Bayesianism, frequentism, and the planted clique, or do algorithms believe in unicorns?

Boaz Barak

(See also pdf version)

The divide between frequentists and Bayesians in statistics is one of those interesting cases where questions of philosophical outlook have actual practical implications. At the heart of the debate is Bayes' theorem:

$\Pr[A|B] = \Pr[A \cap B] / \Pr[B]$.

Both sides agree that it is correct, but they disagree on what the symbols mean. For *frequentists*, probabilities refer to the fraction that an event happens over repeated samples. They think of probability as counting, or an extension of *combinatorics*. For *Bayesians*, probabilities refer to degrees of belief, or, if you want, the odds that you would place on a bet. They see probability as an extension of logic.[1]

A Bayesian is like Sherlock Holmes, trying to use all available evidence to guess who committed the crime. A frequentist is like the lawmaker who comes up with the rules how to assign guilt or innocence in future cases, fully accepting that in some small fraction of the cases the answer will be wrong (e.g., that a published result will be false despite having statistical significance). The canonical question for a Bayesian is "what can I infer from the data about the given question?" while for a frequentist it is "what experiment can I set up to answer the question?". Indeed, for a Bayesian probability is about the degrees of belief in various answers, while for a frequentist probability comes from the random choices in the experiment design.

If we think of algorithmic analogies, then given the task of finding a large clique in graphs, a frequentist would want to design a general procedure that has some assurances of performance on all graphs. A Bayesian would be only interested in the particular graph he's given. Indeed, Bayesian procedures are often exponential in the worst case, since they want to use all available information, which more often than not will turn out to be computationally costly. Frequentists on the other hand, have more "creative freedom" in the choice of which procedure to use, and often would go for simple efficient ones that still have decent guarantees (think of a general procedure that's meant to adjudicate many cases as opposed to deploying Sherlock Holmes for each one).

Given all that discussion, it seems fair to place theoretical computer scientists squarely in the *frequentist* camp of statistics. But today I want to discuss what

[1] As discussed in my previous post, this is somewhat of a caricature of the two camps, and most practicing statisticians are pragmatic about this and happy to take ideas from either side as it applies to their particular situation.

a *Bayesian* theory of computation could look like. As an example, I will use my recent paper with Hopkins, Kelner, Kothari, Moitra and Potechin, though my co authors are in no way responsible to my ramblings here.

Peering into the minds of algorithms.

What is wrong with our current theory of algorithms? One issue that bothers me as a cryptographer is that we don't have many ways to give evidence that an average-case problem is hard beyond saying that "we tried to solve it and we couldn't". We don't have a web of reductions from one central assumption to (almost) everything else as we do in worst-case complexity. But this is just a symptom of a broader lack of understanding.

My hope is to obtain general heuristic methods that, like random models for the primes in number theory or the replica method in statistical physics, would allow us to predict the right answer to many questions in complexity, even if we can't rigorously prove it. To me such a theory would need to not just focus on questions such as "compute $f(x)$ from x" but tackle head-on the question of *computational knowledge*: how can we model the inferences that computationally bounded observers can make about the world, even if their beliefs are incorrect (or at least incomplete). Once you start talking about observers and beliefs, you find yourself deep in Bayesian territory.

What do I mean by "computational knowledge"? Well, while generally if you stop an arbitrary `C` program before it finishes its execution then you get (to use a technical term) bubkas, there are some algorithms, such as Monte Carlo Markov Chain, belief propagation, gradient descent, cutting plane, as well as linear and semidefinite programming hierarchies, that have a certain "knob" to tune their running time. The more they run, the higher quality their solution is, but one could try to interpret their intermediate state as saying something about the knowledge that they accumulated about the solution up to this point. Even more ambitiously, one could hope that in some cases one of those algorithms is the *best*, and hence its intermediate state can be interpreted as saying something about the knowledge of *every* computationally bounded observer that has access to the same information and roughly similar computational resources.

Modeling our knowledge of an unknown clique

To be more concrete, suppose that we are given a graph G on n vertices, and are told that it has a unique maximum clique of size $k = n^{0.49}$. Let $w \in \{0,1\}^n$ denote the characteristic vector of the (unknown to us) clique. We now ask what is the probability that $w_{17} = 1$. This question seems to make no sense. After all, I did not specify any distribution on the graphs, and even if I did, once G is given, it completely fixes the vector x and so either $w_{17} = 0$ or $w_{17} = 1$.

But if you consider probabilities as encoding beliefs, then it's quite likely that a computationally bounded observer is not *certain* whether 17 is in the clique or not. After all, finding a maximum clique is a hard computational problem. So if T is much smaller than the time it takes to solve the k-clique problem (which is $n^{const\cdot k}$ as far as we know), then it might make sense for time T observers to assign a probability between 0 and 1 to this event. Can we come up with a coherent theory of such probabilities?

