CISC 371 Class 29

Support Vector Machine – Primal Formulation

Texts: [1] pp. 417–422; [2] pp. 150–153

Main Concepts:

- *Support vector: data vector $\vec{x}_s$ closest to a hyperplane*
- *SVM: Support Vector Machine, way to compute optimal hyperplane*
- *SVM: quadratic equation with an inequality constraint*

Sample Problem, Machine Inference: How can we optimally separate data vectors?

In the previous class in this course, we defined a *binary classification* problem as a decision of whether a given data vector $\vec{x}$ was in Class -1 or Class +1. Following the convention in machine learning, we will re-define a binary classification problem as a decision of whether a given data vector $\vec{x}$ is in

Class -1 or *Class +1*

We will now make this decision using a *separating hyperplane*, which we will write as $\mathbb{H}$. In an artifical neuron the hyperplane $\mathbb{H}$ was – and is here – a vector $\vec{w}$ and a scalar b. The vector $\vec{x}$ will be predicted, or classified, as:

Vector $\vec{x}$ is in Class +1 if and only if $\vec{x}^T\vec{w} + b \geq 0$

An equivalent mathematical definition is that the classification is performed in two stages. In the first stage, a *score* is assigned to a data vector. The score will be implicitly dependent on the weight vector $\vec{w}$ and the bias scalar b. We will write the score as z so, for the above classification, we would write

$$z(\vec{x}) \stackrel{\text{def}}{=} \vec{x}^T\vec{w} + b \tag{29.1}$$

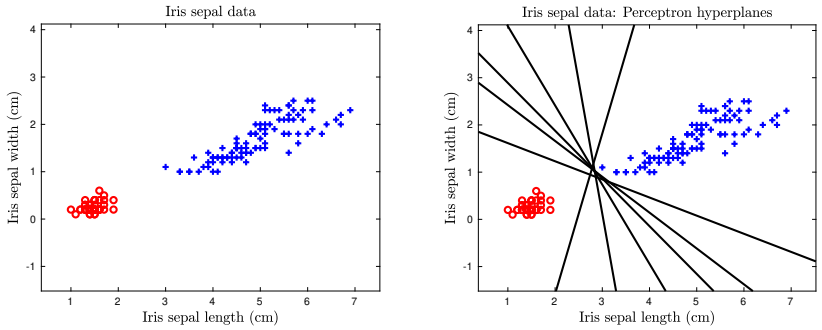
The next stage is to use the score of a vector to predict the classification of that vector. This can be done by using a quantization of the form

$$q : \mathbb{R} \to \{-1, +1\} \qquad \text{or} \qquad q(z) = \pm 1$$

Binary classification by means of a hyperplane can be written as

$$q(\vec{x}) = \text{sgn}(z(\vec{x})) \tag{29.2}$$

One of the difficulties we had with finding a separating hyperplane using the Perceptron Algorithm was that, depending on the initial estimate of the hyperplane, there were many less than ideal solutions to which the algorithm converges. Figure 29.1 illustrates 6 actual hyperplanes that were found for the Iris data using the Perceptron Algorithm; for the Iris data set, we defined label +1 as the type P Iris plant and label -1 as the type B Iris plant.



(A) **(B)**

Figure 29.1: Linearly separable Iris data. (A) Sepal measurements of the Iris data. (B) Separating hyperplanes that were found using the Perceptron Algorithm from random initial estimates. Each hyperplane separates the data correctly but with little or no margin.

We would prefer a hyperplane that, is in some sense, "mid-way" between the data vectors for the set with label +1 and the set with label -1. Such a hyperplane depends only on the data vectors that are *nearest* to the hyperplane. For the Iris data set that is used in Figure 29.1, each hyperplane is nearer to one set than to the other set. This is the situation that we want to correct. The method we will use is based on the concept of a *support vector* for the hyperplane.

29.1 Support Vector: Nearest Data Vector to Hyperplane

Suppose that the data vectors $\vec{x}_j$ are correctly separated by a hyperplane that is specified using $\vec{w}$ and b. This implies that each data vector in the set with label +1 is on the positive side of the hyperplane and that each data vector in the set with label -1 is on the negative side of the hyperplane.

