Machine Learning, 29, 165–180 (1997)

The Sample Complexity of Learning Fixed-Structure Bayesian Networks

SANJOY DASGUPTA dasgupta@cs.berkeley.edu
Department of Computer Science, University of California, Berkeley, CA 94720

Editor: Gregory Provan

Abstract. We consider the problem of PAC learning probabilistic networks in the case where the structure of the net is specified beforehand. We allow the conditional probabilities to be represented in any manner (as tables or specialized functions) and obtain sample complexity bounds for learning nets with and without hidden nodes.

Keywords: Bayesian networks, PAC learning, sample complexity

1. Introduction

Bayesian networks are a means of representing probability distributions in a natural and compact manner. In this paper, the problem of learning such nets is examined in an appropriate PAC (probably approximately correct) framework, using some recent work on the uniform convergence of empirical estimates. To illustrate some of the advantages of this model, sample complexity bounds are derived for various kinds of nets, including those containing hidden units.

Broadly speaking, the paper has two main results. First, if the structure of the net is specified and there are no hidden units, then the sample complexity of learning is proportional to the *pseudo dimension* of its component conditional probability distributions, a quantity analogous to VC dimension. Second, the use of hidden units in a Bayesian net does not cause a drastic increase in the sample complexity of learning. This further justifies the use of such units, which can potentially make a Bayesian net both more compact and easier to understand.

2. The usefulness of sample complexity bounds

It is often possible to give upper and lower bounds on the number of examples needed for learning a hypothesis class in the PAC framework. Since different learning algorithms may utilize their training data with varying degrees of efficacy, such bounds are based on the assumption that the learning algorithm does not waste its data; that is to say, it picks a hypothesis close to the one that best fits the training set. Finding such a hypothesis may in some cases be computationally very expensive, but removing this assumption would cripple the generality of a sample complexity bound, and require individual bounds to be derived for each different learning algorithm.

The other aspect of sample complexity bounds which merits discussion is that they are often (as in the case of this paper) *distribution free*, that is, they are valid regardless of the distribution from which the training data is drawn, however skewed this distribution might be. This generality can potentially cause a sample complexity bound to be unduly pessimistic. We specifically address this issue in an example later in the paper.

To get a feel for sample complexity bounds, consider the following simple learning situation: suppose we have a coin whose heads probability $p \in [0, 1]$ we wish to determine. Letting 0 denote tails and 1 denote heads, the natural learning algorithm would take the result $X_1, \ldots, X_m \in \{0, 1\}$ of m coin tosses and return the average of these numbers. The question then arises: how large must m be for this answer to be within ϵ of p, with probability at least $1 - \delta$? A simple application of Chernoff-type bounds shows that $m = \frac{1}{2\epsilon^2} \ln \frac{2}{\delta}$ will suffice. This is an upper bound on the sample complexity and is distribution free in the sense that it makes no assumptions about the value of p.

To get a lower bound, we can look at the simple case when $p = \frac{1}{2}$. By bounding the relevant binomial coefficients, we obtain inequalities for the probability that our estimate is more than ϵ away from p and thereby conclude that in general we need $m \geq \frac{1}{5\epsilon^2} \ln \frac{2}{\delta}$.

In this case, the upper and lower bounds differ only by a constant multiplicative factor. Such tightness may be very hard to achieve in more complicated scenarios. The upper bounds obtained in this paper should therefore not be treated as strict measures of the number of examples needed; this is not where their main use lies. Rather, the expressions for these bounds can be very useful in revealing what aspects of the hypothesis class can be altered to significantly reduce the sample complexity of learning.

3. The specification of a Bayesian network

Suppose we want to model a probability distribution P over n variables $x_1, \ldots, x_n$, where the value of x_i is drawn from some set X_i. Although most of our examples will focus on the case where $X_i = \{0, 1\}$, the results apply to any discrete X_i and will be stated as such. With the possible exception of lemma 6, they also generalize immediately to continuous X_i. The continuous case will occasionally be discussed in situations where it differs in some significant way from the discrete case.

In a Bayesian net, the distribution P on $X = X_1 \times \cdots \times X_n$ is specified by an n-vertex directed acyclic graph G representing the dependencies among the variables, and by functions f_i specifying the conditional probability distribution of each variable x_i given the values of its parents in G. If π_i refers to the parents of node i, then the probability of any given instance $x = (x_1, x_2, \ldots, x_n)$ is

$$\prod_{i=1}^{n} f_i(x_i | \pi_i(x)),$$

where $\pi_i(x)$ denotes the values given to the parents of node i by x. In the case where X_i is not a discrete space, f_i expresses a conditional probability density, and the result of the above product is also a density. The hypothesis class from which the functions f_i are drawn

is very important. The first two classes described below are canonical in Bayesian network literature, and we will refer to them throughout the paper to illustrate our results.

EXAMPLE 1 If the X_i are discrete, say $\{0, 1\}$, then the brute-force approach is to represent each f_i by an explicit conditional probability table, that is to say one probability value, corresponding to $f_i(x_i = 1)$, for each possible combination of parent values. For a node with k parents, a hypothesis is thus specified by 2^k parameters.

EXAMPLE 2 Although the "explicit" hypothesis class above is the most general option, certain distributions may permit a decent approximation by a more compact "intensional" representation. One such class of functions, of which a slightly less noisy variant is discussed by Pearl (1988), is the noisy-OR class for Boolean variables. If a node y has k parents $\pi = (x_1, \ldots, x_k)$, then each noisy-OR function is specified by $k+1$ real values $\Delta, p_1, \ldots, p_k \in [0, 1]$, where

$$\mathbf{P}(y = 1|x_1, \ldots, x_k) \;=\; 1 - (1 - \Delta) \prod_{i=1}^{n} (1 - x_i p_i).$$

In other words, for $\Delta = 0$, p_i is the probability that $y = 1$ if $x_i = 1$ and all the other x_j's are 0. A regular OR gate can be achieved using $\Delta = 0, p_1 = p_2 = \cdots = p_k = 1$. It is probably desirable to control the noisiness by requiring that $\Delta \leq$ (say) $1/4$, and so we shall henceforth assume that this is the case.

