# E6892 Bayesian Models for Machine Learning

## Columbia University, Fall 2015

## Lecture 10, 11/19/2015

Instructor: John Paisley

Mixture models with Dirichlet priors

- Review: We have data $X = \{x_1, \ldots, x_n\}$. We will assume $x \in \mathbb{R}^d$, but the following discussion doesn't require it. We want to use a mixture model to partition the data into clusters and learn the statistical properties of each cluster.

- Mixture model: Let $p(x|\theta)$ be a probability distribution on the support of $x$, for example, a distribution on $\mathbb{R}^d$. Let $p(\theta)$ be a prior on $\theta$. A mixture model can be used to generate data as follows:

  - Model: $c_i \sim \text{Discrete}(\pi), \qquad x_i \sim p(x|\theta_{c_i})$

  - Priors: $\pi \sim \text{Dirichlet}(\underbrace{\alpha, \ldots, \alpha}_{K \text{ times}}), \qquad \theta_j \overset{iid}{\sim} p(\theta)$

- Gaussian mixture model (GMM): Last time we saw how the GMM results from $\theta_j = \{\mu_j, \Lambda_j\}$, $p(x|\theta_j) = \text{Normal}(\mu_j, \Lambda_j^{-1})$ and $p(\theta) = \text{Normal} - \text{Wishart}$.

- A key question is, how many clusters are there? Up to now we've been presetting $K$ to a value and assuming we know what that value should be. While one approach is to do cross-validation, the approach we will discuss next uses Bayesian nonparametrics.

- A Bayesian nonparametric (BNP) prior is a prior distribution like any other, except it involves "infinity" in a way that requires extra theoretical analysis. We won't discuss the theory, but only be interested in the practical aspects of a BNP prior called a "Dirichlet process."

- Dirichlet processes (high-level): Imagine we used the prior

$$\pi \sim \text{Dirichlet}(\underbrace{\tfrac{\alpha}{K}, \ldots, \tfrac{\alpha}{K}}_{K \text{ times}}) \tag{1}$$

  The only difference is that we divide each parameter $\alpha$ (some non-negative, preset number) by the dimensionality of $\pi$ ($K$). We then have $K$ possible clusters as before, but we let $K \to \infty$. The result is called a Dirichlet process.

- Again, there are two aspects to thinking about this:

    1. One is theoretical, in which it's necessary to make sure things remain well-defined and where we want to understand what $\pi$ looks like in this case, etc. There is no concern about the fact that we have an infinite number of $\theta_j$ sitting around, and that $\pi$ is infinite dimensional, so we can't actually do this in practice.

    2. Thus, the second aspect is practical. This essentially addresses the question of how we can do inference for a model where there are an infinite number of parameters. We clearly can't keep an infinite number of things in memory, so this issue boils down to finding equivalent representations that we can use that don't require this infinite number of objects.

- We only focus on the practical and discuss an equivalent representation. Proving equivalence is itself a theoretical question, so #1 and #2 aren't mutually exclusive. We'll give some hints as well about the theoretical ideas as it relates to the Gibbs sampling algorithm we derive next.

- We won't discuss the theory in depth, but the following can help with some intuition about what $\pi$ looks like as $K \to \infty$ and why it is useful.

    1. First, things remain well-defined. Second, the number of dimensions of $\pi$ that have probability greater than some small number will be small. Virtually all dimensions will have vanishingly small probabilities and technically an infinite number will have probability equal to zero. Therefore, even though there are an infinite number of clusters, for a finite amount of data only a small number (e.g., $10$) will actually be used.

    2. Since the number is variable and learned as part of inference, the Dirichlet process in some sense uncovers the "correct" number of clusters. (The use of "correct" is not at all part of the theory!) The posterior distribution will return a number of "occupied" clusters that is dependent on the data set used.

    3. Ultimately, the *theory* is only used to argue why this prior will be *useful* for our purposes of not presetting $K$, and not why it's the "correct" way to learn $K$.

    4. Dirichlet processes are sometimes referred to as sparsity-promoting priors for this reason.

- Again, our concern is practical, so we are interested in how to infer $(\pi, \theta_1, \ldots, \theta_K, c_1, \ldots, c_n)$ as $K \to \infty$.

