Are there Bayesian networks in which posterior inference is often difficult?

George Matheos, May 8, 2024

1 Background

1.1 Probabilistic graphical models and inference problems

Definition 1. A **probabilistic graphical model on binary variables** is a tuple (V, E, P), where V is an ordered, finite set of variables $V = \{v_1, \ldots, v_n\}$, E is a set of directed edges between the variables, and P is a *conditional probability table*. The directed graph (V, E) must be acyclic. For $v \in V$, $\mathrm{Pa}(v)$ denotes the set of parent variables of v: $\mathrm{Pa}(v) = \{u : (u \mapsto v) \in E\}$. Given any assignment $a_{\mathrm{Pa}(v)} \in \{0,1\}^{|\mathrm{Pa}(v)|}$ to the parent variables of v, the conditional probability table P stores value $P(v = \cdot; a_{\mathrm{Pa}(v)})$, which is a probability vector $[p_{v=0}, p_{v=1}]$ in $\mathbb{R}^2$.

A general probabilistic graphical model lifts the restriction that each variable v_i is binary, and allows it to have arbitrary finite domain. In this report, I will focus on binary probabilistic graphical models. Because a variable with a domain of size k can be represented using $\log k$ binary variables, all the results in this case carry to the general case, except those which restrict the sizes of sets of variables under consideration. Henceforth, the phrase "probabilistic graphical model" should be understood as a graphical model on binary variables.

Given a graphical model (P, E, V), we can define a joint distribution on all the variables in V, with probability mass function

$$P(a) = \prod_{v_i \in V} P(v_i = a_i; a_{\mathrm{Pa}(v_i)}) \quad \forall a \in \{0,1\}^{|V|}$$

where $a_{\mathrm{Pa}(v_i)}$ is the assignment to the parent variables of v_i in a. I will often write P to refer to the whole graphical model, the joint distribution on all its variables, and also marginal and conditional distributions on subsets of its variables.

Definition 2. An **inference problem** consists of a graphical model (V, E, P), a set of *observed variables* $Y \subseteq V$, a set of *query variables* $X \subseteq V$, and an assignment $y \in \{0,1\}^{|Y|}$ to the observed variables such that $P(Y = y) > 0$.

The goal of an inference problem is to compute some piece of information about the posterior distribution $P(X = \cdot | Y = y)$, which is a probability distribution on $\{0,1\}^{|X|}$.

Definition 3. An **inference problem schema** is the tuple $I = (V, E, P, X, Y)$ as in an inference problem, but not fixing an assignment y to the observed variables.

1.2 Worst and typical case inference algorithms

Definition 4. A **deterministic, worst-case additive PDF approximation algorithm with tolerance** ϵ is a Turing machine A which on input (I, y, x), where I is any inference problem schema, y is any assignment to the observed variables, and x is an assignment to the query variables, outputs a rational number $A(I, y, x)$ such that

$$|A(I, y, x) - P(X = x | Y = y)| < \epsilon$$

In 1993, Dagum and Luby [1] showed that if there exists a worst-case additive PDF approximation algorithm with tolerance $< 1/2$, and it has polynomial runtime, then $\mathbf{P} = \mathbf{NP}$.

Probabilistic graphical models are typically used to model aspects of the world. That is, each variable in V represents some aspect of the world; Y represents a set of values which we have observed; and X represents a set of values which we wish to infer. In this setting, the probability distribution P is a description of our beliefs about how probable different joint outcomes of events in the world are. Therefore, given a graphical model P in which we must do inference, it may be acceptable to us if there is exist assignments y under

which computing the posterior distribution is very expensive, so long as these instances are extremeley rare. Since we have a probability distribution P on hand which ought to roughly correspond to the distribution of y values which will occur in the world, and on which we will have to run inference, a natural notion of "rare" is available. Say there is a small set of observation assignments $\mathcal{Y}_{\text{hard}} \subseteq \{0,1\}^{|Y|}$ such that $P(\mathcal{Y}_{\text{hard}}) < \rho$ for very small ρ (e.g. $\rho = 0.00001$), such that for all $y \notin \mathcal{Y}_{\text{hard}}$, we can compute the posterior distribution $P(X = \cdot|Y = y)$ efficiently. Then we can say that inference is easy in the typical case, and for many purposes this is sufficient.

