USING KERNEL FUNCTIONS TO OPTIMIZE DECISION HYPERPLANE ALGORITHM

CHENG CHEN

ABSTRACT. This article uses visualization methods to introduce the concept of support vector machines, conducts an in-depth study of the mathematical principles of support vector machines, mainly explores methods for solving decision hyperplanes, and uses kernel tricks to simplify complex operations caused by high-dimensional matrices. This article also provides an overview of real-world applications of support vector machines.

1. INTRODUCTION AND PRELIMINARIES

In machine learning, support vector machines are supervised max-margin models with associated learning algorithms that analyze data for classification and regression analysis.[1]

Example 1.1. In a two-dimensional space, suppose we have some data points that belong to two classes. We need to find a line that separates the two classes of data points to categorize the data. To find an optimal separating line, we need to make sure that the line is as widely spaced as possible for both types of data points.

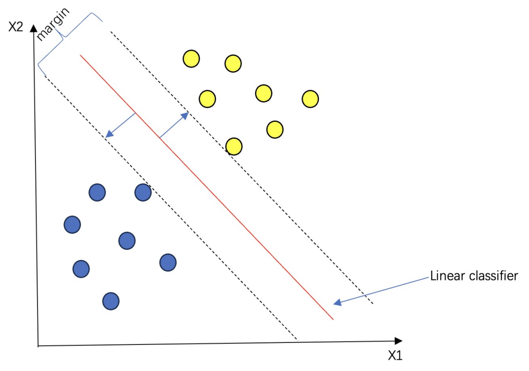

FIGURE 1.

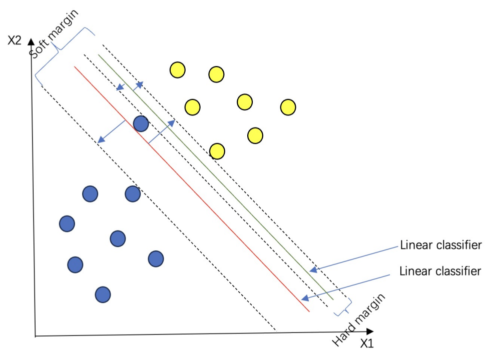

FIGURE 2.

In Figure 1[2], we can see that in the two-dimensional plane, we have two sets of data points, the blue and yellow points. To find the optimal separating line, or, in other words, the linear classifier,[1] we need to maximize the margin.

In Figure 2[2], suppose there exists an outlier. The margin will be significantly reduced. If we ignore this outlier, the separation distance also increases significantly. This introduces new concepts, soft margin and hard margin. Compared with hard margin, soft margin provides a certain fault tolerance rate.

Example 1.2. Let's look at Figure 3. The blue and yellow points are also in a two-dimensional space. However, we can't find a straight line to separate them.

In this case, we need to do a dimension-raising transformation. Turn two-dimensional space into three-dimensional space. We can find a hyperplane to separate these two types of data points, make it look like Figure 4.

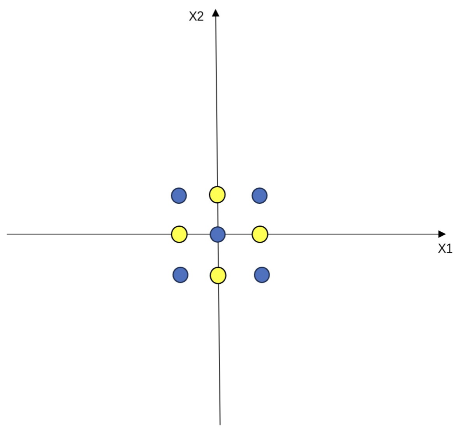

FIGURE 3.

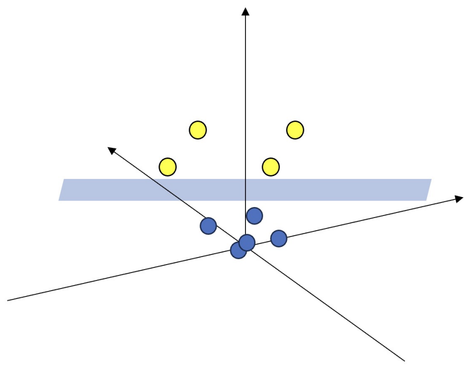

FIGURE 4.

