Outline

1. Administrative matters
   - Project and Homework

2. Review of last lecture

3. EM Algorithm

4. Relation between K-means and GMMs
Mini project description released
Please get working on it as early as possible

# of homework now reduces to 5, namely two more remaining

Pool for Quiz 2
closed-book (short, in-class) versus open-book (long, possibly take-home)
Outline

1. Administrative matters

2. Review of last lecture
   - K-means clustering
   - Gaussian mixture models

3. EM Algorithm

4. Relation between K-means and GMMs
K-means clustering

Setup Given $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$ and $K$, we want to output

- $\{\mathbf{\mu}_k\}_{k=1}^K$: prototypes of clusters
- $A(\mathbf{x}_n) \in \{1, 2, \ldots, K\}$: the cluster membership, i.e., the cluster ID assigned to $\mathbf{x}_n$
- Distortion measure (clustering objective function, cost function)

$$J = N \sum_{n=1}^N \sum_{k=1}^K r_{nk} \| \mathbf{x}_n - \mathbf{\mu}_k \|^2_2$$

where $r_{nk} = 1$ if and only if $A(\mathbf{x}_n) = k$

Algorithm alternate optimization between $\{r_{nk}\}$ and $\{\mathbf{\mu}_k\}$

- **Step 0** Initialize $\{\mathbf{\mu}_k\}$ to some values
- **Step 1** Assume the current value of $\{\mathbf{\mu}_k\}$ fixed, minimize $J$ over $\{r_{nk}\}$
- **Step 2** Assume the current value of $\{r_{nk}\}$ fixed, minimize $J$ over $\{\mathbf{\mu}_k\}$
- **Step 3** Determine whether to stop or return to Step 1
Gaussian mixture models: formal definition

A Gaussian mixture model has the following density function for $\mathbf{x}$

$$p(\mathbf{x}) = \sum_{k=1}^{K} \omega_k N(\mathbf{x} | \mu_k, \Sigma_k)$$

where

- $K$: the number of Gaussians — they are called (mixture) components
- $\mu_k$ and $\Sigma_k$: mean and covariance matrix of the $k$-th component
- $\omega_k$: mixture weights — they represent how much each component contributes to the final distribution. It satisfies two properties:

$$\forall k, \omega_k > 0, \quad \text{and} \quad \sum_k \omega_k = 1$$

The properties ensure $p(\mathbf{x})$ is a properly normalized probability density function.
GMM as the marginal distribution of a joint distribution

The joint distribution between $x$ and $z$ (representing color) are

\[
\begin{align*}
p(x \mid z = 'red') &= N(x \mid \mu_1, \Sigma_1) \\
p(x \mid z = 'blue') &= N(x \mid \mu_2, \Sigma_2) \\
p(x \mid z = 'green') &= N(x \mid \mu_3, \Sigma_3)
\end{align*}
\]

The marginal distribution is thus

\[
p(x) = p('red') N(x \mid \mu_1, \Sigma_1) + p('blue') N(x \mid \mu_2, \Sigma_2) + p('green') N(x \mid \mu_3, \Sigma_3)
\]
Parameter estimation for Gaussian mixture models: the easy case with complete data

\[ \mathcal{D}' = \{x_n, z_n\}_{n=1}^{N}, \quad \theta = \arg \max \log \mathcal{D}' = \sum_n \log p(x_n, z_n) \]

Solution:

\[ \omega_k = \frac{\sum_n \gamma_{nk}}{\sum_k \sum_n \gamma_{nk}}, \quad \mu_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} x_n \]

\[ \Sigma_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (x_n - \mu_k)(x_n - \mu_k)^T \]

where \( \gamma_{nk} \) is 1 if \( z_n = k \).

