Chapter 4
Linear Support Vector Machines

Abstract Support vector machine (SVM) is the most popular classifier based on a linear discriminant function. It is ideally suited for binary classification. It has been studied extensively in several pattern recognition applications and in data mining. It has become a baseline standard for classification because of excellent software packages that have been developed systematically over the past three decades. In this chapter, we introduce SVM-based classification and some of the essential properties related to classification. Specifically we deal with linear SVM that is ideally suited to deal with linearly separable classes.

Keywords Linear SVM · Perceptron and SVM · Maximum margin · Dual problem · Binary classifier · Multiclass classification

4.1 Introduction

Support vector machine (SVM) [1–5] can be used as a binary classifier based on a *linear discriminant function*. In this sense it resembles the perceptron.

4.1.1 Similarity with Perceptron

1. Both perceptron and SVM can be seen as employing the linear discriminant function of the form $W^t X + b$.
2. In the case of perceptron, if the classes are *linearly separable* then it is possible to get more than one W as shown in Fig. 3.1. In theory, there could be *infinite solutions or W vectors*. In the case of SVM, we constrain the W to be a globally optimal solution of a well-formulated optimization problem. So, W is unique.
3. If there is no linear discriminant in the input space or in the given variables, then it is possible to get a linear discriminant in a high-dimensional space. We have seen that in the case of boolean functions, we can transform any function into a linear form in the space of all possible minterms.

M.N. Murty and R. Raghava, *Support Vector Machines and Perceptrons*,
SpringerBriefs in Computer Science, DOI 10.1007/978-3-319-41063-0_4

4. For example, $xor(x_1, x_2)$ is not linear in (x_1, x_2). However, it is linear in (x_1, x_2, x_1x_2) as examined in the previous chapter.
5. Let $X = (x_1, x_2)^t$ be a two-dimensional vector and let $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^5$ given by $\phi(X) = (1, x_1, x_2, x_1^2, x_2^2, x_1x_2)$. Then
   $g(X) = a_0 + a_1x_1 + a_2x_2 + a_3x_1^2 + a_4x_2^2 + a_5x_1x_2$ is a *quadratic function* in $\mathbb{R}$, the *input space*, and
   $g(\phi(X)) = a_0 + a_1x_1 + a_2x_2 + a_3x_1^2 + a_4x_2^2 + a_5x_1x_2$ is a *linear function* in the 5-dimensional $\phi(X)$ space, called the *feature space*.
6. SVM and perceptron are *linear classifiers*.
7. Both SVM and perceptron are inherently binary classifiers. They can be extended to deal with multiclass classification using similar techniques which we will discuss later.

4.1.2 Differences Between Perceptron and SVM

1. ***W* Vector**:
   Perceptron can converge to different W vectors based on the order in which the training patterns are processed.
   However, SVM will produce the same W.
2. **Optimization**:
   Perceptron criterion function, $J(W)$. has value 0 if all the training patterns are classified correctly by W. In other words, $W^tX_i > 0$ for all i. So, multiple solutions, or W vectors could exist that lead to the same error.
   It is possible to show, on the contrary, that the SVM criterion function will result in the same W vector. Here, the W vector corresponds to the *decision boundary that maximizes separation between the two classes*.

4.1.3 Important Properties of SVM [1–5]

1. Maximizing the separation between classes is based on a *well-behaved optimization* problem. In the linearly separable case, it is possible to obtain the *globally optimal* W.
2. It can learn *nonlinear boundaries in the input space* by mapping from the input space to a high-dimensional *feature space* and learning a linear boundary in the feature space; such a linear boundary corresponds to a nonlinear boundary in the input space.
3. It employs a suitable *similarity function in the input space* and avoids making expensive computations in the high-dimensional feature space.
4. It combines the training data points to obtain W and use the W for classification.

In its simplest form, the SVM can be used to classify patterns belonging to two classes that are linearly separable.