However, increasing the dimension requires a clear dimension conversion function, as well as more data storage and computing requirements. Then we need to use Kernel Trick. Since the essence of SVM is to quantify the difference between two types of data sets, the kernel function can provide a measure of high-dimensional vector similarity. By selecting an appropriate kernel function, we can directly obtain the high-dimension difference of the data without knowing the specific dimension conversion function, and use this to make classification judgments.[2]

Result 1.3. This section mainly introduces the basic concepts of SVM, pointing out that we need to achieve the goal of maximum margin in linear classification. In the case of nonlinear classification, the ascending transformation is used to perform linear classification in high dimensions. The following sections will talk about the algorithm of solving decision hyperplane and the technique of using kernel function.

2. Solve SVM decision hyperplane

Classifying data is a common task in machine learning. Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in. In the case of support vector machines, a data point is viewed as a m-dimensional vector (a list of m numbers), and we want to know whether we can separate such points with a (m-1)-dimensional hyperplane.[1]

This type of problem can be summarized as follows: In order to distinguish two types of data, let n be the number of samples of the data, and m be the number of dimensions. How we can design a hyperplane with dimension (m-1) to distinguish two types of data.[2]

$$\begin{bmatrix} X_{11} & X_{12} & X_{13} & ... & X_{1m} \\ X_{21} & X_{22} & X_{23} & ... & X_{2m} \\ X_{31} & X_{32} & X_{33} & ... & X_{3m} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ X_{n1} & X_{n2} & X_{n3} & ... & X_{nm} \end{bmatrix}$$

The n $\times$ m matrix above shows this type of problem, and the equation of the hyperplane can be written as $W_1X_1 + W_2X_2 + W_3X_3 + ... + W_mX_m + B = 0$, where W can be considered as the weight corresponding to X.[2]

Example 2.1. Let's look at the simplified graph: Figure 5.[2]

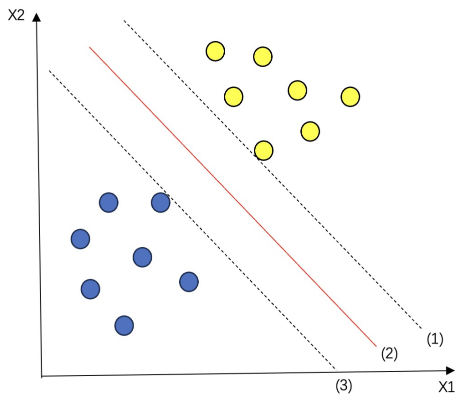

FIGURE 5.

The three lines are labelled as (1),(2),(3). The linear equation for line (2) is given above, which is $w_1x_1 + w_2x_2 + b = 0$. Line (1) and line (3) are formed by line (2) being translated upward or downward by c units respectively. Therefore, we have three equations for the lines.

(1) $w_1x_1 + w_2x_2 + b = c$
(2) $w_1x_1 + w_2x_2 + b = 0$
(3) $w_1x_1 + w_2x_2 + b = -c$

Since $c \neq 0$, we can simplify these equations,

(1) $\frac{w_1}{c}x_1 + \frac{w_2}{c}x_2 + \frac{b}{c} = 1$
(2) $\frac{w_1}{c}x_1 + \frac{w_2}{c}x_2 + \frac{b}{c} = 0$
(3) $\frac{w_1}{c}x_1 + \frac{w_2}{c}x_2 + \frac{b}{c} = -1$

Let $w_1' = \frac{w_1}{c}$, $w_2' = \frac{w_2}{c}$, $b' = \frac{b}{c}$, we have

(1) $w_1'x_1 + w_2'x_2 + b' = 1$
(2) $w_1'x_1 + w_2'x_2 + b' = 0$
(3) $w_1'x_1 + w_2'x_2 + b' = -1$

Since w' and b' are just the notations we need to solve, replacing them with w and b will not affect the calculation. So finally we have

(1) $w_1x_1 + w_2x_2 + b = 1$
(2) $w_1x_1 + w_2x_2 + b = 0$
(3) $w_1x_1 + w_2x_2 + b = -1$[2]

Result 2.2. In this part, the linear equations of positive and negative linear classification in two-dimensional plane are derived. In a two-dimensional plane, if we know the edge points, we can easily obtain the numerical expression of the edge lines. In the most ideal case, the edge lines are perpendicular with the line formed by connecting to the two edge points, and then the margin can reach its maximum.

Example 2.3.

