S21CSE471: Statistical Methods in AI **Spring 2021**

Gaussian Mixture Models

Mohee Datta Gupta:2018112005 CH N V B Dattatreya:2020201011 Pawan Patidar:2020201031

1 Recap

Definition 1.1. ***Discriminative classifiers*** *tries to model the decision boundary between classes*
Ex. SVM

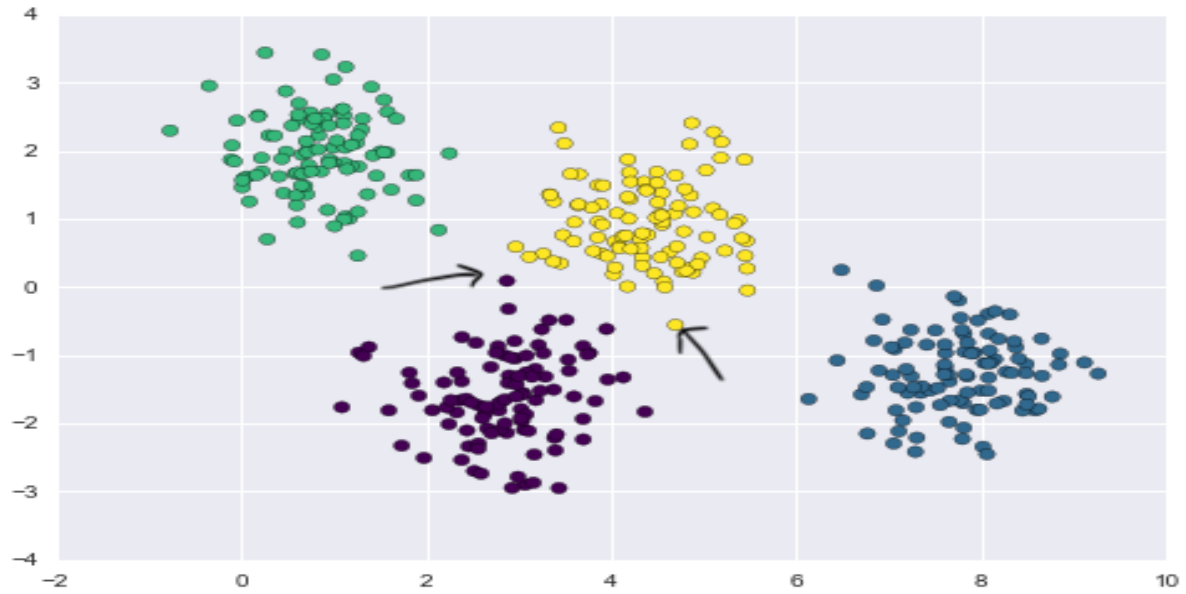
Definition 1.2. ***Generative classifiers*** *tries to model the actual distribution of each class*
Ex. GMM

Theorem 1.3. *To find the probability that a value x_i belongs to a distribution y with mean μ_y and variance σ_y^2 we use*

