Lecture 18.
Gaussian Mixture Model.
Expectation Maximization.

COMP90051 Statistical Machine Learning

Semester 2, 2019

Lecturer: Ben Rubinstein

This lecture

- Unsupervised learning
  - Diversity of problems
- Gaussian mixture model (GMM)
  - A probabilistic approach to clustering
  - The GMM model
  - GMM clustering as an optimisation problem
- The Expectation Maximization (EM) algorithm

Unsupervised Learning

A large branch of ML that concerns with learning the structure of the data in the absence of labels

Previously: Supervised learning

- Supervised learning: Overarching aim is making predictions from data
- We studied methods such as random forest, ANN and SVM in the context of this aim
- We had instances $\boldsymbol{x}_i \in \boldsymbol{R}^m$, $i = 1, \ldots, n$ and corresponding labels y_i as inputs, and the aim was to predict labels for new instances
- Can be viewed as a function approximation problem, but with a big caveat: ability to generalise is critical
- Bandits: a setting of partial supervision

Now: Unsupervised learning

- Next few lectures: unsupervised learning methods
- In unsupervised learning, there is no dedicated variable called a "label"
- Instead, we just have a set of points $\boldsymbol{x}_i \in \boldsymbol{R}^m$, $i = 1, \ldots, n$
- The aim of unsupervised learning is to explore the structure (patterns, regularities) of the data
- The aim of "exploring the structure" is vague

Unsupervised learning tasks

- Diversity of tasks fall into unsupervised learning category
  - Clustering (now)
  - Dimensionality reduction (soon)
  - Learning parameters of probabilistic models (before/now)
- Applications and related tasks are numerous :
  - Market basket analysis. E.g., use supermarket transaction logs to find items that are frequently purchased together
  - Outlier detection. E.g., find potentially fraudulent credit card transactions
  - Often unsupervised tasks in (supervised) ML pipelines

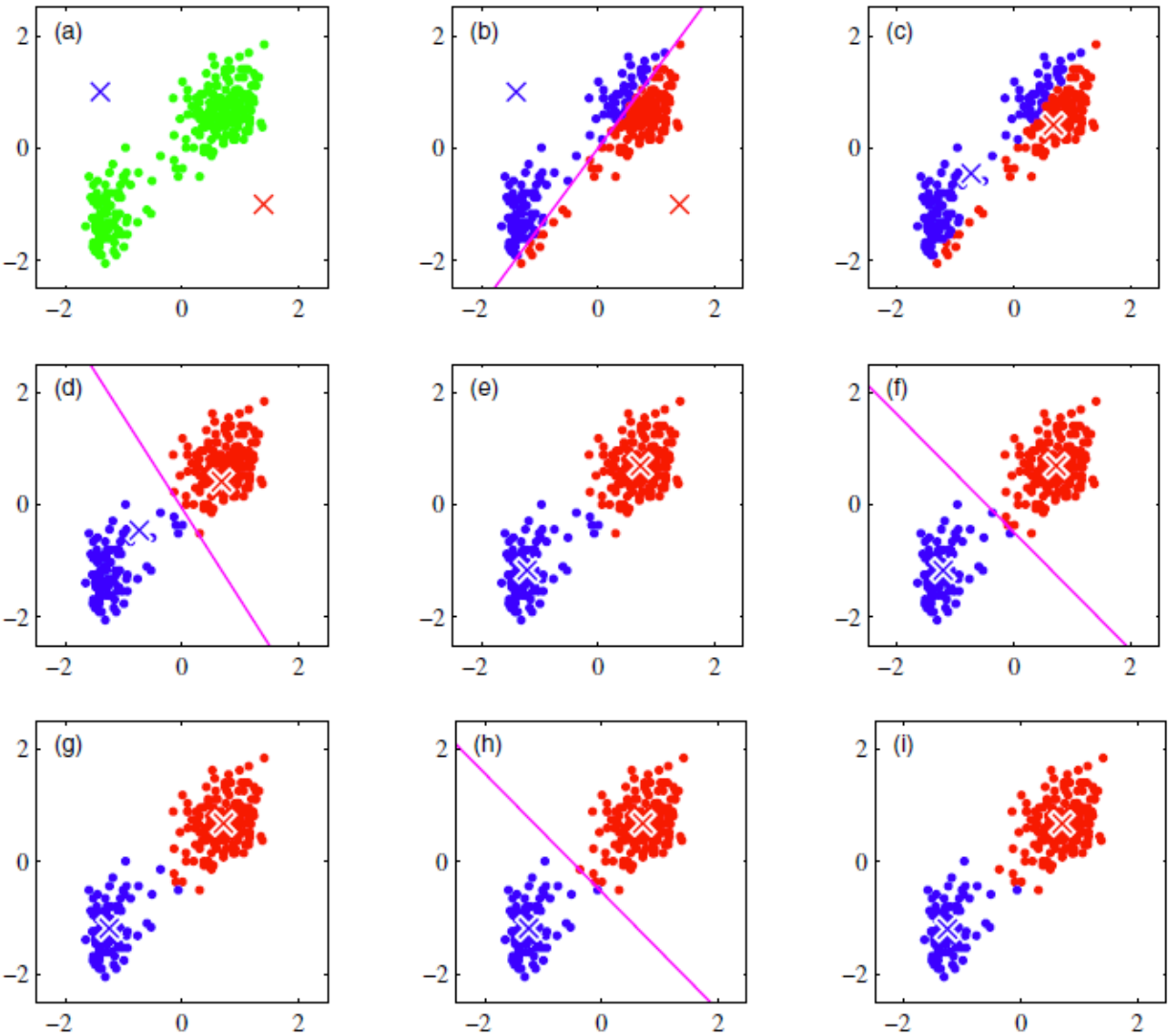
Refresher: k-means clustering

1. Initialisation: choose k cluster centroids randomly
2. Update:
   a) Assign points to the nearest* centroid
   b) Compute centroids under the current assignment
3. Termination: if no change then stop
4. Go to Step 2

*Distance represented by choice of metric typically L_2

Still one of the most popular data mining algorithms.

