Efficient Estimation exploiting Independence Constraints

Russell Greiner
Dept of Computer Science
University of Alberta
Edmonton, AB T6G 2H1, Canada
Email: greiner@cs.ualberta.ca

Dale Schuurmans
Dept of Computer Science
University of Waterloo
Waterloo, ON N2L 3G1, Canada
Email: dale@cs.uwaterloo.ca

Chris O'Brien
Dept of Computer Science
University of Alberta
Edmonton, AB T6G 2H1, Canada
Email: chrisb@cs.ualberta.ca

Abstract

There is an obvious way to obtain an unbiased estimate of the response to a query $P(\, C = c \,|\, \mathbf{E} = \mathbf{e}\,)$ from a sufficiently large data sample: just use the empirical frequency. Unfortunately, this can have high variance (or even be undefined) when the conditioning event $\mathbf{E} = \mathbf{e}$ is rare. In some situations, we are able to reduce this variance by exploiting our knowledge about the underlying distribution — in particular, by using a known set of probabilistic indepedencies amongst the variables, encoded in the structure of a Bayesian belief network. The report investigates ways to exploit those independencies when estimating the response to a probabilistic query. In particular, we show a general approach that produces an unbiased estimate with small variance. Of independent interest, we also present a closed form expression for the (asymptotic) variance of a response wrt a given structure and sample.

1 Introduction

To diagnose patients, a doctor needs to determine the probability that a patient with a specified set of symptoms has a specific disease — *i.e.*, to compute the value $q = P(\, C = c \,|\, \mathbf{E} = \mathbf{e}\,)$. Given a training sample that includes the symptoms and diseases of a large set of patients, we can estimate this conditional probability just using the observed frequency: *i.e.*, as the ratio of the number of patients with this same set of symptoms and the disease, divided by the number of patients with these symptoms — $\#(C = c, \mathbf{E}_1 = \mathbf{e}_1, \ldots, \mathbf{E}_k = \mathbf{e}_k)/\#(\mathbf{E}_1 = \mathbf{e}_1, \ldots, \mathbf{E}_k = \mathbf{e}_k)$. While this "query maximum likelihood" (ML_Q[1]) estimate is unbiased, it can be problematic, especially when this specific set of symptoms is rare — *e.g.*, if each of the $k = 100$ symptoms occurs half the time, independently of each other. Here, we anticipate our sample will have no instances that match this "probability $1/2^{100}$" conditioning event, and so the empirical ratio will be undefined (think "$0/0$").

[1] §2 provides more precise definitions of this, and other, terms.

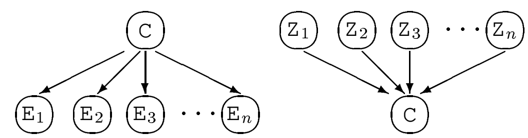


Figure 1: (a) Naïve-Bayes (b) Inverted-Naïve-Bayes

In many such situations, however, we may know something about the underlying distribution; in particular, we may know a set of probabilistic independencies, represented as a belief net (BN; see below). These constraints can be extremely valuable in estimating the probabilities: For example, imagine we knew the distribution corresponds to a "naïve-bayes" structure; see Figure 1(a). Here, we can obtain an unbiased estimate of this conditional probability by first using the sample to instantiate the parameters for the network, then computing the response from the instantiated BN. We will see below that this approach, ML_Θ, can often give a much better estimate, as it has lower variance than simply using the sample; in fact, here it needs a sample that is *exponentially* smaller than is required by the first ML_Q approach!

Unfortunately, this second approach — ML_Θ: first instantiate the parameters of the given structure, then use this instantiated structure to compute the response — is also problematic; there are situations where it will require a sample that is exponentially *larger* than the first ML_Q approach!

This report formally investigates this challenge: What is the best approach to compute

- the response to a single specified $P(\, C = c \,|\, \mathbf{E} = \mathbf{e}\,)$ query, given
- a fixed BN-structure (that is, fixed set of independency claims) and
- a (complete) data sample.

§2 provides the foundations by outlining belief nets, our two standard algorithms and related notions. The paper then deals with following three situations. §3 quickly discusses the situation where the given structure is wrong —

S_1 =

X	Y	Z
1	1	0
1	0	0
0	1	0
0	0	1
0	0	1

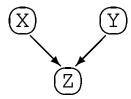


Figure 2: (a) Sample S_1 (b) BN-structure G_{Full}

that is, when the structure omits some essential arc. Our main contribution is in §4, which deals with the situation where the structure is correct. Here, we provide an approach that produces an unbiased estimate, with small variance. Both of these sections deal with the frequentist perspective; §5 quickly overviews the Bayesian perspective, characterizing the posterior distribution of the response. §6 compares our analysis, and results, to the standard literature. We sketch the longer proofs in the Appendix. (For additional details, see [GSO02].)

2 Foundations

We assume there is a stationary underlying distribution $P(\cdot)$ over n (discrete) random variables $\mathcal{V} = \{V_1, \ldots, V_n\}$. We can encode this, or any other distribution, as a "(Bayesian) belief net" (BN) — a directed acyclic graph $B = \langle \mathcal{V}, \mathcal{A}, \Theta \rangle$, whose nodes $\mathcal{V}$ represent variables, and whose arcs $\mathcal{A}$ represent dependencies. More imporantly, the *absense* of an arc connecting a pair of nodes X and Y indicates there is no additional dependency between these variables, given the rest of the graph [Pea88]. If this set of independencies matches the true underlying distribution, we say the graph is "correct", as it is an "I-map of the distribution".

Each node $D_i \in \mathcal{V}$ also includes a conditional-probability-table (CPtable) $\theta_i \in \Theta$, that specifies how D_i's values depend (stochastically) on the values of its parents. In particular, given a node $D \in \mathcal{V}$ with immediate parents $\mathbf{F} \subset \mathcal{V}$, the parameter $\theta_{D=d|\mathbf{F}=\mathbf{f}}$ represents the network's term for $P(\,D=d\,|\,\mathbf{F}=\mathbf{f}\,)$ [Pea88].

