Latent Variable Models and Expectation Maximization

Oliver Schulte - CMPT 726

Bishop PRML Ch. 9
Learning Parameters to Probability Distributions

- We discussed probabilistic models at length
- Assignment 3: given fully observed training data, setting parameters $\theta_i$ for Bayes nets is straight-forward
- However, in many settings not all variables are observed (labelled) in the training data: $x_i = (x, h_i)$
  - e.g. Speech recognition: have speech signals, but not phoneme labels
  - e.g. Object recognition: have object labels (car, bicycle), but not part labels (wheel, door, seat)
- Unobserved variables are called latent variables

figs from Fergus et al.
Latent Variable Models: Pros

- Statistically powerful, often good predictions. Many applications:
  - Learning with **missing data**.
  - **Clustering**: “missing” cluster label for data points.
  - **Principal Component Analysis**: data points are generated in linear fashion from a small set of unobserved components. (more later)
  - **Matrix Factorization, Recommender Systems**:
    - Assign users to unobserved “user types”, assign items to unobserved “item types”.
    - Use similarity between user type, item type to predict preference of user for item.
    - Winner of $1M Netflix challenge.
- If latent variables have an intuitive interpretation (e.g., “action movies”, “factors”), discovers **new features**.
Latent Variable Models: Cons

- From a user’s point of view, like a black box if latent variables don’t have an intuitive interpretation.
- Statistically, hard to guarantee convergence to a correct model with more data (the identifiability problem).
- Harder computationally, usually no closed form for maximum likelihood estimates.
- However, the Expectation-Maximization algorithm provides a general-purpose local search algorithm for learning parameters in probabilistic models with latent variables.
Outline

K-Means

The Expectation Maximization Algorithm

EM Example: Gaussian Mixture Models
Outline

K-Means

The Expectation Maximization Algorithm

EM Example: Gaussian Mixture Models
Unsupervised Learning

- We will start with an unsupervised learning (clustering) problem:
- Given a dataset \( \{x_1, \ldots, x_N\} \), each \( x_i \in \mathbb{R}^D \), partition the dataset into \( K \) clusters
  - Intuitively, a cluster is a group of points, which are close together and far from others
Distortion Measure

- Formally, introduce prototypes (or cluster centers) \( \mu_k \in \mathbb{R}^D \)
- Use binary \( r_{nk}, 1 \) if point \( n \) is in cluster \( k \), 0 otherwise (1-of-\( K \) coding scheme again)
- Find \{\( \mu_k \}\}, \{\( r_{nk} \}\) to minimize distortion measure:

\[
J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \| x_n - \mu_k \|^2
\]
Minimizing Distortion Measure

- Minimizing $J$ directly is hard

\[
J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \| x_n - \mu_k \|^2
\]

- However, two things are easy
  - If we know $\mu_k$, minimizing $J$ wrt $r_{nk}$
  - If we know $r_{nk}$, minimizing $J$ wrt $\mu_k$

- This suggests an iterative procedure
  - Start with initial guess for $\mu_k$
  - Iteration of two steps:
    - Minimize $J$ wrt $r_{nk}$
    - Minimize $J$ wrt $\mu_k$
  - Rinse and repeat until convergence
Minimizing Distortion Measure

- Minimizing $J$ directly is hard
  
  $$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \| x_n - \mu_k \|^2$$

- However, two things are easy
  
  - If we know $\mu_k$, minimizing $J$ wrt $r_{nk}$
  - If we know $r_{nk}$, minimizing $J$ wrt $\mu_k$

- This suggests an iterative procedure
  
  - Start with initial guess for $\mu_k$
  - Iteration of two steps:
    - Minimize $J$ wrt $r_{nk}$
    - Minimize $J$ wrt $\mu_k$
  - Rinse and repeat until convergence
Minimizing Distortion Measure

- Minimizing $J$ directly is hard

\[
J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \| x_n - \mu_k \|^2
\]

- However, two things are easy
  - If we know $\mu_k$, minimizing $J$ wrt $r_{nk}$
  - If we know $r_{nk}$, minimizing $J$ wrt $\mu_k$

- This suggests an iterative procedure
  - Start with initial guess for $\mu_k$
  - Iteration of two steps:
    - Minimize $J$ wrt $r_{nk}$
    - Minimize $J$ wrt $\mu_k$
  - Rinse and repeat until convergence
Determining Membership Variables