Refresher: k-means clustering



Requires specifying the number of clusters in advance

Measures "dissimilarity" using Euclidean distance

Finds "spherical" clusters

An iterative optimization procedure

Data: Old Faithful Geyser Data: waiting time between eruptions and the duration of eruptions

Figure: Bishop, Section 9.1

Gaussian Mixture Model

A probabilistic view of clustering

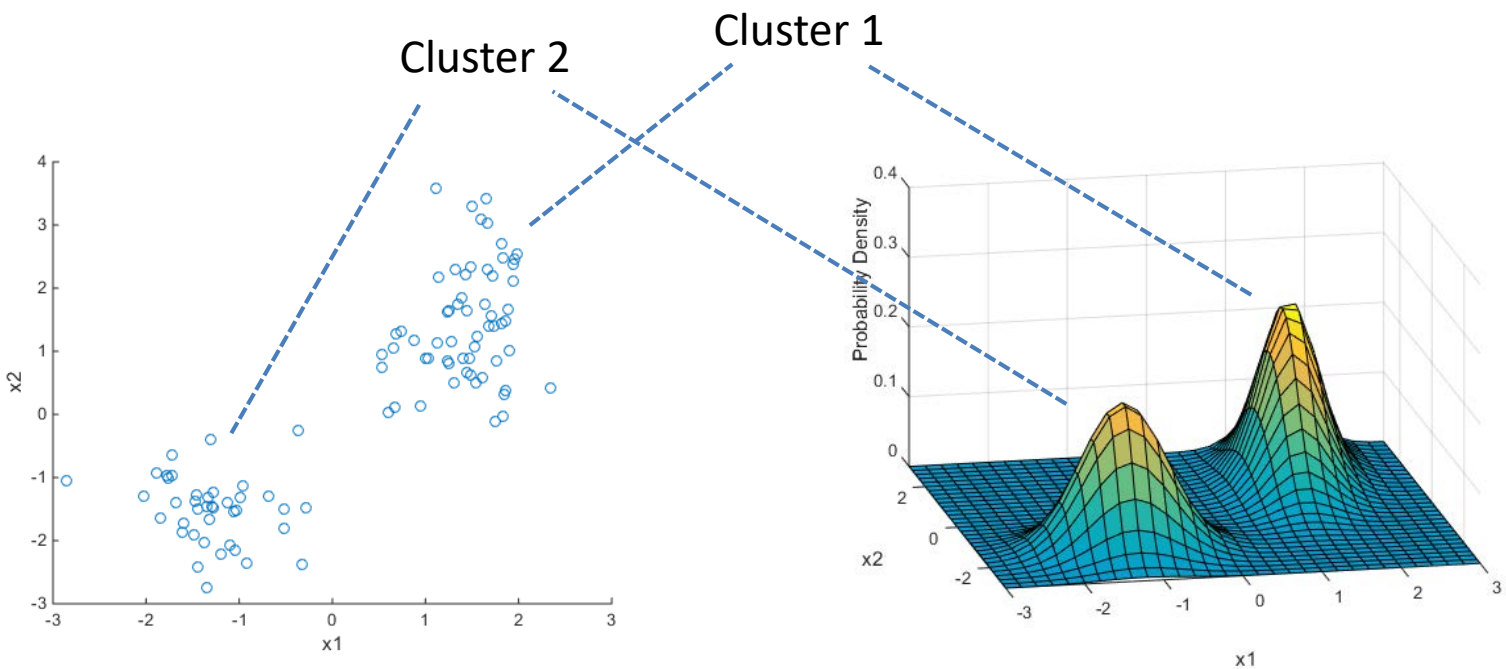
Modelling uncertainty in data clustering

- k-means clustering assigns each point to exactly one cluster
- Similar to k-means, a probabilistic mixture model requires the user to choose the number of clusters in advance
- Unlike k-means, the probabilistic model gives us a power to express uncertainly about the origin of each point
  - Each point originates from cluster c with probability w_c, $c = 1, \dots, k$
- That is, each point still originates from one particular cluster (aka component), but we are not sure from which one
- Next
  - Individual components modelled as Gaussians
  - Fitting illustrates general Expectation Maximization (EM) algorithm

Clustering: probabilistic model

Data points x_i are independent and identically distributed (i.i.d.) samples from a mixture of K distributions (components)

Each component in the mixture is what we call a cluster



In principle, we can adopt any probability distribution for the components, however, the normal distribution is a common modelling choice → Gaussian Mixture Model

Normal (aka Gaussian) distribution

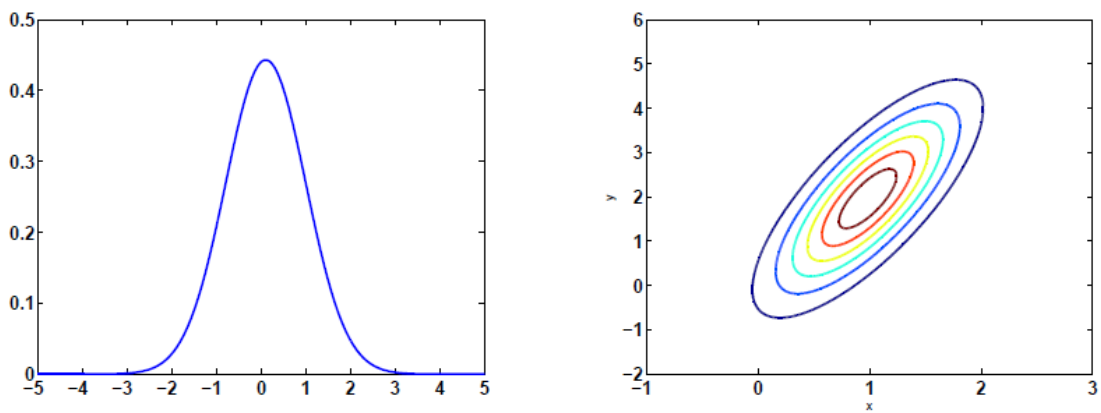
- Recall that a 1D Gaussian is

$$\mathcal{N}(x|\mu,\sigma) \equiv \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- And a d-dimensional Gaussian is

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) \equiv (2\pi)^{-\frac{d}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

  - $\boldsymbol{\Sigma}$ is a PSD symmetric $d \times d$ matrix, the covariance matrix
  - $|\boldsymbol{\Sigma}|$ denotes determinant
  - No need to memorize the full formula.



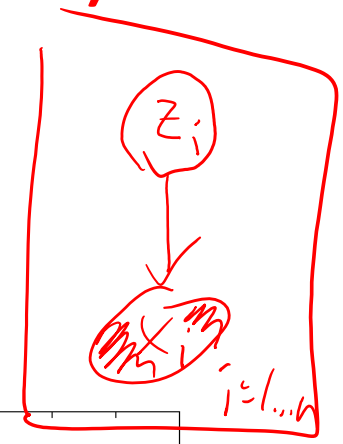
(a) 1-Dim (b) 2-Dim

Figure: Bishop

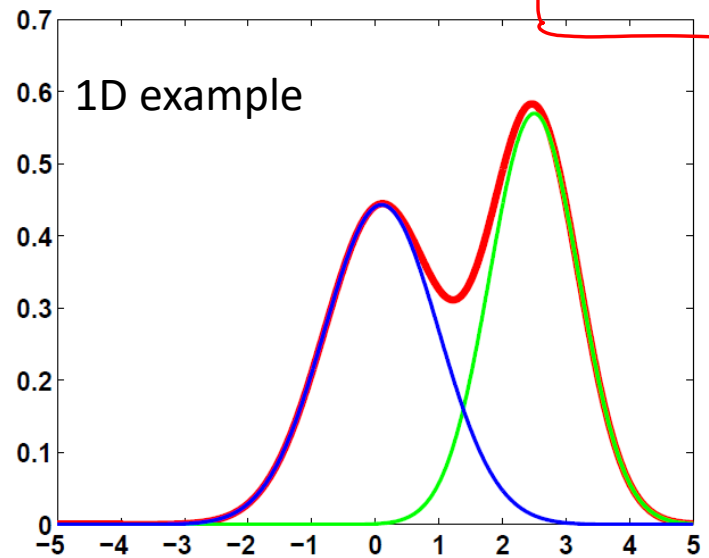
Gaussian mixture model (GMM)

