1 Introduction

2 Everything observed

3 kMeans Clustering
   • Problem description
   • An algorithm

4 Mixtures of Gaussians
   • Problem description
   • Optimising the likelihood with latent variables

5 The General EM Algorithm
   • Definitions
   • MAP learning with EM
   • A closer look at the EM algorithm
   • Extensions to EM
1 Introduction

2 Everything observed

3 kMeans Clustering
   - Problem description
   - An algorithm

4 Mixtures of Gaussians
   - Problem description
   - Optimising the likelihood with latent variables

5 The General EM Algorithm
   - Definitions
   - MAP learning with EM
   - A closer look at the EM algorithm
   - Extensions to EM
Introduction

Last week, we have seen how the factorisation of probability distributions could be represented as graphs.

- We have seen algorithms to efficiently compute marginal probabilities on such graphs.
- We have seen how this could be used to compute conditional probabilities.
- However in we needed to assume that the parameters of the distributions were known.

Today, we see how to learn these parameters efficiently.
Learning the parameters

We want to find the parameters that maximise the likelihood

- MAP treatment is very similar
- (exact) full Bayesian treatment is often not tractable
- We’ll see approximations in lecture 13

How can we find the maximum of the likelihood?

- If everything is observed, it’s easy
- If we have latent variables, it’s hard
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>Everything observed</td>
</tr>
<tr>
<td>3</td>
<td>kMeans Clustering</td>
</tr>
<tr>
<td></td>
<td>• Problem description</td>
</tr>
<tr>
<td></td>
<td>• An algorithm</td>
</tr>
<tr>
<td>4</td>
<td>Mixtures of Gaussians</td>
</tr>
<tr>
<td></td>
<td>• Problem description</td>
</tr>
<tr>
<td></td>
<td>• Optimising the likelihood with latent variables</td>
</tr>
<tr>
<td>5</td>
<td>The General EM Algorithm</td>
</tr>
<tr>
<td></td>
<td>• Definitions</td>
</tr>
<tr>
<td></td>
<td>• MAP learning with EM</td>
</tr>
<tr>
<td></td>
<td>• A closer look at the EM algorithm</td>
</tr>
<tr>
<td></td>
<td>• Extensions to EM</td>
</tr>
</tbody>
</table>
Fully observed model

Example

![Graph showing inter-eruption time vs. duration with symbols and labels](graph.png)
Fully observed model

\[ p(x_i, z_i) = \begin{cases} 
  p(C_1) p(x_i|C_1) & \text{if } z_i = 1 \\
  p(C_2) p(x_i|C_2) & \text{if } z_i = 0 
\end{cases} \]

- Parametrisation:
  \[ p(C_1) = \pi \]
  \[ p(C_2) = 1 - \pi \]
  \[ p(x_i|C_1) = \mathcal{N}(x_i; \mu_1, \Sigma_1) \]
  \[ p(x_i|C_2) = \mathcal{N}(x_i; \mu_2, \Sigma_2) \]
  \[ \theta = \{ \pi, \mu_1, \Sigma_1, \mu_2, \Sigma_2 \} \]

(Complete) likelihood:

\[ p(x_i, z_i|\theta) = [p(C_1) \ p(x_i|C_1)]^{z_i} [p(C_2) \ p(x_i|C_2)]^{1-z_i} \]
Fully observed model

\[ p(x_i, z_i) = \begin{cases} 
  p(C_1) p(x_i|C_1) & \text{if } z_i = 1 \\
  p(C_2) p(x_i|C_2) & \text{if } z_i = 0 
\end{cases} \]

- Parametrisation:
  \[ p(C_1) = \pi \]
  \[ p(C_2) = 1 - \pi \]
  \[ p(x_i|C_1) = \mathcal{N}(x_i; \mu_1, \Sigma_1) \]
  \[ p(x_i|C_2) = \mathcal{N}(x_i; \mu_2, \Sigma_2) \]
  \[ \Theta = \{ \pi, \mu_1, \Sigma_1, \mu_2, \Sigma_2 \} \]

(Complete) likelihood:

\[ p(x_i, z_i|\Theta) = \left[ p(C_1) p(x_i|C_1) \right]^{z_i} \left[ p(C_2) p(x_i|C_2) \right]^{1-z_i} \]
Parameter optimisation

Complete likelihood:

\[
p(\{x_i, z_i\} | \theta) = \prod_{i=1}^{N} \left[ \pi \mathcal{N}(x_i; \mu_1, \Sigma_1) \right]^{z_i} \left[ (1 - \pi) \mathcal{N}(x_i; \mu_2, \Sigma_2) \right]^{1-z_i}
\]

