

EECE5644 Fall 2018 – Homework 4

This assignment is due on Blackboard by 10:00am ET on Tuesday, November 13, 2018. Please submit your solutions on Blackboard in a single PDF file that includes all math, visual and quantitative results (plots, tables, etc), as well as your code (appended after your answers/solutions for each question). In this PDF file, start your answer for each question on a new page and clearly indicate on this new page which question is being answered. Clearly label axes of plots, table rows/columns, and use descriptive captions for results displayed in this fashion. It is recommended that you use \LaTeX to prepare this PDF document containing your answers. If you have not used \LaTeX before, you can start quickly by opening an account for the online latex editor service by Overleaf, for instance. If using Matlab for your code, please use built-in functions whenever possible. Many useful functions were explained in the ML online tutorial.

1 Order Selection for Gaussian Mixture Model (GMM) Fitting (30 points)

Let $g(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ indicate the probability density function (pdf) of a Gaussian random vector with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Write code that implements the following:

- Generate N samples from $p(\mathbf{x}; \boldsymbol{\theta}) = \beta_1 g(\mathbf{x}; \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}) + \beta_2 g(\mathbf{x}; \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}) + \beta_3 g(\mathbf{x}; \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix})$.
- Fit a mixture of M Gaussian pdfs to the data generated in the previous step using the EM algorithm to obtain $\hat{\boldsymbol{\theta}}_M$. Initialize the EM algorithm as you see appropriate but do not use the true model parameters.
- Evaluate the Bayesian Information Criterion (BIC) for the GMM with $\hat{\boldsymbol{\theta}}_M$. Note that there are fewer than $7M$ independent parameters for order- M 2-dimensional GMM. Explain how you decide on the number of parameters that you use in the model order penalty term of BIC.

Repeat the following for $R = 1000$ Monte Carlo runs:

- Generate $N = 100$ samples from $p(\mathbf{x}; \boldsymbol{\theta})$ for each of the following mixture coefficients: $\boldsymbol{\beta} = \begin{pmatrix} 0.75 \\ 0.20 \\ 0.05 \end{pmatrix}$, $\boldsymbol{\beta} = \begin{pmatrix} 0.80 \\ 0.15 \\ 0.05 \end{pmatrix}$, $\boldsymbol{\beta} = \begin{pmatrix} 0.90 \\ 0.05 \\ 0.05 \end{pmatrix}$.
- For all three datasets (obtained using different $\boldsymbol{\beta}$ values), and for all $M \in \{1, \dots, 5\}$, determine $\hat{\boldsymbol{\theta}}_M$ and compute $\text{BIC}(\hat{\boldsymbol{\theta}}_M)$.

Prepare three separate plots, one for each value of $\boldsymbol{\beta}$, demonstrating the values of median BIC values for different M , as well as 5% and 95% levels of the BIC values obtained in R Monte Carlo runs for each M , in order to indicate 90% confidence intervals around these median values. Also, prepare three additional separate bar plots, one for each value of $\boldsymbol{\beta}$, in which you show the histogram of M values at which BIC is minimized (i.e. the number of Gaussians that would have been selected for the GMM by BIC). Indicate how you initialized each algorithm. Briefly discuss results.

2 Clustering Using GMMs and K-Means Algorithm (30 points)

Let $g(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ indicate the probability density function (pdf) of a Gaussian random vector with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Write code that implements the following:

- Generate N samples from $p(\mathbf{x}; \boldsymbol{\theta}) = \beta_1 g(\mathbf{x}; \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}) + \beta_2 g(\mathbf{x}; \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix}) + \beta_3 g(\mathbf{x}; \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 0.5 & 0 \\ 0 & 1 \end{pmatrix})$.
- Fit a mixture of $M = 3$ Gaussian pdfs to the data generated in the previous step using the EM algorithm to obtain $\hat{\boldsymbol{\theta}}_M$. Initialize the EM algorithm as you see appropriate but do not use the true model parameters.
- Using the components of the estimated GMM, assign cluster labels to samples, such that the index of the Gaussian component with the largest posterior is assigned to each sample (in the textbook, see Chapter 10, equation 27, on page 526 for the calculation of the posterior probabilities).
- Using the K-Means clustering algorithm with $K = 3$, and employing Euclidean distance between centroids and samples to make label assignment updates during iterations, assign cluster labels to each sample. Initialize the centroids as you see appropriate but do not use the true model parameters.

Run the code you developed to generate $N = 100$ samples from a GMM with $\boldsymbol{\beta} = \begin{pmatrix} 0.4 \\ 0.3 \\ 0.3 \end{pmatrix}$ and cluster these samples using both GMM and K-Means procedures. Present your results in the form of three separate plots that show the generated data set as a 2-dimensional scatter plot (1) with all samples having the same marker shape/color; (2) with the marker shape/color for each sample indicating the assigned cluster label for the GMM procedure; (3) with the marker shape/color for each sample indicating the assigned cluster label for the K-Means algorithm. Indicate how you initialized each algorithm. Briefly discuss results.

3 Training an SVM Classifier (40 points)

Let $g(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ indicate the probability density function (pdf) of a Gaussian random vector with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Write code that implements the following:

- Generate $N_{train} = 100$ samples from $p(\mathbf{x}; \boldsymbol{\theta}) = \beta_1 g(\mathbf{x}; \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}) + \beta_2 g(\mathbf{x}; \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}) + \beta_3 g(\mathbf{x}; \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix})$ with $\boldsymbol{\beta} = \begin{pmatrix} 0.5 \\ 0.25 \\ 0.25 \end{pmatrix}$. Label the samples that originate from Component 1 (with mean at the origin) as coming from class +1, and the remaining samples as coming from class -1.
- For a support vector machine classifier that uses spherically symmetric Gaussian kernels with width parameter σ , use K-fold cross validation to determine the best kernel parameter σ and the overlap penalty term weight parameter (referred to as C in class), in order to achieve minimum empirical probability of error estimate as the cross-validation objective.
- Using the best hyper parameters σ and C , train a final SVM on the entire data set.
- Generate $N_{test} = 10000$ samples from the same GMM that generated the training set, and assign labels in the same way. This large test data set is independent from and is identically distributed to our training set that was used to train the SVM classifier. We can use it to assess test (generalization) performance.

Apply the trained SVM to classify the test samples and report the following results:

- (1) empirical probability of error estimate on the test set;
- (2) a single plot that shows the following:
 - (a) all test data, each sample colored *green* if correctly classified xor *red* if incorrectly classified by the SVM;
 - (b) the decision boundary of the SVM overlaid on the data.