data-microscopes: Bayesian non-parametric inference made simple in Python

Stephen Tu
tu.stephenl@gmail.com

SF Python - August 20, 2014

Why do we need yet another machine learning library?

Why do we need yet another machine learning library?

There are already many Python libraries out there which specialize to some area of machine learning:

Why do we need yet another machine learning library?

There are already many Python libraries out there which specialize to some area of machine learning:

- `scikit-learn`

Why do we need yet another machine learning library?

There are already many Python libraries out there which specialize to some area of machine learning:

- `scikit-learn`
- `pymc`

Why do we need yet another machine learning library?

There are already many Python libraries out there which specialize to some area of machine learning:

- `scikit-learn`
- `pymc`
- `pybrain`

Why do we need yet another machine learning library?

There are already many Python libraries out there which specialize to some area of machine learning:

- `scikit-learn`
- `pymc`
- `pybrain`
- `pystan`

Why do we need yet another machine learning library?

There are already many Python libraries out there which specialize to some area of machine learning:

- `scikit-learn`
- `pymc`
- `pybrain`
- `pystan`
- `shogun`

Why do we need yet another machine learning library?

There are already many Python libraries out there which specialize to some area of machine learning:

- `scikit-learn`
- `pymc`
- `pybrain`
- `pystan`
- `shogun`
- Countless more (sorry if I missed yours)

Why do we need yet another machine learning library?

Why do we need yet another machine learning library?

Goal

Do one thing well: inference (via *Markov chain Monte Carlo*) for a fixed set of non-parametric models.

Why do we need yet another machine learning library?

Goal

Do one thing well: inference (via *Markov chain Monte Carlo*) for a fixed set of non-parametric models.

Doing it well means being **correct** and **fast**!

Great, so what does it mean to be Bayesian and non-parametric?

Great, so what does it mean to be Bayesian and non-parametric?

- **Disclaimer**: *way* too much to possibly cover in a short time!

Great, so what does it mean to be Bayesian and non-parametric?

- **Disclaimer**: *way* too much to possibly cover in a short time!
- *Bayesian*: encode our beliefs about the world with prior distributions.

Great, so what does it mean to be Bayesian and non-parametric?

- **Disclaimer**: *way* too much to possibly cover in a short time!
- *Bayesian*: encode our beliefs about the world with prior distributions.
- *Non-parametric*: extend these priors to incorporate fixed parameters (e.g. number of clusters).

Great, so what does it mean to be Bayesian and non-parametric?

- **Disclaimer**: *way* too much to possibly cover in a short time!
- *Bayesian*: encode our beliefs about the world with prior distributions.
- *Non-parametric*: extend these priors to incorporate fixed parameters (e.g. number of clusters).
- **We'll focus on one specific problem**: given a set of points, find the most likely clustering.

Great, so what does it mean to be Bayesian and non-parametric?

- **Disclaimer**: *way* too much to possibly cover in a short time!
- *Bayesian*: encode our beliefs about the world with prior distributions.
- *Non-parametric*: extend these priors to incorporate fixed parameters (e.g. number of clusters).
- **We'll focus on one specific problem**: given a set of points, find the most likely clustering.
- This will lead us to the *Dirichlet process mixture model*.

Dirichlet process mixture model

Dirichlet process mixture model

- Why not just k-means?

Dirichlet process mixture model

- Why not just *k*-means?

$$\operatorname*{argmin}_{C_1,\ldots,C_k} \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - \mu_i\|^2 \qquad \mu_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j$$

Dirichlet process mixture model

- Picking the k is annoying.

Dirichlet process mixture model

- Picking the k is annoying.

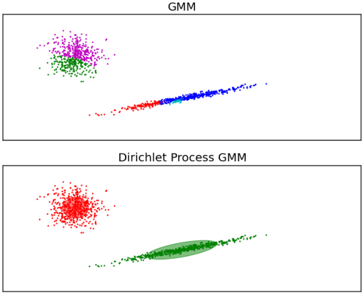

Figure: http://scikit-learn.org/stable/_images/plot_gmm_0011.png

Dirichlet process mixture model

- What if you have a *model* of the data? E.g. you know the data is from a mixture of gaussian distributions.

Dirichlet process mixture model

- What if you have a *model* of the data? E.g. you know the data is from a mixture of gaussian distributions.
- What if there is no *metric* on the data type? E.g. the data is categorical?

Dirichlet process mixture model

- The DPMM deals with these issues in a elegant framework.

