

# A guide to the literature on learning probabilistic networks from data

Wray Buntine

*Abstract*—This literature review discusses different methods under the general rubric of learning Bayesian networks from data, and includes some overlapping work on more general probabilistic networks. Connections are drawn between the statistical, neural network, and uncertainty communities, and between the different methodological communities, such as Bayesian, description length, and classical statistics. Basic concepts for learning and Bayesian networks are introduced and methods are then reviewed. Methods are discussed for learning parameters of a probabilistic network, for learning the structure, and for learning hidden variables. The presentation avoids formal definitions and theorems, as these are plentiful in the literature, and instead illustrates key concepts with simplified examples.

*Keywords*—Bayesian networks, graphical models, hidden variables, learning, learning structure, probabilistic networks, knowledge discovery.

## I. INTRODUCTION

Probabilistic networks or probabilistic graphical models are a representation of the variables in a problem and the probabilistic relationships among them. Bayesian networks, a popular kind of probabilistic network, have been used in different applications including fault diagnosis, medical expert systems, and software debugging [1]. In this review of learning I focus mainly on Bayesian networks which are based on directed graphs.

Probabilistic networks are increasingly being seen as a convenient high-level language for structuring an otherwise confusing morass of equations. They are an explicit representation of dependencies or independencies between variables that ignores the specific numeric or functional details. Depending on interpretation, they can also represent causality [2], [3], [4], [5]. Probabilistic networks in this broad sense were independently developed in a number of communities [6]: in genetics [7], in social science, in statistics to factor multi-dimensional contingency tables; in artificial intelligence to model probabilistic intelligent systems [8]; and in decision theory to model complex decisions [9]. An area not considered in this review is graphical modeling in social science which has had rich development and application, and strong interactions with the artificial intelligence and statistical communities [10], [3], [11], [12].

Networks in general play the role of a high-level language, as is seen in artificial intelligence, statistics, and to a lesser degree in neural networks (where biological views offer an alternative interpretation). See the survey by Ripley [13]. Networks are used to build complex models from simple components. Networks in this broader sense include prob-

abilistic graphical models of the kind considered here, as well as neural networks [14], and decision trees [15]. Probabilistic networks have the distinguishing characteristic that they *specify a probability distribution*—they therefore have a clear semantics that allow them to be processed in order to do diagnosis, learning, explanation and many other inference tasks necessary for intelligent systems. For instance, a new research area considered briefly in the last section is where a probabilistic network is the input specification for a compiler that generates a learning algorithm. This compilation is made easier because the network defines a probability distribution.

Why is learning probabilistic networks of particular interest? Most of the earlier work in artificial intelligence on building expert systems involved a tedious process of manual knowledge acquisition [16]. This tedium spurred two developments that more or less continued independently until recently: machine learning which originally focused on learning rule based systems [17], [18], and uncertainty in artificial intelligence which focused on developing coherent probabilistic knowledge structures whose elicitation suffered less pitfalls. For instance, Henrion and Cooley give a detailed case study [19], and Heckerman developed similarity networks [20] which allow a complex network to be elicited more simply than one would expect. The interest in artificial intelligence in learning of probabilistic networks is a result of the marriage of machine learning and uncertainty in artificial intelligence.

Neural network learning has developed concurrently, based almost exclusively on learning from data. The networks in the computational side of neural networks (interested in information processing as opposed to biological modeling) have increasingly been moving in the direction of probabilistic models. Therefore, there is some overlap between learning of probabilistic networks and neural networks [21], [22], [23]. In statistics, many general inference techniques [24], [25], [26] have been developed that have been applied to learning of probabilistic networks. Computer scientists, for instance in artificial intelligence, have often contributed more in terms of combining and scaling up these techniques, and generalizing them to classes of representations. More examples of the variety of probabilistic networks and their applications to learning are given in [23], [27].

Learning of probabilistic networks includes a number of complications: learning the structure, the parameters given a structure, hidden variables whose values are never present in the data, and values of a variable that are sometimes missing. This review describes some current literature addressing these various tasks, reviews the major methodolo-

gies applied, and describes some of the major algorithms. Available software for learning Bayesian networks is not discussed in this review. An extensive list of software for general inference on probabilistic networks is maintained on the World Wide Web [28]. A list of relevant online tutorial articles and slides, several of those mentioned here, is also available at [29]. Another area not considered in this review is the empirical evaluation of learning algorithms for probabilistic networks. Empirical evaluation of learning algorithms is fraught with difficulties [30]. Notwithstanding, interesting empirical studies appear in [31], [32], [33], [34], [35], [36], [37], [38].

## II. AN INTRODUCTION TO PROBABILISTIC NETWORKS

This section introduces Bayesian networks, and some more general probabilistic networks. For tutorial articles on Bayesian networks see [39], [40], [41]. For an introduction from the artificial intelligence perspective, see [8]. For a statistical introduction to graphical models in general see [42], and a tutorial introduction see [43]. For an introduction to Bayesian networks and Bayesian methods for learning them see [44]. Other kinds of networks include Markov (undirected) networks and Markov random fields are considered widely in image analysis, spatial statistics [45] and neural networks [14].

This section introduces Bayesian networks with a simple example, and then illustrates the richness of the representation with additional examples. Consider Bayesian networks on discrete variables. In their simplest form these consist of a network structure and its associated conditional probability tables. The example below is adapted from [39].

### A. The structure, $S$

The network structure is represented by a Directed Acyclic Graph (DAG) as given in Fig. 1. This network

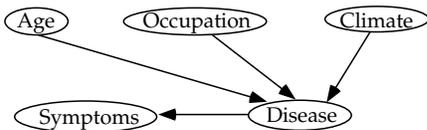


Fig. 1. A simple Bayesian network

is by definition equivalent to the following functional decomposition for the joint probability (full variable names have been abbreviated):

$$\begin{aligned}
 p(\text{Age}, \text{Occ}, \text{Clim}, \text{Disease}, \text{Symptoms}) = & \quad (1) \\
 & p(\text{Age}) p(\text{Occ}) p(\text{Clim}) p(\text{Disease}|\text{Age}, \text{Occ}, \text{Clim}) \\
 & p(\text{Symptoms}|\text{Disease}) ,
 \end{aligned}$$

which is in turn equivalent to the following set of conditional independence statements:

$$\begin{aligned}
 \text{Occ} & \perp\!\!\!\perp \text{Age} , \\
 \text{Clim} & \perp\!\!\!\perp \{\text{Age}, \text{Occ}\} , \\
 \text{Symptoms} & \perp\!\!\!\perp \{\text{Age}, \text{Occ}, \text{Clim}\} | \text{Disease} .
 \end{aligned}$$

TABLE I

TWO OF THE FIVE PROBABILITY TABLES

$\text{Age} < 45$	0.46
$\text{Age} \geq 45$	0.54

$p(\text{Age})$

$\text{Symptoms}$	$\text{Disease}$		
	stomach ulcer	myocardial infarction	neither
stomach pain	0.80	0.05	0.05
chest pain	0.15	0.90	0.10
neither	0.05	0.05	0.85

$p(\text{Symptoms}|\text{Disease})$

Here,  $A \perp\!\!\!\perp B | C$  reads that  $A$  and  $B$  are independent given  $C$  [8], [46]. Take the node for  $\text{Symptoms}$  as an example. This node only has one parent,  $\text{Disease}$ , but three other ancestors,  $\text{Age}, \text{Occ}, \text{Clim}$ . From this one reads the assumption that the symptoms are only dependent on age, occupation and climate indirectly through their influence on the disease. This network substructure, by definition, translates into the third independence statement above. Bayesian networks therefore simplify the full joint probability distribution for a set of variables,  $p(\text{Age}, \text{Occ}, \text{Clim}, \text{Disease}, \text{Symptoms})$  and show independencies between the variables.

### B. The conditional probability tables, parameters $\theta$

Conditional probability tables are needed to specify a probability distribution based on the network. For the structure in Fig. 1, we see from Equation (1) that the tables for  $p(\text{Age})$ ,  $p(\text{Occ})$ ,  $p(\text{Clim})$ ,  $p(\text{Disease}|\text{Age}, \text{Occ}, \text{Clim})$ , and  $p(\text{Symptoms}|\text{Disease})$  need to be specified. These tables may be specified in any form: implicitly by some parametric probability distribution, or explicitly as tables. Two such tables are given below for  $p(\text{Age})$  and  $p(\text{Symptoms}|\text{Disease})$ . Notice that  $\text{Age}$ , while being a real valued variable, is discretized to create a binary variable.  $\text{Symptoms}$  is a three valued discrete variable, as is  $\text{Disease}$ . Without the assumptions of the network which leads to Equation (1), instead of five smaller tables, one large joint table on all five variables would be required. Networks provide a way of simplifying the representation of a probability distribution.

### C. Some extensions

While the variables above are treated as simple discrete variables, and the conditional probabilities in the example above are simple tables, in general a variety of variables and functions can be used on Bayesian networks. Variables could be real valued, integer valued, or multivariate. A real-valued variable may have a probability density function such as a Gaussian. Instead of giving a probability table for it as above, the mean and variance of the Gaussian would be given as functions of the parent variables. These

constructions allow Bayesian networks to represent standard statistical models such as regression with Gaussian error, and log-linear models [42]. Furthermore, graphical models are not restricted to be directed. Undirected arcs can be used in problems such as diagnosis where association between symptoms might be represented, and image analysis, for associations between regions of an image. The combination of directed and undirected graphical models, developed by Lauritzen and Wermuth [47], forms a rich representation language. For an introduction to these combinations see [48]. As an example of this richness, I consider feed-forward neural networks next.

#### D. Connections to feed-forward neural networks

Fig. 2 shows the transformation of a feed-forward neural network predicting real valued variables into a probabilistic network. Fig. 2(a) shows a feed-forward network in

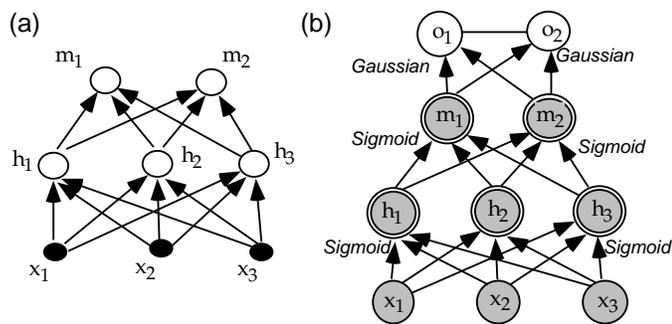


Fig. 2. A feed-forward network to a Bayesian network

the form used in [14], and Fig. 2(b) shows a corresponding probabilistic network with a bivariate Gaussian error distribution grafted onto the output nodes of the network. The feed-forward neural network has the three lower nodes filled in to indicate they are input nodes. The bivariate Gaussian has been represented on the probabilistic network as two nodes with a directed arc between them; an equivalent representation would use an undirected arc. The transformation into the Bayesian network needs to be qualified in several ways. Notice that the interior nodes in the Bayesian network are labeled as Sigmoids, the transfer function typically used in a feed-forward network. The nodes are also double ovals rather than single ovals. This is short-hand to say that the variable is a deterministic function of its inputs, rather than a probabilistic function. Neural networks usually have a weight associated with each arc, giving in some sense the strength of the association. In probabilistic networks, the arc indicates some form of probabilistic dependence or correlation, and any weights are instead associated with each node, and are used to parameterize the functions at the node instead. Furthermore, the probabilistic network explicitly includes the measured output variables in the network,  $o_1$  and  $o_2$ , whereas the neural network only includes the predicted output variables  $m_1$  and  $m_2$ . The probabilistic network therefore explicitly represents the error function, whereas the neural network leaves it unspecified. In summary, the Bayesian network indicates

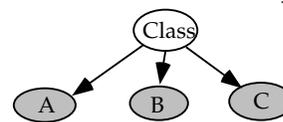


Fig. 3. A simple clustering model

that the output variables,  $o_1$  and  $o_2$ , have a Gaussian distribution based on the variables  $m_1$  and  $m_2$ , which themselves are deterministic Sigmoid functions of the hidden variables  $h_1, h_2, h_3$ , and so forth.

