Instituto Superior Técnico

Departamento de Engenharia Electrotécnica e de Computadores

Machine Learning

5^{th} Lab Assignment

Shift_________ Group number_________

Number_________ Name_______________________________________

Number_________ Name_______________________________________

EM and k-Means Algorithms

Unsupervised learning methods are important tools for the analysis and processing of data. Among other uses, this class of algorithms can play an important role in the interpretation, classification and organization of unlabeled data.

In this lab assignment we'll experiment with two algorithms from this class: the *Expectation-Maximization* (EM) algorithm, applied to the estimation of probability density functions, and the *k-Means* algorithm, applied to data clustering.

1 Estimation of Gaussian distributions

1. Assume that a given data set was drawn from a multidimensional Gaussian distribution.

1.1. Indicate which parameters need to be estimated to characterize the distribution.

1.2. When estimating a model from data, knowing the number of parameters to be estimated is important, because it is related to the number of training patterns that are needed for a good estimation of the model. Indicate the number of distinct real parameters that need to be estimated for the multidimensional Gaussian distribution, if the data consist of P patterns, each pattern being a D-dimensional vector.

1.3. Indicate the equations that allow the estimation of these parameters, and the kind of estimator that those equations correspond to.

2 Estimation of Gaussian mixtures

A convenient form of parametric estimation of probability density functions is to approximate the distribution with a mixture of Gaussians, the parameters of the mixture being estimated through the EM algorithm.

The probability density function (pdf) of a mixture of K Gaussians in D dimensions is given by

$$p(\mathbf{x}) = \sum_{k=1}^{K} w_k g_k(\mathbf{x}),$$

where w_k represents the weight of the k-th component, and $g_k(.)$ is the pdf of that component. The pdf normalization condition is equivalent to

$$\sum_{k=1}^{K} w_k = 1,$$

and a sufficient condition for $p(\mathbf{x})$ to be non-negative is $w_k \geq 0$ for all k. The pdf of each Gaussian component is given by

$$g_k(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^D \left|\mathbf{\Phi}_k\right|}}\; e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \mathbf{\Phi}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)},$$

where $\boldsymbol{\mu}_k$ represents the mean of the k-th Gaussian and $\mathbf{\Phi}_k$ is its covariance matrix. The purpose of the EM algorithm, applied to this problem, is the maximum likelihood estimation of the parameters $(w_k, \boldsymbol{\mu}_k, \mathbf{\Phi}_k)$ from a data sample (here *sample* is used in the statistical sense: it means a set of observations, not a single observation).

In MATLAB, the generation of random data with a Gaussian distribution can be performed with the RANDN function. More specifically, RANDN(P,Q) generates a $P \times Q$ matrix, with independent observations from a Gaussian distribution with zero mean and unit variance.[1]

2.1. Assume that the vector $\mathbf{s}$ represents a D - dimensional random variable with independent components, all of them with a distribution with zero mean and unit variance.

2.1.1. Indicate the mean and the covariance matrix of $\mathbf{s}$.

2.1.2. Now assume that the random variable is subject to the transformation

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{b},$$

where $\mathbf{A}$ is a $D \times D$ matrix and $\mathbf{b}$ is a D-dimensional vector. Find the mean of $\mathbf{x}$ and show that the covariance matrix of $\mathbf{x}$ is $\mathbf{A}\mathbf{A}^{\mathbf{T}}$.

In what follows, $\mathbf{A}$ is called the *generation matrix*.

2.2. In this assignment, two MATLAB programs are available for testing the EM algorithm. The first one, *tstem.m*, reads a file with a description of a mixture, generates data according to that mixture, and then proceeds to estimate the mixture parameters from those data, allowing a comparison between the actual mixture parameters and the estimated ones. The second program, *emest.m*, performs the parameter estimation based on a file containing observations. A brief description of the programs and of all auxiliary functions is available, in MATLAB, through the command *help [program or function]*.

