K Means Clustering, GMM's and EM Algorithm

Anirudha Brahma & Arpit Maheshwari

February 21, 2022

1 Unsupervised Learning Algorithms

Suppose we have N data-points in the 2d space represented by $\mathbf{x} = (x_1, x_2)$ without any labels attached to them. Our goal is to assign these data-points to K classes simply based on the position of the points in the space. The algorithms by which we classify these points into the K classes are known as Unsupervised learning algorithms.

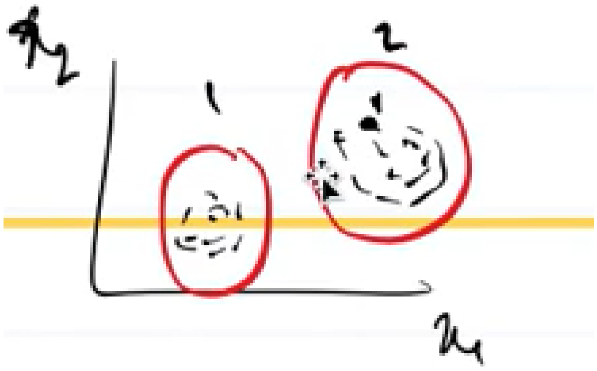

Figure 1: Grouping of data-points

2 K-means Clustering

2.1 Introduction

K means clustering is an algorithm to do unsupervised learning. The intuition for this algorithm is that the data-points which are very close to each other should ideally be grouped into the same class. The algorithm tries to learn an optimal division of space to group the data-points into K different classes.

2.2 Some Terminologies

- K means algorithm defines the mean of a cluster as just another point in the space. This point is equal to the mean of the data-points assigned to this cluster.
- Mean of the k^{th} cluster is μ_k
- r_{nk} is called the membership function, and is equal to 1 if the n^{th} data-point belongs to the k^{th} cluster, and 0 otherwise. It is called hard clustering as one data-point is assigned just one cluster.

2.3 Learning Algorithm

According to the K means clustering algorithm, a data point x_i is assigned to the cluster k' which gives the minimum value of $(x_i - \mu_{k'})^T(x_i - \mu_{k'})$.

Which boils down to finding the minimum of :

$$\text{Distortion Measure: } L = \sum_{n,k} r_{nk}(x_n - \mu_k)^T(x_n - \mu_k) \tag{1}$$

$$= \sum_{n,k} r_{nk}||x_n - \mu_k||^2 \tag{2}$$

This quantity is the sum of all the distances of the data-points from their assigned cluster. For minimising distortion measure, we have to find optimum cluster means μ_k as well as the optimum membership functions r_{nk}. To achieve the optimum parameters we follow the following iterative algorithm:

First of all we choose a k, which is a hyperparameter. Then initialize k random cluster means. After that we loop over the following 2 steps.

- **Step 1: Assigning the data-points to the clusters** For each data-point x_n, find k that minimises $(x_n$-$\mu_k)^T(x_n$-$\mu_k)$(i.e., find the closest cluster mean) and set $r_{nk} = 1$, and $r_{nj} = 0$ for all j $\neq$ k. Mathematically,

$$r_{nk} = \begin{cases} 1, & \text{if k} = \underset{k'}{\text{argmin}}\ ||x_n - \mu_{k'}|| \\ 0, & \text{otherwise} \end{cases}$$

- **Step 2: Updating the clusters means** We update the cluster means using data-points and membership function:

$$\frac{\partial L}{\partial \mu_{k'}} = 2\sum_n r_{nk'}(x_n - \mu_{k'}) \tag{3}$$

$$0 = \sum_n r_{nk'}x_n - \mu_{k'}\sum_n r_{nk'} \tag{4}$$

$$\mu_{k'} = \frac{\sum_n r_{nk'}x_n}{\sum_n r_{nk'}} \tag{5}$$

The algorithm iterates over the above mentioned 2 steps until we reach the convergence criteria.