Conceptually, a *support vector* $\vec{x}_s$ of a hyperplane $\mathbb{H}$ is a vector that is "closest" to $\mathbb{H}$. We will follow the convention that "closest" is measured as the Euclidean norm of the difference between any vector $\vec{x}$ and the hyperplane $\mathbb{H}$.

There are two common ways of interpreting the concept of a support vector:

- The set of feasible vectors is the entire set of m given data vectors $\vec{x}_j$

- There are two distinct sets of feasible vectors, those data vectors for which $q(\vec{x}_j) = -1$ and those data vectors for which $q(\vec{x}_j) = +1$

We will use the first interpretation. We note that important implementations, such as the current implementation of `MATLAB`, use the second interpretation.

We can specify a hyperplane $\mathbb{H}$ using an alternative method, for the purpose of defining a support vector and for our subsequent derivations. We can use a unit normal vector $\vec{n}$, for which $\|\vec{n}\| = 1$, and a bias scalar a, that are derived from the vector $\vec{w}$ and bias b as

$$\begin{aligned} \vec{n} &\stackrel{\text{def}}{=} \vec{w}/\|\vec{w}\| \\ a &\stackrel{\text{def}}{=} b/\|\vec{w}\| \end{aligned} \tag{29.3}$$

Using the conversions of Equation 29.3, the signed Euclidean distance from a point $\vec{x}$ to $\mathbb{H}$ is

$$\vec{n} \cdot \vec{x} + a \tag{29.4}$$

Our definition of a support vector is that it is a data vector $\vec{x}_j$ what is indexed by a natural number s, where $1 \leq s \leq m$. We can define the set $\mathbb{N}_S$ as the natural numbers that index the support vectors of $\mathbb{H}$, which is

$$\mathbb{N}_S = \underset{j \in \mathbb{N}_+ : j \leq m}{\operatorname{argmin}} \; |\vec{n} \cdot \vec{x}_j + a| \tag{29.5}$$

If $\mathbb{H}$ satisfies the basic property of having classifying at least one data vector as Class -1 and at least one as Class +1 then, in the vector space $\mathbb{R}^n$, there are at least n support vectors in the data set $\vec{x}_j \in \mathbb{R}^n$. Usually, there are many more than the minimum.

An example is a simple data set, with five vectors of size 2. Such a data set is illustrated in Figure 29.2, with the support vectors for the hyperplane being circled and being shown in larger fonts.

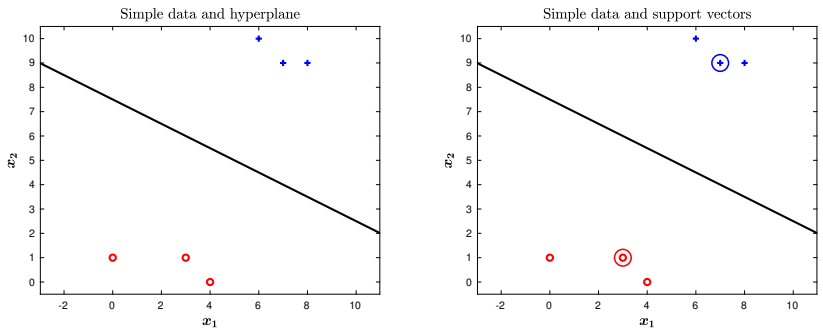


Figure 29.2: Simple data that are linearly separable. (A) Data and a separating hyperplane. (B) Support vectors for the hyperplane are circled.

29.2 Support Vector: Optimization Constraint

The illustration in Figure 29.1 suggests that requiring the classification of data to match the labels is a desirable objective, but it is not sufficient to describe what we really want. We seek a hyperplane $\mathbb{H}$ that is as "far" as possible from its support vectors, while still classifying as many data vectors as possible to their associated labels.