- Gaussian mixture distribution (for one data point):

$$P(\boldsymbol{x}) \equiv \sum_{j=1}^{k} w_j \mathcal{N}\left(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right) \equiv \sum_{j=1}^{k} \mathrm{P}(C_j)\, P\left(\boldsymbol{x}|\boldsymbol{C}_j\right)$$



- $P(\boldsymbol{x}|\boldsymbol{C}_j) \equiv \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ is class/component conditional density (modeled as a Gaussian) for class j
- Here $\mathrm{P}(C_j) \geq 0$ and $\sum_{j=1}^{k} \mathrm{P}(C_j) = 1$
- Parameters of the model are $\mathrm{P}(C_j)$, $\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, j = 1, \dots, k$



Mixture and individual component densities are re-scaled for visualisation purposes

Figure: Bishop

Consider a GMM with five components for 3D data. How many independent scalar parameters does this model have?

$$49 = 6 \times 5 + 3 \times 5 + 4$$

$$50 = 6 \times 5 + 3 \times 5 + 5$$

w_c

$$65 = 9 \times 5 + 3 \times 5 + 5$$

μ_c

Clustering as model estimation

- Given a set of data points, we assume that data points are generated by a GMM
  - Each point in our dataset originates from the j-th normal distribution component with probability $w_j = \mathrm{P}(C_j)$
- Clustering now amounts to finding parameters of the GMM that "best explain" the observed data
- Call upon old friend MLE principle to find parameter values that maximise $p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$

Fitting the GMM

- Modelling the data points as independent, aim is to find $\boldsymbol{P}(\boldsymbol{C_j}), \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, j = 1, \dots, k$ that maximise

  $P(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = \prod_{i=1}^{n} \sum_{j=1}^{k} \mathrm{P}(C_j)\, P(\boldsymbol{x}_i | C_j)$

  where $P(\boldsymbol{x}|\boldsymbol{C_j}) \equiv \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$

  Can be solved analytically?

- Taking the derivative of this expression is pretty awkward, try the usual log trick

$$\log P(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = \sum_{i=1}^{n} \log\left(\sum_{j=1}^{k} \mathrm{P}(C_j)\, P\left(\boldsymbol{x}_i | \mathbf{C}_j\right)\right)$$

→ Expectation-Maximisation (EM)

Expectation Maximisation Algorithm

For a moment, let's put GMM problem aside – to come back to later.

Motivation of EM

- Consider a parametric probabilistic model $p(\boldsymbol{X}|\boldsymbol{\theta})$, where $\boldsymbol{X}$ denotes data and $\boldsymbol{\theta}$ denotes a vector of parameters
- According to MLE, we need to maximise $p(\boldsymbol{X}|\boldsymbol{\theta})$ as a function of $\boldsymbol{\theta}$
  - equivalently maximise $\log p(\boldsymbol{X}|\boldsymbol{\theta})$
- There can be a couple of issues with this task

1. Sometimes we don't observe some of the variables needed to compute the log likelihood
   - Example: GMM cluster membership is not known in advance
2. Sometimes the form of the log likelihood is inconvenient to work with
   - Example: taking a derivative of GMM log likelihood results in a cumbersome equation

MLE vs EM

- MLE is a frequentist *principle* that suggests that given a dataset, the "best" parameters to use are the ones that maximise the probability of the data
  - MLE is a way *to formally pose* the problem
- EM is an *algorithm*
  - EM is a way *to solve* the problem posed by MLE
  - Especially convenient under unobserved latent variables
- MLE can be found by other methods such as gradient descent (but gradient descent is not always the most convenient method)

Expectation-Maximisation (EM) Algorithm

- <u>Initialisation Step</u>:
  - Initialize K clusters: $C_1, \ldots, C_K$

    (μ_j, Σ_j) and $P(C_j)$ for each cluster j.

- <u>Iteration Step</u>:
  - Estimate the cluster of each datum

    $$p(C_j \mid x_i)$$

    → Expectation

  - Re-estimate the cluster parameters

    → Maximisation

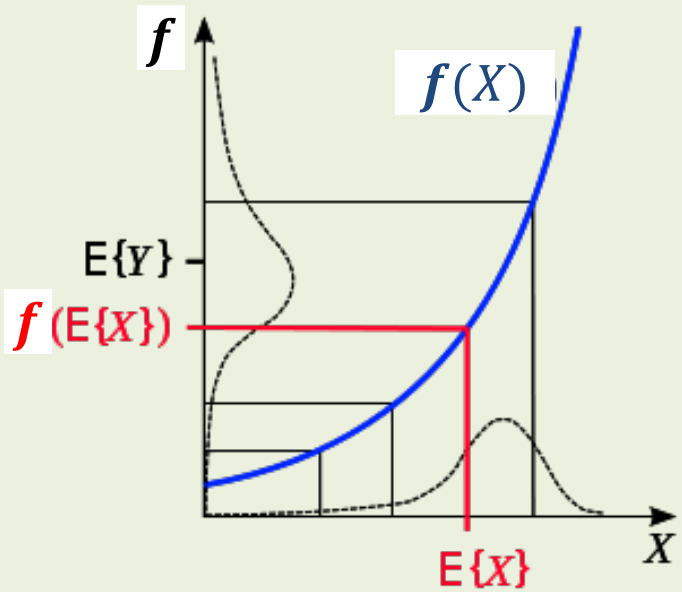
    $(\mu_j, \Sigma_j), p(C_j)$ for each cluster j

EM for GMM and generally

- EM is a general approach, goes beyond GMMs
  - Purpose: Implement MLE under latent variables **Z** ('latent' is fancy for 'missing')
- What are variables, parameters in GMMs?
  - Variables: Point locations **X** and cluster assignments **Z**
    - let z_i denote true cluster membership for each point $\boldsymbol{x}_i$, computing the likelihood with known values $\boldsymbol{z}$ is simplified (see next section)
  - Parameters: **θ** are cluster locations and scales
- What is EM really doing?
  - Coordinate ascent on a lower bound on the log-likelihood
    - M-step: ascent in modeled parameters **θ**
    - E-step: ascent in the marginal likelihood P(**Z**)
  - Each step moves towards a ***local*** optimum
  - Can get stuck, can need random restarts

