MODULE 12
Linear Discriminant Function

LESSON 27

Learning the Discriminant Function

Keywords: Perceptron, Convergence, Updation of Weights, Multi-Class

Learning the Linear Discriminant Functions

- **Learning the Discriminant Function**
  It is convenient to use the normalized data for automatically learning the decision boundary when the classes are linearly separable. In general, it is possible to obtain the decision boundary in the form of a hyperplane if we learn the vector w. It is simpler to learn the weight vector w when the classes are linearly separable. We describe an algorithm, called **Perceptron learning algorithm**, for learning the weight vector when the classes are linearly separable.

- **Learning the weight vector**
  This algorithm considers patterns sequentially and updates the weight vector w if a pattern is misclassified by the weight vector. It iterates through the set of patterns till there is no updation, or equivalently no pattern is misclassified during an entire iteration. It is convenient to consider patterns in the normalized form while applying this algorithm. This would mean that a pattern X is misclassified if $w^t X \leq 0$. In order to use the homogeneous form, we need to consider patterns after transformation and normalization. For example, the set of 3-dimensional vectors corresponding to the patterns given in lesson 26 is given in Table 1. The problem is to find a 3-dimensional weight vector w which classifies all the patterns correctly. Equivalently, w should be such that $w^t X$ is greater than 0 for all the patterns; for example for all the patterns in Table 1.

Pattern No.	$feature_1$	$feature_2$	$feature_3$
1	- 1.0	- 1.5	- 1.0
2	- 1.5	- 2.0	- 1.0
3	- 1.5	- 2.5	- 1.0
4	- 2.0	- 2.5	-1.0
5	3.0	1.0	1.0
6	3.5	2.0	1.0
7	4.0	2.0	1.0

Table 1: Description of the normalized patterns

- **Perceptron Learning Algorithm**
  Now we describe an algorithm, called perceptron learning algorithm, for learning w. Let the normalized patterns in the $d+1$-dimensional space be X_1, X_2, $\cdots$, X_n.

  1. Initialize w by choosing a value for w_1 (for simplicity it is adequate to choose $w_1 = 0$).
  2. For j = 1 to n do
     if $w_i^t X_j \leq 0$ (that is when the normalized pattern X_j is misclassified) then i = i+1 and $w_i = w_{i-1} + X_j$ (update the current w vector to classify X_j better).
  3. Repeat step 2 till the value of i has not changed during the entire iteration (that is the current w vector classifies all the training patterns correctly).

- **Updation of Weight Vector**
  It is possible to explain the intuition behind the proposed scheme for updating the weight vector as follows. Let w_i be the current weight vector which has misclassified the pattern X_j; that is $w_i^t X_j < 0$. So, using the above algorithm, we have

  $$w_{i+1} = w_i + X_j$$

  Note that

  $$w_{i+1}^t X_j = w_i^t X_j + \| X_j \|^2$$

  As a consequence, $w_{i+1}^t X_j$ is larger than $w_i^t X_j$ by $\| X_j \|^2$ because $\| X_j \|^2$ is positive. This would mean that the updated vector w_{i+1} is better suited, than w_i, to classify X_j correctly because $w_{i+1} X_j$ is larger than $w_i X_j$ and so can be positive even if $w_i X_j$ were not.

  Example 1

  We can illustrate the working of the algorithm using the data in Table 1 with the help of the following steps.

  1. $w_1 = (0,0,0)^t$ *and* $X_1 = (-1.0, -1.5, -1.0)^t$. Here, $w_1^t X_1 = 0$ and so $w_2 = w_1 + X_1$ which is represented by