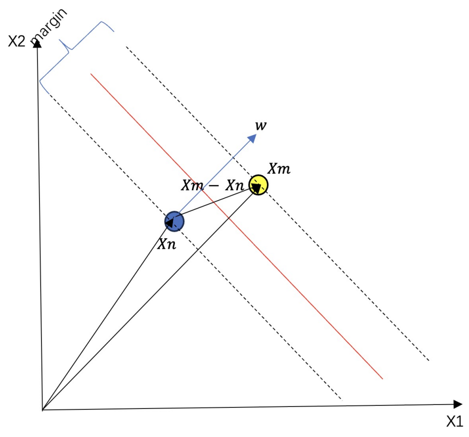

FIGURE 6.

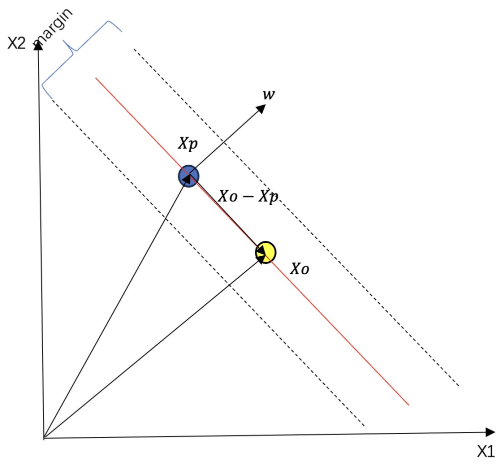

FIGURE 7.

In Figure 6[3], we pick two points x_m, x_n on the positive and negative hyperplane and have

$$w_1x_{1m} + w_2x_{2m} + b = 1 \tag{2.1}$$
$$w_1x_{1n} + w_2x_{2n} + b = -1 \tag{2.2}$$

(2.1)-(2.2), we have

$$w_1(x_{1m} - x_{1n}) + w_2(x_{2m} - x_{2n}) = 2$$
$$\vec{w} \cdot (\vec{x}_m - \vec{x}_n) = 2$$

In Figure 7[3], we pick two points x_o, x_p on the decision hyperplane and we have

$$w_1x_{1o} + w_2x_{2o} + b = 0 \tag{2.3}$$

$$w_1 x_{1p} = w_2 x_{2p} + b = 0 \tag{2.4}$$

(2.3)-(2.4), we have

$$w_1(x_{1o} - x_{1p}) + w_2(x_{2o} - x_{2p}) = 0$$
$$\vec{w} \cdot (\vec{x}_o - \vec{x}_p) = 0$$

So, we have

$$||\vec{x}_m - \vec{x}_n|| \times cos\theta \times ||\vec{w}|| = 2$$

Suppose the margin is L, we have

$$||\vec{x}_m - \vec{x}_n|| \times cos\theta = L$$

So,

$$L \times ||\vec{w}|| = 2$$
$$L = \frac{2}{||\vec{w}||}$$ [3]

Therefore, to obtain the longest margin L, we need to minimize $||\vec{w}||$.

For the positive hyperplane, we have

$$y_i = 1$$
$$\vec{w} \cdot \vec{x_i} + b \geq 1$$

For the negative hyperplane, we have

$$y_i = -1$$
$$\vec{w} \cdot \vec{x_i} + b \leq -1$$

Summarize them, we have

$$y_i \times (\vec{w} \cdot \vec{x_i} + b) \geq 1$$

So we need to minimize

$$||\vec{w}||$$

subject to

$$y_i \times (\vec{w} \cdot \vec{x_i} + b) \geq 1$$
$$i = 1, 2, 3, 4, ..., s, s \text{ is the total number of the data points}$$

where

$$||\vec{w}|| = \sqrt{\vec{w_1} + \vec{w_2}}$$

This means we need to minimize

$$f(w) = \frac{||\vec{w}||^2}{2}$$

subject to

$$g_i(w, b) = y_i \times (\vec{w} \cdot \vec{x_i} + b) - 1 \geq 0$$
$$i = 1, 2, 3, 4, ..., s, s \text{ is the total number of the data points}$$
$$\Downarrow$$
$$g_i(w, b) = y_i \times (\vec{w} \cdot \vec{x_i} + b) - 1 = p_i^2$$
$$i = 1, 2, 3, 4, ..., s, s \text{ is the total number of the data points}$$

So we can get Lagrange's equation

$$L(w,b,\lambda_i,p_i) = ||\vec{w}|| = \sqrt{\vec{w_1}+\vec{w_2}} - \sum_{i=1}^{s} \lambda_i \times (y_i \times (\vec{w}\cdot\vec{x}_i + b) - 1 - p_i^2)^{[3]} \quad (2.5)$$

