EM Algorithm and Semi-Supervised Learning

The aim of this tutorial is to try EM algorithm and get familiar with semi-supervised learning.

1 EM algorithm

1. Generate random numbers with univariate normal distribution using matlab function `randn`. Display empirical probability density (using function `bar(centers, bins)`).

   Beware: `[bins centers] = hist(x)`

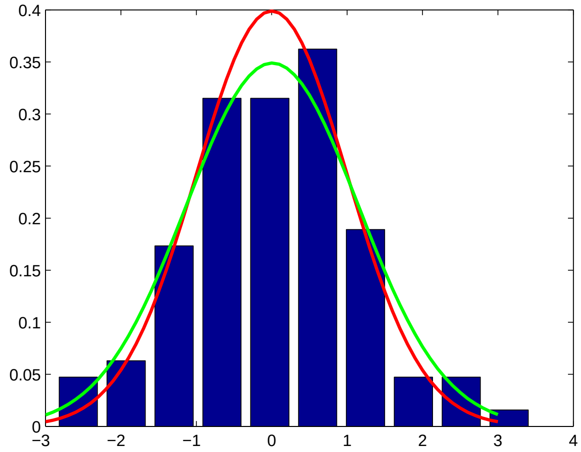
   Function `hist` returns histogram with counts instead of relative frequencies. After normalisation you get empirical probability function. To get empirical density you should divide the empirical probability function by the width of the histogram's bins. Why?

   Display theoretical probability density in the same figure using matlab function `normpdf`.

   Estimate parameters of the normal distribution from the generated data and display corresponding probability density (using function `normpdf`) in the same figure with empirical and theoretical probability densities. Expected result is shown in figure 1.

2. Generate 200 samples from mixture of two normal distributions ($\sigma_1 = 4$, $\mu_1 = 3$, $\sigma_2 = 2$, $\mu_2 = 15$).

   In case of two different normal distributions (Gaussian Mixture Model) it is not possible to calculate mean and standard deviation for each component of the mixture as the empirical mean and empirical standard deviation from the whole dataset. You need to split the observed data



Obrázek 1: Expected result from task 1. Theoretical probability density is shown in red colour, estimated probability density is shown in green colour.

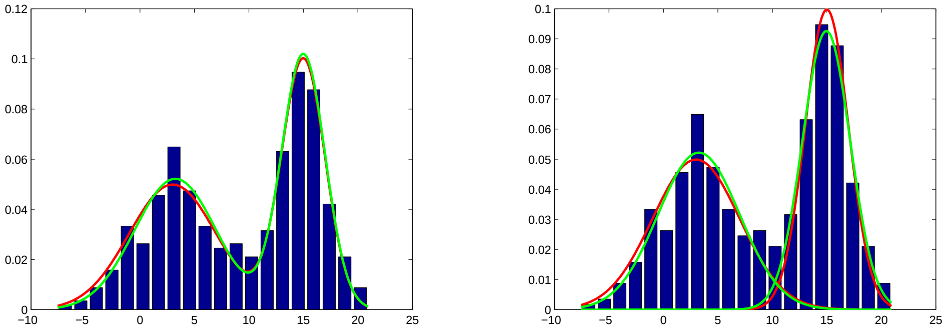
into the particular normal distributions, each with own mean and standard deviation. One option how to get these parameters for particular normal deviations is to use EM algorithm.

Find means and standard deviations for each component of the mixture using function ”EM.m”. Display empirical, theoretical and estimated probability density for the mixture of normal distributions in one figure. (Expected result is shown in the left panel of figure 2.)

Display empirical, theoretical and estimated probability density for each component of the mixture in another figure. (Expected result is shown in the right panel of figure 2.)

2 Semi-Supervised Learning

3. Modify EM algorithm for semi-supervised clustering. Add a new input parameter *classification* to the function EM, which will contain assignment of some samples to clusters (1,2,...). Use 0 for samples for which you do not have assignment. In EM algorithm, fix assignment to

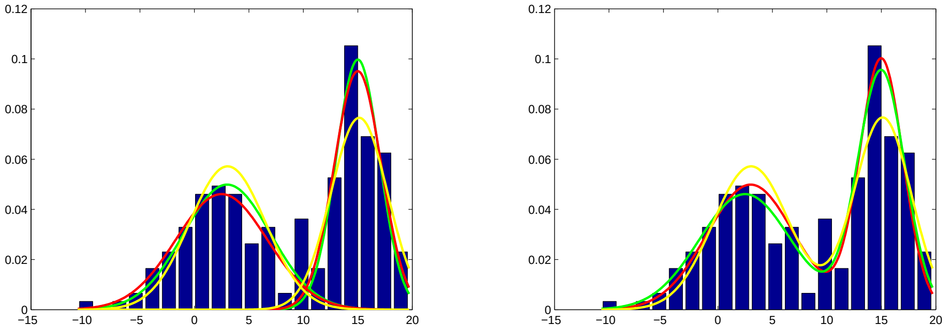


Obrázek 2: Expected results from task 2. Theoretical probability density os shown in red colour, estimated probability density is shown in green colour.

clusters for those samples, for which the assignment is known from the input.

Perform the following experiment. Modify generation of data - add classification, i.e. information about membership of samples to clusters. Use this information just for a randomly chosen subset of samples for training - the rest will be used for testing. Estimate parameters of the mixture in two ways. First, estimate parameters only from samples, for which you know classification (here, you do not use EM algorithm). Second, estimate parameters from all samples using modified EM algorithm. Compare accuracy of sample assignment to clusters using these two methods in cases when you know classification for 10%, 20% and 50% samples. Exploit the fact, that you know classification for every sample. Specifically, use all samples for training - recall that you know assignment to clusters just for some of them. Calculate accuracy as the fraction of correctly assigned samples for which you did not know the assignment at the beginning.

Expected result is shown in figure 3.



Obrázek 3: Expected result from task 3. Theoretical probability density is shown in red colour, probability density estimated using EM algorithm is shown in green colour, probability density estimated only from annotated samples is shown in yellow colour.