Michael Jordan: Bayesian Nonparametrics

March 29, 2016

0.1 Bayesian Nonparametrics and Combinatorial Stochastic Processes

What's troubling is that there are two ways of doing statistics: Bayesian and frequentist, and they hate each other - there's no other field with such a disconnect at the core of the field. We hope eventually this issue will go away. Most of my work the past few years has become frequentist: I need to prove theorems about how computing engages with inference. In particular using optimization tools has been really useful. At some level what a frequentist can tell you is - I'm going to develop a method and 95% of the time it'll work. But I want to get the best method I want for a specific setting: so for more applied work, the Bayesian perspective works out better.

As the pendulum swings, I see a theoretical side coming. This is an area I don't usually talk about. Nonparametrics are important: they allow the model to grow as you get more and more data.

Example: proteomic dataset - angles of a protein with respect to the previous amino acid. How should we model data like this? A lot of the clusters aren't wellexplained by physical principles. As a statistician, this looks like a clustering problem. Here's another one: Speaker diarization problem. You have a single microphone and you want to infer who spoke when. Looks kind of like a changepoint detection problem but you can reapply what you learn at one segment for other segments, but you don't know how many people are in the room. In the last one: motion capture analysis; want to find coherent "behaviors" that transfer to other time series that should transfer to other datasets. You don't have a dictionary of exercise routines or who's doing what at each time.

Bayesian nonparametrics brings the two threads of computer science and statistics together. We have instead *stochastic processes* for representing flexible data structures - organically growing data structures that handle a lot of data under uncewrtainty.

Examples of stochastic processes that are useful in papplications include distributions on - directed trees, unbounded (genetics), distributions on partitions - distributions on grammars, copulae, distributions - you get the recursivity which is in the spirit of computer science.

General mathematical tool - completely random measures: a nice core computer science-y, mathematical structure.

If you're doing Bayes you have $posterior \propto likelihood \times prior$.

For parametric models:

$$p(\theta|x) \propto p(x|\theta)p(\theta)$$

For nonparametric model: G is a general stochastic process - infinite dimensional random variable

$$P(G|x) \propto p(x|G)P(G)$$

This frees us to work with flexible data structures. So I need to put something smooth, nonnegative - a density on the space of angles for the proteomic dataset (Ramachandran diagrams). E.g. gaussian mixture models or density estimate? You get an estimate which is no very useful. Problem: too many neighborhoods around the atoms. In statistics you say you can aggregate over all the data, but in a real life problem you don't have the right granularity - but then we get unevenness. People want to segment the data in industry. They want a special model for different metropolitans, etc. So this is the tension you will face in statistic. Bayes will speak to this directly.

The real answer is to break them apart and share information among them.

Stochastic processes that we will use to solve them: Chinese Restaurant Process: n customers sit down in a Chinese restaurant with an ifninite number of tables -¿ preferential attachment dynamics where there's a probability of starting a new table.

This looks like a clustering setting - partitioning - to put this on data we have to put parameters adn distributionsover our clusters. Take the CRp and augment it. put a distribution indexed by our parameter. E.g. put Gaussian data on the clusters, each table is a 'bump" in our density. Bayesian aguments this partitioning stochastic process with a likelihood. So you have a distribution indexed by the parameter.

Now we can assign probabilities to any arbitrary number of clusters (infinite). Prior depends on the amount of data. You can now generate data from this. inverse Wishart - standard thing to do. Let's try and study a little bit more the mathematical structure.

This is a distribution on partitions, has the property of exchangeability - this means that the distribution is independent on th elabeling of the customers. E.g. if they came in a different order - the probability that Mike sits with barbara is independent of ordering. This is nontrivial and deeply interesting. This tells you by De Finetti - there must exist an underlying rnadom measure such that the CRP is obtained by integrating out the random measure. This turns out to be the Dirichlet process - this is interesting for statistical modeling purposes. (Blackwell & MacQueen). It's like 4 pages.

Let me prove exchangeability:O write down the probability of the partition. It's elementary. Let N be the total number of customers. Write $\Pi[n]$ in terms of cells of the partition, K groups in the partition.

$$P(\pi_{[N]}) = \frac{\alpha^K}{\alpha^(N)} \Pi_{c \in \pi_{[N]}} (|c| - 1)!$$

The second person sat there w.p. probability proportional to 1, 2, 3... ends when you get to total number of customers -1. Do this for all clusters in the partition.

This only depends on the cardinality of the cluster. Product says it's independent of the ordering of the cluster. So this is exchangeable.

