CDA Spring 2026

Sneha Das

February 25, 2026

Week 4: Classification and regularized classification

1. In this exercise, you will implement Fisher's Linear Discriminant Analysis (LDA) from scratch to classify the famous Iris dataset. Instead of relying on a pre-built machine learning library to fit the model, you will manually compute the plug-in estimates and the discriminant functions based on the theoretical formulas (dataset: `FisherIris.csv`)

   (a) **Data Preparation**
   Load the Fisher Iris dataset. The skeleton code already includes the logic to read the CSV, extract the feature matrix X (sepal length, sepal width, petal length, petal width), and encode the target labels y (Setosa, Versicolour, Virginica).

   (b) **Calculate Plug-in Estimates**
   To build the LDA model, compute the sample estimates for the model parameters. For each of the K classes (where $K = 3$), calculate:

   - **Prior Probabilities (π_k):** The proportion of observations belonging to class k.
   - **Class Means (μ_k):** The mean feature vector for observations in class k.
   - **Pooled Covariance Matrix (Σ):** Calculate the shared empirical covariance matrix across all classes.

   (c) **Implement the Discriminant Function**
   Using your estimated parameters, calculate the discriminant score $\delta_k(x)$ for every observation across all 3 classes. Use the following discriminant function formula:

   $$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$$

   Hint: You will need to write a function (e.g., `ProduceDiscriminantLine`*) that computes this score using the inverse of your covariance matrix Σ.*

   (d) **Predict Class Belongings**
   Classify every observation in the training data. An observation x_i should be assigned to the class k that yields the highest discriminant score $\delta_k(x_i)$. Store your predictions in a variable called `yhat`.

   (e) **Evaluate the Model**
   Calculate and visualize the performance of your LDA model on the training data.

   - Compute the non-normalized confusion matrix comparing the true labels y against your predicted labels `yhat`.

- Use `sklearn.metrics.ConfusionMatrixDisplay` to plot the matrix, ensuring that the display labels are set to `['Setosa', 'Versicolour', 'Virginica']`.

2. Logistic regression: Given a logistic model for lung cancer (yes/no) as a function of smoking (number of cigarettes per day) with $\beta = 0.02$. Show that one unit increase in smoking means an increase in lung cancer risk (odds-ratio) of $\exp(0.02) = 1.02 = 2\%$.

3. We have a data material (Golub et al 1999) with gene expression levels from 72 patients with two forms of leukemia, acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). Gene expression levels (how actively the cells are using the information in different genes) are measured for 7127 genes. We would like to build a biomarker for classification of the two cancer forms. Ideally, we would like to use only a few variables.

   (a) How can you use logistic regression here?

   (b) Build a classifier for training data in `GolubGXtrain.csv`. What regularization method do you prefer if you want to have few genes in the biomarker?

   (c) How many variables do you end up with?

   (d) Use the obtained model to calculate accuracy on the test data.

4. Implement and calculate a Regularized Discriminant Analysis (RDA) for the Silhouette data in `Silhouettes.mat`.

   (a) What happens when we vary γ in RDA?