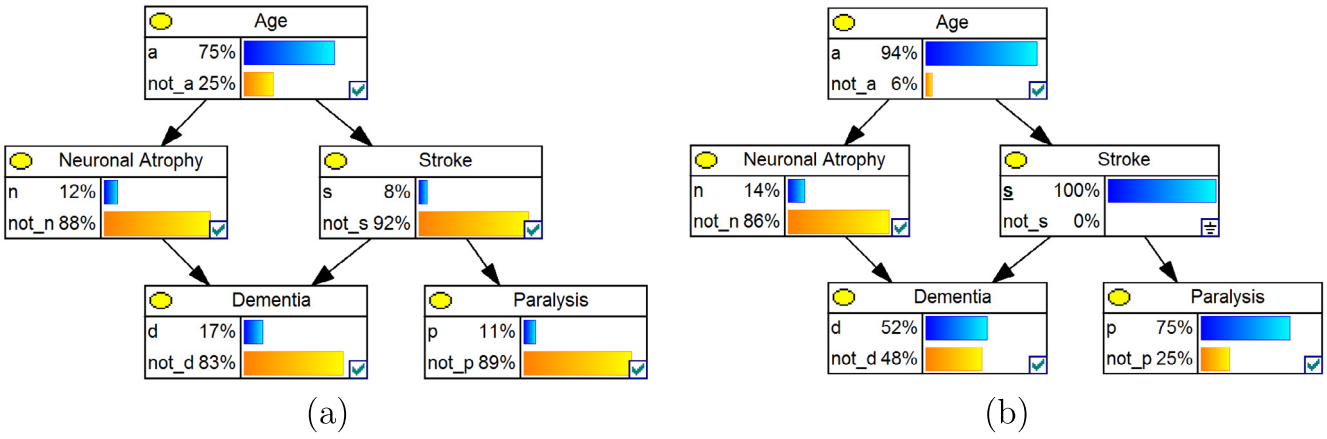


Fig. 2. Exact inference on the risk of dementia example. (a) Prior distributions $p(X_i)$ are shown as bar charts, for each node X_i. For example, the prior probability of paralysis is 11%. (b) After observing someone who has had a stroke ($S = s$), the distributions are updated as $p(X_i|s)$ and the posterior probability of paralysis is 75%. Figure from [43].

for example, arrow directionality can be confusing to an untrained user when it lacks a causal meaning [54]. There are thus tools that provide textual descriptions of the model and its conditional independencies [55].

Understanding the model's reasoning is critical for user adoption. For example, the Pathfinder lymph diagnosis system [27] provides simple explanations by showing how the different values x of a variable X favor one of two competing diagnoses, D_1 and D_2, in terms of the weight of evidence [56], $\log\frac{P(x|D_1)}{P(x|D_2)}$. Madigan et al. [57] also use the weight of evidence to explain the magnitude of the effect of each finding on the variable of interest. A number of solutions explain the reasoning verbally [e.g.,][58,59], for example with step-by-step stories that describe the propagation of evidence while representing probabilities as numbers and/or phrases [55].

The explanation of evidence in BNs is a realization of *inference to the best explanation*, or abduction, a reasoning mode considered common in both science and everyday life [60]. Abduction consists in choosing the best among a set of competing hypotheses on the basis of how well they explain the evidence. In BNs, the hypotheses correspond to states of unobserved (non-evidence) variables and best usually means the most probable assignment to either all (MPE), or some (MAP), of the unobserved variables [24]. Since both MPE and MAP are overspecified when many of the unobserved variables are irrelevant to the observed evidence, a number of solutions seek to provide concise explanations with fewer variables [61–63]. Yuan et al. [64], on the other hand, find concise explanations, consisting of relevant variables, by maximizing the generalized Bayes factor instead of the posterior probability. While very common in BNs, the posterior probability is one of many criteria for hypothesis selection that are used in the wider context of abduction. Alternatives include the weight of evidence [56], explanatory power [65,66], likelihood of evidence [67], and the product coherence measure [68], with ongoing debate regarding their merits and shortcomings [69].

2.6. Learning Bayesian networks from data

Learning a BN from a data set $\mathscr{D} = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ of N observations of $\mathbf{X}$ involves two steps: (a) learning the DAG $\mathscr{G}$; and (b) learning $\boldsymbol{\theta}$, the parameters of the local conditional distributions. There are two main approaches to learning $\mathscr{G}$ from $\mathscr{D}$: (a) by testing for conditional independence among triplets of sets of variables (the *constraint-based* approach); and (b) by searching the space of DAGs in order to optimize a score such as the penalized likelihood (the *score-based* approach).

The prototypical constraint-based algorithm is the PC algorithm [70]. It begins by establishing that a pair of variables X and Y are connected in $\mathscr{G}$, without setting the direction of the arc, if it cannot find a set $\mathbf{Z}$ such that X and Y are independent conditionally on $\mathbf{Z}$. This is established heuristically with sequences of conditional independence tests for X and Y given $\mathbf{S}$, performed with increasingly large sets $\mathbf{S}$ until either: (a) non-rejection of the independence hypothesis; (b) reaching a size limit on $\mathbf{S}$. The algorithm then derives directions for some of the arcs with rules such as the following: if X and Y are marginally independent (and thus non-adjacent in $\mathscr{G}$) but not independent given their common neighbour Z in $\mathscr{G}$, then the arcs are oriented as $X \rightarrow Z \leftarrow Y$. More recent algorithms such as HITON [71,72] and Grow-Shrink [73] use additional heuristics to reduce runtime.

Score-based algorithms have two components: a network score and a discrete optimization technique (i.e., a search algorithm) that is used to maximize the score among candidate networks. A typical score-based search algorithm is hill climbing, a local search which, starting from some initial DAG $\mathscr{G}$, greedily adds, removes or reverses arcs as long as that improves the score. Other algorithms include the tabu meta-heuristic [74], which allows for score-degrading operators while, for efficiency, avoiding those that undo the effect of recently applied ones, and genetic algorithms [75]. The score-based approaches tend to be more robust [25] than constraint-based ones, as they may reconsider previous steps in the search by removing or reversing previously added arcs. A commonly used group of network scores is that of penalized log-likelihood scores, such as the Bayesian information criterion (BIC) [76].

Given $\mathscr{G}$, learning $\boldsymbol{\theta}$ is generally straightforward when data are complete. For discrete variables X_i and $\mathbf{Pa}(X_i)$, we can compute the Bayesian estimates in closed form by assuming a Dirichlet prior over $\boldsymbol{\theta}$. With all Dirichlet hyper-parameters equal to α,

$$\hat{\theta}_{ijk} = \frac{N_{ijk} + \alpha}{N_{ij\cdot} + |\Omega_{X_i}|\alpha}, \tag{2}$$

where N_{ijk} is the number of instances in $\mathscr{D}$ such that $X_i = k$ and $\mathbf{Pa}(X_i) = j$, corresponding to the j-th possible instantiation of $\mathbf{Pa}(x_i)$, $N_{ij\cdot}$ is the number of instances in which $\mathbf{Pa}(x_i) = j$, while $|\Omega_{X_i}|$ is the cardinality of X_i. Setting $\alpha = 0$ in Eq. (2) yields the maximum likelihood estimate of θ_{ijk}. With incomplete data, the parameters of local distributions are no longer independent and we cannot separately maximize the likelihood for each X_i as in Eq. (2). Optimizing the likelihood requires a time-consuming algorithm like expectation maximization [77] which does not guarantee convergence to the global optimum.

2.7. Bayesian network classifiers

For BN classifiers [78,79], a common space of structures to search in is that of augmented naive Bayes [80] models, which factorize $P(\mathbf{X}, C)$ as

$$P(\mathbf{X}, C) = P(C)\prod_{i=1}^{n} P(X_i|\mathbf{Pa}(X_i)), \tag{3}$$

where C is the class variable and $\mathbf{X}$ the predictors, and $C \in \mathbf{Pa}(X_i)$ for all X_i and $\mathbf{Pa}(C) = \varnothing$.

Models of different complexity arise by extending or shrinking the parent sets $\mathbf{Pa}(X_i)$, ranging from the naive Bayes [81] with $\mathbf{Pa}(X_i) = \{C\}$ for all X_i, to those with a limited-size $\mathbf{Pa}(X_i)$ [80,82], to those with unbounded $\mathbf{Pa}(X_i)$ [83]. While the naive Bayes can only represent linearly separable classes [84], more complex models are more expressive [85]. Simpler models, with sparser $\mathbf{Pa}(X_i)$, may perform better with less training data, due to their lower variance, yet worse with more data as the bias due to wrong independence assumptions will tend to dominate the error.

The algorithms commonly used to produce the above structures are generally instances of greedy hill-climbing [82,86], with arc inclusion and removal as their search operators. Some add node inclusion or removal [87], thus embedding feature selection [88] within structure learning. Alternatives include the adaptation [80] of the Chow-Liu algorithm [89] to find the optimal one-dependence estimator with respect to decomposable penalized log-likelihood scores in time quadratic in n.

A special case is multi-dimensional BN classifier [90,91], a BN in which two or more nodes correspond to the class variables and the remaining ones to the predictors.

2.8. Dynamic Bayesian networks

Dynamic BNs [92,93] model domains that evolve over time as discrete-time stochastic processes. Given vector of random variables $\mathbf{X}^t = (X_1^t, \ldots, X_n^t)$ at each time slice $t = 1, \ldots, T$ and assuming a first-order Markovian transition model for the process, i.e., that $p(\mathbf{X}^t|\mathbf{X}^{t-1}, \ldots, \mathbf{X}^1) = p(\mathbf{X}^t|\mathbf{X}^{t-1})$, we have that

$$p(\mathbf{X}^1,\ldots,\mathbf{X}^T) = p(\mathbf{X}^1)\prod_{t=2}^{T} p(\mathbf{X}^t|\mathbf{X}^{t-1}),$$

where, $p(\mathbf{X}^1)$ corresponds to the initial conditions and is factorized according to a *prior BN*. If we also assume that the process is stationary, then $p(\mathbf{X}^t|\mathbf{X}^{t-1})$ does not depend on t and is common for all time slices. It can then be factorized according to a *transition network*, over X_i as $\prod_{i=1}^{n} p(X_i|\mathbf{Pa}(X_i))$, where $\mathbf{Pa}(X_i)$ may contain nodes from both the same and the previous time slice. For inference purposes, we unroll the transition network over all time slices in order to obtain a standard BN structure. An algorithm for learning dynamic BNs is the dynamic hill-climbing algorithm [94], which improves a score of both the prior and the transition networks.

3. Bayesian networks in machine learning

BNs have been widely applied for machine learning in many fields, ranging from forensic science [95] to bioinformatics [96] to fault diagnosis [97] and neuroscience [98,43]. We now present a number of illustrative applications in neuroscience and the industry.

3.1. Neuroscience

The human nervous system is the most complex biological system. In order to effectively detect and respond to changes in the environment, it is capable of learning, self-awareness, and gives rise to the intellect. While many fundamental aspects of neuronal structure and function are well understood, many questions remain open. Answering them is becoming more urgent, mainly due to enormous social and economic cost of nervous system disorders. Brain disorders, such as dementia, depression, and addiction, account for 36% of the burden of all disease in high-income countries [99], with eight million attributable deaths per year [100]. The monetary cost of brain disorders in Europe was estimated to 798 billion euros in 2010 [101], while that of Alzheimer's disease alone in the United States in 2010 was estimated to be between 157 and 215 billion American dollars [102].

Progressing towards understanding the brain is a monumental endeavor. To this end, ambitious neuroscience projects have been launched globally [103] over the last decade or so. These include the Human Brain Project [104,105] in the European Union, the Brain Research through Advancing Innovative Neurotechnologies (BRAIN) initiative [106] and the Allen Institute for Brain Science in United States of America, and others in Canada, China, Japan, Korea, and Israel [103,107]. Most of these are extremely large projects, reflecting the complexity of the task. The Human Brain Project, for example, is one of the largest European-funded research projects ever, with the total funding planned to be around one billion euros. It is an interdisciplinary effort, including experts in computer science, physics, and mathematics [105], in addition to those in neuroscience and related life sciences.

BNs have been widely applied in neuroscience research. Bielza and Larrañaga[98] review many such applications, including more than 40 papers on applications in neuroimaging. In particular, dynamic BNs have been applied to problems in fMRI (dyslexia, Parkinson's disease, schizophrenia, dementia in elder subjects), MRI (mild cognitive impairment) and EEG (motor task). Below we describe studies tackling classification of cortical interneurons, the simulation of virtual somas of pyramidal neurons, as well as an application in neurodegenerative diseases.

