Machine Learning
Chapter 1: Mixture models and the EM algorithm

1 Introduction

The purpose of this set of experiments is to learn the basics of Gaussian Mixture models and how to code an Expectation Maximization algorithms as well as how to track its behaviour. The results will be compared with those of the k-means algorithm.

The code must be written in Matlab of Python. For a proper training of the student in this topic, the existing EM or GMM libraries should be avoided. Students are supposed to code all functions and procedures from scratch. Nevertheless, comparisons with the results of existing libraries are encouraged, once the students hve coded and tested their own programs.

2 GMM function

In this part, the Gaussian Mixture model function is to be coded. Write a function called GMM.m whose input is the set of variables needed for the Gaussian Mixture Model, this is, a matrix containing a M column vectors of dimension D representing $\boldsymbol{\mu}_i$, $1 \leq i \leq M$, a set of matrices containing the corresponding covariance functions, and a vector π of mixture values.

The input should also contain an arbitrary set of vectors in the space. The corresponding output is their probability density value according to the Gaussian Mixture distribution

3 Gaussian distribution

In order to produce artificial data, construct a function whose inputs are a D dimensional vector (mean), a D dimensional matrix (covariance) and a scalar N. The output will produce N data according to the corresponding Gaussian distribution. It is recommended to use a circular Gaussian distribution and a linear transformation according to the Gaussian parameters.

Usin g this distribution, generate a 2 dimensional distribution with 400 data, with the parameters:

```
mu1 = [1 2];
sigma1 = [3 1; 1 2];
mu2 = [-1 -2];
sigma2 = [2 0; 0 1];
mu3 = [3 -3];
sigma3 = [1 .3; .3 1];
```

Generate 100 data for the first and second distributions, and 200 for the third one.

4 Expectation and maximization

Using the GMM function, write a function that computes the probability density values of the data and then updates the mixture parameters π_i. In order to compute the maximization, use another function that computes the mean and the variance of the GMM.

5 Experiments

5.1 EM algorithm

Write a script that iteratively updates the parameters of the GMM using the EM algorithm. For representation of the results, use the function contour of Matlab.

For representation purposes, you can use the following code

```
% Grid of coordinates for representation in the 2D space
x=linspace(-6,6,30);x=repmat(x,length(x),1);
```

```
y=x';
% Vectorization of the coordinates
z=[x(:),y(:)];

%%%Add there the code to compute the pdf 't' at each point of 'z'

figure(1)
scatter(X(1,:),X(2,:),10,'ko') %Plots the data
hold on
contour(x,y,buffer(t,sqrt(length(t)),0)) %Contours the pdf (Called 't' here)
hold off
%Represents the PDF in a 3D plot
figure(2)
surf(x,y,buffer(t,sqrt(length(t)),0),'FaceColor','interp',...
'EdgeColor','none',...
'FaceLighting','phong')

axis tight
view(-50,30)
camlight left
```

5.2 Representation of the data likelihood

Using the previously developed code, compute the log-likelihood of the dataset at each iteration and represent it. From this, propose a stopping criterion. Propose an alternate criterion based on the evolution of the parameters and compare them. Which one is better in terms of accuracy and computational burden?

5.3 Unsupervised classification

Plot a scatter of the data as a function of its PDF. Assume that each mode belongs to a different latenty variable z_i. Propose a classification scheme based on the GMM model. Represent a 2D plot of the result.

Is it possible to determine the quality of the classifier in terms of classification accuracy? Is the Occam Razor implicitly applied here? Explain it in a qualitative, intuitive manner.

5.4 Comparison to the K-means algorithm

5.4.1 Mahalanobis distance

Define the Mahalanobis distance and write a Matlab function that is able to determine the Mahalanobis distances from a set of data to a set of points $\boldsymbol{\mu}_j$ as a function of their corresponding covariance functions $\boldsymbol{\Sigma}_j$.

5.4.2 K-means procedure

Write a function that classifies a set of data as belonging to class z_j as a function to their Mahalanobis distances to points $\boldsymbol{\mu}_j$.

5.4.3 Parameter update

Write a function that updates $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ using an estimate that uses the points belonging to class z_j.

5.4.4 Parameter initialization, growing-prunning and use in an 10 dimensional problem

Summarize the parameter initialization and growing-pruning heuristics that have been presented in class. Repeat the comparison between GMM and k-means in the 8 dimensional problem of first assignment. It is known that for this problem the optimum number of clusters is 8. If a growing prunning strategy is used starting with 4 clusters and starting with 16 clusters, show that they tend to the optimum value.