Another way to state the constraint is that we want a hyperplane that has the maximum *margin of separation* from its support vectors. Using our preferred definition of support vectors, in which the hyperplane $\mathbb{H}$ is not biased by the classification, we want$\mathbb{H}$ to be equally distant from the label +1 support vectors and from the label -1 support vectors. Suppose that this margin of separation is written as r. We can modify Figure 29.2 to include the margin; a small margin is illustrated in Figure 29.3(A) and the maximum margin is shown in Figure 29.3(B).

A method for finding the hyperplane that has the maximum margin of separation is referred to as a *Support Vector Machine*, or an SVM. Technically, linearly separable data have a hyperplane that can be specified as a weight vector $\vec{w}$ and a scalar bias value b, satisfying an objective function with a constraint:

$$\begin{aligned} \{\vec{w}^*, b^*\} &= \underset{\vec{w}\in\mathbb{R}^n, b\in\mathbb{R}}{\text{argmin}}\ \vec{w}^T\vec{w} \\ \text{such that, } \forall i \in \mathbb{N}_S: \quad y_i(\vec{w}^T\vec{x}_i + b) &\geq 1 \end{aligned} \tag{29.6}$$

Problem 29.6 is a direct, or primal, statement of the SVM problem. A derivation of this fundamental SVM constraint, using basic linear algebra, is provided in the extra notes for this class.

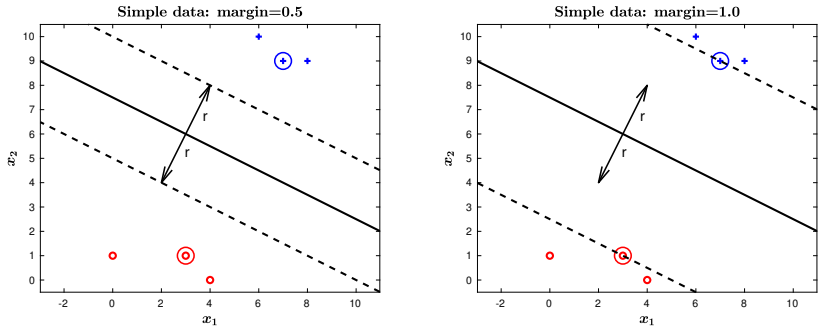


Figure 29.3: A separating hyperplane is shown as a solid line, and various margins as dashed lines. (A) A small margin r is partway between the hyperplane and the nearest data vectors. (B) A maximum margin r is the distance from the hyperplane to its support vectors.

When we apply the SVM method to the Iris data set, we find a hyperplane that optimally separates the two clusters of data vectors. The SVM hyperplane is illustrated in Figure 29.4.

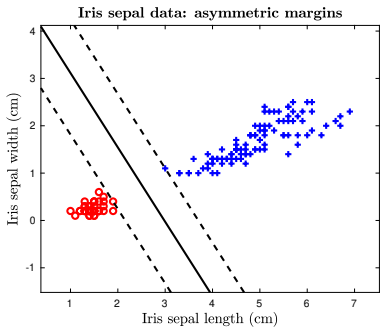


Figure 29.4: Separation of the Iris data vectors by an SVM. The hyperplane is shown as the solid line and hyperplanes at the support vectors are shown as dashed lines.

A consequence of the derivation of Problem 29.6 is that the bias b does not appear in the objective function. This implies that an SVM searches for the optimal *direction* of the weight vector $\vec{w}^*$; in 2D, this is the tilt of the hyperplane $\mathbb{H}$ from a reference axis. The bias b arises from the inequality constraints.

To solve Problem 29.6 effectively, and to manage the constraints, we will use the *dual* formulation of the SVM problem.

Extra Notes

Extra Notes on the Derivation of the Primal SVM Objective Function

A hyperplane can be specified using a unit normal vector $\vec{n}$, so we can simplify the constraints that it satisfies. If hyperplane is at margin distance r to a positive support vector $\vec{x}_i$, then

$$\begin{aligned} \vec{n} \cdot (\vec{x}_i - \vec{p}) &= r \\ \equiv \quad \vec{n}^T \vec{x}_i + a &= r \end{aligned} \tag{29.7}$$

A difficulty in using Equation 29.7 in optimization is the length constraint on the vector $\vec{n}$, which is that $\|\vec{n}\| = 1$.