In fact, Dagum and Luby's construction of a worst-case inference problem I involves selecting an observed assignment y to a set Y of one variable ($|Y| = 1$) which has extremely low marginal probability: $P(Y = y) << 1$. Theorem **??** later in this report shows that for every inference problem schema (P, V, E, X, Y) with $|Y| = 1$, efficient inference is possible on typical case observations. This indicates that Dagum and Luby's strategy for proving the hardness of inference in the worst case does not directly carry through to showing the hardness of inference with typical-case observations.

Definition 5. A **deterministic, typical-case additive PDF approximation algorithm with tolerances** (ϵ, ρ) is a Turing machine A which accepts inputs of the form (I, y, x), where I is any inference problem schema, y is any assignment to the observed variables, and x is an assignment to the query variables, and outputs a rational number $A(I, y, x)$ with the following property. For any inference schema I and any value x,

$$\Pr_{y \sim P(Y=\cdot)}[|A(I, x, y) - P(X = x|Y = y)| \geq \epsilon] < \rho$$

Given such an algorithm A, a given inference schema I, and a value x, I will write $\mathcal{Y}_{\text{good}}$ to denote the set

$$\mathcal{Y}_{\text{good}} := \{y : |A(I, x, y) - P(X = x|Y = y)| < \epsilon\}$$

Note that $P(\mathcal{Y}_{\text{good}}) > 1 - \rho$.

1.3 One-way functions and pseudorandom generators

Proving the hardness of a computational problem C is often done by reducing from an **NP**-hard problem like 3SAT to problem C, thereby showing that if C could be solved in polynomial time, $\mathbf{P} = \mathbf{NP}$. However, problems like 3SAT are stated in terms of behavior on all, and thus worst-case, inputs. Therefore, to prove the hardness of a computational problem on typical-case inputs, it is preferable to reduce to a hardness conjecture stated directly in terms of typical-case behavior.

In this report I will reference two such conjectures, which are known to be related. The first conjecture is the existance of *one-way functions*, functions which can be efficiently computed, but not efficiently inverted for the majority of inputs. It is widely believed that such functions exist [], as there are a number of functions like multiplication of prime numbers, for which no inversion algorithms are known which are efficient in the typical case. Such functions are widely used in public-key cryptography.

Definition 6. (Arora and Barak Def. 9.4) A **one-way function** f is a function $f : \{0,1\}^* \to \{0,1\}^*$ such that f can be computed in polynomial time, and for every probabilistic polynomial time algorithm A,

$$\Pr_{x \sim \text{Uniform}(\{0,1\}^n)}[A(f(x), 1^n) \in f^{-1}(f(x))] \underset{n \to \infty}{\to} 0$$

where $f^{-1}(f(x))$ is the set $\{x' \in \{0,1\}^* : f(x') = f(x)\}$.

The second conjecture is the existance of *secure pseudorandom generators*, which are functions which take a short random seed and expand it into a long string which is indistinguishable from a truly random string. Specifically, I consider PRGs which are secure against all polynomial-time adversaries, once the strings being generated are sufficiently long.

Definition 7. (Arora and Barak Def. 9.8) A *secure pseudorandom generator* (PRG) is a polynomial time computable function $\{0,1\}^* \to \{0,1\}^*$ such that $|G(x)| = |l(|x|)|$ for some function $l : \mathbb{N} \to \mathbb{N}$, such that for every probabilistic polynomial time algorithm A,

$$|\Pr_{s\sim\text{Uniform}(\{0,1\}^n)}[A(G(s)) = 1] - \Pr_{y\sim\text{Uniform}(\{0,1\}^{l(n)})}[A(y) = 1]| \underset{n\to\infty}{\to} 0$$

The function l is called the stretch of the PRG.

It is known that the existence of one-way functions implies the existence of secure pseudorandom generators. Thus, to the extent that it is believed that one-way functions exist, it is also believed that secure pseudorandom generators exist.