**Understanding the intuition**

- For \( \omega_k \): count the number of data points whose \( z_n \) is \( k \) and divide by the total number of data points (note that \( \sum_k \sum_n \gamma_{nk} = N \))
- For \( \mu_k \): get all the data points whose \( z_n \) is \( k \), compute their mean
- For \( \Sigma_k \): get all the data points whose \( z_n \) is \( k \), compute their covariance
Trick: estimation with soft $\gamma_{nk}$

$$
\gamma_{nk} = p(z_n = k | x_n)
$$

$$
\omega_k = \frac{\sum_n \gamma_{nk}}{\sum_k \sum_n \gamma_{nk}}
$$

$$
\mu_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} x_n
$$

$$
\Sigma_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (x_n - \mu_k)(x_n - \mu_k)^T
$$

In other words, every data point $x_n$ is assigned to a component fractionally according to $p(z_n = k | x_n)$ — sometimes, this quantity is also called “responsibility”.
Iterative procedure

Since we do not know $\theta$ to begin with, we cannot compute the soft $\gamma_{nk}$. However, we can invoke an iterative procedure and alternate between estimating $\gamma_{nk}$ and using the estimated $\gamma_{nk}$ to compute the parameters:

- **Step 0:** guess $\theta$ with initial values
- **Step 1:** compute $\gamma_{nk}$ using the current $\theta$
- **Step 2:** update $\theta$ using the just computed $\gamma_{nk}$
- **Step 3:** go back to Step 1

Questions: i) is this procedure correct, for example, optimizing a sensible criteria? ii) practically, will this procedure ever stop instead of iterating forever?

The answer lies in the EM algorithm — a powerful procedure for model estimation with unknown data.
Outline

1. Administrative matters
2. Review of last lecture
3. EM Algorithm
4. Relation between K-means and GMMs
EM algorithm: motivation and setup

As a general procedure, EM is used to estimate parameters for probabilistic models with hidden/latent variables. Suppose the model is given by a joint distribution

\[ p(x|\theta) = \sum_z p(x, z|\theta) \]

where \( x \) is the observed random variable and \( z \) is hidden.

We are given data containing only the observed variable \( D = \{x_n\} \) where the corresponding hidden variable values \( z \) is not included. Our goal is to obtain the maximum likelihood estimate of \( \theta \). Namely, we choose

\[
\theta = \arg \max \log D = \arg \max \sum_n \log p(x_n|\theta)
\]

\[
= \arg \max \sum_n \log \sum_{z_n} p(x_n, z_n|\theta)
\]

The objective function \( \ell(\theta) \) is called incomplete log-likelihood.
Expected (complete) log-likelihood

The difficulty with incomplete log-likelihood is that it needs to sum over all possible values that $z_n$ can take, then take a logarithm. This log-sum format makes computation intractable. Instead, the EM algorithm uses a clever trick to change this into sum-log form.

To this end, we define the following

$$Q_q(\theta) = \sum_n \mathbb{E}_{z_n \sim q(z_n)} \log p(x_n, z_n | \theta)$$

$$= \sum_n \sum_{z_n} q(z_n) \log p(x_n, z_n | \theta)$$

which is called expected (complete) log-likelihood (with respect to $q(z)$. $q(z)$ is a distribution over $z$. Note that $Q_q(\theta)$ takes the form of sum-log, which turns out to be tractable.
Examples

Consider the previous model where \( x \) could be from 3 regions. We can choose \( q(z) \) any valid distribution. This will lead to different \( Q_q(\theta) \). Note that \( z \) here represents different colors.

- \( q(z = k) = 1/3 \) for any of 3 colors. This gives rise to

\[
Q_q(\theta) = \sum_n \frac{1}{3} \left[ \log p(x_n, 'red'|\theta) + \log p(x_n, 'blue'|\theta) + \log p(x_n, 'green'|\theta) \right]
\]

- \( q(z = k) = 1/2 \) for 'red' and 'blue', 0 for 'green'. This gives rise to

\[
Q_q(\theta) = \sum_n \frac{1}{2} \left[ \log p(x_n, 'red'|\theta) + \log p(x_n, 'blue'|\theta) \right]
\]
Which $q(z)$ to choose?