- We will show how marginalization (of $\pi$) can save the day in an MCMC Gibbs sampling setup. Before doing this, notice the difference in advance:

    - In EM, we used "de-marginalization" to help. That is, we wanted to maximize a marginal distribution over some variables and we did this by adding new variables such that they produced the correct marginal.

    - In this lecture, we will have difficulty doing inference over all the model variables, so we will integrate out one of them *permanently* and only sample the remaining ones.

    - Generally speaking, while EM uses one direction, MCMC finds both directions useful.

- Before we derive the generic Gibbs sampling algorithm for the mixture model where $K \to \infty$, let's review the Gibbs sampler for the full model with $\pi$ from a finite Dirichlet prior with repeated parameter $\alpha/K$ (and so $K < \infty$).

- Recall the Gibbs sampling procedure (in the context of this model)

  1. For each iteration:

     (a) Sample each $c_i$ given the most recent $\pi$, $\theta$ and other $c_{-i}$

     (b) Sample each $\theta_i$ given the most recent $c$, $\pi$ and $\theta_{-i}$

     (c) Sample $\pi$ given the most recent $c$ and $\theta$

- We use the subscript $-i$ to indicate all variable indexes except for the $i$th one. We sample each variable from their conditional posteriors. These are:

**Gibbs sampling for the finite Dirichlet mixture model**

- Sample $c_i$: The conditional posterior $p(c_i|\pi, \theta, c_{-i}, X)$ does not depend on $c_{-j}$ and $X_{-i}$. Thus

$$\begin{aligned} p(c_i = j|\pi, x_i, \theta) &\propto p(x_i|\theta, c_i = j)p(c_i = j|\pi) \\ &= \frac{p(x_i|\theta_j)\pi_j}{\sum_{\ell=1}^{K} p(x_i|\theta_\ell)\pi_\ell} \end{aligned} \tag{2}$$

  And so we calculate how likely each cluster is for the data point considered $x_i$, and pre-weight this be how likely the cluster is in general. This gives a $K$ dimensional vector that we normalize and then use as the parameter of a Discrete distribution to sample a new index value for $c_i$.

- Sample $\theta_j$ The conditional posterior $p(\theta_j|\theta_{-j}, c, \pi, X)$ only depends on the data in $X$ that has been assigned to cluster $j$. We can write this as,

$$\begin{aligned} p(\theta_j|X, c) &\propto p(X|\theta_j, c)p(\theta_j) \\ &\propto \left[\prod_{i=1}^{n} p(x_i|\theta_j)^{\mathbb{1}(c_i=j)}\right] p(\theta_j) \end{aligned} \tag{3}$$

  This is problem specific and the prior is often chosen to be conjugate to the likelihood so we can calculate the posterior in closed form and sample from it.

- Sample $\pi$: This is a simple Dirichlet prior—multinomial likelihood setting. The conditional posterior of $\pi$ only depends on $c$. Therefore,

$$\begin{aligned} p(\pi|c) &\propto p(c|\pi)p(pi) \\ &\propto \left[\prod_{i=1}^{n} p(c_i|\pi)\right] p(\pi) \\ &\propto \left[\prod_{i=1}^{n}\prod_{j=1}^{K} \pi_j^{\mathbb{1}(c_i=j)}\right]\left[\prod_{j=1}^{K} \pi_j^{\frac{\alpha}{K}-1}\right] \\ &\propto \prod_{j=1}^{K} \pi_j^{\frac{\alpha}{K}+\sum_{i=1}^{n}\mathbb{1}(c_i=j)-1} \end{aligned} \tag{4}$$

Define $n_j = \sum_{i=1}^{n} \mathbb{1}(c_i = j)$. Then the posterior is

$$p(\pi|c) = \text{Dirichlet}(\tfrac{\alpha}{K} + n_1, \ldots, \tfrac{\alpha}{K} + n_K) \tag{5}$$

- Now we need to address how to sample $\pi$ and each $\theta_j$ as $K \to \infty$. This will be done by integrating out $\pi$, which will produce a new method for sampling each $c_i$.

