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

George Matheos, May 9, 2024

1 Posterior inference and computation

1.1 Posterior inference

A central computational problem in science and artificial intelligence is posterior inference in probabilistic models. Consider a probability distribution P on a space $X \times Y \times Z$. For the purposes of this paper, we will take $X, Y, Z \subseteq \{0,1\}^*$. The problem of posterior inference is to characterize the posterior distribution

$$P(X = \cdot | Y = y)$$

for a value $y \in Y$ s.t. $P(Y = y) > 0$.

To define this formally, we must first define the marginal distributions

$$P(X = x, Y = y) := \sum_{z \in Z} P(x, y, z); \quad P(Y = y) := \sum_{(x,z) \in X \times Z} P(x, y, z)$$

With these definitions in hand, the posterior $P(X = \cdot | Y = y)$ is the probability distribution such that for all $x \in X$,

$$P(X = x | Y = y) := \frac{P(X = x, Y = y)}{P(Y = y)}$$

Stated informally, the problem of posterior inference is: *given a probabilistic model P and a value y with $P(Y = y) > 0$, characerize $P(X = x|Y = y)$.*

1.2 Computational formulations of exact inference

To formalize the problem of posterior inference as a specification a computer algorithm could satisfy, we must specify what it would mean for an algorithm to "characterize" a probability distribution on the space X, as this is the type of mathematical object that $P(X = x|Y = y)$ is. There are several distinct algorithmic formulations:

1. **Probability mass function (PMF) computation.** Given a value $x \in X, y \in Y$, compute the number $P(X = x|Y = y)$.

2. **Cumulative distribution function (CDF) computation.** Given a value $x \in X, y \in Y$, and letting strings in $x \in X \subseteq \{0,1\}^*$ have lexographic ordering, compute the number $P(X \leq x|Y = y)$.

3. **Sampling.** Given a value $y \in Y$, sample a random value in X according to the distribution $P(X = x|Y = y)$. This requires the use of a probabilistic Turing machine. Under standard formulations, this means we want a Turing machine A which, given $y \in Y$ and an uniformly random bit sequence $r \in \{0,1\}^*$, computes a value $A(y, r) \in X$ such that for all $x \in X$, $P(X = x|Y = y) = \Pr_r[x = A(y, r)]$.

1.3 Computational formulations of approximate inference

It usually suffices for algorithms to perform approximate probabilistic inference. There are several standard formulations of approximate probabilistic inference as computational problems:

1. **PMF evalution up to additive error.** Fix $\epsilon > 0$. Algorithm A does PMF evaluation in P up to additive error ϵ if for any $x \in X, y \in Y$, it computes a number $A(x, y)$ where $|A(x, y) - P(X = x|Y = y)| < \epsilon$.

2. **PMF evaluation up to relative error.** Fix $\epsilon > 0$. Algorithm A does PMF evaluation in P up to relative error ϵ if for any $x \in X, y \in Y$, it computes a number $A(x,y) \in (\frac{1}{1+\epsilon}P(X = x|Y = y), (1+\epsilon)P(X = x|Y = y))$

3. **CDF evaluation up to additive error.** Fix $\epsilon > 0$. Algorithm A does CDF evaluation in P up to additive error ϵ if for any $x \in X, y \in Y$, it computes a number $A(x,y)$ where $|A(x,y) - P(X \leq x|Y = y)| < \epsilon$.

4. **CDF evaluation up to relative error.** Fix $\epsilon > 0$. Algorithm A does CDF evaluation in P up to relative error ϵ if for any $x \in X, y \in Y$, it computes a number $A(x,y) \in (\frac{1}{1+\epsilon}P(X \leq x|Y = y), (1+\epsilon)P(X \leq x|Y = y))$.

5. **Approximate sampling with pointwise probability bounds.** There are several formulations of approximate sampling. One formulation is to say that probabilistic algorithm $A(y,r)$ (for random bitstring r) approximately samples from $P(X = x|Y = y)$ up to pointwise error δ if for all $x \in X$, $|\Pr_r[A(y,r) = x] - P(X = x|Y = y)| < \delta$.

