Overview

- First, brief overview of Expectation-Maximization algorithm.
  - In the lecture we were using Gaussian Mixture Model fitted with Maximum Likelihood (ML) estimation.
- Today, practice with the E-M algorithm in an image completion task.
- We will use Mixture of Bernoullis fitted with Maximum a posteriori (MAP) estimation.
  - Learning the parameters
  - Posterior inference
The Generative Model

- We’ll be working with the following generative model for data $D$
- Assume a datapoint $x$ is generated as follows:
  - Choose a cluster $z$ from $\{1, \ldots, K\}$ such that $p(z = k) = \pi_k$
  - Given $z$, sample $x$ from a probability distribution. (Earlier you saw Gaussian $\mathcal{N}(x|\mu_z, I)$, now we will work with Bernoulli($\theta_z$))
- Can also be written:
  $$p(z = k) = \pi_k$$
  $$p(x|z = k) = \mathcal{N}(x|\mu_k, I)/\text{Bernoulli}(\theta_k)$$
Maximum Likelihood with Latent Variables

- How should we choose the parameters $\{\pi_k, \mu_k\}_{k=1}^K$?
- Maximum likelihood principle: choose parameters to maximize likelihood of observed data
- We don’t observe the cluster assignments $z$, we only see the data $x$
- Given data $D = \{x^{(n)}\}_{n=1}^N$, choose parameters to maximize:

$$\log p(D) = \sum_{n=1}^N \log p(x^{(n)})$$

- We can find $p(x)$ by marginalizing out $z$:

$$p(x) = \sum_{k=1}^K p(z = k, x) = \sum_{k=1}^K p(z = k)p(x | z = k)$$
Log-likelihood derivatives

\[ \frac{\partial}{\partial \theta} \log p(x) = \frac{\partial}{\partial \theta} \log \sum_z p(x, z) \]
Log-likelihood derivatives

\[
\frac{\partial}{\partial \theta} \log p(x) = \frac{\partial}{\partial \theta} \log \sum_z p(x, z) \\
= \frac{\partial}{\partial \theta} \frac{\sum_z p(x, z)}{\sum_{z'} p(x, z')} 
\]
Log-likelihood derivatives

\[
\frac{\partial}{\partial \theta} \log p(x) = \frac{\partial}{\partial \theta} \log \sum_z p(x, z)
= \frac{\partial}{\partial \theta} \frac{\sum_z p(x, z)}{\sum_{z'} p(x, z')}
= \frac{\sum_z \frac{\partial}{\partial \theta} p(x, z)}{\sum_{z'} p(x, z')}
\]
Log-likelihood derivatives

\[
\frac{\partial}{\partial \theta} \log p(x) = \frac{\partial}{\partial \theta} \log \sum_z p(x, z)
\]

\[
= \frac{\frac{\partial}{\partial \theta} \sum_z p(x, z)}{\sum_{z'} p(x, z')}
\]

\[
= \frac{\sum_z \frac{\partial}{\partial \theta} p(x, z)}{\sum_{z'} p(x, z')}
\]

\[
= \frac{\sum_z p(x, z) \frac{\partial}{\partial \theta} \log p(x, z)}{\sum_{z'} p(x, z')}
\]
Log-likelihood derivatives

\[
\frac{\partial}{\partial \theta} \log p(x) = \frac{\partial}{\partial \theta} \log \sum_z p(x, z)
\]

\[
= \frac{\partial}{\partial \theta} \frac{\sum_z p(x, z)}{\sum_{z'} p(x, z')}
\]

\[
= \frac{\sum_z \frac{\partial}{\partial \theta} p(x, z)}{\sum_{z'} p(x, z')}
\]

\[
= \frac{\sum_z p(x, z) \frac{\partial}{\partial \theta} \log p(x, z)}{\sum_{z'} p(x, z')}
\]

\[
= \sum_z \left( \frac{p(x, z)}{\sum_{z'} p(x, z')} \frac{\partial}{\partial \theta} \log p(x, z) \right)
\]
Log-likelihood derivatives

\[
\frac{\partial}{\partial \theta} \log p(x) = \frac{\partial}{\partial \theta} \log \sum_z p(x, z)
\]

\[
= \frac{\partial}{\partial \theta} \frac{\sum_z p(x, z)}{\sum_{z'} p(x, z')}
\]

\[
= \frac{\sum_z \frac{\partial}{\partial \theta} p(x, z)}{\sum_{z'} p(x, z')}
\]

\[
= \frac{\sum_z p(x, z) \frac{\partial}{\partial \theta} \log p(x, z)}{\sum_{z'} p(x, z')}
\]

\[
= \sum_z \left( \frac{p(x, z)}{\sum_{z'} p(x, z')} \frac{\partial}{\partial \theta} \log p(x, z) \right)
\]

\[
= \sum_z p(z \mid x) \frac{\partial}{\partial \theta} \log p(x, z)
\]
The Expectation-Maximization algorithm alternates between two steps:

1. **E-step**: Compute the posterior probabilities $r_k^{(n)} = p(z^{(n)} = k | x^{(n)})$ given our current model - i.e. how much do we think a cluster is responsible for generating a datapoint.