4.2 Linear SVM [1, 5]

Given the training set $\{X_1, X_2, \ldots, X_n\}$, $X_i \in \mathbb{R}^l$, $i = 1, 2, \ldots, n$

Let the two classes be linearly separable. This means there is a $W \in \mathbb{R}^l$ and a $b \in \mathbb{R}$ satisfying

1. $W^t X_i + b > 0 \; \forall i$ with $y_i = 1$
2. $W^t X_j + b < 0 \; \forall j$ with $y_j = -1$
   We can put these two sets of inequalities together to write
3. $y_k(W^t X_k + b) > 0 \; \forall k$ with $1 \leq k \leq n$.
4. Note that the decision boundary is given by $W^t X + b = 0$. There could be infinitely many possible separating hyperplanes unless we constrain the selection.

4.2.1 Linear Separability

We can study the implication of linear separability as follows:

- Let the training set be $\{(X_1, -1), (X_2, -1), \ldots, (X_{n_-}, -1), (X_{n_-+1}, +1), \ldots, (X_n, +1)\}$
- Note that $W^t X_j + b' = -\varepsilon_j$, where $\varepsilon_j > 0$, $\forall j$ with $y_j = -1$
  Similarly, $W^t X_i + b' = \varepsilon_i$, where $\varepsilon_i > 0$, $\forall i$ with $y_i = 1$
- So, we have $W^t X_j + b' \leq -\varepsilon_-$ where $-\varepsilon_- = \max_j -\varepsilon_j$ and
- We have $W^t X_i + b' \geq \varepsilon_+$ where $\varepsilon_+ = \min_i \varepsilon_i$.
- From these two sets of inequalities, we get
  $W^t X_i + b \leq -\varepsilon \; \forall i$ with $y_i = -1$ and
  $W^t X_j + b \geq \varepsilon \; \forall j$ with $y_j = 1$
  where $\varepsilon = \frac{\varepsilon_+ + \varepsilon_-}{2}$ and $b = b' - \frac{\varepsilon_+ - \varepsilon_-}{2}$
- By dividing the two inequalities by ε both sides, we get
  $W_n^t X_i + b_n \leq -1 \; \forall i$ with $y_i = -1$ and
  $W_n^t X_j + b_n \geq 1 \; \forall j$ with $y_j = 1$
  where $W_n = (\frac{w_1}{\varepsilon}, \frac{w_2}{\varepsilon}, \ldots, \frac{w_l}{\varepsilon})^t$ and $b_n = \frac{b}{\varepsilon}$
- Instead of using W_n and b_n, we use W and b, respectively, for the sake of brevity.
  So, we get the following inequalities
  $W^t X_i + b \leq -1 \; \forall X_i$ such that $y_i = -1$ and
  $W^t X_i + b \geq 1 \; \forall X_i$ such that $y_i = 1$
- Equivalently, we have
  $y_i(W^t X_i + b) \geq 1, \; \forall i$ (because $y_i \in \{-1, +1\}$)

- Note that a pattern X_i with $y_i = 1$ will either lie on the hyperplane $W^t X_i + b = 1$ or it is in the positive side satisfying $W^t X_i + b > 1$.
  Similarly, a pattern X_i with $y_i = -1$ will either fall on the hyperplane $W^t X_i + b = -1$ or it is in the negative side satisfying $W^t X_i + b < -1$.
  So, there is no X_i such that $-1 < W^t X_i + b < 1$ when the classes are linearly separable.
- The hyperplanes $W^t X_i + b = 1$ and $W^t X_i + b = -1$ are called *support planes*.
- The set of training vectors that fall on these support planes can be support vectors.
- When the classes are linearly separable, we can suitably scale W and b to obtain the support planes to satisfy $W^t X_i + b = 1$ and $W^t X_i + b = -1$.
- There is no pattern X_i falling between the two support planes. Further, the two support planes are parallel to each other as shown in Fig. 2.4.

4.2.2 Margin

The distance between the two planes is called the *Margin*. It is possible to show that the margin is a function of W. Training the SVM consists of learning a W that maximizes the margin. So, margin is important in theory.

Consider the point X shown in Fig. 4.1. Let X_{Proj} be the projection of X onto the hyperplane characterized by $g(X) = 0$. Let d be the normal distance between X and the hyperplane, or the distance between X and X_{Proj}, as shown in the figure.

- It is possible to write X in terms of X_{Proj} and d as
  $X = X_{\text{Proj}} + d\frac{W}{||W||}$ because d is the magnitude and the direction is same as that of W. The unit vector in the direction of W is $\frac{W}{||W||}$.