More sophisticated dynamic networks are the recurrent neural networks [49]—roughly, these might be thought of as a flexible, non-linear extension to probabilistic models like Kalman filters and hidden Markov models. While these networks are based on feed-forward neural networks, the relationship of these to probabilistic networks is still under development.

#### E. Connections to statistics and pattern recognition

Whittaker [42], and Wermuth and Lauritzen [50] provide a rich set of examples of modeling statistical hypotheses using graphical models, some using mixed graphs incorporating both undirected and directed networks.

Consider clustering, a style of unsupervised learning. A Bayesian network can be drawn for a clustering algorithm such as Autoclass [51], where it is assumed that the observed variables are independent given the hidden class. In clustering, the cases are to be grouped in some coherent manner. The probabilistic network in Fig. 3. suggests a way of doing this. A discrete variable *class* is introduced that is termed a *latent* or *hidden* variable. Its value never appears in the data, and it indicates the unknown class to which each case belongs. The advantage of this construction is that once the class value is known for a case, the probability distribution becomes a simple one with  $A$ ,  $B$  and  $C$  independent, needing only 3 real valued parameters to define it. This model is called a *mixture model* because the joint probability is a mixture of the data obtained for the different classes. For a visual illustration of the power of mixture models, consider real valued variables  $X, Y$ . A bivariate Gaussian places an oval shaped cloud centered at a point. A mixture of four bivariate Gaussians is illustrated in Fig. 4 has four clouds of points. When the mixture contains many classes, the density can become quite complex.

Popular models used in pattern recognition, speech recognition and control, the Kalman filter and the hidden Markov model (HMM) can also be modeled with Bayesian networks [52], [53]. A simple hidden Markov model is given in Fig. 5. A sequence of observations are made, such as phonemes in an utterance. These are indicated with the shaded nodes  $observe_1, \dots, observe_i, observe_{i+1}$ . Shading indicates the variables have been observed. The observations are dependent on the hidden states  $hidden_1, \dots, hidden_i, hidden_{i+1}$  of the underlying system. If the observations are phonemes, then the hidden states may be letters of the underlying word being spoken, which are of course hidden from the observer. These kinds of models are

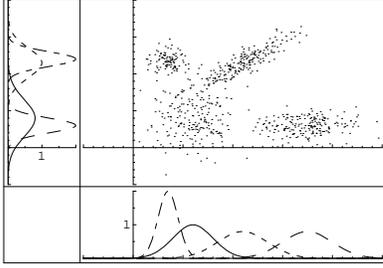


Fig. 4. Data from a 2-dimensional mixture of Gaussians

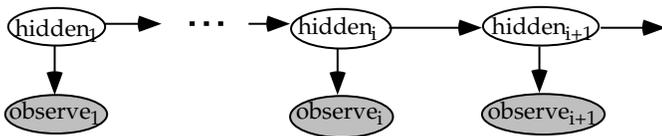


Fig. 5. A simple hidden Markov model.

dynamic, in the sense that the network is a set of repeated units that are expanded in time, as for instance used in forecasting [54].

#### F. Causal networks

A useful trick used in the elicitation of Bayesian networks is to assume the arcs represent causality. Consider the network from [39], reproduced in Fig. 1. One could imagine the environmental variables causing the disease, which in turn causes the symptoms, and this is a nice way of explaining this particular graph to the expert. When Bayesian networks have this interpretation, they are sometimes referred to as causal networks [2], [3], [4], [55]. Causality is of fundamental importance in science because of the notion of intervention [55], [5]. While identifying the observed probabilities relating smoking, sex, and lung cancer is an interesting task in itself, the real goal of such a study is to establish that the act of changing someone's smoking habits will change their susceptibility to lung cancer. This kind of action is an *external intervention* on the variables. A causal model is expected to be stable under acts of external intervention: conclusions drawn from them are still valid. In the probabilistic interpretation of networks used elsewhere in this review, there is an assumption that cases are got through passive observation of independently and identically distributed examples. Networks can be used to represent causality in this manner, but these networks have a different interpretation to the probabilistic networks considered here. Causality, networks and learning causality are not covered in this review. Learning and identification of causality is considered in [56], [3], [57], [58], [59].

TABLE II

A SAMPLE DATABASE IN A RELATIONAL TABLE

case	$A$	$B$	$C$
1	T	F	T
2	T	T	T
3	F	T	T
4	F	T	T

### III. SOME SIMPLE EXAMPLES, AND SOME BASIC CONCEPTS

As an example of learning, consider data about three binary variables,  $A, B, C$ . This data would take the form of a table, as given in the simple example in Table II. The 4 rows in the table give 4 *cases*, which might be different patients. More typically, hundreds or thousands of cases would exist in a relational database. In Table II, each case has three variables measured and their values recorded. The *values* for each variable are either true, indicated by  $T$  or false, indicated by  $F$ . A variable could also have the value “?”. This represents a *missing value*, which means the value for the variable is unknown. Missing values are common in some domains, especially where variables are expensive to measure.

#### A. The hypothesis space

Some example Bayesian networks that might match this problem are given in Fig. 6. First consider structure (a),

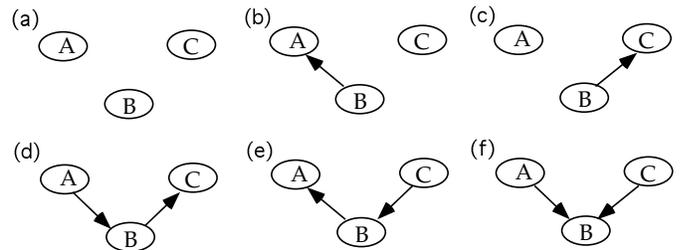


Fig. 6. Some Bayesian networks on three variables,  $A, B, C$ .

I will denote  $S_a$ , which represents that the three variables are independent. For this structure, probability tables for  $p(A)$ ,  $p(B)$  and  $p(C)$  are needed. Since the variables are binary, these three probabilities are specified by three real numbers between 0 and 1. Denote these tables by the parameter set  $\theta_a \in \mathbb{R}^3$ . For structure (c) denoted  $S_c$ , probability tables for  $p(A)$ ,  $p(B)$  and  $p(C|B)$ , denoted  $\theta_c$ , are needed. This parameter set is in  $\mathbb{R}^4$  because while  $p(A)$  and  $p(B)$  are specified by one value each,  $p(C|B)$  is specified by two values, for instance  $p(C = T|B = T)$  and  $p(C = T|B = F)$ . Consider the conditional probability distributions that complete the network  $S_m$ . The probability table for  $p(X|Y)$  will be a subset of the real space of  $(|X| \perp 1)|Y|$ -dimensions, where  $|X|$  is the number of values of the variable  $X$ . The *fully connected* network matching Table II, where every two variables are connected, will have

7 real values, where 7 is calculated from  $2^3 \perp 1$ . So a network of  $k$  binary variables needs between  $k$  and  $2^k \perp 1$  real values to specify its conditional probability tables. A real-valued node whose conditional probability distribution is a Gaussian with  $k$  parents will require  $k(k+1)/2$  real values to specify the mean and the covariance matrix. In general, the real values used to specify conditional probability tables either explicitly (in a table) or implicitly (in some formula) are referred to as the *parameters* of the network.

A simple counting argument shows there are 25 different networks on just the three variables in Fig. 6. However, it happens that several of these are equivalent in the sense that they represent equivalent independence statements. For these networks there are only 11 different *equivalence classes* of networks on three variables. For instance, consider the last three networks given in Fig. 6, (d), (e) and (f). The networks have the following functional decompositions respectively (labeled  $d$ ,  $e$  and  $f$ ):

$$\begin{aligned} p_d(A, B, C) &= p_d(A) p_d(B|A) p_d(C|B) , \\ p_e(A, B, C) &= p_e(C) p_e(B|C) p_e(A|B) , \\ p_f(A, B, C) &= p_f(A) p_f(C) p_f(B|A, C) . \end{aligned}$$

Some basic algebra using the laws of conditional probability show that the Bayesian networks (d) and (e) have equivalent functional decompositions and therefore equivalent independence properties, but the Bayesian network for (f) is different. The structures  $S_d$  and  $S_e$  are said to be *equivalent probability models*. Properties of this equivalence relation have been worked out in general for Bayesian networks [2] (this is discussed further in Section V). Since there are  $k(k \perp 1)/2$  different undirected arcs one can place on a network of  $k$  variables, that means there are  $2^{k(k-1)/2}$  different undirected networks on the  $k$  variables. If the variables are ordered ahead of time so that an arc can only point towards a variable later in the ordering, then there are  $2^{k(k-1)/2}$  different directed networks. There would be many more if the ordering is allowed to vary (although some will be equivalent probability models).

### B. The sample likelihood

The *maximum likelihood* approach is the starting point of most statistical theory, so it is introduced here. First, fix a structure  $S_m$  and its parameters  $\theta_m$  for the model matching the problem of Table II, and calculate the likelihood of the sample as follows:

$$p(\text{sample} | S_m, \theta_m) = \prod_i p(\text{case}_i | S_m, \theta_m) , \quad (2)$$

where the case probabilities  $p(\text{case}_i | S_m, \theta_m)$  are calculated using the probability tables given by  $\theta_m$ . This formulation assumes that each case is independent of the others given the “true” model  $S_m, \theta_m$ , that is they are *independently and identically distributed*. The “true” model is the unknown model believed to represent the process generating the data, and is assumed to exist for purposes of modeling (perhaps a reasonable approximation exists, perhaps not).

For instance, for structure  $S_d$  from Fig. 6,

$$\begin{aligned} p(\text{case}_1 | S_d, \theta_d) &= p(A = T | \theta_d) p(B = F | A = T, \theta_d) \\ &\quad p(C = T | B = F, \theta_d) . \end{aligned}$$

The three terms on the right of this equation are found from the corresponding entries in the probability tables  $\theta_d$ . This quantity Equation (2) is called the *sample likelihood*. The maximum likelihood approach for fixed structure  $S_m$  chooses the parameters  $\theta_m$  to maximize the sample likelihood.

It is important to notice the structure of the maximum likelihood calculation. The probability  $p(A = T | \theta_d)$  appearing in the likelihood for case 1 is a function of the parameters used in the conditional probability table for the variable  $A$ . The parameters  $\theta_d$  for the Bayesian network structure  $S_d$  can be partitioned into the different parameters at each node ( $A, B$  and  $C$ ):

$$\theta_d = \{\theta_{d,A}, \theta_{d,B}, \theta_{d,C}\} ,$$

where  $\theta_{d,B}$  represents the parameters for the conditional probability table for the variable  $B$ . The sample likelihood now becomes:

$$\begin{aligned} p(\text{sample} | S_d, \theta_d) & \quad (3) \\ &= \prod_i p(A_i | \theta_{d,A}) p(B_i | A_i, \theta_{d,B}) p(C_i | B_i, \theta_{d,C}) . \end{aligned}$$

Notice this product has separate terms for  $\theta_{d,A}$ ,  $\theta_{d,B}$ , and  $\theta_{d,C}$ , so maximum likelihood optimization of  $\theta$  can be decomposed into maximum likelihood optimization of these three different variable sets individually. This can be represented as

$$\begin{aligned} p(\text{sample} | S_d, \theta_d) & \\ &= p(\text{sample}_A | \theta_{d,A}) p(\text{sample}_{B|A} | \theta_{d,B}) p(\text{sample}_{C|B} | \theta_{d,C}) , \end{aligned}$$

to show we have three local maximum likelihood problems, one for each node. The sample likelihood is said to *decompose* for Bayesian networks which have neither deterministic variables, missing or hidden values, nor undirected arcs [60], [61], [23], [37], [62]. This decomposition also applies as a network is incrementally modified, for instance during search [23], [60].