Then we have

$$\frac{\partial L}{\partial w} = 0 \quad \Rightarrow \quad \vec{w} - \sum_{i=1}^{s} \lambda_i y_i \vec{x}_i = 0 \quad (2.6)$$

$$\frac{\partial L}{\partial b} = 0 \quad \Rightarrow \quad -\sum_{i=1}^{s} \lambda_i y_i = 0 \quad (2.7)$$

$$\frac{\partial L}{\partial b} = 0 \quad \Rightarrow \quad y_i \quad \times (\vec{w}\cdot\vec{x}_i + b) - 1 - p_i^2 = 0 \quad (2.8)$$

$$\frac{\partial L}{\partial b} = 0 \quad \Rightarrow \quad 2\lambda_i p_i = 0 \Rightarrow \lambda_i p_i^2 = 0 \quad (2.9)$$

By (2.8) and (2.9), we have

$$\lambda_i(y_i \times (\vec{w}\cdot\vec{x}_i + b) - 1) = 0$$

Since

$$y_i \times (\vec{w}\cdot\vec{x}_i + b) \geq 1$$

We only have two possible situations

$$y_i \times (\vec{w}\cdot\vec{x}_i + b) - 1 > 0, \lambda_i = 0$$

or

$$y_i \times (\vec{w}\cdot\vec{x}_i + b) - 1 = 0, \lambda_i \neq 0$$

Return to (2.5)

Suppose

$$\lambda_i < 0$$

to minimize $||\vec{w}||$, we need to make

$$y_i \times (\vec{w}\cdot\vec{x}_i + b) - 1 < 0$$

and then we have

$$-\lambda_i(y_i \times (\vec{w}\cdot\vec{x}_i + b) - 1 - p_i^2) < 0$$

However, we have

$$y_i \times (\vec{w}\cdot\vec{x}_i + b) - 1 \geq 0$$

This is a contradiction.

Thus,

$$\lambda_i \geq 0^{[3]}$$

Return to the original question,

minimize $f(w) = \frac{||\vec{w}||^2}{2}$

subject to $g_i(w,b) = y_i \times (\vec{w}\cdot\vec{x}_i + b) - 1 \geq 0, i = 1,2,3,4,...,s$

Suppose the optimal solutions for w, b are w^*, b^*

We have the new equation

$q(\lambda_i) = minimize(L(w, b, \lambda_i)) = minimize(f(w) - \sum_{i=1}^{s} \lambda_i \times g_i(w, b)), i = 1, 2, 3, 4, ..., s$

Since

$$\lambda \geq 0$$
$$g_i(\vec{w^*}, b^*) \geq 0$$

we have

$$q(\lambda_i) \leq f(\vec{w^*}) - \sum_{i=1}^{s} \lambda_i \times g_i(\vec{w^*}, b^*) \leq f(\vec{w^*}) \leq f(w)$$

Suppose we have a optimal solution λ_i^*, we need to make $q(\lambda_i^*)$ and $f(\vec{w^*})$ as similar as possible, which means

$$q(\lambda_i) \leq q(\lambda_i^*) \leq f(\vec{w^*}) \leq f(w)$$

Now we convert the original problem to a dual[4] problem

maximize $q(\lambda_i) = maxmize(minimize(L(w, b, \lambda_i))$

subject to $\lambda_i \geq 0, i = 1, 2, 3, 4, ...s$

and $q(\lambda_i^*) < f(\vec{w^*})$ is called weak dual, $q(\lambda_i^*) = f(\vec{w^*})$ is called strong dual.[3]

$q(\lambda_i^*) = f(\vec{w^*})$ satisfies slater's condition[5], and we can proof that we will get the optimal λ_i^* and $\vec{w^*}$ at the same time if the equation holds.

Proof.

$$f(w) \geq q(\lambda_i^*) = f(\vec{w}*) \quad \Rightarrow \quad f(w) \geq f(\vec{w^*})$$
$$f(w) \geq q(\lambda_i^*) = f(\vec{w^*}) \geq q(\lambda_i) \quad \Rightarrow \quad q(\lambda_i^*) \geq q(\lambda_i)$$

□

Use the KKT[6] solutions (2.6) and (2.7), we can simplify the original problem.