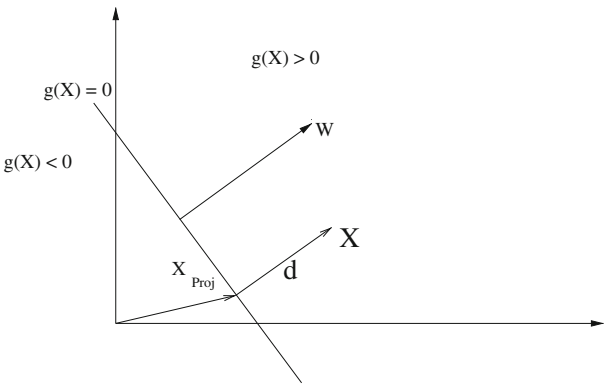

Fig. 4.1 Distance between a point and a hyperplane

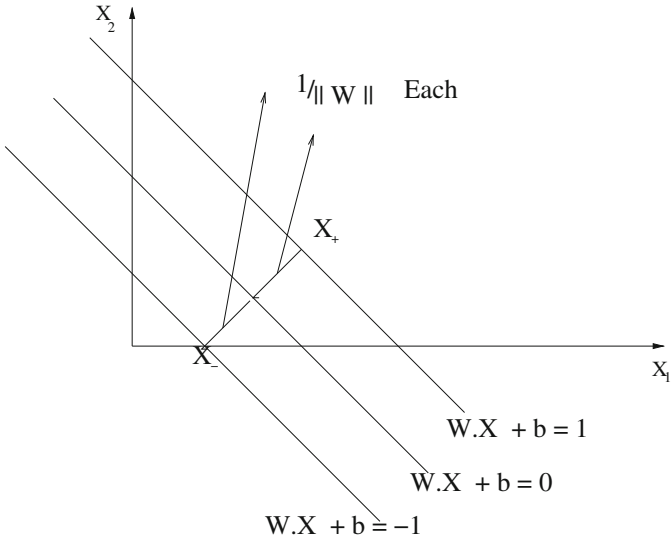

Fig. 4.2 Margin of the SVM

- Observe that
  $g(X) = W^t X + b = W^t(X_{\text{Proj}} + d\frac{W}{||W||}) + b = W^t X_{\text{Proj}} + b + d\frac{W^t W}{||W||} \Rightarrow$
  $g(X) = 0 + d\frac{W^t W}{||W||} = d\frac{W^t W}{||W||}$ because $W^t X_{\text{Proj}} + b = 0$ as X_{Proj} is on $g(X) = 0$.
- $g(X) = d\frac{W^t W}{||W||} \Rightarrow g(X) = d \mid\mid W \mid\mid .$
  So, $d = \frac{g(X)}{||W||}$.
- Hence, the distance between X and the hyperplane $g(X) = 0$ is given by $d = \frac{g(X)}{||W||}$. This result is useful in quantifying the margin.
- Consider Fig. 4.2. We have depicted three parallel lines in the two-dimensional space where $W.X$ is the *dot product* and it is equal to $W^t X$. These are

  1. $W.X + b = -1$ is the support line corresponding to the negative class.
  2. $W.X + b = 0$ which characterizes the decision boundary between the two classes.
  3. $W.X + b = 1$ corresponds to the support plane of the positive class.

- Consider the point X_+ on $W.X + b = W^t X + b = 1$. The normal distance from X_+ to the hyperplane (line in the two-dimensional case) $W.X + b = 0$ $(g(X) = 0)$ is given by
  $d = \frac{g(X_+)}{||W||}$; however, $g(X_+) = 1$ because X_+ is on the line (hyperplane in higher dimensions) $g(x) = W.X + b = 1$.
  So, the distance $d = \frac{1}{||W||}$.

- Similarly for the point X_- on $W.X + b = -1$, the normal distance to the line $W.X + b = 0$ is $d = \frac{1}{||W||}$.
- So, *Margin* is characterized by the sum of these distances and is Margin $= \frac{1}{||W||} + \frac{1}{||W||} = \frac{2}{||W||}$.