We will choose a special $q(z) = p(z|\mathbf{x}; \theta)$, i.e., the posterior probability of $z$. We define

$$Q(\theta) = Q_{z \sim p(z|x; \theta)}(\theta)$$

and we will show

$$\ell(\theta) = Q(\theta) + \sum_n \mathbb{H}[p(z|x_n; \theta)]$$

where $\mathbb{H}[p]$ is the entropy of the probabilistic distribution $p$:

$$\mathbb{H}[p(\mathbf{x})] = - \int p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x}$$
Proof

\[ Q(\theta) = \sum_n \sum_{z_n} p(z_n | x_n; \theta) \log p(x_n, z_n | \theta) \]

\[ = \sum_n \sum_{z_n} p(z_n | x_n; \theta) [\log p(x_n | \theta) + \log p(z_n | x_n; \theta)] \]

\[ = \sum_n \sum_{z_n} p(z_n | x_n; \theta) \log p(x_n | \theta) \]

\[ + \sum_n \sum_{z_n} p(z_n | x_n; \theta) \log p(z_n | x_n; \theta) \]

\[ = \sum_n \log p(x_n | \theta) \sum_{z_n} p(z_n | x_n; \theta) - \sum_n \text{H}[p(z | x_n; \theta)] \]

\[ = \sum_n \log p(x_n | \theta) - \sum_n \text{H}[p(z | x_n; \theta)] \]

\[ = \ell(\theta) - \sum_n \text{H}[p(z | x_n; \theta)] \]
A computable $Q(\theta)$

As before, $Q(\theta)$ cannot be computed, as it depends on the unknown parameter values $\theta$ to compute the posterior probability $p(z|x; \theta)$. Instead, we will use a known value $\theta^{OLD}$ to compute the expected likelihood

$$Q(\theta, \theta^{OLD}) = \sum_n \sum_{z_n} p(z_n|x_n; \theta^{OLD}) \log p(x_n, z_n|\theta)$$

Note that, in the above, the variable is $\theta$. $\theta^{OLD}$ is assumed to be known. By its definition, the following is true

$$Q(\theta) = Q(\theta, \theta^{OLD})$$

However, how does $Q(\theta, \theta^{OLD})$ relates to $\ell(\theta)$? We will show that

$$\ell(\theta) \geq Q(\theta, \theta^{OLD}) + \sum_n \mathbb{H}[p(z|x_n; \theta^{OLD})]$$

Thus, in a way, $Q(\theta)$ is better than $Q(\theta, \theta^{OLD})$ (because we have equality there) except that we cannot compute the former.
Proof

\[
\ell(\theta) = \sum_n \log \sum_{z_n} p(z|x_n; \theta^{\text{OLD}}) \frac{p(x_n, z_n|\theta)}{p(z|x_n; \theta^{\text{OLD}})}
\]

\[
\geq \sum_n \sum_{z_n} p(z|x_n; \theta^{\text{OLD}}) \log \frac{p(x_n, z_n|\theta)}{p(z|x_n; \theta^{\text{OLD}})}
\]

\[
= \sum_n \sum_{z_n} p(z|x_n; \theta^{\text{OLD}}) \log p(x_n, z_n|\theta)
\]

\[\quad - \sum_n \sum_{z_n} p(z|x_n; \theta^{\text{OLD}}) \log p(z|x_n; \theta^{\text{OLD}})\]

\[
= Q(\theta, \theta^{\text{OLD}}) + \sum_n \mathbb{H}[p(z|x_n; \theta^{\text{OLD}})]
\]

The inequality (\(\geq\)) is true because \(\log\) is a concave function:

\[
\log \sum_i w_i x_i \geq \sum_i w_i \log x_i, \quad \forall \ w_i \geq 0, \quad \sum_i w_i = 1
\]

And in our case, the \(w_i\) is \(p(z|x_n; \theta^{\text{OLD}})\).
Putting things together: auxiliary function

So far we have shown a lower bound on the log-likelihood

\[ \ell(\theta) \geq A(\theta, \theta^{\text{OLD}}) = Q(\theta, \theta^{\text{OLD}}) + \sum_n \mathbb{H}[p(z|x_n; \theta^{\text{OLD}})] \]

We will call the right-hand-side an \textit{auxiliary function}.