EXAMPLE 3 Some variables in the net may not have discrete values. Pearl (1988) considers one such case in which a continuous-valued node y is expected to be related linearly to its parents $x_1, \ldots, x_k$, with some Gaussian noise added in. Each hypothesis is parameterized by $k + 1$ real numbers $a_1, \ldots, a_k, \sigma$, and is specified by the conditional probability density function

$$\mathbf{P}(y|x_1, \ldots, x_k) \;=\; \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{ -\frac{1}{2\sigma^2}(y - a_1 x_1 - \cdots - a_k x_k)^2 \right\}.$$

Here, y takes on values in the range $(-\infty, \infty)$. This can be restricted to a smaller set by multiplying the function above by an appropriate normalizing constant.

In these examples, the hypothesis classes are for conditional probabilities at a particular node of a Bayesian net. We will also discuss hypothesis classes for the entire net. The context will make it clear which of these two is intended.

Since Bayesian nets are generally designed to represent a certain causal structure, it is often useful to have "hidden" nodes which do not correspond to observations but to unobservable variables that might, for instance, represent higher-level concepts. This can make the graph easier to understand, and also potentially reduces its connectivity, leading to a more compact representation of the distribution. We can divide the variables into two groups: one consisting of (say) k hidden variables drawn from the (joint) space X_H and the other consisting of the $n - k$ observables drawn from the (joint) space X_O, so that $X = X_O \times X_H$. For convenience, we number the hidden nodes $1, \ldots, k$ and the observable nodes $k + 1, \ldots, n$. Then, the probability of a given instance $(x_{k+1}, \ldots, x_n)$ is taken to be

$$\sum_{(x_1,\ldots,x_k)\in X_H} \prod_{i=1}^{n} f_i(x_i|\pi_i(x)); \tag{1}$$

that is, the sum, over all possible values $(x_1, \ldots, x_k)$ of the hidden nodes, of the probabilities of $x = (x_1, \ldots, x_n)$. For continuous variables, this sum is replaced by the equivalent integral.

4. A learning model

The task of learning Bayesian nets is typically divided into four categories, based on whether the structure is specified in advance and whether there are hidden nodes. A theoretical analysis of the case of variable structure and no hidden nodes can be found in the recent work of Friedman and Yakhini (1996). We use a learning framework with fewer assumptions, and treat the fixed-structure, hidden-variable case.

4.1. Learning a probability distribution

To reiterate, we will consider the learning situation where the structure (underlying graph) of the network is fixed and the goal is to learn the conditional probability functions accurately. The model we use is adapted from Haussler's (1992) extension of the PAC framework (Valiant, 1984). Assume that we want to learn an unknown distribution P over n variables $x_1, \ldots, x_n$ drawn from a (joint) instance space X, and that the answer must be chosen from some hypothesis class H. A learning algorithm $\mathcal{A}$, given (1) an approximation parameter $\epsilon > 0$, (2) a confidence parameter $0 < \delta < 1$, and (3) an oracle which randomly generates instances of X according to P, must output a hypothesis $h \in H$, such that with probability $> 1 - \delta$,

$$d(P, h) \;<\; d(P, h_{opt}) + \epsilon,$$

where $d(\cdot, \cdot)$ is a distance measure which we will discuss later and h_{opt} is the concept $h' \in H$ that minimizes $d(P, h')$. *There is no assumption that P corresponds exactly to some distribution in H.* Moreover, if $\mathcal{A}$ runs in time polynomial in $n, \frac{1}{\epsilon}, \log \frac{1}{\delta}$, and $size(h_{opt})$ (in some reasonable representation), then $\mathcal{A}$ is an efficient learning algorithm.

4.2. The distance measure

Although the most obvious choice of distance measure is perhaps the L_1 norm, much of the work in information theory focuses on one that is not a norm at all, the *Kullback-Leibler divergence* or *relative entropy* (Cover & Thomas, 1991):

$$d_{KL}(P, h) = \sum_{x\in X} P(x) \log \frac{P(x)}{h(x)}.$$

This stringent measure is widely known to have the property that finding a hypothesis h that minimizes d_{KL} with respect to the empirically observed distribution P^* (consisting, say, of m samples $s_1, \ldots, s_m$), is equivalent to finding the (not necessarily unique) hypothesis that maximizes the probability of the sample data (Abe, Takeuchi, & Warmuth, 1990):

$$\begin{aligned} d_{KL}(P^*, h) &= \sum_{i=1}^{m} P^*(s_i) \log \frac{P^*(s_i)}{h(s_i)} \\ &= \sum_{i=1}^{m} \frac{1}{m} \log \frac{1}{m} - \sum_{i=1}^{m} \frac{1}{m} \log h(s_i) \\ &= \log \frac{1}{m} - \frac{1}{m} \log \prod_{i=1}^{m} h(s_i). \end{aligned}$$

That is, minimizing d_{KL} with respect to the empirically observed distribution is equivalent to solving the *maximum likelihood* problem. A significant disadvantage of d_{KL} is that it is unbounded, which complicates convergence proofs. However, since it seems to be rather popular, we will adopt it as our distance measure, with one slight modification: we will use $\ln$ instead of $\log$ to simplify the analysis.

We also want to relate the complexity of learning a distribution to the complexity of learning each of its component conditional distributions. In a Bayesian net, these take the form $\mathbf{P}(x_i|\pi_i)$, where x_i refers to the value of the i^{th} node and π_i to its parents. Let c, h be distributions over all the variables in a Bayesian net, and let $c_i(\cdot|\cdot), h_i(\cdot|\cdot)$ be the corresponding conditional distributions at node i. A useful distance measure between c_i and h_i, with respect to instances drawn from P, is

$$d_{CP}(P, c_i, h_i) = \sum_{\pi_i} P(\pi_i) \sum_{x_i} P(x_i|\pi_i) \ln \frac{c_i(x_i|\pi_i)}{h_i(x_i|\pi_i)}.$$

When we are learning a distribution over $x_1, \ldots, x_n$, and our hypotheses are products of conditional probability functions (as in a Bayesian net without hidden nodes), then it turns out that for any two such hypotheses c, h (with components c_i, h_i at node i), we have

$$d_{KL}(P, h) - d_{KL}(P, c) = \sum_{i=1}^{n} d_{CP}(P, c_i, h_i). \tag{2}$$

Setting c to the best hypothesis for P, we get a very useful expression for the error of any hypothesis h compared to optimal.