A key challenge in neuroscience is the classification of GABAergic interneurons [108]. These neurons constitute around 20–30% of the neurons in the cerebral cortex and are the main component of inhibitory cortical circuits (see Fig. 3 for basics of neuron morphology), which in turn are associated with disorders such as epilepsy [109,110], autism [111], and schizophrenia [112–115]. While high-throughput generation of data may enable learning a systematic taxonomy from data in the near future [116–118], by clustering [119,120] molecular, morphological, and electrophysiological features, researchers currently use [e.g.,][121] and refer to established morphological types such as chandelier, Martinotti, neurogliaform, and basket [122–125]. Having a model to automatically classify interneurons [126] into these morphological types could bring insight and be useful to practitioners [123]. A simple and accurate model could provide an interpretable mapping from the quantitative characteristics to the types.

A number of studies have approached the problem of interneuron classification with methods based on BNs. They are all based on a landmark study of consensus among the scientific community on interneuron classification, in which 42 expert neuroscientists classified 320 interneurons according to a predefined taxonomy (see Fig. 4 for the definition of the taxonomy and details on the study). Since the taxonomy consisted of five morphological features, in addition to interneuron type, the study produced a data set [133] of the 320 neurons classified according to six variables (see Fig. 4 for definitions) by each of the 42 neuroscientists. In addition, the morphologies of 240 out of the 320 interneurons were digitally reconstructed which enabled studing the supervised classification of quantified interneuron morphologies into the type and the morphological features [134–136].

In the original study [123], the authors used BNs to study the classification choices of the neuroscientists (see Fig. 5). In particular, they learned a BN for each neuroscientist in order to model his or her reasoning in terms of the six variables. This enabled them to study, for example, how each expert related the morphological features, such as whether the axon was intra- or translaminar, with the interneuron type. As Fig. 5 shows, these networks let us identify similarities and differences in the reasoning among experts. After observing differences among the experts' BNs, López-Cruz et al. [137] sought to identify distinct schools of thought among the neuroscientists. They clustered the experts' BNs into six clusters and then learned from data a representative BN for each cluster (see Fig. 5), thereby modeling the characteristic reasoning patterns of its members. The authors then combined the clusters' BNs into a consensus Bayesian multinet (i.e., a weighted combination of multiple networks) thus modeling the reasoning patterns of all 42 experts. They performed inference with this consensus model in order to, for example, obtain properties of different interneuron types; for example, Martinotti cells were mainly translaminar, displaced, and ascending. Note that, while the JPDs in this setting (six variables) were not prohibitively large in order to be computationally tractable, they are hardly tractable cognitively for a domain expert, unlike with a BN which gives a compact, graphical representation of the domain by leveraging conditional independencies.

A second goal was to predict interneuron type and four morphological features from the digitally reconstructed morphologies. Mihaljević et al. [134] predicted each of these five variables separately with discrete BN classifiers. Unlike in a typical supervised classification setting, there were up to 42 labels for each instance, provided by the different neuroscientists. The level of agreement among experts varied across the cells: while there were 29 neurons such that at least 35 neuroscientists agreed on their interneuron type, there were 67 other cells such that no more than 15 of them agreed on a single type. The authors thus labelled each cell with the most common among the labels given by the 42 neuroscientists, yet repeated the classification on different subsets of neurons, formed by filtering out cells below a certain threshold on label reliability, defined as the minimal number of neuroscientists agreeing on the majority type. The models were accurate, with up to 89.52%

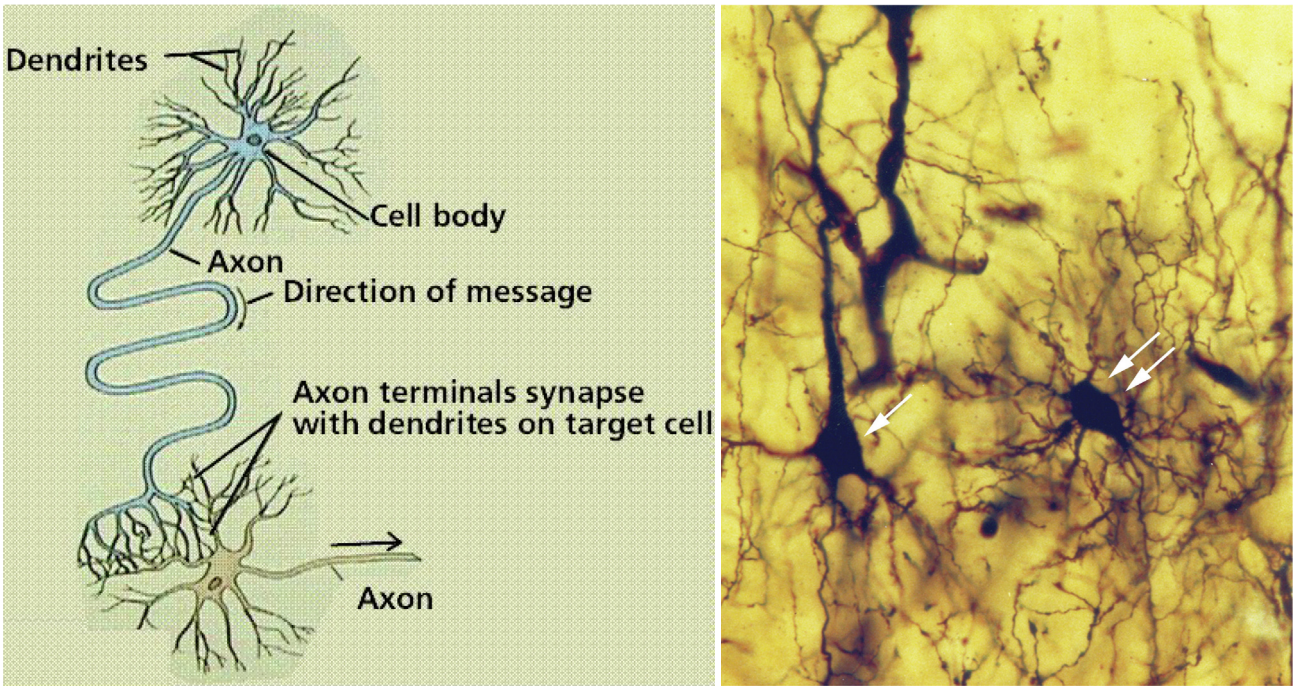


Fig. 3. (a) Neuronal morphology. The basic structural and functional unit of the nervous system is the nerve cell or neuron. There are around 10^{11} [127] neurons in the human brain, with 10^{15} connections among them [128]. A neuron's function is to receive and integrate information from sensory receptors or other neurons and transmit it to other neurons or organs. Each neuron has a single cell body, or *soma*, with branching processes, or neurites, called *dendrites* and *axon*, emerging from it. The dendrites receive chemical signals, or *neurotransmitters*, from axons of other neurons and transform them into electrical signals. The soma integrates incoming signals and may send a signal to other neurons, by an electrical potential that travels down the axon and away from the soma. At axon terminals, or *boutons*, this potential triggers the release of a neurotransmitter, into the *synapse*, the region between two adjacent neurons, passing the signal to the post-synaptic neuron. (b) Two main types of cortical neurons. Between 70% and 80% of neocortical neurons are excitatory pyramidal neurons (one arrow in the graphic) [129–131]. These cells are relatively uniform in terms of morphological, physiological and molecular properties [129]. The remaining 20–30% neurons are interneurons (two arrows in the graphic). They are mostly inhibitory, that is, use the gamma-amino butyric acid (GABA) as their neurotransmitter, and have short axons that do not leave the cortex. Photomicrograph from Cajal's preparation of the occipital pole of a cat stained with the Golgi method, taken from [132].

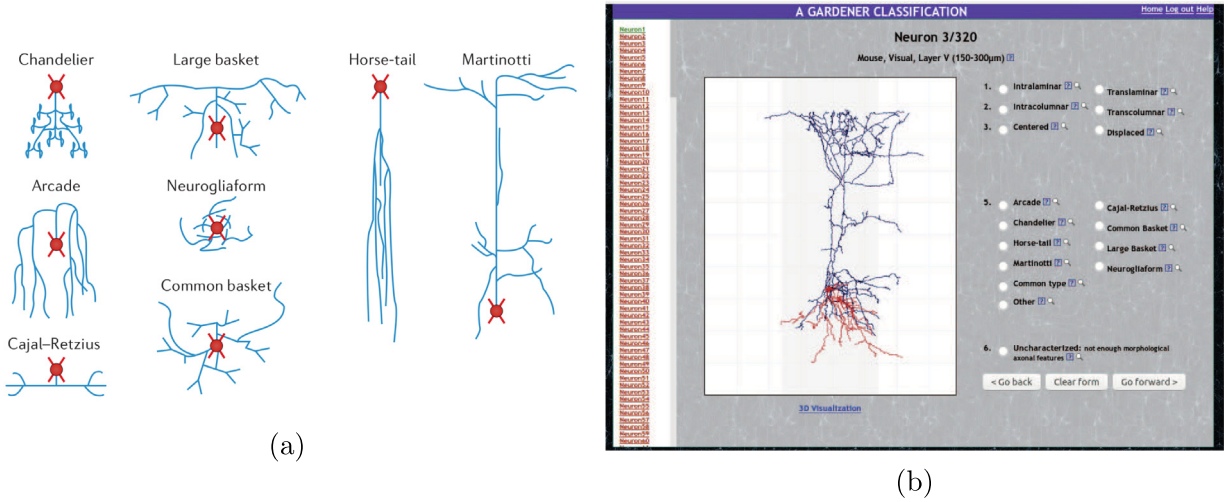


Fig. 4. Interneuron types and morphological features in the classification scheme by DeFelipe et al. [123]. The scheme contemplates ten interneuron types (a): chandelier, large basket, horse-tail, Martinotti, arcade, neurogliaform, Cajal-Retzius, common basket, common type, and other (common type and other not shown in the graphic). Other is meant to be chosen when the neuroscientist finds none of the remaining nine types adequate and prefers to use an alternative name. In addition to interneuron type, the classification scheme contemplates five high-level morphological features, such as whether or not the axon is restricted to the layer that contains the soma. These features, termed F1, F2, F3, F4, and F6 (F5 being the previously discussed interneuron type) have the following categories: (F1) intralaminar and translaminar; (F2) intracolumnar and transcolumnar; (F3) centered and displaced; (F4) ascending, descending, and both; (F6) characterized and uncharacterized. The uncharacterized category of F6 means that a cell's reconstruction is not good enough to reliably classify it. When labeling a cell as uncharacterized in feature F6, the neuroscientist cannot annotate it according to any of the remaining five features, F1-F5. F4 is only applicable for cells that are labeled as translaminar and displaced in F1 and F3, respectively. (b) The web application used to gather the neuroscientists' classification choices for the set of 420 interneurons. Figure (a) from [123].