2. **M-step**: Use the equations on the last slide to update the parameters, assuming $r_k^{(n)}$ are held fixed- change the parameters of each distribution to maximize the probability that it would generate the data it is currently responsible for.

$$
\frac{\partial}{\partial \theta} \log p(D) = \frac{\partial}{\partial \theta} \sum_{n=1}^{N} \sum_{k=1}^{K} p(z^{(n)} = k, x^{(n)})
$$

$$
= \sum_{i=1}^{N} \sum_{k=1}^{K} p(z^{(n)} = k | x^{(n)}) \frac{\partial}{\partial \theta} \log p(x^{(n)}, z^{(n)})
$$

$$
= \sum_{i=1}^{N} \sum_{k=1}^{K} r_k^{(i)} \left[ \frac{\partial}{\partial \theta} \log \Pr(z^{(i)} = k) + \frac{\partial}{\partial \theta} \log p(x^{(i)} | z^{(i)} = k) \right]
$$
A probabilistic model for the task of image completion.

We observe the top half of an image of a handwritten digit, we would like to predict what's in the bottom half.

Given these observations... ... you want to make these predictions

1 Source

Intro ML (UofT)
Our dataset is a set of $28 \times 28$ binary images represented as 784-dimensional binary vectors.

- $N = 60,000$, the number of training cases. The training cases are indexed by $i$.
- $D = 28 \times 28 = 784$, the dimension of each observation vector. The dimensions are indexed by $j$.

Conditioned on the latent variable $z = k$, each pixel $x_j$ is an independent Bernoulli random variable with parameter $\theta_{k,j}$:

$$p(x^{(i)} | z = k) = \prod_{j=1}^{D} p(x_{j}^{(i)} | z = k)$$

$$= \prod_{j=1}^{D} \theta_{k,j}^{x_{j}^{(i)}} (1 - \theta_{k,j})^{1-x_{j}^{(i)}}$$
This can be written out as the following generative process:

Sample $z$ from a multinomial distribution $\pi$.

For $j = 1, \ldots, D$:

Sample $x_j$ from a Bernoulli distribution with parameter $\theta_{k,j}$, where $k$ is the value of $z$.

It can also be written mathematically as:

$$z \sim \text{Multinomial}(\pi)$$

$$x_j \mid z = k \sim \text{Bernoulli}(\theta_{k,j})$$
Part 1: Learning the Parameters

- In the first step, we will learn the parameters of the model given the responsibilities (M-step of the E-M algorithm).
- We want to use the MAP criterion instead of maximum likelihood (ML) to fit the Mixture of Bernoullis model.
  - The only difference is that we add a prior probability term to the ML objective function in the M-step.
  - ML objective function:
    \[
    \sum_{i=1}^{N} \sum_{k=1}^{K} r_k^{(i)} \left[ \log \Pr(z^{(i)} = k) + \log p(x^{(i)} | z^{(i)} = k) \right]
    \]
  - MAP objective function:
    \[
    \sum_{i=1}^{N} \sum_{k=1}^{K} r_k^{(i)} \left[ \log \Pr(z^{(i)} = k) + \log p(x^{(i)} | z^{(i)} = k) \right] + \log p(\pi) + \log p(\Theta)
    \]
Part 1: Learning the Parameters (Prior Distribution)

- Use Beta distribution as the prior for $\Theta$: Every entry is drawn independently from a beta distribution with parameters $a$ and $b$:

$$p(\theta_{k,j}) \propto \theta_{k,j}^{a-1}(1 - \theta_{k,j})^{b-1}$$

- Use Dirichlet distribution as the prior over mixing proportions $\pi$:

$$p(\pi) \propto \pi_1^{a_1-1} \pi_2^{a_2-1} \cdots \pi_K^{a_K-1}.$$
Derive the M-step update rules for $\Theta$ and $\pi$ by setting the partial derivatives of the MAP objective function to zero.