4.3. Minimizing the log loss

Our choice of distance measure is geared towards maximum likelihood algorithms, which given m samples $\{s_i\}$ from a distribution P, try to find a hypothesis $h \in H$ that minimizes

$$\sum_{i=1}^{m} P^*(s_i) \ln \frac{P^*(s_i)}{h(s_i)},$$

where P^* is the empirical distribution consisting of just the m samples. This is equivalent to picking an $h \in H$ that minimizes $\mathbf{E}^*(-\ln h(\cdot))$, where $\mathbf{E}^*$ denotes the expectation with respect to P^* (rather than the true expected value $\mathbf{E}$ which is computed with respect to P).

EXAMPLE 4 Consider the toy situation described in section 2, where we are trying to estimate the heads probability of a coin given the values $X_1, X_2, \ldots, X_m$ of a few coin tosses. In this case our hypothesis class $H = [0, 1]$, and any particular hypothesis $p \in H$ has observed value $\mathbf{E}^*(-\ln p(\cdot)) = -\frac{1}{m}\ln(p^k(1-p)^{m-k})$, where k is the number of heads in the sample data $X_1, \ldots, X_m$. The maximum likelihood hypothesis (and therefore also the one that minimizes d_{KL} with respect to the observed distribution) is then easily verified to be $p = \frac{k}{m}$, the sample average.

The maximum likelihood hypothesis can always be found by an exhaustive search through all the hypotheses in H. For the resulting answer to be good, we require only that the estimates $\mathbf{E}^*(-\ln h(\cdot))$ be close to $\mathbf{E}(-\ln h(\cdot))$. The functions $F = -\ln H = \{-\ln h : h \in H\}$ are called the *log loss* functions for H. The main question, then, is: how many samples are needed for the estimates $\mathbf{E}^*(f)$ to be accurate, for all $f \in F$?

This question can be answered by various extensions of Hoeffding's inequality (Pollard, 1984). The d_{KL} distance measure causes a slight problem, though. The hypothesis class H of probability distributions consists of functions whose range is $[0, 1]$, so that the corresponding log loss functions have range $[0, \infty)$. However, Hoeffding's inequality applies to bounded random variables. One way of dealing with this is formalized nicely by Abe, Takeuchi, and Warmuth (1990). Let $H^{(\epsilon,\gamma)} \subseteq H$, called an (ϵ, γ)-*bounded approximation* of H, have the property that for all $h \in H, \exists h^{(\epsilon,\gamma)} \in H^{(\epsilon,\gamma)}$ such that for all x,

$$h^{(\epsilon,\gamma)}(x) \geq \gamma \text{ and } \ln \frac{h(x)}{h^{(\epsilon,\gamma)}(x)} \leq \epsilon.$$

That is, we restrict attention to a subset[1] of H that provides a good approximation to all of H and whose log loss functions are bounded (in particular, they have range $[0, \ln \frac{1}{\gamma}]$). This is often not hard to manage. Suppose our hypothesis class is for the conditional probabilities at a particular node in a Bayesian net. We will demonstrate how to handle our two main example cases from Section 3.

When H is the hypothesis class consisting of explicitly tabulated conditional probabilities for Boolean variables, each probability value can be adjusted as necessary:

- If $p \in [\gamma, 1-\gamma]$, the adjusted value $p^{(\epsilon,\gamma)} = p$;
- If $p \in [0, \gamma)$, then $p^{(\epsilon,\gamma)} = \gamma$;
- If $p \in (1-\gamma, 1]$, then $p^{(\epsilon,\gamma)} = 1-\gamma$.

To ensure that $\ln p$ exceeds $\ln p^{(\epsilon,\gamma)}$ by at most ϵ (where we are assuming $\epsilon \leq \frac{1}{2}$), we need only set $\gamma \leq \frac{\epsilon}{1+\epsilon}$.

When H is the family of noisy-OR functions for $k \geq 1$ parents, any $h \in H$ with parameters $\Delta, p_1, \ldots, p_k$ can be shifted to $h^{(\epsilon,\gamma)}$ with new parameters $\Delta', p_1', \ldots, p_k'$ using the transformation

$$\gamma = \left(\frac{\epsilon}{2}\right)^k, \;\; p_i' = \min\left\{p_i, \frac{1}{1+\epsilon}\right\}, \;\; \Delta' = \max\{\Delta, \gamma\}.$$

We are assuming here that $\Delta \leq \frac{1}{4}$ and $\epsilon \leq \frac{1}{2}$, as before.

EXAMPLE 5 Let us return to our toy example of trying to estimate the bias of a coin. Our learning algorithm will sample some coin tosses $X_1, \ldots, X_m \in \{0, 1\}$, compute the sample average p^*, and then output some shifted value p^{**}. If p is the true bias of the coin, we want

$$d_{KL}(p, p^{**}) = p \ln \frac{p}{p^{**}} + (1-p) \ln \frac{1-p}{1-p^{**}} \ \leq \ \epsilon$$

for some $\epsilon \in (0, \frac{1}{2}]$. We can shift as in the first case above, setting $p^{**} = p^{*(\epsilon/2, \gamma)}$, where $\gamma = \frac{\epsilon/2}{1+\epsilon/2}$. We then observe (for instance by calculus) that for $p \in [0, 1]$ and $p^{**} \in [\gamma, 1-\gamma]$, we have

$$d_{KL}(p, p^{**}) \ \leq \ \frac{(p-p^{**})^2}{2\gamma(1-\gamma)} \ < \ \frac{25(p-p^{**})^2}{16\epsilon}.$$

Thus, noting that $|p - p^{**}| \leq |p - p^*| + |p^* - p^{**}| \leq |p - p^*| + \frac{\epsilon}{2}$, it is sufficient to ensure that $|p - p^*| \leq \frac{3\epsilon}{10}$ in order to guarantee $d_{KL}(p, p^{**}) \leq \epsilon$. Using Hoeffding's inequality, this can be achieved with confidence $\geq 1 - \delta$ by selecting $m = \frac{6}{\epsilon^2} \ln \frac{2}{\delta}$ samples.

