MTAT.03.227 Machine Learning
Spring 2016 / Exercise session XI
Nominal score: 10p
Maximum score: 15p
Deadline: 19th of April 16:15 EET

1. Let 0010, 1011, 1001, 0011, 1011 represent features present or missing in the variations of the same text coming from the different sources. Find out the most probable history based on naive mutation model. Recall that the maximal likelihood solution can be found by solving the following minimisation task:

$$|E| + \tau(p) \cdot \sum_{\boldsymbol{u} \to \boldsymbol{v}} h(\boldsymbol{u}, \boldsymbol{v}) \to \min$$

where $|E|$ is the number of arcs in the tree and $\tau(p)$ depends on the mutation probability. Find an optimal solution for each tree size $|E| = 6, \ldots, 16$ and corresponding regions of mutation probabilities $p_k \in [a_k, b_k]$ where the tree of size k provides an optimal solution. (**3p**)

Clarification: Since the task is pure combinatorial minimisation task that is easier to program in any other language than GNU R, you are free to use any program language at your will, provided that you document how you look through all shapes of the trees.

2. Recall that the k-means algorithm is a two-step minimisation procedure for maximising the likelihood of data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ by tuning cluster centres $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ and labels $z_1, \ldots, z_n$. According to the underlying probabilistic model each data point $\boldsymbol{x}_i$ is generated by taking the corresponding cluster centre $\boldsymbol{\mu}_{z_i}$ and adding a random Gaussian noise $\mathcal{N}(0, \sigma)$ for each coordinate. We showed in the lecture that the likelihood maximisation is equivalent to the following minimisation task:

$$F(\boldsymbol{\Theta}, \boldsymbol{z}) = \sum_{i=1}^{n} \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_{z_i} \right\|^2 \to \min$$

where $\boldsymbol{\Theta} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k)$ denotes all model parameters.

   (a) Implement the two-step minimisation procedure. First, implement the step that finds optimal values for $\boldsymbol{z}$ given fixed $\boldsymbol{\Theta}$. Second, implement the step that finds optimal $\boldsymbol{\Theta}$ values given fixed $\boldsymbol{z}$. Load the data `cdata1` from the file `gmm-challenges.Rdata` and initialise three cluster centres by choosing $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3$ randomly form the range $[-4, 4] \times [-4, 4]$. Repeat minimisation steps 100 times and visualise the end result by colouring data points into three different colours and showing the final locations of the cluster centres. (**1p**)

(b) Now that you know a good estimate for the labels and cluster centres, you can find the mixture probabilities $\lambda_j = \Pr[z = j]$ as fractions of corresponding cluster labels. You can also estimate the parameter σ in the original model by computing standard deviations of x and y values of error components $\boldsymbol{x}_i - \boldsymbol{\mu}_{z_i}$. They should be roughly equal. Now that you have fixed all model parameters, you can test the applicability of the model by generating the same amount of data. Write a corresponding function and generate the data and compare it visually with `cdata1`. Do you believe that the model adequately describes the data? (**1p**)

(c) Load the data `cdata2` and `cdata3` from `gmm-challenges.Rdata` and find cluster centres by repeating the k-means minimisation procedure. As before estimate mixture probabilities $\lambda_1, \lambda_2, \lambda_3$ and overall variance parameter σ. Simulate the data as before and compare with the originals. Does the model adequately describes the data? (**1p**)

3. As the original k-means model seems to be inadequate for the dataset `cdata2` from the file `gmm-challenges.Rdata`, we need to modify the probabilistic model. As the visual inspection clearly shows that the variance in different clusters is different, we should consider a modified data generation where each data point is generated by taking the corresponding cluster centre $\boldsymbol{\mu}_{z_i}$ and adding a random Gaussian noise $\mathcal{N}(0, \sigma_{z_i})$ for each coordinate. The corresponding likelihood can be expressed as

$$\mathrm{p}[\boldsymbol{x_i}|z_i = j, \boldsymbol{\Theta}] = \frac{1}{2\pi\sigma_j^2} \cdot \exp\left(-\frac{1}{2\sigma_j^2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right)$$

and thus we must solve the following minimisation task

$$F(\boldsymbol{\Theta}, \boldsymbol{z}) = 2 \cdot \sum_{i=1}^{n} \log(\sigma_{z_i}) + \sum_{i=1}^{n} \frac{\left\|\boldsymbol{x}_i - \boldsymbol{\mu}_{z_i}\right\|^2}{2\sigma_{z_j}^2} \to \min$$

where $\boldsymbol{\Theta} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k, \sigma_1, \ldots, \sigma_k)$ denotes all model parameters.