Proposition 1. (Arora and Barak Thm. 9.9) If there exists a one-way function, then there exists a secure pseudorandom generator with stretch $l(n) = n^c$.

2 Main result

Theorem 2. If there exists a polnomial-time, deterministic, typical-case additive PMF approximation algorithm with tolerances $\epsilon < \frac{1}{2}$, $\rho < \frac{1}{4}$, then there does not exist a secure pseudorandom generator with stretch l s.t. $l(n) - n \underset{n\to\infty}{\to} \infty$.

Corollary 3. If there exists a polnomial-time, deterministic, typical-case additive PMF approximation algorithm with tolerances $\epsilon < \frac{1}{2}$, $\rho < \frac{1}{4}$, then one-way functions do not exist.

The corollary follows because if one-way functions exist, there exists a secure PRG with stretch $l(n) = n^2$.

Proof. (Proof of Theorem 2.)

Proof setup. For contradiction, suppose A is a polynomial-time typical-case additive PMF approximation algorithm with tolerances $\epsilon < \frac{1}{2}$ and $\rho < \frac{1}{4}$. Let $\delta > 0$ be s.t. $\epsilon < \frac{1}{2} - \delta$. Suppose G is a secure PRG with stretch $l(n)$ s.t. $\lim_{n\to\infty}[l(n) - n] = \infty$. By lemma 4, I will assume without loss of generality that l is a bijection and that the l^{-1} can be computed in polynomial time.

Proof outline. I will construct a Turing machine B which can distinguish the distributions $\text{Uniform}(\{0,1\}^n)\circ G^{-1}$ and $\text{Uniform}(\{0,1\}^{l(n)})$ for all $n > N$, where N is a constant. This contradicts that G is a secure PRG.

Given input $y \in \{0,1\}^{l(n)}$, Turing machine B will guess whether it was sampled from the pseudo-random generator, or from the uniform on $\{0,1\}^{|y|}$. If a string y is given such that $|y|$ is not in the range of l, then B can immediately reject. Otherwise, B will compute $n = l^{-1}(|y|)$ to find the seed length needed for G to produce y.

Second, B will construct the description of an inference schema $I = (P, V, E, X, Y)$ where $|Y| = l(n)$ and $|X| = 1$, such that if $n > N$,

$$y' \in G(\{0,1\}^n) \implies P(X = 1|Y = y') > 1 - \delta \quad (*)$$

and

$$y' \notin G(\{0,1\}^n) \implies P(X = 1|Y = y') = 0 \quad (**)$$

Third, B will compute $a \leftarrow A(I, 1, y)$. If $a < 1/2$, B will output 0; otherwise B will output 1. I will first show that this implies that on typical y values, B exactly decides whether $y \in \text{range}(G)$. Specifically, there is a set $\mathcal{Y}_{\text{good}} \subseteq \{0,1\}^{l(n)}$ with $P(\mathcal{Y}_{\text{good}}) > 3/4$ such that $y \in \mathcal{Y}_{\text{good}} \implies B(y) = 1_{y\in\text{range}(G)}$. Using this, I will then show that, because we can guarantee $P(y \in \mathcal{Y}_{\text{good}}) > 3/4 = 1 - \rho$, this implies that there exists a $\gamma > 0$ such that

$$|\Pr_{s\sim\text{Uniform}(\{0,1\}^n)}[B(G(s)) = 1] - \Pr_{\bar{y}\sim\text{Uniform}(\{0,1\}^{l(n)})}[B(\bar{y}) = 1]| \geq \gamma \quad (***)$$

for all sufficiently large n, contradicting that G is a secure pseudorandom generator.

I now proceed to complete the proof by (1) proving that B can construct an inference schema with properties $(*)$ and $(**)$, and (2) proving that $(***)$ holds.