Dirichlet process mixture model

- The DPMM deals with these issues in a elegant framework.
- Describes the generative process of n i.i.d. observations $Y_1, ..., Y_n$ as:

$$
\begin{aligned}
G &\sim \mathrm{DirichletProcess}(\alpha, H) \\
\theta_i | G &\sim G \\
Y_i | \theta_i &\sim F(\theta_i)
\end{aligned}
$$

Dirichlet process mixture model

- The DPMM deals with these issues in a elegant framework.
- Describes the generative process of n i.i.d. observations $Y_1, ..., Y_n$ as:

$$
\begin{aligned}
G &\sim \text{DirichletProcess}(\alpha, H) \\
\theta_i | G &\sim G \\
Y_i | \theta_i &\sim F(\theta_i)
\end{aligned}
$$

Dirichlet process mixture model

- The DPMM deals with these issues in a elegant framework.
- Describes the generative process of n i.i.d. observations $Y_1, ..., Y_n$ as:

$$\begin{aligned} G &\sim \text{DirichletProcess}(\alpha, H) \\ \theta_i | G &\sim G \\ Y_i | \theta_i &\sim F(\theta_i) \end{aligned}$$

where $F(\cdot)$ is a likelihood model (e.g. Gaussian), $H(\cdot)$ is the *prior* distribution (e.g. Normal-Inverse-Wishart) over the parameters of $F(\cdot)$, and $\alpha \in \mathbb{R}^+$ is chosen a-priori.

Dirichlet process mixture model

- The previous mathematical description is too abstract!

Dirichlet process mixture model

- The previous mathematical description is too abstract!

Dirichlet process mixture model

- The previous mathematical description is too abstract! (For instance, I didn't define what a Dirichlet Process is...)

Dirichlet process mixture model

- The previous mathematical description is too abstract! (For instance, I didn't define what a Dirichlet Process is...)

Dirichlet process

Dirichlet process

Definition

Let $H(\cdot)$ be a measure over $\mathcal{S}$ and $\alpha > 0$. We say G is drawn from a Dirichlet Process, written as $G \sim DP(\alpha, H)$ if for any (measurable) partition of $\mathcal{S} = (P_1, ..., P_n)$ we have

$$(G(P_1), ..., G(P_n)) \sim Dirichlet(\alpha H(P_1), ..., \alpha H(P_n))$$

Wait... what?!

Wait... what?!

Don't worry! Alternative view known as the **Chinese Restaurant Process** which is *way* more intuitive.

Chinese restaurant process

- To make things more concrete and simple, let's say each $Y_i \in \{0, 1\}^D$.

Chinese restaurant process

- To make things more concrete and simple, let's say each $Y_i \in \{0, 1\}^D$.
- Imagine a Chinese restaurant in SF with *infinite* tables.

Chinese restaurant process

- To make things more concrete and simple, let's say each $Y_i \in \{0,1\}^D$.
- Imagine a Chinese restaurant in SF with *infinite* tables.
- Each table T_i has a vector $\Theta^{(i)} = (\theta_1^{(i)}, ..., \theta_D^{(i)})$, with each $\theta_j^{(i)} \sim \text{Beta}(\gamma, \beta)$, j=1, ..., D.

Chinese restaurant process

- To make things more concrete and simple, let's say each $Y_i \in \{0, 1\}^D$.
- Imagine a Chinese restaurant in SF with *infinite* tables.
- Each table T_i has a vector $\Theta^{(i)} = (\theta_1^{(i)}, ..., \theta_D^{(i)})$, with each $\theta_j^{(i)} \sim \text{Beta}(\gamma, \beta)$, j=1, ..., D.
- To draw a value Y_i, first pick a table T_j, and then draw $Y_i \sim (\text{Bernoulli}(\theta_1^{(j)}), ..., \text{Bernoulli}(\theta_D^{(j)}))$.

Chinese restaurant process

- To make things more concrete and simple, let's say each $Y_i \in \{0,1\}^D$.
- Imagine a Chinese restaurant in SF with *infinite* tables.
- Each table T_i has a vector $\Theta^{(i)} = (\theta_1^{(i)}, ..., \theta_D^{(i)})$, with each $\theta_j^{(i)} \sim \text{Beta}(\gamma, \beta)$, j=1, ..., D.
- To draw a value Y_i, first pick a table T_j, and then draw $Y_i \sim (\text{Bernoulli}(\theta_1^{(j)}), ..., \text{Bernoulli}(\theta_D^{(j)}))$.
- To reference our previous notation, $H = \text{Beta}(\gamma, \beta)^D$ and $F = \text{Bernoulli}(\cdot)^D$.

