MTAT.03.227 Machine Learning
Spring 2016 / Exercise session XII
Nominal score: 10p
Maximum score: 15p
Deadline: 26th of April 16:15 EET

1. We argued in the lecture that hard-clustering methods can become unstable due to outliers, which corrupt the estimation of cluster parameters. One way to make the clustering method more robust is to throw out 5-10% of cluster points that have the lowest likelihood to be part of the cluster.

   (a) Modify the hard-clustering algorithm for two-dimensional Gaussian mixture model given in the file `hard-clustering-gmm.R` so that up to α fraction of points assigned to the cluster are removed before parameter estimation. Study whether the modified algorithm remains stable on challenge datasets stored in the file `gmm-challenges.Rdata`. For that use the same initial state and see whether the modified algorithm converges to the same set of parameters. (**1p**)

   Clarification: A simple comparative graph, which shows cluster centres and data ellipses for both methods are enough.

   Hint: The command `dataEllipse(x, y, groups, levels = 0.9)` is enough to visualise the end results of both clustering methods. You can use `ellipse` command to visualise covariance matrices directly.

   (b) Another alternative is to do the removal globally. That is, for each data point you compute the maximal likelihood

   $$p(\boldsymbol{x}_i) = \max_i \mathrm{p}[\boldsymbol{x}_i|\boldsymbol{\Theta}_j]$$

   and in each iteration remove α fraction of points with lowest $p(\boldsymbol{x}_i)$ scores. Modify the hard-clustering algorithm to accommodate this tweak. Modify both tweaked algorithms so that they would output also the log-likelihood of selected data $\log \mathrm{p}[\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_\ell}|\boldsymbol{z}, \boldsymbol{\Theta}]$ in each iteration. Plot corresponding graphs? Do these methods have different behaviour? If so which of them is more stable? (**1p**)

   Clarification: Study the behaviour of both modified algorithms on `cdata3` on which you have added 10% outlier data which is uniformly sampled form the range $[-10, 10] \times [-10, 10]$.

   (c) Study the robustness of the modified hard-clustering algorithm. For that use `cdata3` and an initial configuration of parameters so that method converges to sensible outcome. Now add various levels of outlier data which is uniformly sampled form the range $[-10, 10] \times [-10, 10]$ to `cdata3` and study when the algorithm still converges to the sensible outcome (observe cluster centres). Try different outlier levels $\beta = 1, 2, 5, 10, 15, 20, 30, 40, 50\%$ and $\alpha = 1, 5, 10\%$. How much noise can be tolerated by the modified algorithm. (**1p**)

2. The soft-clustering does not assign a point into a particular cluster. Instead, it defines a probability w_{ij} that a point $\boldsymbol{x}_i$ belongs to the jth cluster. As such, it it often intractable to get a closed-form update step for the cluster parameters $\boldsymbol{\Theta}_j$. Here, we study how to convert the corresponding parameter estimation step in the hard-clustering algorithm:

$$\boldsymbol{\Theta}_j^{(i+1)} = \operatorname*{argmax}_{\boldsymbol{\Theta}_j} \mathrm{p}[\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_{n_j}} | \boldsymbol{\Theta}_j]$$

into a parameter estimation step in the soft-clustering algorithm. We demonstrate this on the two-dimensional Gaussian distribution and on the two-dimensional Laplacian distribution. The first has a closed form solution for parameter fitting if the data is weighted, while the second has no closed form solution and thus the method has a practical value.

   (a) The file `fractional-weights.Rdata` contains two-dimensional data that has been split into two clusters. For the Gaussian data the corresponding cluster weights are stored in the matrix `gauss.weights`. Estimate the corresponding cluster parameters using the standard hard-clustering update step

$$\boldsymbol{\mu} = \frac{1}{|\mathcal{I}|} \cdot \sum_{i \in \mathcal{I}} \boldsymbol{x_i}$$
$$\Sigma = \frac{1}{|\mathcal{I}|} \cdot X_c^T X_c$$

   where $\mathcal{I}$ is the set of cluster points and X_c is a centred data matrix, i.e., the kth row is $\boldsymbol{x}_{i_k} - \boldsymbol{\mu}$. As the counts are fractional, replace points with $\ell = 10, 100, 1000$ copies and assign $c_{ij} = \texttt{round}(w_{ij} \cdot \ell)$ copies to the jth cluster. Verify that the estimates converge to the correct soft-clustering update step

$$n_j = \sum_{i=1}^{n} w_{ij}$$
$$\boldsymbol{\mu}_j = \frac{1}{n_j} \cdot \sum_{i=1}^{n} w_{ij} \boldsymbol{x_i}$$
$$\Sigma_j = \frac{1}{n_j} \cdot X_c^T \mathsf{diag}(w_{*,j}) X_c$$

   where X_c is centred matrix with rows $\boldsymbol{x}_i - \boldsymbol{\mu}_j$ and $\mathsf{diag}(w_{*,j})$ is a diagonal matrix containing the weights of the data points in the main diagonal.