Maximise: take logarithm, set derivative = 0

\[
\begin{align*}
\pi_1 &= \frac{\sum_{i=1}^{N} z_i}{N} \\
\mu_1 &= \frac{\sum_{i=1}^{N} z_i x_i}{\sum_{i=1}^{N} z_i} \\
\Sigma_1 &= \frac{\sum_{i=1}^{N} z_i x_i x_i^\top}{\sum_{i=1}^{N} z_i} - \mu_1 \mu_1^\top \\
\pi_2 &= \frac{\sum_{i=1}^{N} (1 - z_i)}{N} \\
\mu_2 &= \frac{\sum_{i=1}^{N} (1 - z_i) x_i}{\sum_{i=1}^{N} z_i} \\
\Sigma_2 &= \frac{\sum_{i=1}^{N} (1 - z_i) x_i x_i^\top}{\sum_{i=1}^{N} z_i} - \mu_2 \mu_2^\top
\end{align*}
\]
Hidden class label

So, what happens when the class label is not observed?

- List all possible assignments, pick best
Hidden class label

So, what happens when the class label is not observed?

- List all possible assignments, pick best
Hidden class label

So, what happens when the class label is not observed?

- List all possible assignments, pick best
Hidden class label

So, what happens when the class label is not observed?

- List all possible assignments, pick best
Hidden class label

So, what happens when the class label is not observed?

- List all possible assignments, pick best
Hidden class label

So, what happens when the class label is not observed?

- List all possible assignments, pick best
Hidden class label

So, what happens when the class label is not observed?

- List all possible assignments, pick best
Hidden class label

So, what happens when the class label is not observed?

- List all possible assignments, pick best
Hidden class label

So, what happens when the class label is not observed?

- List all possible assignments, pick best
1 Introduction

2 Everything observed

3 kMeans Clustering
   • Problem description
   • An algorithm

4 Mixtures of Gaussians
   • Problem description
   • Optimising the likelihood with latent variables

5 The General EM Algorithm
   • Definitions
   • MAP learning with EM
   • A closer look at the EM algorithm
   • Extensions to EM
Example: kMeans clustering

Cluster data in two groups in an unsupervised manner:
Properties of clustering

In order to cluster the data, we need:

- Some representation of what a cluster looks like
  - Let’s assume for now that each cluster is fully defined by its centre.

- An assignment of each datapoint to one of the clusters
  - Let’s assume that this is defined by the Euclidean distance to the clusters’ centres.

The best configuration is the one where all datapoints are as close as possible to their cluster’s centre.
Properties of clustering

In order to cluster the data, we need:

- Some representation of what a cluster looks like
  - Let’s assume for now that each cluster is fully defined by its centre.

- An assignment of each datapoint to one of the clusters
  - Let’s assume that this is defined by the Euclidean distance to the clusters’ centres.

The best configuration is the one where all datapoints are as close as possible to their cluster’s centre.
In order to cluster the data, we need:

- Some representation of what a cluster looks like
  - Let’s assume for now that each cluster is fully defined by its centre.

- An assignment of each datapoint to one of the clusters
  - Let’s assume that this is defined by the Euclidean distance to the clusters’ centres.

The best configuration is the one where all datapoints are as close as possible to their cluster’s centre.
An exhaustive search for the optimal clustering is intractable and requires $C^N$ operations.

- Where $C$ is the number of clusters
- and $N$ is the number of datapoints.

How do we find the optimal clustering without exhaustive search?

Solve the clustering iteratively:

- Initialise the cluster means at random
- Repeat until convergence
  1. Assign each data point to the closest cluster mean
  2. Update each cluster’s centre according to the associated data
An exhaustive search for the optimal clustering is intractable and requires $C^N$ operations

- Where $C$ is the number of clusters
- and $N$ is the number of datapoints.