- Step 1 in an iteration of K-means is to minimize distortion measure $J$ wrt cluster membership variables $r_{nk}$

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

- Terms for different data points $x_n$ are independent, for each data point set $r_{nk}$ to minimize

$$\sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2$$

- Simply set $r_{nk} = 1$ for the cluster center $\mu_k$ with smallest distance
Determining Membership Variables

- Step 1 in an iteration of K-means is to minimize distortion measure $J$ wrt cluster membership variables $r_{nk}$

\[
J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2
\]

- Terms for different data points $x_n$ are independent, for each data point set $r_{nk}$ to minimize

\[
\sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2
\]

- Simply set $r_{nk} = 1$ for the cluster center $\mu_k$ with smallest distance
Determining Membership Variables

- Step 1 in an iteration of K-means is to minimize distortion measure $J$ wrt cluster membership variables $r_{nk}$

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

- Terms for different data points $\mathbf{x}_n$ are independent, for each data point set $r_{nk}$ to minimize

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

- Simply set $r_{nk} = 1$ for the cluster center $\mathbf{\mu}_k$ with smallest distance
Determining Cluster Centers

- Step 2: fix $r_{nk}$, minimize $J$ wrt the cluster centers $\mu_k$

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

switch order of sums

- So we can minimize wrt each $\mu_k$ separately

- Take derivative, set to zero:

$$2 \sum_{n=1}^{N} r_{nk} (x_n - \mu_k) = 0$$

$$\iff \mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$

i.e. mean of datapoints $x_n$ assigned to cluster $k$
Determining Cluster Centers

- Step 2: fix $r_{nk}$, minimize $J$ wrt the cluster centers $\mu_k$

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

switch order of sums

- So we can minimize wrt each $\mu_k$ separately
- Take derivative, set to zero:

$$2 \sum_{n=1}^{N} r_{nk} (x_n - \mu_k) = 0$$

$$\iff \mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$

i.e. mean of datapoints $x_n$ assigned to cluster $k$
K-means Algorithm

- Start with initial guess for $\mu_k$
- Iteration of two steps:
  - Minimize $J$ wrt $r_{nk}$
    - Assign points to nearest cluster center
  - Minimize $J$ wrt $\mu_k$
    - Set cluster center as average of points in cluster
- Rinse and repeat until convergence
K-means example
K-means example
K-Means

The Expectation Maximization Algorithm

EM Example: Gaussian Mixture Models

K-means example

(d)
K-means example
K-means example

(f)
K-means example

(g)
K-means example
Next step doesn’t change membership – stop
K-means Convergence

- Repeat steps until no change in cluster assignments
- For each step, value of $J$ either goes down, or we stop
- Finite number of possible assignments of data points to clusters, so we are guaranteed to converge eventually
- Note it may be a local maximum rather than a global maximum to which we converge
K-means clustering on pixel colour values

Pixels in a cluster are coloured by cluster mean

Represent each pixel (e.g. 24-bit colour value) by a cluster number (e.g. 4 bits for $K = 10$), compressed version

This technique known as **vector quantization**

- Represent vector (in this case from RGB, $\mathbb{R}^3$) as a single discrete value
K-Means The Expectation Maximization Algorithm

EM Example: Gaussian Mixture Models

K-means Generalized: the set-up

Let’s generalize the idea. Suppose we have the following set-up.

- $X$ denotes all observed variables (e.g., data points).
- $Z$ denotes all latent (hidden, unobserved) variables (e.g., cluster means).
- $J(X, Z|\theta)$ where $J$ measures the “goodness” of an assignment of latent variable models given the data points and parameters $\theta$.
  - e.g., $J$ = -dispersion measure.
  - parameters = assignment of points to clusters.
- It’s easy to maximize $J(X, Z|\theta)$ wrt $\theta$ for fixed $Z$.
- It’s easy to maximize $J(X, Z|\theta)$ wrt $Z$ for fixed $\theta$. 
K-mean Generalized: The Algorithm

The fact that conditional maximization is simple suggests an iterative algorithm.