Consider the use of a weight vector $\vec{w}$ that is a multiple of the unit vector $\vec{n}$. This allows us to replace occurrences of the unit normal vector $\vec{n}$ with occurrences of a scaled weight vector $\vec{w}$:

$$\begin{aligned} \|\vec{w}\| &\stackrel{\text{def}}{=} \frac{1}{r} \\ \Rightarrow \quad \vec{w} &= \frac{\vec{n}}{r} \end{aligned} \tag{29.8}$$

We can also define the value of a scalar b to be

$$b \stackrel{\text{def}}{=} \frac{a}{r}$$

We can take the support-vector constraint of Equation 29.7 and divide both sides by r. Using the term $\|\vec{w}\|$ of Equation 29.8, and the new scalar b, the constraint can be written as

$$\begin{aligned} \vec{n}^T \vec{x}_i + a &= r \\ \equiv \quad \frac{\vec{n}^T}{r} \vec{x}_i + \frac{a}{r} &= \frac{r}{r} \\ \equiv \quad \vec{w}^T \vec{x}_i + b &= 1 \end{aligned} \tag{29.9}$$

The constraint for a label -1 support vector $\vec{x}_i$ can likewise be written as

$$-\vec{w}^T \vec{x}_i - b = -1 \tag{29.10}$$

A common way to unify the expressions for support vectors, In Equation 29.9 and Equation 29.10, is to observe that the known label value for a positive data vector $\vec{x}_i$ is $y_i = +1$, and for a negative data vector $\vec{x}_i$ is $y_i = -1$. If we multiply both sides of Equation 29.9 by $y_i = +1$, and both sides of Equation 29.10 by $y_i = -1$, then the constraint equation for each support vector is the same:

$$y_i(\vec{w}^T \vec{x}_i + b) = 1 \tag{29.11}$$

Suppose that data vector $\vec{x}_g$ is a general data vector from the set with label +1 that is not a support vector. The distance from data vector $\vec{x}_g$ to the hyperplane must be greater than the distance r for a support vector. This means that the equality $= r$ of Equation 29.7 must be an inequality $> r$. Reasoning likewise for a general data vector with label -1, and using the multiplication by y_g to handle which set the general vector is in, we have an inequality that *every* data vector $\vec{x}_j$ must satisfy:

$$y_j(\vec{w}^T\vec{x}_j + b) \quad \geq \quad 1 \tag{29.12}$$

Referring back to the definition of the weight vector $\vec{w}$ in Equation 29.8, we want the margin of separation r to be as large as possible. These statements are equivalent:

$$\begin{aligned} & \text{maximize} \quad r \\ \equiv \quad & \text{minimize} \quad 1/r \\ \equiv \quad & \text{minimize} \quad \|\vec{w}\| \\ \equiv \quad & \text{minimize} \quad \|\vec{w}\|^2 \\ \equiv \quad & \text{minimize} \quad \vec{w}^T\vec{w} \end{aligned} \tag{29.13}$$

Equation 29.13 implies that, over the space of all possible weight vectors $\vec{w}$ and bias scalars b, we have a constrained optimization problem:

$$\begin{aligned} \{\hat{w}, \hat{b}\} \quad &= \quad \text{argmin}_{\vec{w}\in\mathbb{R}^n}\, \vec{w}^T\vec{w} \\ y_j(\vec{w}^T\vec{x}_j + b) \quad &\geq \quad 1 \end{aligned} \tag{29.14}$$

Equation 29.14 is a convex optimization problem with linear inequality constraints, expressed in the *primal* formulation. Equation 29.13 is the objective function that minimizes $\vec{w}$; the bias b is unbounded in the objective and enters the problem in the constraints.

End of Extra Notes

References

[1] Hastie T, Tibshirani R, Friedman J: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York, 2009

[2] Beck A: Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB. Siam Press, 2014