De Finetti's theorem: a set of random variables is infinitely exhangeable if and only if the joint probability of any finite subsequence can be written as - integral of some product where you integrate out G. One direction is trivial; RHS implies exchangeability. Deep direction is the forward: fourier analysis on groups.

Fundamental theorem of Bayesian analysis! There must exist a prior probability behind this.

Consequences of exchangeability are quite important. Must have underlying random measure behind these processes; can't just be parameter vector - you need to be nonparametric (special classes of stochastic processes).

Now if you build posterior inferene algorithms for the CRP: your mathematics instantiate the rnadom measure - from this you can build actual algorithms.

What classes of random measures do we study? CS hat: Completely Random Measures. If these measures are entangled we can't get scaleable algorithms: we need this random measure to breka apart in pieces. Completely random measures assign independent mass to nonintersecting subsets of Ω. [e.g. diagram of the completely random measures] They live on some space. If when I define a set A, B, here - the amount of mass dropping into those - if the RVs are independent, then this random measure is called completely random.

In terms of divide and conquer this says I can put disjoint subsets on different processors adn not worry aobut them. Tons of them are of interest. Prof. Cinlar has a great book. Poisson processes, gamma processes, beta, ... compound Poisson, limits thereof. Dirichlet not completely random but obtained by normalizing gamma.

Characterization of these processes: there's only way to get these processes. This is it. Take your original space upon which you want to put this random measure. Add to this another dimension: on this space add a rate function: sample from this poisson process and connect the samples vertically to their coordinates in Ω. Draw from a poisson point process with a certain rate function. Drop a line from the x's down to this Ω space. This picture is now a completely random measure by the Poisson process. Amount of mass is independent.

What Kingman proved is that this is the only way you can get random processes. So the Poisson process is really very important: it's a generator of many other interesting objects. In the combinatorial world it's the generator, just as Brownian is generator for finance context.

So there's this object called the Dirichlet process. This arises as: let the rate function to be a Gamma density, goes from $0 \to \infty$ and decays. It's going to be improper. Take that object adn then normalize: sum over all atoms. That defines the Dirichlet process. Its marginal prob. density is the Chinese restaurant process.

[Pictures of doing this] - Dirichlet process defined by two parameters, α_0, G_0. Going to build a dirichlet process mixture model. Repeatly fix G and draw one of the atoms from G. That will be a parameter to draw data from. $x_i|\theta_i \sim F_\theta$.

Tutorial on how to do these things coming out summer 2016 how to get stick breaking? Chained polya urns. These densities aren't distributions anymore, they're atoms.

We can ask: what's the probability that the first atom is equal to the second atom? Is that a well-defined question? What's the problem that the atom is also in the same place? That's on the marginal density on the θs. That sounds hopelessly hard.

The answer is $\frac{1}{2}$. The general answer is the Chinese Restaurant Process. That's what was proved in 4 pgs by Blackwell.

As the years have worn on, there's many of these relationships between an underlying stochastic process and a combinatorial object you might use for dat analysis - Dirichlet Process -¿ CRP (Polya urn).

Beta -¿ Indian buffet process).

Hierarhical Dirichlet -¿ Chinese restaurant franchise

HDP-HMM -¿ infinite HMM.

Nested dirichlet process -¿ nested Chinese Restaurant process.

Now let's go to this stratification process. The main reason you're a Bayesian is so that you can do hierarchical model: so you can learn a little about someone and use it for someone else. Build hierarchies where the ingredients are completely random measure. In the frequentist setting this is shrinkage.

E.g. multiple estimation problems. Want to estimate multiple Gaussian means, $x_{ij} \sim N(\theta_i, \sigma_i^2)$. If you do MLE on each groups, it doesn't work well in practice or in theory.

Bayesian solution is to pretend the θs are related, drawn from an underlying θ, which generates data. So if you learn something, posterior inferenceyields shrinkage - the posterior mean for each θ_i combines data from all the groups.

Notation: the plate notation is equivalent to the tree notation that doesn't replicate.

Question: can we do this with Bayesian nonparametrics?

Classic Efron/Morris machinery. Suppose we have multiple neighborhoods/groups; one Dirichlet process for each one of them. But now we need to tie these processes together somehow. Maybe liek $G_i \sim DP(\alpha G_0)$ where G_0 is an underlying base measure that we estimate. However this approach **fails**: none of the clusters are shared across the groups.

The answer is instead starting with the underlying measure G_0: from G_0, sample a G a discrete measure; then the children discrete measures share atoms because their source is discrete. So you get sharing of atoms because each one gets a base measure drawn from something everyone shares. **Then this becomes a hierarchical dirichlet process mixtures.** This is just a nonparametric version of LDA.

