1 EM for Mixture of Gaussians

Mixture of K Gaussians is represented as

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k), \tag{1}$$

where π_k represents the probability that a data point belongs to the kth component. As it is probability, it satisfies $0 \leq \pi_k \leq 1$ and $\sum_k \pi_k = 1$. In this problem, we are going to represent this in a slightly different manner with explicit latent variables. Specifically, we introduce 1-of-K coding representation for latent variables $z^{(k)} \in \mathbb{R}^K$ for $k = 1, ..., K$. Each $z^{(k)}$ is a binary vector of size K, with 1 only in kth element and 0 in all others. That is,

$$\begin{aligned} z^{(1)} &= [1; 0; ...; 0] \\ z^{(2)} &= [0; 1; ...; 0] \\ &\vdots \\ z^{(K)} &= [0; 0; ...; 1]. \end{aligned}$$

For example, if the second component generated data point x^n, its latent variable z^n is given by $[0; 1; ...; 0] = z^{(2)}$. With this representation, we can express $p(z)$ as

$$p(z) = \prod_{k=1}^{K} {\pi_k}^{z_k},$$

where z_k indicates kth element of vector z. Also, $p(x|z)$ can be represented similarly as

$$p(x|z) = \prod_{k=1}^{K} \mathcal{N}(x|\mu_k, \Sigma_k)^{z_k}.$$

By the sum rule of probability, (1) can be represented by

$$p(x) = \sum_{z \in Z} p(z)p(x|z). \tag{2}$$

where $Z = \{z^{(1)}, z^{(2)}, ..., z^{(K)}\}$.

(a) Show that (2) is equivalent to (1). [5 pts]

Answer:

$$\begin{aligned} p(x) &= \sum_{z \in Z} p(z)p(x|z) \\ &= \sum_{i=1}^{K} p(z^{(i)})p(x|z^{(i)}) \\ &= \sum_{i=1}^{K} \left(\prod_{k=1}^{K} \pi_k^{z_k^{(i)}} \right) \left(\prod_{k=1}^{K} \mathcal{N}(x|\mu_k, \Sigma_k)^{z_k^{(i)}} \right) \\ &= \sum_{i=1}^{K} \pi_i \mathcal{N}(x|\mu_i, \Sigma_i) \end{aligned}$$

So that (2) is equivalent to (1).

(b) In reality, we do not know which component each data point is from. Thus, we estimate the responsibility (expectation of z_k^n) in the E-step of EM. Since z_k^n is either 1 or 0, its expectation is the probability for the point x_n to belong to the component z_k. In other words, we estimate $p(z_k^n|x_n)$. Derive the formula for this estimation by using Bayes rule. Note that, in the E-step, we assume all other parameters, i.e. π_k, μ_k, and Σ_k, are fixed, and we want to express $p(z_k^n|x_n)$ as a function of these fixed parameters. [10 pts]

Answer:

We use the Bayes rule to derive the conditional probability of z given x:

$$\begin{aligned} p(z_k^n = 1|x) &= \frac{p(z_k^n = 1)p(x|z_k = 1)}{p(x)} \\ &= \frac{p(z_k^n = 1)p(x|z_k = 1)}{\sum_{i=1}^{K} p(z_i^n = 1)p(x|z_i = 1)} \\ &= \frac{\pi_k \mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_{i=1}^{K} \pi_i \mathcal{N}(x|\mu_i, \Sigma_i)} \end{aligned}$$

(c) In the M-Step, we re-estimate parameters π_k, μ_k, and Σ_k by maximizing the log-likelihood. Given N i.i.d (Independent Identically Distributed) data samples, derive the update formula for each parameter. Note that in order to obtain an update rule for the M-step, we fix the responsibilities, i.e. $p(z_k^n|x_n)$, which we have already calculated in the E-step. [15 pts]

Hint: Use Lagrange multiplier for π_k to apply constraints on it.