Needed tool: Jensen's inequality

- Compares effect of averaging before and after applying a convex function: $f\big(Average(\boldsymbol{x})\big) \leq Average\big(f(\boldsymbol{x})\big)$
- Example:
  - Let f be some convex function, such as $f(x) = x^2$
  - Consider $\boldsymbol{x} = [1,2,3,4,5]'$, then $f(\boldsymbol{x}) = [1,4,9,16,25]'$
  - Average of input $Average(\boldsymbol{x}) = 3$
  - $f\big(Average(\boldsymbol{x})\big) = 9$
  - Average of output $Average\big(f(\boldsymbol{x})\big) = 12.4$
- Proof follows from the definition of convexity
  - Proof by induction
- General statement:
  - If $\boldsymbol{X}$ random variable, f is a convex function
  - $f(\mathbb{E}[\boldsymbol{X}]) \leq \mathbb{E}[f(\boldsymbol{X})]$



plot: MHz'as at Wikimedia Commons (public domain)

Putting the latent variables to use

- We want to maximise $\log p(\boldsymbol{X}|\boldsymbol{\theta})$. We don't observe $\boldsymbol{Z}$ (here discrete), but can introduce it nonetheless.
- $\boxed{\log p(\boldsymbol{X}|\boldsymbol{\theta})} = \log \sum_{\boldsymbol{Z}} p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$ ← Marginalisation (here $\sum_{\boldsymbol{Z}}$ … iterates over all possible values of $\boldsymbol{Z}$)
- $= \log \sum_{\boldsymbol{Z}} \left(p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}) \frac{p(\boldsymbol{Z})}{p(\boldsymbol{Z})} \right)$ ← Need $\boldsymbol{Z}$ to have non-zero marginal
- $= \log \sum_{\boldsymbol{Z}} \left(p(\boldsymbol{Z}) \frac{p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})}{p(\boldsymbol{Z})} \right)$
- $= \log \mathbb{E}_{\boldsymbol{Z}} \left[\frac{p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})}{p(\boldsymbol{Z})} \right]$
- $\geq \mathbb{E}_{\boldsymbol{Z}} \left[\log \frac{p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})}{p(\boldsymbol{Z})} \right]$ ← Jensen's inequality holds since $\log(\dots)$ is a concave function
- $= \boxed{\mathbb{E}_{\boldsymbol{Z}}[\log p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})] - \mathbb{E}_{\boldsymbol{Z}}[\log p(\boldsymbol{Z})]}$

Maximising the lower bound (1/2)

- $\log p(\boldsymbol{X}|\boldsymbol{\theta}) \geq \mathbb{E}_{\boldsymbol{Z}}[\log p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})] - \mathbb{E}_{\boldsymbol{Z}}[\log p(\boldsymbol{Z})]$
- The right hand side (RHS) is a lower bound on the original log likelihood
  - This holds for any $\boldsymbol{\theta}$ and any non zero $p(\boldsymbol{Z})$
- Intuitively, we want to push the lower bound up
- This lower bound is a function of two "variables" $\boldsymbol{\theta}$ and $p(\boldsymbol{Z})$. We want to maximise the RHS as a function of these two "variables"
- It is hard to optimise with respect to both at the same time, so EM resorts to an iterative procedure

Maximising the lower bound (2/2)

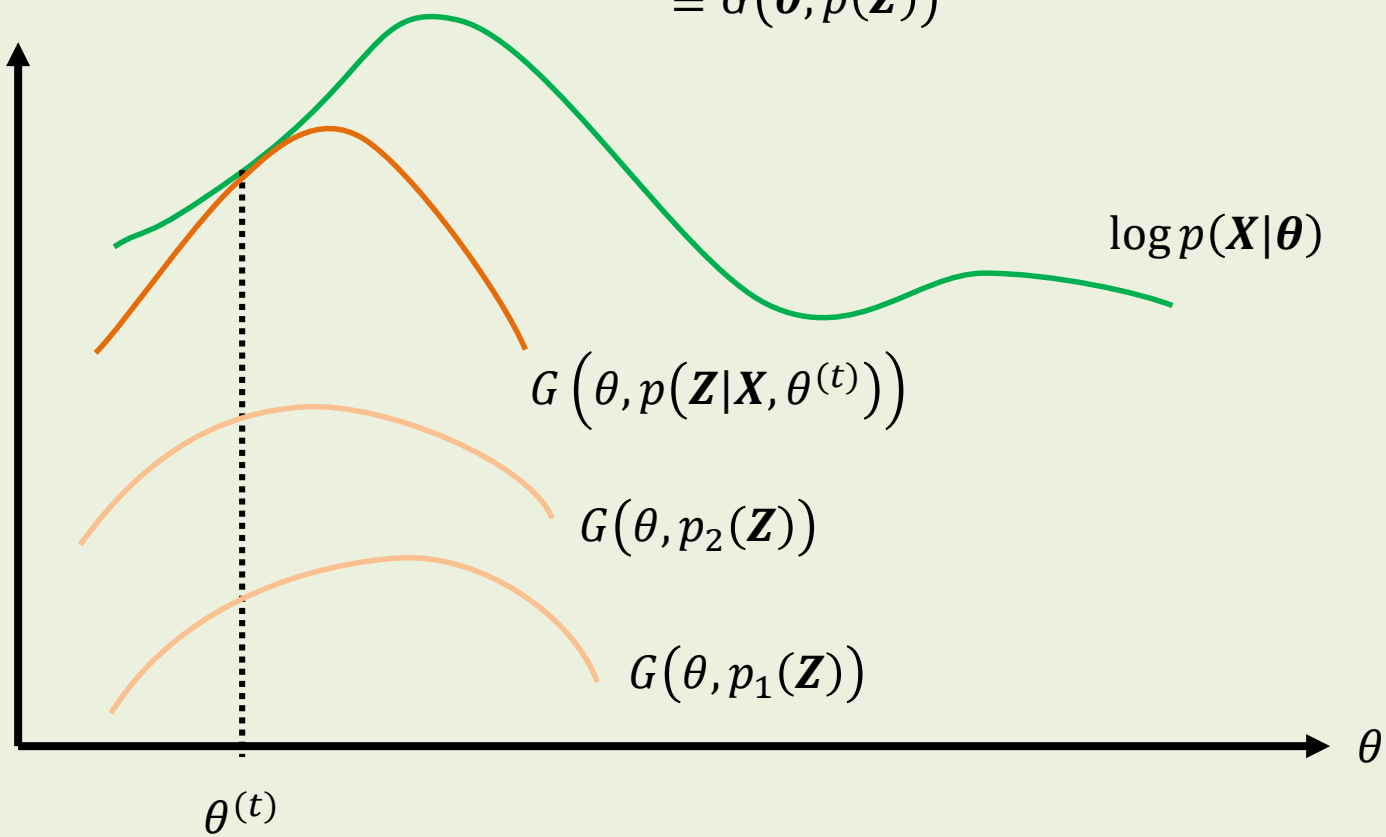
- $\log p(\boldsymbol{X}|\boldsymbol{\theta}) \geq \mathbb{E}_{\boldsymbol{Z}}[\log p(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\theta})] - \mathbb{E}_{\boldsymbol{Z}}[\log p(\boldsymbol{Z})]$
- EM is essentially coordinate ascent:
  - Fix $\boldsymbol{\theta}$ and optimise the lower bound for $p(\boldsymbol{Z})$
  - Fix $p(\boldsymbol{Z})$ and optimise for $\boldsymbol{\theta}$
- The convenience of EM comes from the following