Notice these independencies reduce the dimensionality of the space. *E.g.*, while $O(2^n)$ parameters are required to specify the distribution over n binary variables in general, a distribution that corresponds to a naïve-bayes model (Figure 1(a)) requires only $O(n)$ parameters.

A user can pose *queries* to a distribution (whether it is encoded by a belief net or not), each of the form "$P(\,C = c\,|\,\mathbf{E}=\mathbf{e}\,) = ?$", where $C \in \mathcal{V}$ is a single "query variable", $\mathbf{E} \subset \mathcal{V}$ is the subset of "evidence variables", and c (resp., $\mathbf{e}$) is a legal assignment to C (resp., $\mathbf{E}$). The distribution assigns a probability $p \in [0, 1]$ to this (conditional) event; we refer to this value as the *response* to the query.

In many situations, we do not have access to a distribution, but only to a *sample* — a set of tuples, drawn iid from the distribution. Our basic challenge is to produce an accurate estimate of the response to a query, based on a (complete) sample.[2] Below we describe two algorithms for this task; the next sections will compare these two, and also present another approach that is appropriate for some situations.

Estimation Algorithm#1: The ML_Q process, suggested above, returns the maximum likelihood (ML) estimate of the response from a complete sample, which is just the empirical frequency (when defined). For example, given the S_1 sample shown in Figure 2(a), ML_Q would return

$$P(\,Z=1\,|\,X=0\,) \;=\; \frac{\#(Z=1,\ X=0)}{\#(X=0)} \;=\; \frac{2}{3}$$

Notice this process uses only the sample; it does not use any belief net structure.

Estimation Algorithm#2: The ML_Θ process uses the given BN structure. It first fills in the CPtable parameters, using their respective ML values. Hence, given the S_1 sample and the structure G_{Full}, ML_Θ would first set $\tilde{\theta}_{X=1} = 2/5$ as 2 of the 5 tuples have $X = 1$, similarly $\tilde{\theta}_{Y=1} = 2/5$; as well as[3]

x	y	$\tilde{\theta}_{Z=1\|X=x,Y=y}$
1	1	0/1
1	0	0/1
0	1	0/1
0	0	2/2

In general, ML_Θ sets each $\theta_{d|\mathbf{f}}$ parameter, corresponding to the $D=d$ assignment to the child given the $\mathbf{F}=\mathbf{f}$ configuration of its parents, to its maximum likelihood estimate from S, written $\tilde{\theta}_{d|\mathbf{f}} = \hat{P}_S(\,D=d\,|\,\mathbf{F}=\mathbf{f}\,)$, which (when defined) is the observed conditional frequency of a child variable $D=d$ given a configuration of its parents $\mathbf{F}=\mathbf{f}$. We let $\tilde{\Theta}^{(G,S)}$ refer to the set of parameters of the BN-structure G, computed in this fashion using the sample S; and let $\tilde{B}^{(G,S)} = \langle G, \tilde{\Theta}^{(G,S)} \rangle$ refer to the associated distribution. Here, $\tilde{B}^{(G,S)}(q)$ is the response it produces to the query q.

ML_Θ then computes the response to the query based on this instantiated belief net $\tilde{B}^{(G,S)}(\cdot)$. Hence, as

$$\begin{aligned} P(\,Z=1\,|\,X=0\,) \;&=\; \textstyle\sum_y P(\,Z=1, Y=y\,|\,X=0\,) \\ &= \textstyle\sum_y P(\,Z=1\,|\,X=0, Y=y\,)\,P(\,Y=y\,|\,X=0\,) \\ &= \textstyle\sum_y P(\,Z=1\,|\,X=0, Y=y\,)\,P(\,Y=y\,) \end{aligned}$$

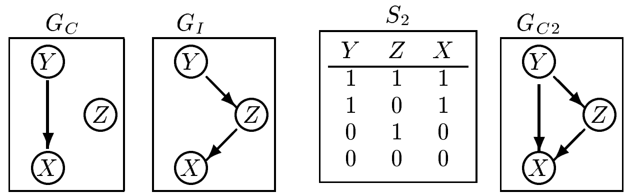
we see

$$\begin{aligned} &\tilde{B}^{(G_{Full},S_1)}(Z=1\,|\,X=0) \;=\; \textstyle\sum_y \tilde{\theta}_{Z=1|X=0,Y=y}\,\tilde{\theta}_{Y=y} \\ &= \quad (0/1 \times 2/5) \;+\; (2/2 \times 3/5) \;=\; 3/5. \end{aligned}$$

(Notice this ML_Θ response differs from the $2/3$ produced by ML_Q.)

[2]A *complete sample* has a value specified for every variable of each tuple.

[3]We will in general use $\tilde{X}$ to refer to the ML estimate of X When necessary, our notation will also explicitly show the sample, and perhaps other factors such as the network structure.



Y	Z	X
1	1	1
1	0	1
0	1	0
0	0	0

Figure 3: Correct and Incorrect Models, and Sample

Our Goal: We will see that there are situations when one or more of these estimates is undefined. Moreover, as this single example shows, even when both estimates are defined, they can still produce different answers. Our task is to find an *unbiased estimator, with minimum variance*. (§5 addresses a variant of this task from the Bayesian perspective; and §7.1 discusses yet other approaches.)

3 Dealing with Incorrect Structures

The ML_Q algorithm is obviously unbiased. Moreover,

Observation 1 *When the BN-structure is an I-map of the true distribution, the ML_Θ approach is asymptotically correct, in that it will recover the true distribution $\tilde{B}^{(G,S)}(\cdot) \sim P(\,\cdot\,)$ if given a sufficiently large sample.*

Proof: The true distribution P corresponds to any graph that is an I-map over the variables, with the correct CPtable values; now observe that the value of each $\tilde{\theta}_{d|\mathbf{f}}$ will approach the true $P(\,D = d \,|\, \mathbf{F} = \mathbf{f}\,)$ value. This means the resulting $\tilde{B}^{(G,S)}$ distribution will (in the limit) produce correct answers to any query; *i.e.*, $\tilde{B}^{(G,S)}(\,c \,|\, \mathbf{e}\,)$ will converge to $P(\,c \,|\, \mathbf{e}\,)$ with probably 1. ∎

The claim that ML_Θ is unbiased, does require the assumption that the structure is correct. Otherwise, when the given structure omits some essential arc, it is easy to see that ML_Θ can produce very bad responses.

