Expectation Maximization

Objective

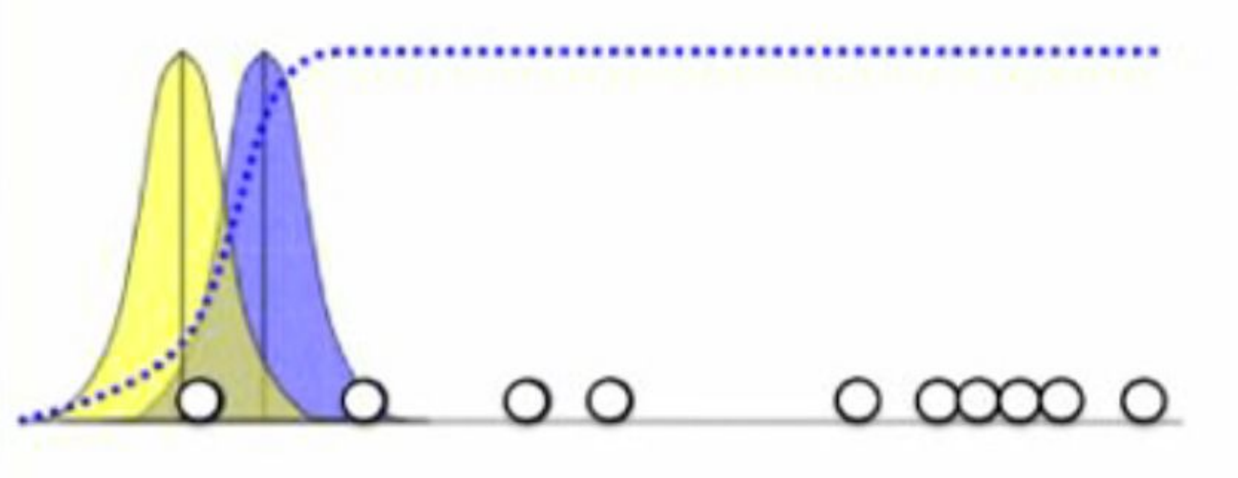
- Classify different class clusters using their features with probability distribution functions.

Overview: Gaussian

- Given a Data point of set **D.**
- A data point may belong one of the **classes (I)** that also belongs to one of the **k features (K).**
- Each class **I** can be represented as a **K-degree** gaussian distribution (pdf).
- You are assumed that you don’t know the mean and variance of all pdf hence they are arbitrary given.
- The mean is a **K** length vector while the variance is a **K*K** matrix (aka covariance matrix)

$$p_k(\underline{x}|\theta_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(\underline{x}-\underline{\mu}_k)^t \Sigma_k^{-1}(\underline{x}-\underline{\mu}_k)}$$

Gaussian distribution in d-dimensions



In the pic, you don’t know if the white samples belong to the yellow or the blue class cluster while you also don’t know the correct gaussian distribution of the yellow or the blue class cluster -pic from Victor Lavrenko

Overview: Probability of probability

- Now that each sample **x** has a probability **p** on how likely belongs to the **l** pdf, we arbitrary set a prior probability **P** to each **l** pdf as long as the sum of all **P** from **l** pdfs is 1.
- The probability that sample **x** belongs to to **i**th pdf of the prior probability **P** is given by the second equation.

$$p(\mathbf{x}|\Theta)=\sum_{l=1}^{M} P_l\, p_l(\mathbf{x}|\boldsymbol{\theta}_l)$$

$$P(l|x_i,\boldsymbol{\Theta})=\frac{P_l\, p_l(x_i|\theta_l)}{p(x_i|\boldsymbol{\Theta})}=\frac{P_l\, p_l(x_i|\theta_l)}{\sum_{l=1}^{M} P_l\, p_l(x_i|\theta_l)}$$

First Eq: Sum of the product of the prior probabilities and their respective pdf along M classes.
Secon Eq: Likelyhood of the sample to belong to one of the classes M