we will prove this shortly

- For any point $\boldsymbol{\theta}^*$, it can be shown that setting $p(\boldsymbol{Z}) = p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta}^*)$ makes the lower bound tight
- For any $p(\boldsymbol{Z})$, the second term does not depend on $\boldsymbol{\theta}$
- When $p(\boldsymbol{Z}) = p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta}^*)$, the first term can usually be maximised as a function of $\boldsymbol{\theta}$ in a closed-form
  - If not, then probably don't use EM

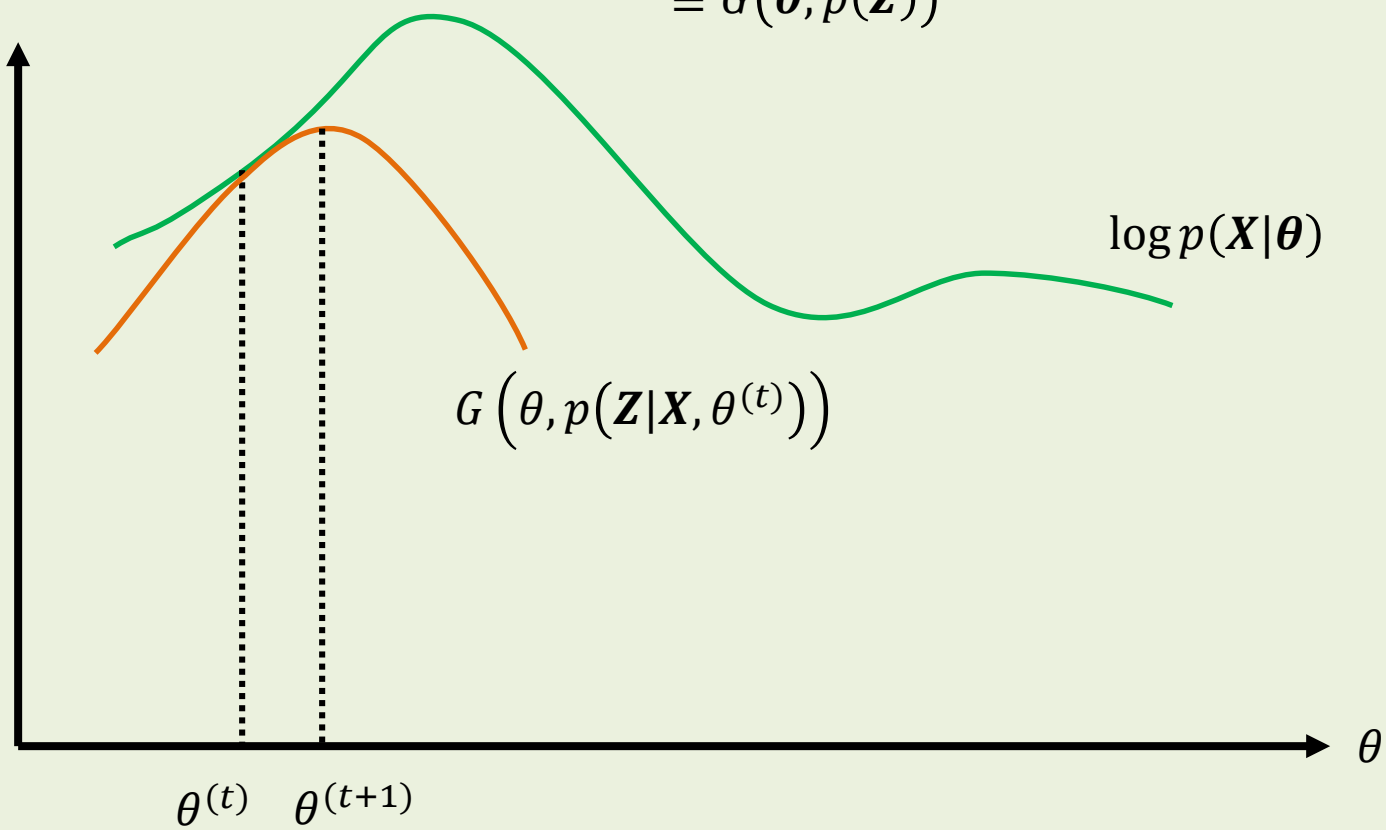
Example (1/3)

$$\log p(\boldsymbol{X}|\boldsymbol{\theta}) \geq \underbrace{\mathbb{E}_{\boldsymbol{Z}}[\log p(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\theta})] - \mathbb{E}_{\boldsymbol{Z}}[\log p(\boldsymbol{Z})]}_{\equiv\, G(\boldsymbol{\theta},\, p(\boldsymbol{Z}))}$$