For any of the PMF or CDF computation problems, a probabilistic version can be defined in which a probabilistic algorithm A is said to do PMF or CDF computation iff it satisfies the above specification with probability $\geq 2/3$ on any input.

1.4 Some hardness relationships among these formulations

A number of relationships are known between the hardness of inference according to these different formulations. Below, I will articulate a subset of these which indicate that if it is computationally hard to do probabilistic PMF computation up to additive error, then it is computationally difficult to solve any of the aformentinoed inference problems.

1. **Relative approximation for is harder than additive approximation for equal ϵ.** Consider a true PMF or CDF value p. If $\epsilon \geq 1$ then additive PMF or CDF approximation is trivial, so consider an $\epsilon < 1$. Since $p \in [0,1]$, if $\hat{p}$ satisfies $\hat{p} \in (p/(1+\epsilon), (1+\epsilon)p)$, then

   $$|\hat{p} - p| \leq \max(|(1+\epsilon)p - p|, |p/(1+\epsilon) - p|) \leq \max(|1+\epsilon-1|, |1/(1+\epsilon) - 1|) = \epsilon$$

   Thus if it is possible to compute a relative ϵ approximation to either CDF or PMF, it is also possible to compute an additive ϵ approximation to the same quantity.

2. **CDF computation is harder than PMF computation.**

   (a) *Exact computation.* Say $\mathsf{CDF}(x,y) := P(X \leq x|Y = y)$ can be computed exactly. Then we can compute $\mathsf{PMF}(x,y) := P(X = x|Y = y) = \mathsf{CDF}(x,y) - \mathsf{CDF}(x-1,y)$.

   (b) *Additive approximation.* Say we can compute an additive ϵ approximation $\mathsf{CDF}_\epsilon(x,y)$ to $P(X \leq x|Y = y)$. Then $\mathsf{PMF}_{2\epsilon}(x,y) := \mathsf{CDF}_\epsilon(x,y) - \mathsf{CDF}_\epsilon(x-1,y)$ is a 2ϵ additive approximation to $P(X = x|Y = y)$.

3. **Sampling is harder than probabilistic PMF computation up to additive error.** Say that it is possible to sample with pointwise error $\epsilon/2$ from $P(X = \cdot|Y = y)$. Let Q be the exact distribution which can be sampled from, which has pointwise error $\leq \epsilon/2$ from the posterior. Consider an algorithm which generates N independent samples $x_i \sim$ and computes $\hat{p} := \frac{1}{N}\sum_{i=1}^{N} 1_{x_i = x}$. By the Hoeffding theorem, since $\mathbb{E}[\hat{p}] = Q(x)$, $\Pr[|\hat{p} - Q(x)| \geq \epsilon/2] \leq 2e^{-\epsilon^2/(2N)}$. Thus if $N = 100/\epsilon^2$, $\Pr[|\hat{p} - Q(x)| \geq \epsilon/2] << 1/3$. Because $|Q(x) = P(X = x|Y = y)| < \epsilon/2$, this means that $\Pr[|\hat{p} - P(X = x|Y = y)| > \epsilon] << 1/3$. Thus it is possible to compute an ϵ approximation to PMF at a cost $O(1/\epsilon^2)$ times the cost of performing approximate pointwise sampling. Therefore a polynomial time approximate sampling algorithm implies the existance of a polynomial time additive PMF approximation algorithm.

2 Background for the main results

2.1 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 $V = \{v_1, \dots, v_n\}$, 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 $v \in V$, $\text{Pa}(v)$ denotes the set of parent variables of v: $\text{Pa}(v) = \{u : (u \mapsto v) \in E\}$. Given any assignment $a_{\text{Pa}(v)} \in \{0,1\}^{|\text{Pa}(v)|}$ to the parent variables of v, the conditional probability table P stores value $P(v = \cdot; a_{\text{Pa}(v)})$, 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

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

where $a_{\text{Pa}(v_i)}$ 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.