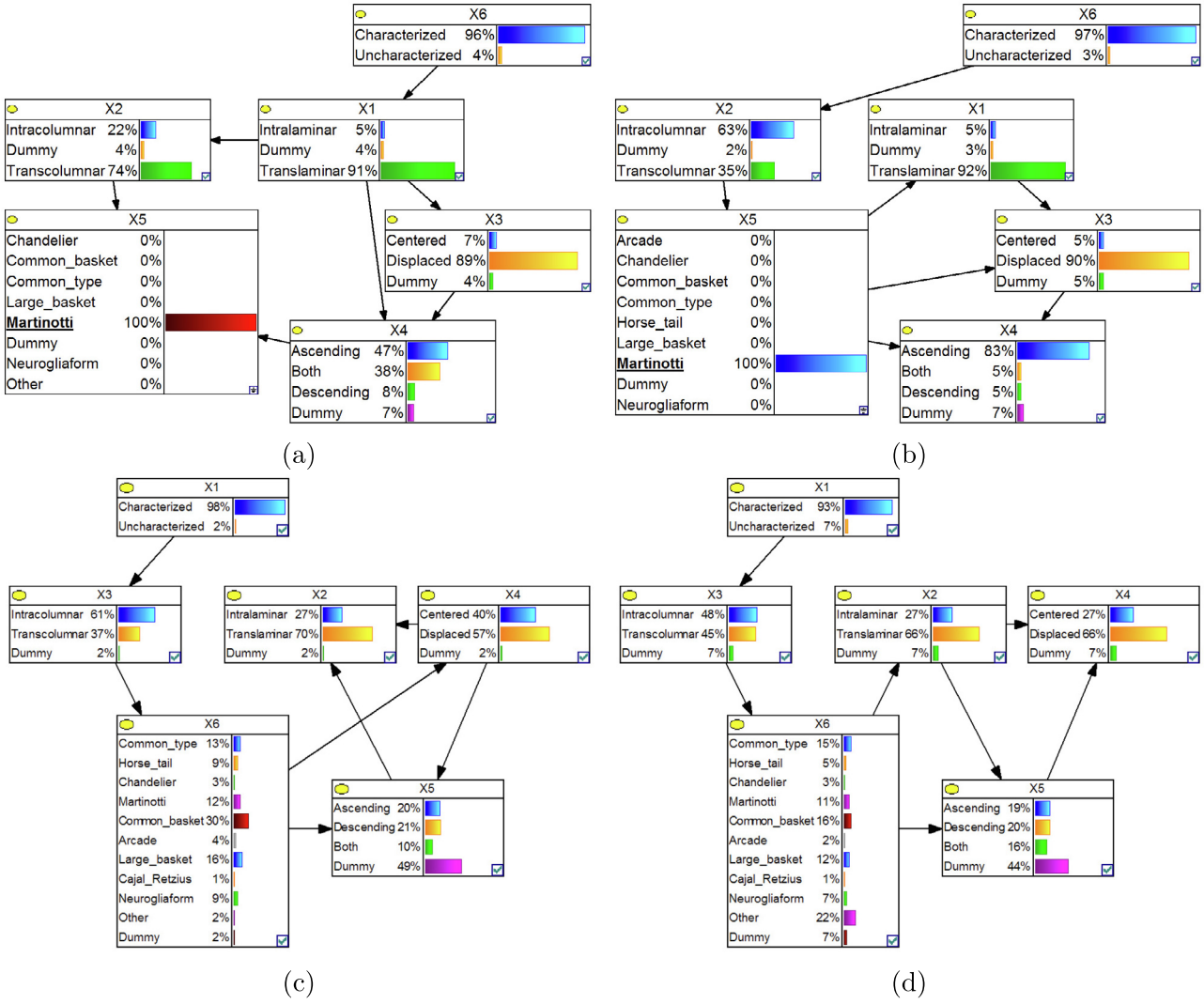


Fig. 5. Above: BNs for two of the 42 neuroscientists. Bar charts show the propagated probabilities of the remaining features conditioned on the Martinotti type. While the two neuroscientists agreed on the features X_1 and X_3 of Martinotti cells, they disagreed in terms of their features X_2 and X_4: for example, the probability of Martinotti cells being ascending is 47% in (5) a yet 83% in (5) b. Below: BNs for two clusters of neuroscientists. Bar charts show the marginal probabilities of the variables. Cluster (5) c consisted of 15 experts, with common basket as the mode for the interneuron type, whereas cluster (5) d consisted of seven experts that had a high probability for other, that is, considered than an alternative interneuron type was appropriate. Figure from [123].

accuracy for the interneuron type and even higher accuracy for the morphological features. Fig. 6 illustrates how a tree augmented naive Bayes can be used to explain the reasoning behind the classification of a cell, providing insight about the quantitative features of two interneuron types.

As an altenative to majority labels and data filtering, Mihaljević et al. [135] used probabilistic class labels while predicting the five variables at once. They encoded the multi-dimensional ([90]; i.e., corresponding to five class variables) class labels with BNs, learning the network for each neuron from a dataset of up to 42 instances (one for each neuroscientist) and five variables. They then predicted the labels of a neuron with an instance-based approach, that is, by combining the BNs labels of its neighbouring neurons. Fig. 7 shows examples of the true and predicted BNs labels. Besides high accuracy in predicting all the variables, encoding labels with BNs provided a representation of how the class variables interact at the single neuron level.

As discussed above, brain disorders impose a severe social and economic burden on modern societies. After Alzheimer's disease, the second most common neurodegenerative disorder is Parkinson's disease. Borchani et al. [138] used BNs to predict European Quality of Life-5 Dimensions (EQ-5D) from the 39-item Parkinson's Disease Questionnaire (PDQ-39). The EQ-5D is a generic health-related quality of life (HRQoL) measure consisting of five items — mobility, self-care, usual activities, pain/discomfort, and anxiety/depression— with three options for each item —no problems, some problems and severe problems. PDQ-39, on the other hand, is a Parkinson disease-specific HRQoL measure, containing 39 questions that capture the patient's perception of his or her illness across eight dimensions such as mobility and emotional well-being (see Fig. 8 for details). The authors used a dataset of EQ-5D and PDQ-39 questionnaires from 488 Parkinson's disease patients to learn a multi-dimensional BN classifier between the two HRQoL measures. The authors developed MB-MBC, an algorithm that learns the classifier by adapting the HITON [71,72] algorithm to simultaneously learn the Markov blankets of the class variables. The learned model structure (see Fig. 8) uncovered relationships among the EQ-5D items and PDQ-39 questions, as well as among the EQ-5D items themselves. Some PDQ-39 questions were irrelevant for predicting EQ-5D items and thus did not appear in the net-

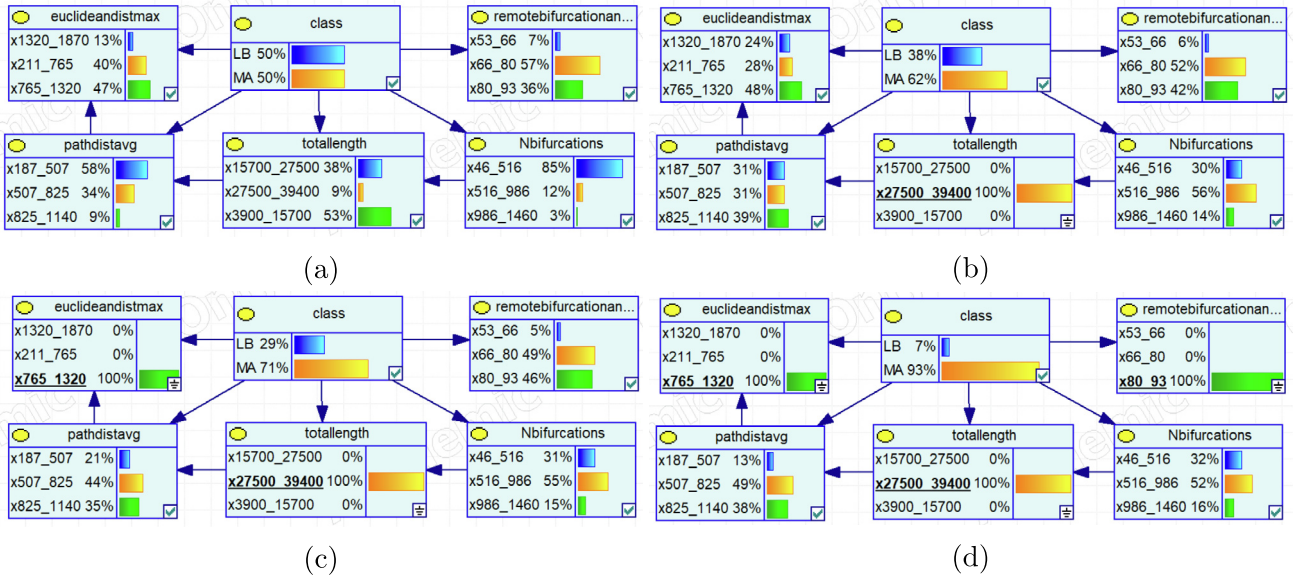


Fig. 6. Illustrating the classification of a neuron with a discrete tree augmented BN clasifier that distinguishes between the Martinotti (MA) and large basket (LB) interneuron types, learned from 101 of the 240 digital reconstructions used by DeFelipe et al. [123]. The class node is the interneuron type whereas the rest nodes correspond to predictor variables. Initially, without any evidence on the predictors, a given cell is equally likely to belong to either class (a). If we learn that a neuron has high total length (in the range 27500–39400 μm) and set that as evidence in the network (b), the probability of the neuron being a Martinotti cell increases to 62%. Subsequent observations of the maximal Euclidean distance to soma (c) and remote bifurcation angle (d) of the neuron further increase this probability, up to 93% in (d). Thus, the practitioner can understand the models' prediction and gain insight regarding the quantitative features of the two interneuron types. Figure from [135].

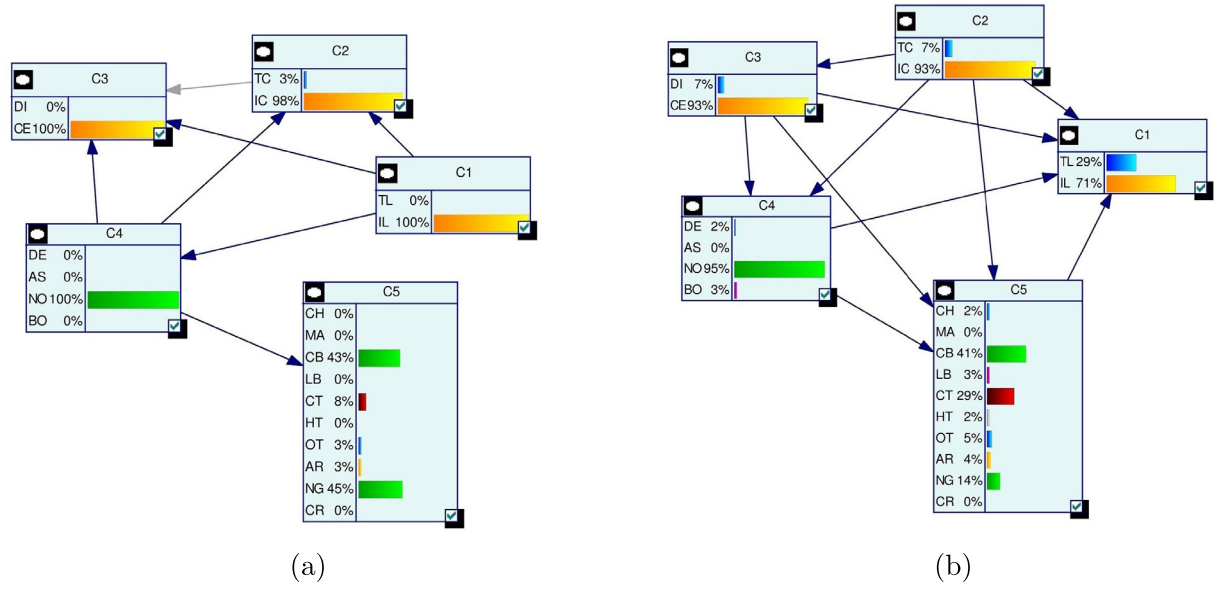


Fig. 7. Example of the true (a) and predicted (c) label BNs (LBNs) for one interneuron. The true network is learned from the 42 neuroscientist's labels for the interneuron. The predicted distributions are similar to the true ones for many nodes —e.g., 98% true vs. 93% predicted for `IC` ('intracolumnar', node C_2). Some marginal probabilities do differ, such as that of the `NG` (neurogliaform) type —45% true vs. 14% predicted; a lot of its probability mass was assigned to the more numerous `CT` 'common type' class.

work (Fig. 8). The model predicted 71% of the class labels correctly, outperforming multiple other methods.

Computational modeling of neuronal morphology is a useful tool for understanding neuronal development and examining relationships between morphology and neuronal function [139]. Since digital reconstructions of neurons are relatively scarce —e.g., human neurons are mainly extracted for study during surgeries and post-mortem— such models allow neuroscientists to reason, make predictions and suggest new hypotheses. Luengo-Sanchez et al. [140] used BNs to cluster and then simulate of morphologies of the human pyramidal somas. They characterized 39 somas with directional (i.e., involving angles) and linear (Gaussian) variables of their multiresolutional Reeb graph representations (Fig. 9). They defined a finite mixture-model based on the extended Mardia-Sutton density and then learned it with the structural expectation–maximization algorithm to maximize the BIC score. They found three clusters and used the RIPPER [141] algorithm to extract a set of rules that characterize each cluster. The BN clustering model identified a set of probabilistic dependencies among the variables, showing that, for example, the linear (Gaussian) vari-

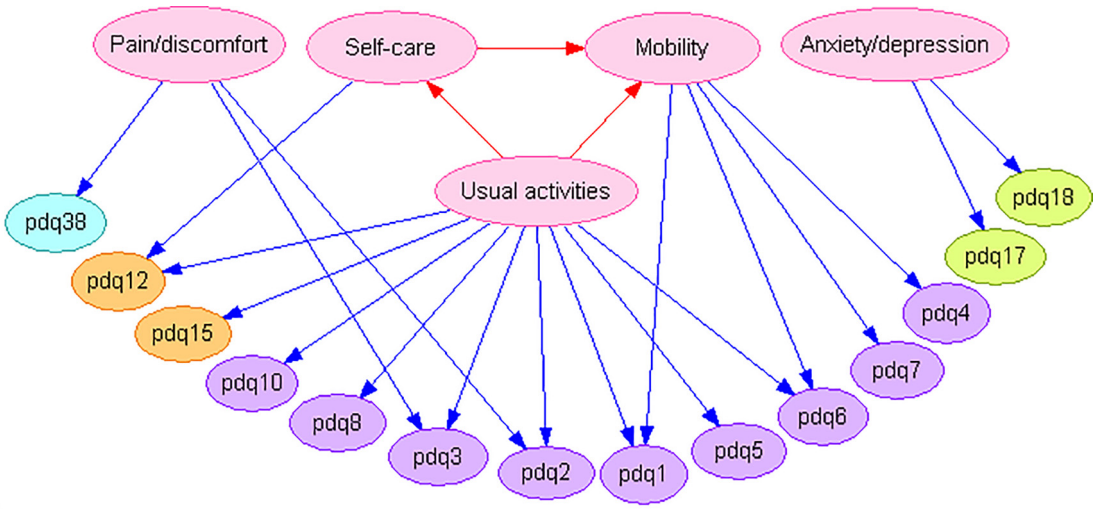


Fig. 8. A multi-dimensional BN classifier for mapping the PDQ-39 measures into EQ-5D ones. The EQ-5D variables are shown on top and the PDQ-39 ones below them, with their labels beginning with 'pdq'. While there were 39 PDQ-39 questions, grouped into eight domains —mobility, activities of daily living, emotional well-being, stigma, social support, cognitions, communication and bodily discomfort— only 14 of them —from domains mobility, activities of daily living, emotional well-being, and bodily discomfort— appear in the network, with each domain represented with a different color. The remaining questions were irrelevant for predicting EQ-5D and are thus omitted from the model. The arcs suggest dependencies between EQ-5D items and PDQ-39 questions. For example, EQ-5D mobility item is directly associated with nodes pdq1, pdq4, pdq6, and pdq7, all belong to questions in the mobility domain of the PDQ-39. There were also arcs between the EQ-5D items mobility, self-care and usual activities, revealing probabilisitc dependence among them. The lack of arcs among the PDQ-39 questions is due to restrictions imposed on the learning algorithm. Figure from [138].

