

Lecture 23. Gaussian Mixture Models.

COMP90051 Statistical Machine Learning

Lecturer: Ben Rubinstein



THE UNIVERSITY OF
MELBOURNE

This lecture

- Unsupervised learning
 - * Diversity of problems
 - * k -means refresher
- Gaussian mixture model (GMM)
 - * A probabilistic approach to clustering
 - * The GMM model
 - * GMM clustering as an optimisation problem
- Starting Expectation-Maximisation (EM) algorithm

Unsupervised Learning

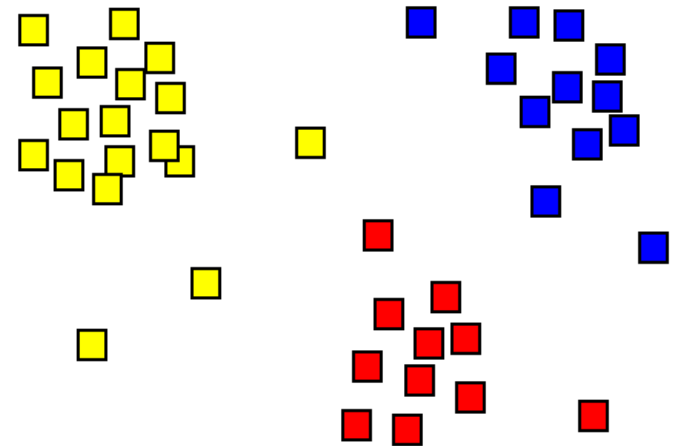
A large branch of ML that concerns
with learning the structure of the
data in the absence of labels

Main learning paradigms so far

- Supervised learning: Overarching aim is making predictions from data
- We studied methods in the context of this aim: e.g. linear/logistic regression, DNN, SVM
- We had instances $\mathbf{x}_i \in \mathbf{R}^m$, $i = 1, \dots, n$ and corresponding labels y_i for model fitting, aiming 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 where subroutine in contextual bandits requires supervised learning

Now: Unsupervised learning

- In unsupervised learning, there is no dedicated variable called a “label”
- Instead, we just have a set of points $\mathbf{x}_i \in \mathbf{R}^m, i = 1, \dots, n$
- Aim of unsupervised learning is to **explore the structure** (patterns, regularities) of data
- The aim of “exploring the structure” is vague



public domain

Unsupervised learning tasks

- Diversity of tasks fall into unsupervised learning category
 - * Clustering (now)
 - * Dimensionality reduction (autoencoders)
 - * 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