Let us return to our main case, where H consists of probability distributions for an entire Bayesian net without hidden nodes, specifically $H = H_1 \times H_2 \times \cdots \times H_n$, where H_i is the hypothesis class of conditional probabilities at node i. We have seen above how to obtain (ϵ, γ)-bounded approximations for various hypothesis classes H_i. A little algebra shows that:

LEMMA 1 *$H_1^{(\epsilon/n, \gamma)} \times \cdots \times H_n^{(\epsilon/n, \gamma)}$ is an (ϵ, γ^n)-bounded approximation of H.*

Next, let us consider the case of Bayesian nets with hidden nodes. For an n-node net, let H denote the hypothesis class if we pretend that none of the nodes are hidden; that is, each $h \in H$ maps $X = X_O \times X_H$ into $[0, 1]$, where X_O and X_H are as defined in Section 3. Let H' be the corresponding class when there are k hidden nodes, so that we can set up a surjective mapping $\psi : H \to H'$, whereby, as in equation (1), for all $h \in H, x \in X_O$, we have

$$\psi(h)(x) = \sum_{y \in X_H} h(x, y). \tag{3}$$

This sum should be interpreted as the equivalent integral if X_H is a continuous space. Now, if each $h \in H$ has range $[\rho, 1]$ for some $\rho > 0$, the corresponding $\psi(h)$ has range $[|X_H|\rho, 1]$. For instance, if each node is Boolean, then $\psi(h)$ has range $[2^k \rho, 1]$. We then get the following lemma.

LEMMA 2 *Let $H^{(\epsilon, \gamma^n)} = H_1^{(\epsilon/n, \gamma)} \times \cdots \times H_n^{(\epsilon/n, \gamma)}$ be the hypothesis class obtained from H by shifting the conditional probability functions at each node. Then the corresponding class $\psi(H^{(\epsilon, \gamma^n)})$ where k of the nodes are treated as hidden is an $(\epsilon, |X_H|\gamma^n)$-approximation of $\psi(H)$.*

To summarize, assume we have an algorithm that draws m samples from a distribution P and tries to find the best fitting hypothesis in H. The algorithm evaluates each $h^{(\epsilon,\gamma)} \in H^{(\epsilon,\gamma)}$ by computing the empirical log loss $\mathbf{E}^*(-\ln h^{(\epsilon,\gamma)})$ and returns the hypothesis with the smallest value. The next section describes one natural way to derive the sample complexity m that guarantees that all these estimates are reasonably accurate.

5. A basic sample complexity argument

To illustrate some of the ideas discussed in the previous section and introduce some techniques that will be generalized later, we will start by deriving the sample complexity in the simplest case: when all the variables are Boolean, there are no hidden units, and the hypothesis class is that of explicit conditional probability tables. Let P denote the target distribution. As always, we will not require that P correspond exactly to one of our hypotheses H, and we will seek a hypothesis $h \in H$ which, with probability $> 1-\delta$, is at most distance ϵ further away from P than the best hypothesis in H.

Fix a node y in the graph, with k parent variables π. The parents can take on 2^k different values. The hypothesis class of conditional probability distributions at this node, denoted H_y, has the feature that each $h \in H_y$ can be described by 2^k real numbers, corresponding to $h(y=0|\pi=j)$ as j varies over $\{0,1\}^k$, and of course $h(y=1|\pi=j) = 1-h(y=0|\pi=j)$. In this sense we can write $H_y = [0,1]^{2^k}$. This can be further decomposed by conditioning on the value of the parent variables: $H_{y,j} = [0,1]$ is the hypothesis class for the specific case when $\pi = j$. We have to find a hypothesis that comes close to the optimal $h_{opt}(y=0|\pi=j)$, henceforth abbreviated $h_{opt}(0|j)$. One can show quite easily that $h_{opt}(i|j) = P(y=i|\pi=j)$, as expected. When (σ,γ)-shifted, this becomes $h_{opt}^{(\sigma,\gamma)} \in H_{y,j}^{(\sigma,\gamma)} = [\gamma, 1-\gamma]$. We will use the particular parameters $\sigma = \frac{\epsilon}{3n}$, $\gamma = \frac{\epsilon/3n}{1+\epsilon/3n}$, where n is the number of nodes in the graph. We will assume for convenience that $\epsilon \leq 6n$.

For any parent value j, in order to find a hypothesis close to $h_{opt}(\cdot|j)$, the general learning algorithm described above would try all possible hypotheses $h \in H_{y,j}^{(\sigma,\gamma)}$ and return the one that best fits the sample data (that which minimizes $-\ln h$ with respect to the data). However, $H_{y,j}^{(\sigma,\gamma)}$ is infinitely large, and we would like uniform convergence of our estimates on all of its hypotheses. We circumvent this difficulty by selecting a finite but representative subset $H'_{y,j} \subset H_{y,j}^{(\sigma,\gamma)}$, specifically $H'_{y,j} = \{h \in H_{y,j}^{(\sigma,\gamma)} : \exists k \in \mathbf{N}, h(0|j) = \gamma(1+\frac{\epsilon}{3n})^k\}$. Then $|H'_{y,j}| \leq \frac{6n}{\epsilon}\ln\frac{1}{\gamma}$. For any hypothesis $h \in H_{y,j}$, there is a corresponding $h' \in H'_{y,j}$ such that $\ln\frac{h}{h'} \leq \frac{2\epsilon}{3n}$. The analysis now breaks down into two parts:

(a) We will ensure that with probability $> 1-\frac{\delta}{n}$, for all $j \in \{0,1\}^k$, for all $h \in H'_{y,j}$, the empirical estimate $\mathbf{E}^*(-\ln h)$ is within α_j of its true value $\mathbf{E}(-\ln h)$, where $\alpha_j = \epsilon/6n2^{k/2}\sqrt{P(\pi=j)}$. That is, we will permit more error on those j that are unlikely. Thus, with probability $> 1-\frac{\delta}{n}$, our selected hypotheses $h(\cdot|j) \in H'_{y,j}$ will have $\mathbf{E}(\ln h_{opt}(\cdot|j) - \ln h(\cdot|j)) \leq 2\alpha_j + \frac{2\epsilon}{3n}$ for all j, whereupon (thinking of h_{opt}, h as

conditional probability distributions for node y):

$$d_{CP}(P, h_{opt}, h) = \sum_{j \in \{0,1\}^k} P(\pi = j) \sum_{i \in \{0,1\}} P(i|j) \ln \frac{h_{opt}(i|j)}{h(i|j)}$$
$$\leq \frac{2\epsilon}{3n} + \sum_{j \in \{0,1\}^k} \frac{\epsilon \sqrt{P(\pi = j)}}{3n 2^{k/2}} \leq \frac{\epsilon}{n},$$

where the last step follows from the Cauchy-Schwarz inequality. Equation (2) then tells us that, with probability $> 1 - \delta$, $d_{KL}(P, h) \leq d_{KL}(P, h_{opt}) + \epsilon$, as required.