Construction of the inference schema. Let (P, V, E) be the probabilistic graphical model which implements the following process. P samples $s = s_1 s_2 \dots s_n$ uniformly at random from $\{0,1\}^n$. P samples $\bar{y} = \bar{y}_1 \bar{y}_2 \dots \bar{y}_{l(n)}$ uniformly at random from $\{0,1\}^{l(n)}$. P samples x uniformly at random from $\{0,1\}$. If $x = 0$, P computes $y = \bar{y}$. Otherwise, P computes $y = G(s)$. Clearly, writing a graphical model which does the sampling and multiplexing steps can be done in polynomial time. And writing out the part of the graphical model which computes $G(s)$ can be done in polynomial time in n using the Cook-Levin reduction for writing a circuit which implements the same behavior as the Turing machine for G, on inputs of length n.

Analysis of the posteriors (proof of $(*)$ and $(**)$). First, note that if $y \notin \text{range}(G)$,

$$P(X = 1|Y = y) = \frac{P(X = 1, Y = y)}{P(Y = y)} = \frac{0}{P(Y = y)} = 0$$

because it is impossible for P to sample $x = 1$ yet output a value not in the range of G. This proves $(**)$.

Now, say $y \in \text{range}(G)$. Let $m = |G^{-1}(y)|$, the number of seeds in $\{0,1\}^n$ that get mapped to y. Then

$$P(X = 1|Y = y) = \frac{P(X = 1, Y = y)}{P(X = 0, Y = y) + P(X = 1, Y = y)} = \frac{\frac{1}{2}\frac{m}{2^n}}{\frac{1}{2}\frac{1}{2^{l(n)}} + \frac{1}{2}\frac{m}{2^n}}$$

The numerator here is $P(X = 1, Y = y) = \frac{1}{2}\frac{m}{2^n}$ because $X = 1$ with probability $\frac{1}{2}$, and given that $X = 1$, we will have $Y = y$ iff the seed s is in $G^{-1}(y)$, which occurs with probability $\frac{m}{2^n}$. The denominator contains the term $P(X = 0, Y = y) = \frac{1}{2}\frac{1}{2^{l(n)}}$ because there is a $1/2$ probability that $X = 0$, and if it is 0, $Y = y$ only if one of the $2^{l(n)}$ equally probable values in $\{0,1\}^{l(n)}$ is chosen as $\bar{y}$.

Simplifying this,

$$P(X = 1|Y = y) = \frac{m/2^n}{1/2^{l(n)} + m/2^n}$$

so

$$P(X = 0|Y = y) = \frac{1/2^{l(n)}}{1/2^{l(n)} + m/2^n} = \frac{1}{1 + m2^{l(n)-n}}$$

Since $2^{l(n)-n} > 0$, this expression decreases in m. Since $y \in \text{range}(G)$, $m \geq 1$. Thus this expression is maximized when $m = 1$. Therefore, for all m,

$$P(X = 0|Y = y) \leq \frac{1}{1 + 2^{l(n)-n}} < \frac{1}{2^{l(n)-n}}$$

Let N_1 be an integer such that $n > N_1 \implies l(n) - n > \log(1/\delta)$. Then $n > N_1 \implies 2^{l(n)-n} > 1/\delta$, so for all such n,

$$P(X = 0|Y = y) < \delta$$

and thus

$$P(X = 1|Y = y) > 1 - \delta$$

This proves $(*)$.

Proof that on $\mathcal{Y}_{\text{good}}$, B exactly identifies range(G). Let $\mathcal{Y}_{\text{good}}$ be the good set of observations for inference schema I constructed above, and assignment $x = 1$, as defined in Definition 5. Then if the given y satisfies $y \in \mathcal{Y}_{\text{good}}$, $|A(I, x, y) - P(X = x|Y = y)| < \epsilon < \frac{1}{2} - \delta$. Thus if $y \notin \text{range}(G)$, by $(**)$, $|A(I, x, y) - 0| < \frac{1}{2} - \delta$ so $A(I, x, y) < 1/2$. And if $y \in \text{range}(G)$ but $A(I, x, y) < 1/2$, by $(*)$ we would have

$|P(X=1|Y=y) - A(I,x,y)| \geq |(1-\delta) - 1/2| \geq 1/2 - \delta > \epsilon$, so it must be the case that $A(I,x,y) > 1/2$. That is,

$$y \in \mathcal{Y}_{\text{good}} \implies [y \notin \text{range}(G) \implies A(I,x,y) < 1/2 \wedge y \in \text{range}(G) \implies A(I,x,y) > 1/2]$$

Thus for $y \in \mathcal{Y}_{\text{good}}$, algorithm B exaclty decides whether $y \in \text{range}(G)$: $y \in \mathcal{Y}_{\text{good}} \implies B(y) = 1_{y \in \text{range}(G)}$.

