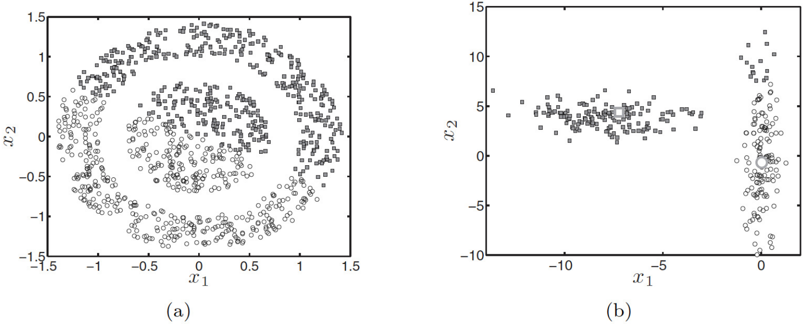
Clustering – Gaussian mixture model

Rohit Budhiraja

Machine Learning for Wireless Communications (EE798L)

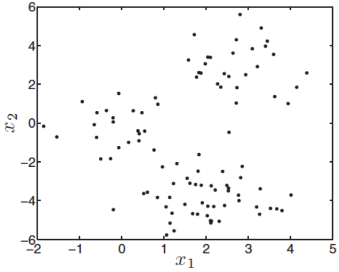
March 4, 2024

Recap and agenda of today's class



- Figure shows datasets for which the original K-means failed
- Problem with K-means algorithm is that its definition of a cluster was too crude
  - Characteristics of stretched clusters cannot be represented by a single point and squared distance
- Need to incorporate notion of shape
  - Statistical mixtures represent each cluster as a probability density
- Probabilistic mixture models helps in clustering a wide variety of shapes in almost any type of data
- Probabilistic mixture models helps in determine number of clusters

Mixture model – generative process



- Our earlier clustering dataset - how could we generate data that looks like this?
- Above data does not look like samples from any density function that we have encountered
  - There appear to be three disjoint regions in which data are concentrated
  - None of the density functions that we have seen can produce data with this complex structure
- However, each of the three regions looks simple enough to generate on its own
  - In fact, they all look a bit like samples from two-dimensional Gaussians

Mixture model – generative process

- Assuming that data was generated by three separate Gaussians we propose
- Two-step procedure for sampling the nth data object $\mathbf{x}_n$:
  1. Select one of the three Gaussians
  2. Sample $\mathbf{x}_n$ from this Gaussian
- Both steps are straightforward. Step 1 chooses one value from a discrete set, like rolling a die
  - To do this, we just need to define the probability of each outcome π_k such that $\sum_k \pi_k = 1$
- Having chosen which Gaussian to sample from, the second step is straightforward
- As in K-means, we will use z_{nk} as an indicator variable
  - If we choose kth component as the source of nth object, we set $z_{nk} = 1$, and $z_{nj} = 0$ for all $j \neq k$
- We will use $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ to denote the parameters of kth Gaussian
- Density function for $\mathbf{x}_n$ given that it was produced by the kth component $(z_{nk} = 1)$ is
  - Gaussian with mean and covariance $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, respectively

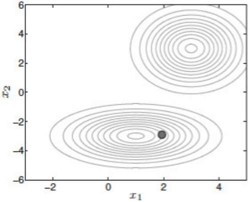
$$p\left(\mathbf{x}_n \mid z_{nk} = 1, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) = \mathcal{N}\left(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$$

Mixture model – generative process (2)

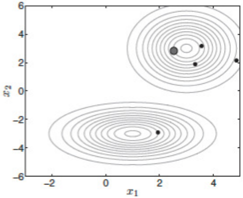
- To illustrate this process, we will sample some data from a setup with $K = 2$ Gaussians
- We will use the following means and covariances for the two components

$$\boldsymbol{\mu}_1 = [3, 3]^\mathsf{T}, \ \boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \qquad \boldsymbol{\mu}_2 = [1, -3]^\mathsf{T}, \ \boldsymbol{\Sigma}_2 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

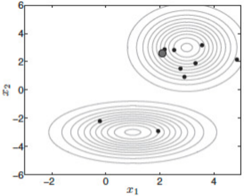
- Finally, we need to define π_k: assume component 1 is more likely, we will use $\pi_1 = 0.7$ and $\pi_2 = 0.3$



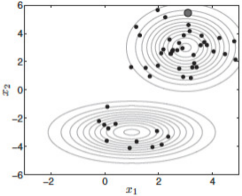
(a) The first object.



(b) The first five objects.