To illustrate, suppose we are trying to find the parameters for the incorrect G_I graph structure, shown in Figure 3, when the true distribution is G_C (Figure 3) with say $P(\,Y = 1\,) = 1/2$, $P(\,X = 1 \,|\, Y = 1\,) = P(\,X = 0 \,|\, Y = 0\,) = 1$, and $P(\,Z = 1\,) = 1/2$. We expect data like S_2. Given this data, the first step of the $\text{ML}_\Theta(G_I, S_2)$ process will produce

$$\tilde{\Theta}^{(G_I,S_2)} \quad = \quad \left\{ \begin{array}{lcl} \tilde{\theta}_{+y} & = & 0.5 \\ \tilde{\theta}_{+z|+y} & = & 0.5 \\ \tilde{\theta}_{+z|-y} & = & 0.5 \\ \tilde{\theta}_{+x|+z} & = & 0.5 \\ \tilde{\theta}_{+x|-z} & = & 0.5 \end{array} \right\}$$

These parameters do maximize the likelihood of the S_2 data, for the structure G_I. However, these parameter values are problematic for most queries: Given the true distribution, as expressed by the sample S_2 (and consistent with the structure G_C), the appropriate response to the query $P(\,+y \,|\, +x\,)$ should be 1; however, the value of $\tilde{B}^{(G_I,S_2)}(\,+y \,|\, +x\,) = 1/2$. By contrast, ML_Q will correctly compute this response as 1.

N.b., this is not a small-sample problem; in the limit $\tilde{B}^{(G_I,S_2)}(\,+y \,|\, +x\,)$ will converge to (the incorrect value) $1/2$, while the unbiased estimator ML_Q will converge to (the correct value) 1.[4]

4 Dealing with Correct Structures

The earlier example shows why we expect ML_Q to be better than ML_Θ when the structure is wrong. This section considers the situation when the structure is correct. Here, the story is not so simple.

First, ML_Q will sometimes produce better answers in this correct-structure situation. For example, consider the "inverted-naïve-bayes" structure shown in Figure 1(b), where the query is $P(\,C = 0\,)$. Here, we expect ML_Θ will do poorly even if this structure was correct, as ML_Θ must explicitly estimate the entire set of $n + 2^n$ CPtable entries (assuming binary valued variables) before it could provide a meaningful answer. As each of C's CPtable rows corresponds to some specific assignment $\langle Z_1 = z_1, \ldots, Z_n = z_n \rangle$, ML_Θ can only provide a meaningful estimate of this row if the sample includes (at least) one instance matching this assignment. As each training instance provides information about only one row of C's CPtable, we will have no information about the vast majority of the rows until seeing $O(2^n)$ instances. We will refer to this as the "$0/0$" problem, as ML_Θ needs to compute the ratio of two frequencies, each of which is empirically 0.[5]

By contrast, as ML_Q does not have this constraint, it can provide a reasonable answer to this query even from a small sample.

This example shows that ML_Q can require exponentially fewer training instances than ML_Θ. It is also possible to produce the opposite result:

Recall the simple naïve-bayes structure (Figure 1(a)), where the query is "$P(\,C = 0 \,|\, E_1 = 0, \ldots, E_n = 0\,)$". As noted in §1, ML_Θ is very efficient here, as it will quickly produce reliable values for all $O(n)$ parameters, and can then use the resulting instantiated BN to provide a good

[4] This is consistent with the standard arguments that explain why discriminately training is better than generatively training when the underlying model is wrong [Rip96].

[5] Of course, we could use Laplacian correction, or law of succession or some other trick, to produce a real value for each of these "$0/0$" quantities; see [ZE01, Ris95]. However, the result will not be unbiased, and can lead to problematic results. Imagine, for example, the target concept was $C \equiv (Z_1 \wedge Z_2) \oplus Z_3 \oplus \ldots \oplus Z_n$; where each Z_i was *a priori* "+" half of the time. The Laplacian correction rule would set most entries of C's CPtable to 1/2, which would lead to a marginal probability of $\hat{P}(\,C = +\,) = 1/2$. However, the correct answer should be $P(\,C = +\,) = 1/4$. Of course, different rules will produce different values for these "$0/0$" terms; but no matter what rule we use, we can easily produce a target concept that does very poorly for the associated "$0/0$" value.

answer to this query. By contrast, ML_Q will need to wait until it has seen many (well, at least one!) instance that matches this $\langle Y_1 = 0, \ldots, Y_n = 0 \rangle$ conditioning events; if these Y_i's each occur with probability $P(Y_i = 0) \approx 1/2$, this will take an exponentially large sample.

Notice we have two exponential separations between the sample size requirements for ML_Θ and ML_Q: The inverted-naïve-bayes example (Figure 1(b)) provides a situation where ML_Q requires exponentially fewer instances than ML_Θ, basically because ML_Q can *ignore* the irrelevant parts of the structure, and the naïve-bayes example (Figure 1(a)) provides a situation where ML_Q requires exponentially more instances than ML_Θ, basically because ML_Q can *exploit* the relevant parts of the structure.

Below we present an approach that takes the best of both algorithms: use parts of the structure when appropriate, but only when appropriate.

Estimation Approach#3: The ML_Θ algorithm is basically doing the right thing. Given a sufficiently large sample, it will produce unbiased estimates of the parameters; as noted above, the belief net produced using ML_Θ on these samples will converge (for sufficiently large samples) to the correct distribution (as the structure is an I-map), from which we can produce the correct answers to any query. However, given a limited sample, there is a problem: The above inverted-naïve-bayes case showed that some CPtable entries can be impossible to estimate, as there will be no training instance that match the conditioning event.

Our $\text{RML}_{\Theta,Q}$ approach avoids this problem by first *reducing* the given BN-structure, to produce a smaller structure that avoids the $0/0$-problem (given some other conditions; see below). It then computes the appropriate parameters for *this reduced* structure, which is used to compute the response to the query — *i.e.*, $\text{RML}_{\Theta,Q}$ corresponds to ML_Θ, differing only by dealing with a "smaller" version of the given structure. ($\text{RML}_{\Theta,Q}$ computes $\tilde{B}^{(G',S)}(q)$, but for some reduced $G' \subset G$.)

