CS281 Section #7 Notes, v 1.1

Note: These notes are more of a lecture review than a collection of exercises.

1 Information Theory

Working at Bell Labs, Shannon tried to formalize communication mathematically. "The fundamental problem is that of reproducing at one point either exactly or approximately a message selected at another point" (*A Mathematical Theory of Communication*, 1948). Think in terms of sending a telegram:

- **information source**: telegram writer, wants to send a message.
- **encoder**: telegram machine, translates the message into a **code** for transmission
- **channel**: electrical wire connecting telegram machines, (often noisy) transmission medium
- **decoder**: telegram machine, tries to recover original message from the codes received

1.1 Entropy and related concepts

If a random variable X has K possible outcomes (e.g. the weather can only be rainy or sunny or cloudy), and the probability distribution is denoted by $p(X)$, then **entropy** is given by

$$H(X) = -\sum_k p(X=k) \log p(X=k) = -\mathbb{E}\big[\log p(x)\big].$$

Exercise: which distribution over K outcomes maximizes entropy? **Solution**: uniform distribution.

Entropy in statistical physics measures the orderlessness of a system. Entropy in info theory measures the uncertainty of a distribution, and information is the reduction of uncertainty. For continuous RVs:

$$H(X) = -\int_x p(x) \log p(x).$$

Conditional entropy of X conditioned on Y measures the *expected* uncertainty over X after a observing Y. $H(X|Y=y)$ is the uncertainty of X given a particular $Y=y$ but $H(X|Y)$ is an average over all $Y=y$

$$H(X|Y) = \int_y p(x) H(X|Y=y) = -\int_x \int_y p(x,y) \log p(x|y) = -\mathbb{E}_{x,y} \log p(x|y)$$

Relative entropy, or **Kullback-Leibler divergence**, measures how different two distributions are

$$KL(p||q) = \int_x p(x) \log \frac{p(x)}{q(x)}.$$

Note that KL-divergence is not symmetric. That is, $KL(p||q) \neq KL(q||p)$. Another measure of the degree of difference between two distributions is the **Jensen-Shannon divergence**

$$JSD(p,q) = \frac{1}{2} KL(p||m) + \frac{1}{2} KL(q||m),$$

where $m = \frac{1}{2}(p+q)$.

Interpret $H(X)$ as the average number of bits needed to encode an outcome $x \sim p(X)$ using an optimal (minimum-length) coding scheme. **Cross entropy** $H_q(p)$ is the expected number of bits needed to encode outcomes $x \sim p(X)$ from distribution p using a scheme optimized for another distribution q

$$H(p,q) = H_q(p) = -\mathbb{E}_{p(X)}\big[\log q(X)\big] = H(p) + KL(p||q)$$

Now $KL(p||q)$ can be more concretely defined as the *increase in average bits* when switching from a coding scheme optimized for p to one optimized for q while continuing to send items from $p(X)$

Exercise: Show that the entropy of X is the average number of bits needed to encode an outcome under an optimal encoding scheme.
Solution: If you have K objects, you need $\log_2(K)$ bits to represent any one of the objects with a unique binary number. Using the optimal variable-length coding scheme (like Huffman Encoding), take the weighted average of the number of bits.

Mutual Information is a symmetric measurement that reflects how much information two distributions share

$$I(X;Y) = KL(p(X,Y)||p(X)p(Y)) = \int_x \int_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

As a consequence of Jensen's inequality, which states that $f(\mathbb{E}[x]) \leq \mathbb{E}[f(x)]$ for convex f, mutual information is always non-negative. The mutual information is the *difference in uncertainty* $H(X) - H(X|Y)$. This is the amount of uncertainty of X explained by observing Y.

1.2 Some concepts in information geometry

To approximate a distribution p with another distribution q subject to some constraints, we choose q from a family of distributions Q and "project" p onto Q one of two ways

- **Information projection** (reverse) from p to Q is to find the q that minimizes $KL(q||p)$.
- **Moment projection** (forward) from p to Q is to find the q that minimizes $KL(p||q)$.

