CS-ML SEMINAR

BAYESIAN NONPARAMETRICS AND COMBINATORIAL STOCHASTIC PROCESSES MARCH 28, 2016

LECTURER: MICHAEL JORDAN SCRIBE: KIRAN VODRAHALLI

Contents

1 Introduction

ABSTRACT: Bayesian nonparametric modeling and inference are based on using general stochastic processes as prior distributions. Despite the great generality of this definition, the great majority of the work in Bayesian nonparametrics is based on only two stochastic processes: the Gaussian process and the Dirichlet process. Motivated by the needs of applications, I present a broader approach to Bayesian nonparametrics in which priors are obtained from a class of combinatorial stochastic processes known as "completely random measures" (Kingman, 1967). In particular I will present models based on the beta process, the Bernoulli process, the gamma process and the Dirichlet process, and on hierarchical and nesting constructions that use these basic stochastic processes as building blocks. I will discuss applications of these models to several problem domains, including image processing, protein structural modeling, natural language processing and statistical genetics.

Being a statistician, there's really two ways of doing statistics: Bayesian and Frequentist. They kind of conflict with each other. There's such a disconnect at the core of the field. It still sits there as a gnawing issue, and we hope it will go away and there will be a reconciliation.

Lately I've been a frequentist. It's very useful to be able to give any procedure to prove things about it. If I'm working on a particular problem, I want the best solution I can get for a specific problem. If I do more applied work, then it'll be more Bayesian. As the pendulum swings, I think there will eventually be a theoretical side to Bayesian things.

Specifically this talk is about nonparametrics. We allow the model to grow in size as it needs.

2 Data Types

2.1 Proteomic Data Set

Proteins are a string of amino acids, and you can describe angles between the amino acids. Quantum mechanics explains some of these clusters, but a lot of them aren't. How should we model this kind of data?

2.2 Speaker Diarization

You have a waveform varying over time; you're trying to decide the chain of speakers: Who speaks when? You don't know how many people there are in the room.

2.3 Motion Capture Analysis

A person has 22 sensors on his body, and each has 3 degrees of freedom. The goal is to find coherent behaviors in the time series which transfer to other time series. You don't necessarily want to just throw deep learning at it. Let me give you a philosophical slide: Computer science and statistics are kind of merging at some level (I'll talk about this tomorrow). There's lots of reasons why. I like data structures, you have these organic objects. What it didn't really do well is it didn't have a notion of uncertainty and so on, now you see that. Statistics was complementary, but it didn't have data structures. If you put the two together, data structures and stats, you get stochastic processes. That's exactly what Bayesian nonparametrics is.

Some stochastic processes:

1. Directed trees of unbounded dpeth and fan out
2. Partitions
3. Grammars (rules)
4. Sparse binary infinite-dimensional matrices
5. Copulae
6. Distributions (recursive, you get distributions on distributions)

We have a general mathematical tool called **completely random measures**.

3 Bayesian Nonparametrics

You have to write posterior $\sim$ likelihood $\times$ prior. In other words,

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

Nonparametric means that θ is no longer a Euclidean vector. Instead of θ, we have G, an infinite dimensional random variable, like a tree process. So we write

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

One of the tensions of statistics is at some level, you want to partition your data into groups (different models for different parts of the U.S., or something), but you want to avoid having two few data points. So the idea is you have different views and you let them share information with each other.

3.1 Chinese Restaurant Processes

This is a preferential attachment kind of dynamics. That looks like a clustering setting, but now you have to put parameters and distributions on these clusters. The Bayesian augments this process with parameters at every table, and we can assign probabilities to densities with arbitrary shapes and locations. Compared to classicial finite mixture models, you have the number of clusters growing at rate $\log n$. The number of clusters grows with the data, which is typical for nonparametrics.

So now you have a generative prior, and there are many procedures which do this. Let's see if we can use this stratification to model our may densities.

The CRP is a distribution on partitions, and has a very powerful idea called **exchangeability**. This just says that the distribution is independent of the order that the customers come in. A theorem by De Finetti, there must exist a random measure such that the CRP is obtained by integrating out that random measure. That random measure turns out to be the Dirichlet process (Blackwell and MacQueen, 1972).

The proof of exchangeability is as follows. Let $\pi_{[N]}$ denote a partition of the integers, and let $\pi_{[N]} = (c_1, \cdots, c_K)$. Then,

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

So this just depends on the size of the cluster, and the product is commutative. Thus we have exchangeability.