How do we find the optimal clustering without exhaustive search? Solve the clustering iteratively:

- Initialise the cluster means at random
- Repeat until convergence
  1. Assign each data point to the closest cluster mean
  2. Update each cluster’s centre according to the associated data
An iterative algorithm

Example: kMeans Clustering

- Duration
- Average distance
- Iteration
- Inter-erruption time
An iterative algorithm

Example: kMeans Clustering

- Duration
- Inter-erruption time
- Average distance

Intelligent Autonomous Systems
An iterative algorithm

Example: kMeans Clustering

- Inter-erruption time vs Duration
- Average distance vs Iteration
An iterative algorithm

Example: kMeans Clustering

-3 -1.5 0 1.5 3
Duration
-3
-1.5
0
1.5
3
Inter-erruption time
1 2 3 4 5 6
Iteration
0
2.5
5
Average distance
An iterative algorithm

Example: kMeans Clustering
An iterative algorithm

Example: kMeans Clustering

- Duration
- Inter-erruption time
- Average distance

Graph showing data points and iterative changes over iterations.
An iterative algorithm

Example: kMeans Clustering

![Graph showing kMeans Clustering results with durations and average distances over iterations. The plot includes various data points and lines indicating the progress of the algorithm.]
An iterative algorithm

Example: kMeans Clustering

- Duration
- Average distance
- Inter-erruption time

The figure shows a scatter plot and a line graph illustrating the progression of kMeans Clustering over iterations.
An iterative algorithm

Example: kMeans Clustering

Duration

Average distance

Inter-erruption time
An iterative algorithm

Example: kMeans Clustering
An iterative algorithm

Example: kMeans Clustering

- Duration
- Inter-erruption time
- Average distance

Graph showing the relationship between duration, inter-erruption time, and average distance with iterations.
Problems with kMeans

There are some disadvantages to kMeans:

- **Euclidean distance:**
  - Only useful for some types of data
  - Not robust to outliers
  - Sensitive to scaling of data
  - Solution: Other distance measures

- **Hard assignments** At each iteration, each datapoint is assigned to exactly one cluster, even for doubtful cases.
**k-means sensitivity to scaling**

**Example: kMeans Clustering**

- **Inter-erruption time**
- **Average distance**

---

**Intelligent Autonomous Systems**
k-means sensitivity to scaling

Example: kMeans Clustering

Inter-erruption time
Inter-erruption time
Average distance
Average distance
k-means sensitivity to scaling

Example: kMeans Clustering

- Inter-erruption time
- Average distance
k-means sensitivity to scaling

Example: kMeans Clustering

Inter-erruption time vs. Inter-erruption time

Average distance vs. Average distance
k-means sensitivity to scaling

Example: kMeans Clustering

- Inter-erruption time vs. Inter-erruption time
- Average distance vs. Average distance
k-means sensitivity to scaling

Example: kMeans Clustering

- Inter-erruption time vs. Inter-erruption time
- Average distance vs. Average distance
k-means sensitivity to scaling

Example: kMeans Clustering

Inter-erruption time
Inter-erruption time
Average distance
Average distance

Intelligent Autonomous Systems
k-means sensitivity to scaling
k-means sensitivity to scaling

Example: kMeans Clustering

- Inter-erruption time
- Average distance
**k-means sensitivity to scaling**

Example: kMeans Clustering

- Inter-erruption time
- Average distance

**Intelligent Autonomous Systems**
k-means sensitivity to scaling

Example: kMeans Clustering

Inter-erruption time vs. Average distance
Introduction

Everything observed

kMeans Clustering
  - Problem description
  - An algorithm

Mixtures of Gaussians
  - Problem description
  - Optimising the likelihood with latent variables

The General EM Algorithm
  - Definitions
  - MAP learning with EM
  - A closer look at the EM algorithm
  - Extensions to EM
A mixture of Gaussians is a linear combination of Gaussians:

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

(1)

where \( 0 \leq \pi_k \leq 1 \) and

\[ \sum_{k=1}^{K} \pi_k = 1 \]  

(2)
Mixtures of Gaussians

A mixture of Gaussians is a linear combination of Gaussians:

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

where \( 0 \leq \pi_k \leq 1 \) and

\[ \sum_{k=1}^{K} \pi_k = 1 \]
Mixtures of Gaussians

A mixture of Gaussians is a linear combination of Gaussians:

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

where $0 \leq \pi_k \leq 1$ and

$$\sum_{k=1}^{K} \pi_k = 1$$ \hspace{1cm} (2)$$
Alternative view of the mixture model

We introduce a binary random variable $z$ in 1-of-K encoding, the probability of $z$ can be written as

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

We choose the conditional distribution of $x$ given a $z_k$ as

$$p(x|z_k = 1) = \mathcal{N}(x|\mu_k, \Sigma_k), \text{ so that}$$

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

Then we have:

$$p(x) = \sum_z p(z)p(x|z) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$
Optimising the likelihood with latent variables

Similar to kMeans, finding the optimal parameters for $p(x)$ in a mixture of Gaussians is hard. However, with our new representation, we can now work with $p(x, z)$ rather than $p(x)$. In particular, consider the log-likelihood of $N$ datapoints $X$:

$$\ln p(X | \pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left[ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right]$$  \hspace{1cm} (3)

Setting the first derivative with respect to $\mu_k$ equal to zero, gives:

$$0 = - \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)}{\sum_{\ell} \pi_{\ell} \mathcal{N}(x_n | \mu_{\ell}, \Sigma_{\ell})} \Sigma_k^{-1} (x_n - \mu_k)$$  \hspace{1cm} (4)
Optimising the likelihood with latent variables

Optimising the likelihood

Similar to kMeans, finding the optimal parameters for $p(x)$ in a mixture of Gaussians is hard.