2 Latent Variables, Mixture Models, and EM

2.1 Latent Variables

A **latent variable** is one that is not observed at training time, but that is still part of the data generation process. Let $\{x_n\}$ be a set of observed variables and $\{z_n\}$ be a set of latent variables with joint density $p(z,x)$. The inference problem is to compute $p(z|x)$. We can write

$$p(z|x) = \frac{p(z,x)}{p(x)}$$

but the **evidence** $p(x)$ can be intractable to compute

$$p(x) = \int p(z,x)dz$$

The evidence is needed to compute the conditional from the joint.

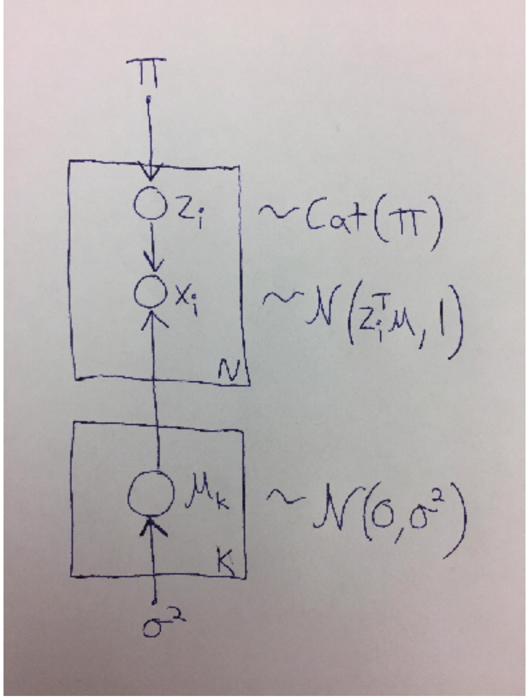
2.2 Bayesian Mixture of Gaussians

Note: Mixtures are much more general than Mixture of Gaussians

Consider a mixture of K unit-variance univariate Gaussians with means $\mu = \{\mu_1, ..., \mu_K\}$. The mean parameters are drawn independently from a common prior $p(\mu_k)$, for now assume $\mathcal{N}(0, \sigma^2)$ with given σ^2. To generate an observation x_i, first draw a cluster assignment z_i that indicates which latent cluster x_i comes from, drawn from $\text{Cat}(\pi)$. Then, draw x_i from $\mathcal{N}(z_i^\top \mu, 1)$ (z_i is one-hot and selects the proper μ_k).

$$\begin{aligned} \mu_k &\sim \mathcal{N}(0, \sigma^2) \\ z_i &\sim \text{Cat}(\pi) \\ x_i | z_i, \mu &\sim \mathcal{N}(z_i^\top \mu, 1) \end{aligned}$$

Exercise: Draw the DGM for this model
Solution:


π
z_i
$\sim \text{Cat}(\pi)$
x_i
$\sim \mathcal{N}(z_i^T \mu, 1)$
N
μ_k
$\sim \mathcal{N}(0, \sigma^2)$
K
σ^2

The complete-data likelihood (given the current $\{z_n\}$ one-hot assignments) is

$$\log p(\{x_n\}, \{z_n\} | \pi, \{\mu\}) = \sum_n \sum_k z_{nk} \big[\log \pi_k + \log p(x_n | \mu_k)\big]$$

Integrating out over z^N assignments to get $p(x)$ is intractable.

2.3 Expectation Maximization

Consider a distribution over z so $q_{nk} = p(z_n = k | x_n, \pi, \mu)$. Don't worry about where it comes from for the moment. Then we can write **expected likelihood**

$$\mathbb{E}_{z \sim p(z_n | x_n \ldots)} \big[\log p(x_n, z_n | \pi, \mu)\big] = \sum_n \sum_k q_{nk} \log \pi_k + q_{nk} \log p(x_n | \mu_k)$$

Notice that we now have soft assignments. Using coordinate ascent, we can do the following:

1. **Expectation**: Compute $q_{nk} = p(z_n = k|x_n, \pi_k, \mu_k)$ using fixed parameters (this answers where q_nk comes from... fixed parameters)
2. **Maximization**: Compute MLE of π and $\{\mu\}$ using q by maximizing the expected likelihood.