Here is one approach. Since we are given no information on G other than that it has an k-sized clique, it makes sense for us to model our prior knowledge using the maximum entropy prior of the uniform distribution over k-sized sets. But of course once we observe the graph we learn something about this. If the degree of 17 is smaller than k then clearly $w_{17} = 0$ with probability one. Even if the degree of 17 is larger than k but significantly smaller than the average degree, we might want to adjust our probability that $w_{17} = 1$ to something smaller than the a priori value of k/n. Of course by looking not just at the degree but the number of edges, triangles, or maybe some more global parameters of the graph such as connected components, we could adjust the probability further. There is an extra complication as well. Suppose we were lied to, and the graph does not really contain a clique. A computationaly bounded observer cannot really tell the difference, and so would need to assign a probability to the event that $w_{17} = 1$ even though w does not exist. (Similarly, when given a SAT formula close to the satisfiability threshold, a computationally bounded observer cannot tell whether it is satisfiable or not and so would assign probability to events such as "the 17^{th} variable gets assigned `true` in a satisfying assignment" even if the formula is unsatisfiable.) This is analogous to trying to compute the probability that a unicorn has blue eyes, but indeed computationally bounded observers are in the uncomfortable positions of having to talk about their beliefs even in objects that mathematically cannot exist.

Computational Bayesian probabilities

So, what would a consistent theory of "computational Bayesian probabilities" would look like? Let's try to stick as closely as possible to standard Bayesian inference. We think that there are some (potentially unknown) parameters θ (in our case consisting of the planted vector w) that yield some observable X (in our case consisting of the graph G containing the clique encoded by w, say in adjacency matrix representation). As in the Bayesian world, we might denote $X \sim p(X|\theta)$ to say that X is sampled from some distribution conditioned on θ, and $\theta \sim p(\theta|X)$ to denote the conditional distribution on θ given x, though more often than not, the observed data X will completely determine the parameters θ in the information theoretic sense (as in the planted clique case). Our goal is to infer a value $f(\theta)$ for some "simple" function f mapping θ to a real number (in our case $f(w)$ is simply w_{17}). We denote by $\tilde{f}(X)$ the computational estimate for $f(\theta)$ given X. As above, we assume that the estimate $\tilde{f}(X)$ is based on some prior distribution $p(\theta)$

A crucial property we require is *calibration*: if θ is truly sampled from the prior distribution $p(\theta)$ then it should hold that

$\mathbb{E}_{\theta \sim p(\theta)} f(\theta) = \mathbb{E}_{\theta \sim p(\theta), X \sim p(X|\theta)} \tilde{f}(X) \quad (*)$

Indeed, the most simple minded computationally bounded observer might ignore X completely and simply let $\tilde{f}(X)$ to be the a priori expected value of $f(\theta)$. A computationally unbounded observer will use $\tilde{f}(X) = \mathbb{E}_{\theta \sim p(\theta|X)} f(\theta)$ which in particular means that when (as in the clique case) X completely determines θ, it simply holds that $\tilde{f}(X) = f(\theta)$.

But of course we want to also talk about the beliefs of observers with intermediate powers. To do that, we want to say that $\tilde{f}$ should respect certain computationally efficient *rules of inference*, which would in particular rule out things like assigning a positive probability for an isolated vertex to be contained in a clique of size larger than one. For example, if we can infer using this system from X that $f(\theta)$ must be zero, then we must define $\tilde{f}(X) = 0$. We also want to satisfy various internal consistency conditions such as linearity between our estimates for different functions f (i.e., that $\widetilde{f+g} = \tilde{f} + \tilde{g}$).
Finally, we would also want to ensure that the map $X \mapsto \tilde{f}(X)$ is "simple" (i.e., computationally efficient) as well.

Different rules of inference or *proof systems* lead to different ways of assigning these probabilities. The Sum of Squares algorithm / proof system is one choice I find particularly attractive. Its main advantages are:

- It encapsulates many algorithmic techniques and for many problems it captures the *best known* algorithm. That makes it a better candidate for capturing (for some restricted subset of all computational problems) the beliefs of *all* computationally bounded observers.

- It corresponds to a very natural proof system that contains in it many of the types of arguments, such as Cauchy Schwarz and its generalizations, that we use in theoretical computer science. For this reason it has been used to find constructive versions of important results such as the invariance principle.