To illustrate, consider the graph and data sample in Figure 4, where the query is $q = P(Z = 1)$. Notice we cannot obtain meaningful instantiations of all 4 of Z's CPtable entries here based on S_3 — in particular, there are no instances corresponding to $\langle X = 0, Y = 0 \rangle$. However, we can obtain meaningful instantiations of the reduced graph G_{Red}, shown in Figure 4(c). Here, we would set $\tilde{\Theta}^{(G_{Red},S_3)}$:

$$\{ \tilde{\theta}_{+x} = 2/3 \quad \tilde{\theta}_{+z|+x} = 1/2 \quad \tilde{\theta}_{+z|-x} = 1/1 \}$$

and then use this BN to compute $\tilde{B}^{(G_{Red},S_3)}(+z) = 2/3 \times 1/2 + 1/3 \times 1/1 = 2/3$.

S_3

X	Y	Z
1	1	1
1	0	0
0	1	1

G_{Full} $\quad$ G_{Red}

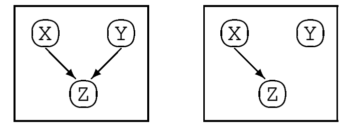


Figure 4: (a) S_3 $\quad$ (b) G_{Full} $\quad$ (c) G_{Red}

4.1 Unbiased Estimator

The challenge is determining the appropriate substructure. As our goal is a minimum-variance unbiased estimate, the first step is to define the space of substructures that could provide an unbiased estimate:

Definition 2 *Given any query $q = P(C = c \mid E_1 = e_1, \ldots, E_k = e_k)$ and any BN-graph (sub)structure $G\ldots$*

$\star$ *Let $G|_q$ refer to the portion of the graph G relevant to the query q — i.e., the nodes and arcs that participate in computing the value of q.*[6] *See [Pea88].*

$\star$ *An evidence variable E is* relevant *to the query q (wrt the structure G) iff it is not d-separated from the query variable C by the remaining evidence variable $\mathbf{E} - E$.*

$\star$ *The graph G' is a subgraph of G, written $G' \subset G$, if we can form G' by removing zero or more arcs from G.*

Theorem 3 *Let G be an I-map of the distribution and $G' \subset G$ be a subgraph. Then $\tilde{B}^{(G',S)}(q)$, the estimator of $q = P(C = c \mid E_1 = e_1, \ldots, E_k = e_k)$ wrt a sample S that uses this G' subgraph, is guaranteed to be unbiased iff all of the following conditions hold:*

C#1. *for every CPtable row $\theta_{D|\mathbf{F=f}}$ in the subgraph $G'|_q$, there is at least one instance in the sample S that matches the conditioning event $\mathbf{F} = \mathbf{f}$ — i.e., there are no $0/0$-problems in $G'|_q$*

C#2. *every E_i that is relevant to q in G, remains relevant to q in G'*

C#3. *the relevant subgraph $G'|_q$ is an I-map, over the nodes in $G'|_q$*

Any subgraph that satisfies these conditions is called a "legal subgraph of G (wrt the query q and sample S)".

Proof (sketch): $\Longrightarrow$: Immediate corollary of Observation 1.

$\Longleftarrow$: C#1 follows from Footnote 5.

To show that C#2 is required, consider again the graph G_{Full} (Figure 4(b)), but now suppose the query is $q' \equiv P(Z = 1 \mid Y = 0)$. The reduced G_{Red} violates this condition as the evidence variable Y, which was not d-separated in the original G_{Full}, is now d-separated in the reduced G_{Red}. Now imagine the true distribution has $Y \equiv Z$ and $P(Z = 1) = 1-\epsilon$, and X is uncorrelated. The estimate of $P(Z = 1 \mid Y = 0)$ from G_{Full} will asymptote to the correct $P(Z = 1 \mid Y = 0) = 0$ value; however, the estimate from G_{Red} will approach $P_{G_{Red}}(Z = 1) = 1 - \epsilon$.

The arguments in §3 show that C#3 is required. ∎

The first two conditions are trivial to check, given a sample and query. The third condition, however, is not: Observe first that, as a subgraph of an I-map graph need not be an I-map, C#3 is not automatically satisfied if we begin with

[6] This excludes the nodes that are d-separated from the query — *e.g.*, the ones that have no evidence or query nodes below them, or which are above an evidence variable, etc.

$$\sum_{\theta(\mathbf{D}\,|\,\mathbf{F}=\mathbf{f})\in\Theta} \frac{1}{\#(\mathbf{F}=\mathbf{f})} \left[\begin{array}{l} \sum_d \frac{1}{\tilde{\theta}_{D=d\,|\,\mathbf{F}=\mathbf{f}}} [\hat{P}(\,D=d, \mathbf{F}=\mathbf{f}, C=c \,|\, \mathbf{E}=\mathbf{e}\,) \;-\; \hat{P}(\,C=c \,|\, \mathbf{E}=\mathbf{e}\,)\, \hat{P}(\,D=d, \mathbf{F}=\mathbf{f} \,|\, \mathbf{E}=\mathbf{e}\,)]^2 \\ -\; [\hat{P}(\,\mathbf{F}=\mathbf{f}, C=c \,|\, \mathbf{E}=\mathbf{e}\,) \;-\; \hat{P}(\,C=c \,|\, \mathbf{E}=\mathbf{e}\,)\, \hat{P}(\,\mathbf{F}=\mathbf{f} \,|\, \mathbf{E}=\mathbf{e}\,)\,]^2 \end{array} \right]$$

Figure 5: Asymptotic Variance of $P(\,C=c \,|\, \mathbf{E}=\mathbf{e}\,)$

an I-map. For example, suppose the correct distribution corresponds to G_C from Figure 3 — $Y \equiv X$ and Z is an independent coin-flip. While G_{C2} (Figure 3) is an I-map of this distribution, its subgraph G_I is not. Second, while there are tests to see if a BN structure is an I-map relative to an empirical sample, they all require that this sample be large. (Of course, for a large sample, it is easy to obtain an accurate estimate…)

This test is trivial in one special case, however:

Observation 4 *Assume that G is an I-map. If G is a poly-tree, then any connected subgraph $G' \subset G$ is an I-map over its nodes.*

Proof: Here we can simply marginalize out the nodes outside the subgraph; see poly-tree inference algorithms [Pea88]. ∎

This means any substructure G' of a poly-tree G will produce an unbiased estimate as long as it avoids $0/0$s (C#1) and its connected component that contains the query node also contains all relevant evidence nodes (C#2).

