

The aim of this practical work is to implement standard simple generative and discriminative classifiers. The considered generative approach for classification is the Gaussian Discriminant Analysis (LDA and QDA), and the discriminative one consists in the binary logistic regression classifier.

The codes may be written in Matlab/Octave/Scilab, R or Python. Here in the text we assume that Matlab is used. Interesting tutorials on Matlab for new users `matlab1` `matlab2`

1 Linear Discriminant Analysis (LDA)

1. Download the training data and the test data on your Matlab work directory. The labels of the training set are given in the last column of `data_train_2class.mat`
2. Learn and test a LDA classifier first by using the matlab function `classify.m`
3. Show the results (both the density ellipses for each class and the decision boundary); to plot the data you can use the Matlab functions `plot`, `scatter`, `gscatter`. You can also use the `contour` function to show the ellipse densities
4. Implement LDA (e.g., create functions `train_lda.m` and `test_lda.m`) and show the results

2 Quadratic Discriminant Analysis (QDA)

1. Learn and test a QDA classifier first by using the matlab function `classify.m`
2. Show the results (both the density ellipses for each class and the decision boundary)
3. Implement QDA (e.g., create functions `train_qda.m` and `test_qda.m`) and show the results

Now do the same thing for the following three-class problem. The data are available in the following links : `training data` `test data`

3 Binary Logistic Regression

Implement the IRLS algorithm for binary logistic regression model where $\mathbb{P}(y_i = 1 | \mathbf{x}_i; \mathbf{w}) = \frac{\exp(\mathbf{w}^T \mathbf{x}_i)}{1 + \exp(\mathbf{w}^T \mathbf{x}_i)}$.

The IRLS algorithm is given by : $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + (\mathbf{X}^T \mathbf{W}^{(t)} \mathbf{X})^{-1} \mathbf{X}^T (\mathbf{y} - \mathbf{p}^{(t)}) = (\mathbf{X}^T \mathbf{W}^{(t)} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W}^{(t)} \mathbf{y}^*$ where t is the current iteration number and :

\mathbf{X} is the $n \times (d + 1)$ matrix whose rows are the input vectors \mathbf{x}_i

\mathbf{y} is the $n \times 1$ column vector whose elements are the binary labels y_i : $\mathbf{y} = (y_1, \dots, y_n)^T$

\mathbf{p} is the $n \times 1$ column vector of logistic probabilities corresponding to the i th input

$\mathbf{p} = (\pi(\mathbf{x}_1; \mathbf{w}), \dots, \pi(\mathbf{x}_n; \mathbf{w}))^T$.

\mathbf{W} is the $n \times n$ diagonal matrix whose diagonal elements are $\pi(\mathbf{x}_i; \mathbf{w}) (1 - \pi(\mathbf{x}_i; \mathbf{w}))$ for $i = 1, \dots, n$.

$\mathbf{y}^* = \mathbf{X} \mathbf{w}^{(t)} + (\mathbf{W}^{(t)})^{-1} (\mathbf{y} - \mathbf{p}^{(t)})$

For that you can create the functions `train_RegLog.m` and `test_RegLog.m` and then apply them on the previous two-class data set.