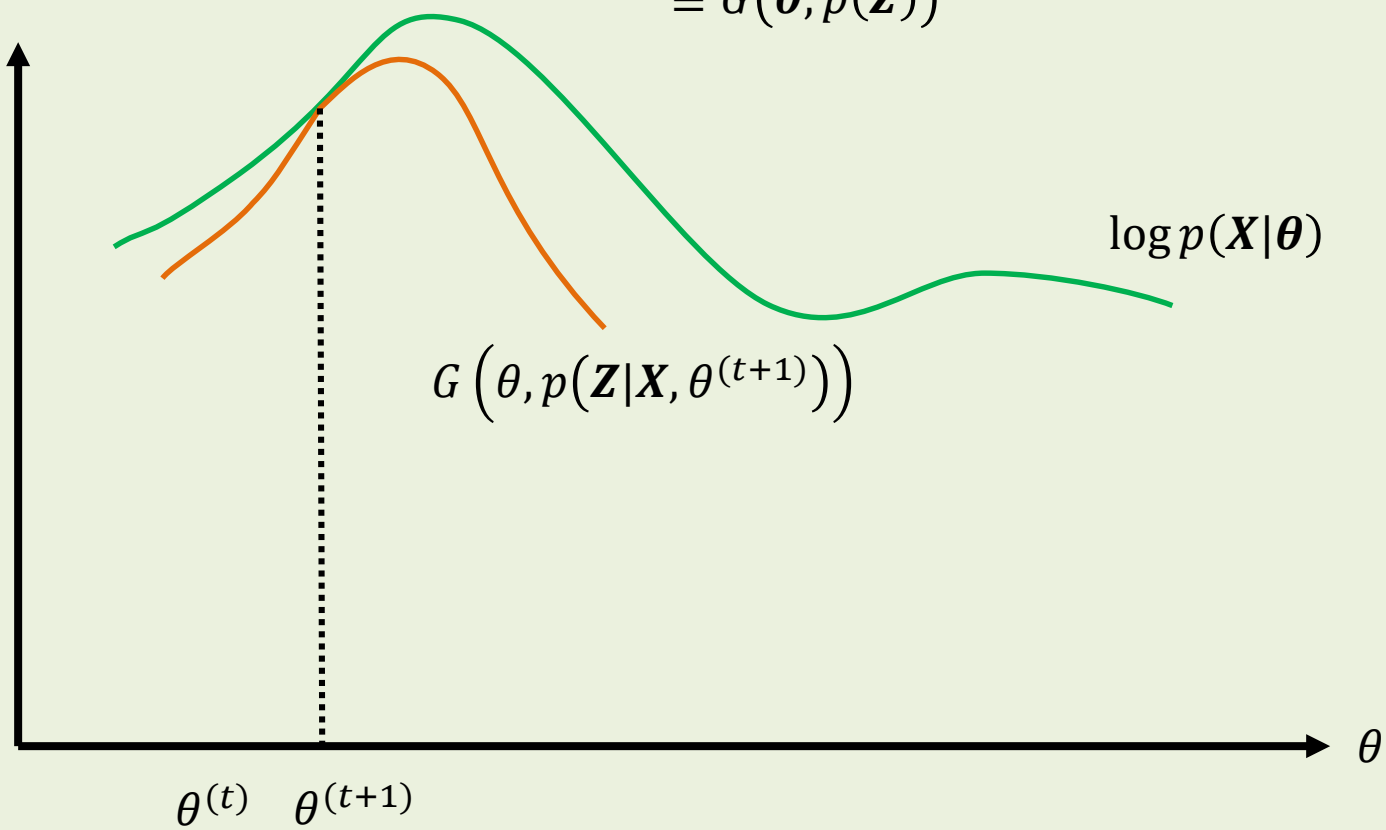
Example (2/3)

$$\log p(\boldsymbol{X}|\boldsymbol{\theta}) \geq \underbrace{\mathbb{E}_{\boldsymbol{Z}}[\log p(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\theta})] - \mathbb{E}_{\boldsymbol{Z}}[\log p(\boldsymbol{Z})]}_{\equiv\, G(\boldsymbol{\theta},\, p(\boldsymbol{Z}))}$$



Example (3/3)

$$\log p(\boldsymbol{X}|\boldsymbol{\theta}) \geq \underbrace{\mathbb{E}_{\boldsymbol{Z}}[\log p(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\theta})] - \mathbb{E}_{\boldsymbol{Z}}[\log p(\boldsymbol{Z})]}_{\equiv\, G(\boldsymbol{\theta},\, p(\boldsymbol{Z}))}$$



EM as iterative optimisation

1. Initialisation: choose (random) initial values of $\boldsymbol{\theta}^{(1)}$
2. Update:
   * E-step: compute $Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}\right) \equiv \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta}^{(t)}}[\log p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})]$
   * M-step: $\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}\right)$
3. Termination: if no change then stop
4. Go to Step 2

This algorithm will eventually stop (converge), but the resulting estimate can be only a local maximum

Maximising the lower bound (2/2)

- $\log p(\boldsymbol{X}|\boldsymbol{\theta}) \geq \mathbb{E}_{\boldsymbol{Z}}[\log p(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\theta})] - \mathbb{E}_{\boldsymbol{Z}}[\log p(\boldsymbol{Z})]$
- EM is essentially coordinate descent:
  - Fix $\boldsymbol{\theta}$ and optimise the lower bound for $p(\boldsymbol{Z})$
  - Fix $p(\boldsymbol{Z})$ and optimise for $\boldsymbol{\theta}$
- The convenience of EM follows from the following

we will prove this now

- For any point $\boldsymbol{\theta}^*$, it can be shown that setting $p(\boldsymbol{Z}) = p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta}^*)$ makes the lower bound tight
- For any $p(\boldsymbol{Z})$, the second term does not depend on $\boldsymbol{\theta}$
- When $p(\boldsymbol{Z}) = p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta}^*)$, the first term can usually be maximised as a function of $\boldsymbol{\theta}$ in a closed-form
  - If not, then probably don't use EM

Putting the latent variables in use

- We want to maximise $\log p(\boldsymbol{X}|\boldsymbol{\theta})$. We don't know $\boldsymbol{Z}$, but consider an arbitrary non-zero distribution $p(\boldsymbol{Z})$
- $\boxed{\log p(\boldsymbol{X}|\boldsymbol{\theta})} = \log \sum_{\boldsymbol{Z}} p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$

← Rule of marginal distribution (here $\sum_{\boldsymbol{Z}}$ … iterates over all possible values of $\boldsymbol{Z}$)

- $= \log \sum_{\boldsymbol{Z}} \left(p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}) \frac{p(\boldsymbol{Z})}{p(\boldsymbol{Z})} \right)$
- $= \log \sum_{\boldsymbol{Z}} \left(p(\boldsymbol{Z}) \frac{p(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\theta})}{p(\boldsymbol{Z})} \right)$
- $= \log \mathbb{E}_{\boldsymbol{Z}} \left[\frac{p(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\theta})}{p(\boldsymbol{Z})} \right]$

← Jensen's inequality holds since $\log(\dots)$ is a concave function

- $\boxed{\geq \mathbb{E}_{\boldsymbol{Z}} \left[\log \frac{p(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\theta})}{p(\boldsymbol{Z})} \right]}$
- $= \mathbb{E}_{\boldsymbol{Z}}[\log p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})] - \mathbb{E}_{\boldsymbol{Z}}[\log p(\boldsymbol{Z})]$

Setting a tight lower bound (1/2)

- $\log p(\boldsymbol{X}|\boldsymbol{\theta}) \geq \mathbb{E}_{\boldsymbol{Z}}\left[\log \frac{p(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\theta})}{p(\boldsymbol{Z})}\right]$
  - $= \mathbb{E}_{\boldsymbol{Z}}\left[\log \frac{p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})p(\boldsymbol{X}|\boldsymbol{\theta})}{p(\boldsymbol{Z})}\right]$ ← Chain rule of probability
  - $= \mathbb{E}_{\boldsymbol{Z}}\left[\log \frac{p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})}{p(\boldsymbol{Z})} + \log p(\boldsymbol{X}|\boldsymbol{\theta})\right]$
  - $= \mathbb{E}_{\boldsymbol{Z}}\left[\log \frac{p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})}{p(\boldsymbol{Z})}\right] + \mathbb{E}_{\boldsymbol{Z}}[\log p(\boldsymbol{X}|\boldsymbol{\theta})]$ ← Linearity of $\mathbb{E}[.]$
  - $= \mathbb{E}_{\boldsymbol{Z}}\left[\log \frac{p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})}{p(\boldsymbol{Z})}\right] + \log p(\boldsymbol{X}|\boldsymbol{\theta})$ ← $\mathbb{E}[.]$ of a constant
- $\log p(\boldsymbol{X}|\boldsymbol{\theta}) \geq \mathbb{E}_{\boldsymbol{Z}}\left[\log \frac{p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})}{p(\boldsymbol{Z})}\right] + \log p(\boldsymbol{X}|\boldsymbol{\theta})$