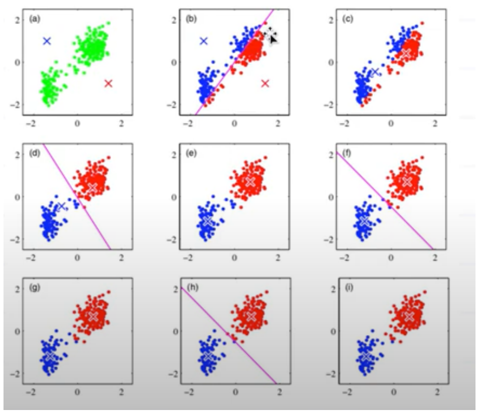

Figure 2: Algorithm in action

3 Gaussian Mixture Models

3.1 Why GMMs and intuition behind them

K-Means clustering suffers from several drawbacks due to its simplicity. It has a non-probabilistic nature which essentially does hard clustering using a simple metric of distance of the point from cluster center. This hard clustering leads to poor performance. In order to address these problems we try to do cluster assignment probabilistically by measuring the uncertainties in assigning clusters by comparing distance of the point to every cluster center. We also would like that cluster boundaries be allowed to be ellipses rather than circles. These modifications eventually lead to **Gaussian mixture models**.

3.2 Latent Variable Models

Latent variables help complicated distributions to be formed from simpler components.

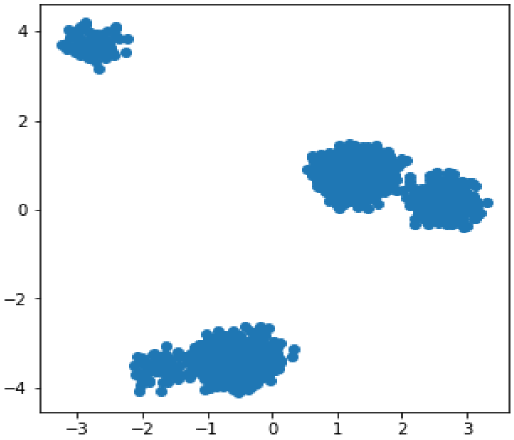

Figure 3: Gaussian mixture consisting of 5 Gaussians

This data does not look like to be sampled from a standard probability distribution. Rather samples in each cluster come from a 2-D Gaussian. Such complicated distributions can be modelled through latent variable technique with the probability density function being:

$$p(\boldsymbol{x}) = p(\boldsymbol{x}|z=1)p(z=1) + ... + p(\boldsymbol{x}|z=k)p(z=k) \tag{6}$$

$$p(\boldsymbol{x}) = \sum\nolimits_k p(\boldsymbol{x}|z=k)p(z=k) \tag{7}$$

Sample x comes from the distribution specified by the latent variable z, $p(z=k)$ is the probability that the k$^{\text{th}}$ distribution samples the data point in consideration. Also,

$$\sum\nolimits_k p(z=k) = 1 \tag{8}$$

3.3 Considering the GMM case

A GMM is comprised of several Gaussians with each k-th Gaussian having certain parameters namely:

- Mixing probability
- Its center
- The covariance matrix

The overall probability density function is a weighted sum of the individual Gaussians which when generalised over N samples gives:

$$
\begin{aligned}
p(\boldsymbol{x}) &= \sum\nolimits_k p(z_k)p(\boldsymbol{x}|z_k) \qquad (9)\\
&= \sum\nolimits_k \pi_k \, \mathcal{N}(\boldsymbol{x}|\mu_k, \Sigma_k) \qquad (10)
\end{aligned}
$$

The learnable parameters of the model are $\boldsymbol{\theta} = \{\pi_k, \mu_k, \Sigma_k; k = 1, 2, ..., K\}$ whereas K (Number of clusters) is the hyperparameter for the model.

3.4 Parameter Estimation

We will use MLE estimate to get the learnable parameters. So we first find the log-likelihood: The likelihood function is:

$$
L = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \, \mathcal{N}(x_n|\mu_k, \Sigma_k) \tag{11}
$$

The log-likelihood is given by:

$$
\begin{aligned}
\log(L) &= \sum_{n=1}^{N} \log\left(\sum_{k=1}^{K} \pi_k \, \mathcal{N}(x_n|\mu_k, \Sigma_k)\right) \qquad (12)\\
&= \sum_{n=1}^{N} \log\left(\sum_{k=1}^{K} \frac{\pi_k}{(\sqrt{2\pi})^D(|\Sigma_k|)^{1/2}} \exp\left\{-\frac{1}{2}(x_n-\mu_k)^T\Sigma_k^{-1}(x_n-\mu_k)\right\}\right) \qquad (13)
\end{aligned}
$$

Now, we differentiate the expression of Log-Likelihood with respect to the particular parameter and equate it to 0.

