Advanced Machine Learning **Fall 2020**

Lecture 11: Non-parametric Bayesian methods

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

Disclaimer: *These notes are adapted from ETH's Advanced Machine Learning Course and "Machine Learning: a Probabilistic Perspective" book.*

11.1 Model Selection

We start our discussion by modeling the posterior distribution:

$$
\begin{aligned}
&p(\theta|x) = e^{-\beta w(\theta,x)} \\
&w(\theta,x) = R(\theta,x) - F(x) \\
&F(x) = -\frac{1}{\beta}\log\int e^{-\beta R(\theta,x)}d\theta \qquad \text{(Normalization term for } p(\theta|x))
\end{aligned}
$$

Next, we can compute the validation error on training set x' and validation set x'':

$$
\begin{aligned}
&\mathbb{E}_{\theta|x'}\Big[-\log p(\theta|x'')\Big] \\
&= \mathbb{E}_{\theta|x'}\Big[\beta w(\theta,x'')\Big] \\
&= \beta\Big(\underbrace{\mathbb{E}_{\theta|x'}\Big[R(\theta,x'')\Big]}_{Loss}\ \underbrace{-F(x'')}_{\text{Free Energy}}\Big) \qquad \text{(Note that } F(x'') \text{ does not depend on } \theta \text{ since we integrate it out)}
\end{aligned}
$$

The problem, in practice, is that people just minimize the loss while completely ignoring the free energy. This is the draw back of using the validation error for model selection.

To overcome this limitation, we can perform "posterior selection":

$$
\begin{aligned}
&\min_{p(.|.)} \mathbb{E}_{\theta|x'}\Big[-\log p(\theta|x'')\Big] \geq \min_{p(.|.)} -\log \mathbb{E}_{\theta|x'}\Big[p(\theta|x'')\Big] \qquad \text{(Jensen's inequality)} \\
&= \min_{p(.|.)} -\log\Big(\int p(\theta|x')p(\theta|x'')d\theta\Big) \\
&= -\max_{p(.|.)} \log\Big(\underbrace{\int p(\theta|x')p(\theta|x'')d\theta}_{\text{Probability Kernel } k(x',x'')}\Big)
\end{aligned}
$$

A few remarks on this:

- The kernel $k(x', x'')$ measures the "agreement" of the two posteriors.
- This strategy chooses a posterior that is concentrated (peaked) and agrees between x' and x''
- When we apply Jensen's inequality we have no guarantee that the posteriors (the one of the original optimization problem and the one from Jensen's) are similar.
- Maximizing the $\mathbb{E}_{x',x''}\Big[\log k(x', x'')\Big]$ is a metric concept, since the kernel captures the similarity between x' and x''. Whereas minimizing $\mathbb{E}_{\theta|x'}\Big[R(\theta, x'')\Big]$ is a search strategy based on a partial order. The argument we make here is that the latter is more sensible to noise. (In the space of θ, two solutions might be far away from each other and still have the same cost value)

11.2 Dirichlet Processes

The principle problem with finite mixture models is how to choose the number of components **K**. However, in many cases, there is no well- defined number of clusters. It would be much better if we did not have to choose K at all. In this section, we discuss infinite mixture models, in which we do not impose any a priori bound on K. To do this, we will use a non-parametric prior based on the Dirichlet process (DP). This allows the number of clusters to grow as the amount of data increases.

11.2.1 Stick-breaking Construction

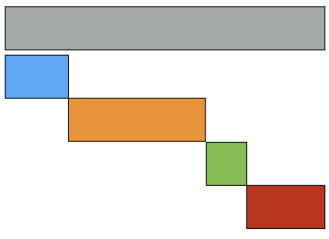
To build the mixture model we need to sample the cluster probabilities: $\rho_{1:K} \sim Dir(a_{1:K})$.
However, there are two problems:

- As $K \to \infty$ we cannot sample infinite points from $Dir()$.
- The sum of these probabilities should sum to 1.

We observe that:

$$\rho_{1:K} \sim Dir\Big(a_{1:K}\Big) \iff \rho_1 = Beta\Big(a_1, \sum_{k=1}^{K} a_k - a_1\Big) \perp\!\!\!\perp Dir\Big(a_{2:K}\Big)$$

Thus, we can sample from the Dirichlet using this technique:



- "Stick breaking"

$V_1 \sim \text{Beta}(a_1, a_2 + a_3 + a_4)$ $\quad \rho_1 = V_1$

$V_2 \sim \text{Beta}(a_2, a_3 + a_4)$ $\quad \rho_2 = (1 - V_1)V_2$

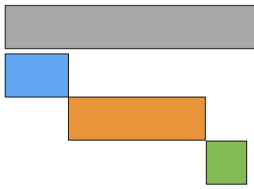
$V_3 \sim \text{Beta}(a_3, a_4)$ $\quad \rho_3 = (1 - V_1)(1 - V_2)V_3$

$$\rho_4 = 1 - \sum_{k=1}^{3} \rho_k$$

How can we generate $K \to \infty$ probabilities that strictly sum to one?
We fix the Betas in the stick-breaking process to $Beta(1, \alpha)$.