**Discussion on sampling $\theta$**

- We will integrate out $\pi$ in the next section. Therefore we want to sample each $\theta_j$ from its conditional posterior given that $\pi$ is integrated out. In this case, the conditional posterior distribution has exactly the same form as before

$$\begin{aligned} p(\theta_j|X, c, \theta_{-j}) &\propto p(X|\theta_j, c)p(\theta_j) \\ &\propto \left[\prod_{i=1}^{n} p(x_i|\theta_j)^{\mathbb{1}(c_i=j)}\right] p(\theta_j) \end{aligned} \tag{6}$$

- Therefore, the posterior distribution is calculated exactly the same and then $\theta_j$ can be sampled from its conditional posterior. However, now since there are and infinite number of $\theta_j$ (since $j \in \{1, 2, 3, \ldots\}$, we can't actually do this. We can break this process down into two categories:

  1. $n_j > 0$. There can only be a finite number of $j$ such that this is the case (since the amount of data is finite). Therefore, we definitely need to sample these indices from their conditional posterior.

  2. $n_j = 0$. There are an infinite number of these. We can't actually sample them, however, a crucial observation is that, for a $j$ such that $n_j = 0$, $\mathbb{1}(c_i = j) = 0$ for all $i$.

     - Therefore, $p(\theta_j|X, c, \theta_{-j}) \propto p(\theta_j)$ for these $j$. That is, the conditional posterior is equal to the prior (there is no data in cluster $j$, so there is no prior-to-posterior update!).
     - So even though we can't sample an infinite number of times from this distribution, we do *know* the distribution we would sample from *if we could*.
     - In this case, we would sample an infinite number of i.i.d. samples from the prior $p(\theta)$. This fact will come in handy later.

**Discussion on dealing with $\pi$ by integrating it out**

- The solution to working with an infinite dimensional $\pi$ is to integrate it out (or marginalize it) from the model. Notice that this is the opposite solution as EM, where we expand the model by adding more variables.

- In other words, before we were Gibbs sampling from $p(\pi, c, \theta|X)$. Now we want to Gibbs sample from

$$p(c, \theta|X) = \int p(\pi, c, \theta|X) d\pi \tag{7}$$

- We already showed how to sample $\theta$ in this model (exactly the same as before). Next we need to show how to sample each $c_i$. However, before doing that we briefly discuss what the difference between these two Gibbs samplers is in more detail.
    - Imagine we obtained a large set of samples of the set of model variables $\{\pi^{(t)}, c^{(t)}, \theta^{(t)}\}$ using Gibbs sampling of $p(\pi, c, \theta|X)$. The samples are this triplet for many (say 1000) different values of $t$ as discussed more generally in an earlier lecture.
    - Then, imagine that we threw away each $\pi^{(t)}$ and only kept $\{c^{(t)}, \theta^{(t)}\}$
    - Statistically, the set of "thinned" samples we would have would be statistically identical to if we sampled this pair from the marginalized model $p(c, \theta|X)$.
    - Therefore, if we directly sample this pair of variables from the marginal $p(c, \theta|X)$, we are not sampling from a different model, but only sampling a subset of variables from the same original model.
    - Often (such as in this case) this subset of variables is sufficient to give us information about all we might care about—after all, $\pi^{(t)}$ would only look like the histogram of $c^{(t)}$.

  Sampling $c_i$

- We go back to a finite $K$, do some calculations, and then let $K \to \infty$.
- Since we integrate out $\pi$, sampling $c_i$ turns out to be different from before. By Bayes rule,

$$p(c_i = j|X, \theta, c_{-i}) \;\propto\; \underbrace{p(X|c_i = j, \theta)}_{= p(x_i|\theta_j) \text{ as before}} \underbrace{p(c_i = j|c_{-i})}_{= ?} \tag{8}$$