Proof that B identifies outputs from the PRG with nontrivial probability (proof of $(***)$). In this section I will write $\Pr_s$ as shorthand for $\Pr_{s \sim \text{Uniform}(\{0,1\}^n)}$ and $\Pr_{\bar{y}}$ as shorthand for $\Pr_{\bar{y} \sim \text{Uniform}(\{0,1\}^{l(n)})}$.

The goal of this section is to establish that there exists a $\gamma > 0$ such that

$$|\Pr_s[B(G(s)) = 1] - \Pr_{\bar{y}}[B(\bar{y}) = 1]| \geq \gamma$$

Let $p_s := \Pr_s[B(G(s)) = 1]$ and $p_{\bar{y}} := \Pr_{\bar{y}}[B(\bar{y}) = 1]$. Let $p_{s,1} := \Pr_s[B(G(s)) = 1 \wedge G(s) \in \mathcal{Y}_{\text{good}}]$, $p_{s,2} := \Pr_s[B(G(s)) = 1 \wedge G(s) \notin \mathcal{Y}_{\text{good}}]$, $p_{\bar{y},1} = \Pr_{\bar{y}}[B(\bar{y}) = 1 \wedge \bar{y} \in \mathcal{Y}_{\text{good}}]$, and $p_{\bar{y},2} = \Pr_{\bar{y}}[B(\bar{y}) = 1 \wedge \bar{y} \notin \mathcal{Y}_{\text{good}}]$. Observe that $p_s = p_{s,1} + p_{s,2}$ and $p_{\bar{y}} = p_{\bar{y},1} + p_{\bar{y},2}$. By the triangle inequaltiy,

$$|p_s - p_{\bar{y}}| = |p_{s,1} + p_{s,2} - p_{\bar{y},1} - p_{\bar{y},2}| \geq |p_{s,1} - p_{\bar{y},1}| - |p_{s,2} - p_{\bar{y},2}| \tag{1}$$

We have

$$\begin{aligned} |p_{s,2} - p_{\bar{y},2}| &= |\Pr_s[G(s) \notin \mathcal{Y}_{\text{good}}] \Pr_s[B(G(s)) = 1 | G(s) \notin \mathcal{Y}_{\text{good}}] - \Pr_{\bar{y}}[\bar{y} \notin \mathcal{Y}_{\text{good}}] \Pr_{\bar{y}}[B(\bar{y}) = 1 | \bar{y} \notin \mathcal{Y}_{\text{good}}]| \\ &\leq \max(\Pr_s[G(s) \notin \mathcal{Y}_{\text{good}}], \Pr_{\bar{y}}[\bar{y} \notin \mathcal{Y}_{\text{good}}]) \end{aligned} \tag{2}$$

We also have

$$\begin{aligned} p_{s,1} - p_{\bar{y},1} &= \Pr_s[B(G(s)) = 1 \wedge G(s) \in \mathcal{Y}_{\text{good}}] - \Pr_{\bar{y}}[B(\bar{y}) = 1 \wedge \bar{y} \in \mathcal{Y}_{\text{good}}] \\ &= \Pr_s[G(s) \in \text{range}(G) \wedge G(s) \in \mathcal{Y}_{\text{good}}] - \Pr_{\bar{y}}[\bar{y} \in \text{range}(G) \wedge \bar{y} \in \mathcal{Y}_{\text{good}}] \\ &= \Pr_s[G(s) \in \mathcal{Y}_{\text{good}}] - \Pr_{\bar{y}}[\bar{y} \in \text{range}(G) \wedge \bar{y} \in \mathcal{Y}_{\text{good}}] \\ &\geq \Pr_s[G(s) \in \mathcal{Y}_{\text{good}}] - \Pr_{\bar{y}}[\bar{y} \in \text{range}(G)] = \Pr_s[G(s) \in \mathcal{Y}_{\text{good}}] - 2^n/2^{l(n)} \end{aligned} \tag{3}$$