- **Dirichlet process stick-breaking**: $a_k = 1, b_k = \alpha > 0$
- Griffiths-Engen-McCloskey (**GEM**) distribution:

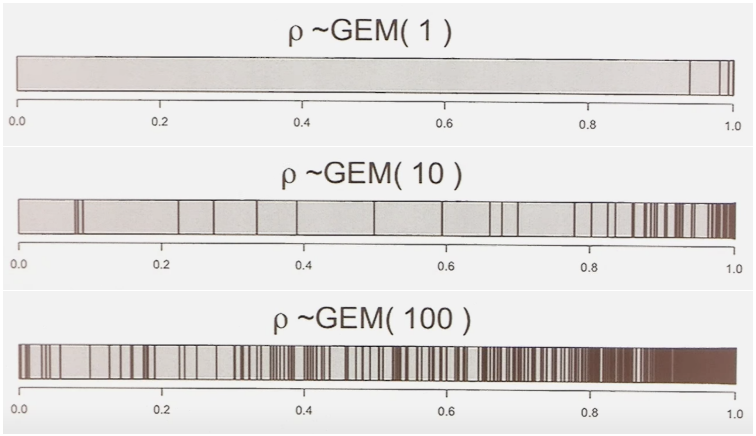
$$\rho = (\rho_1, \rho_2, \ldots) \sim \text{GEM}(\alpha)$$



$$V_1 \sim \text{Beta}(a_1, b_1) \qquad \rho_1 = V_1$$
$$V_2 \sim \text{Beta}(a_2, b_2) \qquad \rho_2 = (1 - V_1)V_2$$
$$\cdots \quad V_k \sim \text{Beta}(a_k, b_k) \qquad \rho_k = \left[\prod_{j=1}^{k-1}(1 - V_j)\right] V_k$$

[McCloskey 1965; Engen 1975; Patil and Taillie 1977; Ewens 1987; Sethuraman 1994; Ishwaran, James 2001]

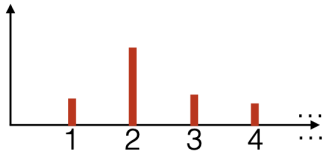
We finally obtain a distribution $\rho \sim GEM(\alpha)$ from which we can sample our infinite probabilities. Observe that α is an hyper-parameter that regulates how much stick the first draws will obtain. The larger the α the smaller the first piece of the stick will be.



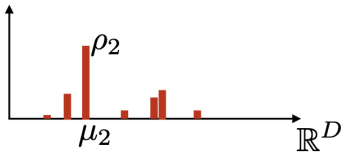
11.2.2 Mixture Model

We now have all we need to build our Dirichlet Process Mixture Model.

First, we draw an infinity of cluster probabilities from $\rho \sim GEM(\alpha)$



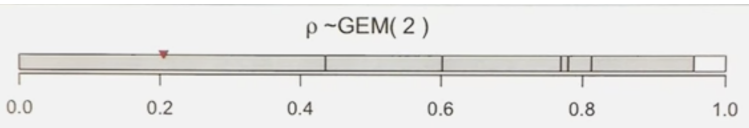
Second, we draw an infinity of $\mu_k \sim \mathcal{N}\Big(\mu_0, \Sigma_0\Big), k = 1, 2 \ldots$ and we assign each u_k to the respective probability ρ.



The resulting distribution is $G = \sum_{k=1}^{\infty} \rho_k \delta \mu_k = DP\Big(\alpha, \mathcal{N}(\mu_0, \Sigma_0)\Big)$. We have successfully constructed a Dirichlet Process.

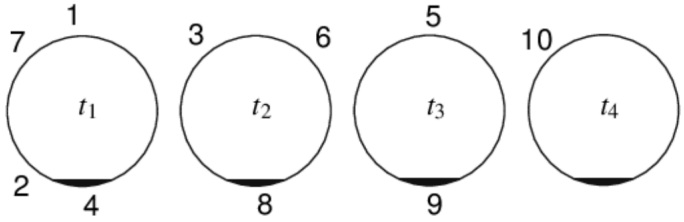
To generate data from this model, we can sample $z_n \sim Categorical(\rho)$ and $\mu_n = \mu_{z_n}$ i.e. $\mu_n \sim G$. Next, we sample the data point from $x_n \sim \mathcal{N}(\mu_n, \Sigma_0)$.