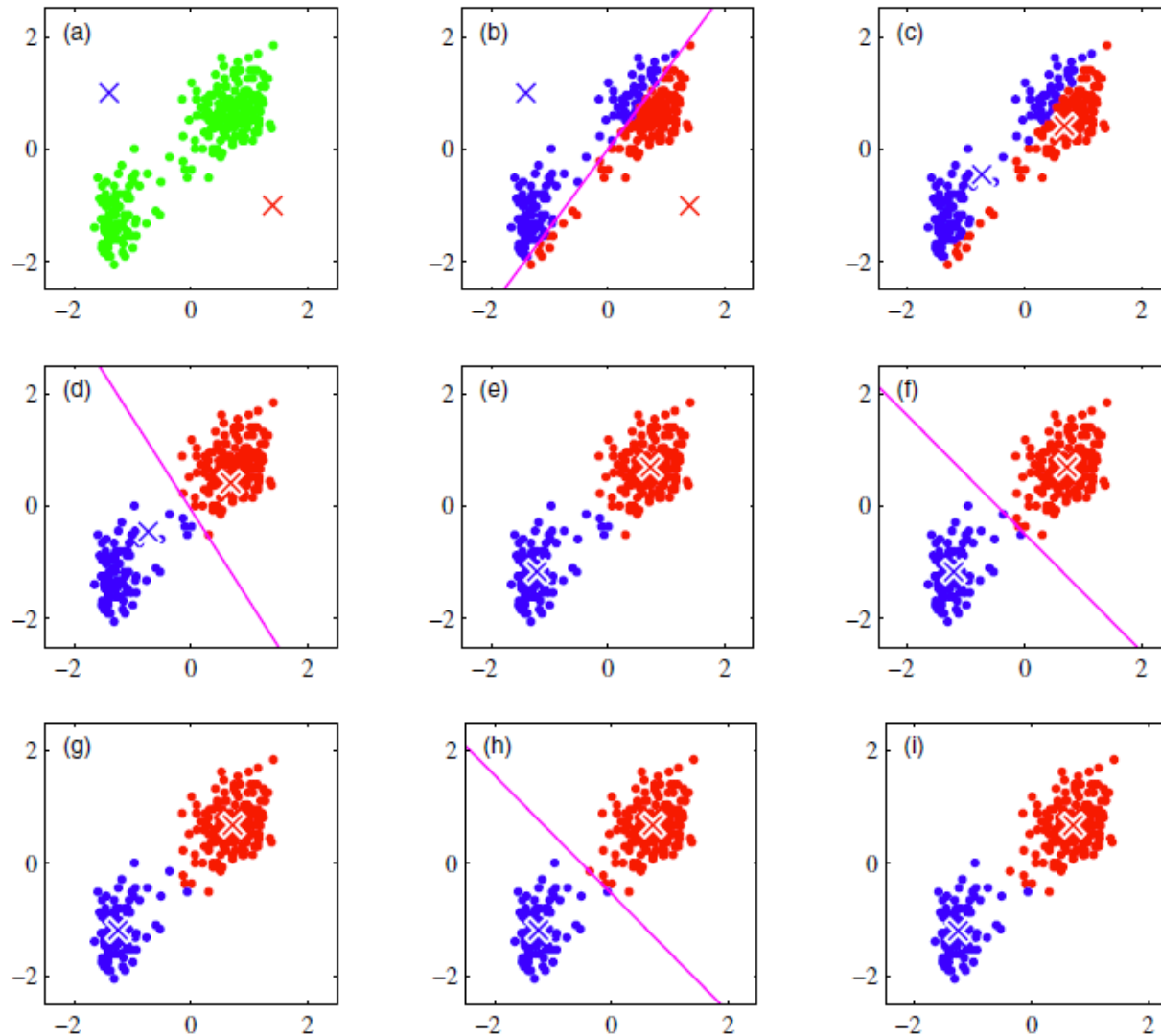
(Refresher?) *k*-means clustering

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

Mini Summary

- Unsupervised learning
 - * Face value: drop labels from training. That's it
 - * Actually: catch-all for many many ML tasks, even as steps in supervised learning pipelines
- Refresher: k -means
 - * Import next as we introduce GMMs

Next: The Gaussian mixture model

Gaussian Mixture Model

*A probabilistic view of clustering.
Simple example of a latent variable model.*

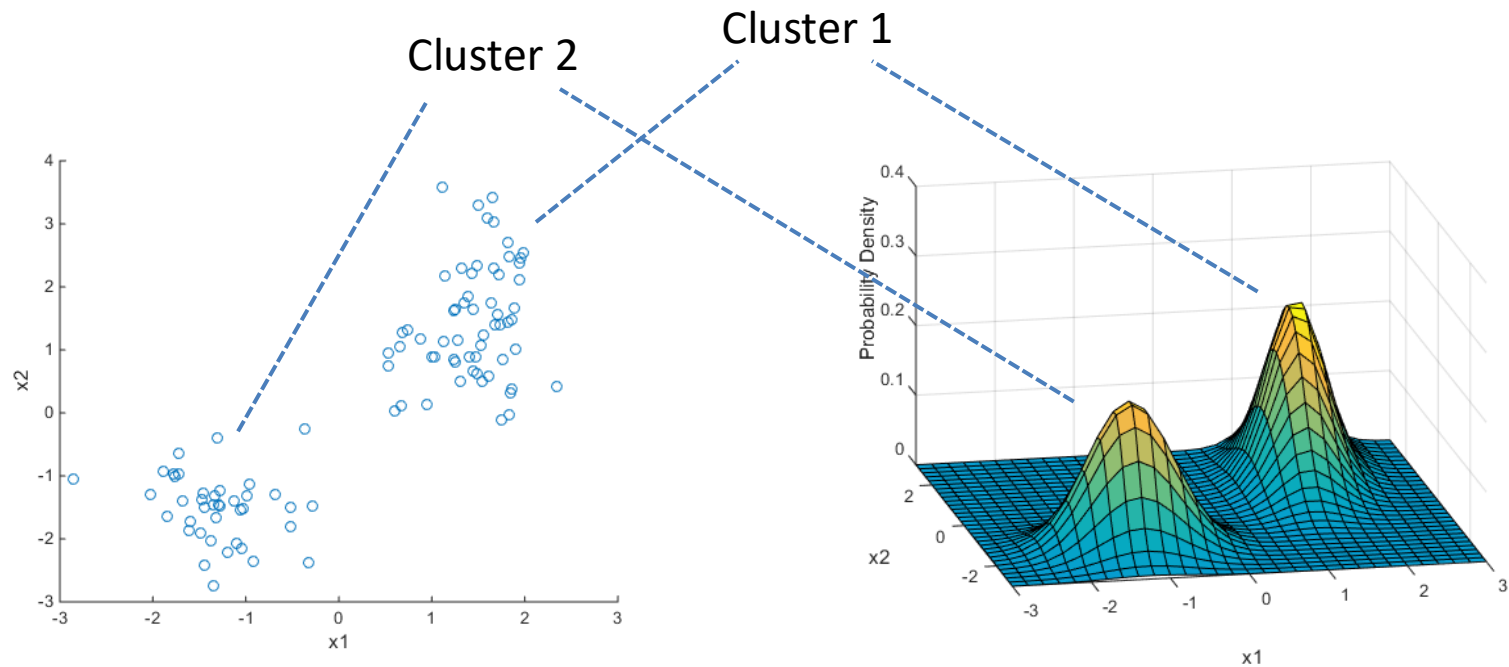
Modelling uncertainty in data clustering