Answer:

The likelihood of N i.i.d data samples is:

$$\prod_{i=1}^{N} p(x_i) = \prod_{i=1}^{N} \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k) \right)$$

So the log-likelihood is:

$$L = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k) \right)$$

[μ_k] We take the partial derivative for μ_k and set it to zero:

$$\begin{aligned}
\frac{\partial L}{\partial \mu_k} =& \sum_{i=1}^{N} \left(\frac{\pi_k}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)} \frac{\partial \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\partial \mu_k} \right) \\
=& \sum_{i=1}^{N} \left(\frac{\pi_k}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)} \times \frac{\partial \frac{1}{\sqrt{(2\pi)^t |\Sigma|}} \exp\left(-\frac{1}{2}(x_i - \mu_k)^T \Sigma^{-1}(x_i - \mu_k)\right)}{\partial \mu_k} \right) \\
=& \sum_{i=1}^{N} \left(\frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)} \times -\Sigma_k^{-1}(x_i - \mu_k) \right) \\
=& 0
\end{aligned}$$

By multiplying Σ_k to the both sides of equation, we can see

$$\begin{aligned}
\mu_k =& \frac{1}{\sum_{i=1}^{N} \frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}} \sum_{i=1}^{N} \frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)} x_i \\
=& \frac{1}{\sum_{i=1}^{N} p(z_k^i = 1|x_i)} \sum_{i=1}^{N} p(z_k^n = 1|x_i) x_i
\end{aligned}$$

[Σ_k] Similarly, we take the partial derivative for Σ_k and set it to zero:

$$\begin{aligned}
\frac{\partial L}{\partial \Sigma_k} =& \sum_{i=1}^{N} \left(\frac{\pi_k}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)} \frac{\partial \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\partial \Sigma_k} \right) \\
=& \sum_{i=1}^{N} \left(\frac{\pi_k}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)} \times \frac{\partial \frac{1}{\sqrt{(2\pi)^t |\Sigma|}} \exp\left(-\frac{1}{2}(x_i - \mu_k)^T \Sigma^{-1}(x_i - \mu_k)\right)}{\partial \Sigma_k} \right) \\
=& \frac{1}{2} \sum_{i=1}^{N} \left(\frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)} \times \Sigma_k^{-1} \left((x_i - \mu_k)(x_i - \mu_k)^T \Sigma_k^{-1} - \mathbb{I} \right) \right) \\
=& \frac{1}{2} \Sigma_k^{-1} \sum_{i=1}^{N} p(z_k^i = 1|x_i) \left((x_i - \mu_k)(x_i - \mu_k)^T \Sigma_k^{-1} - \mathbb{I} \right) \\
=& \frac{1}{2} \Sigma_k^{-1} \left(\sum_{i=1}^{N} p(z_k^i = 1|x_i)(x_i - \mu_k)(x_i - \mu_k)^T \Sigma_k^{-1} - \sum_{i=1}^{N} p(z_k^i = 1|x_i) \mathbb{I} \right) \\
=& 0
\end{aligned}$$

By multiplying Σ_k^{-1} and 2 to both sides of equation, we can get the maximum likelihood estimation of Σ_k as:

$$\Sigma_k = \frac{\sum_{i=1}^{N} p(z_k^i = 1|x_i)(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^{N} p(z_k^i = 1|x_i)}$$

[π_k] We formulate the maximum-likelihood of π_k as a constrained optimization problem:

$$\begin{aligned}
\arg\min_{\pi_k} \quad & -\sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k) \right) \\
\text{s.t.} \quad & \sum_{k=1}^{K} \pi_k = 1
\end{aligned}$$

By introducing the Lagrange multiplier λ, we can get the unconstrained min-max:

$$\min_{\pi_k}\max_{\lambda} -\sum_{i=1}^{N}\log\left(\sum_{k=1}^{K}\pi_k\mathcal{N}(x_i|\mu_k,\Sigma_k)\right)+\lambda(\sum_{k=1}^{K}\pi_k-1)$$

Take the derivative of π_k and set it to be zero, we can get:

$$\begin{aligned}\frac{\partial}{\partial\pi_k} =& -\sum_{i=1}^{N}\frac{\mathcal{N}(x_i|\mu_k,\Sigma_k)}{\sum_{j=1}^{K}\pi_j\mathcal{N}(x_i|\mu_j,\Sigma_j)}+\lambda\\ =&\frac{1}{\pi_k}\left(-\sum_{i=1}^{N}p(z_k^i=1|x_i)+\pi_k\lambda\right)\\ =&0\end{aligned}$$