- The derivation of $p(c_i = j|c_{-i})$ is the main new calculation we need. The result is very interpretable, but requires the following derivation,

$$\begin{aligned}
p(c_i = j|c_{-i}) &= \int p(c_i = j|\pi)p(\pi|c_{-i})d\pi \\
&= \int \pi_j \cdot \text{Dirichlet}(\pi|\tfrac{\alpha}{K} + n_1^{(-i)}, \ldots, \tfrac{\alpha}{K} + n_K^{(-i)})d\pi \\
&\quad \longrightarrow \left(n_j^{(-i)} = \sum_{s \neq i} \mathbb{1}(c_s = j)\right) \\
&= \int \pi_j \frac{\Gamma(\alpha + n - 1)}{\prod_\ell \Gamma(\frac{\alpha}{K} + n_\ell^{(-i)})} \prod_{\ell=1}^{K} \pi_\ell^{\frac{\alpha}{K} + n_\ell^{(-i)} - 1} d\pi \\
&= \frac{\Gamma(\alpha + n - 1)}{\prod_\ell \Gamma(\frac{\alpha}{K} + n_\ell^{(-i)})} \underbrace{\int \pi_j^{\frac{\alpha}{K} + n_j^{(-i)} + 1 - 1} \prod_{\ell \neq j} \pi_\ell^{\frac{\alpha}{K} + n_\ell^{(-i)} - 1} d\pi}_{= \frac{\Gamma\left(\frac{\alpha}{K} + n_j^{(-i)} + 1\right) \prod_{\ell \neq j} \Gamma\left(\frac{\alpha}{K} + n_\ell^{(-i)}\right)}{\Gamma(\alpha + n)}} \\
&= \frac{\Gamma(\alpha + n - 1)}{\Gamma(\alpha + n)} \frac{\Gamma(\frac{\alpha}{K} + n_j^{(-i)} + 1)}{\Gamma(\frac{\alpha}{K} + n_j^{(-i)})}
\end{aligned} \tag{9}$$

- We solved the integral by recognizing that this is the normalizing constant for a Dirichlet distribution with the parameterization indicated in the equation.

- To complete the calculation, we use the property that $\Gamma(y) = (y-1)\Gamma(y-1)$. Applying this equality to $\Gamma(\alpha + n)$ and $\Gamma(\frac{\alpha}{K} + n_j^{(-i)} + 1)$ in the equation above, we find that

$$p(c_i = j | c_{-j}) \;=\; \frac{\frac{\alpha}{K} + n_j^{(-i)}}{\alpha + n - 1} \tag{10}$$

- As a result,

$$p(c_i = j | X, \theta, c_{-i}) \;\propto\; p(x_i|\theta_j) \left( \frac{\frac{\alpha}{K} + n_j^{(-i)}}{\alpha + n - 1} \right) \tag{11}$$

Letting $K \to \infty$

- The problem is that as $K \to \infty$, more and more $n_j^{(-i)} = 0$, which poses a problem. When $n_j^{(-i)} > 0$, there's no problem. We discuss these two cases below.

- Case $n_j^{(-i)} > 0$: In the limit $K \to \infty$,

$$p(c_i = j | X, \theta, c_{-i}) \;\propto\; p(x_i|\theta_j) \frac{n_j^{(-i)}}{\alpha + n - 1} \tag{12}$$

- Case $n_j^{(-i)} = 0$: In the limit $K \to \infty$,

$$p(c_i = j | X, \theta, c_{-i}) \;\propto\; 0 \tag{13}$$

- Does this mean that $c_i$ is limited to the existing cluster? No! Instead of asking the probability $c_i = j$ for a particular $j$ in the case where $n_j^{(-i)} = 0$, we ask

$$\begin{aligned} p(c_i = new | X, \theta, c_{-i}) \;&=\; \sum_{j\,:\,n_j^{(-i)}=0} p(c_i = j | X, \theta, c_{-i}) && (14) \\ &\propto\; \lim_{K\to\infty} \sum_{j\,:\,n_j^{(-i)}=0} p(x_i|\theta_j) \frac{\alpha/K}{\alpha + n - 1} && (15) \\ &\propto\; \lim_{K\to\infty} \frac{\alpha}{\alpha + n - 1} \sum_{j\,:\,n_j^{(-i)}=0} \frac{p(x_i|\theta_j)}{K} && (16) \end{aligned}$$