maximize $\quad q(\lambda_i) = maximize(minimize(\frac{||\vec{w}^2||}{2} - \sum_{i=1}^{s} \lambda_i \times (y_i \times (\vec{w} \cdot \vec{x}_i + b) - 1)))$

subject to

$$\lambda_i \geq 0, i = 1, 2, 3, 4, ..., s$$
$$\Downarrow$$

maximize $\quad q(\lambda_i) = maximize(\frac{1}{2}(\sum_{i=1}^{s} \lambda_i y_i \vec{x}_i) \cdot (\sum_{i=1}^{s} \lambda_j y_j \vec{x}_j) - \sum_{j=1}^{s} \lambda_i \times (y_i \times ((\sum_{i=1}^{s} \lambda_j y_j \vec{x}_j) \cdot \vec{x}_i + b) - 1)))$

$$\Downarrow$$

$$\text{maxmize} \quad q(\lambda_i) = maximize(\sum_{i=1}^{s} \lambda_i - \frac{1}{2} \sum_{i=1}^{s} \sum_{j=1}^{s} \lambda_i \lambda_j y_i y_j \vec{x}_i \cdot \vec{x}_j)$$

subject to

$$\lambda_i \geq 0, i = 1, 2, 3, 4, ..., s$$

Result 2.4. In this part, we suppose the margin is L. To maximize the margin, we need to get its expression. By choosing two points on the decision hyperplane, we can get $L = \frac{2}{||\vec{w}||}$. Now the problem we need to solve is to minimize $||\vec{w}||$. Since $||\vec{w}|| = \sqrt{\vec{w_1} + \vec{w_2}}$, we can use Lagrange's equation to minimize it and we get $\lambda_i \geq 0$. By using duality, Slater's condition and some equation transformation, we finally turn the question into maxmize $\quad q(\lambda_i) = maximize(\sum_{i=1}^{s} \lambda_i - \frac{1}{2}\sum_{i=1}^{s}\sum_{j=1}^{s} \lambda_i\lambda_j y_i y_j \vec{x}_i \cdot \vec{x}_j)$ subject to $\lambda_i \geq 0, i = 1, 2, 3, 4, ..., s$. We need to solve λ_i firstly, then use $\vec{w} = \sum_{i=1}^{s} \lambda_i y_i \vec{x}_i$ to solve $\vec{w}$. Finally, we can use $y_i \times (\vec{w} \cdot \vec{x}_i + b) - 1 = 0$ to solve b.[3]

3. Kernel trick

Let's return back to Figure 3 and Figure 4. For the previous problem

maxmize $\quad q(\lambda_i) = maximize(\sum_{i=1}^{s} \lambda_i - \frac{1}{2}\sum_{i=1}^{s}\sum_{j=1}^{s} \lambda_i\lambda_j y_i y_j \vec{x}_i \cdot \vec{x}_j)$

If the above objective equation has no solution on the original dimension, we can transform the original vectors, and get new dimension vectors.

The original dimension vector $x \Rightarrow$ the new dimension vector $T(x)$

So the original problem convert to

maxmize $\quad q(\lambda_i) = maximize(\sum_{i=1}^{s} \lambda_i - \frac{1}{2}\sum_{i=1}^{s}\sum_{j=1}^{s} \lambda_i\lambda_j y_i y_j T(\vec{x}_i) \cdot T(\vec{x}_j))$

subject to

$$\lambda_i \geq 0, i = 1, 2, 3, 4, ..., s$$

Now we have

$$\vec{x}_i = (x_{i1}, x_{i2}) \xrightarrow{T} T(\vec{x}_i) = (x_{i1}, x_{i2}, x_{i3}, ..., x_{in})$$

$$\vec{x}_j = (x_{j1}, x_{j2}) \xrightarrow{T} T(\vec{x}_j) = (x_{j1}, x_{j2}, x_{j3}, ..., x_{jn})$$

$$T(\vec{x}_i) \cdot T(\vec{x}_j) = \sum_{m=1}^{n} x_{im}x_{jm}$$ [7]

We can also use the kernel function $K(\vec{x_i}, \vec{x_j})$
and let

$$T(\vec{x_i}) \cdot T(\vec{x_j}) = K(\vec{x_i}, \vec{x_j})$$ [7]

Let's compare the two different methods.