4.2 Minimize Variance

There can be many legal substructures, that each produce an unbiased estimate of the query response. Here, we want to use the "unbiased substructure" that produces the least variance [Rip96]. This corresponds to a *largest* structure, which involves the most variables:

Theorem 5 *Let G^+ and G be two legal structures wrt q and S, where G^+ extends G by including additional variables — in particular, G^+ includes all of the variables and arcs present in G, as well as additional variables and the arcs connecting those variables to G. Then*

$$Var[\,\tilde{B}^{(G^+,S)}(q)\,] \quad \leq \quad Var[\,\tilde{B}^{(G,S)}(q)\,]\,.$$

This result may at first seem counter-intuitive: As we include more and more variables, we will typically have to estimate more and more parameters, which often means poorer performance as it can be easy to overfit such models. Moreover, there will be fewer tuples that match each parameter in the larger model, which means we have a poorer estimate of each. Nevertheless, this theorem states the opposite claim — namely that these additional parameters *reduce* the variance. As shown in the proof (in the Appendix), our result follows from the fact that conditioning on additional variables will in fact reduce the expected variance.

We close this section with several comments about this largest substructure.

(1) Given a sufficiently large sample — with at least one tuple that matches the conditioning event of each CPtable row[7] — the "largest substructure" will in fact be the entire structure. Here, we see that $\text{RML}_{\Theta,Q}$ reduces to ML_Θ, which is the appropriate algorithm.

(2) The largest subgraph need not be unique; *e.g.*, in the Figure 4 example, dealing with $P(\,Z=1\,)$, we could have removed $X \to Z$ rather than $Y \to Z$, and still have produced a legal subgraph; notice each of these removals produces a largest legal subgraph for the $P(\,Z=1\,)$ query.

In these cases, we may want to actually compute the variance associated with each legal substructure, and then select the argmin. To do this, we first need to estimate the variance of the response, when using the ML_Θ-based approach.

Theorem 6 *Given a fixed correct structure G, a complete sample S, and a query $q \equiv P(\,C=c \,|\, \mathbf{E}=\mathbf{e}\,)$, let $\#(\chi)$ be the number of tuples in S that match the event χ, and let $\hat{P}(\,\chi\,)$ be result of computing the probability of the event χ, using ML_Θ on G and S. (Here, we assume the estimates of G's CPtable parameters are well-defined — i.e., the conditioning event for each CPtable estimate occurred at least once in S.) Then the variance of the response to q, based on the ML_Θ algorithm on the data S and structure G, is given in Figure 5. Here, the summation is over all CPtable entries.*

While this result is only guaranteed asymptotically, [VGH01b] showed that it is quite accurate, over a wide range of situations, even for very small sample sizes.[8] (Of course, one problem with this approach is computational: In general, we need to a belief net inference process to compute each $\hat{P}(\,\chi\,)$, and this process is NP-hard [Coo90].)

(3) There need not be *any* legal substructures — *i.e.*, there are situations where there are no $0/0$-free subgraphs that respect a specified query. For example, consider again the $[Y \to Z \to X]$ structure (G_I from Figure 3, which we are now viewing as an I-map), the simple sample

X	Y	Z
1	1	1

[7] Of course, we only need to consider the $G|_q$ subgraph, which contains the nodes that are not d-seperated from the query.

[8] The sum in Figure 5 decomposes nicely into the sum of the contributions associated with each individual variable, and in fact, of each row of the variable's CPtable. Note that each such CPtable-row term is non-negative. As a sanity check, observe that these terms are each 0 if D and $\mathbf{F}$ are d-seperated from C by $\mathbf{E}$; hence, nodes D that do not influence the response will not contribute to its variance.

and the query $P(Y = 1 \mid X = 1)$. Here we need to keep both $Y \to Z$ and $Z \to X$ links, to respect this $P(Y = 1 \mid X = 1)$ query. However, this G_I is not 0/0-free wrt this sample. In these situations, there can be no unbiased estimators based on subgraphs G_I, and so we need some other measure to decide what to do next; see [ZE01, Ris95], or perhaps restructure the graph (*e.g.*, to include only a direct $Y \to X$ link). However, we will not pursue this issue further in the current paper.

(In this "single parent" context, these "0/0" values are basically irrelevant, as the underlying computation will, in effect, multiply this by 0. For example, $P(+y, +x) = P(+y)[P(+z \mid +y)P(+x \mid +z) + P(-z \mid +y)P(+x \mid -z)]$. While $P(+x \mid -z)$ is 0/0, we multiply this by $P(-z \mid +y)$ which is 0, which means we can ignore this product.[9] Of course, this claim does *not* hold when there are multiple parents, like G_{Full} from Figure 4(b).)