(a) Implement the two-step minimisation procedure. First, implement the step that finds optimal values for $\boldsymbol{z}$ given fixed $\boldsymbol{\Theta}$. Second, implement the step that finds optimal $\boldsymbol{\Theta}$ values given fixed $\boldsymbol{z}$. Load the data `cdata2` from the file `gmm-challenges.Rdata` and initialise three cluster centres by choosing $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3$ randomly form the range $[-4, 4] \times [-4, 4]$ and set all variance parameters $\sigma_1 = \sigma_2 = \sigma_3 = 1$. Repeat minimisation steps 100 times and visualise the end result by colouring data points into three different colours and showing the final locations of the cluster centres. (**1p**)

Hint: The simplest but good enough way to fix the cluster variance is to compute the standard deviation for the x and y coordinates and set the variance as an average.

Bonus: You get an extra point if you find the optimal value of the variance σ_j that minimises $F(\boldsymbol{\Theta}, \boldsymbol{z})$.

(b) Now that you know a good estimate for the labels and cluster parameters, you can find the mixture probabilities $\lambda_j = \Pr[z = j]$ as fractions of corresponding cluster labels. Now that you have fixed all model parameters, you can test the applicability of the model by generating the same amount of data. Write a corresponding function and generate the data and compare it visually with `cdata2`. Do you believe that the model adequately describes the data? (**1p**)

4. The dataset `cdata3` from the file `gmm-challenges.Rdata` contains clusters with non-symmetric shapes and thus we need the full-blown Gaussian mixture model to adequately capture the data. That is, the likelihood that $\boldsymbol{x_i}$ is garaged by the cluster j can be expressed as

$$\mathrm{p}[\boldsymbol{x_i}|z_i = j, \boldsymbol{\Theta}] = \frac{1}{2\pi\,|\Sigma_j|} \cdot \exp\left(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right)$$

where the correlation matrix Σ fixes the shape and scale of the cluster.

(a) Implement the two-step minimisation procedure. First, implement the step that finds optimal values for $\boldsymbol{z}$ given fixed $\boldsymbol{\Theta}$. Second, implement the step that finds optimal $\boldsymbol{\Theta}$ values given fixed $\boldsymbol{z}$. Load the data `cdata3` from the file `gmm-challenges.Rdata` and initialise three cluster centres by choosing $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3$ randomly form the range $[-4, 4] \times [-4, 4]$ and set all covariance matrixes to identity matrices. Repeat minimisation steps 100 times and visualise the end result by colouring data points into three different colours and showing the final locations of the cluster centres. (**1p**)

Hint: Take the update formulas for cluster centres and shape parameters from the lecture slides.

Bonus: You get an extra point if you derive these formulae by finding closed form solution for parameters that minimises the log-likelihood of the data $F(\boldsymbol{\Theta}, \boldsymbol{z}) = -\log \Pr[\boldsymbol{x_1}, \ldots, \boldsymbol{x}_n | \boldsymbol{z}, \boldsymbol{\Theta}]$.

(b) Now that you know a good estimate for the labels and cluster parameters, you can find the mixture probabilities $\lambda_j = \Pr[z = j]$ as fractions of corresponding cluster labels. Now that you have fixed all model parameters, you can test the applicability of the model by generating the same amount of data. Write a corresponding function and generate the data and compare it visually with `cdata3`. Do you believe that the model adequately describes the data? (**1p**)

5. Let us consider the following simplified path reconstruction task. A user uses a finger to draw a shape on the tablet screen, which registers the centre of the finger at equal time intervals. Each measurement has a spherical Gaussian noise with constant variance due to alteration of the

contact area. Find out the most probable shape candidate. For simplicity, consider only piecewise linear paths specified by angle points $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ where the distance between adjacent ankle points is the same. Assume that each measurement is by some ankle point $\boldsymbol{\mu}_j$. Note that many points can be generated by the same ankle point.