Example 3.1. If we use method 1, suppose we need to transform the dimension from 2 to 6 then

$$\vec{x} = (x_1, x_2) \xrightarrow{T} T(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

$$\begin{aligned} T(\vec{x}_i) \cdot T(\vec{x}_j) &= (1, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}, x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2}) \cdot (1, \sqrt{2}x_{j1}, \sqrt{2}x_{j2}, x_{j1}^2, x_{j2}^2, \sqrt{2}x_{j1}x_{j2}) \\ &= 1 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} + x_{i1}^2x_{j1}^2 + 2x_{i1}x_{i2}x_{j1}x_{j2} \end{aligned}$$ [7]

Example 3.2. If we use method 2, the kernel function

we have

$$\begin{aligned} K(\vec{x}_i\vec{x}_j) &= (1 + \vec{x}_i \cdot \vec{y}_i)^2 \\ &= (1 + x_{i1}x_{j1} + x_{i2}x_{j2})^2 \\ &= 1 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} + x_{i1}^2x_{j1}^2 + 2x_{i1}x_{i2}x_{j1}x_{j2} \end{aligned}$$ [7]

Let's look at the general form

$$K(\vec{x}_i\vec{x}_j) = (c + \vec{x}_i \cdot \vec{y}_i)^d$$

The constant term parameter c in the polynomial kernel function plays an important role. Because of its existence, the expression of the final dot product result contains data combinations from low-order terms to high-order terms. If the constant term parameter is missing, only the higher-order term will be included in the expression.[7]

Let's look at the following examples

Example 3.3. $c = 1, d = 2$

$$\begin{aligned} K_1(\vec{x}_i\vec{x}_j) &= (1 + \vec{x}_i \cdot \vec{y}_i)^2 \\ &= (1 + x_{i1}x_{j1} + x_{i2}x_{j2})^2 \\ &= 1 + \underline{2x_{i1}x_{j1} + 2x_{i2}x_{j2}} + \underline{x_{i1}^2x_{j1}^2 + 2x_{i1}x_{i2}x_{j1}x_{j2}} \end{aligned}$$ [7]

Example 3.4. $c = 0, d = 2$

$$\begin{aligned} K_2(\vec{x}_i\vec{x}_j) &= (\vec{x}_i \cdot \vec{y}_i)^2 \\ &= (x_{i1}x_{j1} + x_{i2}x_{j2})^2 \\ &= \underline{x_{i1}^2x_{j1}^2 + 2x_{i1}x_{i2}x_{j1}x_{j2}} \end{aligned}$$ [7]

For this type of polynomial kernel function without a constant parameter c, we can also add and combine multiple polynomial kernel functions of different degrees so that the result has both high and low degree terms, thereby enriching the diversity of dimensions.[7]

Example 3.5. $(c = 0, d = 1) + (c = 0, d = 2)$

$$\begin{aligned} K_1'(\vec{x}_i\vec{x}_j) + K_2(\vec{x}_i\vec{x}_j) &= \vec{x}_i \cdot \vec{y}_i + (\vec{x}_i \cdot \vec{y}_i)^2 \\ &= (x_{i1}x_{j1} + x_{i2}x_{j2}) + (x_{i1}x_{j1} + x_{i2}x_{j2})^2 \\ &= \underline{x_{i1}x_{j1} + xi_2x_{j2}} + \underline{x_{i1}^2x_{j1}^2 + 2x_{i1}x_{i2}x_{j1}x_{j2}}^{[7]} \end{aligned}$$

Let's find a method to solve the decision hyperplane in an infinite dimensional space.

Look at the example,

Example 3.6.

$$\vec{x} = (x_1, x_2) \xrightarrow{T} T(\vec{x}) = (a_1x_1, a_1x_2, a_1x_1x_2, a_2x_1^2, a_2x_2^2, ..., a_nx_1^n, a_nx_2^n, ..., \infty)$$

Method 1

$$T(\vec{x}_i) \cdot T(\vec{x}_j) = \infty \cdot \infty$$

We can't use this method.

Method 2

$$T(\vec{x}_i) \cdot T(\vec{x}_j) = K(\vec{x}_i, \vec{x}_j)$$

Then we can use radial basis function kernel[8]

$$K(\vec{x}_i, \vec{x}_j) = e^{-\gamma||\vec{x}_i - \vec{x}_j||^2}$$

Since the value of the RBF kernel decreases with distance and ranges between zero (in the infinite-distance limit) and one (when $x_i = x_j$), it has a ready interpretation as a similarity measure.[8]