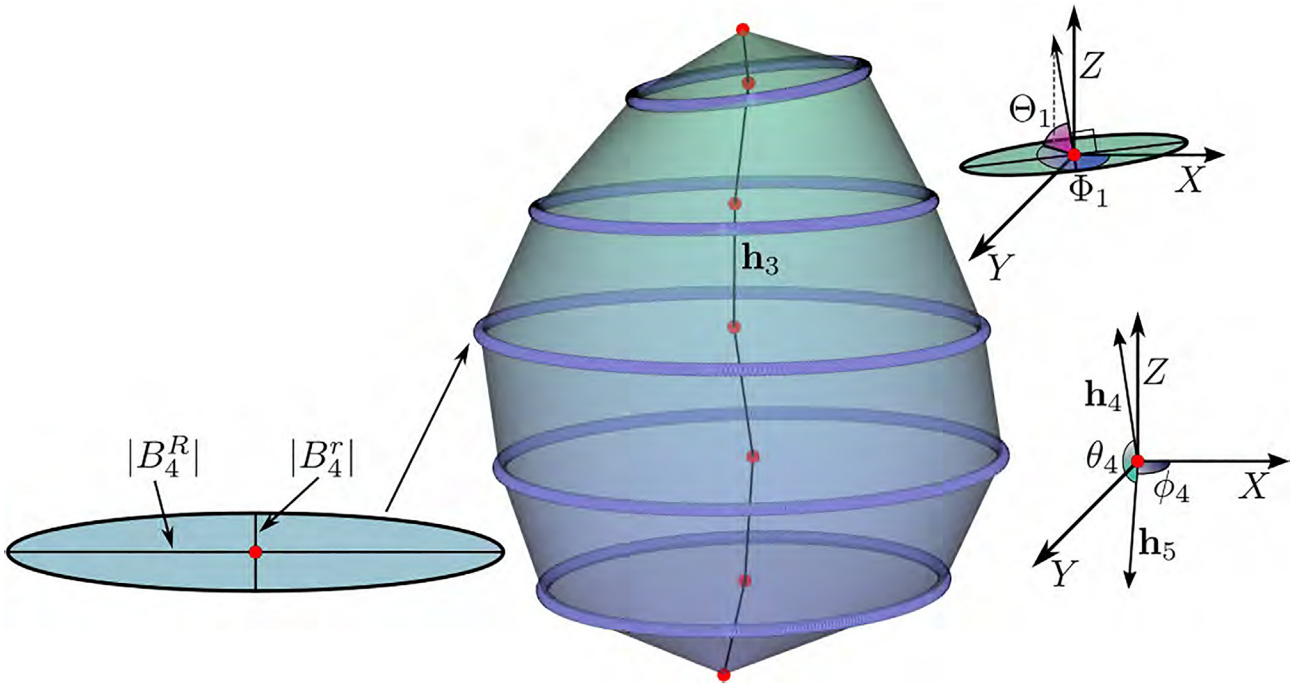


Fig. 9. Computation of linear and directional features. A set of ellipses is first identified, roughly separating the surface of the soma into regions with respect to the geodesic distance to the apical insertion point, located towards the top of the soma. Each ellipse B_i is defined by its centroid and major $|B_i^R|$ and minor $|B_i^r|$ axes. The height of each region is given by the length of the vector $\mathbf{h}_i$ between the centroids of the ellipses. Vectors $\mathbf{h}_i$ and $\mathbf{h}_{i+1}$ define a direction in spherical coordinates from which ϕ and θ_j are obtained. Φ_i and Θ_i are computed from the perpendicular vector to each ellipse B_i. Figure from [140].

ables were interrelated in consecutive regions of the soma (Fig. 10). The authors then used the model to simulate synthetic 3D somas from each cluster.

3.2. Industry

Larrañaga et al. [142] report an application of Bayesian netwoks to the automatic detection of possibly defective manufactured products for their immediate revision. In particular, they developed an automated visual inspection system for the quality control of the heat treatment of steel cylinders with laser beams. They learned the model from a set of images capturing the laser surface heat treatment of 32 steel cylinders. Since all 32 cylinders were correctly processed the authors used them to model the normal behavior of the system, tagging new processed units as possibly defective if they were anomalous according to this model.

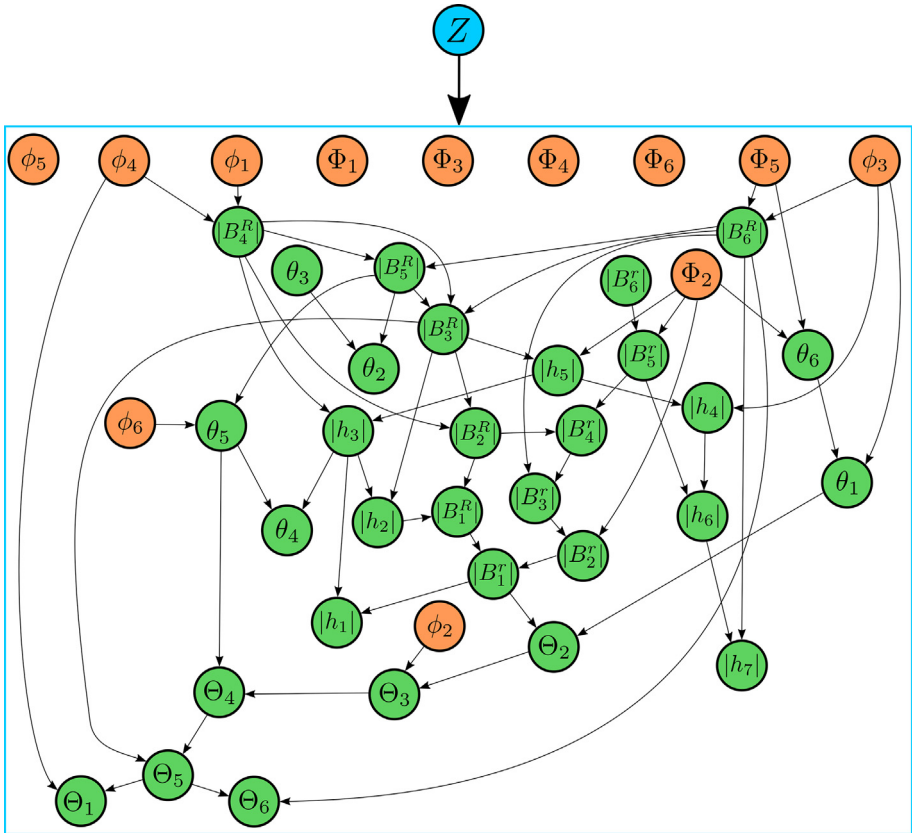


Fig. 10. The network structure for the clustering of somas. There iare 12 directional (orange) and 31 are linear (green) variables; note that linear nodes cannot be parents of directional nodes and thus there are no arcs from green to orange nodes. Angular variables θ and Θ are modelled as linear because they are restricted to the $[0, 2\pi]$ interval and are thus not circular. The latent (unobserved) variable Z (on top) encodes the assignments to clusters. To avoid cluttering the BN, arcs from Z the each variable are represented as a single arc from Z to an enclosing box. The structure shows that linear variables are interrelated in consecutive regions, such as $|B_4^r| \rightarrow |B_3^r| \rightarrow |B_2^r|_1$ and that curvature variables θ and Θ are mostly correlated with directional variables or other curvature variables. Figure from [140].

The data set contained 21,500 images (frames) for each processed cylinder and each image had 1024 pixels with values in the range 0 to 1023 (i.e., the range of colors encoded with 10 bits). The authors reduced the dimensionality of each image by grouping correlated and neighbouring pixels into clusters. They identified nine regions of interest and five background regions that were ignored in subsequent analysis (Fig. 11). They quantified the temperatures within each region of interest with four variables —the

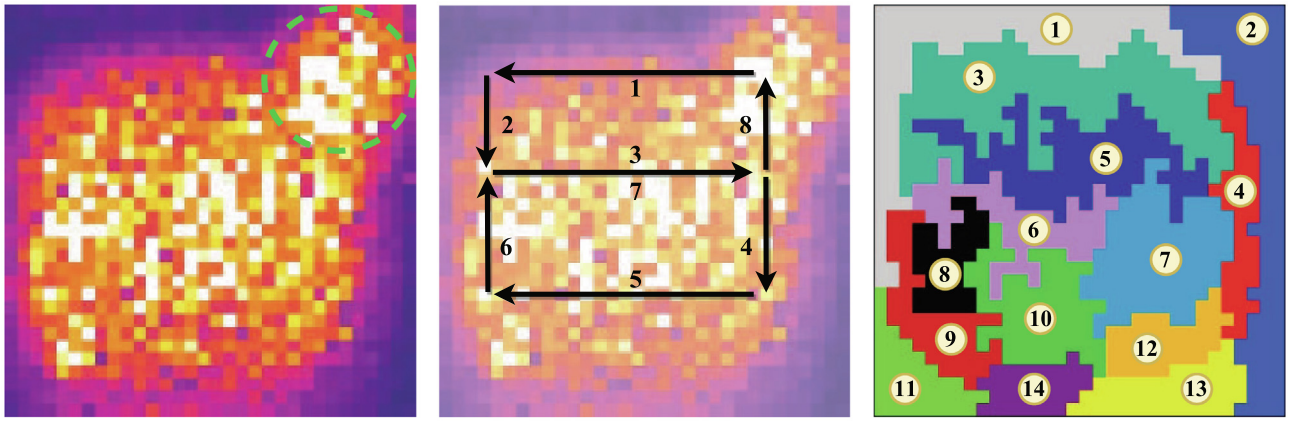


Fig. 11. The laser beam moved quickly according to a predefined pattern in order to heat the whole surface of the cylinder. This movement produced a heat-affected zone (HAZ) that was recorded by the high-speed thermal camera. The camera recorded 1,000 frames per second over a region of interest of 32 × 32 pixels, with up to 1024 different colors (10 bits per pixel) proportional to the temperature reading. A rotation of the surface of each cylinder took 21.5 s and a total of 21,500 frames were output for each processed cylinder. (a) The laser spot is noticeable at the top right of the image (green circle). (b) The spot was programmed to move along the steel surface according to a pattern. The numbers indicate the order in which the different segments of the pattern were formed. (c) The 14 regions into which the frame was segmented. The regions adjacent to the edges were considered to be background. Figure from [142].

median, the standard deviation, the maximum and the minimum—and then mapped the obtained values from the range 0 to 1023 into one of ten discrete intervals of width 102 between 0 and 1023.

The authors used BNs to model the density of the laser process. In particular, they used dynamic BNs to model the sequence of images over time, assuming a first-order Markovian transition model. They used the dynamic hill-climbing algorithm to maximize the BIC score of the prior and transition networks. They imposed a number of constraints on the learning algorithm, such as having at most two parents for each variable and only allowing arcs between variables of the same type (e.g., medians with medians) across regions. All future image sequences with a larger negative log-likelihood than any of those observed in the training data were then to be considered as anomalous, and thus possibly defective, according to the model.