Now, De Finetti's theorem says that there must exist an underlying measure G such that

$$P(X_1 \in C_1, \cdots, X_n \in C_n) = \int \prod_{i=1}^{n} P(X_i \in C_i | G) P(dG)$$

This is a Fourier analysis on groups kind of story. This is the math. This is sometimes called the Fundamental Theorem of Bayesian Analysis, when the labels don't matter.

This says that underlying random measures exist, and more over, that you have to be nonparametric in order to access these random measures. You explicitly instantiate that random measure in order to build actual algorithms on your computer.

This is kind of a mathematician's theorem. Now let's put on a computer scientist's hat. We need to get the random measure to break apart into pieces, to apply divide and conquer.

Kingma in 1968 defined **completely random measures**. You have some probability space Ω, and each location (atom) has some mass. You assign independent mass to nonintersecting subsets of Ω (for completely random). They don't necessarily sum to 1.

Now are there any interesting random measures. Professor Cinlar wrote a very interesting book on this. Poisson processes, gamma processes, beta processes, and others are completely random measures. Dirichlet processes are obtained by normalizing a completely random measure (but are not completely random). Now how do you get these objects? Why are they useful? Kingma proved a characterization theorem for these processes: There's only one way to get these processes. You take your original space Ω where we can have some distribution G_0 and add to this space another dimension, $\Omega \otimes R$ with a rate function specified as a product measure. Then sample from this Poisson process and connect the samples vertically to their coordinates in (project onto) Ω. So the Poisson process is the core: It's the generator of all these other processes in the combinatorical world, much like Brownian motion is a generator of many objects in stochastic processes.

There's this object called a Dirichlet process. You take the rate function to be a gamma density, which goes from $0 \to \infty$. Take that object, and now normalize it. This defines the famous object known as a Dirichlet process. If you take that object and call it G, and take $P(G)$ to be the Poisson process that generated it, you get the Chinese Restaurant Process.

Now there's many many relations between combinatorial processes and objects for data analysis. If you have feature vector problems, there's a naturally defined object called the beta process, and the Indian buffet process, which again are related by De Finiti's theorem. There's also Hierarchical Dirichlet Process and Chinese Restaurant Franchise, and others.

Now the main reason you're Bayesian is you can do hierarchical modeling. You learn a bit of data about you, and you learn about your friend, and you try to relate them.

What if you are trying to estimate multiple Gaussian means? If you do maximum likelihood on each of the separate groups, you get an estimator. So you need to share statistical strength. The Bayesian solution is hierarchical. This is also known as shrinkage.

The answer was to go to another level of hierarchy in order to do nonparametric Bayesian modeling. That's an object called the hierarchical Dirichlet process. This is now one of my most cited papers. This is the nonparametric version of LDA (David Blei).

You get an interesting process called a Chinese Restaurant Franchise, which has a global menu across the franchise, where you have multiple Chinese Restaurants. Basically you go up to the menu, select a food, put a checkmark, and then go sit at your table giving food to everyone. Then someone else comes and picks a table, and if it's a new table, they go up to the menu, pick a dish with probability proportional to the number of check marks, and bring it back to their table. And so on.

That's what you get by marginalizing out hierarchical Dirichlet process.

3.2 Bayesian Nonparametric Approach to HMM

Every time you enter a certain state, you move into a particular Chinese Restaurant Process. Each current state is a different Chinese restaurant. If you want to allow transitions from state 5 to state 3, you need to allow hierarchy.

3.3 Beta process

The Chinese Restaurant Process says you have to sit at one table. Well there are cases where I may be able to sit at multiple tables. Lots of real-world industrial clustering problems have that flavor nowadays. You buy a few books, and after you do that, you get put in a cluster with other people like you. The problem is the number of clusters was going out of control. You really want a bit of a feature vector. So it's a bit vector that describes me. The bit-vector metaphor vector is good. How do you get a stochastic process which generates bit vectors? You generate draws from a beta process. This gives you a countable number of atoms. You toss these coins, and get a bit vector. Toss them again and get a different bit vector. You get some things shared among us, and some individual things. It's a sparsity generating prior. You need numbers between 0 and 1. Each person who comes in the room draws from a Bernoulli process. Now the likelihood is a big infinite dimensional HMM. Now you bring down your parameters from the infinite space, fit your own personal HMM, and you're done. The next person who comes in will do the same thing, as well as fit the overlap between your features. Then both will refit their personal HMM.

Peopleare talking a lot about supervised learning (neural nets) these days. People keep saying all open problems are unsupervised, and this has been that way for 40 years. This is completely unsupervised.

I wouldn't say it's a good idea to learn one tool (i.e. stochastic gradient descent on deep nets), and I think this is making a pendulum swing back to prominence. I'm not working on it directly right now, but I think this should be very interesting.