2.2 Worst and typical case inference algorithms

Definition 4. A **deterministic, worst-case additive PMF 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

$$|A(I, y, x) - P(X = x | Y = y)| < \epsilon \tag{1}$$

Definition 5. A **probabilistic, worst-case additive PMF approximation algorithm with tolerance** ϵ is a probabilistic Turing machine A which for every I, y, x, satisfies condition 1 with probability $\geq 2/3$.

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

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.

Definition 6. A **deterministic, typical-case additive PMF 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 $P(\mathcal{Y}_{\text{good}}) > 1 - \rho$.

Definition 7. A **probabilistic, typical-case additive PMF approximation algorithm with tolerances** (ϵ, ρ) is a probabilistic Turing machine A which accepts inputs (I, y, x) as above, such that for any inference schema I and any value x, there exists a set $\mathcal{Y}_{\text{good}} \subseteq \{0,1\}^{|Y|}$ where $P(\mathcal{Y}_{\text{good}}) > 1 - \rho$ and

$$y \in \mathcal{Y}_{\text{good}} \implies \Pr[|A(I,x,y) - P(X=x|Y=y)| < \epsilon] > 2/3$$

where the probability is taken over the randomness of the algorithm.

In this report, I prove a new result showing that if either a deterministic or probabilistic typical-case additive PMF approximation algorithm exists with tolerances $\epsilon < 1/2$, $\rho < 1/4$, then one-way functions do not exist (Theorem 2). This hardness result suggests that typical-case inference is computationally difficult, and it is not merely the worst-case nature of the observations considered in Dagum and Luby's reduction which made inference difficult in their setting. Due to the typical-case formulation studied here, a different proof strategy is needed to show this hardness result.

2.3 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 8. (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, for every $c \in \mathbb{R}$,

$$n^c \cdot \Pr_{x \sim \text{Uniform}(\{0,1\}^n)}[A(f(x), 1^n) \in f^{-1}(f(x))] \underset{n \to \infty}{\to} 0 \tag{2}$$

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. This is known to follow from the existence of one-way functions, and this weaker conjecture is all the is needed for the main result of this report. Specifically, I consider PRGs which are secure against all polynomial-time adversaries, once the strings being generated are sufficiently long.

Definition 9. (Weakened variant of 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,

$$|\Pr_{s \sim \text{Uniform}(\{0,1\}^n)}[A(G(s)) = 1] - \Pr_{y \sim \text{Uniform}(\{0,1\}^{l(n)})}[A(y) = 1]| \underset{n\to\infty}{\to} 0 \tag{3}$$

The function l is called the stretch of the PRG.

This definition is weaker than the definition given in Arora and Barak, as it only requires the expression in eq. 3 to tend to zero, rather than to zero at a superpolynomial rate, as in eq. 2, and as in Arora and Barak's definition of a secure PRG. The reduction in this report shows that the existence of a secure PRG, as defined here, with superlinear stretch implies the non-existence of typical-case inference algorithms with certain tolerances. The existance of secure PRGs as defined in Arora and Barak is a strictly stronger statement than the existance of secure PRGs as defined here, and their PRGs are instances of secure PRGs as defined here.

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 $l(n) = n^c$.

The reason I include the superpolynomial convergence rate in Def. 8 is to ensure that this theorem goes through. As far as I am aware, it may be that given my weaker definition of a secure PRG, a weaker definition of one-way function would also suffice to imply the existance of such PRGs, but I have not verified this.

3 Main result

Theorem 2. If there exists a polynomial-time typical-case additive PMF approximation algorithm A with tolerances $\epsilon < \frac{1}{2}$, $\rho < \frac{1}{4}$, then there does not exist a secure pseudorandom generator with an invertible stretch function l such that $\lim_{n\to\infty}[l(n) - n] = \infty$. This holds whether A is deterministic or probabilistic.

This holds even if we only require A to be capable of typical-case inference in Bayesian networks with in-degree ≤ 2 and with probability values in the set $\{0, 1/2, 1\}$.