(b) How many samples are needed? Hoeffding and Chernoff bounds show that in order to get an α_j-accurate estimate of $\mathbf{E}(-\ln h(\cdot|j))$ for one specific hypothesis h, with confidence $1 - \delta$, it is sufficient to draw $\frac{288 n^2 2^k}{\epsilon^2} \ln^2 \frac{1}{\gamma} \ln \frac{3}{\delta}$ samples. Suppose that no node has more than k parents. The sample complexity of learning is then upper-bounded by $\frac{288 n^2 2^k}{\epsilon^2} \ln^2 \left(1 + \frac{3n}{\epsilon}\right) \ln \frac{18 n^2 2^k \ln(1+3n/\epsilon)}{\epsilon \delta}$.

Incidentally, the uniform distribution over $\{0,1\}^n$ has KL-distance at most n from any other distribution on this space. One can use this as a guide and select error rates of the form $\epsilon = \alpha n$, for small constants α.

The most interesting term in the bound we have derived is 2^k, because it suggests that reducing the connectivity of the graph can lead to tremendous reductions in sample complexity. This is not simply an aberration caused by a loose upper bound; it is easily shown to be a necessary term in the sample complexity. The usual manner of doing this involves demonstrating that, if a learning algorithm does not draw enough samples, then there is some target distribution for which it has a decent chance of failing. This kind of analysis can be carried out for the simple hypothesis class currently under consideration. However, to illustrate that it is not just a pessimistic worst-case bound, we will adopt a slightly different approach, by showing that a very natural learning algorithm will have trouble learning a specific, extremely simple, distribution if it does not get enough training data.

Consider a learning algorithm that sets the conditional probability tables at each node according to sample averages. For a specific node y in the underlying graph, if it does not receive any training data with some specific parent value $\pi = j$, then it picks some preset hypothesis, say $h(0|j) = \frac{1}{2}$. Let the target distribution P treat all variables as independent, where the probability that any of them is 1 is $\frac{3}{4}$. In particular, this distribution is necessarily contained in the hypothesis class of explicit conditional probability tables, regardless of the dependency graph.

For each node y in the graph with $\geq k$ parents, if there are no training samples available for a constant fraction of its possible parent values, then the resulting hypothesis at that node will have a d_{CP} error of $\Omega(1)$. If there are $\Omega(n)$ such nodes, the overall hypothesis will have KL-distance $\Omega(n)$ from P. Thus at least $\Omega(2^k)$ samples must be seen, for some constant error rate $\alpha = \epsilon/n$. This analysis can be refined quite routinely to incorporate a dependence on $(\frac{n}{\epsilon})^{2/3}$ in the lower bound.

6. Sample complexity via small covers of function classes

An important aspect of the proof in the previous section was the approximation of an infinitely large hypothesis class by a finite representative subset. Over the course of the last few decades, this technique has been formalized for application to general hypothesis classes. We will briefly outline this theory and then apply it to our case.

6.1. Covering numbers

Following work by Vapnik and Chervonenkis (1982), Dudley (1978), Pollard (1984), and others, Haussler (1992) presents a wonderful treatment of sample complexity based upon the notion of the *covering number* of a set. Let F be a class of functions from some input space X into $[0, M]$. We are interested in the particular case when F is a set of log loss functions $\{-\ln h(\cdot)\}$. We can impose a pseudo metric on these functions that depends upon a probability distribution P on X: for $f, g \in F$,

$$d_1^P(f, g) \;=\; \mathbf{E}(|f - g|) \;=\; \sum_{x \in X} |f(x) - g(x)|P(x).$$

This is not a metric because $d_1^P(f, g) = 0$ does not necessarily imply that $f = g$. An *ϵ-cover* of F is a subset $F' \subseteq F$ such that for any function $f \in F$, there is some $f' \in F'$ with $d_1^P(f, f') < \epsilon$. Although F itself might be gigantic or even infinite, often there is a small (and in particular finite) ϵ-cover of F, and this effectively makes the search space much smaller since we do not require exact learning. The covering number $\mathcal{N}(\epsilon, F, d_1^P)$ is the cardinality of the smallest ϵ-cover of F. We can thus expect the sample complexity of learning F to be bounded linearly by the logarithm of this number, just as for finite F the sample complexity can be bounded linearly in terms of $\log |F|$.

Suppose we draw a set $S = \{s_1, \ldots, s_m\} \subseteq X$ of examples according to the underlying distribution P, and then use this sample data to estimate each of the log loss functions in F. The examples specify an empirical distribution P^* that assigns probability $1/m$ to each of them. The corresponding covering number is thus $\mathcal{N}(\epsilon, F, d_1^{P^*})$.

LEMMA 3 Sample complexity of learning (Pollard, 1984; Haussler, 1992). *Let F be a set of functions[2] from X into $[0, M]$ and let $S = \{s_1, \ldots, s_m\}$ be a set of examples from X, drawn independently according to distribution P. Suppose S is used to calculate empirical estimates $\mathbf{E}^*(f) = \frac{1}{m}\sum_{i=1}^m f(s_i)$ for every $f \in F$. In order to ensure that $\mathbf{P}(\exists f \in F : |\mathbf{E}^*(f) - \mathbf{E}(f)| > \epsilon) < \delta$, it is enough to set*

$$m > \frac{128M^2}{\epsilon^2} \ln \frac{4\mathbf{E}(\mathcal{N}(\frac{\epsilon}{16}, F, d_1^{P^*}))}{\delta}.$$

In other words, the sample complexity of learning depends upon the logarithm of $\mathbf{E}(\mathcal{N}(\epsilon/16, F, d_1^{P^*}))$, where this expectation is taken over the possible sets of sample data S (whose choice depends upon P). We would like a distribution-independent bound on this covering number. More generally, we need a bound on $\mathcal{N}(\epsilon, F, d_1^P)$ that does not depend upon P.

