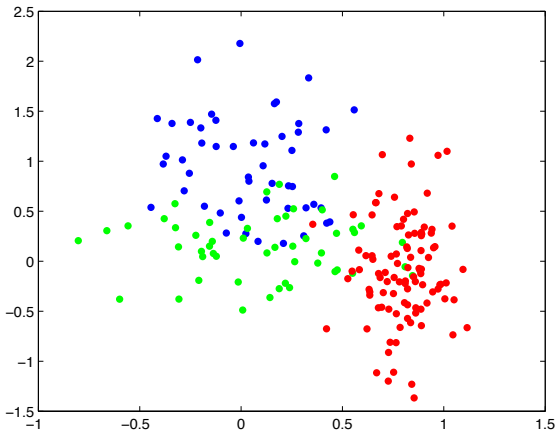
Exercise Nr 5, Machine Learning, Fall 2015

An elementary K-means case study

I created a set of 200 2-dimensional data points (x_i, y_i) by sampling 50 points from a 2-dimensional Gaussian, another 50 points from another such Gaussian, and 100 points from a third such Gaussian. The figure below shows the points in three colors representing the three Gaussian sources.



The dataset is available (in text format) on the course homepage (file `xypoints.txt`).

Your task (using Python or Matlab as usual):

Run K-means to cluster the dataset, in three separate runs with $K = 2, 3, 4$. Write your code from scratch (don't use a ready-made K-means routine, of which there are many free online). For each run, generate a figure where the points belonging to a particular cluster are rendered in a cluster-specific color. Mark in your figures the cluster means by bold points. Furthermore, add to your figures the cluster-separating lines. For any pair of clusters, say clusters C_i and C_j, the line separating the two clusters is the line made of points that are equidistant from the two cluster means. So for $K = 2$ there is one such line, for $K = 3$ there are three, and for $K = 4$ there are 6. Your separating lines may extend through the entire image (you don't have to compute and draw the Voronoi cells which are made of certain segments of these lines).

Deliverable: a zipfile containing (1) a copy of the data file `xypoints.txt` and (2) a self-contained Matlab or Python script which, when run, computes the three K-means clustering tasks and generates three figures. **Submission deadline:** Sunday November 2, midnight.

You are invited but not required to play with different initializations in order to assess experimentally how sensitive your clustering results are to the initialization. A nicely done documentation of this experimentation (add to the zipfile as a pdf document, code need not be delivered) may give you up to 30 bonus points.