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

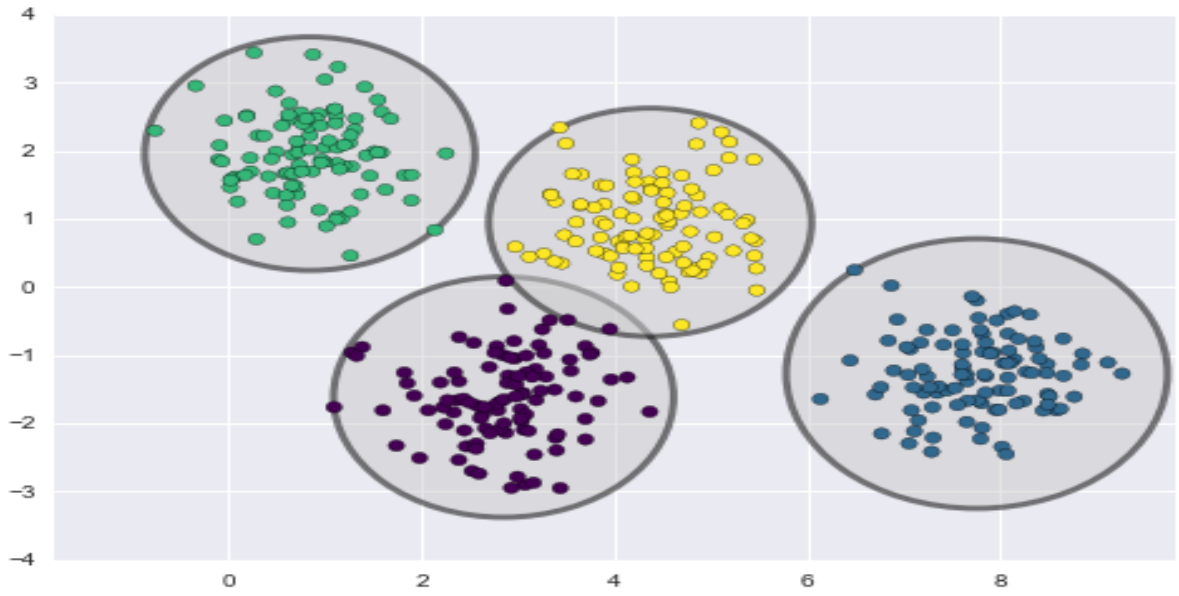
2 Motivation for GMM

Lets look at some of the weaknesses of the k-means and how we might improve the cluster model. For a well separated data K-means achieves satisfactory results.

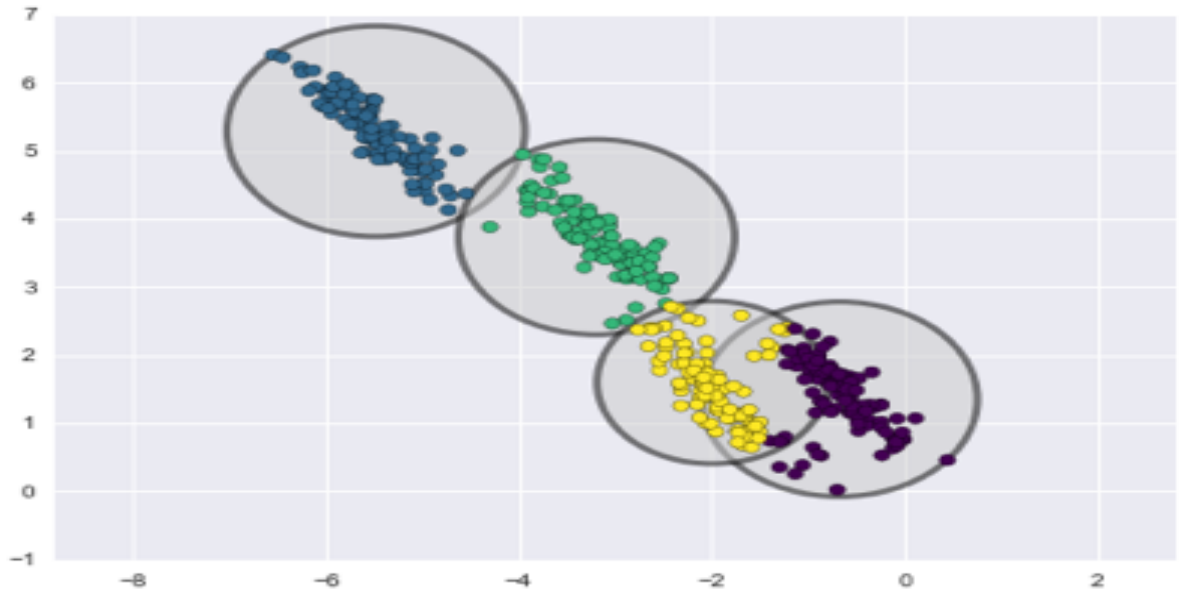


From a visual perspective, we might expect the clustering assignment for some points is more certain than others. In the above figure two points(shown by arrow) can be treated as uncertain points.

What k-means actually does is it places a hyper-sphere of a radius defined by the farthest point in the cluster. This radius acts as a threshold for cluster assignment. Any point outside this radius is considered as not part of the cluster.



But this works well for only nicely separable data. Consider situation where data points are close to each other.



Clearly K means does not do very well in assigning a cluster to the data for the yellow and purple colored points. This is because K-means only considers the mean of a cluster and neglects the variance with in a cluster.

This is where Gaussian Mixture Models come in and try to fit the data in a generative way rather than discriminative way. It tries to describe data by fitting the data as weighted guassians which will be described below.