If all the parameters  $\theta_d$  describe probability tables for binary variables, as in Table II, then Equation (3) corresponds to a product of binomials. For instance,

$$p(\text{sample}_A | \theta_{d,A}) = \theta_{d,A}^{p_A} (1 \perp \theta_{d,A})^{n_A}$$

where the counts  $p_A$  and  $n_A$  give the occurrences of  $A = T$  and  $A = F$  respectively in the data. As is the case for the binomial, the maximum likelihood is given by the observed frequency,  $\hat{\theta}_{d,A} = \frac{p_A}{n_A + p_A}$ . Likewise for the other variables and all the entries in the other tables.

An important and common assumption used in computing the sample likelihood is the *complete data* assumption. This holds when no case has missing values. This can be an

unrealistic assumption. For instance, if data comes from a historical medical database it is likely that expensive measurements would not have been taken and recorded if they were not considered critical to the diagnosis. The complete data assumption simplifies calculation of the sample likelihood for a network. For instance, consider the model for Fig. 6(f), and consider the likelihood for case 3. Suppose the variable  $C$  had a missing value, “?”.

$$p(\text{case}_3|S_d, \theta_d) = \sum_{C \in \{T, F\}} p(A = F|\theta_f) p(C|\theta_f) p(B = F|A = T, C, \theta_f)$$

As before, the three terms on the right of this equation are simply the corresponding entries in the probability tables  $\theta_f$ . However, notice the summation outside this. When there are many of these summations, there is no longer a simple closed form solution for maximizing the sample likelihood. Furthermore, the optimization problem no longer decomposes, as was demonstrated with Equation (3). Hidden variables lead to the same problem, and violate the complete data assumption, because the summations above always appear in the sample likelihood.

A concept central to these and subsequent techniques is the family of statistical distributions known as the *exponential family* [26], [63]. An introduction in the context of probabilistic networks appears in [23]. This family, which includes the Gaussian, the Bernoulli, and the Poisson has the general functional form of

$$p(x|\theta) \propto \exp\left(\sum_i s_i(\theta)t_i(x)\right),$$

which lends itself to many convenient computational properties including compact storage of the training sample, simple calculation of derivatives, and fitting guaranteed to be linear in the size of the sample. One needs to become familiar with these features of the exponential family in order to understand many of the recent developments in learning probabilistic models. Many of the properties of the sample likelihood, the impact of complete data assumption, exact solutions to the maximum likelihood equations and so forth follow directly from standard results for the exponential family—the effort is usually expended in formulating the probabilistic network as a member of the exponential family, and then the standard results for exponential family follow [26], [63].

### C. Basic statistical considerations

Suppose the structure  $S_m$  of a network on discrete or Gaussian variables is fixed. Then it remains to learn the parameters,  $\theta_m$ . For the probability tables considered earlier and with enough data, the sample likelihood is a well-behaved differentiable function of its parameters. This is often called a *parametric* problem. A non-parametric problem, in contrast, has potentially an infinite number of parameters, or no coherent likelihood function is defined so it is un-parameterized. This is not always clear from the

literature because in some cases a model is presented in a non-parametric manner, whereas it can be given a parametric basis (classification trees are an example [64], [15]). Now consider the problem of learning the structures as well, and remember there are a finite number of them. A fixed network structure has its own distinct set of parameters. When allowing a set of different structures, each with its own parameters, the full probability density has no single, natural, global real-valued parameterization, but has different parameterizations depending on which structure is used. Such problems are sometimes referred to as semi-parametric, but the same qualifications apply. Of course, a clever mathematician can coerce a full specification of the network and its parameters into some single real number. However, this would be an artificial construct with complex non-continuous derivatives. Furthermore, for the structures of Fig. 6, the probability distributions represented by structure  $S_a$  are a set of measure zero in the probability distributions with structure  $S_b$ , which themselves are a set of measure zero within  $S_e$ <sup>1</sup>. By offering these structures as valid alternatives, the set of measure zero is not to be ignored. I will refer to this combination of detail—for a given structure there is a neat parametric model, and structures form nested hierarchies with some being a subset of measure zero of others—as the *parametric structure* of the problem.

Learning network structures from data is sometimes termed a *model selection* problem in the sense that each network corresponds to a distinct model, and one is to be selected based on the data. Both non-parametric methods and model selection are active research areas in modern statistics [65], [25], [66]. More recently, researchers in statistics have focused on *model uncertainty* because it is accepted that selection of a single “best” model from an exponential-sized family of models—as is the case for learning Bayesian networks—is often infeasible [67], [68], [25]. Rather than selecting a single best model, one looks at a subset of “reasonable” models, attempting to quantify uncertainty about them.

### D. The complexity of learning

So network learning involves choosing from, possibly, an exponential number of network structures, and giving values to, possibly, an exponential number of real values. Why is this a problem? Basic results from computational learning theory show how difficult this can be, both in terms of the number of cases required for training, and the time or space required for the optimization. These two aspects are referred to as *sample complexity* and *computational complexity* respectively.

In learning there are roughly three distinct phases as more cases are obtained to learn from: the small sample, medium sample, and large sample phases. Initially with

<sup>1</sup>For the purposes of this paper, a subspace has measure zero if its integrated area relative to the full space is zero. Usually this means it is a space of lower dimension. A line has measure zero in a finite plane, but a rectangle on the finite plane has non-zero measure. A two-dimensional slice of a cube has measure zero in the full three-dimensional cube.

a small sample, learning corresponds to going with one’s biases or priors. With a large sample, learning close to the “true” model is possible with high probability, where “close” is measured according to some reasonable utility criteria such as mean-square error or Kullback-Leibler distance. This learning should be possible by many reasonable algorithms that asymptotically converge to the “truth”. In between the small and large sample phase is a medium sample phase where some algorithms should perform better than others, depending on how well their particular biases align with the “true” model. I use the term biases here in a loose sense. As more cases are obtained to learn from, performance may increase gradually or sometimes in jumps as the algorithm better approximates the “truth”. This is illustrated by the learning curve in Fig. 7 which plots error of some idealized algorithm as it gains more cases (represented by the sample size  $N$ ). The asymptotic

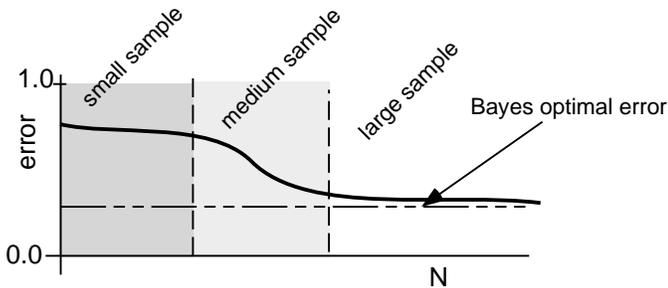


Fig. 7. An idealized learning curve.

error in this example approaches the *Bayes optimal error rate* from above. Without prescience, there will be a lower bound on what error rate can be achieved by any algorithm (for instance, in predicting coin tosses from a fair coin, the Bayes optimal error rate is 50%). The theory of learning curves is developed, for instance, in [69]. Suppose the hypothesis space is a family of probabilistic networks  $(S_i, \theta_i)$  for  $i = 1, \dots, K$ . Results from computational learning theory [70] show that under many conditions the transition to the large sample phase is made when the sample size is given by

$$N = O(\max_i \dim(\theta_i)) + O(\log K) .$$

This sample size is the sample complexity. For the discrete Bayesian networks discussed earlier, the first term will be exponential in  $k$  (the number of variables), and the second term quadratic.

Of course, this ignores the issue of computational complexity. Given that there are an exponential number of networks, it should not be surprising that in some formulations, learning a Bayesian network is an NP-complete problem [71], [72], [36]. In some formulations, learning is viewed as a maximization problem: find the network maximizing some quality measure. As is the case for the the sample likelihood, these scores usually decompose, often because they are based on the sample likelihood, see for instance [61], [23], [37], [62]. The optimization problem is to find a

network  $S$  on variables  $\mathcal{X}$  maximizing some function of the form:

$$quality(S|sample) = \sum_{x \in \mathcal{X}} quality(x|parents_S(x), sample)$$

where the network  $S$  influences the quality measure through the parents function,  $parents_S(\cdot)$ , and the quality measure may be a log-probability, log-likelihood, or a complexity measure (to be minimized). These measures are discussed further in Section VIII. This maximization problem is an instance of a maximum branchings problem (see the discussion in [37]) which in general (allowing any quality function at the nodes) is NP-complete even if variables in network are restricted to have at most 2 parents. It is polynomial if each variable has at most 1 parent. Another variation of this problem, discussed in [37], is to find the best  $l$  networks in terms of the quality measure. For Bayesian networks, this search problem is also confounded because of the existence of equivalent networks. Nevertheless, experience with existing systems shows that standard search algorithms such as greedy algorithms and iterated local search algorithms often perform well. Basic greedy search is explored in [35]. Furthermore, the search problem adapts nicely to branch and bound using some standard methods from information theory to provide the bounds [73], and savings over an exhaustive search appear to be many orders of magnitude.

#### IV. PARAMETER FITTING

For a fixed graphical structure,  $S_m$ , the parameter fitting problem is to learn the parameters  $\theta_m$  from data. The mathematics of fitting parameters to a Bayesian/Markov network is an extension of standard fitting procedures in statistics. Fitting algorithms exist for Bayesian networks and more general probabilistic networks in the cases of complete and missing data [74], [42], [75], [76]. See Whittaker for a more extensive discussion and review of methods and theory. In the case of a Bayesian network with complete data, where the distributions at the nodes are discrete probability tables or Gaussians, fast close form solutions exist that can be computed in time proportional to the size of the data set. As an example, consider fitting the model of Fig. 6(a) to the data in Table 6. Each of the probabilities  $\phi$  in this model occurs in the sample likelihood in the form  $\phi^n (1 - \phi)^m$ , which has a maximum at  $\hat{\phi} = \frac{n}{n+m}$ . The maximum likelihood solution for the parameters is therefore equal to the observed frequency of the relevant probabilities:

$$\begin{aligned} p(A = T|\theta_a) &\approx 1/2 , \\ p(B = T|\theta_a) &\approx 3/4 , \\ p(C = T|\theta_a) &\approx 1 . \end{aligned}$$

In other cases, a variety of iterative algorithms exist that make use of these fast closed form solutions as a subroutine. Some common techniques I shall not explain here are the expectation maximization (EM) algorithm [77] and the

iterative proportional fitting (IPF) algorithm [75]. Once again, the exponential family is important here.

Maximum likelihood approaches suffer from so-called *sparse data* because, for instance, they may become undefined whenever a table of counts total to zero. Consider the model of Fig. 6(e) and consider estimating  $p(B = T|C = F, \theta_e)$ . Notice there are no instances of  $C = F$  in the sample, so the maximum likelihood estimate for this probability is undefined since the sample likelihood does not exist. For  $k$  binary variables and a fully connected Bayesian network (where every two variables are directly connected), clearly need greater than  $2^{k-1}$  cases in the sample for the maximum likelihood estimate to be defined.