- What is this last limit? Remember that for all of these infinite number of $j$ such that $n_j^{(-i)} = 0$, $\theta_j \overset{iid}{\sim} p(\theta)$. Skipping to the end, we can say that as $K \to \infty$ this *Monte Carlo integral* converges to the true integral it approximates when $K$ is finite. Therefore,

$$\lim_{K\to\infty} \sum_{j\,:\,n_j^{(-i)}=0} \frac{p(x_i|\theta_j)}{K} \;=\; \mathbb{E}[p(x_i|\theta)] \;=\; \int p(x_i|\theta) p(\theta) d\theta \tag{17}$$

- To summarize, we have shown that

$$c_i = \begin{cases} j & \text{w.p. } \propto p(x_i|\theta_j)\frac{n_j^{(-i)}}{\alpha+n-1} \text{ if } n_j^{(-i)} > 0 \\ \text{a new} & \\ \text{index} & \text{w.p. } \propto \frac{\alpha}{\alpha+n-1}\int p(x_i|\theta)p(\theta)d\theta \end{cases}$$

- Finally, if we pick a new index, the question is, what index did we pick? The answer is "who cares"!? In the end, there is a 1-to-1 mapping between the new index $j'$ and new variable $\theta_{j'}$. So we only need to pick the new variable. This is not trivial to prove how to do, but the solution is, if $c_i = j'$ and $j'$ is a new index (i.e., $n_{j'}^{(-i)} = 0$), then

$$\theta_{j'} \mid \{c_i = j'\} \sim p(\theta|x_i) \tag{18}$$

- This gives a general algorithm for sampling from the posterior of a Dirichlet process mixture model. Specific problems that need to be addressed depending on what type of mixture it is (e.g., Gaussian mixture) are calculating the posterior of $\theta$ given $x$ and the marginal $\int p(x|\theta)p(\theta)d\theta$.

- Below, we give the general outline of the algorithm, and then the two specific equations used for a Gaussian mixture model.

---

**A marginalized sampling method for the Dirichlet process mixture model**

- Initialize in some way, e.g., set all $c_i = 1$ for $i = 1, \ldots, n$, and sample $\theta_1 \sim p(\theta)$.

- At iteration $t$, re-index clusters to go from 1 to $K_{t-1}$, where $K_{t-1}$ is the number of occupied clusters after the previous iteration. Sample all variables below using the most recent values of the other variables.

  1. For $i = 1, \ldots, n$

     (a) For all $j$ such that $n_j^{(-i)} > 0$, set

$$\hat{\phi}_i(j) = p(x_i|\theta_j)n_j^{(-i)}/(\alpha + n - 1)$$

     (b) For a new value $j'$, set

$$\hat{\phi}_i(j') = \frac{\alpha}{\alpha + n - 1}\int p(x_i|\theta)p(\theta)d\theta$$

     (c) Normalize $\hat{\phi}_i$ and sample the index $c_i$ from a discrete distribution with this parameter.

     (d) If $c_i = j'$, generate $\theta_{j'} \sim p(\theta|x_i)$

  2. For $j = 1, \ldots, K_t$ generate

$$\theta_j \sim p(\theta|\{x_i : c_i = j\})$$

  ($K_t$ = # non-zero clusters after completing Step 1)

---

Comments:

1. There is a lot of bookkeeping with this algorithm that can very easily lead to coding issues.
2. $n_j^{(-i)}$ needs to be updated after each $c_i$ is sampled. This means that a new cluster is possibly created, or destroyed, after each $i$ in Step 1. It is created if $c_i$ picks a new $j'$.
3. A cluster is destroyed if the *previous* value of $c_i$ was the *only* instance of this value in the data set. When we check for the new value of $c_i$, we erase the value previously assigned to it, in which case the cluster no longer exists because $x_i$ was the only observation in that cluster. Then, if $c_i$ joins an existing clusters, the number of clusters is reduced by one. If $c_i$ still creates a new cluster $j'$, a new value of $\theta_{j'}$ must be generated and so the $\theta$ variable is still changed.
4. $n_j^{(-1)}$ needs to always be up to date. When a new $c_i$ is sampled, these counts need to be updates.
5. Therefore the "new value $j'$" in Step 1b can be changing (e.g., for one $i$, $j' = 10$, for the next $i$ perhaps $j' = 11$ if the previous $c_i$ picked $j'$ when $j' = 10$)
6. Keeping indexing correct is also crucial to getting things to work. After each iteration, it would be good to re-index clusters to start from one and increase by one.