(4) The final issue is computational. Unfortunately, it is NP-hard to finding the *smallest*-variance unbiased estimator; this follows immediately from the following theorem:

Theorem 7 *It is NP-hard to compute the largest legal substructure of a graph G, with respect to a query q and a sample S.*

5 Bayesian Analysis

So far, we have focused on the frequentist approach. In the standard Bayesian models, this task is trivial: Making the standard assumption that *each individual CPtable row is an independent Beta/Dirichlet distribution*, the posterior for any inference can be defined in terms of the posteriors of these rows. This suggests that an algorithm like ML_Θ, which deals with the given structure whose parameters are influenced by the sample, is most meaningful. Moreover, Cooper and Herskovits [CH92] prove that the expected response to a query is just the result of a single belief net inference that uses, for each CPtable entry, the expected value of that entry's posterior distribution. Hence, assuming each row of the G_{Full} structure (Figure 4(b)) has a "*Beta*$(1, 1)$" prior, then the posterior of the rows, after training with S_3, would have $\langle\theta_{X=0}, \theta_{X=1}\rangle \sim$ *Beta*$(1+1,\ 1+2) =$ *Beta*$(2, 3)$ (as S_3 includes one instance with $X = 0$ and two with $X = 1$), $\langle\theta_{Y=0}, \theta_{Y=1}\rangle \sim$ *Beta*$(1+1,\ 1+2)$ and for Z,

x	y	$\langle\theta_{Z=0 \mid X=x, Y=y},\ \theta_{Z=1 \mid X=x, Y=y}\rangle$
1	1	*Beta*$(1+0,\ 1+1)$
1	0	*Beta*$(1+1,\ 1+0)$
0	1	*Beta*$(1+0,\ 1+1)$
0	0	*Beta*$(1,\ 1)$

Note there is a well-defined distribution even for the $\langle X = 0, Y = 0\rangle$ case. (Of course, this follows from the Bayesian claim that we *a priori* know something of these parameters.) The mean of the variable pair distributed as a Beta *Beta*(a, b) is simply $\langle a/(a+b),\ b/(a+b)\rangle$. Hence, we would set $\tilde{\theta}_{X=0} = 2/5$, $\tilde{\theta}_{X=1} = 3/5$, $\tilde{\theta}_{Z=0 \mid X=1, Y=1} = 1/3$ and also $\tilde{\theta}_{Z=0 \mid X=0, Y=0} = 1/2$, etc. Given this instantiation, we next compute the response as a simple BN-inference; *e.g.*, $\tilde{B}^{(Bayes:G_{Full}, S_3)}(Z = 1 \mid X = 0) = \sum_y \tilde{\theta}_{Z=1 \mid Y=y, X=0}\, \tilde{\theta}_{Y=y} = (0/1 \times 2/5) + (1/1 \times 3/5) = 3/5$. [CH92] proves this value is the expected value of this posterior distribution over the query responses.

Recall that our goal (mirroring our frequentist analysis) is to find a *minimum variance* estimate. To do this, we must first again compute the variance of the response to a query. Here, [VGH01a] provides a formula very similar to Figure 5 for the asymptotic variance. That formula is in terms of the posterior distributions of the various CPtable rows, which are $\langle\theta_{D=d_1 \mid \mathbf{F}=\mathbf{f}}, \ldots, \theta_{D=d_k \mid \mathbf{F}=\mathbf{f}}\rangle \sim$ *Beta*$(\alpha_{D=d_1 \mid \mathbf{F}=\mathbf{f}}, \ldots, \alpha_{D=d_k \mid \mathbf{F}=\mathbf{f}})$ for positive values α_i. If the priors were uniform — *i.e.*, *Beta*$(1, 1, \ldots, 1)$ for each parameter — each $\alpha_{D=d_i \mid \mathbf{F}=\mathbf{f}}$ would be $1 + \#(D = d_i, \mathbf{F} = \mathbf{f})$. The resulting Bayesian variance then differs from Figure 5 only by using $\hat{P}(\cdot)$ and $\tilde{\theta}_{D=d \mid \mathbf{F}=\mathbf{f}}$ values based on these expected values of these distributions, and replacing the $\#(\mathbf{F} = \mathbf{f})$ in the denominator by $1 + \sum_i \alpha_{D=d_i \mid \mathbf{F}=\mathbf{f}}$; see [VGH01a].

As the variance, again, depends on the structure used, we can again consider performing the same modifications suggested above, of removing some arcs. It is easy to show that a theorem isomorphic to Theorem 5 holds in this Bayesian context. This means the minimum variance estimator should be the complete original structure — *i.e.*, we can never reduce variance by removing arcs.

6 Related Results

Our $\mathrm{RML}_{\Theta,Q}$ approach works by using only a *subset* of the conditioning event, in situations when the entire conditioning event did not appear in the training sample. While $\mathrm{RML}_{\Theta,Q}$ uses this when dealing with one or more rows of the CPtable, many previous researchers have used this idea wrt a single estimation (*i.e.*, the entire "query"). For example, Kyburg's "Reference class" [Kyb91] takes "the most specific class with suffient support". That is, if there are too few examples of $\langle\mathbf{E}_1 = \mathbf{e}_1, \ldots, \mathbf{E}_k = \mathbf{e}_k\rangle$, Kyburg suggests estimating $P(C = c \mid \mathbf{E}_1 = \mathbf{e}_1, \ldots, \mathbf{E}_k = \mathbf{e}_k)$ using $P(C = c \mid \mathbf{E}_1 = \mathbf{e}_1, \ldots, \mathbf{E}_\ell = \mathbf{e}_\ell)$ for some $\ell < k$,

[9] In general, any estimate that uses 1 parent $\hat{P}_1(+c)$ is identical to an estimate using 0 parents, $\hat{P}_0(+c)$: $\hat{P}_1(+c) = \hat{P}_1(+c \mid +a)\hat{P}_1(+a) + \hat{P}_1(+c \mid -a)\hat{P}_1(-a) = \frac{\#(+c,+a)}{\#(+a)}\frac{\#(+c,+a)}{m} + \frac{\#(+c,-a)}{\#(-a)}\frac{\#(+c,-a)}{m} = \frac{\#(+c,+a)+\#(+c,-a)}{m} = \frac{\#(+c)}{m} = \hat{P}_0(+c)$.

or even $P(C=c)$ (*i.e.*, $\ell = 0$). This corresponds to our approach, with "sufficient support" mapping to "at least one tuple". However, his estimates are not necessarily unbiased; by contrast, our $\text{RML}_{\Theta,Q}$ was motivated by the goal of remaining unbiased.

A related idea, used in "Stein Shrinkage" [EM77] and interpolated Markov models [SDKW98], involves *combining* the estimates based on several different conditioning sets. While these estimates can be biased, they still tend to produce superior estimate, in terms of the L_2 measure. Our $\text{RML}_{\Theta,Q}$ takes just one of these estimates, and is unbiased.