   (b) Now that you know how to approximate the soft-clustering update step with the hard-clustering step, use the function `FitLaplacian2D` to find new parameters for the Laplacian clusters. (**1p**)

3. The expectation-maximisation algorithm for clustering consists of two steps. In the E-step, we compute cluster weights:

$$w_{ij} = \Pr\left[z_i = j | \boldsymbol{\Theta}, \boldsymbol{x}_i\right] = \frac{\Pr\left[\boldsymbol{x}_i, z_i = j | \boldsymbol{\Theta}\right]}{\Pr\left[\boldsymbol{x}_i | \boldsymbol{\Theta}\right]}$$

and in the M-step we recalibrate cluster parameters. For the Gaussian and Laplacian mixture distributions, the corresponding update step was specified in the previous exercise.

(a) Convert the hard-clustering algorithm for the Gaussian mixture model given in the file `hard-clustering-gmm.R` into the EM algorithm. Compare the behaviour of both algorithms on `gauss.data` from the file `fractional-weights.Rdata`. Choose a reasonable initial condition but not a perfect solution. Draw the corresponding output plots and interpret results. (**1**p)

Hint: The function `dataEllipse` also accepts weights.

(b) Modify the algorithm further so that it would compute the lower bound $F(q_i, \boldsymbol{\Theta}^{(i)})$ and the actual log-likelihood

$$\log \Pr\left[\boldsymbol{\Theta}^{(i)} | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\right] = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} \Pr\left[z_i = j | \boldsymbol{\Theta}\right] \cdot \mathrm{p}[\boldsymbol{x}_i | \boldsymbol{z}_i = j, \boldsymbol{\Theta}_j] \right)$$

and visualise the convergence similarly to the `HardGMMClustering2D` function, i.e., representing the value of $F(q, \boldsymbol{\Theta})$ after both steps. (**1p**)

(c) Study the stability of the soft-clustering algorithm using the `cdata3` challenge stored in the file `gmm-challenges.Rdata`. Again, add various levels of outlier data which is uniformly sampled form the range $[-10, 10] \times [-10, 10]$ to `cdata3` and study when the algorithm still converges to the sensible outcome (observe cluster centres). Try different outlier levels $\beta = 1, 2, 5, 10, 15, 20, 30, 40, 50\%$. (**1p**)

(d) Use global removal of outlier points to make the soft-clustering algorithm more robust. That is, for each data point you compute the maximal likelihood

$$p(\boldsymbol{x}_i) = \max_i \mathrm{p}[\boldsymbol{x}_i | \boldsymbol{\Theta}_j]$$

and in each iteration remove α fraction of points with lowest $p(\boldsymbol{x}_i)$ scores. Does this modification make the algorithm more stable against the noise? Test against the same noise levels. (**1p**)

(e) Modify the soft-clustering algorithm so that the clusters are defined in terms of Laplacian distributions. For that use the update step from the previous exercise. Test the algorithm on `laplace.data`. Choose a reasonable initial condition but not a perfect solution. Draw the corresponding output plots and interpret results. (**1**p)

4. The aim of this exercise is to model measurements of epigenetic data. Your task is to build a corresponding simplified mixture model and corresponding EM algorithm for reconstructing the data sources.

   (a) Biologists often measure signals on the DNA molecules extracted from a group of cells. Let us assume that there are k prototype profiles for the epigenetic data. Each of them is a m-element zero-one vector $\boldsymbol{a}_j$ containing a long streaks of zeroes and ones. Define a data generation algorithm that generates such vectors with average block length ℓ. Justify your choice. (**2p**)

   (b) Individual observations are generated by randomly selecting a two-element window and coping the window with mutation probability δ. The corresponding observation $\boldsymbol{y}$ consists of the start location of the window and measurements of two copies. Each profile $\boldsymbol{a}_j$ is selected for copying with the probability λ_i. Define the corresponding probabilistic model and write the corresponding data generation algorithm. (**2p**)

   (c) Define corresponding hard-clustering algorithm that takes in individual observations and reconstructs epigenetic profiles $\boldsymbol{a}_j$. Test the corresponding algorithm and estimate what must be the average coverage of individual locations in order to get a reasonable reconstruction. Try different parameters. (**5p**)

   (d) Define corresponding soft-clustering algorithm that takes in individual observations and reconstructs epigenetic profiles $\boldsymbol{a}_j$. Test the corresponding algorithm and estimate what must be the average coverage of individual locations in order to get a reasonable reconstruction. Try different parameters. (**5p**)