

ELE888 Intelligent Systems

Lab 2: Linear Discriminant Functions

Objective

To classify Iris datasets using linear discriminant functions.

Background

Linear discriminant functions are those functions which are either linear in the components of feature vector x or linear in some given set of functions of x . Linear classifiers can be easily built using discriminant functions and does not need the knowledge on the underlying probability densities of the given data. Linear classifiers are attractive due to their simplicity and ideal for initial, trial classifiers. The classifier parameters can be computed using a set of training samples and minimizing a criterion function.

A discriminant function $g(\mathbf{x})$ that is linear in the component of the feature vector \mathbf{x} can be written as:

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 \quad (1)$$

where \mathbf{w} is the weight vector and w_0 is the bias or threshold. In general for a two category class problem we decide on ω_1 (class 1) by observing the feature vector \mathbf{x} if $g(\mathbf{x}) > 0$ and ω_2 (class 2) if $g(\mathbf{x}) < 0$.

A generalized form of linear discriminant functions can be given as:

$$g(\mathbf{x}) = \mathbf{a}^T \mathbf{y} \quad (2)$$

where the *augmented feature vector* y and *augmented weight vector* a are given by:

$$\mathbf{y} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_d \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ \vdots \\ w_d \end{bmatrix} \quad (3)$$

The goal is to compute the weight vector using training data samples. Gradient descent procedures are commonly used to achieve this by iteratively updating the weight vector while minimizing some criterion function. The following pseudo code outlines the basic gradient descent algorithm:

1. begin initialise: \mathbf{a} , criterion θ , $\eta(\cdot)$, $k = 0$
2. do $k \leftarrow k + 1$
3. $\mathbf{a} \leftarrow \mathbf{a} - \eta(k) \nabla J(\mathbf{a})$
4. until $|\eta(k) \nabla J(\mathbf{a})| < \theta$
5. return \mathbf{a}
6. end

Laboratory Exercises

Let's first consider the following test cases from the Iris dataset investigated in Lab 1:

- **data set A:** a 50×2 matrix (50 rows, 2 columns) of the samples from the *Iris Setosa* class, with columns as the features x_2 and x_3 respectively.
- **data set B:** a 50×2 matrix of the samples from the *Iris Versicolour* class, with columns as the features x_2 and x_3 respectively.
- **data set C:** a 50×2 matrix (50 rows, 2 columns) of the samples from the *Iris Virginia* class, with columns as the features x_2 and x_3 respectively.

Using the gradient descent approach and the *perceptron criterion* compute the weight vectors and classify the given data for the conditions given below. The perceptron criterion function $J_p(\mathbf{a})$ and its gradient ∇J_p are given as:

$$J_p(\mathbf{a}) = \sum_{\mathbf{y} \in Y} (-\mathbf{a}^T \cdot \mathbf{y}) \quad (4)$$

$$\nabla J_p = \sum_{\mathbf{y} \in Y} (-\mathbf{y}) \quad (5)$$

where $Y(\mathbf{a})$ are the set of samples misclassified by \mathbf{a} .

1. Construct a new data set from A and B by setting aside 30% of the samples from these original sets (this will be used for *training* purposes). Construct a second set from the remaining 70% of data (this will be used for *testing*). Use the 30% set (training samples) to compute the weight vector for $\eta(k) = \eta(\cdot) = 0.01$, $\theta = 0$, and an initial value of $\mathbf{a}(0) = [0 \ 0 \ 1]$. Limit your maximum iterations to 300 (i.e even if you cannot achieve $\theta = 0$).
2. Use the 70% set (test samples) and the weight vector computed in the previous step to calculate the classification accuracy of the classifier.
3. Repeat the above steps by changing the data split to 70%(training) and 30%(testing).
4. Repeat all the above for test and training sets constructed from the combination of data sets B and C (i.e. *Iris Vericolour* vs. *Iris Virginia*)
5. Compute the weight vectors and study the gradient descent algorithms for two different values of $\eta(k)$, and the initial values of \mathbf{a} .
6. In each of the above steps, plot **the training data on** the feature space, the decision boundary for training set, the perception criterion function over the iterations and the number of iterations required for convergence.
7. Discuss in your report the following: (i) the effect of using different sizes for training and testing data, (ii) the effect of different learning rates, threshold, initial weight values, (iii) the criterion function over the iterations, and (iv) comment on the achieved classification accuracy for the given data.

What to Submit?

- The programs should be demonstrated to the TA. A report should include all the required plots, discussions, and your final observations and conclusions.