The authors evaluated the method by simulating two different types of defects in the 32 normal sequences: (a) defect in the laser power supply unit, simulated with negative offsets of 3.5% and 4% to the pixel colors; and (b) camera sensor wear, simulated with added Gaussian noise to the pixel values. The method correctly

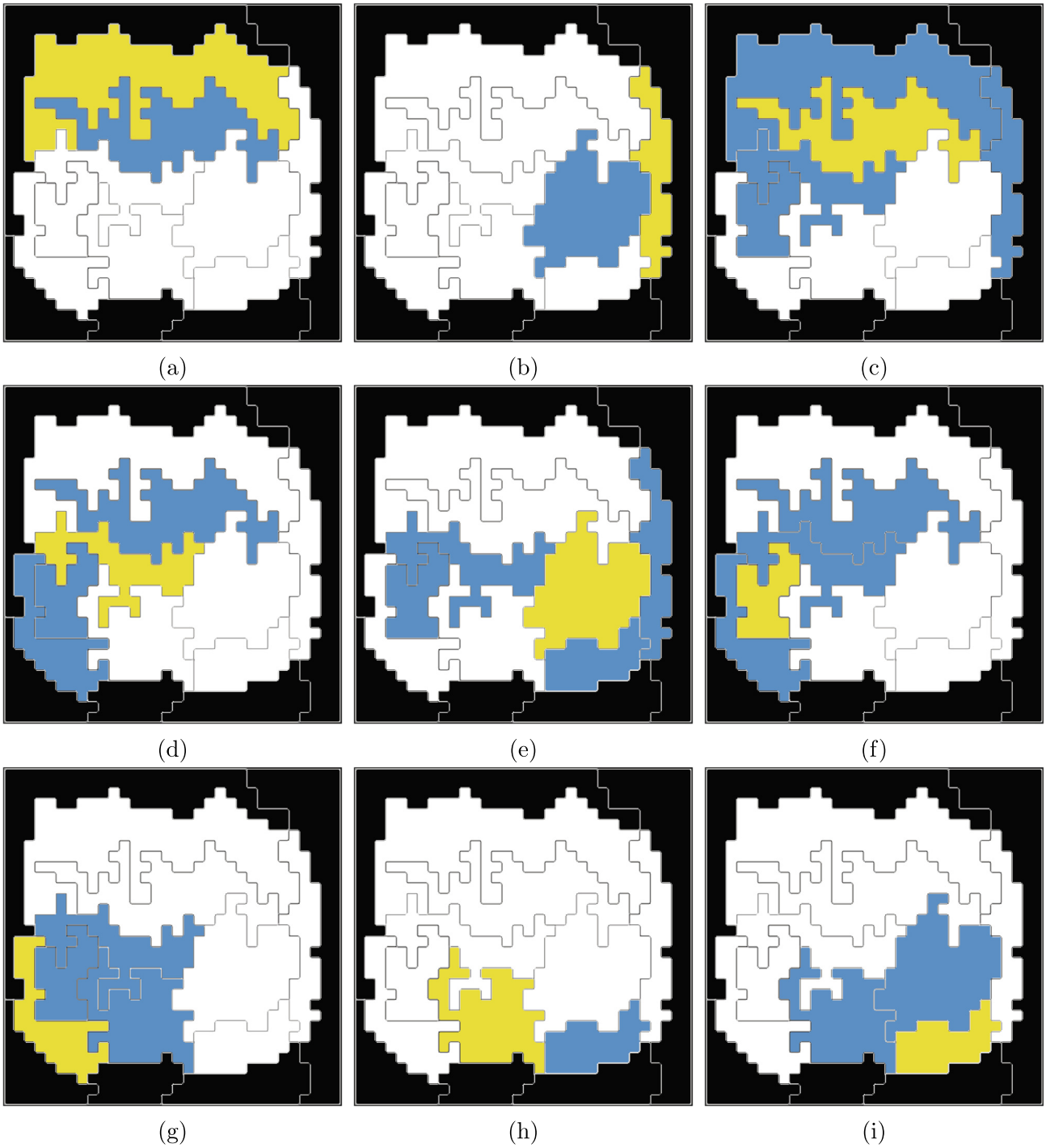


Fig. 12. Illustration of the regions with variables within the Markov blanket (in blue) of the variables of the target region (in yellow). Knowledge of the state of these regions shielded the target region from the influence of other regions (in white). As expected, both the regions and their Markov blanket regions were close. Markov blanket of (a) region 3, (b) region 4, (c) region 5, (d) region 6, (e) region 7, (f) region 8, (g) region 9, (h) region 10 and (i) region 12. Figure from [142].

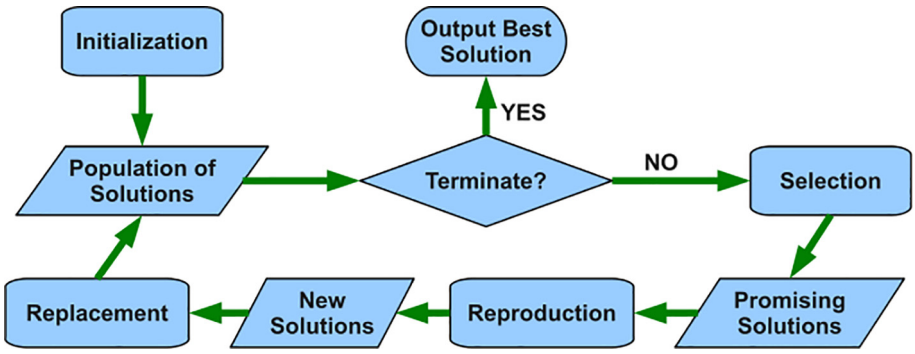


Fig. 13. The flowchart of a typical evolutionary algorithm. Figure from [149].

classified 93.8% of the normal sequences, 78.1% of the anomalies with a negative offset of 3.5%, 100% of the anomalies with a negative offset of 4%, and 100% of the camera noise anomalies.

The authors then analysed the transition network to identify the spatio-temporal properties of the thermal process that were learned from data. They found that the median, maximum and minimum were persistent (connected in different time slices) variables in 85.2% of the cases, which was particularly important for the median of the regions as the temperature of the HAZ ought to be stable at a high enough value in order to reach the austenite phase. Network centrality measures, such as the outdegree and the reversed PageRank [143,144], indicated that the median was the most influential type of variable. The authors identified the Markov blankets of each region (see Fig. 12) —defined as the union of the Markov blankets of the region's four variables— thus identifying nearby regions that affected the state of a particular region while making it independent from the states of remaining regions. Thus, a separate characterization of a region could not suffice to model the thermal properties of the process since there were spatio-temporal dependencies among regions, induced by the movement of the laser beam.

4. Bayesian networks in optimization

Optimization problems, such as finding optimal routes for the vehicles of a transportation company, are common in many domains. They can be cast as the minimization (or maximization) of one or more functions, subject to a set of constraints. Many relevant scientific and industrial problems are too complex to be solved optimally and we can, at best, hope to find a good solution. This may be done by intelligently searching the enormous space of possible solutions with meta-heuristic algorithms [38]. One group of such meta-heuristics are evolutionary algorithms. These algorithms follow a framework inspired by natural evolution (Fig. 13). Namely, given a fitness function that evaluates the quality of a solution, the algorithm iteratively evolves a population of candidate solutions. Offspring solutions are produced from the fitter solutions of the population (survival of the fittest), by combining them (crossover); the offspring then may be randomly altered (mutation). Usually, the solutions improve over time and search is stopped at some point, returning the best solution up to that point. Examples of evolutionary algorithms include genetic algorithms [145], evolutionary strategies [146], evolutionary programming [147] and genetic programming [148]. While most evolutionary algorithms tend to identify, preserve and effectively combine partial solutions during the evolution, they may be limited when certain characteristics are present in the problem. In particular, the traditional operators such as crossover often do not properly account for the dependencies among the variables of the problem, thus ignoring information that could speed up convergence. Non-linearity, ill-conditioning and deception can pose significant challenges unless such dependencies are taken into account.

Probabilistic modeling can account for such dependencies in order to improve the speed and accuracy of problem solving [150,151]. Instead of traditional genetic operators, a new generation of candidate solutions is generated as follows: (1) estimating a probabilistic model based on a set of candidate solutions (usually the fitter ones); and (2) sampling the new generation from the learned probabilistic model. The class of evolutionary algorithms based on probabilistic modeling is referred to as estimation of distribution algorithms (EDAs) [39–42]. As an effective optimization technique, they have been widely applied to complex optimization problems [e.g.,][152]. Below we describe the basics of EDAs as well as two applicatons.

4.1. Estimation of distribution algorithms

Fig. 14 shows the basic scheme of an EDA. At iteration t the algorithm selects a set of solutions S_t from which to learn the probabilistic model $\hat{p}_t(\boldsymbol{x})$. Since S_t usually consists of fitter solutions, $\hat{p}_t(\boldsymbol{x})$ is thus an explicit model of promising regions of the search space. The new generation of solutions U_t is generated by sampling from $\hat{p}_t(\boldsymbol{x})$.

EDA algorithms are commonly grouped according to the degree of interaction among variables into the: (a) univariate, (b) bivariate, and (c) multivariate EDAs. Univariate EDAs, such as PBIL [153], cGA [154] and UMDA [39], assume that all variables are independent and factorize the JPD as a product of univariate mar-

Input:
Representation of solutions
Objective function f

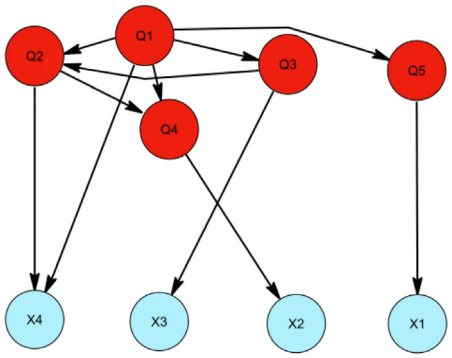
$P_0 \leftarrow$ Generate initial population according to the representation
$F_0 \leftarrow$ Evaluate each individual x in P_0 using f
$t \leftarrow 0$
while termination criteria are not met **do**
  $S_t \leftarrow$ Select a subset of P_t according to F_t
  $\hat{p}_t(\boldsymbol{x}) \leftarrow$ Estimate the probability density of solutions in S_t
  $U_t \leftarrow$ Sample from $\hat{p}_t(\boldsymbol{x})$ according to the representation
  $H_t \leftarrow$ Evaluate U_t using f
  $P_{t+1} \leftarrow$ Incorporate U_t into P_t according to F_t and H_t
  $F_{t+1} \leftarrow$ Update F_t according to the solutions in P_{t+1}
  $t \leftarrow t+1$
end while

Output: The best solution(s) in P_t

Fig. 14. The basic steps of an EDA.

$$
\begin{aligned}
Q_1(\boldsymbol{x}) &= a + 2 \cdot h_1\left(g_2\left(x_1\right), g_2\left(x_2\right), g_2\left(x_3\right)\right), \\
Q_2(\boldsymbol{x}) &= a + 4 \cdot h_2\left(g_2\left(x_1\right), g_2\left(x_2\right), g_2\left(x_3\right)\right), \\
Q_3(\boldsymbol{x}) &= a + 6 \cdot h_3\left(g_2\left(x_1\right), g_2\left(x_2\right), g_2\left(x_3\right)\right) \\
Q_4(\boldsymbol{x}) &= a + 8 \cdot h_4\left(g_2\left(x_1\right), g_2\left(x_2\right)\right) \\
Q_5(\boldsymbol{x}) &= a + 10 \cdot h_5\left(g_2\left(x_1\right)\right) \\
a &= g_1\left(x_5, \ldots, x_{16}\right)
\end{aligned}
$$

(a)



(b)

Fig. 15. Joint modeling of objectives and variables for the 5-objective WFG1 optimization problem [164]. (a) The formulas give a simplified definition of the five objective functions. There are five objectives and 16 variables in the problem, with four of them especially relevant. Namely, the first four variables determine the position of a solution in the objective space via shape functions h_1 to h_4, and this position is then added to a distance parameter a computed from the last 12 variables. (b) Part of the learned network structure, showing the most significant arcs and their corresponding nodes. The objectives are shown above, in red, and the variables below, in light blue. For example, the model correctly identified that all four variables influence the value of Q_1, as Q_1 is not marginally independent of any of the variables X_1 to X_4. On the other hand, Q_1 is independent of variables X_1 to X_3 given the other objectives and variable X_4. Figure from [162].

ginal distributions. Bivariate EDAs, such as MIMIC [155], represent pairwise dependencies between variables, for example with a chain BN where all variables but one are conditioned on the preceding variable in the chain. While univariate and bivariate models can be efficiently and reliably estimated from data, they may be too simple in some cases. Multivariate EDAs do not necessarily limit the degree of interactions among variables and can be modelled with unrestricted BNs. Early examples include EBNA [156] and BOA [157].