Setting a tight lower bound (2/2)

Ultimate aim: maximise this

Lower bound of what we want to maximise

$$\underbrace{\log p(\boldsymbol{X}|\boldsymbol{\theta})}_{\text{Ultimate aim: maximise this}} \geq \underbrace{\underbrace{\mathbb{E}_{\boldsymbol{Z}}\left[\log \frac{p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})}{p(\boldsymbol{Z})}\right]}_{\text{First, note that this term* } \leq 0} + \log p(\boldsymbol{X}|\boldsymbol{\theta})}_{\text{Lower bound of what we want to maximise}}$$

First, note that this term* ≤ 0

Second, note that if $p(\boldsymbol{Z}) = p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})$, then

$$\mathbb{E}_{p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})}\left[\log \frac{p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})}{p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})}\right] = \mathbb{E}_{p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})}[\log 1] = 0$$

For any $\boldsymbol{\theta}^*$, setting $p(\boldsymbol{Z}) = p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta}^*)$ maximises the lower bound on $\log p(\boldsymbol{X}|\boldsymbol{\theta}^*)$ and makes it tight

*Negative Kullback-Leibler divergence between $p(\boldsymbol{Z})$ and $p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\theta})$

Estimating Parameters of Gaussian Mixture Model

A classical application of the Expectation-Maximisation algorithm

Latent variables of GMM

- Let $z_1, \dots, z_n$ denote true origins of the corresponding points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$. Each z_i is a discrete variable that takes values in $1, \dots, k$, where k is a number of clusters
- Now compare the original log likelihood

$$\log p(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = \sum_{i=1}^{n} \log \left(\sum_{c=1}^{k} w_c \mathcal{N}(\boldsymbol{x}_i | \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \right)$$

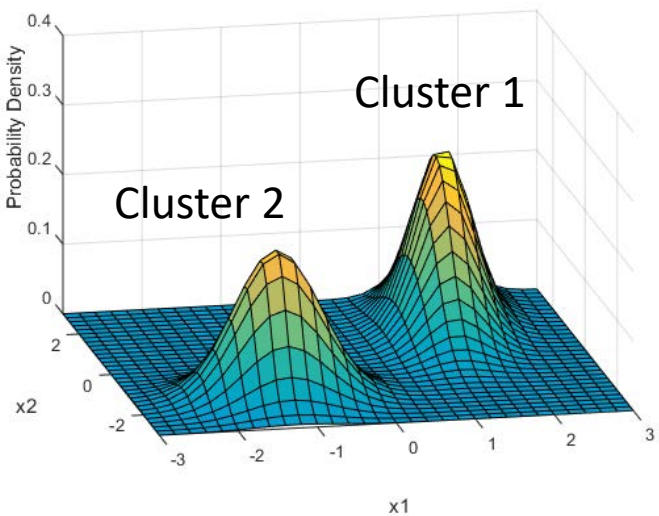
- With *complete log likelihood* (if we knew $\boldsymbol{z}$)

$$\log p(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n, \boldsymbol{z}) = \sum_{i=1}^{n} \log \left(w_{z_i} \mathcal{N}\left(\boldsymbol{x}_i | \boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i}\right) \right)$$

- Recall that taking a log of a normal density function results in a tractable expression

Handling uncertainty about $\boldsymbol{z}$

- We cannot compute complete log likelihood because we don't know $\boldsymbol{z}$
- EM algorithm handles this uncertainty replacing $\log p(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\theta})$ with expectation $\mathbb{E}_{\boldsymbol{z}|\boldsymbol{X},\boldsymbol{\theta}^{(t)}}[\log p(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\theta})]$
- This in turn requires the distribution of $p\left(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}\right)$ given current parameter estimates
- Assuming that z_i are pairwise independent, we need $P\left(z_i = c|\boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right)$
- E.g., suppose $\boldsymbol{x}_i = (-2, -2)$. What is the probability that this point originated from Cluster 1



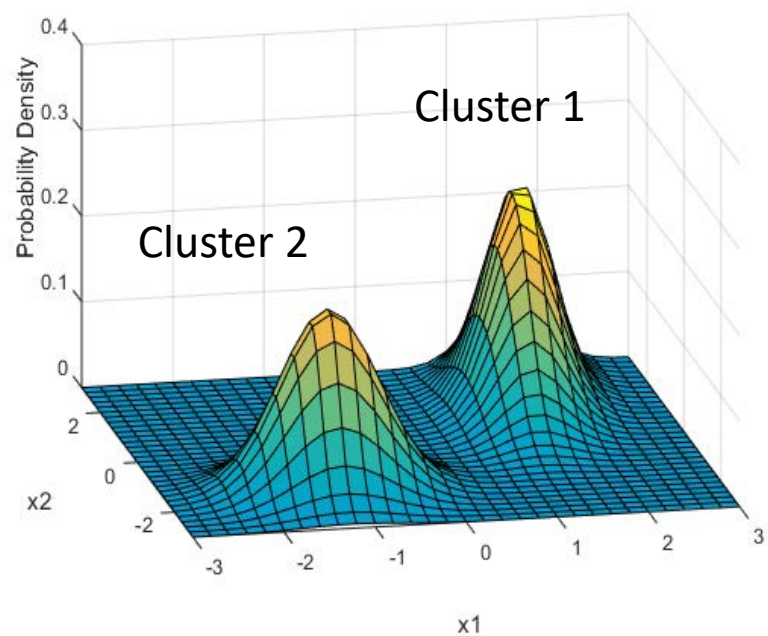
E-step: Cluster responsibilities

- Setting latent Z as originating cluster, yields (via Bayes rule)

$$P\left(z_i = c|\boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right) = \frac{w_c \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{l=1}^{k} w_l \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}$$

- This probability is called *responsibility* that cluster c takes for data point i

$$r_{ic} \equiv P\left(z_i = c|\boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right)$$