1. Guess an initial value for latent variables $Z$.
2. Repeat until convergence:
   2.1 Find best parameter values $\theta$ given the current guess for the latent variables. Update the parameter values.
   2.2 Find best value for latent variables $Z$ given the current parameter values. Update the latent variable values.
Outline

K-Means

The Expectation Maximization Algorithm

EM Example: Gaussian Mixture Models
EM Algorithm: The set-up

- We assume a probabilistic model, specifically the complete-data likelihood function $p(X, Z|\theta)$.
- “Goodness” of the model is the log-likelihood $\ln p(X, Z|\theta)$.
- Key difference: instead of guessing a single best value for latent variables given current parameter settings, we use the conditional distribution $p(Z|X, \theta^{old})$ over latent variables.
- Given a latent variable distribution, parameter values $\theta$ are evaluated by taking the expected “goodness” $\ln p(X, Z|\theta)$ over all possible latent variable settings.
EM Algorithm: The procedure

1. Guess an initial parameter setting $\theta^{old}$.
2. Repeat until convergence:
3. The E-step: Evaluate $p(Z|X, \theta^{old})$. (Ideally, find a closed form as a function of $Z$).
4. The M-step:
   4.1 Evaluate the function
   $$Q(\theta, \theta^{old}) = \sum_Z p(Z|X, \theta^{old}) \ln p(X, Z|\theta).$$
   4.2 Maximize $Q(\theta, \theta^{old})$ wrt $\theta$. Update $\theta^{old}$.
5. This procedure is guaranteed to increase at each step. the data log-likelihood
   $$\ln p(X|\theta) = \sum_Z \ln p(X, Z|\theta).$$
6. Therefore converges to local log-likelihood maximum. More theoretical analysis in text.
Outline

K-Means

The Expectation Maximization Algorithm

EM Example: Gaussian Mixture Models
In the K-means algorithm, a hard assignment of points to clusters is made. However, for points near the decision boundary, this may not be such a good idea. Instead, we could think about making a soft assignment of points to clusters.
The Gaussian mixture model (or mixture of Gaussians MoG) models the data as a combination of Gaussians.

- a: constant density contours. b: marginal probability $p(x)$. c: surface plot.
- Widely used general approximation for multi-modal distributions.
Gaussian Mixture Model

- Above shows a dataset generated by drawing samples from three different Gaussians.
Generative Model

- The mixture of Gaussians is a generative model
- To generate a datapoint $x_n$, we first generate a value for a discrete variable $z_n \in \{1, \ldots, K\}$
- We then generate a value $x_n \sim \mathcal{N}(x|\mu_k, \Sigma_k)$ for the corresponding Gaussian
Graphical Model

- Full graphical model using plate notation
  - Note $z_n$ is a latent variable, unobserved
- Need to give conditional distributions $p(z_n)$ and $p(x_n|z_n)$
- The one-of-$K$ representation is helpful here: $z_{nk} \in \{0, 1\}$, $z_n = (z_{n1}, \ldots, z_{nK})$
Graphical Model - Latent Component Variable

- Use a Bernoulli distribution for $p(z_n)$
  - i.e. $p(z_{nk} = 1) = \pi_k$
  - Parameters to this distribution $\{\pi_k\}$
  - Must have $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$

$$p(z_n) = \prod_{k=1}^{K} \pi_k^{z_{nk}}$$
Graphical Model - Observed Variable

- Use a **Gaussian distribution** for $p(x_n|z_n)$
  - Parameters to this distribution $\{\mu_k, \Sigma_k\}$

\[
p(x_n|z_{nk} = 1) = \mathcal{N}(x_n|\mu_k, \Sigma_k)
\]

\[
p(x_n|z_n) = \prod_{k=1}^{K} \mathcal{N}(x_n|\mu_k, \Sigma_k)^{z_{nk}}
\]
• The full joint distribution is given by:

\[
p(x, z) = \prod_{n=1}^{N} p(z_n)p(x_n | z_n)
\]

\[
= \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} \mathcal{N}(x_n | \mu_k, \Sigma_k)^{z_{nk}}
\]
MoG Marginal over Observed Variables

• The marginal distribution $p(x_n)$ for this model is:

$$p(x_n) = \sum_{z_n} p(x_n, z_n) = \sum_{z_n} p(z_n)p(x_n|z_n)$$

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

• A mixture of Gaussians
To apply EM, need the conditional distribution 
\[ p(z_{nk} = 1 | x_n, \theta) \]
where \( \theta \) are the model parameters.