Chinese restaurant process

- So how do we pick a table?

Chinese restaurant process

- So how do we pick a table?
- Suppose there are n_j existing people at table T_j, and suppose we are the $n+1$-th observation.

Chinese restaurant process

- So how do we pick a table?
- Suppose there are n_j existing people at table T_j, and suppose we are the $n + 1$-th observation.
- Pick table T_j with probability $\frac{n_j}{n+\alpha}$, otherwise pick an empty table with probability $\frac{\alpha}{n+\alpha}$.

Chinese restaurant process

- So how do we pick a table?
- Suppose there are n_j existing people at table T_j, and suppose we are the $n+1$-th observation.
- Pick table T_j with probability $\frac{n_j}{n+\alpha}$, otherwise pick an empty table with probability $\frac{\alpha}{n+\alpha}$.
- Note: can prove that expected number of tables filled is $O(\alpha \log n)$.

How do we learn the model?

How do we learn the model?

Two major approaches:

How do we learn the model?

Two major approaches:

- Markov chain Monte Carlo (e.g. Gibbs sampling)

How do we learn the model?

Two major approaches:

- Markov chain Monte Carlo (e.g. Gibbs sampling)
- Variational methods

How do we learn the model?

Two major approaches:

- Markov chain Monte Carlo (e.g. Gibbs sampling)
- Variational methods

How do we learn the model?

Two major approaches:

- Markov chain Monte Carlo (e.g. Gibbs sampling)
- Variational methods

`data-microscopes` only implements MCMC (for now!).

Gibbs sampling for DPMM

Gibbs sampling for DPMM

Problem statement

Given n data points $\mathcal{Y} = (Y_1, ..., Y_n)$, our goal is to learn the distribution $p(\mathcal{C}|\mathcal{Y})$, where $\mathcal{C}$ is the clustering (assignment vector) of $\mathcal{Y}$.

Gibbs sampling for DPMM

Problem statement

Given n data points $\mathcal{Y} = (Y_1, ..., Y_n)$, our goal is to learn the distribution $p(\mathcal{C}|\mathcal{Y})$, where $\mathcal{C}$ is the clustering (assignment vector) of $\mathcal{Y}$.

Note: when we say "learn the distribution" we mean draw (independent) samples from.

Gibbs sampling for DPMM

Due to time constraints, please take on faith the following assertions:

Gibbs sampling for DPMM

Due to time constraints, please take on faith the following assertions:

- An exact analytical solution for $p(\mathcal{C}|\mathcal{Y})$ is not readily available.

Gibbs sampling for DPMM

Due to time constraints, please take on faith the following assertions:

- An exact analytical solution for $p(\mathcal{C}|\mathcal{Y})$ is not readily available.
- Sampling $c_i^{(t)} \leftarrow p(c_i|\mathcal{C}_{\neg i}^{(t-1)}, \mathcal{Y}),\ i = 1, ..., n$ over and over (and over) will *eventually* get us $p(\mathcal{C}|\mathcal{Y})$.

Gibbs sampling for DPMM

Due to time constraints, please take on faith the following assertions:

- An exact analytical solution for $p(\mathcal{C}|\mathcal{Y})$ is not readily available.
- Sampling $c_i^{(t)} \leftarrow p(c_i|\mathcal{C}_{\neg i}^{(t-1)}, \mathcal{Y}),\ i = 1, ..., n$ over and over (and over) will *eventually* get us $p(\mathcal{C}|\mathcal{Y})$.
- The above strategy is called *Gibbs sampling*.

Gibbs sampling for DPMM

To Gibbs sample, all we need to do is derive $p(c_i{=}k|\mathcal{C}_{\neg i}, \mathcal{Y})$.

Gibbs sampling for DPMM

To Gibbs sample, all we need to do is derive $p(c_i{=}k|\mathcal{C}_{\neg i}, \mathcal{Y})$.

$$\begin{aligned} p(c_i{=}k|\mathcal{C}_{\neg i}, \mathcal{Y}) &\propto p(c_i{=}k, \mathcal{C}_{\neg i}, \mathcal{Y}) \\ &= p(c_i{=}k|\mathcal{C}_{\neg i}) p(Y_i|\mathcal{Y}^{(k)}) \\ &= p(c_i{=}k|\mathcal{C}_{\neg i}) \int_\theta p(Y_i|\theta) p(\theta|\mathcal{Y}^{(k)}) \, d\theta \end{aligned}$$