\[
J(\theta, \pi) = \sum_{i=1}^{N} \sum_{k=1}^{K} r^{(i)}_{k} \left[ \log \Pr(z^{(i)} = k) + \log p(x^{(i)} | z^{(i)} = k) \right] \\
+ \log p(\pi) + \log p(\Theta)
\]

\[\pi_{k} \leftarrow \ldots\]

\[\theta_{k,j} \leftarrow \ldots\]
Part 1: Learning the Parameters

\[ J(\Theta, \pi) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_k^{(i)} \left[ \log \Pr(z^{(i)} = k) + \log p(x^{(i)} | z^{(i)} = k) \right] + \log p(\pi) + \log p(\Theta) \]

\[ = \sum_{i=1}^{N} \sum_{k=1}^{K} r_k^{(i)} \left[ \log \pi_k + \sum_{j=1}^{D} x_j^{(i)} \log \theta_{k,j} + (1 - x_j^{(i)}) \log(1 - \theta_{k,j}) \right] \]

\[ + \sum_{k=1}^{K} (a_k - 1) \log \pi_k + \sum_{k=1}^{K} \sum_{j=1}^{D} [(a - 1) \log \theta_{k,j} + (b - 1) \log(1 - \theta_{k,j})] + C \]
Derivative wrt. $\theta_{k,j}$

$$J(\Theta, \pi) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_k^{(i)} \left[ \log \pi_k + \sum_{j=1}^{D} x_j^{(i)} \log \theta_{k,j} + (1 - x_j^{(i)}) \log(1 - \theta_{k,j}) \right]$$

$$+ \sum_{k=1}^{K} (a_k - 1) \log \pi_k + \sum_{k=1}^{K} \sum_{j=1}^{D} [(a - 1) \log \theta_{k,j} + (b - 1) \log(1 - \theta_{k,j})] + C$$

- First we take derivative wrt. $\theta_{k,j}$:

$$\frac{\partial J}{\partial \theta_{k,j}} = \sum_{i=1}^{N} r_k^{(i)} \left[ x_j^{(i)} \frac{1}{\theta_{k,j}} + (1 - x_j^{(i)}) \frac{1}{\theta_{k,j} - 1} \right] + (a - 1) \frac{1}{\theta_{k,j} - 1} + (b - 1) \frac{1}{\theta_{k,j} - 1}$$

$$= \frac{1}{\theta_{k,j}} \left( \sum_{i=1}^{N} [r_k^{(i)} x_j^{(i)}] + (a - 1) \right) + \frac{1}{\theta_{k,j} - 1} \left( \sum_{i=1}^{N} [r_k^{(i)}] - \sum_{i=1}^{N} [r_k^{(i)} x_j^{(i)}] + (b - 1) \right)$$
Derivative wrt. $\theta_{k,j}$

\[
\frac{\partial J}{\partial \theta_{k,j}} = \sum_{i=1}^{N} r^{(i)}_k \left[ x^{(i)}_j \frac{1}{\theta_{k,j}} + (1 - x^{(i)}_j) \frac{1}{\theta_{k,j} - 1} \right] + (a - 1) \frac{1}{\theta_{k,j}} + (b - 1) \frac{1}{\theta_{k,j} - 1}
\]

\[
= \frac{1}{\theta_{k,j}} \left( \sum_{i=1}^{N} [r^{(i)}_k x^{(i)}_j] + (a - 1) \right) + \frac{1}{\theta_{k,j} - 1} \left( \sum_{i=1}^{N} [r^{(i)}_k] - \sum_{i=1}^{N} [r^{(i)}_k x^{(i)}_j] + (b - 1) \right)
\]

Setting this to zero, and multiplying both sides by $\theta_{k,j} (\theta_{k,j} - 1)$ yields:

\[
0 = (\theta_{k,j} - 1) \left( \sum_{i=1}^{N} [r^{(i)}_k x^{(i)}_j] + (a - 1) \right) + \theta_{k,j} \left( \sum_{i=1}^{N} [r^{(i)}_k] - \sum_{i=1}^{N} [r^{(i)}_k x^{(i)}_j] + (b - 1) \right)
\]
Derivative wrt. $\theta_{k,j}$

$$\frac{\partial J}{\partial \theta_{k,j}} = \sum_{i=1}^{N} r^{(i)}_k \left[ x^{(i)}_j \frac{1}{\theta_{k,j}} + (1 - x^{(i)}_j) \frac{1}{\theta_{k,j} - 1} \right] + (a - 1) \frac{1}{\theta_{k,j}} + (b - 1) \frac{1}{\theta_{k,j} - 1}$$

$$= \frac{1}{\theta_{k,j}} \left( \sum_{i=1}^{N} [r^{(i)}_k x^{(i)}_j] + (a - 1) \right) + \frac{1}{\theta_{k,j} - 1} \left( \sum_{i=1}^{N} [r^{(i)}_k] - \sum_{i=1}^{N} [r^{(i)}_k x^{(i)}_j] + (b - 1) \right)$$

