Project 2: Linear Classification

Due: March 19, 2019

EE/CS5841: Machine Learning

Project Summary

In this assignment you will apply two different linear classifiers (linear discriminant analysis (LDA) and logistic regression (LR)) to synthetic data. You may also implement a quadratic discriminant analysis (QDA)-based classifier for +3 points extra credit.

1 Part 1: Generate Training and Testing Data (2 points)

Generate training and testing data to use in the classification problems that follow.

- Generate 200 samples from a normal distribution with mean $\mu_{(+1)} = (1,1)^T$ and covariance $\Sigma_{(+1)} = \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix}$. These data belong to the +1 class.
- Generate 200 samples from a normal distribution with mean $\mu_{(-1)} = (-1,-1)^T$ and covariance $\Sigma_{(-1)} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$. These data belong to the -1 class.
- Partition your data into two groups: training data and testing data. Each group should have 100 samples from each class.

  Plot each group of data in a scatter plot.

1.1 Data Generation Deliverables

Plot the training data and testing data independently via 2 scatter plots. Label the plots as "Training Data" and "Testing Data", respectively. Use different marker types and/or colors for each class, e.g., red x's for the +1 class and blue o's for the -1 class.

2 Part 2: Linear Discriminant Analysis (5 points)

Apply a LDA classifier to the data generated in Part 1. Note that a LDA classifier does not need to be "trained" in the typical sense—the "training" is done by knowing the class means and covariance.

- Estimate the overall covariance of the data using the *testing* set. Note that the data must be centered with respect to the overall mean as outlined below.

  First estimate the overall mean as $\boldsymbol{\mu}_{est} = \frac{1}{200}\sum_{i=1}^{200} \mathbf{x}_i$.

  Then estimate the overall covariance as $\Sigma_{est} = \frac{1}{200-1}(X - \boldsymbol{\mu}_{est})^T(X - \boldsymbol{\mu}_{est})$, where the rows of X are training instances and the matrix $(X - \boldsymbol{\mu}_{est})$ comprises the centered data. Note that X should be 200×2 and Σ_{est} will be 2×2.

  Report the estimated covariance matrix.

- Implement a LDA classifier using the class means ($\mu_{(+1)}$ and $\mu_{(-1)}$) and the overall covariance estimate (Σ_{est}). You may **not** use built-in functions related to LDA.

  Report the learned weight (w) and bias (b).

- Classify the testing data.

  Report the classification accuracy.

  Plot the testing data, highlighting the misclassifications.

2.1 LDA Deliverables

Report the estimated covariance matrix Σ_{est} defined in the problem statement as well as the learned hyperplane parameters, $\mathbf{w}$ and b. Also report the LDA classifier's classification accuracy defined as $Acc = \frac{\text{\# samples correctly classified}}{200} \times 100\%$ (note the denominator is 200 since the number of testing samples is 200). Show a scatter plot of the testing data, giving classes different marker shapes/colors similar to Part 1. Highlight (e.g., draw a small circle around, change the color, change the marker type, etc.) the instances that were misclassified by the LDA classifier.

3 Part 3: Logistic Regression (10 points)

Apply a logistic regression classifier to the data generated in Part 1. You will use the training set to train the logistic regression model and will evaluate its performance with the testing set.

- Train a logistic regression model on the training data. You may **not** use built-in functions related to logistic regression.

Lump a '1' into each datapoint such that the bias is encoded in the weight vector.

Use the iteratively reweighted least squares algorithm for training.

Report the learned weights.

- Classify the testing data using the learned logistic regression model.

  Report the classification accuracy.

  Plot the testing data, highlighting the misclassifications

3.1 LR Deliverables

Report the learned LR parameters, $\mathbf{w}$. Also report the LR classifier's classification accuracy defined as $Acc = \frac{\text{\# samples correctly classified}}{200} \times 100\%$ (note the denominator is 200 since the number of testing samples is 200). Show a scatter plot of the testing data, giving classes different marker shapes/colors similar to Part 1. Highlight (e.g., draw a small circle around, change the color, change the marker type, etc.) the instances that were misclassified by the LR classifier.

4 Extra Credit: Quadratic Discriminant Analysis (+3 points)

Apply a QDA classifier to the data generated in Part 1. Note that a QDA classifier does not need to be "trained" in the typical sense—the "training" is done by knowing the class means and covariance.

- Implement a QDA classifier using the class means ($\mu_{(+1)}$ and $\mu_{(-1)}$) and the covariance matrices $\Sigma_{(+1)}$ and $\Sigma_{(+1)}$. These are given in the problem statement. You may **not** use built-in functions related to QDA.

- Classify the testing data.

  Report the classification accuracy.

  Plot the testing data, highlighting the misclassifications.

4.1 Extra Credit: QDA Deliverables

Report the QDA classifier's accuracy defined as $Acc = \frac{\text{\# samples correctly classified}}{200} \times 100\%$ (note the denominator is 200 since the number of testing samples is 200). Show a scatter plot of the testing data, giving classes different marker shapes/colors similar to Part 1. Highlight (e.g., draw a small circle around, change the color, change the marker type, etc.) the instances that were misclassified by the QDA classifier.

5 Submissions

Your submission should include a PDF with all deliverables clearly labeled. You must also submit all your code. This assignment may be completed using either Matlab or Python. Jupyter notebooks are acceptable, however, please print to PDF for your submission.