Gibbs sampling for DPMM

To Gibbs sample, all we need to do is derive $p(c_i{=}k|\mathcal{C}_{\neg i}, \mathcal{Y})$.

$$\begin{aligned} p(c_i{=}k|\mathcal{C}_{\neg i}, \mathcal{Y}) &\propto p(c_i{=}k, \mathcal{C}_{\neg i}, \mathcal{Y}) \\ &= p(c_i{=}k|\mathcal{C}_{\neg i}) p(Y_i|\mathcal{Y}^{(k)}) \\ &= p(c_i{=}k|\mathcal{C}_{\neg i}) \int_\theta p(Y_i|\theta) p(\theta|\mathcal{Y}^{(k)})\, d\theta \end{aligned}$$

The $p(c_i{=}k|\mathcal{C}_{\neg i})$ term is easy to calculate given the CRP interpretation!

Gibbs sampling for DPMM

To Gibbs sample, all we need to do is derive $p(c_i{=}k|\mathcal{C}_{\neg i}, \mathcal{Y})$.

$$\begin{aligned} p(c_i{=}k|\mathcal{C}_{\neg i}, \mathcal{Y}) &\propto p(c_i{=}k, \mathcal{C}_{\neg i}, \mathcal{Y}) \\ &= p(c_i{=}k|\mathcal{C}_{\neg i}) p(Y_i|\mathcal{Y}^{(k)}) \\ &= p(c_i{=}k|\mathcal{C}_{\neg i}) \int_\theta p(Y_i|\theta) p(\theta|\mathcal{Y}^{(k)}) \, d\theta \end{aligned}$$

The $p(c_i{=}k|\mathcal{C}_{\neg i})$ term is easy to calculate given the CRP interpretation!

If we pick H and F nicely, the integral on the RHS has an analytical solution!

Gibbs sampling for DPMM

Binary $Y_i \in \{0,1\}^D$ case, where H is the Beta distribution and F is Bernoulli, the Gibbs sampler distribution $p(c_i{=}k|\mathcal{C}_{\neg i}, \mathcal{Y})$ is (proportional to):

Gibbs sampling for DPMM

Binary $Y_i \in \{0,1\}^D$ case, where H is the Beta distribution and F is Bernoulli, the Gibbs sampler distribution $p(c_i{=}k|\mathcal{C}_{\neg i}, \mathcal{Y})$ is (proportional to):

$$\frac{|\mathcal{Y}^{(k)}_{\neg i}|}{n-1+\alpha} \prod_{d=1}^{D} \frac{\left(\beta + \sum_{y_k \in \mathcal{Y}^{(k)}_{\neg i}} y_k^{(d)}\right)^{y_i^{(d)}} \left(\gamma + |\mathcal{Y}^{(k)}_{\neg i}| - \sum_{y_k \in \mathcal{Y}^{(k)}_{\neg i}} y_k^{(d)}\right)^{(1-y_i^{(d)})}}{\beta + \gamma + |\mathcal{Y}^{(k)}_{\neg i}|}$$

when k is an existing cluster and

Gibbs sampling for DPMM

Binary $Y_i \in \{0,1\}^D$ case, where H is the Beta distribution and F is Bernoulli, the Gibbs sampler distribution $p(c_i{=}k|\mathcal{C}_{\neg i}, \mathcal{Y})$ is (proportional to):

$$\frac{|\mathcal{Y}_{\neg i}^{(k)}|}{n-1+\alpha} \prod_{d=1}^{D} \frac{\left(\beta + \sum_{y_k \in \mathcal{Y}_{\neg i}^{(k)}} y_k^{(d)}\right)^{y_i^{(d)}} \left(\gamma + |\mathcal{Y}_{\neg i}^{(k)}| - \sum_{y_k \in \mathcal{Y}_{\neg i}^{(k)}} y_k^{(d)}\right)^{(1-y_i^{(d)})}}{\beta + \gamma + |\mathcal{Y}_{\neg i}^{(k)}|}$$

when k is an existing cluster and

$$\frac{\alpha}{n-1+\alpha} \prod_{d=1}^{D} \left(\frac{\beta}{\beta+\gamma}\right)^{y_i^{(d)}} \left(\frac{\gamma}{\beta+\gamma}\right)^{1-y_i^{(d)}}$$

when k is a new cluster.

Using `data-microscopes`

Using `data-microscopes`

`data-microscopes` implements what we talked about (and more!)

Using `data-microscopes`

`data-microscopes` implements what we talked about (and more!)

Now let's see the library in action!