     $$w_2 = (-1.0, -1.5, -1.0)^t.$$

2. Next we consider pattern X_2 and $w_2^t X_2$ is 5.0 (> 0). Similarly, X_3, and X_4 also are properly classified.
   Note that $w_2^t X_3 = 6.25$, and $w_2^t X_4 = 6.75$. So, these patterns do not affect the weight vector.
3. However, $w_2^t X_5$ = -5.5 (< 0). So, $w_3 = w_2 + X_5$, that is
$$w_3 \;=\; (2.0, -0.5, 0.0)^t.$$
   Note that w_3 classifies patterns X_6, and X_7 correctly.
4. However x_1 is misclassified by w_3. Note that $w_3^t X_1 \; is \; - \; 1.25 (< 0)$. So, $w_4 = w_3 + X_1$ is obtained. It is given by
$$w_4 \;=\; (1.0, -2.0, -1.0)^t.$$
   w_4 correctly classifies X_2, X_3, and X_4.
5. However, w_4 misclassifies X_5 as $w_4^t X_5 \;=\; 0.0 \; (\leq 0.0)$. So, $w_5 = w_4 + X_5$, that is
$$w_5 \;=\; (4.0, -1.0, 0.0)^t.$$
   w_5 classifies X_6 and X_7 correctly.
6. Note that w_5 misclassifies X_1 as $w_5^t X_1 \;=\; - \; 2.5 \; < \; 0$. So, $w_6 = w_5 + X_1$, that is
$$w_6 \;=\; (3, -2.5, -1)^t.$$
   w_6 classifies X_2, X_3, X_4, X_5, X_6, X_7, and X_1. Specifically, $w_6^t X_2 \;=\; 3$, $w_6^t X_3 \;=\; 2.75$, $w_6^t X_4 \;=\; 1.25$, $w_6^t X_5 \;=\; 5.5$, $w_6^t X_6 \;=\; 4.5$, $w_6^t X_7 \;=\; 6$, and $w_6^t X_1 \;=\; 1.75$.
7. So, the algorithm converges to w_6 which is the desired vector. In other words, $3X_1$ - $2.5X_2$ - 1 = 0 is the equation of the decision boundary; equivalently, the line separating the two classes is $6X_1$ - $5X_2$ - 2 = 0

- **Convergence of the Perceptron Algorithm**
  In general, it is possible to show that the perceptron learning algorithm will converge to a correct weight vector in a finite number of iterations when the classes are linearly separable. The number of iterations may increase based on the location of the training patterns.

- **Updation of the w vector**
  Let us consider the following. Let the training set of patterns, from two classes, after transformation and normalization be $\{X_1, X_2, \cdots, X_n\}$ and let $w_1 = 0$. Let us say that the first pattern, if any, misclassified by w_1 be X^1 which means that $w_1^t X^1 \leq 0$; further $X^1 \in \{X_1, X_2, \cdots, X_n\}$. Now we get $w_2 = w_1 + X^1$. Let X^2 be the pattern misclassified by w_2; update w_2. In this manner let X^k be misclassified by w_k. Note that $w_{k+1} = w_k + X^k$.

- **Linear Separability**
  If the classes are linearly separable, then there exists a weight vector w such that $w^t X > 0$ for every training pattern X, irrespective of its class label, because of normalization. Now consider w_{k+1} which is given by

  $$w_{k+1} = w_1 + X^1 + X^2 + \cdots + X^k \tag{1}$$

  Now

  $$w^t w_{k+1} = w^t(w_1 + X^1 + X^2 + \cdots + X^k) > k\delta \tag{2}$$

  because $w_1 = 0$ and where $\delta = min\ \{w^t X_i\}$ over all X_i in the training set. Further,

  $$w_{k+1}^t w_{k+1} = (w_k + X^k)^t (w_k + X^k) = || w_k ||^2 + || X^k ||^2 + 2 w_k^t X^k$$

  Note that $w_k^t X^k \leq 0$ and if $\gamma = max\ \{|| X^k ||^2\}$, then

  $$|| w_{k+1} ||^2 < k\gamma \tag{3}$$

  This is obtained by expanding recursively w_k in terms of w_{k-1} and X^{k-1}.