Overview: Update rules

- To improve the accuracy of the mean, variance and prior probability expectation maximization is used thus update rules are applied and iterated.
- The solution is guaranteed to converge
- To stop the iteration a difference threshold is placed. (Mine was if the minimum absolute difference of either the 3 variables is less than 10**-4)

UPDATE RULES:

$$P_l^{new} = \frac{1}{N}\sum_{i=1}^{N} P(l|\mathbf{x}_i, \boldsymbol{\Theta}^g) \qquad \boldsymbol{\mu}_l^{new} = \frac{\sum_{i=1}^{N} \mathbf{x}_i P(l|\mathbf{x}_i, \boldsymbol{\Theta}^g)}{\sum_{i=1}^{N} P(l|\mathbf{x}_i, \boldsymbol{\Theta}^g)}$$

$$\boldsymbol{\Sigma}_l^{new} = \frac{\sum_{i=1}^{N} P(l|\mathbf{x}_i, \boldsymbol{\Theta}^g)(\mathbf{x}_i - \boldsymbol{\mu}_l^{new})(\mathbf{x}_i - \boldsymbol{\mu}_l^{new})^T}{\sum_{i=1}^{N} P(l|\mathbf{x}_i, \boldsymbol{\Theta}^g)}$$

Update rules of mean, variance and prior probability

Case: 2 classes, 1 feature

- It means there are 2 pdfs where the mean and variance are length 1 array and 1x1 matrix respectively.

- <u>Initial Parameters:</u>

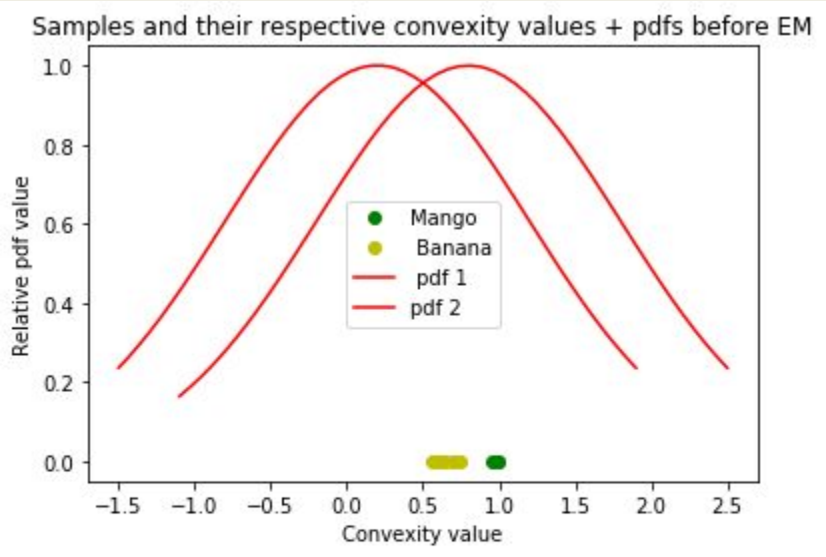
  mean_list = np.array([.2,.8])

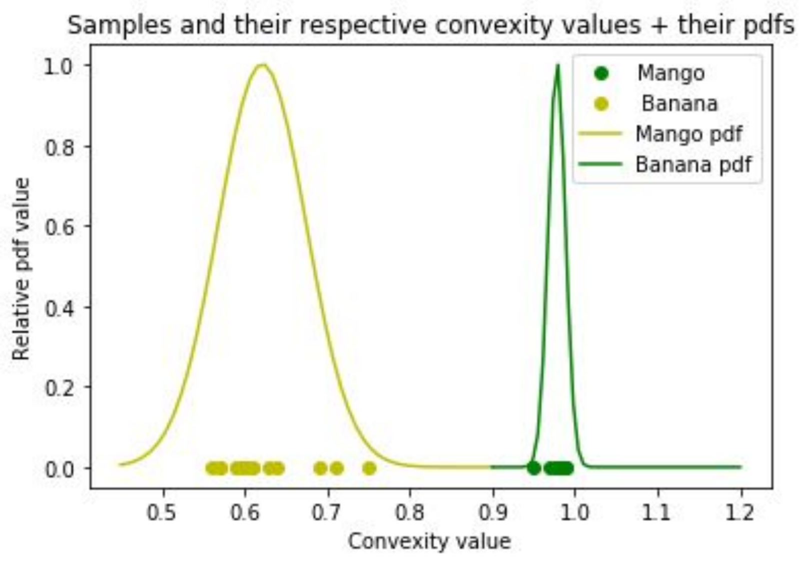
  variance_list = np.array([1,1])

  prior_list = np.array([1/2,1/2])

- <u>Data used:</u>

  Convexity values of mangoes and bananas.