Corollary 3. If there exists a polynomial-time 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, by Proposition 1, there exists a secure PRG with stretch $l(n) = n^2$.

These results suggest that it is unlikely that there exists a universal typical-case inference algorithm which succeeds on a 3/4 typical set of observations in all Bayesian networks.

Proof. (Theorem 2.) For contradiction, suppose A is a deterministic polynomial-time typical-case additive PMF approximation algorithm with tolerances $\epsilon < \frac{1}{2}$ and $\rho < \frac{1}{4}$. (The case where A is probabilistic will be covered at the end of the proof.) Let $\delta > 0$ be s.t. $\epsilon < \frac{1}{2} - \delta$. Suppose G is a secure PRG with invertible stretch function $l(n)$ s.t. $\lim_{n\to\infty}[l(n) - n] = \infty$. Note that l^{-1} is computable in poly(n) time, because given m, one can iterate through the values $n = 1, 2, \ldots, m$ to find when $l(n) = m$.

Proof outline. I will construct a Turing machine B which can distinguish the distributions Uniform($\{0,1\}^n$)$\circ G^{-1}$ and 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 6. 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 exactly 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,

$$|p_s - p_{\bar{y}}| = |p_{s,1} + p_{s,2} - p_{\bar{y},1} - p_{\bar{y},2}| \geq |p_{s,1} - p_{\bar{y},1}| - |p_{s,2} - p_{\bar{y},2}| \tag{4}$$

We have

$$\begin{aligned} |p_{s,2} - p_{\bar{y},2}| = |\Pr_s[G(s) \notin \mathcal{Y}_{\text{good}}] \Pr_s[B(G(s)) = 1 | G(s) \notin \mathcal{Y}_{\text{good}}] - \Pr_{\bar{y}}[\bar{y} \notin \mathcal{Y}_{\text{good}}] \Pr_{\bar{y}}[B(\bar{y}) = 1 | \bar{y} \notin \mathcal{Y}_{\text{good}}]| \\ \leq \max(\Pr_s[G(s) \notin \mathcal{Y}_{\text{good}}], \Pr_{\bar{y}}[\bar{y} \notin \mathcal{Y}_{\text{good}}]) \end{aligned} \tag{5}$$

We also have

$$\begin{aligned} p_{s,1} - p_{\bar{y},1} &= \Pr_s[B(G(s)) = 1 \wedge G(s) \in \mathcal{Y}_{\text{good}}] - \Pr_{\bar{y}}[B(\bar{y}) = 1 \wedge \bar{y} \in \mathcal{Y}_{\text{good}}] \\ &= \Pr_s[G(s) \in \text{range}(G) \wedge G(s) \in \mathcal{Y}_{\text{good}}] - \Pr_{\bar{y}}[\bar{y} \in \text{range}(G) \wedge \bar{y} \in \mathcal{Y}_{\text{good}}] \\ &= \Pr_s[G(s) \in \mathcal{Y}_{\text{good}}] - \Pr_{\bar{y}}[\bar{y} \in \text{range}(G) \wedge \bar{y} \in \mathcal{Y}_{\text{good}}] \\ &\geq \Pr_s[G(s) \in \mathcal{Y}_{\text{good}}] - \Pr_{\bar{y}}[\bar{y} \in \text{range}(G)] = \Pr_s[G(s) \in \mathcal{Y}_{\text{good}}] - 2^n/2^{l(n)} \end{aligned} \tag{6}$$

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

$$\Pr_s[G(s) \in \mathcal{Y}_{\text{good}}] \geq 2P(\mathcal{Y}_{\text{good}}) - 1; \quad \Pr_{\bar{y}}[\bar{y} \in \mathcal{Y}_{\text{good}}] \geq 2P(\mathcal{Y}_{\text{good}}) - 1 \tag{7}$$

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

$$\Pr_s[G(s) \notin \mathcal{Y}_{\text{good}}] \leq 2 - 2P(\mathcal{Y}_{\text{good}}); \quad \Pr_{\bar{y}}[\bar{y} \notin \mathcal{Y}_{\text{good}}] \leq 2 - 2P(\mathcal{Y}_{\text{good}}) \tag{8}$$