6.2. The pseudo dimension

There is a measure called the *pseudo dimension* (Dudley, 1978; Pollard, 1984), denoted $\mathbf{dim_P}$, which adapts the VC dimension for real-valued functions and which can be used to obtain bounds on covering numbers. Again, fix a hypothesis class H consisting of hypotheses $h : X \to \mathbf{R}$. For some set of examples $S = \{s_1, \ldots, s_m\} \subseteq X$, H is said to *shatter* S if there is some translation vector $t \in \mathbf{R}^m$ such that $\forall b \in \{0,1\}^m, \exists h_b \in H$ for which

$$h_b(s_i) = \begin{cases} > t_i & \text{if } b_i = 1 \\ \leq t_i & \text{if } b_i = 0 \end{cases}.$$

In other words, H shatters S if the m-dimensional vectors $\{(h(s_1), h(s_2), \ldots, h(s_m)) : h \in H\}$, shifted by $-t$, intersect all 2^m orthants. The pseudo dimension $\mathbf{dim_P}(H)$ is the size of the largest set S that is shattered by H. This is a direct generalization of the VC dimension for Boolean concept classes.

For a given node y in a Bayesian net with k parents $\pi = (x_1, \ldots, x_k)$, we will derive the pseudo dimensions of various classes of conditional probability functions. In the case when all variables are Boolean, each such hypothesis is a function $h : \{0,1\}^k \times \{0,1\} \to [0,1]$, i.e., $h(\pi, y) = \mathbf{P}(y|\pi)$.

EXAMPLE 6 The simplest class is that of explicit conditional probability tables, which permits all possible dichotomies over the k parent variables. In this case, the set $S = \{(\pi, 0) : \pi \in \{0,1\}^k\}$ is shattered. Larger samples must include a pair of instances of the form $(\pi, 0)$ and $(\pi, 1)$ and thus cannot be shattered. Therefore the pseudo dimension of this hypothesis class is 2^k.

EXAMPLE 7 The noisy-OR hypothesis class has dimension $k + 1$. First we will exhibit $k+1$ inputs that are shattered by this class. Let e_i denote the k-tuple $(0, 0, 0, \ldots, 1, \ldots, 0)$, with a 1 at position i and a 0 everywhere else, and let $\overline{0}$ denote the all-zeros vector of the same length. Consider the set of $k + 1$ instances $S = \{(e_1, 1), \ldots, (e_k, 1), (\overline{0}, 1)\}$, and the translation vector $t = (\frac{1}{2}, \ldots, \frac{1}{2}, \frac{1}{8})$. Any dichotomy can be realized; choose a noisy-OR function h (parameterized by $p_1, \ldots, p_k, \Delta$) such that

- To make $h(e_i, 1) \leq \frac{1}{2}$, pick $p_i = 0$; otherwise, pick $p_i = 1$;
- To make $h(\overline{0}, 1) \leq \frac{1}{8}$, pick $\Delta = 0$; otherwise, pick $\Delta = \frac{1}{4}$.

Thus, $\mathbf{dim_P}(\text{noisy-OR}) \geq k + 1$.

To see that $\mathbf{dim_P}(\text{noisy-OR}) \leq k+1$, let S be a finite set of m examples that is shattered by this hypothesis class, with respect to some translation $(t_1, \ldots, t_m)$. We can assume that the i^{th} example has the form $(s_i, 0)$, where $s_i \in \{0,1\}^k$, since $\mathbf{P}(y = 1|s) = 1 - \mathbf{P}(y = 0|s)$. S is also seen to be shattered by $F = \{\ln h : h \text{ is a noisy-OR function}\}$, with respect to the translation vector $(\ln t_1, \ldots, \ln t_m)$.

With regard to shattering by F, each of the 2^m possible dichotomies corresponds to the natural logarithm of a particular noisy-OR function h, which is indexed by a particular set of parameters $\{p_1, \ldots, p_k, \Delta\}$. Notice that for any $(x_1, \ldots, x_k) \in \{0,1\}^k$,

$(\ln h)((x_1, \ldots, x_k), 0) = \ln(1-\Delta) + \ln(1-p_1x_1) + \cdots + \ln(1-p_kx_k) = \ln(1-\Delta) + x_1 \ln(1-p_1) + \cdots + x_k \ln(1-p_k)$. In this way, we see that S is also shattered by the class of linear functions in k variables, with respect to translation by $(\ln t_1, \ldots, \ln t_m)$. But this latter class has pseudo dimension $k+1$ by a standard result (Dudley, 1978; Haussler, 1992), whereby $m \leq k+1$, as desired.

EXAMPLE 8 In both of the cases above, the pseudo dimension was the same as the number of parameters needed to describe a hypothesis. This is not a general rule. Consider, for instance, a node y that has no parents and that takes values in the range $(-\infty, \infty)$. The hypotheses for y are indexed by a single real parameter $a \in \mathbf{R}$ and have the form $h_a(y) = \frac{1}{\sqrt{\pi}} e^{-(y-a)^2}$. The pseudo dimension of this class is the same as that of $\{f_a : f_a(y) = (y-a)^2\}$, which is readily seen to be two.

6.3. Bounding the covering number

The sample complexity of learning a Boolean function is proportional to the VC dimension of the relevant concept class. Similarly, we might expect that for a class of functions F with a distribution P over the instance space, it is possible to bound $\log \mathcal{N}(\epsilon, F, d_1^P)$ in terms of $\mathbf{dim_P}(F)$.