A related problem is the problem of *over-fitting*. Suppose sparse data is not a problem. Observe the maximum likelihood estimate above for  $p(C = T|\theta_a)$ . This was equal to 1.0 because in the data, all observed cases of the variable  $C$  had the value  $T$ . Now this is based on four cases. It would seem reasonable that the “true” value could be 0.9, and by chance have all  $T$ ’s in the data. The estimate 1.0 must be an upper bound on the probability. By definition, the maximum likelihood value ( $1.0^4$ ) must be an over-estimate of the “true” sample likelihood ( $0.9^4$ ). As the sample size gets larger and larger, the over-estimate will gradually converge to the “true” value; assured in most cases by large sample properties of maximum likelihood theory (for an introduction see [78]). However, for small samples, the maximum likelihood value may be much larger than the “true” likelihood, and in general the maximum likelihood solution will attempt to fit the data as well as possible—for instance, regression using 10 degree polynomials will fit 11 data points exactly, whereas for 11 data points one might more reasonably attempt to fit a 2 or 3 degree polynomial and assume the remaining lack of fit is due to noise in the data. The maximum likelihood parameter values are therefore said to *over-fit* the data. This is a well-known problem in supervised learning, for instance as addressed by pruning methods for classification trees [64], [15].

The Bayesian *Maximum a-posterior* (MAP) approach extends the maximum likelihood approach by introducing a prior probability. Good introductions to this simplified Bayesian approach and some of its extensions can be found in [79], [80]. The approach places a probability distribution on the unknown parameters  $\theta$  and reasons about them using the axioms of probability theory. The likelihood is augmented with a prior that gives the initial belief about  $\theta$  before seeing any data. Consider just the column of data for  $A$  in Table II, and consider  $\theta_A$ , the parameter giving the probability of  $A$ . By Bayes Theorem:

$$p(\theta_A | \text{sample}) = \frac{p(\text{sample} | \theta_A) p(\theta_A)}{p(\text{sample})},$$

where the numerator contains the sample likelihood and the prior, and the denominator is obtained by integrating the numerator,

$$p(\text{sample}) = \int p(\text{sample} | \theta_A) p(\theta_A) d\theta_A.$$

Again, these computations become simplified in some cases of the exponential family, mentioned previously, Gaussians, Bernoulli, and so forth. An example is given in Fig. 8. The

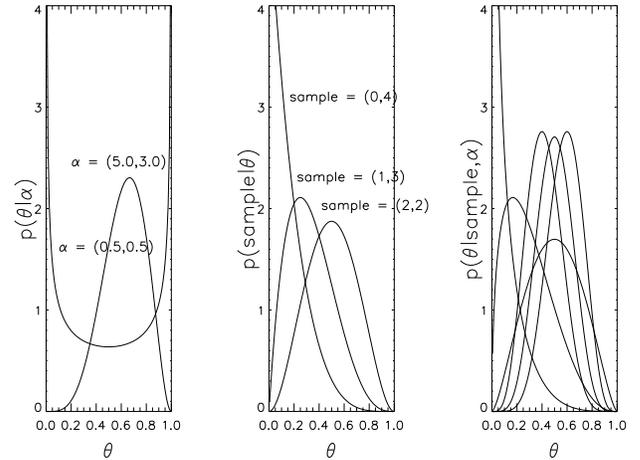


Fig. 8. Priors, likelihoods and posteriors for  $\theta_A$ .

left graph shows two different priors. These priors are Beta distributions with parameters  $\alpha$  marked on the plot. The second prior with  $\alpha = (5, 3)$  has a mild preference for  $\theta$  to be about 0.625, whereas the other prior is agnostic. The middle graph shows the likelihoods for 3 different samples (0,1 or 2 counts of  $A = T$  in a sample size of 4), and the right graph shows the resulting posterior for the  $(2 \times 3)$  posteriors resulting. The cluster of three peaks at the top are three posteriors for the prior  $\alpha = (5, 3)$ . Notice how the agnostic prior ( $\alpha = (0.5, 0.5)$ ) is more influenced by the likelihood, whereas the three posterior peaks for the mild prior reflect the shape of the prior quite strongly. The maximum posterior value is the value of  $\theta$  at the maximum of each curve. Notice how it is effected by both the prior and the likelihood.

Many general algorithms exist for addressing parameter fitting problems of probabilistic networks: missing and latent variables, large samples, recursive or incremental techniques, special nodes, and subjective priors [26], [24], [81], [25], [23], [42] Table III lists the major techniques and their application. References given are introductions, new extensions or examples of their use, and are by no means a thorough list of references in the area. The common versions of the EM and IPF algorithms, and mean field theory are based on the exponential family, although generalizations exist. Used in conjunction with these methods are a large number of optimization techniques, for finding a MAP, or computing the various quantities used in the Laplace approximation. Several optimization techniques are specific to parameter fitting in learning. This includes the Fisher Scoring method [89], which is an approximate Newton-Raphson algorithm, and stochastic optimization which computes gradients on subsamples or individual cases at a time [90]. Variations of this method are popular in neural networks [91], having been a feature of early methods [92], and have proven to yield computational

Algorithm	Problems	Refs.
MAP	general	[25]
Laplace	2nd-order approx.	[25], [82]
EM	missing and hidden values	[77], [76], [83]
IPF	undirected network	[75]
mean field	approximate moments	[84], [22]
Gibbs	approximate moments	[85], [86]
MCMC	approximate moments	[87], [88]

TABLE III  
SOME GENERAL ALGORITHMS FOR PARAMETER FITTING

savings in many studies.

An extension of parameter fitting to handle sequential (on-line) learning and missing data is described in [93]. This uses Bayesian methods to overcome the problems of sparse data, by defining a Dirichlet prior of entries for the probability tables. A full implementation is described in [94]. Extensions have been made to Gaussians and other popular nodes types for the Bayesian network [95]. When combined with some structure elicitation, techniques for parameter fitting can prove powerful in applications, for instance in dynamic models in the medical domain [96], [54].

## V. STRUCTURE IDENTIFICATION METHODS

Ignoring the issue of sample size for the moment, a difficult question is whether particular network structures with or without latent variables are identifiable in the limit with probability 1. That is, assuming there are large amounts of data to accurately estimate various probabilities, can the “true” probabilistic network be reconstructed at all in the sense that a learning algorithm, given a sufficiently large sample, will invariably return a hypothesis (graphical structure and parameters) close to the “truth”? This question is formalized and addressed from several angles in computational learning theory [97] under the name of identification and learnability, as well as in statistics [78], [26] under the name of consistency. This is the situation of  $N \rightarrow \infty$  in Fig. 7.

In Bayesian networks, this question is confounded by the existence of equivalence classes of graphs (one example of a *redundant model* [78]) and by the use of hidden or latent variables. For instance, consider the networks given in Fig. 6 again. The Bayesian networks (d) and (e) have equivalent probability models but the Bayesian network for (f) is different. Therefore, Bayesian networks (d) and (e) have equivalent sample likelihoods and cannot be distinguished from data without some additional criteria or knowledge, whereas the Bayesian network (f) could be identified from data alone. A theoretical tool used to analyze identifiability is the equivalence of graphical models with latent variables [98], [56], [99] and without [100], [101], [2], [102], and more recently involving causality where variables are manipulated [57]. A thorough treatment of the issues of equivalence, latent variables, and causality appears in

[3]. In some cases, only a class of equivalent graphs can be reconstructed from data, and in other cases latent variables and their properties cannot be identified uniquely.

These identification methods have led to some of the earliest algorithms for learning structure from data [103], [56], and a related approach that also combines cross validation to address model selection is [104]. Identification methods are also used in TETRAD II, the successor to TETRAD [12].

The theory of network identification from data and network equivalence are a precursor to techniques for learning from medium sized samples of Fig. 7. Network equivalence is an important concept used in some Bayesian techniques for learning Bayesian networks from data, used in advanced work on priors for Bayesian networks [105], [37]. This will be discussed later.

## VI. DIAGNOSTICS, ELICITATION AND ASSESSMENT

The day to day practice of learning and data analysis may have a learning algorithm at its core but a lot of the work involves modeling and assessment: building a model and trying to find out what is going on with the data, and with the expert’s opinions. Some of the work relevant to learning here comes from statisticians who generally have more experience [106], [107] and decision analysts who use these methods in constructing systems and working with experts [41], [108].

The basic problem of elicitation is a twist on the problem of knowledge acquisition for expert systems.

- In the medium sample regime, which applies frequently, data should be complemented with prior knowledge and constraints if reliable and useful results are to be obtained.
- Prior knowledge can often only be obtained from the domain experts by the manual process of knowledge elicitation.
- Domain experts can be poor at judging their own limitations and capabilities, and estimating probabilities [109]. One of the common mistakes of beginners is to assume that the expert’s claims are valid.

In applications these issues are crucial because a learning problem does not come prepackaged in its own neat wrapper with instructions for assembly: “here’s the data, use these five variables, and try the C4.5 tree program.” A learning problem is usually embedded in some larger problem. A domain expert may be needed just to circumscribe the learning component: which variables might be used, what is being predicted from what, and so forth. Sometimes this is crucial to success, and the learning algorithm used is almost incidental [110].

A number of techniques exist at the interface of learning and knowledge acquisition. Diagnostics are measures used to evaluate particular model assumptions [111], [112] [113]. Sensitivity analysis [114] measures the sensitivity of the results of a study to the model assumptions, using the same techniques taught to engineers everywhere: wiggle the inputs to the model (in the case of learning, this means the constraints and priors) and watch how the output of

the model wiggles. Assessment and elicitation is the usual process discussed in manual knowledge acquisition of interviewing an expert in order to obtain prior estimates of relevant quantities. Because the elicitation and evaluation of probabilistic networks is a well developed area, the further refinement of networks via learning is made possible, as is discussed later under priors.

## VII. LEARNING STRUCTURE FROM DATA

The earliest result in structure learning was the Chow and Liu algorithm for learning trees from data [115]. This algorithm learns a Bayesian network whose shape is a tree. If there are  $k$  variables, then there are  $O(k^2)$  trees, much less than the exponential number of Bayesian networks. The sample complexity is thus  $O(2 \log k)$  more than the sample complexity for each tree, which is  $O(k)$ , thus learning is feasible from small samples. Furthermore, the computational complexity of searching for a tree shaped network requires at most a quadratic number of network evaluations. Herskovits and Cooper [116] demonstrated on a problem of significant size that complex structure learning was possible from quite reasonable sample sizes (in their case, about 10,000), despite being faced with a potentially exponential sample complexity and an NP-complete search problem. Other early work on structure learning was often based on the identification results discussed in the previous section, for instance [103], [56], [104], [117].

Problems like learning the structure of a Bayesian network suffer when samples are smaller. This happens because of over-fitting in the structure space, similar to over-fitting in the parameter space discussed previously. Maximum likelihood and hypothesis-testing methods provide techniques for comparing one structure to another, “shall add an arc here?” “Is model  $S_c$  better than model  $S_f$ ?” This is done, for instance, using the likelihood ratio test [42], [43]. Repeated use of this test can lead to problems because, by chance, hypothesis tests at the 95% confidence level should fail 1 in 20 times, and hundreds of such tests may need to be made when learning a network structure from data. A comparable problem in the statistics literature is variable subset selection in regression. In this problem, one seeks to find a subset of variables on which to base a linear regression. The pitfalls of hypothesis testing in this context are discussed in [67]. The basic problem is that model selection focuses on choosing a single “best” model.

For discrete variables at least, the problem of learning Bayesian networks from complete data is related to the problem of learning classification trees, exemplified by the CART algorithm [64] in statistics and ID3 and C4 in artificial intelligence [15]. This relationship holds because the sample likelihood for a binary classification tree can be represented as a product of independent binomial distributions, just like the sample likelihood for the Bayesian networks on binary variables described in Section III. Both problems also have a similar parametric structure. The classification tree problem has a long history and has been studied from the perspective of applied statistics [64], ar-

tificial intelligence [15], Bayesian statistics [118], minimum description length (MDL) [119], [120], genetic algorithms, and computational learning theory. An adaptation of a successful tree algorithm to an algorithm for learning Bayesian networks appears in [121], and the relationship between the two approaches is discussed in [122].

