Scribe Notes for 300D, Aug 8, 2013
Charles Zheng and Cyrus DiCiccio

Other Bayesian Nonparametric Models

So far we covered clustering, but there are lots of other models people use for data analysis.

- Clustering, including Hidden Markov Chain
- Density estimation more generally (e.g. graphical models)
- Factor analysis
- Regression and Classification

This lecture we will talk about two more Bayesian nonparametric models: nonparametric factor models using the Beta process and nonparametric regression or classification models using Gaussian process priors.

Nonparametric Factor Analysis

We have data $Y_1, \ldots, Y_N$ where $Y_n$ is a vector on $\mathbb{R}^D$. We assume that the variation in $Y_n$ can be explained by unobserved latent variables $U_{n,1}, U_{n,2}, \ldots,$ which are infinite in number, but which only a few are nonzero. These latent variables influence $Y_n$ through a number of common factors $\phi_1, \phi_2, \ldots$ which are shared among the individuals, but which have to be inferred.

So our model is

$$Y_i \sim f\left(\sum_{k=1}^{\infty} U_{n,k}\phi_k\right)$$

where $f$ is some parametric family.

To induce sparsity in the $U_{n,k}$ we let

$$U_{n,k} = Z_{n,k}W_{n,k}$$

where $Z_{n,1}, \ldots$, is an infinite sequence of binary-valued variables, all but finitely which are zero, and $W_{n,1}, \ldots$, is an infinite sequence of real-valued variables which are iid distributed according to some distribution $g$ (e.g. normal distribution). Letting $Z_n$ denote the infinite vector $Z_{n,1}, \ldots$, we can think of $Z_n$ as an “assignment vector” specifying which factors are associated to each observations. We also need to specify a prior for the “latent features” or “principal components” $\phi_k$, let us say they are iid according to some distribution $H$. 

The complication in this model comes from the mechanism for ensuring the sparsity of the $Z_{n,1}, Z_{n,2}, \ldots$ A Beta process $BP(\cdot)$ is used to generate an infinite vector of probabilities $\pi$, and then $Z_{n,k}$ are independently distributed as $Z_{n,k} \sim Bern(\pi_k)$. As long as the total of the probabilities $\pi_k$ is finite, only a finite number of $Z_{n,k}$ will be nonzero for each $n$, thus each observation $Y_n$ will only have a finite number of factors associated with it.

In summary, the model is

**Infinite Factor Analysis Model**

\[
\begin{align*}
\pi &\sim BP(\alpha, \beta) \\
Z_{n,k} &\sim Bern(\pi_k) \quad \text{for } n = \{1, \ldots, N\}, k = \{1, \ldots\} \\
W_{n,k} &\sim g \quad \text{for } n = \{1, \ldots, N\}, k = \{1, \ldots\} \\
\phi_k &\sim H \quad \text{for } k = \{1, \ldots\} \\
Y_n &\sim f\left(\sum_{k=1}^{\infty} Z_k W_{n,k}\right) \quad \text{for } n = \{1, \ldots, N\}
\end{align*}
\]

**The Beta Process**

The Beta Process has two parameters, $\alpha$ and $\beta$. We begin with a finite-dimensional approximation.

**Finite dimensional approximation:**

Let $K$ be a large integer.

Draw $\pi \in [0,1]^K$ by $\pi_1, \ldots, \pi_K$ iid $Beta(\alpha/K, \beta)$.

As $K$ goes to infinity, the distribution of $\pi$ will converge to the Beta process $BP(\alpha, \beta)$.

Using this finite-dimensional approximation allows us to discover some key facts about the Beta Process which will be valuable for inference.

In the finite model, $Z_n$ denotes the $K$-vector $Z_{n,1}, \ldots, Z_{n,K}$. To get a handle on joint distribution of $Z_1, \ldots, Z_N$, we view $Z_1, \ldots$ as a stochastic process starting with $Z_1$, in which the distribution of $Z_n$ is dependent on $Z_1, \ldots, Z_{n-1}$. Of course, our choice to start with $Z_1$ and end with $Z_N$ is completely arbitrary; due to exchangeability, we get the same result under any permutation of the indices.