Combining equations 5 and 8, we get

$$|p_{s,2} - p_{\bar{y},2}| \leq 2 - 2P(\mathcal{Y}_{\text{good}}) \tag{9}$$

and combining equations 6 and 7, we get

$$|p_{s,1} - p_{\bar{y},1}| \geq 2P(\mathcal{Y}_{\text{good}}) - 1 - 2^{n-l(n)} \tag{10}$$

Plugging in these bounds to equation 4, we obtain

$$|p_s - p_{\bar{y}}| \geq 2P(\mathcal{Y}_{\text{good}}) - 1 - 2^{n-l(n)} - (2 - 2P(\mathcal{Y}_{\text{good}})) = 4P(\mathcal{Y}_{\text{good}}) - 3 - 2^{n-l(n)} \tag{11}$$

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 $N = \max(N_1, N_2)$.

Extending the proof to the case where the inference algorithm A is probabilistic. This extension is fairly straightforward; we simply need to track several additional terms in the analysis, to control the probability that algorithm B fails to indicate whether $y \in \text{range}(G)$ for a value $y \in \mathcal{Y}_{\text{good}}$. Since we can upper-bound this probability arbitrarily tightly by boosting the probabilistic version of algorithm B through replication, the ultimate conclusion is the same as before.

Say that the inference algorithm $A(I, x, y)$ is probabilistic. Write $A(I, x, y, r)$ to denote a run of algorithm A that uses internal random bits r. By saying A is a probabilistic typical-case additive-error PMF approximation algorithm, we mean that for $y \in \mathcal{Y}_{\text{good}}$, we have

$$\Pr_r[|A(I, x, y, r) - P(X = x|Y = y)| \geq \epsilon] < 1/3$$

In the procedure described above, the only place where algorithm B uses algorithm A is to check if $P(X = 1|Y = y)$ is greater than 1/2. I will write $B(y, r)$ for a call to B that uses random bits r when calling A. Our preceding analysis shows that for $y \in \mathcal{Y}_{\text{good}}$,

$$|A(I, 1, y, r) - P(X = 1|Y = y)| < \epsilon \implies B(y, r) = 1_{y \in \text{range}(G)}$$

Thus for $y \in \mathcal{Y}_{\text{good}}$,

$$\Pr_r[B(y,1) \neq 1_{y \in \text{range}(G)}] < 1/3$$

By calling B $O(k)$ times with different random r, and taking the most frequent output, we can obtain an algorithm B^*, comsuming randomness r^*, where for $y \in \mathcal{Y}_{\text{good}}$,

$$\Pr_{r^*}[B^*(y,r^*) \neq 1_{y \in \text{range}(G)}] < 1/2^k$$

Except for the final section dedicated to proving statement $(***)$, the preceding proof remains essentially unchanged. We must, however, redo part of the analysis in the final section, replacing the old deterministic algorithm B with our new stochastic algorithm B^*. Much of the analysis remains unchanged, except that the expressions $\Pr_s$ and $\Pr_{\bar{y}}$ should be understood to also close over the randomness r^*. The first place where the analysis diverges nontrivially is in equation 6.

To do the new version of the analysis, I will introduce the following notation. For $y \in \mathcal{Y}_{\text{good}}$, let $\mathcal{R}^y_{\text{good}}$ be the set of random strings r^* such that $B^*(y,r^*) = 1_{y \in \text{range}(G)}$, and for $y \notin \mathcal{Y}_{\text{good}}$ let $\mathcal{R}^y_{\text{good}}$ be the set of all bitstrings r^*.