(a) Derive the corresponding probabilistic model and express the log-likelihood function $F(\boldsymbol{\Theta}, \boldsymbol{z}) = \Pr[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n | \boldsymbol{z}, \boldsymbol{\Theta}]$. (**1p**)

(b) Define a minimisation step that finds the optimal assignment $\boldsymbol{z}$ of individual measurements to ankle points. (**1p**)

(c) Find a heuristic minimisation algorithm that alters the locations of $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$ so that $F(\boldsymbol{\Theta}, \boldsymbol{z})$ decreases while preserving the restriction $\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_{i+1}\| = d$. For instance, you might start to update locations from random ends of the path or a random point on the path. (**1p**)

(d) Write the corresponding two-step minimisation procedure and run it on the `finger-tracking.Rdata` data. As a initial layout you can take some sub-sample of data points. Try different d values and show the best reconstructed shape. (**2p**)

6. Hard-clustering algorithm can be used also in the image processing. We can use it for splitting the colour space of the image into separate clusters. This very useful in robotics, where you can simplify image processing by replacing colours with cluster labels.

(a) Use the package `jpeg` to read in the picture of a beautiful forest scenery (`forest.jpg`). Treat each pixel as a tree dimensional vector and use hard-clustering algorithm to plot the colour space into chunks that are characteristic for objects in the image. (**1p**)

(b) Visualise the result by showing how the entire colorspace is split into different clusters and how individual pixels are divided among individual clusters. For that, you can use dedicated colours for each cluster. (**2p**). You get an extra point if you keep the original intensity of the pixels and mix the cluster labels through the hue channel.

7. Colorspace clustering is not optimal for image segmentation, since it completely neglects the texture. We can use hard-clustering algorithm based on the Gaussian mixture model also for separating textures. For that we can first consider monochrome 16×16 or 32×32 squares. Flatten them into vectors and then apply hard-clustering.

(a) Split the image into sub-images and keep only the intensity channel. Sample enough squares from random locations or sample all possible squares. Choose one option and justify your choice.(**1p**)

(b) After you have extracted the squares you need to flatten them into vectors and apply hard-clustering algorithm. Choose some reasonable

initialisation mechanism. One possibility is to assign clusters to sub-images based on colour space clustering. You can use majority voting to decide the cluster or cluster the histograms of color-space labels. Document and justify your choice.(**1p**)

(c) Recall that fitting a multivariate normal distribution is equivalent to principal component analysis (See Exercise Session X). Hence, you can use the covariance matrix Σ to determine the principal components through the eigenvalue decomposition. Visualise the corresponding components. Alternatively, you can apply PCA directly to the sub-images assigned to the cluster. (**1p**)

(d) Observe the corresponding principal components for each cluster and based on that determine what is the optimal number of clusters. Justify your reasoning. (**1p**)

8. Another more elaborate way to use colour information is by considering all channels of the sub-images. This quickly leads to high-dimensional clustering task, for which hard-clustering algorithm is not optimal. Describe and implement methods for dimensionality reduction which do not lose too much information.

   (a) One way to reduce the search space is to consider only first 15 principal components of the cluster. For each data point, you can remove the contribution of other components and estimate the probability that the data point is generated for the reduced normal distribution. Study the Wikipedia and corresponding probability formulae and implement the corresponding cluster assignment scheme. (**3p**)

   (b) Modify the cluster assignment algorithm so that the effect of noise is also taken into account. For that assume that the noise component is white gaussian noise that is orthogonal with first 15 principal components. Device a procedure for estimating the corresponding variance σ. Visualise the image segmentation into the clusters and corresponding principal components in the clusters. (**3p**)