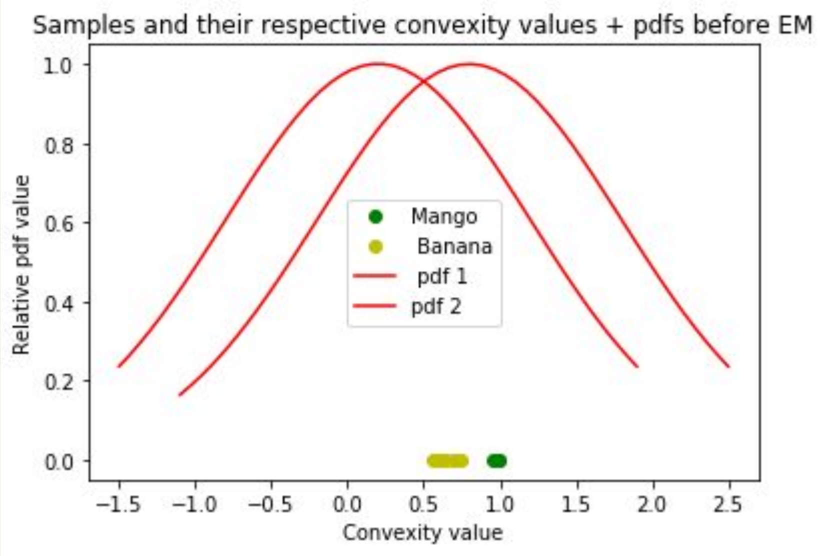


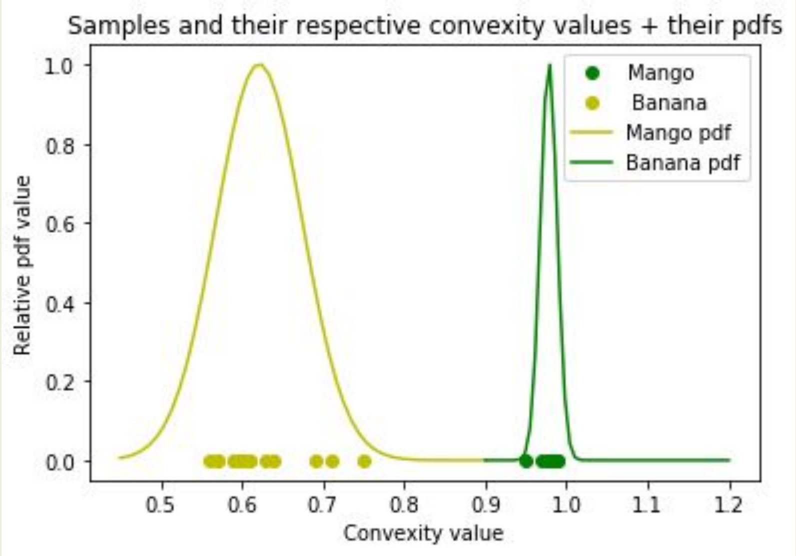
Case: 2 classes, 1 feature

- The plot showed the pdf of the mango and banana class. After a significant amount of iterations (difference threshold was achieved)

- Before, expectation maximization the initial pdfs does not fit either the two class clusters. (I didn't know variance of 1 is big)

- After expectation maximization via update rules iterating it many times, the two pdfs showed an identity where one of them truly belongs to the mango cluster while the other belongs to the banana cluster.





Other cases to try in future:

- Cases of 2 or more features. The mean will become a vector mean while the variance becomes a covariance matrix. I stick to the simplest case first since I had to catch up with the other activities.