This auxiliary function has an important property. When \( \theta = \theta^{\text{OLD}} \),

\[ A(\theta, \theta) = \ell(\theta) \]
Use auxiliary function to increase log-likelihood

Suppose we have an initial guess $\theta^{OLD}$, then we maximize the auxiliary function

$$\theta^{NEW} = \arg \max_\theta A(\theta, \theta^{OLD})$$
Use auxiliary function to increase log-likelihood

Suppose we have an initial guess \( \theta^{\text{OLD}} \), then we maximize the auxiliary function

\[
\theta^{\text{NEW}} = \arg \max_{\theta} A(\theta, \theta^{\text{OLD}})
\]

With the new guess, we have

\[
\ell(\theta^{\text{NEW}}) \geq A(\theta^{\text{NEW}}, \theta^{\text{OLD}}) \geq A(\theta^{\text{OLD}}, \theta^{\text{OLD}}) = \ell(\theta^{\text{OLD}})
\]
Use auxiliary function to increase log-likelihood

Suppose we have an initial guess $\theta^{OLD}$, then we maximize the auxiliary function

$$\theta^{NEW} = \arg\max_\theta A(\theta, \theta^{OLD})$$

With the new guess, we have

$$\ell(\theta^{NEW}) \geq A(\theta^{NEW}, \theta^{OLD}) \geq A(\theta^{OLD}, \theta^{OLD}) = \ell(\theta^{OLD})$$

Repeating this process, we have

$$\ell(\theta^{EVEN NEWER}) \geq \ell(\theta^{NEW}) \geq \ell(\theta^{OLD})$$

where

$$\theta^{EVEN NEWER} = \arg\max_\theta A(\theta, \theta^{NEW})$$
Iterative and monotonic improvement

Thus, by maximizing the auxiliary function, we obtain a sequence of guesses

\[ \theta^{\text{OLD}}, \theta^{\text{NEW}}, \theta^{\text{EVEN NEWER}}, \ldots, \]

that will keep increasing the likelihood. This process will eventually stop if the likelihood is bounded from above (i.e., less than \(+\infty\)). This is the core of the EM algorithm.

Expectation-Maximization (EM)

- Step 0: Initialize \( \theta \) with \( \theta^{(0)} \)
- Step 1 (E-step): Compute the auxiliary function using the current value of \( \theta \)

\[ A(\theta, \theta^{(t)}) \]

- Step 2 (M-step): Maximize the auxiliary function

\[ \theta^{(t+1)} \leftarrow \arg \max A(\theta, \theta^{(t)}) \]

- Step 3: Increase \( t \) to \( t + 1 \) and go back to Step 1; or stop if \( \ell(\theta^{(t+1)}) \) does not improve \( \ell(\theta^{(t)}) \) much.
Remarks

- The EM procedure converges but only converges to a local optimum. Global optimum is not guaranteed to be found.
- The E-step depends on computing the posterior probability

\[ p(z_n|x_n; \theta^{(t)}) \]

- The M-step does not depend on the entropy term, so we need only to do the following

\[ \theta^{(t+1)} \leftarrow \text{arg max } A(\theta, \theta^{(t)}) = \text{arg max } Q(\theta, \theta^{(t)}) \]

We often call the last term \( Q \)-function.
Example: applying EM to GMMs

What is the E-step in GMM? We compute the responsibility

$$\gamma_{nk} = p(z = k | x_n; \theta^{(t)})$$

What is the M-step in GMM? The $Q$-function is

$$Q(\theta, \theta^{(t)}) = \sum_n \sum_k p(z = k | x_n; \theta^{(t)}) \log p(x_n, z = k | \theta)$$

$$= \sum_n \sum_k \gamma_{nk} \log p(x_n, z = k | \theta)$$

$$= \sum_k \sum_n \gamma_{nk} \log p(z = k)p(x_n | z = k)$$

$$= \sum_k \sum_n \gamma_{nk} [\log \omega_k + \log N(x_n | \mu_k, \Sigma_k)]$$

Hence, we have recovered the parameter estimation algorithm for GMMs, seen previously. (We still need to do the maximization to get $\theta^{(t+1)}$ — left as homework.)
Outline

1. Administrative matters
2. Review of last lecture
3. EM Algorithm
4. Relation between K-means and GMMs
GMMs and K-means

GMMs provide probabilistic interpretation for K-means. We have the following observation:

- Assume all Gaussian components have $\sigma^2 I$ as their covariance matrices
- Further assume $\sigma \to 0$
- Thus, we only need to estimate $\mu_k$, i.e., means
- Then, the EM for GMM parameter estimation simplifies to K-means.

For this reason, K-means is often called “hard” GMM or GMMs is called “soft” K-means. The soft posterior $\gamma_{nk}$ provides a probabilistic assignment for $x_n$ to cluster $k$ represented by the corresponding Gaussian distribution.