- Setting this to zero, and multiplying both sides by $\theta_{k,j} (\theta_{k,j} - 1)$ yields:

$$0 = (\theta_{k,j} - 1) \left( \sum_{i=1}^{N} [r^{(i)}_k x^{(i)}_j] + (a - 1) \right) + \theta_{k,j} \left( \sum_{i=1}^{N} [r^{(i)}_k] - \sum_{i=1}^{N} [r^{(i)}_k x^{(i)}_j] + (b - 1) \right)$$

- This gives:

$$\theta_{k,j} = \frac{\sum_{i=1}^{N} [r^{(i)}_k x^{(i)}_j] + (a - 1)}{\sum_{i=1}^{N} [r^{(i)}_k x^{(i)}_j] + (a - 1) + \sum_{i=1}^{N} [r^{(i)}_k] - \sum_{i=1}^{N} [r^{(i)}_k x^{(i)}_j] + (b - 1)}$$

$$= \frac{\sum_{i=1}^{N} [r^{(i)}_k x^{(i)}_j] + a - 1}{\sum_{i=1}^{N} [r^{(i)}_k] + a + b - 2}$$
Derivative wrt. $\pi_k$

- Now we take derivative wrt. $\pi_k$.
- Note that it is a bit trickier because we need to account for the condition $\sum_{k=1}^{K} \pi_k = 1$.
- This can be done with the use of a Lagrange multiplier.
- Let $J_\lambda = J + \lambda(\sum_{k=1}^{K} [\pi_k] - 1)$

$$\frac{\partial J_\lambda}{\partial \pi_k} = \sum_{i=1}^{N} r^{(i)}_k \frac{1}{\pi_k} + (a_k - 1) \frac{1}{\pi_k} + \lambda$$
Derivative wrt. $\pi_k$

- Now we take derivative wrt. $\pi_k$.
- Note that it is a bit trickier because we need to account for the condition $\sum_{k=1}^{K} \pi_k = 1$.
- This can be done with the use of a Lagrange multiplier.
- Let $J_\lambda = J + \lambda(\sum_{k=1}^{K} [\pi_k] - 1)$

\[
\frac{\partial J_\lambda}{\partial \pi_k} = \sum_{i=1}^{N} r_k^{(i)} \frac{1}{\pi_k} + (a_k - 1) \frac{1}{\pi_k} + \lambda
\]

- Setting this to zero, we get:

\[
\pi_k = \frac{(a_k - 1) + \sum_{i=1}^{N} [r_k^{(i)}]}{\lambda}
\]

- Knowing that $\pi_k$ sums to one, we obtain:

\[
\pi_k = \frac{(a_k - 1) + \sum_{i=1}^{N} [r_k^{(i)}]}{\sum_{k=1}^{K} [(a_k - 1) + \sum_{i=1}^{N} [r_k^{(i)}]]} = \frac{(a_k - 1) + \sum_{i=1}^{N} [r_k^{(i)}]}{N + \sum_{k=1}^{K} (a_k - 1)}
\]

- (We used $\sum_{i=1}^{N} \sum_{k=1}^{K} r_k^{(i)} = \sum_{i=1}^{N} 1 = N$)
Part 2: Posterior inference

- We represent partial observations in terms of variables $m^{(i)}_j$, where $m^{(i)}_j = 1$ if the $j$th pixel of the $i$th image is observed, and 0 otherwise.

- Derive the posterior probability distribution $p(z \mid x_{\text{obs}})$, where $x_{\text{obs}}$ denotes the subset of the pixels which are observed.

- Using Bayes rule, we have:

$$p(z = k \mid x) = \frac{p(x \mid z = k)p(z = k)}{p(x)}$$

$$= \frac{\pi_k \prod_{j=1}^{D} \theta_{k,j}^{m_j} x_j (1 - \theta_{k,j}^{m_j} (1-x_j))}{\sum_{l=1}^{K} \pi_l \prod_{j=1}^{D} \theta_{l,j}^{m_j} x_j (1 - \theta_{l,j}^{m_j} (1-x_j))}$$
Part 3: Posterior Predictive Mean

- Computes the posterior predictive means of the missing pixels given the observed ones.
- The posterior predictive distribution is:

\[ p(x_2 | x_1) = \sum_z p(z | x_1)p(x_2 | z, x_1) \]

- Assume that the \( x_i \) values are conditionally independent given \( z \).
- For instance, the pixels in one half of an image are clearly not independent of the pixels in the other half. But they are roughly independent, conditioned on a detailed description of everything going on in the image.
- So we have:

\[ \mathbb{E}[p(x_{mis} | x_{obs})] = \sum_{k=1}^{K} r_k p(x_{mis} = 1 | z = k) = \sum_{k=1}^{K} r_k \text{Bernoulli}(\theta_{k,mis}) \]

\[ = \sum_{k=1}^{K} r_k \theta_{k,mis} \]