Another adaptation, which is not quite as direct, is the Constructor algorithm of [104] which adapts the cost-complexity technique from the CART algorithm for trees. There are a variety of heuristic techniques developed for trees, including the handling of missing values [123] and the discretization of real-valued attributes [124], which have yet to find their way into algorithms for probabilistic networks.

## VIII. STATISTICAL METHODOLOGY

In most work on learning structure, researchers have applied standard statistical methodology for fitting models and handling over-fitting. It is therefore appropriate to discuss these standard methodologies, done so in this section. The problem of over-fitting was encountered and addressed by the earliest methods. It is important to note that *the role of a statistical methodology is to convert a learning problem into an optimization problem*. Some of the statistical methodologies, despite their wide philosophical differences, reduce a learning problem to the same kind of optimization problem, so the practitioner could well be left wondering what all the differences are about. It is also important to note that *most structure learning is built around some form of parameter learning as a sub-problem*. In general, the many different structure learning methods are extensions of the general algorithms summarized in Table III. In some cases, this can be as simple as placing a model selection wrapper around a parameter fitting system [125], in other cases more sophistication is layered on top.

It is perhaps unfortunate that so many different, competing statistical methodologies exist to address essentially the same problem. Partly, this stems from the apparent impossibility of handling smaller sample learning problems in any objective manner, and the difficulty of establishing a basis on which a statistical methodology can be judged. See, for instance, the efforts made to compare different learning algorithms in [30], and consider that a statistical methodology is a higher level of abstraction than a learning algorithm. A discussion of the Bayesian perspective on the issues of learning appears in [26], touching on prior probabilities, and subjective statistical analysis. Different disciplines have addressed these problems in parallel while they attempted to extend the classical maximum likelihood and hypothesis testing approaches from statistics. Each methodology comes with a cast of staunch protagonists and antagonists and a litany of standard claims, dogma, paradoxes, and counter-claims. It is useful to become familiar with the different approaches and the mappings and approximations between them to better understand their differences, however this can be difficult given the confusing state of the literature. Each methodology has its particular strengths that make it suitable under certain conditions: ease of implementation, adequate for large samples, more

appropriate for the engineer, availability of software and training, and so forth. I believe no one methodology is superior in all respects.

My comments in this review are colored from a Bayesian perspective. I have tried to keep my comments below to the realm of what is “generally believed” by those knowledgeable in this area rather than merely repeating the dogma of each community. Also, this section is not an introduction to these methodologies. I include appropriate tutorial references below. Finally, there are really hundreds of different methodologies, one for each small cluster of researchers. The list below presents different corners in a continuum.

#### A. Maximum likelihood and Minimum cross entropy methods

The maximum likelihood approach says to find the network structure  $S_m$  whose maximum likelihood over parameters  $\theta_m$  is the largest

$$\operatorname{argmax}_{S_m} \max_{\theta_m} p(\text{sample}|\theta_m, S_m) .$$

The minimum cross entropy approach says to find the structure whose minimum cross entropy with the data is the smallest. These two approaches are equivalent [126], and they are also well known to suffer from over-fitting, as discussed in Section IV. If the “true” model has one single equivalent representative in the hypothesis space, then the maximum likelihood approach is consistent in the sense that in the limit of a large sample it will converge on this “truth” [78]. The maximum likelihood method can also be viewed as a simplification of most other approaches, so it is an important starting point for everyone. When in a large sample regime, the best strategy is to use the maximum likelihood approach to avoid all the mathematical or implementation details of the more complex approaches. The results from computational learning theory for bounding the onset of the large sample phase are useful for deciding when to do this. For Bayesian networks, the maximum likelihood approach has been applied by [127], [116]. The paper by Herskovits and Cooper was the major breakthrough in learning Bayesian networks. It was clear from this paper that MDL and Bayesian methods, which extend the maximum likelihood approach, could be applied in all their detail.

#### B. Hypothesis testing approaches

Hypothesis testing is the standard model selection strategy from classical statistics. For probabilistic networks methods are well developed and a variety of statistical software exists [28], [43], [13]. As mentioned before, the problem is that this is only a viable approach if a small number of hypotheses are being tested. Clever or greedy search techniques can help here [128] by reducing the number of hypothesis tests required. Another way for thinking about this is to deal with multiple hypotheses: let hypothesis testing return a set of possible models rather than expecting it to isolate a single one [128]. This strategy then resembles a Bayesian approach where multiple models are considered.

This is discussed in the context of probabilistic networks below.

#### C. Extended likelihood approaches

A number of extensions to the maximum likelihood approach have been proposed to overcome the problem of over-fitting, and to overcome the problems inherent in hypothesis testing. These approaches replace the sample likelihood by a modified score that is to be maximized. Examples include the penalized likelihood, Akaike information criteria (AIC), the Bayesian information criteria (BIC) and others [66], [129]. Typically, this involves minimizing a formula such as the BIC formula

$$\begin{aligned} BIC(S_m|\text{sample}) \\ = -\log p(\text{sample}|\widehat{\theta}_m, S_m) + \frac{1}{2} \dim(\theta_m) \log N , \end{aligned}$$

where  $\widehat{\theta}_m$  is the maximum likelihood estimate of  $\theta_m$  fixing the structure to be  $S_m$ ,  $N$  is the sample size and  $\dim(\theta_m)$  is the dimensionality. The BIC criteria and some related variations are asymptotically Bayesian but avoid specification of the prior, and are similar to variations of the minimum information complexity approaches described below. Examples for undirected probabilistic networks with the BIC criteria appear in [67].

#### D. Minimum information complexity approaches

There are several different schools under the general rubric of minimizing some information complexity measure (“code length”), for instance minimum description length (MDL) [130], minimum message length [131], and minimum complexity [132]. A simple approximation for MDL is equivalent to the BIC above, but other variations involve statistical quantities such as the Fisher Information, and hypothesis dependent complexity measures chosen particularly for the domain. These approaches are popular among engineers and computer scientists who learn coding and information theory as undergraduates. From one perspective, these methods are related to Bayesian MAP methods although there are subtle differences [133]. One advantage that some proponents claim of this approach (particularly those in the MDL school) is that it requires no prior and is hence objective. In most instances a corresponding “implicit prior” can be constructed from the code. Some authors use this approach so that they can use Bayesian methods in disguise without being ridiculed by their anti-Bayesian colleagues. Search bounds, for instance [134], are one area where the information complexity approach takes advantage of the techniques developed in information theory. Suzuki has developed a branch and bound technique for learning Bayesian networks based on information-theoretic bounds [73]. For Bayesian networks, MDL has been applied by [61], [135], [136].

#### E. Resampling approaches

Modern statistics has developed a variety of *resampling schemes* for addressing over-fitting in parametric situations

like learning networks. Resampling refers to the fact that pseudo-samples are created from the original sample. A popular approach is cross validation, applied by [104]. Resampling schemes have been used to great success in applied multivariate statistics, see for instance a tutorial in [137]. Their strength lies in the fact that they are reliable black box method that can be used without requiring some of the complex mathematical treatment found in the Bayesian or minimum complexity methods [138]. These resampling schemes therefore provide a good benchmark for comparison with more complex schemes which have additional mathematical and implementation pitfalls. Their theoretical justification is large sample, although they have empirical successes in the small sample case for a wide range of problems.

#### F. Bayesian approaches

There are a rich variety of Bayesian methods, and depending on the approximations and shortcuts made, most of the previous methodologies can be reproduced with some form of Bayesian approximation. In its full form the Bayesian approach requires specification of a prior probability (for a tutorial and a list of references, see [139]). A good general introduction to Bayesian methods for learning Bayesian networks can be found in [79]. Advanced introductions and reviews of Bayesian methods for learning can be found in [25], [26], [24].

The Bayesian approach has many different approximations. The simplest MAP approach seeks to find the structure  $S_m$  maximizing the log-probability

$$\log p(S_m, \text{sample}) = \log p(S_m) + \log p(\text{sample}|S_m) .$$

The term  $p(\text{sample}|S_m)$  is called the *evidence* and differs from the likelihood  $p(\text{sample}|S_m, \theta_m)$ . The evidence is the average sample likelihood rather than the maximum sample likelihood used in the earlier techniques:

$$p(\text{sample}|S_m) = \int_{\theta_m} p(\text{sample}|S_m, \theta_m) p(\theta_m|S_m) d\theta_m .$$

Sometimes a relative value is calculated instead,

$$\frac{p(S_m, \text{sample})}{p(S_0, \text{sample})}$$

for some base structure  $S_0$ . This is called the *Bayes factor* and a variety of techniques and approximations exist for computing it [25], [26], [23].

The basic technique for Bayesian learning of Bayesian network structures from complete data uses standard Bayesian methods, and was worked out in one form or another, by many [140], [35], [121], [111], [112], [68], [141], [142], [143], [37], [38]. Certainly, these techniques use standard Bayesian manipulations and should be obvious to most students of Bayesian theory. The general case for the exponential family is worked through in [105]. Good summaries of this line of work can be found in [111], [68], [144], [37], [23], and a thesis covering many of the issues is [36].

The full Bayesian approach is a predictive one: rather than returning the single “best” network, the aim might be to perform prediction or estimate probabilities for new cases. For instance, one might be interested in the probability of new cases based on the sample,  $p(\text{new-case}|\text{sample})$ . In general this is estimated by averaging the predictions across all possible networks using the probability identity

$$p(\text{new-case}|\text{sample}) = \sum_{S_m} p(\text{new-case}|S_m, \text{sample})$$

This situation is represented in Fig. 9. This approaches

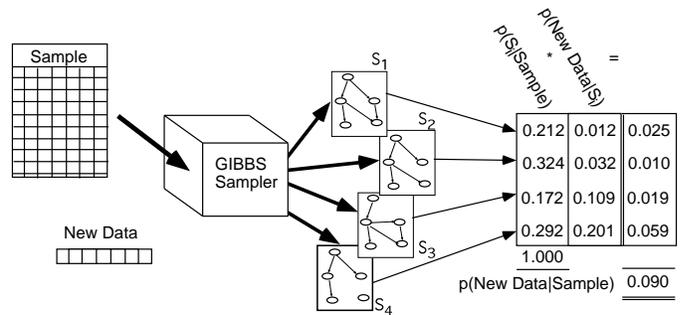


Fig. 9. Averaging over multiple Bayesian networks

matches the intuition: “five different networks all seem quite reasonable so let’s hedge our bets and combine them.” In practice this full summation is not possible so approximations are used. Bayesian methods for learning probabilistic networks in this more general sense can be found in [121], [68], [143], [144], [145], [35], [146], [147]. Computational aspects of finding the best  $l$  networks are discussed in [37]. A related concern is how to combine the posterior network probabilities efficiently and to compute conditional posterior probabilities [148], [111], [32].