It is denoted by \( \gamma(z_{nk}) \) can be computed as:

\[
\gamma(z_{nk}) \equiv p(z_{nk} = 1 | x_n) = \frac{\pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}
\]

\( \gamma(z_{nk}) \) is the responsibility of component \( k \) for datapoint \( n \)
MoG Conditional over Latent Variable

- To apply EM, need the conditional distribution \( p(z_{nk} = 1 | x_n, \theta) \) where \( \theta \) are the model parameters.
- It is denoted by \( \gamma(z_{nk}) \) can be computed as:

\[
\gamma(z_{nk}) \equiv p(z_{nk} = 1 | x_n) = \frac{p(z_{nk} = 1)p(x_n | z_{nk} = 1)}{\sum_{j=1}^{K} p(z_{nj} = 1)p(x_n | z_{nj} = 1)} \cdot \frac{\pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}
\]

- \( \gamma(z_{nk}) \) is the responsibility of component \( k \) for datapoint \( n \)

\[(b)\]

\[(c)\]
• To apply EM, need the conditional distribution $p(z_{nk} = 1 | x_n, \theta)$ where $\theta$ are the model parameters.
• It is denoted by $\gamma(z_{nk})$ can be computed as:

$$\gamma(z_{nk}) \equiv p(z_{nk} = 1 | x_n) = \frac{p(z_{nk} = 1)p(x_n | z_{nk} = 1)}{\sum_{j=1}^{K} p(z_{nj} = 1)p(x_n | z_{nj} = 1)} = \frac{\pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}$$

• $\gamma(z_{nk})$ is the responsibility of component $k$ for datapoint $n$
EM for Gaussian Mixtures: E-step

The complete-data log-likelihood is

$$\ln p(X, Z|\theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} [\ln \pi_k + \ln \mathcal{N}(x_n|\mu_k, \Sigma_k)].$$

**E step**: Calculate responsibilities using current parameters $\theta^{old}$:

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$$

The $z_n$ vectors assigning data point $n$ to components are independent of each other.

Therefore under the posterior distribution $p(z_{nk} = 1|x_n, \theta)$ the expected value of $z_{nk}$ is $\gamma(z_{nk})$. 
EM for Gaussian Mixtures: M-step

- **M step**: Because of the independence of the component assignments, we can calculate the expectation wrt the component assignments by using the expectations of the component assignments.

- So \( Q(\theta, \theta^{old}) = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) [\ln \pi_k + \ln \mathcal{N}(x_n|\mu_k, \Sigma_k)] \).

- Maximizing \( Q(\theta, \theta^{old}) \) with respect to the model parameters is more or less straightforward.
EM for Gaussian Mixtures II

- Initialize parameters, then iterate:
  - **E step**: Calculate responsibilities using current parameters
    \[
    \gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}
    \]
  - **M step**: Re-estimate parameters using these \(\gamma(z_{nk})\)
    \[
    N_k = \sum_{n=1}^{N} \gamma(z_{nk})
    \]
    \[
    \mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n
    \]
    \[
    \Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (x_n - \mu_k)(x_n - \mu_k)^T
    \]
    \[
    \pi_k = \frac{N_k}{N}
    \]
- Think of \(N_k\) as effective number of points in component \(k\).
EM for Gaussian Mixtures II

- Initialize parameters, then iterate:
  - **E step**: Calculate responsibilities using current parameters
    \[
    \gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}
    \]
  - **M step**: Re-estimate parameters using these \(\gamma(z_{nk})\)
    \[
    N_k = \sum_{n=1}^{N} \gamma(z_{nk})
    \]
    \[
    \mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n
    \]
    \[
    \Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T
    \]
    \[
    \pi_k = \frac{N_k}{N}
    \]
- Think of \(N_k\) as effective number of points in component \(k\).
MoG EM - Example

- Same initialization as with K-means before
  - Often, K-means is actually used to initialize EM
• Calculate responsibilities $\gamma(z_{nk})$
MoG EM - Example

- Calculate model parameters \( \{\pi_k, \mu_k, \Sigma_k\} \) using these responsibilities
MoG EM - Example

- Iteration 2
MoG EM - Example

- Iteration 5
MoG EM - Example

- Iteration 20 - converged
EM - Summary

- EM finds local maximum to likelihood

\[ p(X|\theta) = \sum_Z p(X,Z|\theta) \]

- Iterates two steps:
  - **E step** “fills in” the missing variables \( Z \) (calculates their distribution)
  - **M step** maximizes expected complete log likelihood (expectation wrt **E step** distribution)
Conclusion

- Readings: Ch. 9.1, 9.2, 9.4
- K-means clustering
- Gaussian mixture model
- What about K?
  - Model selection: either cross-validation or Bayesian version (average over all values for K)
- Expectation-maximization, a general method for learning parameters of models when not all variables are observed