- It is particularly relevant when the functions f we want to estimate are *low degree polynomials*. If we think of the data that we observe as inherently *noisy* (e.g., the data is a vector of numbers each of which corresponds to some physical measurement that might have some noise in it), then it is natural to restrict ourselves to that case since high degree polynomials are often very sensitive to noise.

- It is a tractable enough model that we can prove lower bounds for it, and in fact have nice interpretations as to what these "estimates" are, in the sense that they correspond to a distribution-like object that "logically approximates" the Bayesian posterior distribution of θ given X.

SoS lower bounds for the planted clique.

Interestingly, a lower bound showing that SoS fails on some instance amounts to talking about "unicorns". That is, we need to take an instance X that did *not* arise from a model of the form $p(X|\theta)$ (e.g., in the planted clique case, a graph G that is random and contains no planted clique) and still talk about various estimate of this fictional θ.

We need to come up with reasonable "pseudo Bayesian" estimates for certain quantities even though in reality these estimates are either completely determined (if X came from the model) or simply non-sensical (if X didn't). That is, for every "simple" function $f(\theta)$, we need to come up with an estimate $\tilde{f}(X)$. In the case of SoS, the notion of "simple" consists of functions that are *low degree polynomials in* θ. For every low degree polynomial $f(w)$, we need to give an estimate $\tilde{f}(G)$ that estimates $f(w)$ from the graph G. (Of course if we had unbounded time and G really was from the planted distribution then we could simply recover the maximum clique w completely from G.)

For example, if 17 is not connected to 27 in the graph G, then our estimate for $w_{17}w_{27}$ should be zero. What might be less clear is what should be our estimate for w_{17}— i.e., what do we think is the conditional probability that 17 is in the clique given our observation of the graph G and our limited time. The a priori probability is simply $\frac{k}{n}$, but if we observe that, for example, the degree of 17 is a bit bigger than expected, say, $\frac{n}{2} + \sqrt{n}$ then how should we update this probability? The idea is to think like a Bayesian. If 17 does not belong to the clique then its degree is roughly distributed like a normal with mean $\frac{n}{2}$ and standard deviation $\frac{\sqrt{n}}{2}$. On the other hand, if it does belong to the clique then its degree is roughly distributed like a normal with mean $\frac{n}{2} + k$ and the same standard deviation. So, we can see that if the degree of 17 was Z then we should update our estimate that 17 is in the clique by a factor of roughly $\Pr[N(\frac{n}{2} + k, \frac{n}{4}) = Z] / \Pr[N(\frac{n}{2}, \frac{n}{4}) = Z]$. This turns out to be $1 + ck/\sqrt{n}$ in the case that the degree was $\frac{n}{2} + \sqrt{n}$.

We can try to use similar ideas to come up with how we should update our estimate for w_{17} based on the number of triangles that contain it, and generalize this to updates of more complicated events based on more general statistics. But things get very complex very soon, and indeed prior work has only been able to carry this out for estimates of polynomials up to degree four.

In our new paper we take an alternate route. Rather than trying to work out the updates for each such term individual, we simply declare by fiat that our estimates should:

- Be simple functions of the graph itself. That is $\tilde{f}(G)$ will be a low degree function of G.
- Respect the calibration condition (*) for all functions f that can depend on the graph only in a low degree way.

This condition turns out to imply that our estimates automatically respect all the low degree statistics. The "only" work that is then left is to show that they satisfy the constraint that the estimate of a $f(\theta)^2$ is always non-negative. This turns out to be quite some work, but it can be thought as following from a recursive structure versus randomness partition. This might seem to have nothing to do with other uses of the "structure vs randomness" approach such as in the setting of the regularity lemma or the prime numbers, but at its core, the general structure vs. randomness argument is really about Bayesian estimates. The idea is that given some complicated object O, we separate it to the part containing the *structure* that we can infer in some computationally bounded way, and then the rest of it, since it contains no discernable structure, can be treated as if it is *random* even if it is fully deterministic since a Bayesian observer will have uncertainty about it and hence assign it probabilities strictly between zero and one. Thus for example, in the regularity lemma, we can think of a bounded observer that cannot store a matrix M in full, and so only remembers the average value in each block, and considers the entry as random inside it. Another example is the case of the set P of primes, where a bounded observer can infer that all but finitely many members of P do not divide $2, 3, 5, 7, 11, \ldots$ up to some not too large number w, but beyond that will simply model P as a random set of integers of the appropriate density. Similarly, in the case of the regularity lemma, we split a matrix into a low rank component containing structure that we can infer, with the rest of it treated as random.

I think that a fuller theory of computational Bayesian probabilities, which would be the dual to our standard "frequentist" theory of pseudorandomness, is still waiting to be discovered. Such a theory would go far beyond just looking at sums of squares.