So

$$\pi_k=\frac{\sum_{i=1}^{N}p(z_k^i=1|x_i)}{\lambda}$$

By applying the sum constrain to π_k, then we get $\lambda=N$, so:

$$\pi_k=\frac{\sum_{i=1}^{N}p(z_k^i=1|x_i)}{N}$$

(d) EM and K-Means [10 pts]

K-means can be viewed as a particular limit of EM for Gaussian mixture. Considering a mixture model in which all components have covariance ϵI, show that in the limit $\epsilon\to 0$, maximizing the expected complete data log-likelihood for this model is equivalent to minimizing objective function in K-means:

$$J=\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma_{nk}\|x_n-\mu_k\|^2,$$

where $\gamma_{nk}=1$ if x_n belongs to the k-th cluster and $\gamma_{nk}=0$ otherwise.

Answer:

If we use the covariance matrix as mentioned in the problem, then we can rewrite the normal distribution of the kth mixture as follows:

$$\begin{aligned}\mathcal{N}(x_i|\mu_k,\Sigma_k) =&\frac{1}{\sqrt{2\pi^t\det(\Sigma)}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)\\ =&\frac{1}{(2\pi)^{\frac{t}{2}}\epsilon^{\frac{t}{2}}}\exp\left(-\frac{1}{2\epsilon}\|x_i-\mu_k\|^2\right)\end{aligned}$$

Where t is the dimensionality of samples X.

Plugin the above equation into the posterior probability $p(z_k^i=1|x_i)$, we get:

$$p(z_k^i=1|x_i)=\frac{\pi_k\exp\left(-\frac{\|x_i-\mu_k\|^2}{2\epsilon}\right)}{\sum_{j=1}^{K}\pi_j\exp\left(-\frac{\|x_i-\mu_j\|^2}{2\epsilon}\right)}$$

Actually, $\forall p,q=1,2,\ldots,K$, if $\|x_i-\mu_p\|^2>\|x_i-\mu_q\|^2$, then wen can get the limit that:

$$\lim_{\epsilon \to 0} \frac{\pi_p \exp\left(-\frac{\|x_i - \mu_p\|^2}{2\epsilon}\right)}{\pi_q \exp\left(-\frac{\|x_i - \mu_q\|^2}{2\epsilon}\right)} = 0$$

That is to say, if we are not choosing the k such that $\|x_i - \mu_k\|^2$ is not the smallest among all other $j = 1, 2, \ldots, K, j \neq k$, then $p(z_k^i = 1|x_i)$ will become zero. Otherwise, it will become one. i.e.,

$$p(z_k^i = 1|x_i) = \begin{cases} 1, k = \arg\min_k \|x_i - \mu_k\|^2 \\ 0, otherwise \end{cases}$$

This is exactly the same as the 'hard' cluster assignment in k-means.

Plugin the above posterior probability into the maximum likelihood estimation for μ_k, then we can also get the similar 'mean' value in k-means:

$$\mu_k = \frac{\sum_{i=1}^N p(z_k^i = 1|x_i) x_i}{\sum_{i=1}^N p(z_k^i = 1|x_i)}$$

According to the equation 9.40 in PRML, we can write the expectation of complete dataset (including samples x and latent variables z) as:

$$E = \sum_{i=1}^N \sum_{k=1}^K p(z_k^i = 1|x_i) \left(\log \pi_k + \log \mathcal{N}(x_i|\mu_k, \Sigma_k)\right)$$

As explained in PRML through equation 9.3.2, as $\epsilon \to 0$, $\gamma(z_{nk})$ goes to r_{nk}.

From the question we know that $\Sigma = \epsilon I$, and so we need to maximize with only parameter μ_k. π_k will be the proportion of data points assigned to each cluster. As these correspond to hard assignments, it means that $\pi_k > 0$ for any reasonable initialization. Considering that, we can maximize only w.r.t μ_k independent of π.

Hence, we get

$$E = \sum_{i=1}^N \sum_{k=1}^K r_{nk} \left(\log \mathcal{N}(x_i|\mu_k, \Sigma_k)\right)$$

Using the derivation for the normal distribution from above, we get

$$E = \sum_{i=1}^N \sum_{k=1}^K r_{nk} \Big(-\frac{1}{2\epsilon} \|x_n - \mu_k\|^2 \Big) + const.$$

In the limit $\epsilon \to 0$, we can say that the above expected likelihood $\mathbb{E}$ is equivalent to the negative of J, the distortion function upto a scaling factor. This scaling factor is independent of our parameter μ_k. Hence, maximizing the expected complete data log likelihood is equivalent to minimizing J.