4.2.3 Maximum Margin

We are given that the classes are *linearly separable*. In such a case, we have the margin that exists between the two support planes and is given by

$$\text{Margin} = \frac{2}{||\ W\ ||}.$$

The idea is to find out a W that maximizes the margin. Once we get the W, $W^t X + b = 0$ gives us the corresponding *decision boundary*.

More precisely, the decision boundary or the *optimal hyperplane* is given by the solution of the following equivalent optimization problem.

Find $W \in \mathbb{R}^l$, $b \in \mathbb{R}$ to maximize $\frac{2}{W^t W}$ subject to $y_i(W^t X_i + b) \geq 1,\ \forall i$.

Instead of maximizing $\frac{2}{W^t W}$, we can *equivalently minimize* $\frac{W^t W}{2}$ to get

minimize $\frac{1}{2} W^t W$

subject to $y_i(W^t X_i + b) \geq 1,\ i = 1, 2, \ldots, n$

This is an optimization problem with quadratic criterion function $\frac{1}{2} W^t W$ and the constraints are in the form of *linear inequalities* $y_i(W^t X_i + b) \geq 1$.

It is possible to transform the constrained optimization problem into an unconstrained optimization problem using the Lagrangian given by

$$\mathscr{L} = \frac{1}{2} W^t W + \sum_{i=1}^{n} \alpha_i (1 - y_i(W^t X_i + b)).$$

The optimization problem is formulated so that the resulting form is *convex* ensuring globally optimal solution. In this case, the KKT conditions are both *necessary and sufficient*. These are

$$\nabla_W \mathscr{L} = W + \sum_{i=1}^{n} \alpha_i (-y_i) X_i = 0 \Rightarrow W = \sum_{i=1}^{n} \alpha_i y_i X_i.$$

$$\frac{\delta \mathscr{L}}{\delta b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0.$$

$$\alpha_i \geq 0\ \ \alpha_i(1 - y_i(W^t X_i + b)) = 0; \quad \text{and}\ \ 1 - y_i(W^t X_i + b) \leq 0,\ \forall i.$$

The important properties of the SVM are given by

1. We are given n training patterns and the training set of patterns is $\{(X_1, y_1), (X_2, y_2), \ldots, (X_n, y_n)\}$
2. The vector W is given by

$$W = \sum_{i=1}^{n} \alpha_i y_i X_i$$

which means W is a sum of the training patterns which are weighted by the corresponding αs and ys.
We will see later that we need not consider all the training patterns; there will be a small number of patterns with the corresponding αs to be nonzero. We need to consider them only. The other patterns will have their corresponding α values to be 0.

3. The equation

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

captures the property that

$$\sum_{i=1}^{n_-} \alpha_i = \sum_{i=n_-+1}^{n} \alpha_i.$$

The sum of the αs corresponding to the negative class is equal to that of the positive class. this property is useful in learning W.

4. Another important property, called the *complementary slackness* condition, is given by $\alpha_i(1 - y_i(W^t X_i + b)) = 0, \ \forall i$.

$$\alpha_i > 0 \Rightarrow y_i(W^t X_i + b) = 1$$

which means that if $\alpha_i > 0$, then the corresponding X_i is on a support plane. It is on the positive support plane if $y_i = 1$ else it is on the negative support plane.

4.2.4 An Example

We illustrate the learning of W, b, and αs using a two-dimensional example shown in Fig. 4.3. We have shown two Xs, negative examples, characterized by $(2, 1)^t$, and $(1, 3)^t$ and a O, a positive example, given by $(6, 3)^t$.