Let us find the law of $Z_1$. Here $Z_{1,1}, \ldots, Z_{1,K}$ are independently distributed with marginals

\[
p(z_{1,k}) = \int_{\pi_k} Beta(\pi_k; \alpha/K, \beta) Bern(z_{1,k}; \pi_k) d\pi_k = Bern(z_{1,k}; \frac{\alpha/K}{\alpha/K + \beta})
\]

Next we find the law of $Z_2$ conditional on $Z_1$. Each component $Z_{2,k}$ depends...
only on $\pi_k|Z_{1,k}$. Recall that

$$\pi_k|Z_{1,k} \sim Beta(\alpha/K + Z_{1,k}, \beta + 1 - Z_{1,k})$$

Thus we get

$$p(z_{2,k}|Z_1) = \int_{\pi_k} Beta(\pi_k, \alpha/K + Z_{1,k}, \beta + 1 - Z_{1,k})Bern(z_{2,k}; \pi_k)d\pi_k$$

$$= Bern\left(z_{2,k}; \frac{\alpha/K + Z_{1,k}}{\alpha/K + \beta + 1}\right)$$

In general we will get

$$p(z_{n,k}|Z_1, \ldots, Z_{n-1}) = Bern\left(z_{n,k}; \frac{\alpha/K + \sum_{i=1}^{n-1} Z_{i,k}}{\alpha/K + \beta + n - 1}\right)$$

That is, the probability that $Z_{n,k}$ is activated depends on the number of times the $k$th factor has previously been activated.

**Limiting Properties**

Let $C_1 = \sum_{i=1}^{K} Z_{1,k}$, the total number of factors activated by the first observation. As $K$ becomes large, $Pr[Z_{1,k} = 1]$ scales with $1/K$, hence we get that

$$C_1 \xrightarrow{D} Poisson(\alpha/\beta)$$

Now conditioning on the first observation, the second observation will activate each of the $C_1$ previously activated factors with probability

$$Pr[Z_{2,k} = 1|Z_{1,k} = 1] = \frac{\alpha/K + 1}{\alpha/K + \beta + 1} \xrightarrow{k \to \infty} \frac{1}{\beta + 1}$$

Hence a total of $Binom\left(\frac{1}{\beta + 1}, C_1\right)$ factors will be activated in this way. However, recall that $K - C_1$ factors have never been previously activated. This time, each one gets activated with probability

$$Pr[Z_{2,k} = 1|Z_{1,k} = 0] = \frac{\alpha/K}{\alpha/K + \beta + 1} \approx \frac{1}{K} \frac{\alpha}{\beta + 1}$$

As $K \to \infty$, this probability again scales at $1/K$, hence letting $C_2$ denote the total number of new factors,

$$C_2 = \sum_{i=1}^{K} Z_{2,k}(1 - Z_{1,k})$$

we get $C_{2,k} \xrightarrow{D} Poisson(\alpha/(\beta + 1))$

In general, the $n$th observation will activate $Poisson(\alpha/(\beta + n - 1))$ new factors, and will activate previously activated factors with probability depending on the number of times each factor has previously been activated.
**Table 1: The Indian Buffet Process**

<table>
<thead>
<tr>
<th></th>
<th>Dish 1</th>
<th>Dish 2</th>
<th>Dish 3</th>
<th>Dish 4</th>
<th>Dish 5</th>
<th>Dish 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer 1</td>
<td>new</td>
<td>new</td>
<td>new</td>
<td>new</td>
<td></td>
<td>new</td>
</tr>
<tr>
<td>Customer 2</td>
<td>✓</td>
<td>new</td>
<td>✓</td>
<td>new</td>
<td>new</td>
<td>new</td>
</tr>
<tr>
<td>Customer 3</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Indian Buffet Analogy**

Just like in the case of the Chinese Restaurant Process, the distribution of $Z_1, \ldots, Z_n$ can be interpreted using a restaurant-related analogy.