Initialize parameters randomly and then repeat the above steps until convergence of parameters.

For the **expectation** step, use

$$q_{nk} = p(z_n = k|x_n, \pi_k, \mu_k) \leftarrow \frac{\pi_k p(x_n|\mu_k)}{\sum_{k'} \pi_{k'} p(x_n|\mu_{k'})}$$

Notice that the expectation step is nothing different from what you would do if you were using an already-trained model to make predictions on test data.

For the **maximization** step, think of q_{nk} as "expected counts". Ask yourself what you would do in the fully-supervised Naive Bayes setting

$$\pi_k \leftarrow \frac{\sum_n q_{nk}}{\sum_n \sum_{k'} q_{nk'}}$$

The only difference is that all data points contribute to parameter updates for all latent classes. But, they are weighed by q_{nk}. Picture Q as an $N \times K$ grid. The numerator is a column sum for a particular k. Notice that the denominator sums to N because the row vector q_n sums to 1 across all K, and you sum across N.

Exercise: What is the $p(x|\mu_k)$ update?
Solution:

$$\mu_k \leftarrow \frac{\sum_n q_{nk}(x_n)}{\sum_n q_{nk}}$$

2.4 Understanding Intractability of $p(x)$

In class, we only talked about the cluster assignments z_n as latent. However, the parameters μ and π are also latent (that is, when we are performing inference on them and are not holding them fixed as we do in the Expectation step). We are generally interested in $p(\mu|x)$. How do we calculate that? Again we would need

$$p(\mu|x) = \frac{p(\mu, x)}{p(x)}$$

But we are stuck with dependence on $p(x)$. See why this is hard. Assuming 3 latent classes:

$$p(\mu_1, \mu_2, \mu_3|x) = \frac{p(\mu_1, \mu_2, \mu_3, x)}{\int_{\mu_1} \int_{\mu_2} \int_{\mu_3} p(\mu_1, \mu_2, \mu_3, x)}$$

The numerator is easy

$$p(\mu_1, \mu_2, \mu_3, x) = p(\mu_1)p(\mu_2)p(\mu_3) \prod_{i=1}^{N} p(x_i|\mu_1, \mu_2, \mu_3)$$

where each likelihood term marginalizes out z_i

$$p(x_i|\mu_1, \mu_2, \mu_3) = \sum_{k=1}^{K} \pi_k p(x_i|\mu_k)$$

But consider the denominator

$$p(x) = \int_{\mu_1} \int_{\mu_2} \int_{\mu_3} p(\mu_1)p(\mu_2)p(\mu_3) \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k p(x_i|\mu_k)$$

Bring the summation outside of the integral

$$p(x) = \sum_{z} \int p(\mu_1)p(\mu_2)p(\mu_3) \prod_{i=1}^{N} p(x_i|\mu_{z_i})$$

Decompose by partitioning data according to z

$$p(x) = \sum_{z} \prod_{k=1}^{3} \left(\int_{\mu,k} p(\mu_k) \prod_{\{i:z_i=k\}} p(x_i|\mu_k) \right)$$

Exercise: Is each term within the large parenthesis computable? How many different assignments of the data must we consider for the whole expression?
Solution: Each term within the parenthesis is a Normal-Normal conjugate pair and is computable. However, there are 3^N assignments to consider. This problem requires **approximate inference**.

3 References

1. CS281 Lectures on Info Theory and Mixture Models (2017), Sasha Rush

2. Bayesian Mixture Models and the Gibbs Sampler (2015), David M. Blei*

3. Variational Inference: A Review for Statisticians (2017), David M. Blei, Alp Kucukelbir, Jon D. McAuliffe*

* These notes borrow heavily from David Blei's tutorials. They are very good resources.