The second equality here follows from the fact established above, that for $y \in \mathcal{Y}_{\text{good}}$, algorithm B exactly decides whether $y \in \text{range}(G)$.

We now need to bound $\Pr_s[G(s) \in \mathcal{Y}_{\text{good}}]$ and $\Pr_{\bar{y}}[\bar{y} \in \mathcal{Y}_{\text{good}}]$. Observe that

$$P(\mathcal{Y}_{\text{good}}) = \frac{1}{2} \Pr_s[G(s) \in \mathcal{Y}_{\text{good}}] + \frac{1}{2} \Pr_{\bar{y}}[\bar{y} \in \mathcal{Y}_{\text{good}}]$$

because under the model P, y is set equal to $G(s)$ half the time and equal to $\bar{y}$ the other half of the time. Since each of these probability terms are no greater than 1, we have

$$\Pr_s[G(s) \in \mathcal{Y}_{\text{good}}] \geq 2P(\mathcal{Y}_{\text{good}}) - 1; \quad \Pr_{\bar{y}}[\bar{y} \in \mathcal{Y}_{\text{good}}] \geq 2P(\mathcal{Y}_{\text{good}}) - 1 \tag{4}$$

By subtracting each side of these inequalities from 1, we also obtain

$$\Pr_s[G(s) \notin \mathcal{Y}_{\text{good}}] \leq 2 - 2P(\mathcal{Y}_{\text{good}}); \quad \Pr_{\bar{y}}[\bar{y} \notin \mathcal{Y}_{\text{good}}] \leq 2 - 2P(\mathcal{Y}_{\text{good}}) \tag{5}$$

Combining equations 2 and 5, we get

$$|p_{s,2} - p_{\bar{y},2}| \leq 2 - 2P(\mathcal{Y}_{\text{good}})$$

and combining equations 3 and 4, we get

$$|p_{s,1} - p_{\bar{y},1}| \geq 2P(\mathcal{Y}_{\text{good}}) - 1 - 2^{n-l(n)}$$

Plugging in these bounds to equation 1, we obtain

$$|p_s - p_{\bar{y}}| \geq 2P(\mathcal{Y}_{\text{good}}) - 1 - 2^{n-l(n)} - (2 - 2P(\mathcal{Y}_{\text{good}})) = 4P(\mathcal{Y}_{\text{good}}) - 3 - 2^{n-l(n)} \tag{6}$$

Thus,

$$P(\mathcal{Y}_{\text{good}}) \geq \frac{3}{4} + \frac{1}{4 \cdot 2^{l(n)-n}} + \frac{1}{4}\gamma \implies |p_s - p_{\bar{y}}| \geq \gamma$$

Since $l(n) - n \to \infty$, the term $\frac{1}{4 \cdot 2^{l(n)-n}} \to 0$, so there exists N_2 such that for all $n > N_2$,

$$P(\mathcal{Y}_{\text{good}}) > \frac{3}{4} \implies \exists \gamma > 0 \text{ s.t. } |\Pr_s[B(G(s)) = 1] - \Pr_{\bar{y}}[B(\bar{y}) = 1]| \geq \gamma$$

Finally, it suffices to take $N = \max(N_1, N_2)$.

□

Finally, here is a proof of the minor lemma used at the beginning of the proof of Theorem 2.

Lemma 4. If there exists a secure PRG G with stretch $l(n)$ s.t. $\lim_{n\to\infty}[l(n) - n] = \infty$, then there exists a PRG G' with stretch l' satisfying the same property, and such that l' is a bijection such that $(l')^{-1}$ can be computed in polynomial time.

Proof. **TODO.** □