2 Density Estimation

Consider a histogram-like density model in which the space x is divided into fixed regions for which density $p(x)$ takes constant value h_i over ith region, and that the volume of region i in denoted as Δ_i. Suppose we have a set of N observations of x such that n_i of these observations fall in regions i.

(a) What is the log-likelihood function? [8 pts]

Answer:

The likelihood function is:

$$\prod_i h_i^{n_i}$$

So the log-likelihood is:

$$L = \sum_i n_i \log h_i$$

(b) Derive an expression for the maximum likelihood estimator for h_i. [10 pts]

Hint: This is a constrained optimization problem. Remember that $p(x)$ must integrate to unity. Since $p(x)$ has constant value h_i over region i, which has volume Δ_i. The normalization constraint is $\sum_i h_i \Delta_i = 1$. Use Lagrange multiplier by adding $\lambda \left(\sum_i h_i \Delta_i - 1\right)$ to your objective function.

Answer:

We can formulate the problem as the following constrained optimization problem:

$$\begin{aligned} \arg\min_{h_i} \quad & -\sum_i n_i \log h_i \\ \text{s.t.} \quad & \sum_i h_i \Delta_i = 1 \end{aligned}$$

By introducing the Lagrange multiplier λ, we can reformulate it as the following unconstrained min-max optimization problem:

$$\min_{h_i} \max_{\lambda} -\sum_i n_i \log h_i + \lambda \left(\sum_i h_i \Delta_i - 1 \right)$$

Take the partial derivative of h_i, and set it to be zero, we get:

$$\begin{aligned} \frac{\partial}{\partial h_i} &= -\frac{n_i}{h_i} + \lambda \Delta_i \\ &= 0 \end{aligned}$$

So

$$h_i = \frac{n_i}{\lambda \Delta_i}$$

Take the consideration of sum constrain (i.e., $\sum_i h_i \Delta_i = 1$), we can infer that

$$\lambda = N$$

So the maximum likelihood estimation of h_i is:

$$h_i = \frac{n_i}{N \Delta_i}$$

(c) Mark T if it is always true, and F otherwise. Briefly explain why. [12 pts]

- Non-parametric density estimation usually does not have parameters.

  Answer: F.

  Non-parametric means there is no assumptions about the probability distributions of the variables being assessed. It doesn't means no parameters. Actually, this kind of models can not be described by a fixed number of parameters. In other words, there could be many many parameters.

- The Epanechnikov kernel is the optimal kernel function for all data.

  Answer: F.

  Epanechnikov kernel minimizes AMISE (Asymptotic Mean Integrated Squared Error) and it is optimal. However, it is optimal in a minimum variance sense. In consideration for the real world dataset, the efficiency and other mathematical properties might be more important than the kernel efficiency. So it is hard to say Epanechnikov kernel is the optimal for all data.

- Histogram is an efficient way to estimate density for high-dimensional data.

  Answer: F.

  If the dimension is d and the number of bins for each dimension is n, then there will be n^d bins. It can be easily larger than the total number of samples, in which there will be many empty bins. So the parameter estimation will fail. What's more, the insufficient training samples will make the model over fitting, which results in the poor generalization ability.

- Parametric density estimation assumes the shape of probability density.

  Answer: F.

  Parametric method has a fixed number of parameters. It makes assumption on the distribution of the probability density. However, they do not necessarily have exactly the same shape. Take the Poisson distribution for example. If we get different parameters, the shape of the distribution will vary a lot.

3 Information Theory

In the lecture you became familiar with the concept of entropy for one random variable and mutual information. For a pair of discrete random variables X and Y with the joint distribution $p(x, y)$, the *joint entropy* $H(X, Y)$ is defined as

$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(x, y) \tag{3}$$

which can also be expressed as

$$H(X, Y) = -\mathbb{E}[\log p(X, Y)] \tag{4}$$

Let X and Y take on values $x_1, x_2, ..., x_r$ and $y_1, y_2, ..., y_s$ respectively. Let Z also be a discrete random variable and $Z = X + Y$.