However, with our new representation, we can now work with $p(x, z)$ rather than $p(x)$. In particular, consider the log-likelihood of $N$ datapoints $X$:

$$\ln p(X|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left[ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right]$$

(3)

Setting the first derivative with respect to $\mu_k$ equal to zero, gives:

$$0 = -\sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{\ell} \pi_\ell \mathcal{N}(x_n|\mu_\ell, \Sigma_\ell)} \Sigma_k^{-1}(x_n - \mu_k)$$

(4)
Likelihood maximisation

The values for $\mu_k$, $\Sigma_k$ and $\pi_k$ that maximise the likelihood are:

$$
\pi_k = \frac{N_k}{N}
$$

$$
\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} p(z_k|x_n) x_n
$$

$$
\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} p(z_k|x_n)(x_n - \mu_k)(x_n - \mu_k)^T
$$

where we defined $N_k = \sum_{n=1}^{N} p(z_k|x_n)$

However $p(z_k|x_n)$ is a function of $\pi_1, \ldots, \pi_K, \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K$
Likelihood maximisation

The values for $\mu_k$, $\Sigma_k$ and $\pi_k$ that maximise the likelihood are:

$$
\pi_k = \frac{N_k}{N}
$$

$$
\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} p(z_k|x_n)x_n
$$

$$
\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} p(z_k|x_n)(x_n - \mu_k)(x_n - \mu_k)^T
$$

where we defined $N_k = \sum_{n=1}^{N} p(z_k|x_n)$

However $p(z_k|x_n)$ is a function of $\pi_1, \ldots, \pi_K, \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K$
The EM algorithm for Gaussian Mixtures

The Expectation-Maximisation algorithm is an iterative update where:

- **E-step** Compute the posterior probabilities of the latent variables given the data and the current parameters (also called *responsibilities*), $p(z_k|x_n, \theta)$

- **M-step** Optimise the expectation of the complete log-likelihood with respect to the parameters

In practice, stop when the increase in likelihood falls below a certain threshold.
Example: Mixtures of Gaussians

Intelligent Autonomous Systems
About EM for mixtures of Gaussians

Some things to notice

- The problem is ill-posed: consider a component $k$ with covariance $\sigma I$. If the mean of the component falls exactly on a data point, its contribution to the likelihood is

$$
\ln p(x_n|\mu_k, \Sigma_k) = \frac{1}{\sqrt{2\pi}\sigma}
$$

which, in the limit of $\sigma \to 0$ goes to infinity.

- A suitable prior on $\theta$ avoids this problem.

- The kMeans algorithm is equivalent with EM for a Gaussian mixture model, where the covariance is $\sigma I$ for all mixture components in the limit $\sigma \to 0$. 
Optimising the likelihood with latent variables

Example

Intelligent Autonomous Systems
1 Introduction

2 Everything observed

3 kMeans Clustering
   - Problem description
   - An algorithm

4 Mixtures of Gaussians
   - Problem description
   - Optimising the likelihood with latent variables

5 The General EM Algorithm
   - Definitions
   - MAP learning with EM
   - A closer look at the EM algorithm
   - Extensions to EM
The General EM Algorithm

In general, the EM algorithm is defined as follows. Optimising the complete log-likelihood

$$p(X, Z|\theta)$$

would be easy, but we only observe $X$. So let’s optimise our best estimate of the complete log-likelihood: the expectation of the complete log-likelihood under our current parameter estimates $\theta^{old}$:

$$Q(\theta, \theta^{old}) = \mathbb{E}_{p(Z|X, \theta^{old})}[\ln p(X, Z|\theta)]$$

$$= \sum_z p(Z|X, \theta) \ln p(X, Z|\theta)$$
The E and M steps

- In the E-step we evaluate the distribution of the latent variables

\[ p(Z|X, \theta^{old}) \] (9)

so that we can compute the expectation of the complete log-likelihood (although we do not need to compute that explicitly)

- In the M-step we maximise the complete log-likelihood with respect to \( \theta \)

\[ \theta^{new} \leftarrow \text{arg max}_\theta Q(\theta, \theta^{old}) \] (10)
If we incorporate a prior over $\theta$, $p(\theta)$, the EM algorithm changes slightly:

**E-Step**: $p(Z|X, \theta^{old})$ does not depend on $p(\theta^{old})$ since $\theta^{old}$ are given, so the E-step remains identical.