$$
\begin{aligned}
\frac{\partial \log{(L)}}{\partial \mu_{k'}} &= \sum_n \frac{\pi_{k'} \, \mathcal{N}(x_n|\mu_{k'}, \Sigma_{k'})}{\sum_k \pi_k \, \mathcal{N}(x_n|\mu_k, \Sigma_k)} \left(-\Sigma_{k'}^{-1}(x_n-\mu_{k'})\right) = 0 \qquad (14)\\
&\implies \sum_n \gamma_{nk'} \left(-\Sigma_{k'}^{-1}(x_n-\mu_{k'})\right) = 0 \qquad (15)\\
&\implies \sum_n \gamma_{nk'}\mu_{k'} = \sum_n \gamma_{nk'} x_n \qquad (16)\\
&\implies \boxed{\mu_{k',\text{MLE}} = \frac{\sum_n \gamma_{nk'} x_n}{N_{k'}}} \qquad (17)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \log{(L)}}{\partial \Sigma_{k'}} = \sum_n \gamma_{nk'} &\left(\frac{-1}{2|\Sigma_{k'}|^{3/2}} \frac{\partial |\Sigma_{k'}|}{\partial \Sigma_{k'}}\right)\\
&+ \gamma_{nk'} \left(\frac{1}{|\Sigma_k|^{1/2}} \frac{\partial}{\partial \Sigma_{k'}} \left\{-\frac{1}{2}(x_n-\mu_{k'})^T\Sigma_{k'}^{-1}(x_n-\mu_{k'})\right\}\right) = 0 \qquad (18)
\end{aligned}
$$

$$
\implies \sum_n \gamma_{nk'} \left((\Sigma_{k'}^T)^{-1} - (\Sigma_{k'}^T)^{-1}(x_n-\mu_{k'})(x_n-\mu_{k'})^T(\Sigma_{k'}^T)^{-1}\right) = 0 \tag{19}
$$

$$
\implies \boxed{\Sigma_{k',\text{MLE}} = \frac{\sum_n \gamma_{nk'}(x_n-\mu_{k'})(x_n-\mu_{k'})^T}{N_{k'}}} \tag{20}
$$

Where we have used the identities:

$$
\frac{\partial(|\mathbf{A}|)}{\partial \mathbf{A}} = |\mathbf{A}|(\mathbf{A}^T)^{-1} \tag{21}
$$

$$\frac{\partial \mathbf{a}^T \mathbf{A}^{-1} \mathbf{b}}{\partial \mathbf{A}} = -(\mathbf{A}^T)^{-1} \mathbf{a} \mathbf{b}^T (\mathbf{A}^T)^{-1} \tag{22}$$

For estimating the parameters $\pi_{k'}$, we use Lagrange Multipliers as this is a case of constrained optimization with the constraint being $\sum_k \pi_{k'} = 1$. So we differentiate $\log(L) + \lambda(\sum_k \pi_{k'} - 1)$ with respect to $\pi_{k'}$ to get:

$$\frac{\partial}{\partial \pi_{k'}} \{\log(L) + \lambda(\textstyle\sum_k \pi_{k'} - 1)\} = \sum_n \frac{\mathcal{N}(x_n|\mu_{k'}, \Sigma_{k'})}{\sum_k \pi_k \, \mathcal{N}(x_n|\mu_k, \Sigma_k)} + \lambda = 0 \tag{23}$$

Multiplying the equation obtained with $\pi_{k'}$ and summing over k' gives:

$$\lambda = -N \tag{24}$$

Hence,

$$\sum_n \frac{\gamma_{nk'}}{\pi_{k'}} + (-N) = 0 \tag{25}$$

$$\implies \boxed{\pi_{k',\text{MLE}} = \frac{N_{k'}}{N}} \tag{26}$$

3.5 How is GMM applied?

The equations for different parameters are dependent on each other, so an iterative method is used until its convergence:

- **1:** Initialize all the parameters.
- **2:** Find γ_{nk} using the parameters .
- **3:** Update the parameters $\boldsymbol{\theta}$ using the update equations.
- **4:** Stop if converged, else go back to 2.