Early EDAs were developed for discrete domains, as it is common in evolutionary algorithms to represent solutions with bit strings. The most common approach for handling continuous variables is to model them as Gaussian [158].

4.2. Uni-objective problems

[159] studied whether the characteristics of the networks learned during the running of an EDA are informative about the algorithm's behaviour and the problem's characteristics. In particular, they studied whether network metrics such as node eccentricity (maximal shortest path length between a pair of nodes) and edge betweenness centrality (fraction of all shortest paths that traverse a given edge) could predict the algorithm's convergence and distinguish between problems with many nearly optimal soluctions and those with few of them. They tested the hypotheses on three groups of synthetic optimization problems, one of which was that of protein folding of the simplified HP protein model, a combinatorial problem consisting in finding a simplified protein model configuration that minimizes an energy representing the interaction between hydrophobic (H) and polar (P) residues. From a dataset of 611 proteins with different folding sequences [160] and the corresponding network metrics obtained after running EDAs on each of them, the authors trained supervised classifiers to predict (a) whether the EDA will converge in 30 iterations; and (b) whether there are few or many nearly optimal solutions. The classifiers had a 71% accuracy for predicting convergence and 91% for predicting the existance of many near optimal solutions, showing that indeed the network metrics were informative.

4.3. Multi-objective problems

Many optimization problems involve multiple, and often conflictive, objectives. For example, when designing a product, a company might want to maximize its quality while minimizing its environmental impact. The optimal solution for a multi-objective optimization problem is not a single solution but a set of Pareto optimal solutions. A solution is Pareto optimal if no other solution improves a given objective without degrading at least one other objective. When applying meta-heuristics, the goal is to approximate the Pareto optimal set with a uniform diversity across the set. Since the objectives' values may give only a partial ordering over the solutions, properties such as diversity are usually taken into account to rank the solutions (e.g., by considering distances in the objective space [161]).

[162] proposed an EDA that models the JPD of objectives and variables with a BN. This allows capturing not only the dependencies between variables but also (a) among the objectives; and (b) between the objectives and the variables. One could also use BN inference to, for example, find the most probable solution given a specific setting for the objectives. The proposed model is analogous to a multi-dimensional BN classifier (see Section 2.7), with the objectives corresponding to class variables (and thus having no variable parents in the graph) and the variables to predictor variables. The authors used a greedy hill-climbing search with random restarts to maximize the BIC score, and assumed that both the variables and the objectives were Gaussian variables, estimating parameters with covariance shrinkage [163]. Extensive comparison to related state-of-the-art algorithms showed that the algorithm found significantly better approximations to the Pareto set for many of the considered problems. An analysis of the structures learned during evolution showed that the algorithm was able to distinguish between relevant and irrelevant variables for the different objectives, and also to identify dependencies between similar objectives. Fig. 15 illustates how the algorithm recovered a good approximation of a synthetic problem with a known structure. Since multi-objetive optimization involves trade-offs between objectives that often requires decision-making, the information uncovered by the model can be valuable to the decision-maker.

5. Conclusions

As artificial intelligence is being increasingly used for high-stakes applications, it is becoming more and more important that the models used be interpretable. Bayesian networks offer a paradigm for interpretable artificial intelligence based on probability theory. They provide a semantics that enables a compact, declara-

tive representation of a joint probability distribution over the variables of a domain by leveraging the conditional independencies among them. The representation consists of a directed acyclic graph that encodes the conditional independencies among the variables and a set of parameters that encodes conditional distributions. This representation has provided a basis for the development of algorithms for probabilistic reasoning (inference) and for learning probability distributions from data. Bayesian networks are used for a wide range of tasks in machine learning, including clustering, supervised classification, multi-dimensional supervised classification, anomaly detection, and temporal modeling. They also provide a basis for estimation of distribution algorithms, a class of evolutionary algorithms for heuristic optimization.

We have illustrated the use of Bayesian networks for interpretable machine learning and optimization by presenting applications in neuroscience, the industry, and bioinformatics, covering a wide range of machine learning and optimization tasks.

Funding

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness through the TIN2016-79684-P project and the Spanish Ministry of Science, Innovation and Universities through the PID2019-109247GB-I00 project, by the BBVA Foundation (2019 Call) through the "Score-based nonstationary temporal Bayesian networks. Applications in climate and neuroscience" project, the BBVA Foundation's grants (2020 Call) for Scientific Investigation Teams SARS-CoV-2 and COVID-19 through the "Outcome prediction and treatment efficiency in patients hospitalized with Covid-19 in Madrid: A Bayesian network approach"project, and European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement N. 945539 (Human Brain Project SGA3).

CRediT authorship contribution statement

Bojan Mihaljević: Conceptualization, Writing - original draft, Writing - review & editing. **Concha Bielza:** Writing - review & editing. **Pedro Larrañaga:** Conceptualization, Writing - original draft, Writing - review & editing.

Declaration of competing interest

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

References

[1] M.I. Jordan, T.M. Mitchell, Machine learning: Trends, perspectives, and prospects, Science 349 (2015) 255–260.
[2] D. Castelvecchi, Can we open the black box of AI?, Nat. News 538 (2016) 20.
[3] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nat. Mach. Intell. 1 (2019) 206–215.
[4] B. Goodman, S. Flaxman, European union regulations on algorithmic decision-making and a "right to explanation", AI Mag. 38 (2017) 50–57.
[5] A. Jobin, M. Ienca, E. Vayena, The global landscape of AI ethics guidelines, Nat. Mach. Intell. 1 (2019) 389–399.
[6] E. Tjoa, C. Guan, A survey on explainable artificial intelligence (XAI): Toward medical XAI, IEEE Trans. Neural Networks Learn. Syst. (2020).
[7] D. Szafron, P. Lu, R. Greiner, D.S. Wishart, B. Poulin, R. Eisner, Z. Lu, J. Anvik, C. Macdonell, A. Fyshe, et al., Proteome analyst: Custom predictions with explanations in a web-based tool for high-throughput proteome annotations, Nucl. Acids Res. 32 (2004) W365–W371.
[8] V. Dhar, D. Chou, F. Provost, Discovering interesting patterns for investment decision making with GLOWER: A genetic learner overlaid with entropy reduction, Data Min. Knowl. Disc. 4 (2000) 251–280.
[9] J. Zhu, A. Liapis, S. Risi, R. Bidarra, G.M. Youngblood, Explainable AI for designers: A human-centered perspective on mixed-initiative co-creation, in: 2018 IEEE Conference on Computational Intelligence and Games, IEEE, pp. 1–8..
[10] K. Sachs, O. Perez, D. Pe'er, D.A. Lauffenburger, G.P. Nolan, Causal protein-signaling networks derived from multiparameter single-cell data, Science 308 (2005) 523–529.
[11] D. Gunning, Explainable artificial intelligence (XAI), Technical Report DARPA-BAA-16-53, DARPA, 2016..
[12] W. Samek, T. Wiegand, K.-R. Müller, Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models, arXiv preprint arXiv:1708.08296 (2017)..
[13] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (2018) 1–42.
[14] Z.C. Lipton, The mythos of model interpretability, Queue 16 (2018) 31–57.
[15] F. Doshi-Velez, B. Kim, Towards a rigorous science of interpretable machine learning, arXiv preprint arXiv:1702.08608 (2017)..
[16] A.B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al., Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fus. 58 (2020) 82–115.
[17] W.J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, B. Yu, Definitions, methods, and applications in interpretable machine learning, Proc. Nat. Acad. Sci. 116 (2019) 22071–22080.
[18] A. Preece, D. Harborne, D. Braines, R. Tomsett, S. Chakraborty, Stakeholders in explainable AI, arXiv preprint arXiv:1810.00184 (2018)..
[19] M.T. Ribeiro, S. Singh, C. Guestrin, Why should I trust you? Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144..
[20] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, B. Baesens, An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models, Decis. Support Syst. 51 (2011) 141–154.
[21] A.A. Freitas, Comprehensible classification models: A position paper, ACM SIGKDD Explorations Newsletter 15 (2014) 1–10.
[22] D.J. Hand, Classifier technology and the illusion of progress, Stat. Sci. (2006) 1–14.
[23] E.T. Jaynes, Probability Theory: The Logic of Science, Cambridge University Press, 2003.
[24] J. Pearl, Probabilistic Reasoning in Intelligent Systems, Morgan Kaufmann, 1988.
[25] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, MIT Press, 2009.
[26] M. Maathuis, M. Drton, S. Lauritzen, M. Wainwright, Handbook of Graphical Models, CRC Press, 2019.
[27] D. Heckerman, E. Horvitz, K. Ng, B. Nathwani, Towards normative expert systems: Part I, Pathfinder Project, Methods Inf. Med. 31 (1992) 90–105.
[28] S. Andreassen, F.V. Jensen, S.K. Andersen, B. Falck, U. Kjærulff, M. Woldbye, A. Sørensen, A. Rosenfalck, F. Jensen, MUNIN: An expert EMG Assistant, in: Computer-aided Electromyography and Expert Systems, Pergamon Press, 1989, pp. 255–277.
[29] M.A. Shwe, B. Middleton, D.E. Heckerman, M. Henrion, E.J. Horvitz, H.P. Lehmann, G.F. Cooper, Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base, Methods Inf. Med. 30 (1991) 241–255.
[30] A. Rajkomar, J. Dean, I. Kohane, Machine learning in medicine, New Engl. J. Med. 380 (2019) 1347–1358.
[31] A. Dannenberg, A. Shapiro, J. Fries, Enhancement of clinical predictive ability by computer consultation, Methods Inf. Med. 18 (1979) 10–14.
[32] R.L. Teach, E.H. Shortliffe, An analysis of physician attitudes regarding computer-based clinical consultation systems, Comput. Biomed. Res. 14 (1981) 542–558.
[33] B.G. Buchanan, E.H. Shortliffe, Rule-based expert systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley, 1984.
[34] W.J. Clancey, Use of MYCIN's rules for tutoring, Rule-Based Expert Systems. Addison-Wesley, Reading 20 (1984).
[35] A. Tversky, D. Kahneman, Judgment under uncertainty: Heuristics and biases, Science 185 (1974) 1124–1131.
[36] P.C. Wason, P.N. Johnson-Laird, Psychology of Reasoning: Structure and Content, volume 86, Harvard University Press, 1972.
[37] C. Lacave, F.J. Díez, A review of explanation methods for Bayesian networks, Knowledge Eng. Rev. 17 (2002) 107–127.
[38] E.-G. Talbi, Metaheuristics, Wiley, From Design to Implementation, 2009.
[39] H. Mühlenbein, G. Paaß, From recombination of genes to the estimation of distributions. I. Binary parameters, in: Lecture Notes in Computer Science 1411: Parallel Problem Solving from Nature, pp. 178–187..
[40] P.A. Bosman, D. Thierens, Linkage information processing in distribution estimation algorithms, in: Proceedings of the Genetic and Evolutionary Computation Conference, volume I, pp. 60–67..
[41] P. Larrañaga, J.A. Lozano (Eds.), Estimation of Distribution Algorithms: A new Tool for Evolutionary Computation, Springer, 2001.
[42] J.A. Lozano, P. Larrañaga, I. Inza, E. Bengoetxea (Eds.), Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms, volume 192, Springer, 2006.