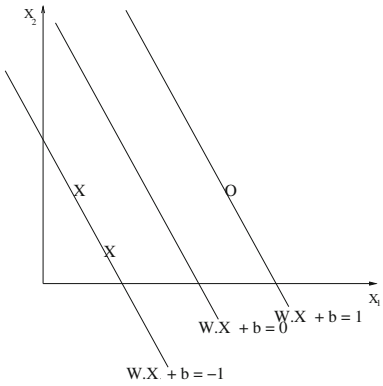

Fig. 4.3 Learning W and b from training data

- $(2, 1)^t$ and $(1, 3)^t$ are on the line $W^t X + b = -1$. So, we have

$$2w_1 + w_2 + b = -1$$

and

$$w_1 + 3w_2 + b = -1.$$

- Similarly, $(6, 3)^t$ is on the line $W^t X + b = 1$. So, we get

$$6w_1 + 3w_2 + b = 1.$$

- Solving the three equations, we get $w_1 = \frac{2}{5}$, $w_2 = \frac{1}{5}$, and $b = -2$.
- Note that $(4, 2)^t$ is on the boundary as $W^t(4, 2)^t + b = (\frac{2}{5}, \frac{1}{5})(4, 2)^t - 2 = 0$. Similarly, $(7, 1)^t$ is in the positive class and $(2, 0)^t$ is in the negative class.
- Further,

$$\sum_i \alpha_i y_i = 0 \Rightarrow -\alpha_1 - \alpha_2 + \alpha_3 = 0 \Rightarrow \alpha_3 = \alpha_1 + \alpha_2.$$

- Also

$$W = (\frac{2}{5}, \frac{1}{5})^t = -\alpha_1(1, 3)^t - \alpha_2(2, 1)^t + \alpha_3(6, 3)^t \Rightarrow$$

$$\alpha_1 = 0;\ \alpha_2 = \alpha_3 = \frac{1}{10}.$$

Note that $\alpha_1 = 0$. So, it is possible that αs corresponding to some of the patterns on the support planes could be 0.

4.3 Dual Problem

If we substitute

$$W = \sum_{i=1}^{n} \alpha_i y_i X_i$$

in the Lagrangian $\mathscr{L}$, we have

$$\mathscr{L} = \frac{1}{2} \sum_{i=1}^{n} \alpha_i y_i X_i^t \sum_{j=1}^{n} \alpha_j y_j X_j + \sum_{i=1}^{n} \alpha_i \left(1 - y_i (\sum_{j=1}^{n} \alpha_j y_j X_j^t X_i + b) \right)$$

By simplifying further, we get

$$\mathscr{L} = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j X_i^t X_j + b \sum_{i=1}^{n} \alpha_i y_i$$

By noting that

$$\sum_{i=1}^{n} \alpha_i y_i = 0,$$

we get

$$\mathscr{L} = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j X_i^t X_j$$

This is the *dual problem* and it is in terms of αs only. We use $\mathscr{L}_D$ for the dual and it is

$$\mathscr{L}_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j X_i^t X_j$$

$$\text{such that } \alpha_i \geq 0, \ \forall i \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

1. This is a *convex optimization problem*. It is possible to obtain α vector corresponding to the *global optimum*.
2. The vector $W = \sum_{i=1}^{n} \alpha_i y_i X_i$. So, optimization is over $\mathbb{R}^n$ *irrespective of the dimension of* X_i.
3. Many of the α_i are 0. *Support Vectors (SVs)* are the X_is corresponding to the nonzero α_is.
4. Let, $S = \{X_i | \alpha_i > 0\}$ be the set of SVs.

a. By *complementary slackness condition*, $X_i \in S \Rightarrow \alpha_i > 0 \Rightarrow$ $y_i(W^t X_i + b) = 1 \Rightarrow$.
X_i is the closest to the decision boundary.
b. We have $W = \sum_i \alpha_i y_i X_i = \sum_{X_i \in S} \alpha_i y_i X_i$.
Optimal W is a *linear combination of the support vectors*.
c. $b = y_j - W^t X_j$, where j is such that $\alpha_j > 0$.
d. Thus, both W and b are determined by $\alpha_j,\ j = 1, 2, \ldots, n$.
e. We can solve the dual optimization problem to obtain the optimal values of α_is. We can use the αs to get optimal values of both W and b.
f. Typically we would like to classify a new pattern Z based on the sign of $W^t X + b$.
Equivalently, by using $W = \sum_i\ \alpha_i y_i X_i$, we can classify a pattern Z based on the sign of $b + \sum_{X_i \in S}\ \alpha_i y_i X_i^t Z$. We do not need to use W explicitly.