- **Cosine of the angle between w and w_{k+1}**
  Let θ be the angle between w and w_{k+1}. So, using (3) and (4), we have

  $$cos\ \theta = \frac{w^t w_{k+1}}{|| w_{k+1} ||} > \frac{k\delta}{\sqrt{k\gamma}}$$

But we know that $cos\,\theta\ <\ 1$. So, we have

$$\frac{\sqrt{k}\delta}{\sqrt{\gamma}}\ <\ 1\ \Rightarrow k\ <\ \frac{\gamma}{\delta^2} \tag{4}$$

Note that k, the number of iterations is bounded because γ is finite for finite size training vectors. Also, δ cannot be zero as the classes are linearly separable and $w^tX_i\ >\ 0$. So, in a finite number of iterations the algorithm converges. It is possible that k is very large if δ is close to zero; this can happen if one of the X_is is almost orthogonal to w.

- **Multi-class Problems**
  A linear discriminant function is ideally suited to separate patterns belonging to two classes that are linearly separable. However, in real life applications, there could be a need for classifying patterns from **three** or more classes. It is possible to extend a binary classifier, in different ways, to classify patterns belonging to C classes where $C >$ 2, We list two of the popularly used schemes below:

  1. Consider a pair of classes at a time. Note that there are $\frac{C(C-1)}{2}$ such pairs when there are C classes in the given collection. Learn a linear discriminant function for each pair of classes. Combine these decisions to arrive at the class label.

  2. Consider two-class problems of the following type. For each class C_i, create the class $\overline{C}_i$ which consists of patterns from all the remaining classes. So, $\overline{C}_i = \bigcup_{j=1;j\neq i}^{C} C_j$. Learn a linear discriminant function to classify each of these two-class problems. Note that there are C such two-class problems. These C linear discriminants give us the overall decision.

- **Assignment**

  1. Consider a two-class problem. There are **five** patterns from one class (labeled 'X'') and **four** patterns from the second class (labeled 'O''). This set of labeled patterns is described in the following table.

     (a) Show that the data is linearly separable.

     (b) Show that $f_1\ =\ f_2$ is a decision boundary.

Pattern No.	f_1	f_2	Class
1	0.5	3.0	'X'
2	1	3	'X'
3	0.5	2.5	'X'
4	1	2.5	'X'
5	1.5	2.5	'X'
6	4.5	1	'O'
7	5	1	'O'
8	4.5	0.5	'O'
9	5.5	0.5	'O'

(c) Obtain the weight vector w. Show that it is orthogonal to the decision boundary.

(d) Suggest another decision boundary for the data.

(e) Compute the distance from the origin ($X = (0,0)^t$) to the decision boundary.

2. By transforming and normalizing the data in the table above, we get the data in the following table. Use perceptron algorithm to learn the weight vector.

Pattern	f_1	f_2	f_3
x_1	-0.5	-3.0	-1
x_2	-1	-3	-1
x_3	-0.5	-2.5	-1
x_4	-1	-2.5	-1
x_5	-1.5	-2.5	-1
x_6	4.5	1	1
x_7	5	1	1
x_8	4.5	0.5	1
x_9	5.5	0.5	1

3. Let θ be the angle between w and X. Show that the value of $cos\,\theta$ is positive if $w^tX > 0$.

4. Consider the three patterns $(1,1)^t$ and $(2,2)^t$ from class 'X' and $(2,0)^t$ from class 'O'. Use perceptron learning algorithm to show that the decision boundary is $f_1 - 3f_2 - 1 = 0$.

5. Consider the truth table of Boolean OR given in the following table. There are two classes '0' and '1' corresponding to the output

f_1	f_2	$f_1 \vee f_2$
0	0	0
0	1	1
1	0	1
1	1	1

values 0 and 1 of $f_1 \vee f_2$ respectively. There is one pattern of class '0' and three patterns of class '1'.

   (a) Show that the classes are linearly separable.

   (b) After transformation and normalization, use the perceptron learning algorithm to obtain the weight vector.

- **References**

Devi V.S; Murty, M.N (2011) *Pattern Recognition: An Introduction*, Universities Press, Hyderabad.
Duda R.O; Hart, P.E; Stork, D.J. (2000) *Pattern Classification*, Wiley-interscience, New York.
Minsky M.L.; Papert, S.A. (1988) *Perceptrons*, Extended Edition, MIT Press, Cambridge, MA.
Theodoridis S.; Koutroumbas K. (2003) *Pattern Recognition*, Academic Press, New York.