(a) Prove that $H(X, Y) \leq H(X) + H(Y)$ [4 pts]

Answer: We can easily verify that $\sum_{j=1}^{s} p(x_i, y_j) = p(x_i)$
So we can get,

$$
\begin{aligned}
H(X)+H(Y) &= \sum_{i=1}^{r} p(x_i)\log\frac{1}{p(x_i)} + \sum_{j=1}^{s} p(y_j)\log\frac{1}{p(y_j)} \\
&= \sum_{i=1}^{r}(\sum_{j=1}^{s} p(x_i,y_j))\log\frac{1}{p(x_i)} + \sum_{j=1}^{s}(\sum_{i=1}^{r} p(x_i,y_j))\log\frac{1}{p(y_j)} \\
&= \sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i,y_j)\log\frac{1}{p(x_i)} + \sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i,y_j)\log\frac{1}{p(y_j)} \\
&= \sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i,y_j)\log\frac{1}{p(x_i)p(x_j)}
\end{aligned}
$$

Before we prove the conclusion, lets use an inequality:

$$\ln x \leq x-1, x>0$$

This is easy to prove, by considering the derivative of $x-1-\ln x$ and $\ln x = x-1 = 0$ when $x=1$. Now we will prove

$$p(x_i,y_j)\log\frac{1}{p(x_i,y_j)} \leq p(x_i,y_j)\log\frac{1}{p(x_i)p(y_j)} + p(x_i).p(y_j) - p(x_i,y_j)$$

Proof:

$$
\begin{aligned}
p(x_i,y_j)\log\frac{1}{p(x_i,y_j)} &= p(x_i,y_j)\left(\log(\frac{1}{p(x_i)p(y_j)}\times\frac{p(x_i).p(y_j)}{p(x_i,y_j)})\right) \\
&= p(x_i,y_j)\left(\log\frac{1}{p(x_i)p(y_j)} + \log\frac{p(x_i).p(y_j)}{p(x_i,y_j)}\right) \\
&\leq p(x_i,y_j)\left(\log\frac{1}{p(x_i)p(y_j)} + \frac{p(x_i).p(y_j)}{p(x_i,y_j)} - 1\right) \\
&= p(x_i,y_j)\log\frac{1}{p(x_i)p(y_j)} + p(x_i).p(y_j) - p(x_i,y_j)
\end{aligned}
$$

Taking the sum over i and j, we get:

$$\sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i,y_j)log\frac{1}{p(x_i,y_j)} \leq \sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i,y_j)log\frac{1}{p(x_i)p(y_j)} + \sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i).p(y_j) - \sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i,y_j)$$

Since, $\sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i).p(y_j) = \sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i,y_j) = 1$, we can get

$$\sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i,y_j)log\frac{1}{p(x_i,y_j)} \leq \sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i,y_j)log\frac{1}{p(x_i)p(y_j)}$$

This proves that,

$$H(X,Y) \leq H(X)+H(Y)$$

(b) Show that $I(X;Y) = H(X)+H(Y)-H(X,Y)$. [2 pts]

Answer:

$$I(X;Y) = H(X) - H(X|Y)$$

The condition entropy can be written as:

$$\begin{aligned}
H(X,Y) &= \sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i|y_j).p(y_j).\log\frac{1}{p(x_i|y_j)} \\
&= \sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i,y_j).\log\frac{y_j}{p(x_i,y_j)} \\
&= \sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i,y_j).\log\frac{1}{p(x_i,y_j)} - \sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i,y_j).\log\frac{1}{p(y_j)} \\
&= \sum_{i=1}^{r}\sum_{j=1}^{s} p(x_i,y_j).\log\frac{1}{p(x_i,y_j)} - \sum_{j=1}^{s} p(y_j).\log\frac{1}{p(y_j)} \\
&= H(X,Y) - H(Y)
\end{aligned}$$

(c) Under what conditions does $H(Z) = H(X) + H(Y)$. [4 pts]

Answer: Here we first prove that, for a discrete R.V X, its entropy is greater than or equal to the entropy of function f over X, i.e.,

$$H(f(X)) \leq H(X)$$

Proof:
Here we assume that $X \in A = \{x_1, x_2, ...x_r\}$ and $f(x) \in B = \{f(x_1), f(x_2), ..., f(x_r)\}$. It is easy to see that $|A| = r \geq |B| = t$. So we rewrite $B = \{f_1, f_2, ...f_t\}$. Let S_i be the set of inverse image of f_i, i.e.