The story is as follows. I am opening a new restaurant with not one but three special features: the menu is infinite, and the dishes all arrive instantly. Each dish has its own unique name, like “Braised Jalapeno Peacock”, but for convenience, I will refer to the first dish ever ordered in the restaurant as Dish 1, and the second as Dish 2, etc.

There is so much anticipation for this restaurant that at 8 AM, people are already lining up. I open the doors, and the first customer walks in. She has no idea about which dishes are good, so she just randomly three dishes that look appealing, which will hereafter be called Dish 1, Dish 2, and Dish 3 (see Table 1). These dishes are encoded into the vector $Z_1$.

The second customer walks in and sees the first customer happily eating Dishes 1, 2, and 3. He instantly decides, “I must also try Dishes 1 and 3” before even taking a look at the menu. Of course, once he gets the menu, two other items catch his eye, and he orders two dishes that have never been tried before: Dish 4 and 5. Dish 1,3,4 and 5 are encoded in $Z_2$.

The third customer walks in and sees:

- Both customers are eating Dishes 1 and 3
- Only one customer is eating each of Dish 2, 4, and 5

So the third customer thinks, “Dish 1 and 3 are really popular,” and decides to try Dish 3 immediately. But since Dish 2, 4, and 5 are not as popular, he does not decide to try any of them. He gets the menu, and although some of the names of the dishes catch his eye, he is somewhat less adventurous than the other two customers, and only orders one of them, Dish 6. Dishes 3 and 6 are encoded in $Z_3$. 
Both regression and classification problems can be viewed as the problem of estimating a function \( f(x) = \mathbf{E}[Y | X = x] \) given observations \((X_1, Y_1), \ldots, (X_N, Y_N)\). In the case of regression, \( Y \) is real-valued, while in classification, \( Y \) is binary-valued.

**GP for regression**

In the simplest case, we may assume that the data is observed directly from \( Y = f(X) \). If \( f \) is a Gaussian process, the posterior distribution for \( f \) will be a Gaussian process conditioned to pass through the observed points. A sample path for this process is given in Figure 1.

In practice, this model may be unrealistically restrictive. To loosen the assumptions, we consider the regression model

\[
Y_i = f(X_i) + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2)
\]

Under this model, a sample from the posterior distribution of \( f \) will not exactly through the data points. A sample path from this posterior is given in Figure 2.

Under the Bayesian approach, we model the underlying function \( f \) as a “random function.” One can construct a Gaussian process model for \( f \) with two ingredients, a mean function \( \mu(x) \) and a covariance kernel \( C(x_1, x_2) \). The mean function is often taken to be zero, and for “smooth” functions, one often uses a covariance function of the form \( C(x_1, x_2) = \tau \exp(x_1 - x_2)^2 / \lambda \).

Then the distribution of \( f \) is specified in terms of the finite-dimensional distributions: for any finite subset \( x_1, \ldots, x_n \), the joint distribution of \( f(x_1), \ldots, f(x_n) \)
Figure 2: Nonparametric Gaussian regression

is multivariate gaussian with mean \( \mu(x_1), \ldots, \mu(x_n) \) and covariance matrix \( (\sigma_{ij}) \) with \( \sigma_{ij} = C(x_i, x_j) \).

Supposing we only care about the value of \( f \) on a finite grid with points \( z_1, \ldots, z_K \) notice that the Gaussian Process model boils down to nothing more than a high-dimensional normal distribution. After all, we can obtain the joint distribution of our observations and the process on the grid,

\[
\begin{pmatrix}
Y_1 \\
Y_2 \\
\vdots \\
Y_N \\
f(z_1) \\
\vdots \\
f(z_K)
\end{pmatrix}
\sim N
\begin{pmatrix}
\mu(X_1) \\
\mu(X_2) \\
\vdots \\
\mu(X_N) \\
\mu(z_1) \\
\vdots \\
\mu(z_K)
\end{pmatrix}
\]