Using this, in place of Equation 6, we now have

$$\begin{aligned}
p_{s,1} - p_{\bar{y},1} &= \Pr_{s,r^*}[B(G(s)) = 1 \wedge G(s) \in \mathcal{Y}_{\text{good}}] - \Pr_{\bar{y},r^*}[B(\bar{y}) = 1 \wedge \bar{y} \in \mathcal{Y}_{\text{good}}] \\
&= \Pr_{s,r^*}[G(s) \in \text{range}(G) \wedge G(s) \in \mathcal{Y}_{\text{good}} \wedge r^* \in \mathcal{R}^{G(s)}_{\text{good}}] + \Pr_{s,r^*}[B(G(s)) = 1 \wedge G(s) \in \mathcal{Y}_{\text{good}} \wedge r^* \notin \mathcal{R}^{G(s)}_{\text{good}}] \\
&\quad - (\Pr_{\bar{y},r^*}[\bar{y} \in \text{range}(G) \wedge \bar{y} \in \mathcal{Y}_{\text{good}} \wedge r^* \in \mathcal{R}^{\bar{y}}_{\text{good}}] + \Pr_{\bar{y},r^*}[B(\bar{y}) = 1 \wedge \bar{y} \in \mathcal{Y}_{\text{good}} \wedge r^* \notin \mathcal{R}^{\bar{y}}_{\text{good}}]) \\
&\geq \Pr_{s,r^*}[G(s) \in \text{range}(G) \wedge G(s) \in \mathcal{Y}_{\text{good}} \wedge r^* \in \mathcal{R}^{G(s)}_{\text{good}}] \\
&\quad - (\Pr_{\bar{y},r^*}[\bar{y} \in \text{range}(G) \wedge \bar{y} \in \mathcal{Y}_{\text{good}} \wedge r^* \in \mathcal{R}^{\bar{y}}_{\text{good}}] + \Pr_{\bar{y},r^*}[B(\bar{y}) = 1 \wedge \bar{y} \in \mathcal{Y}_{\text{good}} \wedge r^* \notin \mathcal{R}^{\bar{y}}_{\text{good}}]) \\
&\geq (1 - \frac{1}{2^k}) \Pr_{s,r^*}[G(s) \in \text{range}(G) \wedge G(s) \in \mathcal{Y}_{\text{good}}] - \Pr_{\bar{y},r^*}[\bar{y} \in \text{range}(G)] - \Pr_{\bar{y},r^*}[r^* \notin \mathcal{R}^{\bar{y}}_{\text{good}}] \\
&\geq (1 - \frac{1}{2^k}) \Pr_s[G(s) \in \mathcal{Y}_{\text{good}}] - 2^n/2^{l(n)} - 1/2^k \qquad (12)
\end{aligned}$$

Rewriting inequality 10 using this new bound yields

$$|p_{s,1} - p_{\bar{y},1}| \geq (1 - \frac{1}{2^k})(2P(\mathcal{Y}_{\text{good}}) - 1) - 2^{n-l(n)} - 1/2^k \qquad (13)$$

Equation 11 becomes

$$\begin{aligned}
|p_s - p_{\bar{y}}| &\geq (1 - \frac{1}{2^k})(2P(\mathcal{Y}_{\text{good}}) - 1) - 2^{n-l(n)} - 1/2^k - (2 - 2P(\mathcal{Y}_{\text{good}})) \\
&= 4P(\mathcal{Y}_{\text{good}}) - 3 - 2^{n-l(n)} - \frac{2P(\mathcal{Y}_{\text{good}})}{2^k} \geq 4P(\mathcal{Y}_{\text{good}}) - 3 - 2^{n-l(n)} - \frac{2}{2^k} \qquad (14)
\end{aligned}$$

where the last inequality follow from the fact that $P(\mathcal{Y}_{\text{good}}) \leq 1$. Therefore

$$P(\mathcal{Y}_{\text{good}}) \geq \frac{3}{4} + \frac{1}{4}(\frac{1}{2^{l(n)-n}} + \frac{2}{2^k}) + \frac{1}{4}\gamma \implies |p_s - p_{\bar{y}}| \geq \gamma \qquad (15)$$