4.3.1 An Example

Let us consider the example data shown in Fig. 4.4. There are five points. These are

- **Negative Class**: $(2, 0)^t$, $(2, 1)^t$, $(1, 3)^t$
- **Positive Class**: $(6, 3)^t$, $(8, 2)^t$

We have seen earlier that $W = (\frac{2}{5}, \frac{1}{5})^t$ and $b = -2$ for the patterns $(2, 1)^t,\ (1, 3)^t$, $(6, 3)^t$, first two from the negative class and the third from the positive class.
The α values are $\alpha_1 = 0, \alpha_2 = \alpha_3 = \frac{1}{10}$.
The remaining two patterns are such that the corresponding αs are 0.

1. $(2, 0)^t$ is from class -1. Based on the complementary slackness condition, we have $\alpha(1 - y(W^t X + b)) = 0$. Here, $y = -1,\ W = (\frac{2}{5}, \frac{1}{5})^t,\ X = (2, 0)^t$, and $b = -2$. So, $\alpha = 0$ because $1 - y(W^t X + b) = -\frac{1}{5} \neq 0$.

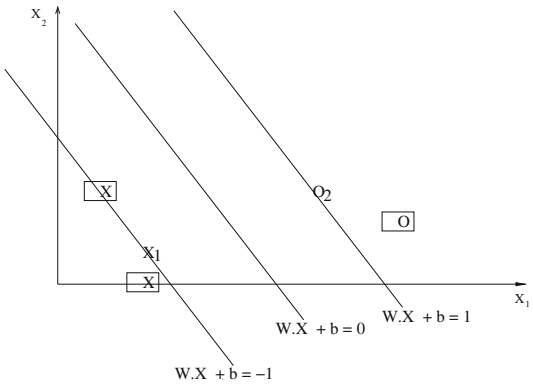

Fig. 4.4 α values

2. $(8, 2)^t$ is from class $+1$. Here, $\alpha = 0$ because $1 - y(W^t X + b) = -\frac{3}{5}$.
3. So, the SVs are $X_1 = (2, 1)^t$, $O_2 = (6, 3)^t$ and both have the same α value of $\frac{1}{10}$. The α values corresponding to the other three patterns are 0.
4. The points which are not support vectors or equivalently points with zero α value are indicated using a rectangular box around them in Fig. 4.4.

4.4 Multiclass Problems [2]

Classifiers like perceptron and SVM are based on linear discriminants and are ideally suited for two-class problems or binary classification problems. So, when the training data is from C (> 2) classes, then we need to build a *multiclass classifier* from a collection of binary classifiers. Some of the well-known possibilities are

1. Consider a *pair of classes* at a time; there are $\frac{C(C-1)}{2}$ such pairs. Learn a linear discriminant function for each pair of classes.
   Consider Fig. 4.5.

   These decisions are combined to arrive at the class label among the three classes C_1, C_2, and C_3. Note that there are three binary classifiers as shown in the figure. A problem is the ambiguous region marked in the middle. It is difficult to classify a point in this region.
2. For class C_i let the complementary region be

$$\bar{C}_i = \bigcup_{j=1, j \neq i}^{C} C_j$$

   Learn a linear discriminant function to classify to C_i or $\bar{C}_i$ for each i. Combine these binary classifiers to classify a pattern.

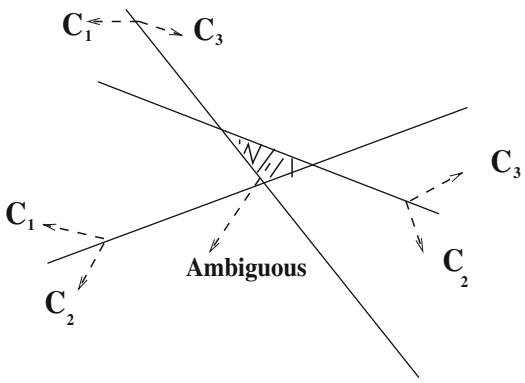

Fig. 4.5 Multiclass classification

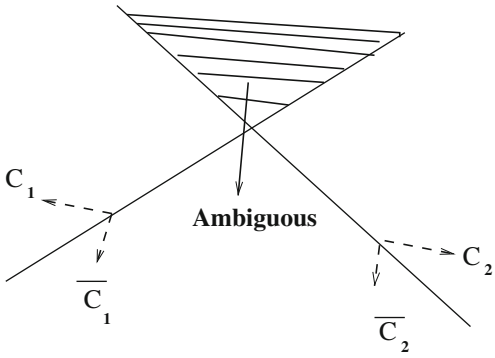

Fig. 4.6 Multiclass classification

Consider Fig. 4.6.