We agree this is a better version of topic modeling.

Chinese Restaurant Franchise Global menu: with multiple restaurants. When you sit at a table you go to the global menu, get a dish from the global menu, bring it back to the table and everyone has to eat that dish. Suppose Han comes in at Restaurant 2 and gets a dish from global menu; gets a dish proportional to the number of checkmarks of the dish. So information will flow between the restaurants and we will get sharing of the atoms.

That is not just a cute metaphor but what you get by marginalizing out the Hierarchical Dirichlet Process.

Returning to our problem: we want to model the neighborhoods of the protein problems. For everyone one of my 400 neighborhoods I will have a dirichlet process but they will be tied by this Hierarchical process. [Diagrams] You can notice some bumps where there is no data: frequentists would call this overfitting. As a nonparametric Bayesian, I'd say - maybe this system is hedging its bets. Maybe there was a little of data from the information sharing. That's exactly what happens that it's transferred over and hedging its bets. We can make this metric quantitative: over 20 amino acids, the log-probability on heldout data of choosing the correct answer. Two different models: improvement over a finite mixture. On a log scale you're seeing good improvement. I think of this as a killer app for this thing: This is solving the real problem.

I don't know how many clusters there are in the diagram. I need to shar einformation among the estimations.

In my last part here I'm going to turn to the problem of speaker diarization, and MCA - 60dimensional time series.

To do this in 10 minutes I'm going to telegraph some of the underlying structures. We're going to build up a dictionary which is combinatorial, have a prior, have a likelihood. This data is very oscillatory and continuous: we need a richer likelihood. We use HMMs for the likelihood. HMMs can be drawn in three different ways: double chain, double matrix (stochastic) where each row is a measure - k atoms that sum to 1 -, Hopefully you can see how you want to develop a hidden markov model with an infinite number of states by replacing a row with a Chinese Restaurant proces.

Issues with HMMs: How many states should I use? - use CRP that generates new states on the fly. How do we structure the state space - ¿ Bayesian nonparametric appraoch.

HDP-HMM.

It's an application of the HDP. Every time you enter a certain state you move into a Chinese restaurant process. Each current state is a different chinese restaurant. If I want states to be shraed among the chinese restaurants I need sharing of the atoms across my Chinese restaurants - this is wha tHDP handles. That's how you can take something like an HMM and make it nonparametric.

Now you have a mother transition dimennsion, each one of the rwos are like the picket diagrams I've been showing earlier on teh data. We did this on some synthetic data: gold standard - different states - actually it inadequately models temporal persistenceof states. It has split one state into two substates (which had nearly the same probability) so you get same probability. The point is that you got it wrong. unrealistically rapid dynamics.

How do you fix this? Add a little bit more prior knowledge: allow yourself to be a Bayesian. Add in the extra self transition probability that wants to tsay in same state on orders of hundreds of milliseconds - Sticky HDP-HMM.

Emily Fox did this by herself and beat the time. Beta process: Dirichlet process naturall yields a nmultinomial random variable (which one of n tables is the customer seitting at?)

But real-world industry clusterin gmodels have combinatorially number of tables. E.g. collaborative filtering. Number of clusters was going out of control. What you want is a bit of a feature vector. The bit vector method is good for segmentation/filtering.

How do you get a stochastic process which generates bit vector? You don't generate draws from a Dirichlet process, you draw beta process: you get a countable number of atoms and you toss all those coins and you get a bit vector. So you get a population of people, underlying draw from a beta process, you get things that are shared among people and some things that are shared. This is domne from the completely random measure picture. That's the rate function for the beta process - sparesely generating prior, stops at 1.

How do we apply this to solve the problem? Multiple time series - we have this data that people are doing these exercise routines. I'm going to use the beta process to generate an infintie number of things; draw from the Bernoulli process to pick out subsets of states. Represent behaviors as states in a nonoparametric HMM. What's the likelihood term? big infinitely dimensional HMM which has all the exercises anyone could ever do, if I come in with my bit vector, I pick out a submatrix, bring those parameters down, adjust, bring parameters to the big thing. So this will be transferred to the next time someone brings an overlapping submatrix. This infinite matrix will get filled in in a blocking structure and it'll be constructed in a blocking structure. 60d time series data: for each one fo those bit vector features, let it run - it found twisiting,

Claim: the open problems are still unsupervised. But this is all unsupervised: there's no data here. As the pendulum moves back from unsupervised, these kinds of methods and models will still be in the air. It's not a good idea to learn one hammer and learn about one kind of thing. Gotta learn about the whole span of things.