A general Bayesian algorithm family for inference that applies in any context, parameter fitting or structure learning, is the *Markov Chain Monte Carlo* (MCMC) family of algorithms. An introduction is given in [149], [23], and an extensive review is given by [87]. This family uses the following kind of trick. Suppose we wish to sample from the distribution  $p(A, B, C)$ . In general this might be a complex distribution and no convenient sampling algorithm may be known. When the complete data assumption is violated for instance, as discussed in Section III-B, it is quite easy to get an intractible sample likelihood distribution for network parameters, and hence the posterior distribution for network parameters may have no convenient functional form to sample from—this is exactly the kind of problem that MCMC methods were designed for. They can even be used for instance, to estimate posterior predictions when learning with complex parametric systems such as sigmoidal feed-forward neural networks [88]. To sample from  $p(A, B, C)$  using the Gibbs sampler, the simplest kind of MCMC method, we start at  $A_0, B_0, C_0$ , and then repeatedly re-sample each variable in turn according to its current conditional distribution (“ $\sim$ ” should be read as “to

be sampled from”):

$$\begin{aligned} A_1 &\sim p(A|B = B_0, C = C_0), \\ B_1 &\sim p(B|A = A_1, C = C_0), \\ C_1 &\sim p(C|A = A_1, B = B_1), \\ A_2 &\sim p(A|B = B_1, C = C_1), \\ B_2 &\sim p(B|A = A_2, C = C_1), \\ &\dots \end{aligned}$$

Probabilistic networks are an ideal framework for developing MCMC methods because these conditional distributions can be generated automatically from the network. MCMC methods can be used for parameter fitting, to sample different network parameters, and for structure learning, to sample from different possible probabilistic network structures. Use of MCMC methods for learning probabilistic networks is discussed in [85], [144], [147], [146], [23]. Madigan, Gavrin and Raftery [146] refer to the use of MCMC methods for averaging over multiple probabilistic networks—the full predictive approach—as *Markov Chain Monte Carlo Model Composition* (MC<sup>3</sup>).

The key distinction between Bayesian and non-Bayesian methods is the use of priors. Priors can unfortunately be complex mathematically, so poorly chosen priors can make a Bayesian method perform poorly against other methods—a real danger in the case of Bayesian networks because of their semi-parametric nature. Both informative priors [68], [111], [121], [37], [35], [38], [146], [147], and non-informative priors can be used. A fundamental assumption is that equivalent network structures should have equivalent priors on their parameters [121], [60], [37], [150]. For instance, consider structures  $S_d$  and  $S_e$  from Fig. 6. The prior probability  $p(\theta_d|S_d)$ , by virtue of equivalence, can be converted into a prior for  $\theta_e$  using a change of variables with the Jacobian for the transformation:

$$q(\theta_e|S_e) = p(\theta_d|S_d) \det \left( \frac{d\theta_d}{d\theta_e} \right).$$

Notice this prior is constructed from the prior for  $S_d$ , and is not necessarily equal to the prior actually used for  $S_e$ ,  $p(\theta_e|S_e)$ . The assumption of prior equivalence sets these two priors equal, something not applicable if the network has a causal interpretation [58]. This gives a set of functional equations that the prior should satisfy. This basic theory and other properties of priors for Bayesian networks is discussed in [105], extending techniques presented in [37].

The ability to use a variety of informative, subjective priors for Bayesian networks is one of their strengths. Informative priors can include constraints and preferences on the structure of the network [121], [37], as well as preferences on the probabilities, and even using the expert to generate “imaginary data” [146]. An example in the language of chain graphs (an extension to Bayesian networks) is given by [38]. The potential for using Bayesian networks as a basis for knowledge refinement has been suggested by [121], [37], [111], [146], and in applications this offers an integrated approach to the development and maintenance

of intelligent systems, long considered one of the potential fruits of artificial intelligence.

## IX. MORE ON LEARNING STRUCTURE

An exact algorithm for handling incomplete data or missing values can be found in [151]. The problems involved here for exact methods were previously explained in [35]. While impractical for larger problems, this could serve as a tool to benchmark on non-trivial sized problems for the many approximate algorithms that exist, for instance, some are mentioned in Table III.

Simple clustering algorithms learn Bayesian networks with a single latent/hidden variable at the root of the network. So these kinds of problems have been addressed in a limited sense for many years in the AI and statistics community [152]. A Bayesian method is [153], [51]. Likewise, missing values can be handled by the well known EM algorithm [76], or more accurately by Gibbs sampling [85]. More recent versions of these clustering algorithms search over possible structures as well [51].

Some algorithms do not fit neatly into the categories above. Learning Markov (undirected) networks from data is related to the early Boltzmann machine from neural networks [21]. Also the earlier Bayesian methods seemed to require as input a strict ordering of variables [35], [121], whereas the identification algorithms did not require this. So one thought is a combination of Bayesian with identification algorithms [33]. But Bayesian methods do equivalent things in the large sample case to the independence tests used by identification algorithms, and the strict ordering is not entirely necessary for the Bayesian algorithms [32], [37]. A variety of hybrid algorithms exist [59], [104], [12], [73] that provide a rich source of ideas for future development.

## X. CONSTRUCTING LEARNING SOFTWARE

For a variety of network structures with latent variables and different parametric nodes (Logistic, Poisson, and other forms), the BUGS program can generate Gibbs samplers automatically [154], [86]. This effectively allows data analysis algorithms to be compiled from specifications given as a probabilistic network, and the technique addresses a number of non-trivial data analysis problems [155], [86]. Unfortunately, Gibbs sampling without much thought to domain specific optimization can be time intensive because convergence may be slow, so other methods need to be developed to make this approach more widely applicable. Other algorithm schemas from Table III can be applied within this compilation framework as well, so it may be possible to construct more efficient algorithms automatically. An exposition of the techniques used by algorithms for learning Bayesian networks—decomposition, exact Bayes factors, and differentiation—all readily automated—can be found in [23], [156].

## REFERENCES

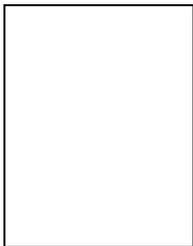
- [1] D. Heckerman, A. Mamdani, and M. Wellman, “Real-world applications of Bayesian networks: Introduction”, *Communications of the ACM*, vol. 38, no. 3, 1995.

- [2] T.S. Verma and J. Pearl, "Equivalence and synthesis of causal models", In Bonissone [157].
- [3] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, Springer-Verlag, New York, 1993.
- [4] D. Heckerman and R. Shachter, "A definition and graphical representation for causality", In Besnard and Hanks [158].
- [5] J. Pearl, "Graphical models, causality, and intervention", *Statistical Science*, vol. 8, no. 3, pp. 266–273, 1993.
- [6] S.L. Lauritzen and D.J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems (with discussion)", *Journal of the Royal Statistical Society B*, vol. 50, no. 2, pp. 240–265, 1988.
- [7] S. Wright, "Correlation and causation", *Journal of Agricultural Research*, vol. 20, pp. 557–585, 1921.
- [8] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, 1988.
- [9] R.A. Howard and J.E. Matheson, "Influence diagrams", in *The Principles and Applications of Decision Analysis*, R.A. Howard and J.E. Matheson, Eds. Strategic Decisions Group, Menlo Park, CA, 1981.
- [10] C. Glymour, R. Scheines, P. Spirtes, and K. Kelly, *Discovering Causal Structure*, Morgan Academic Press, San Diego, CA, 1987.
- [11] S. Mishra and P.P. Shenoy, "Attitude formation models: Insights from TETRAD", In Cheeseman and Oldford [159], pp. 223–232.
- [12] R. Scheines, "Inferring causal structure among unmeasured variables", In Cheeseman and Oldford [159], pp. 197–204.
- [13] B.D. Ripley, "Network methods in statistics", in *Probability, Statistics and Optimization*, F.P. Kelly, Ed., pp. 241–253. Wiley & Sons, New York, 1994.
- [14] J.A. Hertz, A.S. Krogh, and R.G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, 1991.
- [15] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1992.
- [16] F. Hayes-Roth, D.A. Waterman, and D. Lenat, Eds., *Building Expert Systems*, Addison Wesley, 1983.
- [17] D. Michie, "Current developments in expert systems", in *Applications of Expert Systems*, J.R. Quinlan, Ed. Addison Wesley, London, 1987.
- [18] J.R. Quinlan, P.J. Compton, K.A. Horn, and L. Lazarus, "Inductive knowledge acquisition: A case study", in *Applications of Expert Systems*, J.R. Quinlan, Ed. Addison Wesley, London, 1987.
- [19] M. Henrion and D.R. Cooley, "An experimental comparison of knowledge engineering for expert systems and for decision analysis", in *Sixth National Conference on Artificial Intelligence*, Seattle, 1987, American Association for Artificial Intelligence, pp. 471–476.
- [20] D. Heckerman, "Probabilistic similarity networks", *Networks*, vol. 20, pp. 607–636, 1990.
- [21] R.M. Neal, "Connectionist learning of belief networks", *Artificial Intelligence*, vol. 56, pp. 71–113, 1992.
- [22] L.K. Saul, T. Jaakkola, and M.I. Jordan, "Mean field theory for sigmoid belief networks", Technical Report 9501, Computational Cognitive Science, MIT, 1995.
- [23] W. Buntine, "Operations for learning with graphical models", *Journal of Artificial Intelligence Research*, vol. 2, pp. 159–225, 1994, JAIR is mirrored at several sites including URL <http://www.cs.washington.edu/research/jair/home.html>.
- [24] M.A. Tanner, *Tools for Statistical Inference*, Springer-Verlag, New York, second edition, 1993.
- [25] R.E. Kass and A.E. Raftery, "Bayes factors and model uncertainty", *Journal of the American Statistical Association*, vol. 90, pp. 773–795, 1995.
- [26] J.M. Bernardo and A.F.M. Smith, *Bayesian Theory*, John Wiley, Chichester, 1994.
- [27] W.L. Buntine, "Graphical models for discovering knowledge", in *Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. S. Uthurasamy, Eds. MIT Press, 1995.
- [28] R. Almond, "Software for belief networks", World wide web site, URL is <http://bayes.stat.washington.edu/almond/belief.html>, 1995.
- [29] AUAI, *Association for Uncertainty in Artificial Intelligence, Home Page*, sited at Thinkbank, Berkeley, 1995, World wide web site, URL is <http://www.auai.org/>.
- [30] D. Michie, D.J. Spiegelhalter, and C.C. Taylor, Eds., *Machine Learning, Neural and Statistical Classification*, Ellis Horwood, Hertfordshire, England, 1994.
- [31] S.L. Lauritzen, B. Thiesson, and D.J. Spiegelhalter, "Diagnostic systems created by model selection methods: A case study", In Cheeseman and Oldford [159], pp. 143–152.
- [32] Remco R. Bouckaert, "Properties of Bayesian belief network learning algorithms", In de Mantaras and Poole [160].
- [33] M. Singh and M. Valtorta, "An algorithm for the construction of Bayesian network structures from data", In Heckerman and Mamdani [161], pp. 259–265.
- [34] C.F. Aliferis and G.F. Cooper, "An evaluation of an algorithm for inductive learning of Bayesian belief networks using simulated data sets", In de Mantaras and Poole [160], pp. 8–14.
- [35] G.F. Cooper and E.H. Herskovits, "A Bayesian method for the induction of probabilistic networks from data", Report SMI-91-01, Section of Medical Informatics, University of Pittsburgh, January 1991.
- [36] R.R. Bouckaert, *Bayesian Belief Networks: from Inference to Construction*, PhD thesis, Faculteit Wiskunde en Informatica, Utrecht Universiteit, June 1995.
- [37] D. Heckerman, D. Geiger, and D. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data", Technical Report MSR-TR-94-09 (Revised), Microsoft Research, Advanced Technology Division, July 1994, To appear, *Machine Learning Journal*.
- [38] Søren Højsgaard and Bo Thiesson, "BIFROST — Block recursive models Induced From Relevant knowledge, Observations, and Statistical Techniques", *Computational Statistics and Data Analysis*, vol. 19, no. 2, pp. 155–175, 1995.
- [39] R.D. Shachter and D. Heckerman, "Thinking backwards for knowledge acquisition", *AI Magazine*, vol. 8, no. Fall, pp. 55–61, 1987.
- [40] E. Charniak, "Bayesian networks without tears", *AI Magazine*, vol. 12, no. 4, pp. 50–63, 1991.
- [41] M. Henrion, J.S. Breese, and E.J. Horvitz, "Decision analysis and expert systems", *AI Magazine*, vol. 12, no. 4, pp. 64–91, 1991.
- [42] J. Whittaker, *Graphical Models in Applied Multivariate Statistics*, Wiley, 1990.
- [43] D. Edwards, *Introduction to Graphical Modelling*, Springer-Verlag, 1995.
- [44] D. Heckerman, "Bayesian networks for knowledge representation and learning", in *Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. S. Uthurasamy, Eds. MIT Press, 1995, An extended version is available as MSR-TR-95-06 from Microsoft Research, Advanced Technology Division.
- [45] B.D. Ripley, *Spatial Statistics*, Wiley, New York, 1981.
- [46] S.L. Lauritzen, A.P. Dawid, B.N. Larsen, and H.-G. Leimer, "Independence properties of directed Markov fields", *Networks*, vol. 20, pp. 491–505, 1990.
- [47] S. Lauritzen and N. Wermuth, "Graphical models for associations between variables, some of which are qualitative and some quantitative", *Annals of Statistics*, vol. 17, pp. 31–57, 1989.
- [48] W.L. Buntine, "Chain graphs for learning", In Besnard and Hanks [158].
- [49] P. Tino, B.G. Horne, C.L., Giles P.C., and Collingwood, "Finite state machines and recurrent neural networks – automata and dynamical systems approaches", Technical Report UMIACS-TR-95-1, Institute for Advanced Computer Studies, University of Maryland, 1995, To be published in *Progress in Neural Networks* special volume on "Temporal Dynamics and Time-Varying Pattern Recognition," (eds) J.E. Dayhoff and O. Omidvar, Ablex Publishing.
- [50] N. Wermuth and S.L. Lauritzen, "On substantive research hypotheses, conditional independence graphs and graphical chain models", *Journal of the Royal Statistical Society B*, vol. 51, no. 3, 1989.
- [51] R. Hanson, J. Stutz, and P. Cheeseman, "Bayesian classification with correlation and inheritance", In IJCAI91 [162].
- [52] T.L. Dean and M.P. Wellman, *Planning and Control*, Morgan Kaufmann, San Mateo, California, 1991.
- [53] W.B. Poland, *Decision Analysis with Continuous and Discrete Variables: A Mixture Distribution Approach*, PhD thesis, Department of Engineering Economic Systems, Stanford University, Stanford, CA, 1994.