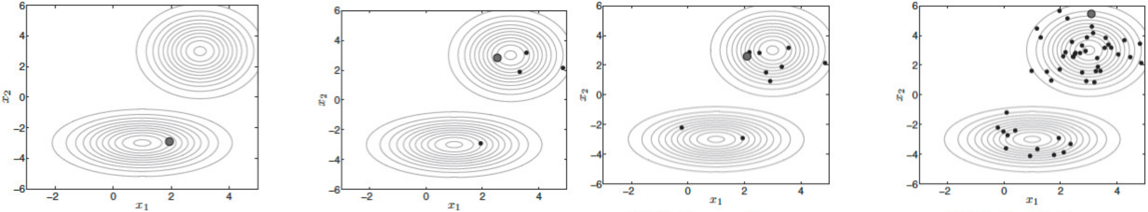
(c) The first t enobjects.



(d) The first 50 objects.

- Figure shows first 50 generated data objects and the density functions of the two Gaussians
- For first point in Figure (a), $k = 2$ is chosen and object sampled from second (lower) component
- Figure (b) shows the first five objects (the most recent is always denoted as a larger circle)
  - All but the first one have come from the first component because $\pi_1 > \pi_2$

Mixture model – generative process (3)



(a) The first object. (b) The first five objects. (c) The first t enobjects. (d) The first 50 objects.

- Data in Fig. (d) looks similar to our earlier figure
  - Generative procedure generates data by sampling it from a mixture of individual density functions
- Mixture models are widely used in data modelling
  - Fitting a set of simple distributions is often more straightforward than fitting one more complex one
- Learning task: Infer, from observed data
  - Component parameters $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ and assignments of objects to components

Mixture model – learning objective

- As with K- means, this is a circular argument:
  - component parameters would be easy to compute if we knew the assignments, and
  - assignments would be easy to compute if we knew the component parameters
- Without either, it is hard to know where to start
- Answer comes in form of Expectation-Maximisation (EM) algorithm - parallel to K-means algorithm
- EM algorithm iteratively maximizes the likelihood, and used for a wide range of models
- Develop EM algorithm as general as possible: we will work with $p(\mathbf{x}_n \mid z_{nk} = 1, \Delta_k)$
  - Δ_k denotes parameters of the kth density (not necessarily Gaussian)
  - Δ will denote the collection of parameters of all mixture components $\Delta = \{\Delta_1, \cdots, \Delta_K\}$
  - Use $\boldsymbol{\pi} = \{\pi_1, \cdots, \pi_K\}$.
- Require likelihood of data objects $\mathbf{x}_n$ under the whole model:$p(\mathbf{x}_n \mid \Delta, \boldsymbol{\pi})$
- We start with likelihood of a particular data object conditioned on $z_{nk} = 1$:

$$p(\mathbf{x}_n \mid z_{nk} = 1, \Delta) = p(\mathbf{x}_n \mid \Delta_k)$$

Mixture model likelihood

- To obtain $p(\mathbf{x}_n \mid \Delta, \boldsymbol{\pi})$, we need to get rid of z_{nk}
- To do this, we first multiply both sides by $p(z_{nk})$, which we have defined as π_k:

$$\begin{aligned} p(\mathbf{x}_n \mid z_{nk} = 1, \Delta)\, p(z_{nk} = 1) &= p(\mathbf{x}_n \mid \Delta_k)\, p(z_{nk} = 1) \\ p(\mathbf{x}_n, z_{nk} = 1 \mid \Delta, \boldsymbol{\pi}) &= p(\mathbf{x}_n \mid \Delta_k)\, \pi_k \end{aligned}$$

- Summing both sides over k (marginalising over the individual components) yields

$$\begin{aligned} \sum_{k=1}^{K} p(\mathbf{x}_n, z_{nk} = 1 \mid \Delta, \boldsymbol{\pi}) &= \sum_{k=1}^{K} p(\mathbf{x}_n \mid \Delta_k)\, \pi_k \\ p(\mathbf{x}_n \mid \Delta, \boldsymbol{\pi}) &= \sum_{k=1}^{K} \pi_k\, p(\mathbf{x}_n \mid \Delta_k) \end{aligned}$$

- Making standard independence assumption, we can extend this to likelihood of all N data objects:

$$p(\mathbf{X} \mid \Delta, \boldsymbol{\pi}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k\, p(\mathbf{x}_n \mid \Delta_k)$$

- Maximise log likelihood to calculate optimal parameter $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\pi}$

$$L = \log p(\mathbf{X} \mid \Delta, \boldsymbol{\pi}) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k\, p(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)) \tag{1}$$