$$S_i = \{x|f(x) = f_i\}, i = 1, 2, ..., t$$

So the entropy of $f(X)$ can be written as:

$$\begin{aligned}
H(f(X)) &= \sum_{i=1}^{t} p(f_i)\log\frac{1}{p(f_i)} \\
&= \sum_{i=1}^{t}\left(\sum_{x\in S_i} p(x)\right)\log\frac{1}{\sum_{x\in S_i} p(x)}
\end{aligned}$$

$\forall i \in 1, 2, ...t$, it is easy to see that

$$\begin{aligned}
\log\frac{1}{\sum_{x\in S_i} p(x)} &\leq min\{\log\frac{1}{p(x)}|x \in S_i\} \\
&\leq \sum_{i=1}^{r} p(x).\log\frac{1}{p(x)} \\
&= H(X)
\end{aligned}$$

Similarly, we can extend the above conclusion to prove that

$$H(Z) = H(X+Y) = H(f(X,Y)) \leq H(X,Y) \leq H(X) + H(Y)$$

What's more, if (X,Y) is the function image of Z, i.e. , there exists a function $g : Z \to X \times Y$, then we can get:

$$H(X,Y) = H(g(Z)) \leq H(Z)$$

If X and Y are Independent at the same time, ie. $H(X,Y) = H(X) + H(Y)$, then we will get the following two inequalities:

$$\begin{cases} H(Z) = H(X+Y) H(X,Y) = H(X) + H(Y) \\ H(X,Y) = H(X) + H(Y) H(Z) \end{cases}$$

So the equality satisfies. $H(Z) = H(X) + H(Y)$ [Conclusion]
We need the following two conditions to satisfy the equality:

1. X and Y are Independent
2. (X,Y) is a function of Z. In other words, if we know the value of Z, then we can simply get the value of X and Y . Here is an example:

$$X = 1, 2, 3, ..., 10$$
$$Y = 100, 200, 300, ..., 1000$$

4 Programming: Text Clustering

In this problem, we will explore the use of EM algorithm for text clustering. Text clustering is a technique for unsupervised document organization, information retrieval. We want to find how to group a set of different text documents based on their topics. First we will analyze a model to represent the data.

Bag of Words

The simplest model for text documents is to understand them as a collection of words. To keep the model simple, we keep the collection unordered, disregarding grammar and word order. What we do is counting how often each word appears in each document and store the word counts into a matrix, where each row of the matrix represents one document. Each column of matrix represent a specific word from the document dictionary. Suppose we represent the set of n_d documents using a matrix of word counts like this:

$$D_{1:n_d} = \begin{pmatrix} 2 & 6 & ... & 4 \\ 2 & 4 & ... & 0 \\ \vdots & & \ddots & \end{pmatrix} = T$$

This means that word W_1 occurs twice in document D_1 . Word W_{n_w} occurs 4 times in document D_1 and not at all in document D_2.

Multinomial Distribution

The simplest distribution representing a text document is multinomial distribution(Bishop Chapter 2.2). The probability of a document D_i is:

$$p(D_i) = \prod_{j=1}^{n_w} \mu_j^{T_{ij}}$$

Here, μ_j denotes the probability of a particular word in the text being equal to w_j, T_{ij} is the count of the word in document. So the probability of document D_1 would be $p(D_1) = \mu_1^2 \cdot \mu_2^6 \cdot ... \cdot \mu_{n_w}^4$.

Mixture of Multinomial Distributions

In order to do text clustering, we want to use a mixture of multinomial distributions, so that each topic has a particular multinomial distribution associated with it, and each document is a mixture of different topics. We define $p(c) = \pi_c$ as the mixture coefficient of a document containing topic c, and each topic is modeled by a multinomial distribution $p(D_i|c)$ with parameters μ_{jc}, then we can write each document as a mixture over topics as

$$p(D_i) = \sum_{c=1}^{n_c} p(D_i|c)p(c) = \sum_{c=1}^{n_c} \pi_c \prod_{j=1}^{n_w} \mu_{jc}^{T_{ij}}$$