**Dirichlet process Gaussian mixture model**

- For this mixture model, $p(\theta) = p(\mu|\Lambda)p(\Lambda)$, where

$$\Lambda \sim \text{Wishart}(a, B), \qquad \mu|\Lambda \sim \text{Normal}(m, (c\Lambda)^{-1}) \tag{19}$$

- In variational inference we didn't need this linked prior because we were using an approximation that allowed for tractable calculations. With Gibbs sampling, we need to be more rigorously correct and so in order to calculate the marginal distribution $p(x)$, we need to connect the priors.
- Imagine that $x_1, \ldots, x_s$ were assigned to cluster $j$, then

$$p(\mu_j, \Lambda_j|x_1, \ldots, x_s) = \text{Normal}(\mu_j|m_j', (c_j'\Lambda_j)^{-1})\text{Wishart}(\Lambda_j|a_j', B_j') \tag{20}$$

where

$$m_j' = \frac{c}{s+c}m + \frac{1}{s+c}\sum_{i=1}^{s} x_i, \qquad c_j' = s + c$$

$$a_j' = a + s, \qquad B_j' = B + \sum_{i=1}^{s}(x_i - \bar{x})(x_i - \bar{x})^T + \frac{s}{as+1}(\bar{x} - m)(\bar{x} - m)^T$$

We define $\bar{x} = \frac{1}{s}\sum_{i=1}^{s} x_i$.

- When we want to sample a new cluster in which $x_i$ is the only observation, we can also use this posterior distribution. We just don't have any sums and $s = 1$. Notice that one of the terms in $B_j'$ ends up equaling zero.

- Next we need to calculate the marginal distribution under the prior

$$p(x) = \int \int p(x|\mu, \Lambda) p(\mu, \Lambda) d\mu d\Lambda \tag{21}$$

- Again, the prior $p(\mu, \Lambda) = p(\mu|\Lambda)p(\Lambda)$ where

$$p(\mu|\Lambda) = \text{Normal}(m, (c\Lambda)^{-1}), \qquad p(\Lambda) = \text{Wishart}(a, B)$$

- Using this prior and a likelihood $p(x|\mu, \Lambda) = \text{Normal}(\mu, \Lambda^{-1})$, the marginal is

$$p(x) = \left(\frac{c}{\pi(1+c)}\right)^{\frac{d}{2}} \frac{|B + \frac{c}{1+c}(x-m)(x-m)^T|^{-\frac{a+1}{2}}}{|B|^{-\frac{a}{2}}} \frac{\Gamma_d\left(\frac{a+1}{2}\right)}{\Gamma_d\left(\frac{a}{2}\right)} \tag{22}$$

- And where

$$\Gamma_d(y) = \pi^{\frac{d(d-1)}{4}} \prod_{j=1}^{d} \Gamma\left(y + \frac{1-j}{2}\right)$$

- Again, $d$ is the dimensionality of $x$.

- When implementing this, there might be scaling issues. For example, in the fraction $x_1/x_2$, the terms $x_1$ and $x_2$ might be too large for a computer (i.e., "rounded up" to $\infty$), but their fraction is something "reasonably sized". This is particularly relevant for the fraction

$$\frac{\Gamma_d\left(\frac{a+1}{2}\right)}{\Gamma_d\left(\frac{a}{2}\right)}$$

- An implementation trick is to represent this as

$$\frac{\Gamma_d\left(\frac{a+1}{2}\right)}{\Gamma_d\left(\frac{a}{2}\right)} = \mathrm{e}^{\sum_{j=1}^{d}\left[\ln\Gamma\left(\frac{a+1}{2}+\frac{1-j}{2}\right)-\ln\Gamma\left(\frac{a}{2}+\frac{1-j}{2}\right)\right]}$$

- By first calculating the value in the exponent, we can work with smaller terms (because of the natural log) and they can cancel each other such that the sum is small. The exponent then returns this final value to the correct "space". This trick is often useful when implementing machine learning algorithms that run into scaling issues (when we know that they shouldn't).