A quick glance at the update equations:

$$\gamma_{nk'} = \frac{\pi_{k'} \, \mathcal{N}(x_n|\mu_{k'}, \Sigma_{k'})}{\sum_k \pi_k \, \mathcal{N}(x_n|\mu_k, \Sigma_k)} \tag{27}$$

$$N_{k'} = \sum_n \gamma_{nk'} \tag{28}$$

$$\mu_k = \frac{\sum_n \gamma_{nk'} x_n}{N_{k'}} \tag{29}$$

$$\Sigma_{k'} = \frac{\sum_n \gamma_{nk'} (x_n - \mu_{k'})(x_n - \mu_{k'})^T}{N_{k'}} \tag{30}$$

$$\pi_{k'} = \frac{N_{k'}}{N} \tag{31}$$

4 Expectation-Maximization(EM) Algorithm

In expressions of Maximum Log-Likelihood Estimation for the parameters of the model, we have to find the value of summation inside the logarithm, which makes finding the optimal values of the parameters difficult.

$$\text{Log-likelihood} = \log(p(\boldsymbol{S}|\boldsymbol{\theta})) = \log\left(\sum_z p(\boldsymbol{S}, z|\boldsymbol{\theta})\right) \tag{32}$$

$$= \log\left(\sum_z p(z|\boldsymbol{\theta}) p(\boldsymbol{S}|z, \boldsymbol{\theta})\right) \tag{33}$$

where $\boldsymbol{S}$ is observed data, z is a latent variable and $\boldsymbol{\theta}$ are learnable parameters.

EM algorithm is used to solve this problem by finding a lower bound on this log-likelihood. Here, instead of maximizing the log-likelihood, we now maximize the lower bound to this log-likelihood. We use the **Jensen's Inequality** to calculate the lower bound.

4.1 Jensen's Inequality

If f is a concave function and X is any random variable, then :

$$E(f(X)) \leq f(E(X)) \tag{34}$$

Our Log-Likelihood function is of the form $\log(E(X))$ and we find the lower bound to this expression using the Jensen's inequality which is $E(\log(X))$. We call this lower bound to be Q :

$$Q = \sum_{z} p(z|\boldsymbol{S}, \boldsymbol{\theta}^{old}) \log(p(\boldsymbol{S}, z|\boldsymbol{\theta})) \tag{35}$$

$$Q = E_{z|\boldsymbol{S},\boldsymbol{\theta}^{old}} \left(\log(p(\boldsymbol{S}, z|\boldsymbol{\theta}))\right) \tag{36}$$

For making the calculation of $\frac{\partial Q}{\partial \theta}$ easier, we can assume that $p(z|S, \theta)$ is calculated at $\theta = \theta^{old}$ since we follow an iterative algorithm, and therefore acts as a constant when we will calculate the $\frac{\partial Q}{\partial \theta}$.

The Gaussian Mixture Model log likelihood also has a summation inside the logarithm term, so we can use the EM algorithm to find estimation of the parameters. The bound Q for the case of GMM(**E-step**) can be calculated as shown below:

$$p(\boldsymbol{S}, z|\boldsymbol{\theta}) = \prod_{n}\prod_{k} \pi_k \, \mathcal{N}(s_n|\mu_k, \Sigma_k)^{z_{nk}} \tag{37}$$

$$\log(p(\boldsymbol{S}, z|\boldsymbol{\theta})) = \sum_{n}\sum_{k} z_{nk} \log\left(\pi_k \, \mathcal{N}(s_n|\mu_k, \Sigma_k)\right) \tag{38}$$

$$Q = \sum_{z} p(z|\boldsymbol{S}, \boldsymbol{\theta}^{old}) \log(p(\boldsymbol{S}, z|\boldsymbol{\theta})) \tag{39}$$

$$= \sum_{n}\sum_{k} \gamma_{nk} \log\left(\pi_k \, \mathcal{N}(s_n|\mu_k, \Sigma_k)\right) \tag{40}$$

4.2 The EM Procedure

The iterative algorithm to do the E and M steps are :

- **Step 1:** Initialize all the trainable parameters ($\boldsymbol{\theta}$).
- **Step 2:** Evaluate $p(z|\boldsymbol{S}, \boldsymbol{\theta})$, i.e., γ_{nk}.

  $$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{old}) = E_{z|\boldsymbol{S},\boldsymbol{\theta}^{old}} \left(\log(p(\boldsymbol{S}, z|\boldsymbol{\theta}))\right) \tag{41}$$

  where the expectation is computed with respect to the conditional distribution of z given $\boldsymbol{S}$ for the current iterate $\boldsymbol{\theta}^{old}$. This step requires calculating expectation , hence the name **E** step.

- **Step 3:** Update the parameters ($\boldsymbol{\theta}$: μ_k, Σ_k, π_k) by Maximizing Q.

  $$\boldsymbol{\theta}^{new} = \underset{\boldsymbol{\theta}}{\text{argmax}} \; Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{old}) \tag{42}$$

  This step involves maximisation of Q, hence is also known as the **M** step.

- **Step 4:** Stop if the convergence criteria is satisfied, otherwise go back to step 2.