EM for Mixture of Multinomials

In order to cluster a set of documents, we need to fit this mixture model to data. In this problem, the EM algorithm can be used for fitting mixture models. This will be a simple topic model for documents. Each topic is a multinomial distribution over words (a mixture component). EM algorithm for such a topic model, which consists of iterating the following steps:

1. Expectation

   Compute the expectation of document D_i belonging to cluster c:

   $$\gamma_{ic} = \frac{\pi_c \prod_{j=1}^{n_w} \mu_{jc}^{T_{ij}}}{\sum_{c=1}^{n_c} \pi_c \prod_{j=1}^{n_w} \mu_{jc}^{T_{ij}}}$$

2. Maximization

   Update the mixture parameters, i.e. the probability of a word being W_j in cluster (topic) c, as well as prior probability of each cluster.

   $$\mu_{jc} = \frac{\sum_{i=1}^{n_d} \gamma_{ic} T_{ij}}{\sum_{i=1}^{n_d} \sum_{l=1}^{m_w} \gamma_{ic} T_{il}}$$

   $$\pi_c = \frac{1}{n_d} \sum_{i=1}^{n_d} \gamma_{ic}$$

Task [20 pts]

Implement the algorithm and run on the toy dataset `data.mat`. You can find detailed description about the data in the `homework2.m` file. Observe the results and compare them with the provided true clusters each document belongs to. Report the evaluation (e.g. accuracy) of your implementation.

Hint: We already did the word counting for you, so the data file only contains a count matrix like the one shown above. For the toy dataset, set the number of clusters $n_c = 4$. You will need to initialize the parameters. Try several different random initial values for the probability of a word being W_j in topic c, μ_{jc}. Make sure you normalized it. Make sure that you should not use the true cluster information during your learning phase.

Extra Credit: Realistic Topic Models [20pts]

The above model assumes all the words in a document belongs to some topic at the same time. However, in real world datasets, it is more likely that some words in the documents belong to one topic while other words belong to some other topics. For example, in a news report, some words may talk about "Ebola" and

"health", while others may mention "administration" and "congress". In order to model this phenomenon, we should model each word as a mixture of possible topics.

Specifically, consider the log-likelihood of the joint distribution of document and words

$$\mathcal{L} = \sum_{d \in \mathcal{D}} \sum_{w \in \mathcal{W}} T_{dw} \log P(d, w), \tag{5}$$

where T_{dw} is the counts of word w in the document d. This count matrix is provided as input.

The joint distribution of a specific document and a specific word is modeled as a mixture

$$P(d, w) = \sum_{z \in \mathcal{Z}} P(z) P(w|z) P(d|z), \tag{6}$$

where $P(z)$ is the mixture proportion, $P(w|z)$ is the distribution over the vocabulary for the z-th topic, and $P(d|z)$ is the probability of the document for the z-th topic. And these are the parameters for the model.

The E-step calculates the posterior distribution of the latent variable conditioned on all other variables

$$P(z|d, w) = \frac{P(z)P(w|z)P(d|z)}{\sum_{z'} P(z')P(w|z')P(d|z')}. \tag{7}$$

In the M-step, we maximizes the expected complete log-likelihood with respect to the parameters, and get the following update rules

$$P(w|z) = \frac{\sum_d T_{dw} P(z|d, w)}{\sum_{w'} \sum_d T_{dw'} P(z|d, w')} \tag{8}$$

$$P(d|z) = \frac{\sum_w T_{dw} P(z|d, w)}{\sum_{d'} \sum_w T_{d'w} P(z|d', w)} \tag{9}$$

$$P(z) = \frac{\sum_d \sum_w T_{dw} P(z|d, w)}{\sum_{z'} \sum_{d'} \sum_{w'} T_{d'w'} P(z'|d', w')}. \tag{10}$$

Task

Implement EM for maximum likelihood estimation and cluster the text data provided in the `nips.mat` file you downloaded. You can print out the top key words for the topics/clusters by using the `show_topics.m` utility. It takes two parameters: 1) your learned conditional distribution matrix, i.e., $P(w|z)$ and 2) a cell array of words that corresponds to the vocabulary. You can find the cell array `wl` in the `nips.mat` file. Try different values of k and see which values produce sensible topics. In assessing your code, we will use another dataset and observe the produces topics.