Since we in fact have $P(\mathcal{Y}_{\text{good}}) > \frac{3}{4}$ with strict inequality, let $\eta > 0$ be such that $P(\mathcal{Y}_{\text{good}}) \geq \frac{3}{4} + \eta$. Choose N_2, k large enough so that $n \geq N_2 \implies \frac{1}{2^{l(n)-n}} + \frac{2}{2^k} < \eta/2$, and set $\gamma = 2\eta$. Because ρ is a property of algorithm A, this choice of k is independent of the length of the string y input to algorithm B, so k is a constant for the purposes of characterizing the complexity of algorithm B. Thus setting k to have this value does not affect the assessment that B is a polynomial-time algorithm. Then (15) holds, so for all $n > N := \max(N_1, N_2)$,

$$\exists \gamma > 0 \text{ s.t. } |\Pr_{s,r^*}[B(G(s)) = 1] - \Pr_{\bar{y},r^*}[B(\bar{y}) = 1]| \geq \gamma$$

□

4 Tractability results

Theorem 4. For any $B \in \mathbb{R}$ and any $\epsilon, \rho \in (0,1)$, there exists a probabilistic polynomial time typical-case inference algorithm A with tolerances ϵ, ρ, which can do inference under any inference schema such that the observation and query variables have mutual information bounded by B.

That is, for any inference schema (P, V, E, X, Y) such that the mutual information $I(X;Y) < B$, and any $x \in \{0,1\}^{|X|}$, there exists a set $\mathcal{Y}_{\text{good}} \subseteq \{0,1\}^{|Y|}$ with $P(\mathcal{Y}_{\text{good}}) > 1 - \rho$ such that

$$\Pr[|A(I, x, y) - P(X = x|Y = y)| < \epsilon] > 1 - \epsilon$$

where the probability is taken over the randomness of A.

Corollary 5. For any fixed C and any $\epsilon, \rho \in (0,1)$, there exists a probabilistic polynomial time typical-case inference algorithm A which can do inference in any inference schema in which the number of observation variables $|Y| < C$.

Proof. (Corollary 5.) Say $|Y| < C$. Then $|\{0,1\}^{|Y|}| < 2^C$. Thus, regarding Y as a random variable, the entropy $H(Y) < C$. Thus the mutual information $I(X;Y) = H(Y) - H(Y|X) < C$. Thus a bound on the number of observed variables yieldds a bound on the mutual information between the observed and query variables, so Theorem 4 can be applied. □

Proof. (Theorem 4.) Say $I(X,Y) < B$.Then $\mathbb{E}_{x,y\sim P}[\log \frac{P(x,y)}{P(x)P(y)}] < B$. By lemma 6, this implies

$$\Pr[\frac{P(x,y)}{P(x)P(y)} > a] \leq \frac{1}{\log a}(B+1)$$

for any a. Thus

$$\Pr[\frac{P(x,y)}{P(x)P(y)} > 2^{T(B+1)}] \leq \frac{1}{T(B+1)}(B+1) = 1/T$$

Chaterjee and Diaconis's result then implies that if likelihood weighting is run using $T \times 2^{T(B+1)}$ samples, prodcing estimate $\hat{p}$ of $P(X = x, Y = y)$, then

$$\mathbb{E}|\hat{p} - P(X = x, Y = y)| \leq (\frac{1}{\sqrt{T}} + 2\sqrt{\Pr_{(x,y)\sim P}[\frac{P(x,y)}{P(x)P(y)} > 2^{T(B+1)}]}) \leq \frac{1}{\sqrt{T}} + 2/\sqrt{T} = 3/\sqrt{T}$$

Thus by Markov's inequality,

$$\Pr[|\hat{p} - P(X = x, Y = y)| \geq \epsilon] \leq \frac{3}{\epsilon\sqrt{T}}$$

For any ϵ and any constant C, we can choose T large enough that this probability is less than ρ.

While the constants involved here are very large, they are still constants depending only on the fixed parameters B, ϵ, ρ.

□

Lemma 6. Let P and Q be discrete probability distributions on the same space. Say $\mathbb{E}_P[\log \frac{P}{Q}] < B$. Then for any $a > 1$,

$$\Pr_{x\sim P}[P(x)/Q(x) > a] \leq \frac{1}{\log a}(B+1)$$