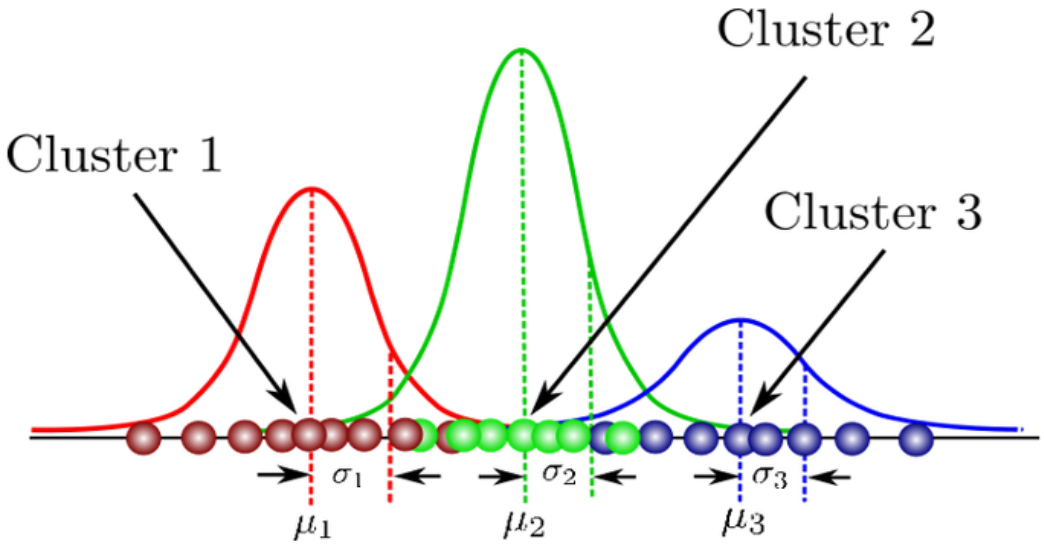
3 Gaussian Mixture Models

A Gaussian Mixture is a function that is comprised of several Gaussians, each identified by k $\in$ {1,...,K} , where K is the number of clusters of our dataset. Each Gaussian k in the mixture is comprised of the

following parameters:

- A mean μ that defines its centre.
- A covariance Σ that defines its width. This would be equivalent to the dimensions of an ellipsoid in a multivariate scenario.
- A mixing probability π that defines how big or small the Gaussian function will be.

Let us now illustrate these parameters graphically:



Here, we can see that there are three Gaussian functions, hence K = 3. Each Gaussian explains the data contained in each of the three clusters available. The mixing coefficients are themselves probabilities and must meet this condition:

$$\sum_{k=1}^{K} \pi_k = 1$$

Now how do we determine the optimal values for these parameters? To achieve this we must ensure that each Gaussian fits the data points belonging to each cluster. This is exactly what maximum likelihood does.

In general, the Gaussian density function is given by:

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

Where x represents our data points, D is the number of dimensions of each data point. μ and σ are the mean and covariance, respectively. If we have a dataset comprised of N = 1000 three-dimensional points (D = 3), then x will be a 1000 × 3 matrix. will be a 1 × 3 vector, and will be a 3 × 3 matrix. For later purposes, we will also find it useful to take the log of this equation, which is given by:

$$\ln \mathcal{N}(\mathbf{x}|\mu,\Sigma) = -\frac{D}{2}\ln 2\pi - \frac{1}{2}\ln \Sigma - \frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)$$

4 Expectation Maximization Algorithm

Expectation maximization is an iterative algorithm for using maximum likelihood to estimate the parameters of a statistical model with unobserved (hidden)variables. It has two main steps. First is the E-step.We compute some probability distribution of the model so we can use it for expectations.Second comes the M-step, which stands for maximization. In this step, we maximize the lower bound of the log-likelihood function by generating a new set of parameters with respect to the expectations.