Note that even in this case, there is a region that is ambiguous as shown in the figure.

4.5 Experimental Results

Here, we considered Iris Setosa and Iris Versicolour classes which are linearly separable. We used the two features *sepal length* and *petal length* in building and testing the classifiers. We have used 60 patterns for training and the remaining 40 for testing. Both the perceptron and linear SVM classifiers have given us 100 % accuracy on the test data set. The weight vectors learnt are given in Table 4.1. Here, W_p is the weight vector learnt using Perceptron and W_s is the weight vector obtained using SVM.

4.5.1 Results on Multiclass Classification

SVM and Perceptron are inherently two-class classifiers. We use the traditional way of one-against-rest method to perform multiclass classification. *Weka*, a popular suite of machine learning software is used in realizing this.

We consider two well-known machine learning data sets: Iris and Pendigits. The number of instances, attributes, and the results are listed below. The data set is split

Table 4.1 Directions of W_p and W_s

W_p	W_s	Cosine (W_p, W_s)
$(2, 3.4, -9.1)^t$	$(-1, 0.9827, -1.96)^t$	0.80

Table 4.2 Results on iris dataset with three classes

No of training training patterns	No of test test patterns	Number of correctly classified patterns	Accuracy (Percentage)
75	75	71	94.67
83	67	63	94.03
90	60	57	95
99	51	49	96.08
105	45	43	95.56
110	40	38	95
113	37	35	94.60
117	33	32	96.67
120	30	29	96.67

into train and test sets, and fed into the multiclass classifier of Weka. Several iterations are carried out with different train-test percentage splits. Finally the Mean and the Standard Deviation are calculated. Also, we have provided the results for multiclass classifier using a tenfold cross validation.

We give below the details of our experiments.

1. **Iris Dataset**
   Number of Classes = 3
   Number of Data Points = 150
   Number of features = 5
   Out of the 5 features, 4 of them are *sepal length, sepal width, petal length*, and *petal width*. The *fifth* feature is a dependent feature; it is the *class label* which can assume one of three values corresponding to the 3 classes, Setosa, Versicolour, and Viriginica. We give the results in Table 4.2.
   By using tenfold cross validation, we obtained an accuracy of 96 %.
2. **Pendigits Dataset**
   Number of Classes = 10
   Number of Data Points = 10992
   Number of Features = 17
   Out of the 17 features, the 17th feature is the class label assuming one of 10 values corresponding the digit that is represented by 16 features. We have used Weka software that is described in the book by Witten, Frank and Hall, the details of which are provided in the references. We give results in Table 4.3.
   Using tenfold cross validation, we could classify with an accuracy of 93.52 %

Table 4.3 Results on pendigit dataset with ten classes

No of training training patterns	No of test test patterns	Number of correctly classified patterns	Accuracy (Percentage)
5496	5496	5145	93.61
5946	4946	4644	93.90
6695	4397	4124	93.79
7255	3737	3499	93.63
7694	3298	3101	94.02
8244	2748	2579	93.85
8574	2418	2270	93.88
8794	2198	2065	93.95

4.6 Summary

Classification based on SVMs is popular and is being used in a variety of applications. It is good to understand why it works and also its shortcomings. Some of the important features are

1. Both SVM and perceptron are *linear classifiers*.
2. It is possible to view the linear classifier to have the form $W^t X + b$. The training patterns are used to *learn* W and b.
3. In both the SVM and perceptron, the W vector may be viewed as a *linear combination* of the training patterns.

    a. In perceptron the iterations converge to a W_{k+1}, a correct weight vector, and it is

    $$W_{k+1} = \sum_{i=1}^{k} X^i,$$

    where X^k is misclassified by W_k.

    b. In SVM, the weight vector W is given by

    $$W = \sum_{X_i \in S} \alpha_i y_i X_i,$$

    where only *support vectors* matter.