\[
\begin{pmatrix}
C(X_1, X_1) + \sigma^2 & C(X_1, X_2) & \cdots & C(X_1, X_N) & C(X_1, z_1) & \cdots & C(X_1, z_K) \\
C(X_1, X_2) & C(X_2, X_2) + \sigma^2 & \cdots & C(X_2, X_N) & C(X_2, z_1) & \cdots & C(X_2, z_K) \\
\vdots & \vdots & \ddots & \vdots & \vdots & \cdots & \vdots \\
C(X_1, X_N) & C(X_2, X_N) & \cdots & C(X_N, X_N) + \sigma^2 & C(X_N, z_1) & \cdots & C(X_N, z_K) \\
C(X_1, z_1) & C(X_2, z_1) & \cdots & C(X_N, z_1) & C(z_1, z_1) & \cdots & C(z_1, z_K) \\
\vdots & \vdots & \ddots & \vdots & \vdots & \cdots & \vdots \\
C(X_1, z_K) & C(X_2, z_K) & \cdots & C(X_N, z_K) & C(z_1, z_K) & \cdots & C(z_K, z_K)
\end{pmatrix}
\]

and then obtain the posterior of \( Z = (f(z_1), \ldots, f(z_K)) \) by the usual formula.
for conditional expectation and covariance
\[ Z \mid (X, Y) \sim N(\mu(Z) + \text{Cov}(Z, Y)\text{Cov}(Y, Y)^{-1}(Y - \mu(X)), \text{Cov}(Z, Z) - \text{Cov}(Z, Y)\text{Cov}(Y, Y)^{-1}\text{Cov}(Z, Z)) \]

Alternatively, suppose you want to simply obtain the posterior mean \( \hat{f}(x) \) given the observations. Again, we use the conditional expectation formula to obtain
\[
\hat{f}(x) = \mu(x) + \text{Cov}(x, Y)\text{Cov}(Y, Y)^{-1}(Y - \mu(X))
\]
But note that \( \text{Cov}(Y, Y)^{-1}(Y - \mu(X)) \) does not depend on \( x \). Let use define \( (w_1(Y), \ldots, w_N(Y)) = \text{Cov}(Y, Y)^{-1}(Y - \mu(X)) \) and also the functions \( c_i(x) = C(X_i, x) \). Then we get the result
\[
\hat{f}(x) = \mu(x) + \sum_{i=1}^{N} w_i(Y)c_i(x)
\]
For fixed design points \( X_i \) the posterior mean is a weighted sum of functions, where only the weights \( w_i \) depend on the observed \( Y_i \).

Hyperparameter Selection

Let’s say we use \( \mu = 0 \) and the covariance kernel \( C(x_1, x_2) = \tau \exp((x_1 - x_2)^2/\lambda) \), but now we want to choose hyperparameters \( \tau \) or \( \lambda \), and also the noise level \( \sigma^2 \) that are most appropriate for our model. Alternatively, we could even put priors on \( \tau, \lambda, \sigma^2 \), and try to draw from the posterior. In either case we need an expression for the marginal
\[
p(Y \mid X, \tau, \lambda, \sigma^2)
\]
But this is simple since given \( X, \tau, \lambda, \sigma^2 \) the joint distribution of \( Y \) is normal with a covariance matrix \( \Sigma(X, \tau, \lambda, \sigma^2) \) that we can directly compute. Then the marginal density we seek is given by the standard formula for multivariate normals
\[
p(Y \mid X, \tau, \lambda, \sigma^2) = (2\pi)^{N/2} |\Sigma(X, \tau, \lambda, \sigma^2)|^{-N/2} \exp -Y^T \Sigma(X, \tau, \lambda, \sigma^2)^{-1}Y
\]

GP for classification

When \( Y \) is binary-valued, the range of regression function \( g(x) = \mathbb{E}[Y \mid X = x] \) is restricted to the unit interval \([0, 1]\). Then it is no longer appropriate to model \( g \) directly by a gaussian process. However, the logit transform of \( g \),
\[
f(x) = \log \frac{g(x)}{1 - g(x)}
\]
can range over the real line, and can be modeled by a gaussian process. Unfortunately, though, it is no longer as straightforward to obtain the posterior distribution of \( f(x) \) given \((X_i, Y_i)\).