- [54] P. Dagum, A. Galper, E. Horvitz, and A. Seiver, "Uncertain reasoning and forecasting", *International Journal of Forecasting*, 1994, Submitted.
- [55] J. Pearl, "Causal diagrams for empirical research", Technical Report R-218-L, Cognitive Systems Laboratory, Computer Science Department, University of California, Los Angeles, 1994, To appear in *Biometrika*.
- [56] J. Pearl and T.S. Verma, "A theory of inferred causation", in *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, J.A. Allen, R. Fikes, and E. Sandewall, Eds., pp. 441-452. Morgan Kaufmann, San Mateo, CA, 1991.
- [57] J. Pearl, "On the identification of nonparametric structural equations", Technical Report R-207, Cognitive Systems Laboratory, Computer Science Department, University of California, Los Angeles, March 1994.
- [58] D. Heckerman, "A Bayesian approach to learning causal networks", In Besnard and Hanks [158].
- [59] P. Spirtes, C. Meek, and T. Richardson, "Causal inference in the presence of latent variables and selection bias", In Besnard and Hanks [158], pp. 499-506.
- [60] A.P. Dawid and S.L. Lauritzen, "Hyper Markov laws in the statistical analysis of decomposable graphical models", *Annals of Statistics*, vol. 21, no. 3, pp. 1272-1317, 1993.
- [61] W. Lam and F. Bacchus, "Using causal information and local measures to learn Bayesian networks", In Heckerman and Mamdani [161], pp. 243-250.
- [62] D. Heckerman and D. Geiger, "Learning Bayesian networks: A unification for discrete and Gaussian domains", In Besnard and Hanks [158].
- [63] O.E. Barndorff-Nielsen, *Information and exponential families in statistical theory*, John Wiley and Sons, New York, 1978.
- [64] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees*, Wadsworth, Belmont, 1984.
- [65] H. Linhart and W. Zucchini, *Model Selection*, Wiley, 1986.
- [66] S.L. Sclove, "Small-sample and large-sample statistical model selection criteria", In Cheeseman and Oldford [159], pp. 31-39.
- [67] A.E. Raftery, "Bayesian model selection in social research (with discussion by gelman & rubin, and hauser, and a rejoinder)", in *Sociological Methodology 1995*, P.V. Marsden, Ed. Blackwells, Cambridge, Mass., 1995.
- [68] D. Madigan and A.E. Raftery, "Model selection and accounting for model uncertainty in graphical models using Occam's window", *Journal of the American Statistical Association*, vol. 89, pp. 1535-1546, 1994.
- [69] D. Haussler, M. Kearns, H.S. Seung, and N. Tishby, "Rigorous learning curve bounds from statistical mechanics", in *Proceedings of the Seventh ACM Conference on Computational Learning Theory*, M. Warmuth, Ed. 1994, pp. 76-87, Morgan Kaufmann.
- [70] D. Haussler, "Decision theoretic generalizations of the PAC model for neural net and other learning applications", *Information and Control*, vol. 100, no. 1, pp. 78-150, Sept. 1992.
- [71] D.M. Chickering, "Learning bayesian networks is np-complete", Submitted to Proceedings of AI and Statistics, 1995.
- [72] K.-U. Höffgen, "Learning and robust learning of product distributions", Research Report Nr. 464, revised May 1993, Fachbereich Informatik, Universität Dortmund, 1993.
- [73] J. Suzuki, "On an efficient mdl learning procedure using branch and bound technique", Technical Report COMP95-27 (1995-06), Institute of Electronics, Information and Communication Engineers, 1995.
- [74] D. Edwards, "Hierarchical interaction models", *Journal of the Royal Statistical Society B*, vol. 51, no. 3, 1989.
- [75] R. Jiroušek and S. Preučil, "On the effective implementation of the iterative proportional fitting procedure", *Computational Statistics and Data Analysis*, vol. 19, no. 2, pp. 177-189, 1995.
- [76] S.L. Lauritzen, "The EM algorithm for graphical association models with missing data", *Computational Statistics and Data Analysis*, vol. 19, no. 2, pp. 191-201, 1995.
- [77] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society B*, vol. 39, pp. 1-38, 1977.
- [78] G. Casella and R.L. Berger, *Statistical Inference*, Wadsworth & Brooks/Cole, Belmont, CA, 1990.
- [79] D. Heckerman, "A tutorial on learning Bayesian networks", Technical Report MSR-TR-95-06, Microsoft Research, Advanced Technology Division, March 1995.
- [80] R.A. Howard, "Decision analysis: perspectives on inference, decision, and experimentation", *Proceedings of the IEEE*, vol. 58, no. 5, 1970.
- [81] R. Musick, J. Catlett, and S. Russell, "Decision theoretic subsampling for induction on large databases", in *Machine Learning: Proc. of the Tenth International Conference*, Amherst, Massachusetts, 1993, Morgan Kaufmann.
- [82] A. Azevedo-Filho and R.D. Shachter, "Laplace's method approximations for probabilistic inference in belief networks with continuous variables", In de Mantaras and Poole [160], pp. 28-36.
- [83] Bo Thiesson, "Accelerated quantification of Bayesian networks with incomplete data", in *Proceedings of First International Conference on Knowledge Discovery and Data Mining*, U. M. Fayyad and R. Uthurusamy, Eds., 1995, To appear.
- [84] Z. Ghahramani, "Factorial learning and the EM algorithm", in *Advances in Neural Information Processing Systems 7 (NIPS'94)*, G. Tesauro, D.S. Touretzky, and T.K. Leen, Eds. 1994, Morgan Kaufmann.
- [85] J. York and D. Madigan, "Markov chain Monte Carlo methods for hierarchical Bayesian expert systems", In Cheeseman and Oldford [159], pp. 445-452.
- [86] W.R. Gilks, A. Thomas, and D.J. Spiegelhalter, "A language and program for complex Bayesian modelling", *The Statistician*, vol. 43, pp. 169-178, 1993.
- [87] R.M. Neal, "Probabilistic inference using Markov chain Monte Carlo methods", Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto, 1993.
- [88] Radford M. Neal, *Bayesian Learning for Neural Networks*, PhD thesis, University of Toronto, Graduate Department of Computer Science, October 1994, Available via FTP from <ftp://cs.toronto.edu/pub/radford/thesis.ps.Z>.
- [89] P. McCullagh and J.A. Nelder, *Generalized Linear Models*, Chapman and Hall, London, second edition, 1989.
- [90] H. Robbins and S. Munro, "A stochastic optimization method", *Annals of Mathematical Statistics*, vol. 22, pp. 400-407, 1951.
- [91] M.F. Møller, *Efficient Training of Feed-Forward Neural Networks*, PhD thesis, Aarhus University, Aarhus, Denmark, 1993.
- [92] David E. Rumelhart, James L. McClelland, and the PDP Research Group, Eds., *Parallel Distributed Processing*, MIT Press, 1986.
- [93] D.J. Spiegelhalter and S.L. Lauritzen, "Sequential updating of conditional probabilities on directed graphical structures", *Networks*, vol. 20, pp. 579-605, 1990.
- [94] K.G. Olesen, S.L. Lauritzen, and F.V. Jensen, "aHUGIN: A systems creating adaptive causal probabilistic networks", In Dubois et al. [163], pp. 223-229.
- [95] F.J. Diez, "Parameter adjustment in Bayesian networks. the generalized noisy OR-gate", In Heckerman and Mamdani [161], pp. 99-105.
- [96] G.M. Provan, "Tradeoffs in constructing and evaluating temporal influence diagrams", In Heckerman and Mamdani [161], pp. 40-47.
- [97] S. Ben-David and M. Jacovi, "On learning in the limit and non-uniform  $(\epsilon, \delta)$ -learning", in *Proceedings of the Sixth ACM Workshop on Computational Learning Theory*, L. Pitt, Ed. 1993, pp. 209-217, Morgan Kaufmann.
- [98] P. Spirtes and T. Verma, "Equivalence of causal models with latent variables", Report CMU-PHIL-33, Philosophy, Carnegie Mellon University, 1992.
- [99] T. Verma and J. Pearl, "An algorithm for deciding if a set of observed independencies has a causal explanation", In Dubois et al. [163].
- [100] M. Frydenberg, "The chain graph Markov property", *Scandinavian Journal of Statistics*, vol. 17, pp. 333-353, 1990.
- [101] D. Geiger, T. Verma, and J. Pearl, "Identifying independence in Bayesian networks", *Networks*, vol. 20, pp. 507-534, 1990.
- [102] S.A. Andersson, D. Madigan, and M.D. Perlman, "On the Markov equivalence of chain graphs, undirected graphs, and acyclic digraphs", Technical Report #281, Department of Statistics, University of Washington, Seattle, WA, December 1994.
- [103] P. Spirtes and C. Glymour, "An algorithm for fast recovery of sparse causal graphs", *Social Science Computing Reviews*, vol. 9, no. 1, pp. 62-72, 1991.
- [104] R.M. Fung and S.L. Crawford, "A system for induction of probabilistic models", in *Eighth National Conference on Artificial*