While our variance analysis (Theorem 6) is related to the work in [Kle96, VGH01a], their results are in a Bayesian context, while §4 is frequentist.

7 Conclusion

7.1 Future work

The analysis in this paper suggests a way to produce the *minimum-variance unbiased estimator* of a *single* query, given a data sample with *complete* tuples. It considers the space of estimators that correspond to the *subgraphs* of a specified belief net structure, which use the *simple tabular encoding* for the CPtables. The obvious extensions to this work address each of these issues: 1. We could minimize the expected L_2-error, seeking an estimator that combines bias with variance. 2. We could get superior estimators if we were allowed to make arbitrary changes to the given structure; this even suggests ways to *learn* a structure that would minimize variance (or bias+variance). 3. There are many ways to encode the "conditional probabily function" that connects one variable to its parents; this report deals only with this standard "tabular" parameterization. There could be better estimators that use other encodings, such as noisy-or, or ones based on decision trees [FG96]. 4. Our analysis has considered only complete data. If we consider incomplete data, we may need to use the results from [Kle96]. 5. There are yet other subtle issues that arise if our goal is to deal effectively and efficiently with a *set* of queries — *i.e.*, perhaps to find a single structure that does best for each of a multitude of queries; see [GGS97].

7.2 Contributions

Our results confirm several intuitions: First, if the given structure is wrong, §3 shows one cannot use (an instantiation of) that structure to produce an unbiased estimator of the response to a query. Second, if the given structure is correct, §5 shows it is easy to find minimal-variance unbiased estimators in the *Bayesian* context: just instantiate the structure, and use that "distribution" to produce an answer — *i.e.*, use the Bayesian version of ML_{Θ}. Our main results address the challenges of finding minimal-variance unbiased estimators in the frequentist context, given a correct structure. §4 shows there are many subtleties here. If the data is plentiful (read "enough to include at least one instance matching each CPtable row"), then ML_{Θ} is appropriate. But when the data is not sufficient, it pays to use a more complicated approach that first removes some arcs before using ML_{Θ} on the depleted structure. We give necessary and sufficient conditions for the reduced structure to be unbiased — in terms of the sample (no "$0/0$"s), the query (avoid d-separating relevant evidence nodes), and the underlying distribution (I-map) — and then give a simple criteria for reducing variance: prefer the largest "legal" substructure, with as many variables as possible. We also provide a formula for the (asymptotic) variance of the response to a query (for a given structure and sample), which can be used to evaluate incomparable substructures.

References

[BFH95] Y. Bishop, S. Fienberg, and P. Holland. *Discrete Multivariate Analysis — Theory and Practice*. MIT Press, 1995.

[CH92] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[Coo90] G. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2–3):393–405, 1990.

[Dar00] A. Darwiche. A differential approach to inference in bayesian networks. In *UAI'00*, 2000.

[EM77] B. Efron and C. Morris. Stein's paradox in statistics. *Scientic American*, 236:119–127, 1977.

[FG96] N. Friedman and M. Goldszmidt. Learning Bayesian networks with local structure. In *Proc. Uncertainty in AI*, 1996.

[GGS97] R. Greiner, A. Grove, and D. Schuurmans. Learning Bayesian nets that perform well. In *UAI*, 1997.

[GJ79] M. Garey and D. Johnson. *Computers and Intractability*. Freeman, 1979.

[GSO02] R. Greiner, D. Schuurmans, and C. O'Brien. Efficient estimation exploiting independence constraints. TR, U. Alberta, 2002.

[Kle96] G. Kleiter. Propagating imprecise probabilities in bayesian networks. *Artificial Intelligence*, 88, 1996.

[Kyb91] H. Kyburg. Evidential probability. In *IJCAI91*, 1991.

[Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[Rip96] B. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[Ris95] E. Ristad. A natural law of succession. TR 495–95, Computer Science, Princeton, May 1995.

[Ros97] S. Ross. *Introduction to Probability Models*. Academic Press, 6th edition, 1997.

[SDKW98] S. Salzberg, A. Delcher, S. Kasif, and O. White. Microbial gene identification using interpolated markov models. *Nucleic Acids Res.*, 26(2):544–548, 1998.

[VGH01a] T. Van Allen, R. Greiner, and P. Hooper. Bayesian error-bars for belief net inference. In *UAI01*, 2001.

[VGH01b] T. Van Allen, R. Greiner, and P. Hooper. Error-bars for belief net inference. TR, U. Alberta, 2001.

[ZE01] B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML*, 2001.

A Proofs

Proof of Theorem 6 (sketch): For fixed structure G and query q, let $f(\Theta) = P_{\langle G,\Theta\rangle}(q)$ be the response, as a function of Θ. The Delta method [BFH95] states that, asymptotically,

$$\mathrm{Var}[\ f(\Theta)\ ] \quad \approx \quad \left[\frac{\partial f(\Theta)}{\partial \theta_i}\right]^T \times\ \Sigma\ \times \left[\frac{\partial f(\Theta)}{\partial \theta_i}\right]$$

where $[\partial f(\Theta)/\partial\theta_i]$ is the vector of partial derivatives (wrt each θ_i), and Σ is the (co)variance matrix, over the parameters $\Theta = \langle\theta_i\rangle$. The (co)variance of $\theta_{D=d|\mathbf{F}=\mathbf{f}}$ with itself is $\tilde\theta_{D=d|\mathbf{F}=\mathbf{f}}(1-\tilde\theta_{D=d|\mathbf{F}=\mathbf{f}})/\#(\mathbf{F}=\mathbf{f})$; of two parameters within same row $\langle\theta_{D=d_i|\mathbf{F}=\mathbf{f}}, \theta_{D=d_j|\mathbf{F}=\mathbf{f}}\rangle$ is $-\tilde\theta_{D=d_i|\mathbf{F}=\mathbf{f}}\tilde\theta_{D=d_j|\mathbf{F}=\mathbf{f}}/\#(\mathbf{F}=\mathbf{f})$; and for two parameters in different rows is 0. [GGS97, Dar00] prove that the derivative of the response to the query $q = P(\,C=c\,|\,\mathbf{E}=\mathbf{e}\,)$ wrt $\theta_{D=d|\mathbf{F}=\mathbf{f}}$ is $\frac{\partial f(\Theta)}{\partial\theta_{D=d|\mathbf{F}=\mathbf{f}}} = \frac{\partial P(\,C=c\,|\,\mathbf{E}=\mathbf{e}\,)}{\partial\theta_{D=d|\mathbf{F}=\mathbf{f}}} =$

$$\frac{1}{\theta_{d|\mathbf{f}}}\left[P(\,d,\mathbf{f},c\,|\,\mathbf{e}\,)\ -\ P(\,c\,|\,\mathbf{e}\,)\,P(\,d,\mathbf{f}\,|\,\mathbf{e}\,)\right]$$