To simplify the calculation, set $\gamma = 0.5$

$$\begin{aligned} K(\vec{x}_i, \vec{x}_j) &= e^{-\frac{||\vec{x}_i = \vec{x}_j||^2}{2}} \\ &= e^{-\frac{1}{2}(\vec{x}_i - \vec{x}_j)\cdot(\vec{x}_i) - \vec{x}_j)} \\ &= e^{-\frac{1}{2}(\vec{x}_i\cdot\vec{x}_i + \vec{x}_j\cdot\vec{x}_j - 2\vec{x}_i\vec{x}_j)} \\ &= e^{-\frac{1}{2}(||\vec{x}_i||^2 + ||\vec{x}_j||^2 - 2\vec{x}_i\vec{x}_j)} \\ &= e^{-\frac{1}{2}(||\vec{x}_i||^2 + ||\vec{x}_j||^2)}e^{\vec{x}_i\vec{x}_j} \end{aligned}$$

let

$$C = e^{-\frac{1}{2}(||\vec{x}_i||^2 + ||\vec{x}_j||^2)}$$

then

$$K(\vec{x}_i, \vec{x}_j) = Ce^{\vec{x}_i\vec{x}_j}$$

Use Taylor series expansion

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

then

$$\begin{aligned} K(\vec{x}_i, \vec{x}_j) &= Ce^{\vec{x}_i \vec{x}_j} \\ &= C\sum_{n=0}^{\infty} \frac{\vec{x}_i \vec{x}_j^n}{n!} \\ &= C\sum_{n=0}^{\infty} \frac{K_{Poly(n)}(\vec{x}_i \vec{x}_j)}{n!} [7] \end{aligned}$$

This expression represents an infinite number of polynomial kernel functions without a constant parameter C, arranged in order from low order to infinite high order, adjusted according to different factorial coefficients and then added. According to the previous introduction, its results can reflect the characteristics of infinite diversity in the combination of high and low order items of infinite dimensional data.[7] Let's look at some numerical examples to understand the meaning of γ values,

Result 3.7. In this part, we compare the difference between using a dimension-raising conversion function and a kernel function. The limitation of the dimension-raising conversion function in infinite dimensions is also pointed out. Then we use radial basis function kernel to show how we can achieve the result of vector similarity in infinite dimensions without substantially stepping into infinite dimensions. Generally speaking, when facing SVM nonlinear classification problems, we can directly use radial basis function kernel.[7]

Example 3.8.

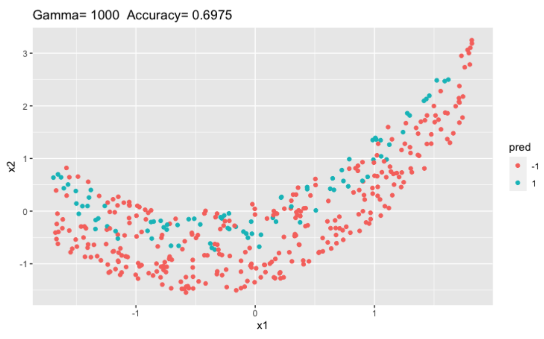

Figure 8. $\gamma = 0.01$

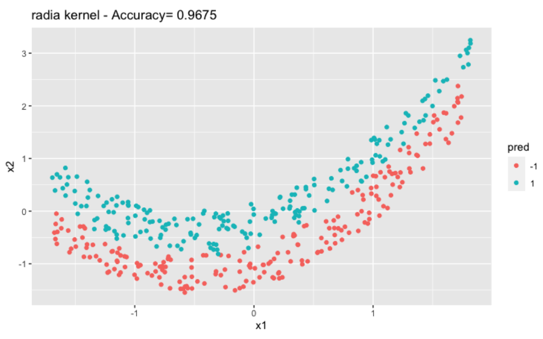

Figure 9. $\gamma = 1$

Based on a quadratic function $2 + 3x + x^2$ and adding corresponding random numbers, we can generate a two-dimensional data set xy. Through the sign of the random number added to the quadratic function, we can distinguish positive data from negative data and add a label of plus or minus 1 to it. Through the ggplot chart, we can see that this is a nonlinear classification problem. As we know, we can directly use radial basis function kernel when facing SVM nonlinear classification problems. Now we set $\gamma = 0.01$, use the test model to train and view the accuracy on the training set.

Let's look at Figure 8[13], we set $\gamma = 0.01$, we can see the accuracy is 0.6975, from the previous result, the smaller the gamma value, the greater the tolerance for distance in calculations. When we increase the value of γ to 1(see Figure 9[13]), we can find the accuracy has been significantly improved, which is 0.9675.