Fundamental theorem of expectation

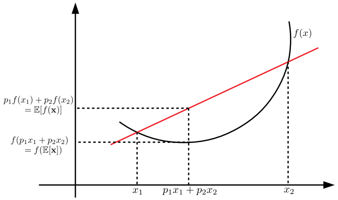
- Summation inside log term complicates maximization – helps comes in form of Jensen inequality
- X is a random variable then $\mathbb{E}X$ is its expectation:

$$\mathbb{E}X = \sum_{x \in \mathcal{X}} xp(x)$$

- Let $X = x_1$ with prob. p_1 and $X = x_2$ with prob. p_2, then $\mathbb{E}X = p_1x_1 + p_2x_2$ with $p_1 + p_2 = 1$
- Recall fundamental theorem of expectation

$$\mathbb{E}f(X) = \sum_{x \in \mathcal{X}} f(x)p(x) = p_1 f(x_1) + p_2 f(x_2)$$

- Let's understand a convex function



$$p_1 f(x_1) + p_2 f(x_2) \quad \geq \quad f(p_1 x_1 + p_2 x_2)$$

Jensen inequality

- Jensen inequality: If f is a convex function and X is a random variable then

$$\mathbb{E}f(X) \geq f(\mathbb{E}X)$$

- Proof: Let $X = x_1$ with probability p_1 and $X = x_2$ with probability p_2, then for convex f

$$\begin{aligned} p_1 f(x_1) + p_2 f(x_2) &\geq f(p_1 x_1 + p_2 x_2) && \text{definition of convexity} \\ \mathbb{E}f(X) &\overset{(a)}{\geq} f(\mathbb{E}X) \end{aligned}$$

- Can be proved using induction for larger number of mass points
- Inequality sign will change for concave functions

$$f(\mathbb{E}X) \geq \mathbb{E}f(X)$$

Simplification of log likelihood using Jensen inequality (1)

- We shall now demonstrate the use of the EM algorithm to maximise the log likelihood

$$L = \log p(\mathbf{X} \mid \Delta, \boldsymbol{\pi}) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k p\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)) \tag{2}$$

- Summation inside logarithm makes finding optimal parameter $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\pi}$ difficult
- EM algorithm overcomes this problem by deriving a lower bound on this likelihood
  - Instead of maximising L directly, we instead maximise the lower bound
- To obtain a lower bound on L we use Jensen's inequality

$$\log \mathbb{E}_{p(z)}\{f(z)\} \geq \mathbb{E}_{p(z)}\{\log f(z)\}$$

- To use Jensen's inequality, we need to make RHS of (2) look like the log of an expectation.
- To do this, we multiply and divide expression inside summation over k by a new variable q_{nk}

$$L = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k p\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \frac{q_{nk}}{q_{nk}}$$

- If we restrict q_{nk} to be positive and satisfy the summation constraint $\sum_{k=1}^{K} q_{nk} = 1$

Simplification of log likelihood using Jensen inequality (2)

- q_{nk} is some probability distribution over the K components for the nth object

$$L = \sum_{n=1}^{N} \log \sum_{k=1}^{K} q_{nk} \frac{\pi_k p\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{q_{nk}} = \sum_{n=1}^{N} \log \mathbb{E}_{q_{nk}} \left\{ \frac{\pi_k p\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{q_{nk}} \right\}$$

- Applying Jensen's inequality, we can lower bound this expression:

$$L = \sum_{n=1}^{N} \log \mathbb{E}_{q_{nk}} \left\{ \frac{\pi_k p\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{q_{nk}} \right\} \geq \sum_{n=1}^{N} \mathbb{E}_{q_{nk}} \left\{ \log \frac{\pi_k p\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{q_{nk}} \right\}$$

- RHS of this expression is the bound (we will denote it $\mathcal{B}$) that we shall optimise
- Expanding the expression gives us something more manageable:

$$\begin{aligned} \mathcal{B} &= \sum_{n=1}^{N} \mathbb{E}_{q_{nk}} \left\{ \log \frac{\pi_k p\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{q_{nk}} \right\} = \sum_{n=1}^{N} \sum_{k=1}^{K} q_{nk} \log \left(\frac{\pi_k p\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{q_{nk}} \right) \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K} q_{nk} \log \pi_k + \sum_{n=1}^{N} \sum_{k=1}^{K} q_{nk} \log p\left(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) - \sum_{n=1}^{N} \sum_{k=1}^{K} q_{nk} \log q_{nk} \end{aligned} \quad (3)$$

- Calculate $q_{nk}, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\pi}$ to find a local maxima of this bound
  - Values will correspond to a local maxima of log-likelihood L