The files with a description of Gaussian mixtures (both as input, for the generation of synthetic data, and as output, for presentation of the estimated

[1] Formally, the observations are not strictly independent, because they are generated by a pseudo-random number generator.

parameters) are text files with the following format:

```
Number of dimensions (D)
Number of Gaussian components (N)
```
$w_1 ... w_N$

$\boldsymbol{\mu}_1$ `(`D `components)`

$\mathbf{B}_1$ `(`D^2 `components)`

$\boldsymbol{\mu}_2$ `(`D `components)`

$\mathbf{B}_2$ `(`D^2 `components)`

`...`

$\boldsymbol{\mu}_N$ `(`D `components)`

$\mathbf{B}_N$ `(`D^2 `components)`

The **B** matrices can represent covariance matrices or generation matrices. When reading (*readmix* function) or writing (savemix function) a file with mixture parameters, it is possible to specify which of the two formats is to be used.

2.2.1. Test the *tstem* program with 100 and with 1000 data points, generated from the mixture specified in the file 'mix.txt', applying the EM algorithm with 2, 4, 5 and 10 Gaussian components. Make several runs for each case, in order to get a good idea of the behavior of the algorithm (each run uses a randomly chosen training set and a randomly chosen initialization). In each run, use the figures to compare the estimated distribution with the distribution used to generate the data. Fill the following table with the log-likelihoods obtained in each of the cases (use, for each of the cases, the log-likelihood from one of the runs that you have made for that case). Then, comment on what you have observed, both the estimated distributions and the log-likelihood values.

Log-likelihood

Number of Gaussians	100 data points	1000 data points
2		
4		
5		
10		

2.2.2. Since the EM algorithm estimates the probability density by maximizing the likelihood, it might seem that the likelihood of the data should be maximal for the true distribution, which was used to generate the data. However, the final likelihood, computed with the model estimated by the EM algorithm, may be higher than the one computed with the true distribution. Comment on this statement and, more generally, on the relationships among likelihoods, sample size, number of Gaussian components and parameter estimates obtained by the algorithm.

2.3.1. Estimate the number of Gaussians and the mixture parameters for the data contained in file 'data_2d.txt'. Indicate the results that you obtained, and explain how you have estimated the number of Gaussians.

2.3.2. Estimate the number of Gaussians and the mixture parameters for the data contained in file 'data_3d.txt'. Indicate the results that you obtained, and explain how you have estimated the number of Gaussians.

2.4. The file 'normal.dat' contains 10000 sets of values from three sensors from a certain system, under normal operating conditions. The file 'unknown.dat' also contains 10000 sets of values from the same sensors, but it is unknown whether the system's operation had any anomalous periods in the time during which these data were collected. Find whether the data in 'unknown.dat' show signs of anomalies. Explain what you have done. If you have found anomalies, indicate the starting and ending times of the anomalous periods, measured in number of samples. You may find the function *mixdis* useful.

3 k-Means Algorithm

The k-means algorithm is an unsupervised clustering algorithm which is very commonly used. The cost function that is minimized is the mean of the squared errors between the training patterns and the corresponding nearest centers. Denoting by $\mathbf{x}_i, i = 1, ..., N$, the training observations, this function is given by

$$E = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}^i - \mathbf{c}^{k(i)}\|^2,$$

where $\mathbf{c}^{k(i)}$ is the center that is closest to $\mathbf{x}^i$.

The test of the k-means algorithm with synthetic data can be performed with program *kmeanstest*. The program first generates synthetic data from a mixture of Gaussians, using a parameter file with a format identical to the one used for the EM algorithm, and then performs the clustering of those data with a user-specified number of centers. As before, more information about the program can be obtained by means of the *help* command.

3.1. Test the k-means algorithm with data generated using the file 'mix1.txt'. Fill the following table, performing three tests for each case (indicate, in the table, the results of the three tests; retain only four digits after the decimal point):

Mean squared error

Number of centers	100 data points	2000 data points
2		
4		
7		

3.2. Test the algorithm using the file 'mix2.txt', using 1000 data points, with 4 and with 6 centers. Again, indicate the results of three tests for each case.

Mean squared error

Number of centers	1000 data points
4	
6	

3.3. Briefly comment on the results of **3.1** and **3.2**, in what concerns (1) the influence of the number of training data; (2) the correctness of the clustering that is obtained, as a function of the number of centers that are used; (3) the appropriateness of the k-means clustering method for identifying clusters, as a function of the clusters' sizes, shapes and relative positions.