First Find the Conditions for which maximum likelihood can be reached. So for that recall the older maximum likelihood function which is in form of sum of logs and finding optimum parameters from that function is hard. So try to reduce derivative of that function in terms of our new introduced variables

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

1. First taking Derivative w.r.t μ_k.

$$0 = -\sum_{n=1}^{N}\frac{\pi_k\mathcal{N}(x_n|\mu_k,\sum_k)}{\sum_j \pi_j\mathcal{N}(x_n|\mu_j,\sum_j)}\sum_k^{(} x_n - \mu_k)$$

   As we defined $\gamma(z_{nk})$ before, it can be used to reduce the term as shown and after simplifying it, μ_k can be estimated.

$$\mu_k = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})x_n$$

   where,

$$N_k = \sum_{n=1}^{N}\gamma(z_{nk})$$

2. Taking Derivative w.r.t π_k we will get,

$$\sum_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T$$

3. Taking Derivative w.r.t pi_k Here we must take account of the constraint $\sum \pi_k = 1$ can be achieved using a Lagrange multiplier

   Now maximize following quantity,

$$lnp(X|\pi, \mu, \sum) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

$$0 = \sum_{n=1}^{N} \frac{(x_n|\mu, \sum_k)}{\sum_j \pi_j(N)(x_n|\mu_j, \sum_j)} + \lambda$$

   which gives,

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

Steps for EM Algorithm

Given a Guassian model, maximize the likelihood function w.r.t parameters

1. Initialize μ_k, σ_k and π_k find the initial value of log liklihood.
2. **E step:** Evaluate the $\gamma(Z_{nk}$ using the current parameter values.

$$\gamma(z_{nk}) = \frac{\pi_k(N)(x_n|\mu_k, \sum_k}{\sum_{j=1}^{K} \pi_j(N)(x_n|\mu_j, \sum_j)}$$

3. **M step:** Re-estimate the parameters from $\gamma(Z_{nk})$

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1} N\gamma(z_{nk})x_n$$

$$\sum_k^{new} = \frac{1}{N_k} \sum_{n=})^{N} (x_n - \mu_(k)^{new}(x_n - \mu_k^{new})^T)$$

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

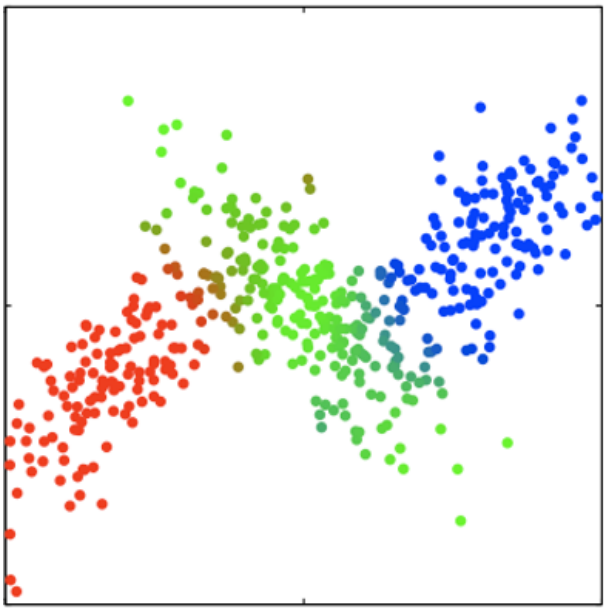
4. Evaluate the log likelihood,

$$lnp(X|\pi,\mu,\sum) = \sum_{n=})^{N} ln\left\{\sum_{k=1}^{K} \pi_k(N)(x_n|\mu_k,\sum_k)\right\}$$

Check the convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied return to step 2.

Results of EM

1. EM gives Soft Assignments



2. All points contribute to estimate all components
3. Each point has unit weight to contribute, but splits it across the K components.
4. Weight contributed by point to component is proportional to the likelihood that point was generated by that component.

5 References

- **An excerpt from the Python Data Science Handbook by Jake Vanderlas**
  https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html
- **The classical EM Algorithm from Machine Learning course of Columbia University by Martin Haugh**
  http://www.columbia.edu/ mh2078/MachineLearningORFE/EM$_A lgorithm.pdf$
- **An handout on Guassian Mixture Model and EM by University of Toronto**
  https://www.cs.toronto.edu/ jlucas/teaching/csc411/lectures/lec15$_1$6$_h andout.pdf$

- **Lecture notes on EM algorithm by Tengyu Ma and Andrew Ng: Stanford University**
  http://cs229.stanford.edu/notes-spring2019/cs229-notes8-2.pdf
- **A blog post on GMM in towards data science**
  https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95