4. Consider the data shown in Fig. 4.7.

    Here, there are two points each from the two classes as given by

    a. Negative Class: $X_1 = (1, 1)^t$, $X_2 = (1, 6)^t$

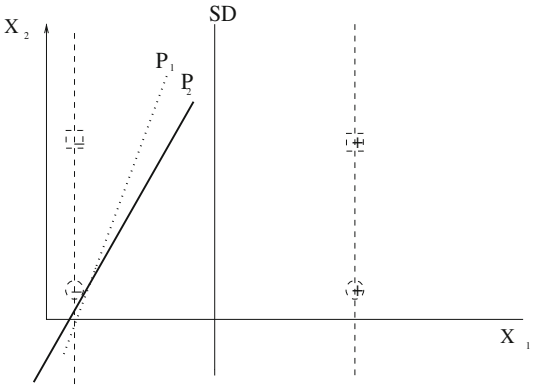

Fig. 4.7 Different support vector Sets

b. Positive Class: $X_3 = (5, 1)^t,\ X_4 = (5, 6)^t i$

The W and b vectors given by Perceptron and SVM can be obtained as follows:

a. If we use the order $X_1,\ X_2,\ X_3,\ X_4$, to compute the augmented W using the perceptron learning algorithm, we get $W = (5, -3)^t$ and $b = -3$, the corresponding decision is indicated using P_1 in the figure.
b. If we use the order $X_3,\ X_4,\ X_2,\ X_1$, we get $W = (12, -9)^t$, and $b = -4$ and we show the corresponding decision boundary using P_2 in the figure.
c. Using the SVM, we get two possible support vector sets. They are
   - The support vectors are $X_1 = (1, 1)^t$ and $X_3 = (5, 1)^t$. Because of $\sum_i \alpha_i y_i = 0$, we get
     $-\alpha_1 + \alpha_3 = 0 \Rightarrow \alpha_1 = \alpha_3 = \alpha$. So,
     $W = \alpha[(5, 1)^t - (1, 1)^t] = (4\alpha, 0)$.
     Also, because $(1, 1)^t$ is on the negative support line, we get
     $w_1 + w_2 + b = -1$.
     Similarly, for $(5, 1)^t$ which is on the positive support line, we have
     $5w_1 + w_2 + b = 1$.
     From these two equations, we get $w_1 = \frac{1}{2}$ and $w_2 = 0$. So, $W = (4\alpha, 0)^t = (\frac{1}{2}, 0)^t \Rightarrow \alpha = \frac{1}{8}$.
     From these, we get $b = -\frac{3}{2}$. So, the decision boundary, SD, is characterized by $\frac{x_1}{2} - \frac{3}{2} = 0$ as shown in the figure.
   - The other possibility is to have $X_2 = (1, 6)^t$ and $X_4 = (5, 6)^t$. Here also we get $W = (\frac{1}{2}, 0)^t$ and $b = -\frac{3}{2}$. Again the decision boundary is given by SD. In both the cases, W is orthogonal to SD.

   Even though both the Support Vector sets are different, we get the same W. So, in the case of the SVM also we can have multiple solutions, in terms of the SV sets. However, the W vector is the same.

5. Linear Support Vector Machine is a simple linear classifier. It is popularly used in linearly separable cases.

6. It is also used in classifying high-dimensional datasets even if the classes are not linearly separable. Some of the popular applications are **text classification** and classification of nodes and edges in **social networks**.
7. Experimental results on Iris data do not show much difference between Perceptron and Linear SVM in terms of accuracy.

References

1. Fan, R.-E., Chang, K.-W., Hsieh, C.-J, Wang, X.-R., Lin, C.-J.: LIBLINEAR: a library for large linear classification. JMLR **9**, 1871–1874 (2008)
2. Hsu, C.W., Lin, C.-J.: A comparison of methods for multiclass support vector machines. IEEE Trans. Neural Networks **13**(2), 415–425 (2002)
3. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
4. Rifkin, R.M.: Multiclass Classification, Lecture Notes, Spring08. MIT, USA (2008)
5. Witten, I.H., Frank, E., Hall, M.A.: Data Mining, 3rd edn. Morgan Kauffmann (2011)