[43] C. Bielza, P. Larrañaga, Data-Driven Computational Neuroscience: Machine Learning and Statistical Models, Cambridge University Press, 2020.
[44] G.F. Cooper, The computational complexity of probabilistic inference using Bayesian belief networks, Artif. Intell. 42 (1990) 393–405.
[45] S.E. Shimony, Finding MAPs for belief networks is NP-hard, Artif. Intell. 68 (1994) 399–410.
[46] S. Arnborg, D. Corneil, A. Proskurowski, Complexity of finding embeddings in a k-tree, SIAM J. Algebraic Discrete Methods 8 (1987) 277–284.
[47] S. Lauritzen, D. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems, J. R. Stat. Soc. Ser. B (Methodol.) 50 (1988) 157–224.
[48] A. Darwiche, A differential approach to inference in Bayesian networks, J. ACM 50 (2003) 280–305.
[49] P. Dagum, M. Luby, Approximating probabilistic inference in Bayesian belief networks is NP-hard, Artif. Intell. 60 (1993) 141–153.
[50] M. Henrion, Propagating uncertainty in Bayesian networks by probabilistic logic sampling, in: Uncertainty in Artificial Intelligence 2, pp. 149–163..
[51] R. Fung, K.-C. Chang, Weighing and integrating evidence for stochastic simulation in Bayesian networks, in: Uncertainty in Artificial Intelligence, North-Holland, 1990, pp. 209–219..
[52] R. Shachter, M. Peot, Simulation approaches to general probabilistic inference on belief networks, in: Uncertainty in Artificial Intelligence 5, North-Holland, 1990, pp. 221–231..
[53] B. Mihaljević, P. Larrañaga, R. Benavides-Piccione, J. DeFelipe, C. Bielza, Comparing basal dendrite branches in human and mouse hippocampal CA1 pyramidal neurons with Bayesian networks, Scientific Reports 10 (2020) 18592.
[54] D. Heckerman, D.M. Chickering, C. Meek, R. Rounthwaite, C. Kadie, Dependency networks for inference, collaborative filtering, and data visualization, J. Mach. Learn. Res. 1 (2000) 49–75.
[55] M. Henrion, M.J. Druzdzel, Uncertainty in Artificial Intelligence, Elsevier, 1991, pp. 17–32..
[56] I. Good, Weight of evidence: A brief survey, with discussion, Bayesian Stat. 2 (1985) 249–270.
[57] D. Madigan, K. Mosurski, R.G. Almond, Graphical explanation in belief networks, Journal of Computational and Graphical Statistics 6 (1997) 160–181.
[58] G.F. Cooper, NESTOR: A Computer-Based Medical Diagnostic Aid That Integrates Causal and Probabilistic Knowledge, Ph.D. thesis, Stanford, 1984..
[59] H.J. Suermondt, Explanation in Bayesian Belief Networks, Ph.D. thesis, Stanford, 1993..
[60] P. Lipton, Inference to the Best Explanation, Routledge, 2003.
[61] S.E. Shimony, Explanation, irrelevance and statistical independence, in: Proceedings of the Ninth National Conference on Artificial intelligence-Volume 1, pp. 482–487..
[62] J. Kwisthout, Most frugal explanations in Bayesian networks, Artif. Intell. 218 (2015) 56–73.
[63] M.J. Flores, J.A. Gámez, S. Moral, Abductive inference in Bayesian networks: Finding a partition of the explanation space, in: European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty, Springer, pp. 63–75..
[64] C. Yuan, H. Lim, T.-C. Lu, Most relevant explanation in Bayesian networks, J. Artif. Intell. Res. 42 (2011) 309–352.
[65] J.N. Schupbach, J. Sprenger, The logic of explanatory power, Phil. Sci. 78 (2011) 105–127.
[66] V. Crupi, K. Tentori, A second look at the logic of explanatory power, with two novel representation theorems, Phil. Sci. 79 (2012) 365–385.
[67] L.M. De Campos, J.A. Gámez, S. Moral, Simplifying explanations in Bayesian belief networks, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9 (2001) 461–489.
[68] D.H. Glass, Coherence, explanation, and hypothesis selection, The British Journal for the Philosophy of Science (2018). Axy063..
[69] C. Glymour, Probability and the explanatory virtues, British J. Phil. Sci. 66 (2015) 591–604.
[70] P. Spirtes, C. Glymour, An algorithm for fast recovery of sparse causal graphs, Soci. Sci. Comput. Rev. 90 (1991) 62–72.
[71] C. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, X. Koutsoukos, Local causal and Markov blanket induction for causal discovery and feature selection for classification, Part I: Algorithms and empirical evaluation, J. Mach. Learn. Res. 11 (2010) 171–234.
[72] C. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, X. Koutsoukos, Local causal and Markov blanket induction for causal discovery and feature selection for classification, Part II: Analysis and extensions, J. Mach. Learn. Res. 11 (2010) 235–284.
[73] D. Margaritis, Learning Bayesian Network Model Structure from Data, Ph.D. thesis, Carnegie-Mellon University, 2003..
[74] F. Glover, M. Laguna, Tabu Search, in: P.M. Pardalos, D.-Z. Du, R.L. Graham (Eds.), Handbook of Combinatorial Optimization, Springer, 2013, pp. 3261–3362.
[75] P. Larrañaga, M. Poza, Y. Yurramendi, R. Murga, C. Kuijpers, Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters, IEEE Trans. Pattern Anal. Mach. Intell. 18 (1996) 912–926.
[76] G. Schwarz, Estimating the dimension of a model, Ann. Stat. 6 (1978) 461–464.
[77] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. Ser. B (Methodol.) 39 (1977) 1–38.
[78] C. Bielza, P. Larrañaga, Discrete Bayesian network classifiers: A survey, ACM Comput. Surv. 47 (2014).
[79] B. Mihaljević, C. Bielza, P. Larrañaga, bnclassify: Learning Bayesian network classifiers, R J. 10 (2018) 455–468.
[80] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Mach. Learn. 29 (1997) 131–163.
[81] M. Minsky, Steps toward artificial intelligence, Trans. Inst. Radio Engrs. 49 (1961) 8–30.
[82] M. Sahami, Learning limited dependence Bayesian classifiers, in: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, volume 96, pp. 335–338..
[83] F. Pernkopf, P. O'Leary, Floating search algorithm for structure learning of Bayesian network classifiers, Pattern Recogn. Lett. 24 (2003) 2839–2848.
[84] M. Jaeger, Probabilistic classifiers and the concept they recognize, in: Proceedings of the 20th International Conference on Machine Learning ICML, 2003, pp. 266–273.
[85] G. Varando, C. Bielza, P. Larrañaga, Decision boundary for discrete Bayesian network classifiers, J. Mach. Learn. Res. 16 (2015) 2725–2749.
[86] E.J. Keogh, M.J. Pazzani, Learning the structure of augmented Bayesian classifiers, Int. J. Artif. Intell. Tools 11 (2002) 587–601.
[87] M. Pazzani, Constructive induction of Cartesian product attributes, in: Proceedings of the Information, Statistics and Induction in Science Conference, pp. 66–77..
[88] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[89] C. Chow, C. Liu, Approximating discrete probability distributions with dependence trees, IEEE Trans. Inf. Theory 14 (1968) 462–467.
[90] C. Bielza, G. Li, P. Larra naga, Multi-dimensional classification with Bayesian networks, Int. J. Approximate Reasoning 52 (2011) 705–727.
[91] S. Gil-Begue, C. Bielza, P. Larrañaga, Multi-dimensional Bayesian network classifiers: A survey, Artif. Intell. Rev. (2020) 1–41.
[92] T. Dean, K. Kanazawa, A model for reasoning about persistence and causation, Comput. Intell. 5 (1989) 142–150.
[93] K.P. Murphy, Dynamic Bayesian Networks: Representation, Inference and Learning, Ph.D. thesis, University of California at Berkeley, 2002..
[94] N. Friedman, K. Murphy, S. Russell, Learning the structure of dynamic probabilistic networks, in: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, pp. 139–147..
[95] F. Taroni, C. Aitken, P. Garbolino, A. Biedermann, Bayesian Networks and Probabilistic Inference in Forensic Science, Wiley, 2006.
[96] R. Neapolitan, Learning Bayesian networks, Prentice Hall, 2004.
[97] J. Cai, Y. Xiao, Bearing fault diagnosis method based on the generalized S transform time–frequency spectrum de-noised by singular value decomposition, Proc. Inst. Mech. Eng. Part C: J. Mech. Eng. Sci. 233 (2019) 2467–2477.
[98] C. Bielza, P. Larrañaga, Bayesian networks in neuroscience: A survey, Front. Comput. Neurosci. 8 (2014) 131.
[99] D. Silberberg, N.P. Anand, K. Michels, R.N. Kalaria, Brain and other nervous system disorders across the lifespan – global challenges and opportunities, Nature 527 (2015) S151–S154.
[100] E.R. Walker, R.E. McGee, B.G. Druss, Mortality in mental disorders and global disease burden implications: A systematic review and meta-analysis, JAMA Psychiatry 72 (2015) 334–341.
[101] J. Olesen, Q. Gustavsson, M. Svensson, H. Wittchen, B. Jonson, The economic cost of brain disorders in Europe, Eur. J. Neurol. 19 (2012) 155–162.
[102] M.D. Hurd, P. Martorell, A. Delavande, K.J. Mullen, K.M. Langa, Monetary costs of dementia in the United States, N. Engl. J. Med. 368 (2013) 1326–1334.
[103] Z.J. Huang, L. Luo, It takes the world to understand the brain, Science 350 (2015) 42–44.
[104] H. Markram, The Human Brain Project, Sci. Am. 306 (2012) 50–55.
[105] K. Amunts, C. Ebell, J. Muller, M. Telefont, A. Knoll, T. Lippert, The human brain project: Creating a European research infrastructure to decode the human brain, Neuron 92 (2016) 574–581.
[106] T.R. Insel, S.C. Landis, F.S. Collins, The NIH BRAIN initiative, Science 340 (2013) 687–688.
[107] S. Grillner, N. Ip, C. Koch, W. Koroshetz, H. Okano, M. Polachek, M. Poo, T.J. Sejnowski, Worldwide initiatives to advance brain research, Nat. Neurosci. 19 (2016) 1118–1122.
[108] G.A. Ascoli, L. Alonso-Nanclares, S.A. Anderson, G. Barrionuevo, R. Benavides-Piccione, A. Burkhalter, G. Buzsáki, B. Cauli, J. DeFelipe, A. Fairén, D. Feldmeyer, G. Fishell, Y. Fregnac, T.F. Freund, D. Gardner, E.P. Gardner, J.H. Goldberg, M. Helmstaedter, S. Hestrin, F. Karube, Z.F. Kisvárday, B. Lambolez, D.A. Lewis, O. Marin, H. Markram, A. Muñoz, A. Packer, C.C.H. Petersen, K.S. Rockland, J. Rossier, B. Rudy, P. Somogyi, J.F. Staiger, G. Tamas, A.M. Thomson, M. Toledo-Rodriguez, Y. Wang, D.C. West, R. Yuste, Petilla terminology: Nomenclature of features of GABAergic interneurons of the cerebral cortex, Nat. Rev. Neurosci. 9 (2008) 557–568.
[109] J. DeFelipe, Chandelier cells and epilepsy, Brain 122 (1999) 1807–1822.
[110] R.F. Hunt, K.M. Girskis, J.L. Rubenstein, A. Alvarez-Buylla, S.C. Baraban, GABA progenitors grafted into the adult epileptic brain control seizures and abnormal behavior, Nat. Neurosci. 16 (2013) 692.
[111] J. Rubenstein, M.M. Merzenich, Model of autism: Increased ratio of excitation/inhibition in key neural systems, Genes Brain Behav. 2 (2003) 255–267.
[112] A.A. Curley, D.A. Lewis, Cortical basket cell dysfunction in schizophrenia, J. Physiol. 590 (2012) 715–724.