- k -means clustering assigns each point to exactly one cluster
 - * Does this make sense for points that are between two clusters?
 - * Clustering is often not well defined to begin with!
- Like k -means, a probabilistic mixture model requires the user to choose the number of clusters in advance
- Unlike k -means, the probabilistic model grants us 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
 - * Clustering becomes model fitting in probabilistic sense. Philosophically satisfying.
 - * 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

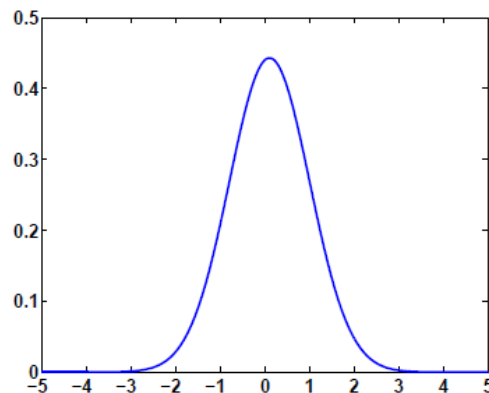
- Recall that a 1D Gaussian is

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

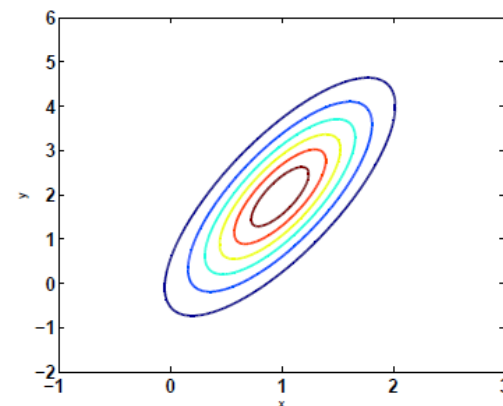
- And a d -dimensional Gaussian is

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

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



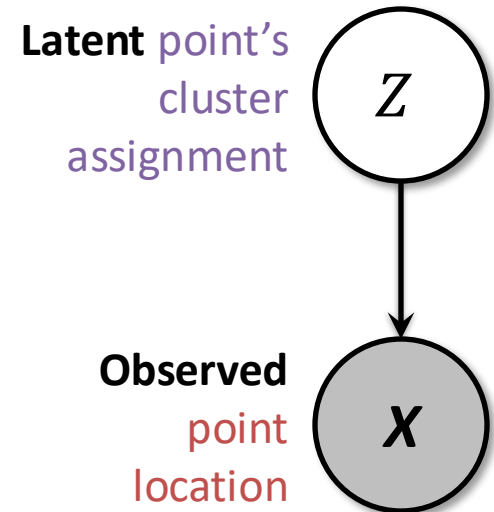
(a) 1-Dim



(b) 2-Dim

Gaussian mixture model (GMM): One point

- Cluster assignment of point
 - * Multinomial distribution on k outcomes
 - * $P(Z = j)$ described by $P(C_j) = w_j \geq 0$ with $\sum_{j=1}^k w_j = 1$
- Location of point
 - * Each cluster has its own Gaussian distribution
 - * Location of point governed by its cluster assignment
 - * $P(X|Z = j) = \mathcal{N}(\mu_j, \Sigma_j)$ class conditional density
- Model's **parameters**: $w_j, \mu_j, \Sigma_j, j = 1, \dots, k$



From marginalisation to mixture distribution

- When fitting the model to observations, we'll be maximising likelihood of observed portions of the data (the \mathbf{X} 's) not the latent parts (the Z 's)
- Marginalising out the Z 's derives the “familiar” mixture distribution

- Gaussian mixture distribution:

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

- A convex combination of Gaussians
- Simply marginalisation at work

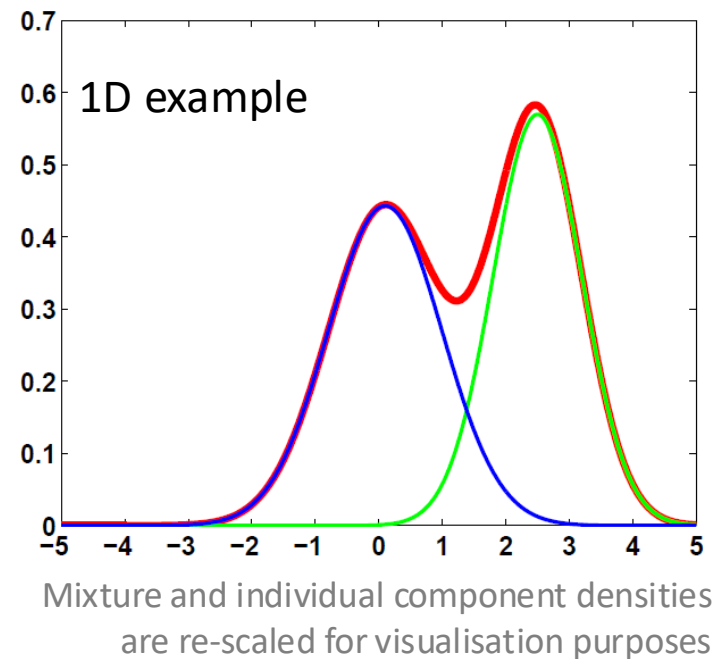
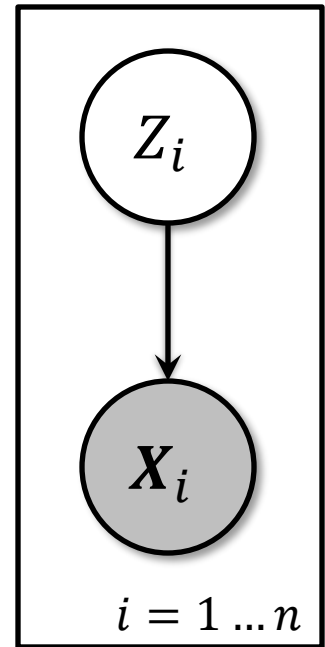


Figure: Bishop

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 our mixture distribution
 - * Shared parameters between points: independence assumption!!
- Clustering now amounts to finding parameters of the GMM that “best explains” observed data
- Call upon old friend **MLE** principle to find parameter values that maximise $p(\mathbf{x}_1, \dots, \mathbf{x}_n)$



Mini Summary

- GMM is just another D-PGM
- Some variables are observed some latent
- Convenient to model location as generated by cluster assignment
- Shared clusters arise from independence b/w points
- Mixture distribution arises algebraically from marginalisation

Next: MLE to fit the model, again motivating EM algorithm

Motivating (again) Expectation-Maximisation Algorithm

*We want to implement MLE but we have
unobserved r.v.'s that prevent clean
decomposition as happens in fully
observed settings*

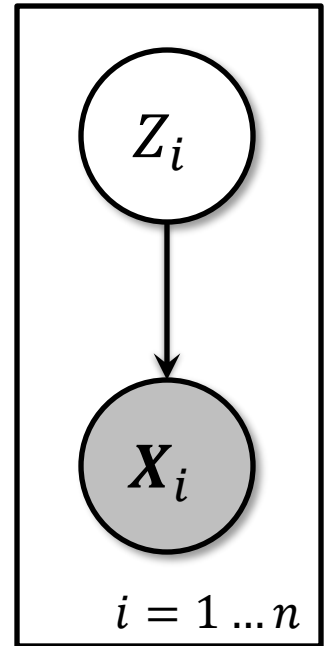
Fitting the GMM

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

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

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

Can be solved analytically?



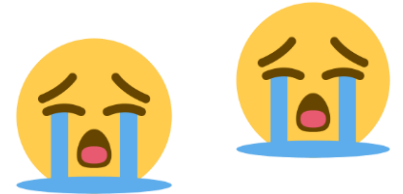
- Taking the derivative of this expression is pretty awkward, **try the usual log trick**

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

→ Expectation-Maximisation (EM)

Motivation of EM

- Consider a parametric probabilistic model $p(\mathbf{X}|\boldsymbol{\theta})$, where \mathbf{X} denotes data and $\boldsymbol{\theta}$ denotes a vector of parameters
 - According to MLE, we need to maximise $p(\mathbf{X}|\boldsymbol{\theta})$ as a function of $\boldsymbol{\theta}$
 - * equivalently maximise $\log p(\mathbf{X}|\boldsymbol{\theta})$
 - 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 Z 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



Expectation-Maximisation (EM) Algorithm

- Initialisation Step:

- * Initialize K clusters: C_1, \dots, C_K
 (μ_j, Σ_j) and $P(C_j)$ for each cluster j .

- Iteration Step:

- * Estimate the cluster of each datum

$$p(C_j | x_i)$$

 Expectation

- * Re-estimate the cluster parameters

 Maximisation

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

Summary

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

Next lecture: Getting to the bottom of EM