However, this is unfeasible in practice because we need to draw an infinity of ρ. The key idea to solve this problem, is to draw the ρ on demand. We can draw from $GEM(2)$ with Uniform$(0, 1)$: if the sample we draw from the uniform distribution (red cursor in figure) is already covered we draw x_1 from the corresponding Gaussian with mean μ_1. If it is not covered (white space in figure) we continue the stick breaking process until it is covered.



11.2.3 Chinese Restaurant Process

Chinese restaurant process or CRP, is based on the seemingly infinite supply of tables at certain Chinese restaurants. The analogy is as follows: The tables are like clusters, and the customers are like observations. When a person enters the restaurant, he may choose to join an existing table with probability proportional to the number of people already sitting at this table (the $|\tau|$ term); otherwise, with a probability that diminishes as more people enter the room (due to the $\frac{1}{\alpha+n}$ term), he may choose to sit at a new table. The result is a distribution over partitions of the integers, which is like a distribution of customers to tables. The fact that currently occupied tables are more likely to get new customers is sometimes called the **rich get richer** phenomenon.



$$\pi_{[10]} = \{\{1,2,4,7\},\{3,6,8\},\{5,9\},\{10\}\}$$

$$P(\text{customer } n+1 \text{ joins table } \tau \mid \pi) = \begin{cases} \frac{|\tau|}{\alpha+n} & \text{if } \tau \in \pi \\ \frac{\alpha}{\alpha+n} & \text{otherwise} \end{cases}$$

The probability for a specific tables configuration is given by:

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

where $\alpha^{(n)}$ is the ascending factorial.

11.2.4 Exchangeability

Let $(X_1, X_2, \ldots)$ be a sequence of random variables. The sequence in exchangeable when, for every permutation π of $\mathcal{N}$, the random vectors:

$$(X_1, X_2, \ldots) \text{ and } (X_{\pi(1)}, X_{\pi(2)}, \ldots)$$

have the same distribution.

Theorem 11.1 *(De Finetti)*
Let $(X_1, X_2, \ldots)$ be an infinitely exchangeable sequence of random variables. Then , $\forall n$:

$$p(X_1, \ldots, X_n) = \int \Big(\prod_{i=1}^{n} p(x_i|G) \Big) dP(G)$$

for some random variable G.

In the case of i.i.d. random variables we would only have one G and the theorem would reduce to $p(X_1, \ldots, X_n) = \prod_{i=1}^{n} p(x_i)$

Notice that CRP is exchangeable $\implies$ we can apply De Finetti's Theorem. In the CRP's case, the Dirichlet Process is the random variable G of De Finetti's theorem. The intuition behind this is that the probability of a particular table configuration is given by a "voting" weighted on all the underlying distributions G from which the probabilities of the tables (remember the ρ's) are picked from.

11.2.5 Fitting

We can leverage exchangeability to fit our model: any point can be considered the last arrived.
Considering the case of CRP: for each observation, we remove the customer dish from the restaurant and resample as if they were the last to enter.

- Take a random guess initially.
- Unassign observation i.
- Compute $p(z_i|z_{-i}, x, \alpha, \mu)$ which represents the cluster assignment for element i.
- Update z_i by sampling from this distribution.
- Keep going.

$p(z_i|z_{-i}, x, \alpha, \mu)$ is computed as follows:

$$p(z_i = k|z_{-i}, \mathbf{x}, \alpha, \mu) \propto \underbrace{p(z_i = k|z_{-i}, \alpha)}_{Prior} \underbrace{p(x_i|\mu, z_i = k, z_{-i}, \mathbf{x_{-i}})}_{Likelihood}$$

The prior computation is simple (CRP):

$$p(z_i = k|z_{-i}, \alpha) = \begin{cases} \dfrac{N_{k,-i}}{\alpha + N - 1} & \text{for existing k} \\ \dfrac{\alpha}{\alpha + N - 1} & \text{otherwise} \end{cases}$$

Finally, for the likelihood we don't need to consider point in x that are not in cluster k:

$$p(x_i|\mu, z_i = k, z_{-i}, \mathbf{x_{-i}}) = \begin{cases} p(x_i|x_{-i,k}, \mu) & \text{for existing k} \\ p(x_i|\mu) & \text{otherwise} \end{cases}$$