[113] D.A. Lewis, The chandelier neuron in schizophrenia, Dev. Neurobiol. 71 (2011) 118–127.
[114] M. Inan, T.J. Petros, S.A. Anderson, Losing your inhibition: Linking cortical GABAergic interneurons to schizophrenia, Neurobiol. Disease 53 (2013) 36–48.
[115] D. Joshi, J.M. Fullerton, C.S. Weickert, Elevated ErbB4 mRNA is related to interneuron deficit in prefrontal cortex in schizophrenia, J. Psychiatr. Res. 53 (2014) 125–132.
[116] H. Zeng, J.R. Sanes, Neuronal cell-type classification: Challenges, opportunities and the path forward, Nat. Rev. Neurosci. 18 (2017) 530–546.
[117] R. Yuste, M. Hawrylycz, N. Aalling, A. Aguilar-Valles, D. Arendt, R.A. Arnedillo, G.A. Ascoli, C. Bielza, V. Bokharaie, T.B. Bergmann, et al., A community-based transcriptomics classification and nomenclature of neocortical cell types, Nat. Neurosci. (2020) 1–13.
[118] B. Mihaljević, P. Larrañaga, R. Benavides-Piccione, S. Hill, J. DeFelipe, C. Bielza, Towards a supervised classification of neocortical interneuron morphologies, BMC Bioinf. 19 (2018) 511.
[119] B. Tasic, V. Menon, T.N. Nguyen, T.K. Kim, T. Jarsky, Z. Yao, B. Levi, L.T. Gray, S. A. Sorensen, T. Dolbeare, D. Bertagnolli, J. Goldy, N. Shapovalova, S. Parry, C. Lee, K. Smith, A. Bernard, L. Madisen, S.M. Sunkin, M. Hawrylycz, C. Koch, H. Zeng, Adult mouse cortical cell taxonomy revealed by single cell transcriptomics, Nat. Neurosci. 19 (2016) 335–346.
[120] B. Cauli, E. Audinat, B. Lambolez, M.C. Angulo, N. Ropert, K. Tsuzuki, S. Hestrin, J. Rossier, Molecular and physiological diversity of cortical nonpyramidal cells, J. Neurosci. 17 (1997) 3894–3906.
[121] H. Markram, E. Muller, S. Ramaswamy, M. Reimann, M. Abdellah, C. Sanchez, A. Ailamaki, L. Alonso-Nanclares, N. Antille, S. Arsever, G. Kahou, T. Berger, A. Bilgili, N. Buncic, A. Chalimourda, G. Chindemi, J.-D. Courcol, F. Delalondre, V. Delattre, S. Druckmann, R. Dumusc, J. Dynes, S. Eilemann, E. Gal, M. Gevaert, J.-P. Ghobril, A. Gidon, J. Graham, A. Gupta, V. Haenel, E. Hay, T. Heinis, J. Hernando, M. Hines, L. Kanari, D. Keller, J. Kenyon, G. Khazen, Y. Kim, J. King, Z. Kisvarday, P. Kumbhar, S. Lasserre, J.-V. Le Be, B. Magalhaes, A. Merchán-Perez, J. Meystre, B. Morrice, J. Muller, A. Munoz-Cespedes, S. Muralidhar, K. Muthurasa, D. Nachbaur, T. Newton, M. Nolte, A. Ovcharenko, J. Palacios, L. Pastor, R. Perin, R. Ranjan, I. Riachi, J.-R. Rodriguez, J. Riquelme, C. Rössert, K. Sfyrakis, Y. Shi, J. Shillcock, G. Silberberg, R. Silva, F. Tauheed, M. Telefont, M. Toledo-Rodriguez, T. Tränkler, W. Van Geit, J. Díaz, R. Walker, Y. Wang, S. Zaninetta, J. DeFelipe, S. Hill, I. Segev, F. Schürmann, Reconstruction and simulation of neocortical microcircuitry, Cell 163 (2015) 456–492.
[122] H. Markram, M. Toledo-Rodriguez, Y. Wang, A. Gupta, G. Silberberg, C. Wu, Interneurons of the neocortical inhibitory system, Nat. Rev. Neurosci. 5 (2004) 793–807.
[123] J. DeFelipe, P.L. López-Cruz, R. Benavides-Piccione, C. Bielza, P. Larra naga, S. Anderson, A. Burkhalter, B. Cauli, A. Fairén, D. Feldmeyer, G. Fishell, D. Fitzpatrick, T.F. Freund, G. González-Burgos, S. Hestrin, S. Hill, P.R. Hof, J. Huang, E.G. Jones, Y. Kawaguchi, Z. Kisvárday, Y. Kubota, D.A. Lewis, O. Marín, H. Markram, C.J. McBain, H.S. Meyer, H. Monyer, S.B. Nelson, K. Rockland, J. Rossier, J.L.R. Rubenstein, B. Rudy, M. Scanziani, G.M. Shepherd, C.C. Sherwood, J.F. Staiger, G. Tamás, A. Thomson, Y. Wang, R. Yuste, G.A. Ascoli, New insights into the classification and nomenclature of cortical GABAergic interneurons, Nat. Rev. Neurosci. 14 (2013) 202–216..
[124] D. Feldmeyer, G. Qi, V. Emmenegger, J.F. Staiger, Inhibitory interneurons and their circuit motifs in the many layers of the barrel cortex, Neuroscience 368 (2018) 132–151.
[125] R. Tremblay, S. Lee, B. Rudy, GABAergic interneurons in the neocortex: From cellular properties to circuits, Neuron 91 (2016) 260–292.
[126] R. Armañanzas, G.A. Ascoli, Towards the automatic classification of neurons, Trends Neurosci. 38 (2015) 307–318.
[127] F.A.C. Azevedo, L.R.B. Carvalho, L.T. Grinberg, J.M. Farfel, R.E.L. Ferretti, R.E.P. Leite, W.J. Filho, R. Lent, S. Herculano-Houzel, Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain, J. Comparat. Neurol. 513 (2009) 532–541.
[128] O. Sporns, Networks of the Brain, MIT Press, 2011.
[129] J. DeFelipe, I. Fari nas, The pyramidal neuron of the cerebral cortex: Morphological and chemical characteristics of the synaptic inputs, Prog. Neurobiol. 39 (1992) 563–607.
[130] E. White, Cortical Circuits: Synaptic Organization of the Cerebral Cortex Structure, Function, and Theory, Birkhäuser, 1989..
[131] A. Peters, E.G. Jones, Cerebral Cortex: Volume 1: Cellular Components of the Cerebral Cortex, Plenum Press, 1984..
[132] J. DeFelipe, Cortical interneurons: From Cajal to 2001, Prog. Brain Res. 136 (2002) 215–238.
[133] B. Mihaljević, R. Benavides-Piccione, C. Bielza, P. Larrañaga, J. DeFelipe, Classification of GABAergic interneurons by leading neuroscientists, Scientific Data 6 (2019) 1–6.
[134] B. Mihaljević, R. Benavides-Piccione, C. Bielza, J. DeFelipe, P. Larrañaga, Bayesian network classifiers for categorizing cortical GABAergic interneurons, Neuroinformatics 13 (2015) 192–208.
[135] B. Mihaljević, C. Bielza, R. Benavides-Piccione, J. DeFelipe, P. Larrañaga, Multi-dimensional classification of GABAergic interneurons with Bayesian network-modeled label uncertainty, Front. Comput. Neurosci. 8 (2014) 150.
[136] B. Mihaljević, R. Benavides-Piccione, L. Guerra, J. DeFelipe, P. Larrañaga, C. Bielza, Classifying GABAergic interneurons with semi-supervised projected model-based clustering, Artif. Intell. Med. 65 (2015) 49–59.
[137] P.L. López-Cruz, P. Larrañaga, J. DeFelipe, C. Bielza, Bayesian network modeling of the consensus between experts: An application to neuron classification, Int. J. Approximate Reasoning 55 (2014) 3–22.
[138] H. Borchani, C. Bielza, P. Martí, P. Larrañaga, Markov blanket-based approach for learning multi-dimensional Bayesian network classifiers: An application to predict the European Quality of Life-5 Dimensions, EQ-5D) from the 39-item Parkinson's Disease Questionnaire (PDQ-39, J. Biomed. Inform. 45 (2012) 1175–1184.
[139] D.E. Donohue, G.A. Ascoli, A comparative computer simulation of dendritic morphology, PLoS Computational Biology 4 (2008) e1000089.
[140] S. Luengo-Sanchez, P. Larrañaga, C. Bielza, A directional-linear Bayesian network and its application for clustering and simulation of neural somas, IEEE Access 7 (2019) 69907–69921.
[141] W.W. Cohen, Fast effective rule induction, in: Proceedings of the Twelfth International Conference on Machine Learning, Morgan Kaufmann, 1995, pp. 115–123..
[142] P. Larrañaga, D. Atienza, J. Diaz-Rozo, A. Ogbechie, C.E. Puerto-Santana, C. Bielza, Industrial Applications of Machine Learning, CRC Press, 2018.
[143] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank citation ranking: Bringing order to the web, Technical Report, Stanford InfoLab, 1999.
[144] D.F. Gleich, PageRank beyond the Web, SIAM Rev. 57 (2015) 321–363.
[145] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975.
[146] I. Rechenberg, Evolutionstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution, Frommann-Holzboog, 1973..
[147] L.J. Fogel, A.J. Owens, M.J. Walsh, Artificial Intelligence through Simulated Evolution, John Wiley & Sons, New York, 1966.
[148] J.R. Koza, Genetic programming: On the programming of computers by means of natural selection, The MIT Press, 1992.
[149] P. Larrañaga, H. Karshenas, C. Bielza, R. Santana, A review on evolutionary algorithms in Bayesian network learning and inference tasks, Inf. Sci. 233 (2013) 109–125.
[150] D.E. Goldberg, The Design of Innovation: Lessons from and for Competent Genetic Algorithms, Kluwer Academic Press, 2002.
[151] M. Pelikan, D.E. Goldberg, F. Lobo, A survey of optimization by building and using probabilistic models, Comput. Optim. Appl. 21 (2002) 5–20.
[152] R. Armañanzas, Y. Saeys, I. Inza, M. Garcia-Torres, C. Bielza, Y. van de Peer, P. Larrañaga, Peakbin selection in mass spectrometry data using a consensus approach with estimation of distribution algorithms, IEEE/ACM Trans. Comput. Biol. Bioinf. 8 (2010) 760–774.
[153] S. Baluja, Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning, Technical Report CMU-CS-94-163, Carnegie Mellon University, 1994..
[154] G.R. Harik, F.G. Lobo, D.E. Goldberg, The compact genetic algorithm, IEEE Trans. Evol. Comput. 3 (1999) 287–297.
[155] J.S. De Bonet, C.L. Isbell, P. Viola, MIMIC: Finding optima by estimating probability densities, in: Proceedings of the 9th International Conference on Neural Information Processing Systems, MIT Press, 1996, pp. 424–430.
[156] R. Etxeberria, P. Larrañaga, Global optimization using Bayesian networks, in: Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99), pp. 151–173..
[157] M. Pelikan, D.E. Goldberg, E. Cantú-Paz, et al., BOA: The Bayesian optimization algorithm, in: Proceedings of the genetic and evolutionary computation conference, volume 1, pp. 525–532..
[158] P. Larrañaga, H. Karshenas, C. Bielza, R. Santana, A review on probabilistic graphical models in evolutionary computation, J. Heurist. 18 (2012) 795–819.
[159] R. Santana, R. Armañanzas, C. Bielza, P. Larrañaga, Network measures for information extraction in evolutionary algorithms, Int. J. Comput. Intell. Syst. 6 (2013) 1163–1188.
[160] N. Krasnogor, B. Blackburne, E.K. Burke, J.D. Hirst, Algorithms for protein structure prediction, in: Parallel Problem Solving from Nature, volume 2439 of Lecture Notes in Computer Science, Springer Verlag, 2002, pp. 769–778..
[161] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2002) 182–197.
[162] H. Karshenas, R. Santana, C. Bielza, P. Larrañaga, Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables, IEEE Trans. Evol. Comput. 18 (2013) 519–542.
[163] J. Schäfer, K. Strimmer, A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics, Statist. Appl. Genet. Mol. Biol. 4 (2005).
[164] S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Trans. Evol. Comput. 10 (2006) 477–506.
[165] Bojan Mihaljević, Pedro Larrañaga, Concha Bielza, Comparing the Electrophysiology and Morphology of Human and Mouse Layer 2/3 Pyramidal Neurons With Bayesian Networks, Frontiers in Neuroinformatics 15 (2021), https://doi.org/10.3389/fninf.2021.580873, In press.

Bojan Mihaljević recieved his B.Sc. degree in Computer Science and Management from the Diego Portales University in 2010. He obtained a M.Sc. degree in Artificial Intelligence in 2013 and a Ph.D. degree in Artificial Intelligence in 2018, both from from the Universidad Politécnica de Madrid (UPM). Since 2018, he has been a post-doc researcher at the UPM, working on the Human Brain Project. He is currently an Assistant Professor at Departamento de Matemáticas, Universidad Autónoma de Madrid.

Concha Bielza received the M.S. degree in mathematics from the Universidad Complutense de Madrid, Madrid, Spain, in 1989, and the Ph.D. degree in computer science from the Universidad Politécnica de Madrid, Madrid, in 1996, where she has been a Full Professor of statistics and operations research with the Departamento de Inteligencia Artificial, since 2010. She has supervised 12 Ph.D. theses. She has published more than 100 papers in impact factor journals. Her research interests are primarily in the areas of probabilistic graphical models, decision analysis, metaheuristics for optimization, data mining, classification models,and real applications, such as biomedicine, bioinformatics, neuroscience, andindustry. She received the 2014 UPM Research Prize and the Extraordinary Doctorate Award.

Pedro Larrañaga received the M.Sc. degree in mathematics (statistics) from the University ofValladolid and the Ph.D. degree in computer science from the University of the Basque Country. His academic career has been developed at the University of the Basque Country at several faculty ranks, as an Assistant Professor, from 1985 to 1998, as an Associate Professor, from 1998 to 2004, and as a Full Professor, from 2004 to 2007. He earned the Habilitation qualification for Full Professor, in 2003. He has been a Full Professor in computer science and artificial intelligence with the Technical University of Madrid (UPM), since 2007. His research interests are primarily in the areas of probabilistic graphical models, metaheuristics for optimization, data mining, classification models, and real applications, such as biomedicine, bioinformatics, and neuroscience. He has supervised more than 25 Ph. D. theses. He has published more than 200 papers in impact factor journals. He has been an ECCAI Fellow, since 2012, and a Fellow of the Academia Europaea, since 2018. He received the 2013 Spanish National Prize in computer science and the Prize of the Spanish Association for Artificial Intelligence,in 2018. He received the Excellence Award from the University of the Basque Country.