As usual, please hand in on paper form your derivations and answers to the questions. You can use any programming language for your source code (submitted on Studium as per the website instructions). All the requested figures should be printed on paper with clear titles that indicate what the figures represent.

Linear classification

Please download the datasets from the website. You get 6 text files: three training sets (`.train` files) and three test sets (`.test` files). Each row of the text file represents a sample of data (x_i, y_i). There are three columns: the first two give the coordinates for $x_i \in \mathbb{R}^2$; the third column gives the class label $y_i \in \{0, 1\}$. There are three different types of datasets (`A`, `B` and `C`), all generated from some kind of mixture of Gaussians generative model. The train and test sets are generated from the same distribution for each types of dataset. To help your interpretation, we give you the actual generating process:

- Dataset `A`: the class-conditionals for this dataset are Gaussians with different means, but with a *shared* covariance matrix Σ.
- Dataset `B`: similar generating process but the covariance matrices are *different* for the two classes.
- Dataset `C`: here one class is a mixture of two Gaussians, while the other class is a single Gaussian (with no sharing).

Note that normally we would not know the information about the generating process. In this assignment, we will compare difference classification approaches.

1. **Generative model (Fisher LDA)**. We first consider the Fisher LDA model as seen in class: given the class variable, the data are assumed to be Gaussians with different means for different classes but with the same covariance matrix:

$$Y \sim \text{Bernoulli}(\pi), \quad X \,|\, Y = j \sim \mathcal{N}(\mu_j, \Sigma).$$

   (a) Derive the form of the maximum likelihood estimator for this model. *Hint*: you can re-use some of the tricks presented in class for the MLE of a multivariate Gaussian, but adapted to this setting. You can get inspiration from Section 7.2 in Mike's book (which covers the case where Σ is diagonal).

   (b) What is the form of the conditional distribution $p(y = 1|x)$? Compare with the form of logistic regression.

   (c) Implement the MLE for this model and apply it to each training dataset. For each dataset, represent graphically the data as a point cloud in $\mathbb{R}^2$ and the line defined by the equation

$$p(y = 1|x) = 0.5\,.$$

2. **Logistic regression:** now implement logistic regression for an affine function $f(x) = w^\top x + b$ (do not forget the constant term – you can use the bias feature trick) using the IRLS algorithm (Newton's method) which was described in class. Hint: never compute the matrix inverse by itself – this is not numerically stable when the Hessian might become ill-conditioned...

   (a) Give the numerical values of the learnt parameters for each training dataset.

   (b) Represent graphically the data as a point cloud in $\mathbb{R}^2$ and the line defined by the equation:

   $$p(y = 1|x) = 0.5\,.$$

3. **Linear regression:** as mentioned in class, we can forget that the class y can only take the two values 0 or 1 and think of it as a real-valued variable on which we can do standard linear regression (least-squares). Here, the Gaussian noise model on y does not make any sense from a generative point of view; but we can still do least-squares to estimate the parameters of a linear decision boundary (you'll be surprised by its performance despite coming from a "bad" generative model!). Implement linear regression (for an affine function $f(x) = w^\top x + b$) by solving the normal equations on each dataset (with no regularization).

   (a) Provide the numerical values of the learnt parameters.

   (b) Represent graphically the data as a point cloud in $\mathbb{R}^2$ and the line defined by the equation

   $$f(x) = 0.5\,.$$

4. Data in the files `classificationA.test`, `classificationB.test` and `classificationC.test` are respectively drawn from the same distribution as the data in the files `classificationA.train`, `classificationB.train` and `classificationC.train`. Test the different models learnt from the corresponding training data on these test data.

   (a) Compute for each model the misclassification error (i.e. the fraction of the data misclassified) on the training data and compute it as well on the test data.

   (b) Compare the performances of the different methods on the three datasets. Is the misclassiffication error larger, smaller, or similar on the training and test data? Why? Which methods yield very similar/dissimilar results ? Which methods yield the best results on the different datasets ? Provide an interpretation.

5. **QDA model**. We finally relax the assumption that the covariance matrices for the two classes are the same. So, given the class label, the data are now assumed to be Gaussian with means and covariance matrices which are a priori different:

   $$Y \sim \text{Bernoulli}(\pi), \quad X \,|\, Y = j \;\sim \mathcal{N}(\mu_j, \Sigma_j).$$

   Implement the maximum likelihood estimator and apply it to the data.

   (a) Provide the numerical values of the learnt parameters.

   (b) Represent graphically the data as well as the conic defined by

   $$p(y = 1|x) = 0.5\,.$$

   (c) Compute the misclassification error for QDA for both train and test data.

   (d) Comment the results as previously.