Expectation step for GMM

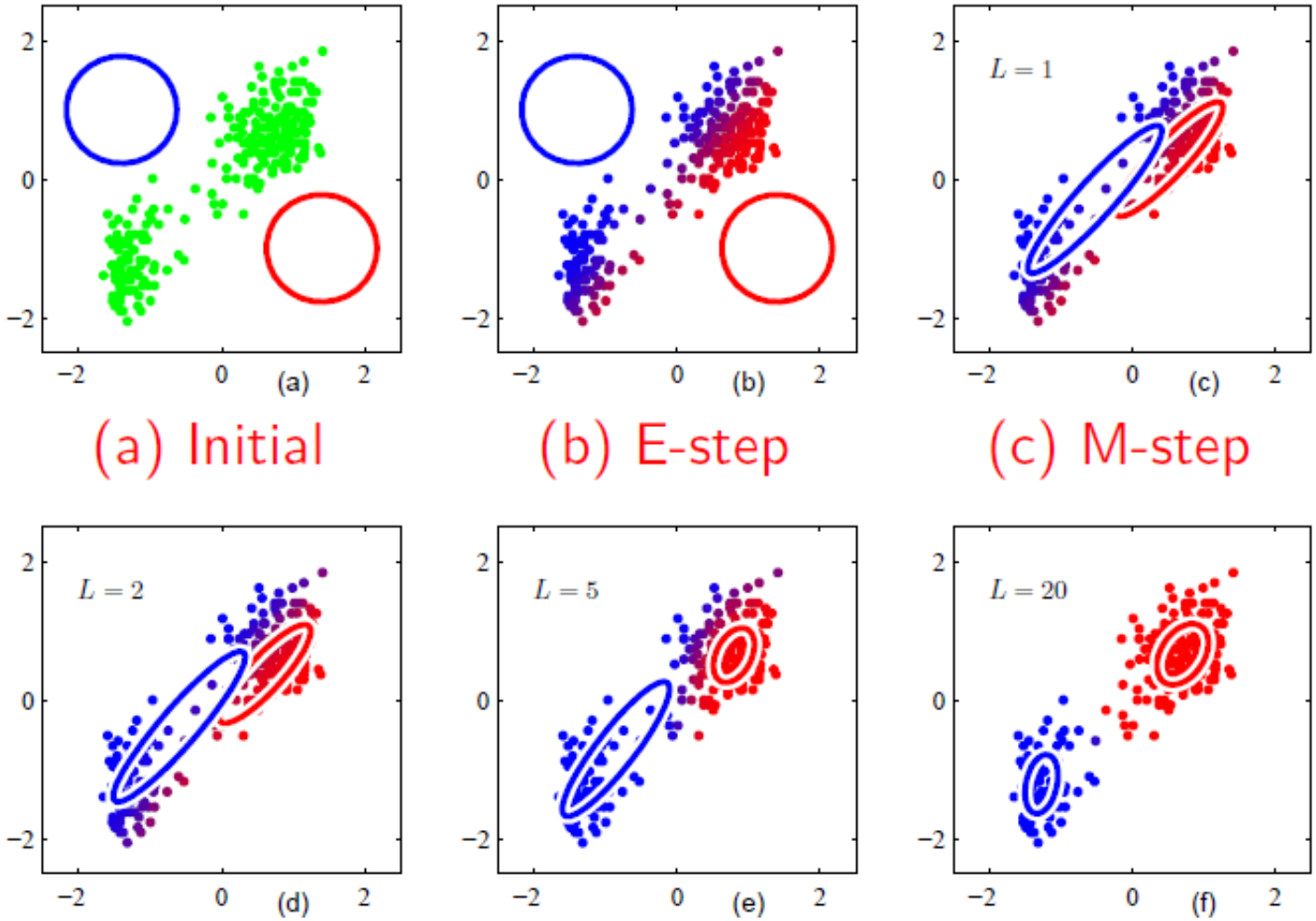
To simplify notation, we denote $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ as $\boldsymbol{X}$

- $Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}\right) \equiv \mathbb{E}_{\boldsymbol{z}|\boldsymbol{X},\boldsymbol{\theta}^{(t)}}[\log p(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\theta})]$
- $= \sum_{\boldsymbol{z}} p\left(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}\right) \log p(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\theta})$
- $= \sum_{\boldsymbol{z}} p\left(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}\right) \sum_{i=1}^{n} \log w_{z_i} \mathcal{N}\left(\boldsymbol{x}_i|\boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i}\right)$
- $= \sum_{i=1}^{n} \sum_{\boldsymbol{z}} p\left(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\theta}^{(t)}\right) \log w_{z_i} \mathcal{N}\left(\boldsymbol{x}_i|\boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i}\right)$
- $= \sum_{i=1}^{n} \sum_{c=1}^{k} r_{ic} \log w_{z_i} \mathcal{N}\left(\boldsymbol{x}_i|\boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i}\right)$
- $= \sum_{i=1}^{n} \sum_{c=1}^{k} r_{ic} \log w_{z_i}$
- $+ \sum_{i=1}^{n} \sum_{c=1}^{k} r_{ic} \log \mathcal{N}\left(\boldsymbol{x}_i|\boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i}\right)$

Maximisation step for GMM

- In the maximisation step, take partial derivatives of $Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}\right)$ with respect to each of the parameters and set the derivatives to zero to obtain new parameter estimates
- $w_c^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} r_{ic}$
- $\boldsymbol{\mu}_c^{(t+1)} = \frac{\sum_{i=1}^{n} r_{ic}\boldsymbol{x}_i}{r_c}$
  - Here $r_c \equiv \sum_{i=1}^{n} r_{ic}$
- $\boldsymbol{\Sigma}_c^{(t+1)} = \frac{\sum_{i=1}^{n} r_{ik}\boldsymbol{x}_i\boldsymbol{x}_i'}{r_k} - \boldsymbol{\mu}_c^{(t)}\left(\boldsymbol{\mu}_c^{(t)}\right)'$
- Note that these are the estimates for step $(t+1)$

Example of fitting Gaussian Mixture model



(a) Initial (b) E-step (c) M-step

(d) 2 cycles (e) 5-cyclces (f) 20-cycles

K-means as a EM for a restricted GMM

- Consider a GMM model in which all components have the same fixed probability $w_c = 1/k$, and each Gaussian has the same fixed covariance matrix $\boldsymbol{\Sigma}_c = \sigma^2 \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix
- In such a model, only component centroids $\boldsymbol{\mu}_c$ need to be estimated
- Next approximate a probabilistic cluster responsibility $r_{ic} = P\left(z_i = c | \boldsymbol{x}_i, \boldsymbol{\mu}_c^{(t)}\right)$ with a deterministic assignment $r_{ic} = 1$ if centroid $\boldsymbol{\mu}_c^{(t)}$ is closest to point $\boldsymbol{x}_i$, and $r_{ic} = 0$ otherwise
- Such a formulation results in a E-step where $\boldsymbol{\mu}_c$ should be set as a centroid of points assigned to cluster c
- In other words, k-means algorithm is a EM algorithm for the restricted GMM model described above*!!!*

This lecture

- Unsupervised learning
  - Diversity of problems
- Gaussian mixture model (GMM)
  - A probabilistic approach to clustering
  - The GMM model
  - GMM clustering as an optimisation problem
- The Expectation Maximization (EM) algorithm
- Next lecture: More unsupervised with dim reduction