LEMMA 4 Bounding covering numbers (Pollard, 1984; Haussler, 1992). *Let F be a set of functions from X into $[0, M]$, such that $\mathbf{dim_P}(F) = d < \infty$. Let P be any probability distribution on X. Then for any $\epsilon \in (0, M]$,*

$$\mathcal{N}(\epsilon, F, d_1^P) < 2\left(\frac{2eM}{\epsilon} \ln \frac{2eM}{\epsilon}\right)^d.$$

It is vital that this holds for any distribution P. Combined with the results mentioned in Section 6.1, it shows that the sample complexity of learning F can be bounded by a quantity which is proportional to $\mathbf{dim_P}(F)$.

6.4. The entire network

Let us now consider the case where we are learning conditional probabilities for a Bayesian net without hidden nodes. Let H_i denote the possible conditional probability hypotheses for some node i, let $H_i^{(\sigma,\gamma)}$ be its shifted approximation (for some σ, γ whose values are not important for this discussion), and let $F_i = \{-\ln h_i^{(\sigma,\gamma)} : h_i^{(\sigma,\gamma)} \in H_i^{(\sigma,\gamma)}\}$ be the corresponding class of log loss functions. We can compute $\mathbf{dim_P}(H_i)$ as in the examples above and then observe that $\mathbf{dim_P}(F_i) = \mathbf{dim_P}(H_i^{(\sigma,\gamma)}) \leq \mathbf{dim_P}(H_i)$, where the latter statement follows from the fact that $H_i^{(\sigma,\gamma)} \subset H_i$. This gives us a bound on the covering number of the log loss functions at node i, which can be used to derive an upper bound

on the sample complexity of learning a good conditional probability function at that node. However, we would also like some kind of sample complexity bound for the entire net.

From now on, we will drop the (σ, γ)-type superscripts for convenience, with the understanding that these are implied. If the net has n nodes, then the entire net's hypothesis class is $H = H_1 \times H_2 \times \cdots \times H_n$. Let F_i be the class of log loss functions at node i, as defined above. The log loss of some $h = (h_1, \ldots, h_n) \in H$ is $-\ln h_1 - \ln h_2 - \cdots - \ln h_n$. Thus the class of log loss functions for the entire net is $F = F_1 + F_2 + \cdots + F_n$. We want to bound the covering number of F in terms of the numbers for $F_1, \ldots, F_n$. In order to do this, we think of each function $f_i \in F_i$ as having input space X, that is to say, it is a function on all the variables and just ignores the inputs that it does not need. The following simple lemma does the trick.

LEMMA 5 Addition rule. *Let $F_1, \ldots, F_n$ be classes of functions from the same input space X into $[0, M]$. Then for any distribution P,*

$$\mathcal{N}(\epsilon, F_1 + \cdots + F_n, d_1^P) \leq \mathcal{N}(\epsilon/n, F_1, d_1^P)\mathcal{N}(\epsilon/n, F_2, d_1^P) \cdots \mathcal{N}(\epsilon/n, F_n, d_1^P).$$

Proof: Fix a distribution P and let $C_1, \ldots, C_n$ be ϵ/n-covers for $F_1, \ldots, F_n$. Because the triangle inequality holds for the pseudo metric d_1^P, $C_1 + C_2 + \cdots + C_n$ is an ϵ-cover for $F_1 + \cdots + F_n$, and it has cardinality $|C_1| \cdot |C_2| \cdots |C_n|$. ■

If each F_i has pseudo dimension d_i and has values in the range $[0, M_i]$, then we can set $\overline{M} = \max_i M_i$ and $d = d_1 + \cdots + d_n$, and use the addition rule in conjunction with lemma 4 to show that F has covering number

$$\mathcal{N}(\epsilon, F, d_1^P) < 2^n \left(\frac{2e\overline{M}n}{\epsilon} \ln \frac{2e\overline{M}n}{\epsilon} \right)^d,$$

whereupon lemma 3 can be invoked to derive the sample complexity.

Let us apply this to our previous example, that of a network where each node has $\leq k$ parents and the conditional probabilities are stored explicitly in tables, giving a total of at most $n2^k$ parameters. To learn a hypothesis with accuracy ϵ, each conditional probability value should be $(\epsilon/2n, \frac{\epsilon/2n}{1+\epsilon/2n})$-shifted, so that an upper bound on the value of our log-loss functions for the entire net is $M = n \ln(1 + \frac{2n}{\epsilon})$ (Lemma 1). The pseudo dimension at each node is at most 2^k, so by Lemma 3, the overall sample complexity is bounded above by

$$\tilde{O}\left(\frac{n^2}{\epsilon^2} \left(n2^k + \ln \frac{1}{\delta} \right) \right),$$

where the $\tilde{O}(\cdot)$ notation suppresses multiplicative terms of the order of $\log^{O(1)} \frac{n}{\epsilon}$.

The convenient generality of this approach has a price: the results are less tight than the specialized proof presented earlier (there is an additional multiplicative factor of n here). However, the key term – the dependence on 2^k – remains the same, and makes it clear that the key to reducing sample complexity is making sure that the conditional probability

functions at the nodes have low pseudo dimension. For instance, if we use noisy-OR functions instead of explicit conditional probability tables, the 2^k in the sample complexity bound gets reduced to k.

6.5. *Bayesian nets with hidden variables*

Consider a net with n nodes such that the conditional probability function at node i is $h_i \in H_i$; again, we are omitting the (σ, γ)-type superscripts for convenience. As before, we will pretend that each such function h_i maps X into $[0, 1]$, ignoring the inputs that it does not need. The function $h \in H$ computed by the entire net is simply the product of these components. If the net has k hidden nodes, however, the situation is much more complicated, and, as in equation (3), we can associate with each $h \in H$ a corresponding hypothesis $\psi(h) \in \psi(H)$, where $\psi(H)$ denotes the class of hypotheses for the net with hidden variables.

Let H_i, H, and $\psi(H)$ have corresponding log loss classes F_i, F, and F^ψ. We are interested in computing covering numbers for F^ψ. We already know (via Lemma 5) how to obtain covering numbers for F.