Putting together these pieces, we obtain the equation in Figure 5. ([GSO02] provides the details required to justify the claim of asymptotic convergence.) ∎

Proof of Theorem 5: We first consider simple "unconditional" queries, of the form $q = P(\,+c\,)$. We can view the training sample as a random event, and hence the statistics we draw from it as r.v.'s — such as the ML values of the CPtable entries $\Theta = \{\theta_{d|f}\}$ (for a fixed structure G), and the ML estimate of the response to the query using only the sample, call it $\chi_q \in [0,1]$. Now let $\phi_q = \tilde B^{(G,S)}(q)$ be the response to q produced by the structure G and the ML estimates of the CPtable entries Θ. As G is an I-map, observe that ϕ_q is the expected value of χ_q, conditioned on Θ — *i.e.*, $\phi_q = \mathrm{E}[\,\chi_q|\Theta\,]$. (Proof: As a standard belief net inference, ϕ_q is the sum of products of CPtable entries $\phi_q = \sum\prod\theta_{d|\mathbf{f}}$. Hence its (conditional) expected value is $\mathrm{E}[\,\chi_q|\Theta\,] = \sum\mathrm{E}[\,\prod\theta_{d|\mathbf{f}}\,]$. Now notice the CPtable values in each product are independent, and so $\mathrm{E}[\,\prod\theta_{d|\mathbf{f}}\,] = \prod\mathrm{E}[\,\theta_{d|\mathbf{f}}\,]$. As $\mathrm{E}[\,\theta_{d|\mathbf{f}}\,] = P(\,d\,|\,\mathbf{f}\,)$, and the structure is an I-map, $\mathrm{E}[\,\chi_q|\Theta\,] = P(\,+c\,)$.)

Now consider the structure G^+ that extends G by adding another node, as well as the arcs (and associated parameters) required to insure that G^+ is an I-map, wrt all of its variables. Let Θ^+ be the parameters associated with this G^+ structure. Through a simple re-parameterization, we will view Θ^+ as a superset of Θ — *i.e.*, $\Theta^+ = \Theta\cup\Delta$. (*E.g.*, consider adding B to the trivial graph $G = \boxed{A}$, forming $G^+ = \boxed{B\to A}$. Here, we would let $\Theta^+ = \{\theta_{+b}, \theta_{+a}, \tau_{B,A}\}$ where $\tau_{B,A} = \theta_{+a|+b} - \theta_{+a|-b}$. Using these Θ^+ parameters, we can now recover $\theta_{+a|+b} = \theta_{+a} - \theta_{+b}\,\tau_{b,a}$, and then $\theta_{+a|-b} = \theta_{+a|+b} - \tau_{A,B}$.)

[Ros97] shows $\mathrm{E}[\,\mathrm{Var}[\,\chi_q|\Theta^+\,]\,] =$

$$\mathrm{E}[\,\mathrm{Var}[\,\chi_q|\Theta,\Delta\,]\,] \;=\; \mathrm{Var}[\,\chi_q|\Theta\,] \;-\; \mathrm{Var}[\,\mathrm{E}[\,\chi_q|\Theta,\Delta\,]\,].$$

As $\mathrm{Var}[\,\cdot\,]\geq 0$, the (expected) variance associated with using the larger structure G^+ can be no larger than the variance of the smaller one G.

The same proof holds for conditional queries $q = P(\,C=c\,|\,\mathbf{E}=\mathbf{e}\,)$; here we need to keep the conditioning $\mathbf{E}=\mathbf{e}$ around for each step. ∎

Proof of Theorem 7: This proof is a reduction from Maximal Clique [GJ79, p194]. Given any graph $H = \langle\{Z_i\}_{i=1}^n, \mathcal{E}\rangle$, form a "inverted-naïve-bayes" structure over binary variables where each of these n Z_is is a parent to the node C; see Figure 1(b). The query here is $P(\,C=0\,)$. Now let S^{all} be a sample of $1+n+|\mathcal{E}|$ tuples: Here $C=0$ everywhere. The first tuple is all 0's; each of the next n tuples sets exactly one Z_i to 1, and each the final $|\mathcal{E}|$ tuples has exactly 2 values equal to 1, corresponding to an edge.

Notice S^{all} includes all 2^2 entries for the $\langle Z_i, Z_j\rangle$ pair if-and-only-if there is an edge joining these nodes in the graph H — while S^{all} includes 3 of the 4 entries for the every $\langle Z_i, Z_j\rangle$ pair (the $\langle 0,0\rangle$ assignment appears in the first tuple, the $\langle 1,0\rangle$ in the "all-0-but-Z_i" row, and the $\langle 0,1\rangle$ in the "all-0-but-Z_j" row), the final assignment $\langle 1,1\rangle$ is only present if there is an edge in the original H graph, as only then is there is a tuple (associated with this edge) that has this pair of 1's.

We can include some set of $\{Z_i\to C\}$ arcs in the final "diminished" subgraph (of the inverted-naïve-bayes structure) only if, for each Z_i, Z_j included, the S^{all} sample includes all 2^2 assignments; as shown above, this only happens if there is an arc connecting these two nodes in H. This means we can include a set of k arcs only if H includes a clique of size k, which is an NP-complete decision problem.

(Note that we are not actually writing down the CPtable here, for either the original set over n variables, nor for the reduced set over $k = O(n)$ variables, as that alone would require $2^{O(n)}$ time.) ∎