Machine Learning and Computer Vision - K-means, EM

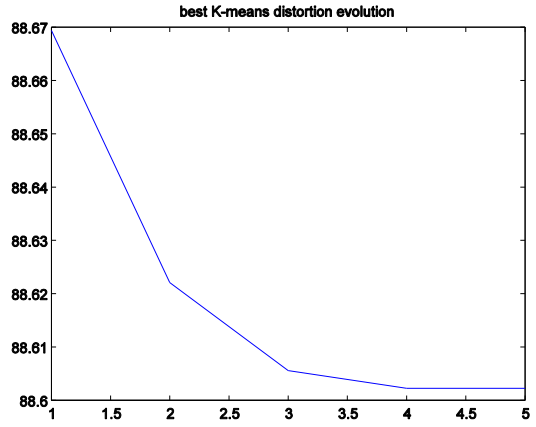
Olivier Jais-Nielsen

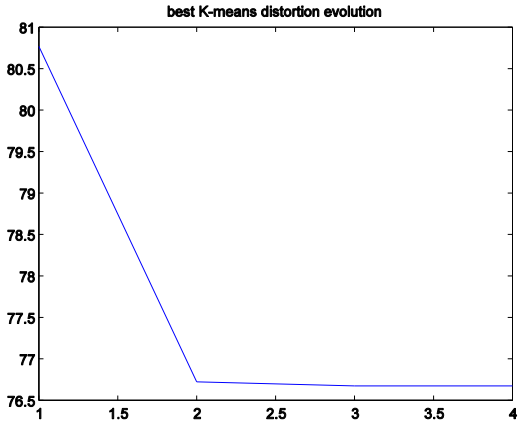
1. K-means clustering

For each sample set, we apply 20 k-means clustering with random initialization centers among the points, and $k = 3$ cluters. Each k-means clustering is considered converged when no change occur in the cluster affectations. For each run, we keep the distortion of the clusters along with the iterations. For some centers $(c_i)_{1 \le i \le k}$, clusters $(C_i)_{1 \le i \le k}$ and points $(x_i)_{1 \le i \le n}$, the distortion is defined as:

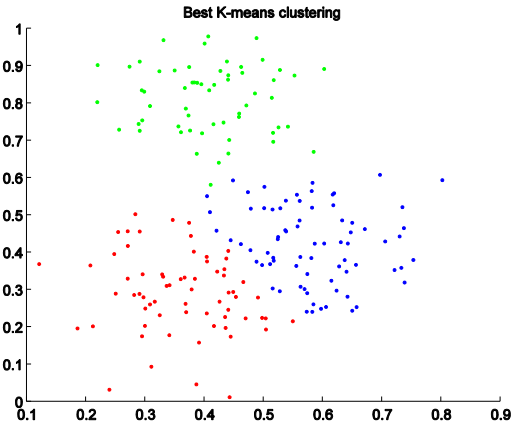
$$D = \sum_{i=1}^{k} \sum_{j \in C_i} \|c_i - x_j\|^2$$

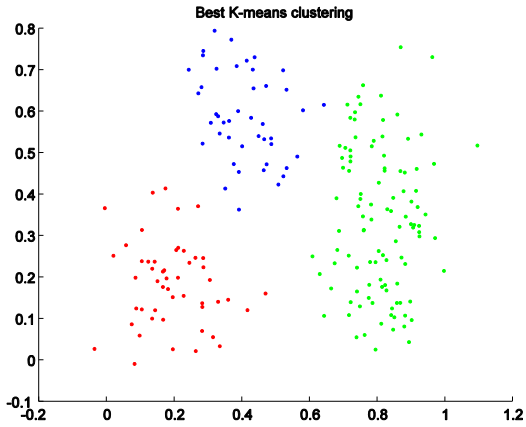
The run that ends up with the lowest distortion is kept. Here are the evolutions of the distortion corresponding to such runs for each data set.





The resulting clusterings are the following.



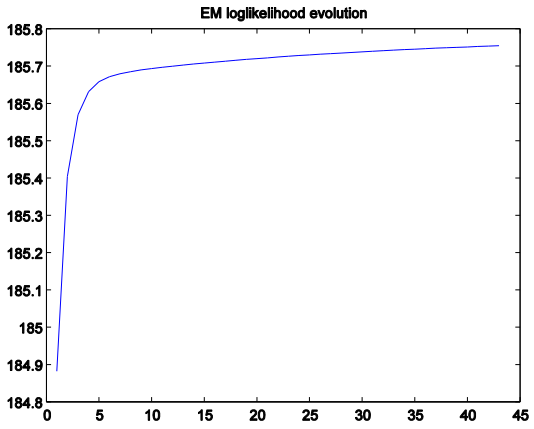


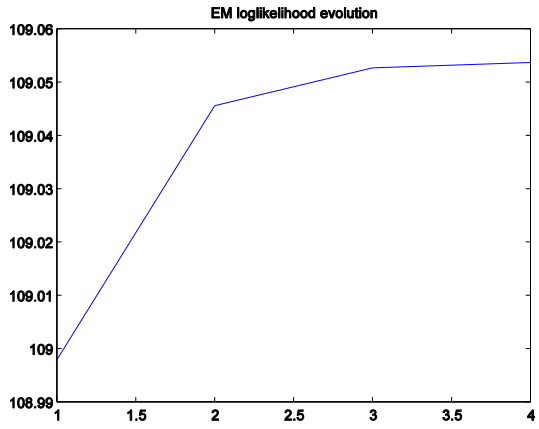
2. EM algorithm

We initialize the EM algorithm for a mixture of three Gaussian distribution using the previous clustering:

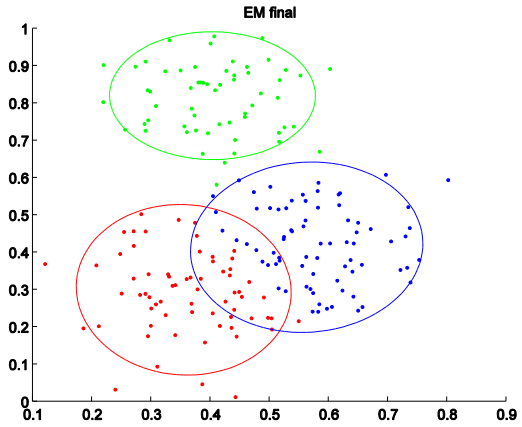
- The initial means of the Gaussians are the centers of the clustering.
- The initial mixing coefficients are set as the empirical probability of belonging to each cluster.
- The initial covariance matrices of the Gaussians are the empirical Covariance matrices for each cluster.

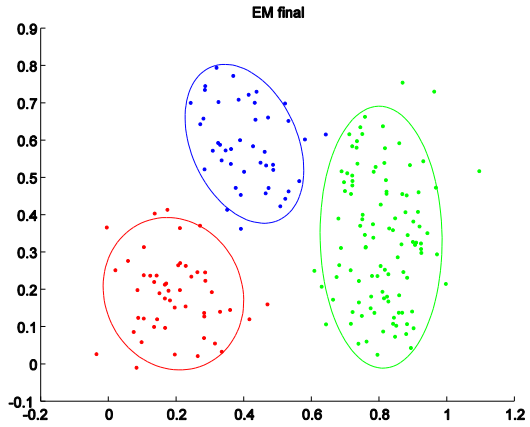
The log likelihood is monitored along the EM iterations and the algorithm is stopped when it stops growing. Here are the evolutions of the log likelihood for each data set.





The resulting Gaussians are represented as ellipsoids centered in the means and with axes and small axes proportional to the principal axes of the covariance matrices.





3. Classification

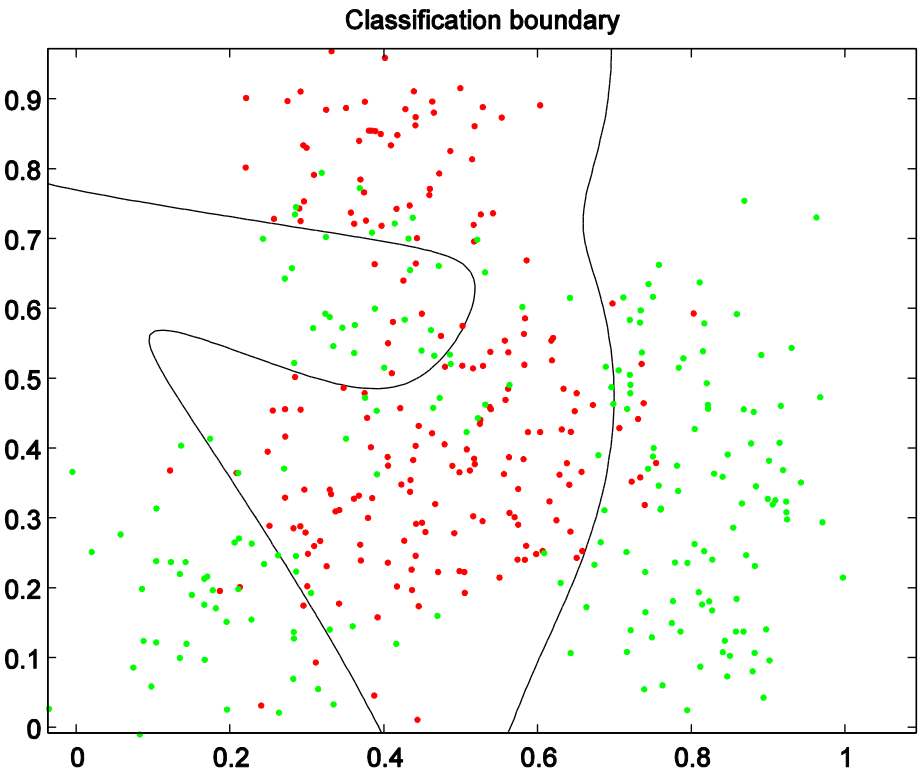
Once the two previous models trained, for any point x with label y, we can evaluate the posterior probability of its belonging to either of the classes to which belong the two data sets (i. e. whether $y = 1$ or $y = 0$). A rule of classification is to give it the class corresponding to the highest of these probabilities. We therefore consider the odd ratio:

$$r(x) = \frac{\mathbb{P}(y = 1|x)}{\mathbb{P}(y = 0|x)}$$

The boundary between both classes is then defined as:

$$r(x) = 1$$

We get the following boundary.



Such a classification rule gives a misclassification rate of approximately 15% on the training data. With the same data, we get a rate of approximately 12% with Adaboost for the training data.