LEMMA 6 Hidden variable rule. *Let H be a class of functions from X into $[0, 1]$ and let $\psi(H)$ be the corresponding class with k hidden variables as defined above in equation (3). We can write $X = X_H \times X_O$, where X_H, X_O refer to the hidden and observable variables, respectively. Then for any distribution P over X_O, $\mathcal{N}(\epsilon, F^\psi, d_1^P) \leq \mathcal{N}(\epsilon/|X_H|, F, d_1^{\hat{P}})$, where the distribution $\hat{P}$ over X is an extension of P which is uniform over the k hidden variables (i.e., $\hat{P}(x, y) = P(x)/|X_H|$ for all $x \in X_O, y \in X_H$).*

Proof: Let C be an $\epsilon/|X_H|$-cover of F with respect to $d_1^{\hat{P}}$, and let $C^\psi = \{-\ln \psi(h) : -\ln h \in C\}$. Now pick any $-\ln \psi(h) \in F^\psi$. There exists $-\ln h_0 \in C$ such that $d_1^{\hat{P}}(-\ln h, -\ln h_0) < \epsilon/|X_H|$, that is,

$$\begin{aligned}
\frac{\epsilon}{|X_H|} &> \sum_{x \in X_O} \sum_{y \in X_H} |\ln h(x, y) - \ln h_0(x, y)|\, \hat{P}(x, y) \\
&= \sum_{x \in X_O} \sum_{y \in X_H} \left| \ln \frac{h(x, y)}{h_0(x, y)} \right| \frac{P(x)}{|X_H|} \\
&\geq \sum_{x \in X_O} \left| \ln \frac{\sum_{y \in X_H} h(x, y)}{\sum_{y \in X_H} h_0(x, y)} \right| \frac{P(x)}{|X_H|} \\
&= \frac{1}{|X_H|} \sum_{x \in X_O} \left| \ln \frac{\psi(h)(x)}{\psi(h_0)(x)} \right| P(x) \\
&= \frac{1}{|X_H|} d_1^P(-\ln \psi(h), -\ln \psi(h_0)),
\end{aligned}$$

as desired, so that C^ψ is an ϵ-cover for F^ψ. The third line in the derivation stems from the fact that, for any two positive-valued sequences $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$,

$$\sum_i \left| \ln \frac{a_i}{b_i} \right| \;\geq\; \left| \ln \frac{\sum_i a_i}{\sum_i b_i} \right|$$

because the left-hand side is always at least $\max\{\ln \max_i \{\frac{a_i}{b_i}, 1\}, -\ln \min_i \{\frac{a_i}{b_i}, 1\}\}$, whereas the right-hand side is always at most this amount. ■

From all this we get an upper bound on the sample complexity of learning $\psi(H)$ that is about $\ln |X_H|$ times our upper bound for learning H. If all the variables are Boolean, the increase is only $O(k)$, where k is the number of hidden variables. This justifies the use of hidden units from a sample complexity viewpoint, particularly if the use of these units allows the connectivity of the underlying graph to be reduced significantly.

THEOREM 1 Learning Bayesian nets. *Given a Bayesian net with fixed structure over n variables, divide the joint instance space X into its observable and hidden components X_O and X_H. Let the hypothesis class of conditional probability functions at each node i have pseudo dimension bounded by d_i and form their $(\epsilon/2n, \gamma)$-bounded approximations for some $\gamma > 0$. Setting $d = d_1 + \cdots + d_n$, the sample complexity of learning is bounded above by*

$$\tilde{O}\left(\frac{n^2}{\epsilon^2} \left(\ln \frac{1}{\delta} + d \max\{\ln |X_H|, 1\} \right) \right),$$

where the $\tilde{O}(\cdot)$ notation suppresses multiplicative terms on the order of $\log^{O(1)} \frac{n \log |X_H|}{\epsilon\gamma}$.

7. Directions for further study

This paper puts the task of learning fixed-structure Bayesian networks in a suitable PAC framework and demonstrates that a judicious choice of conditional probability function classes can lead to tremendous reductions in sample complexity. This partially corroborates, for instance, the popularity of noisy-OR functions rather than general conditional probability tables in much recent experimental work. At the same time, hidden nodes are seen to be relatively inexpensive in terms of sample complexity, compared to the large savings in network connectivity that can result from their use.

If these results are to be extended to the case where the structure of the network is not prespecified, a good starting point might be to restrict attention to the realistic scenario where an approximate structure is known and only small deviations from this are permitted. Another fruitful line of research is deriving good learning algorithms for various classes of conditional probability functions. For instance, there has been some experimental work on learning using gradient-based heuristics (Russell, Binder, Koller, & Kanazawa, 1995); perhaps variants of these techniques can be shown to perform well on specific hypothesis classes like noisy-OR.

Acknowledgments

The author would like to thank Stuart Russell for much help and encouragement, and the editors and anonymous referees for many useful suggestions.

Notes

1. There is actually no need for $H^{(\epsilon,\gamma)}$ to be chosen from H. An altogether different function class that approximates H well would also work. We have only adopted this assumption because it simplifies matters slightly in Section 6.4.
2. The function class F needs to satisfy some mild measurability conditions which need not concern us in practical settings. Further details can be found in (Haussler, 1992).

References

Abe, N., Takeuchi, J., & Warmuth M. (1990). Polynomial learnability of probabilistic concepts with respect to the Kullback-Leibler divergence. *ACM Conference on Computational Learning Theory*. San Mateo, CA: Morgan Kaufmann.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory.* New York: John Wiley.

Dudley, R. M. (1978). Central limit theorems for empirical measures. *Annals of Probability, 6*, 899-929.

Friedman, N., & Yakhini, Z. (1996). On the sample complexity of learning Bayesian networks. *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*. San Mateo, CA: Morgan Kaufmann.

Haussler, D. (1992). Decision-theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation, 100*, 78-150.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems.* San Mateo, CA: Morgan Kaufmann.

Pollard, D. (1984). *Convergence of stochastic processes.* New York: Springer-Verlag.

Russell, S., Binder, J., Koller, D., & Kanazawa, K. (1995). Local learning in probabilistic networks with hidden variables. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*. Los Altos, CA: William Kaufmann.

Valiant, L. (1984). A theory of the learnable. *Communications of the ACM, 27*, 1134-1142.

Vapnik, V. N. (1982). *Estimation of dependencies based on empirical data.* New York: Springer-Verlag.

Received July 11, 1996
Accepted July 28, 1997
Final Manuscript July 29, 1997