- Intelligence*, Boston, Massachusetts, 1990, American Association for Artificial Intelligence, pp. 762–779.
- [105] D. Geiger and D. Heckerman, “A characterization of the Dirichlet distribution with application to learning Bayesian networks”, In Besnard and Hanks [158].
- [106] R.L. Winkler, “The quantification of judgment: Some methodological suggestions”, *SIAM Journal on Computing*, vol. 62, pp. 1105–1120, 1967.
- [107] D.J. Spiegelhalter, R.C.G. Bull, and K. Bull, “Assessment, criticism and improvement of imprecise subjective probabilities for a medical expert system”, In Henrion et al. [164], pp. 285–294.
- [108] M.G. Morgan and M. Henrion, *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*, Cambridge University Press, 1990.
- [109] D. Kahneman, P. Slovic, and A. Tversky, *Judgement under Uncertainty: Heuristics and Biases*, Cambridge University Press, Cambridge, 1982.
- [110] P. Langley and H.A. Simon, “Applications of machine learning and rule induction”, *CACM*, 1995, To appear.
- [111] D.J. Spiegelhalter, A.P. Dawid, S.L. Lauritzen, and R.G. Cowell, “Bayesian analysis in expert systems”, *Statistical Science*, vol. 8, no. 3, pp. 219–283, 1993.
- [112] D.J. Spiegelhalter and R.G. Cowell, “Learning in probabilistic expert systems”, In Bernardo et al. [165], pp. 447–465.
- [113] R.G. Cowell, A.P. Dawid, and D.J. Spiegelhalter, “Sequential model criticism in probabilistic expert systems”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 209–219, 1993.
- [114] K.B. Laskey, “Sensitivity analysis for probability assessments in Bayesian networks”, In Heckerman and Mamdani [161], pp. 136–142.
- [115] C.K. Chow and C.N. Liu, “Approximating discrete probability distributions with dependence trees”, *IEEE Transactions on Information Theory*, vol. 14, pp. 462–467, 1968.
- [116] E.H. Herskovits and G.F. Cooper, “Kutató: An entropy-driven system for construction of probabilistic expert systems from databases”, In Bonissone [157], pp. 54–62.
- [117] S. Srinivas, S. Russell, and A. Agogino, “Automated construction of sparse Bayesian networks”, In Henrion et al. [164], pp. 295–308.
- [118] W.L. Buntine, “Learning classification trees”, In Hand [166], pp. 182–201.
- [119] J. Rissanen, *Stochastic Complexity in Statistical Enquiry*, World Scientific, 1989.
- [120] C.S. Wallace and J.D. Patrick, “Coding decision trees”, *Machine Learning*, vol. 11, pp. 7–22, 1993.
- [121] W.L. Buntine, “Theory refinement of Bayesian networks”, in *Uncertainty in Artificial Intelligence: Proceedings of the Seventh Conference*, B.D. D’Ambrosio, P. Smets, and P.P. Bonissone, Eds., Los Angeles, CA, 1991.
- [122] W.L. Buntine, “Classifiers: A theoretical and empirical study”, In IJCAI91 [162].
- [123] J.R. Quinlan, “Unknown attribute values in induction”, in *Proceedings of the Sixth International Machine Learning Workshop*, A.M. Segre, Ed., Cornell, New York, 1989, Morgan Kaufmann.
- [124] U.M. Fayyad and K.B. Irani, “Multi-valued interval discretization of continuous-valued attributes for classification learning”, in *International Joint Conference on Artificial Intelligence*, Chambéry, France, 1993, IJCAI, Inc., pp. 1022–1027, Morgan Kaufmann.
- [125] Ron Kohavi, George John, Richard Long, David Manley, and Karl Pfleger, “MLC++: A machine learning library in C++”, in *Tools with Artificial Intelligence*. 1994, pp. 740–743, IEEE Computer Society Press, Available by anonymous ftp from: `starry.Stanford.EDU:pub/ronnyk/mlc/toolsmlc.ps`.
- [126] S. Kullback, *Information Theory and Statistics*, John Wiley and Sons, New York, 1959.
- [127] D. Geiger, “An entropy-based learning algorithm of Bayesian conditional trees”, In Dubois et al. [163], pp. 92–97.
- [128] D. Edwards and T. Havránek, “A fast model selection procedure for large families of models”, *Journal of the American Statistical Association*, vol. 82, no. 397, pp. 205–211, 1987.
- [129] R.B. Poland and R.D. Shachter, “Three approaches to probability model selection”, In de Mantaras and Poole [160], pp. 478–483.
- [130] J. Rissanen, “Stochastic complexity”, *Journal of the Royal Statistical Society B*, vol. 49, no. 3, pp. 223–239, 1987.
- [131] C.S. Wallace and P.R. Freeman, “Estimation and inference by compact encoding”, *Journal of the Royal Statistical Society B*, vol. 49, no. 3, pp. 240–265, 1987.
- [132] A.R. Barron and T.M. Cover, “Minimum complexity density estimation”, *IEEE Transactions on Information Theory*, vol. 37, no. 4, 1991.
- [133] J.J. Oliver and R.A. Baxter, “Mml and bayesianism: similarities and differences”, Technical Report 206, Monash University, Melbourne, 1994.
- [134] P. Smyth, “Admissible stochastic complexity models for classification problems”, In Hand [166], pp. 335–347.
- [135] W. Lam and F. Bacchus, “Learning Bayesian belief networks: An approach based on the MDL principle”, *Computational Intelligence*, vol. 10, no. 4, 1994.
- [136] J. Suzuki, “A construction of Bayesian networks from databases based on an MDL scheme”, In Heckerman and Mamdani [161], pp. 266–273.
- [137] B. Efron and R. Tibshirani, “Statistical data analysis in the computer age”, *Science*, vol. 253, pp. 390–395, 1991.
- [138] Ron Kohavi, “A study of cross validation and bootstrap for accuracy estimation and model selection”, in *International Joint Conference on Artificial Intelligence*, Montreal, 1995, IJCAI, Inc., Morgan Kaufmann.
- [139] W.L. Buntine, “Prior probabilities”, Tutorial slides available through URL <http://www.Thinkbank.com/wray/>, 1994.
- [140] G.F. Cooper and E.H. Herskovits, “A Bayesian method for the induction of probabilistic networks from data”, *Machine Learning*, vol. 9, no. 4, pp. 309–348, 1992.
- [141] R.D. Shachter, D.M. Eddy, and V. Hasselblad, “An influence diagram approach to medical technology assessment”, in *Influence Diagrams, Belief Nets and Decision Analysis*, R.M. Oliver and J.Q. Smith, Eds., pp. 321–350. Wiley, 1990.
- [142] G. Consonni and P. Giudici, “Learning in probabilistic expert systems”, in *Workshop on Probabilistic Expert Systems*, R. Scozzafava, Ed., Roma, October 1993, pp. 57–78.
- [143] J.C. York, *Bayesian Methods for the Analysis of Misclassified and Incomplete Multivariate Discrete Data*, PhD thesis, University of Washington, Seattle, WA, 1992.
- [144] D. Madigan and J. York, “Bayesian graphical models for discrete data”, Technical Report #259, Department of Statistics, University of Washington, Seattle, WA, November 1993, Submitted to *International Statistical Review*.
- [145] D. Madigan, A.E. Raftery, J.C. York, J.M. Bradshaw, and R.G. Almond, “Strategies for graphical model selection”, In Cheeseman and Oldford [159], pp. 91–100.
- [146] D. Madigan, J. Gavrin, and A.E. Raftery, “Eliciting prior information to enhance the predictive performance of bayesian graphical models”, *Communications in Statistics*, 1995, To appear.
- [147] J. York, D. Madigan, I. Heuch, and R.T. Lie, “Estimation of the proportion of congenital malformations using double sampling: Incorporating covariates and accounting for model uncertainty”, *Applied Statistics*, vol. 44, pp. 227–242, 1995.
- [148] R. Musick, “Minimal assumption distribution propagation in belief networks”, In Heckerman and Mamdani [161], pp. 251–258.
- [149] B.D. Ripley, *Stochastic Simulation*, John Wiley & Sons, 1987.
- [150] D. Heckerman, D. Geiger, and D. Chickering, “Learning Bayesian networks: The combination of knowledge and statistical data”, In de Mantaras and Poole [160].
- [151] G.F. Cooper, “A method for learning belief networks that contain hidden variables”, *Journal of Intelligent Information Systems*, 1994, To appear. Also in *Proceedings of the Workshop on Knowledge Discovery in Databases*, 1993, 112–124.
- [152] D.M. Titterton, A.F.M. Smith, and U.E. Makov, *Statistical Analysis of Finite Mixture Distributions*, John Wiley & Sons, Chichester, 1985.
- [153] P. Cheeseman, M. Self, J. Kelly, W. Taylor, D. Freeman, and J. Stutz, “Bayesian classification”, in *Seventh National Conference on Artificial Intelligence*, Saint Paul, Minnesota, 1988, American Association for Artificial Intelligence, pp. 607–611.
- [154] A. Thomas, D.J. Spiegelhalter, and W.R. Gilks, “BUGS: A program to perform Bayesian inference using Gibbs sampling”, In Bernardo et al. [165], pp. 837–42.
- [155] W.R. Gilks, D.G. Clayton, D.J. Spiegelhalter, N.G. Best, A.J. McNeil, L.D. Sharples, and A.J. Kirby, “Modelling complexity:

- applications of Gibbs sampling in medicine”, *Journal of the Royal Statistical Society B*, vol. 55, pp. 39–102, 1993.
- [156] W.L. Buntine, “Networks for learning”, in *50th Session of the International Statistical Institute*, Beijing, China, 1995, Invited lecture.
- [157] Piero Bonissone, Ed., *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, Cambridge, Massachusetts, 1990.
- [158] P. Besnard and S. Hanks, Eds., *Uncertainty in Artificial Intelligence: Proceedings of the Eleventh Conference*, Montreal, Canada, 1995.
- [159] P. Cheeseman and R.W. Oldford, Eds., *Selecting Models from Data: Artificial Intelligence and Statistics IV*, Springer-Verlag, 1994.
- [160] R. Lopez de Mantaras and D. Poole, Eds., *Uncertainty in Artificial Intelligence: Proceedings of the Tenth Conference*, Seattle, WA, 1994.
- [161] D. Heckerman and A. Mamdani, Eds., *Uncertainty in Artificial Intelligence: Proceedings of the Ninth Conference*, Washington, DC, 1993.
- [162] IJCAI91, Ed., *International Joint Conference on Artificial Intelligence*, Sydney, 1991. Morgan Kaufmann.
- [163] D. Dubois, M.P. Wellman, B.D. D’Ambrosio, and P. Smets, Eds., *Uncertainty in Artificial Intelligence: Proceedings of the Eight Conference*, Stanford, CA, 1992.
- [164] M. Henrion, R. Shachter, L.N. Kanal, and J. Lemmer, Eds., *Uncertainty in Artificial Intelligence 5*, Elsevier Science Publishers, Amsterdam, 1991.
- [165] J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith, Eds., *Bayesian Statistics 4*, Oxford University Press, 1992.
- [166] D.J. Hand, Ed., *Artificial Intelligence Frontiers in Statistics*, Chapman & Hall, London, 1991.



**Wray Buntine** received his B.Sc. degree in mathematics from the University of Queensland in 1979 and his Ph.D. degree in computer science at the University of Technology, Sydney in 1992. He is currently a Senior Scientist at Thinkbank undertaking consulting, exploratory data analysis, and software development. Part of this survey was written while he was a Principle Investigator at the Research Institute for Advanced Computing Science at the NASA Ames Research Center. His research

interests include experimental and theoretical methods in machine learning, statistics, neural networks, and knowledge discovery. He wrote the C-based decision tree learning package, IND. He is currently writing a compiler for probabilistic networks.