**M-Step**: We now optimise

$$
E \left[ \ln(p(X, Z|\theta)p(\theta)) \right]
$$

where $p(\theta)$ is constant with respect to $p(Z|X, \theta^{old})$, so that we get:

$$
\theta^{new} = \arg \max_{\theta} Q(\theta, \theta^{old}) + \ln p(\theta)
$$
Why EM works

Consider the log likelihood

\[ \ell(\theta) = \ln p(\mathbf{X}|\theta) = \sum_z \ln p(\mathbf{Z}, \mathbf{X}|\theta) \]  

(13)

\[ = \sum_z \ln[q(\mathbf{Z}) \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})}] \]  

(14)

\[ \geq \sum_z q(\mathbf{Z}) \ln[\frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})}] \]  

(15)

By Jensen’s inequality
Jensen’s inequality

\[ f(x) \]

\[ f(a) \]

\[ f(b) \]

\[ \lambda f(a) + (1 - \lambda)f(b) \]

\[ \lambda a + (1 - \lambda)b \]
Jensen’s inequality
Jensen’s inequality

\[ x \ln x \]

\[ -\ln x \]
Mutual entropy

Consider the quantity

$$\sum_{z} q(z) \ln \left[ \frac{p(x, z|\theta)}{q(z)} \right]$$

$$= \sum_{z} q(z) \left[ \ln p(z|x, \theta) + \ln p(x|\theta) - \ln q(z) \right]$$

$$= \sum_{z} q(z) \ln \frac{p(z|x, \theta)}{q(z)} + \ln p(x|\theta) \sum_{z} q(z)$$

$$= \ln p(x|\theta)$$

so that

$$\ln p(x|\theta) = \sum_{z} q(z) \ln \left[ \frac{p(x, z|\theta)}{q(z)} \right] + \left( \sum_{z} q(z) \ln \frac{p(z|x, \theta)}{q(z)} \right)$$

$$\quad \left( L(q, \theta) \right)$$

$$\quad \left( KL(q||p) \right)$$
During the E-step, we maximise $\mathcal{L}(q, \theta)$ with respect to $q(z)$, leaving $\theta^{old}$ untouched. Since $\ell(\theta^{old})$ does not depend on $q(z)$, this can only be achieved by setting

$$KL(q||p) = 0$$

in other words

$$q(z) = p(z|x, \theta)$$

so that

$$\mathcal{L}(q, \theta^{old}) = \ln p(x|\theta^{old})$$
A closer look at the EM algorithm

**M-Step**

During the M-step, we maximise $\mathcal{L}(q, \theta)$ with respect to $\theta$, leaving $q(z)$ untouched. Since $q(x) = p(z|x, \theta^{old})$,

$$
\mathcal{L}(q, \theta) = \sum_z p(z|x, \theta) \ln p(z, x|\theta) - \sum_z p(z|x, \theta) \ln p(z|x, \theta)
$$

$$
= Q(\theta, \theta) + H[q(z)]
$$

- Maximising $\mathcal{L}(q, \theta)$ with respect to $\theta$ changes $\ln p(z|x, \theta)$, so that $KL(q||p)$ increases.
- $p(x|\theta)$ therefore increases at least as much as $\mathcal{L}(q, \theta)$.
EM vs. Gradient Descent
Sometimes, the following extensions of EM are used:

- When the datapoints are independent, the responsibilities \( z_n \) depend on \( x_n \) and \( \theta \) only, so that the E and M step can be computed online rather than in batch. This can converge faster than the batch version.

- Sometimes the E-step or M-step (or both) remain intractable. Increasing the likelihood (rather than maximising it) still guarantees increasing the likelihood. This is called Generalised EM (GEM)
Wrap-up

Today, we have seen:

- Learning when we can do inference (Bishop, p. 439–441)
- Examples of the EM algorithm (Bishop, p. 423–439)
- A formal analysis of EM (Bishop, p. 450–453)
- Compared k-means with mixtures of Gaussians

Lab:

- Implement the EM algorithm for mixtures of Gaussians