In practice, we can use a loop statement to try different gamma values, compare the results, and choose the optimal parameters.[13]

4. SVM applications in real world

SVMs are widely used in the real world and can be used to solve various problems:

i. Text and Hypertext Categorization: SVMs are very well suited for text categorization. SVMs achieve substantial improvements over the currently best performing methods and behave robustly over a variety of different learning tasks. Furthermore they are fully automatic, eliminating the need for manual parameter tuning.[9]

ii. Image Classification: Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevance feedback.[1] Decision systems are modeled on fuzzy natural vision-based reasoning, that exploit techniques for measuring human consensus about spatial-taxon structure. A system based on natural vision-based reasoning is highly non-linear and dynamical. It arrives at an end-point spatial-taxon by adjusting to human input as it iterates.[10]

iii. Handwriting Recognition: SVMs can be used to recognize handwritten characters and digits. Convolutional neural network (CNN) has recently been used as an efficient unsupervised feature vector extractor and a distinct Support Vector Machine (SVM) was used as the corresponding classifier. Recognition accuracies are comparable with the state-of-the-art.[11]

iv. Regression, Time Series, and Forecasting: Multivariate pattern analysis (MVPA) methods such as support vector machines (SVMs) have been increasingly applied to fMRI and sMRI analyses, enabling the detection of distinctive imaging patterns.[12]

5. Conclusion

This article provides a method for solving decision hyperplanes based on hard margin, using duality, Slater's condition, KKT conditions, radial basis function kernel and Tayler series. To extend SVM to cases in which the data are not linearly separable, the hinge loss function is helpful.[1] After having a certain understanding of the mathematical principles of support vector machines, future research should focus on practical applications.

References

1. Wikipedia contributors. (2023, November. 4). *Support vector machine.* Wikipedia. https://en.wikipedia.org/wiki/Support_vector_machine
2. FunInCode. (2022, January. 27). *What is support Vector Machine SVM, eight minutes intuitive understanding of its nature*[Video]. bilibili.

https://www.bilibili.com/video/BV16T4y1y7qj/?spm_id_from=333.788&vd_source=79243681ec5b7cb8ef1c09d532c914eb

3. FunInCode. (2022, February. 25). *Machine Learning Essentials - Math Essentials for SVM Support Vector Machines*[Video]. bilibili.
https://www.bilibili.com/video/BV13r4y1z7AG/?spm_id_from=333.788&vd_source=79243681ec5b7cb8ef1c09d532c914eb
4. Wikipedia contributors. (2023, November. 2). *Duality (mathematics)*. Wikipedia.
https://en.wikipedia.org/wiki/Duality_(mathematics)
5. Wikipedia contributors. (2023, November. 24). *Slater's condition*. Wikipedia.
https://en.wikipedia.org/wiki/Slater%27s_condition
6. Wikipedia contributors. (2023, December. 2). *Karush–Kuhn–Tucker conditions*. Wikipedia.
https://en.wikipedia.org/wiki/Karush%E2%80%93Kuhn%E2%80%93Tucker_conditions
7. FunInCode. (2022, March. 11). *SVM support vector machine - Kernel Trick detailed explanation*[Video]. bilibili.
https://www.bilibili.com/video/BV1Nb4y1s7pE/?spm_id_from=333.788&vd_source=79243681ec5b7cb8ef1c09d532c914eb
8. Wikipedia contributors. (2023, December. 18). *Radial basis function kernel*. Wikipedia.
https://en.wikipedia.org/wiki/Radial_basis_function_kernel
9. Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. In *Machine Learning: ECML-98*(Lecture Notes in Computer Science, Vol. 1398, pp. 137–142). Springer.
https://doi.org/10.1007/BFb0026683
10. Barghout, L. (2015). Spatial-Taxon Information Granules as Used in Iterative Fuzzy-Decision-Making for Image Segmentation. In *Granular Computing and Decision-Making*(Studies in Big Data, Vol. 10, pp. 285–318).
https://doi.org/10.1007/978-3-319-16829-6_12
11. Maitra, D. S., Bhattacharya, U., & Parui, S. K. (2015, August). CNN based common approach to handwritten character recognition of multiple scripts. In *Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR 2015)*(pp. 1021–1025). IEEE.
https://doi.org/10.1109/ICDAR.2015.7333916
12. Gaonkar, B., & Davatzikos, C. (2013). Analytic estimation of statistical significance maps for support vector machine based multi-variate image analysis and classification. *NeuroImage, 78,* 270–283.
https://doi.org/10.1016/j.neuroimage.2013.03.066
13. FunInCode. (2022, April. 22). *Support Vector Machine SVM Final Chapter-R Language Example Sharing*[Video]. bilibili.
https://www.bilibili.com/video/BV1xB4y127cs/?spm_id_from=333.788&vd_source=79243681ec5b7cb8ef1c09d532c914eb

Department of Mathematics, University of Manitoba, Winnipeg, CA.
Email address: chenc8@myumanitoba.ca