Homework 5

Machine Learning 10 December, 2013
1st Semester (FEUP-PDEEC) 18-782PP CMU-Portugal

Submit by 24 December, 2013, 23h59
by email to `jaime.cardoso@fe.up.pt`
and `vikramg@andrew.cmu.edu`

Problem 1: Batman and Joker [20pts]

Even after all the attempts by the Gotham police to capture him, Joker still continues to destroy the property of the city. This time he has planted a deadly computer virus at different ($\hat{K}$) locations in the city. The virus has this strange property that it spreads to *nearby* computers very fast once triggered, but does not go too far from the source. Joker has triggered this virus 2 times already, but with slightly different properties. It is impossible to know which computer exactly is the source of the virus.

One the other hand, Batman is now old, and does not have any skills in Machine Learning. He has to choose a candidate to be the next Batman (and win the Batmobile) based on how well someone is able to identify the spread and source of the virus across the city from the data about the locations of the infected computers.

1. As your first task, analyze the geographical spread of the infected computers by plotting the two data-sets. (You may find Joker's hidden signature in the impact of the virus).

2. Implement K-means and Gaussian Mixture Model techniques to cluster the input data. Choose two different values of $K = \{4, 6\}$.

3. For both the data-sets, and for both the clustering approaches, start with 5 different initial means, and show the obtained classification.

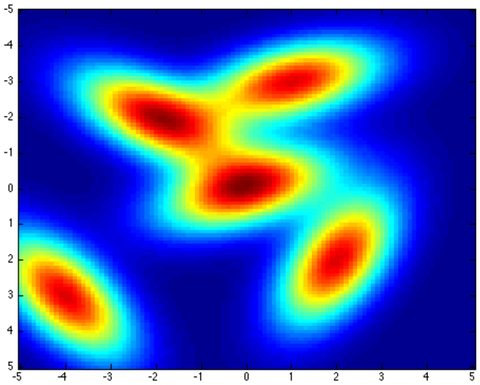
Problem 2: Hidden Markov Models [10pts]

Consider a HMM with continuous outputs. Assume that the HMM only has two states; both states have the same initial probability; the probability of changing between different states is 0.05 (that is, the transition matrix is symmetric with 0.95 in the elements in the main diagonal). The emission density function for state 1 follows a Gaussian distribution with mean 0 and standard deviation 0.2. For state 2 the emission density function is uniform in $[0, 1]$. Change the code provided in the class (file `hmmTest.m`) to compute the probability of the following sequence of length 10:
$\{‘0.7,\ 0.7,\ 0.1,\ 0.2,\ 0.3,\ 0.6,\ 0.2,\ 0.3,\ -0.1,\ 0.2\}$.
Send the modified file and write the computed probability.

Problem 3: Distributed Machine Learning [10pts]

So far we have seen the problems where all the data for clustering is available to us in one shot. But in many problems, it might be the case that different devices or computers contain only one data point, and they have to connect via a network or wireless channels to share the data.

A typical example would be a sensor network with several ($N \times N$) devices, as shown in the figure below where each device senses a physical quantity (temperature, for example). Red means a high value seen by sensor, and blue is low. Our goal is to cluster a given 2-Dimensional distributed data using a method that provides results *similar* to Gaussian Mixture Models, which means that we should be able to identity the approximate mean and approximate covariance (spread) of each cluster. For our purpose, it is sufficient to find the **direction** and the **length of major axis** of each "gaussian ellipse" in addition to the **mean**, but not the exact covariance matrix.



Assuming that each point in the 2D field is a small device able to communicate wirelessly, and when any device transmits (broadcasts), every other device can listen to the transmitted value. Each device has sufficient memory to save a large number of received broadcast messages. The challenge is to obtain the approximate GMM result (mean and the major axis) with the minimum number of broadcast messages.

Let us assume that there are N^2 devices spread uniformly in a square field. Also, state any other assumptions you make clearly.

1. Design an algorithm (a brief psuedocode in plain English) about how we can find the mean and the major axis of gaussians in distributed system model described above. State the estimated cost in terms of the total number of messages required in your algorithm's approach.

   FYI, The worst we can do is when each device shares its value one by one, and hence we can know the entire data set. Then we can use GMM in the regular way. But the total number of messages required is equal to number of devices (N^2). The mean and major axis will be statistically accurate, but we are okay with some inaccuracy if we can save in the number of messages.

2. Now assume that we have a magic tool f_{magic} that can return the global maximum (or minimum or both) of a particular data-type v over all the nodes with a cost equal to only one message.

   $$f_{magic} = MAX(v_i), \quad i = 1, 2, ..., N^2$$

   Can we now do better for finding the approximate means and the major axis in terms of number of messages. Outline a new algorithm and its cost.

(Hint: A device can transmit any one or more types of data ***it knows*** *in one message, or a function of this data. Some of the useful types of data a device can have and broadcast are: its location in x-y coordinates, sensor value etc.)*