SVM and SVR

Zhiyuan Wei

October 2022

Contents

SVM

1 Overview of SVM

SVM is an unsupervised machine learning method, commonly used in binary classification problems. Compared with logistic regression, it introduces the concept of kernel function, which has a better classification effect on non-linear relations; at the same time, due to the introduction of the dual problem, it makes the complexity of calculation change from the size of dimension to the number of samples, avoiding dimensional explosion. However, as the essence of SVM is a quadratic programming problem, the large number of samples requires a lot of storage space and time, which is not easy to implement; at the same time, SVM has some difficulties in solving multi-classification problems.

2 Hard Margin SVM

The problem: We are given a sample set $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n)\}, y_i \in \{+1, -1\}$ and asked to find a hyperplane to classify the samples and maximize the margin (which is γ in the below graph).

Preknowledge: In the n-dimentional space, hyperplane can be represented as $\boldsymbol{w}^T\boldsymbol{x} + b = 0$ where $\boldsymbol{w} = (w_1; w_2; ...; w_n)$ is the normal vector and b is the displacement item. The Euclid distance from point $\boldsymbol{x} = (x_1, x_2, ..., x_m)$ to hyperplane $\boldsymbol{w}^T\boldsymbol{x} + b = 0$ is $\frac{|\boldsymbol{w}^T\boldsymbol{x}+b|}{||\boldsymbol{w}||}$ where $||\boldsymbol{w}|| = \sqrt{w_1^2 + w_2^2 + ... + w_n^2}$.

2.1 Setting up Lagrangian function

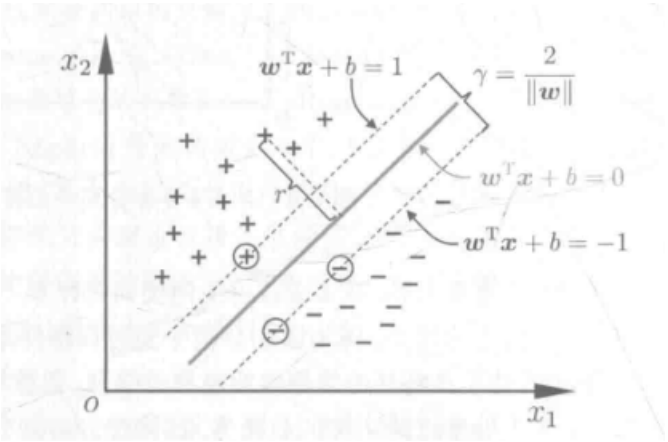

Figure 1: supporting vector and margin

We can set up the problem we want to solve:

$$\max_{\boldsymbol{w},b} \min_{\boldsymbol{x}_i, i=1,2,...,m} \frac{2|\boldsymbol{w}^T\boldsymbol{x}_i + b|}{||\boldsymbol{w}||}$$
$$s.t. \boldsymbol{w}^T\boldsymbol{x}_i + b > 0, y_i > 0$$
$$\boldsymbol{w}^T\boldsymbol{x}_i + b < 0, y_i < 0$$

where $\boldsymbol{x}_i = (x_{1i}; x_{2i}; ...; x_{ni})$ which is the vector of features of the ith sample and n is the number of features. $\boldsymbol{w} = (w_1; w_2; ...; w_n)$ is the normal vector we need to optimize and b is the displacement item.

This problem is equivalent to

$$\max_{\boldsymbol{w},b} \min_{\boldsymbol{x}_i, i=1,2,...,m} \frac{2y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)}{||\boldsymbol{w}||}$$
$$s.t. y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) > 0$$

We further write it as:

$$\max_{\boldsymbol{w},b} \frac{2}{||\boldsymbol{w}||} \min_{\boldsymbol{x}_i, i=1,2,...,m} y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)$$
$$s.t. \exists r > 0, \min_{\boldsymbol{x}_i, y_i, i=1,2,...,m} y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) = r$$

By scaling w and b we get:

$$\max_{\boldsymbol{w},b} \frac{2}{||\boldsymbol{w}||}$$
$$s.t. \min_{\boldsymbol{x}_i, y_i, i=1,2,...,m} y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) = 1$$

By changing the maximization problem to minimization problem and changing the form of the constraint we get:

$$\min_{\boldsymbol{w},b} \frac{||\boldsymbol{w}||^2}{2}$$
$$s.t. y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$$

$(\max_{\boldsymbol{w},b} \frac{2}{||\boldsymbol{w}||})$ is equivalent to $(\min_{\boldsymbol{w},b} \frac{||\boldsymbol{w}||^2}{2})$ because 1. $f(x) = \frac{2}{x}$ is a decreasing function on (0,+∞), so maximizing $\frac{2}{x}$ is equivalent to minimizing x when $x > 0$. Therefore, $(\max_{\boldsymbol{w},b} \frac{2}{||\boldsymbol{w}||})$ is equivalent to $(\min_{\boldsymbol{w},b} ||\boldsymbol{w}||)$. 2. $f(x) = \frac{x^2}{2}$ is an increasing function on (0,+∞), so minimizing x is equivalent to minimizing $\frac{x^2}{2}$ when $x > 0$. Therefore, $(\min_{\boldsymbol{w},b} ||\boldsymbol{w}||)$ is equivalent to $(\min_{\boldsymbol{w},b} \frac{||\boldsymbol{w}||^2}{2})$. Altogether, we have $(\max_{\boldsymbol{w},b} \frac{2}{||\boldsymbol{w}||})$ is equivalent to $(\min_{\boldsymbol{w},b} \frac{||\boldsymbol{w}||^2}{2})$

The reason why we use $\frac{||\boldsymbol{w}||^2}{2}$ is that it helps us form the dual problem correctly and helps calculation. (If we use $||\boldsymbol{w}||^2$, there will not be the quadratic term in the objective function in the dual problem. If we use $||\boldsymbol{w}||$, then there will be a square root term in the objective function in the dual problem and this will make calculation more complex. We will see how these happen in the following part.)

2.2 Dual problem

This is a convex quadratic programming problem and we can solve it directly. However, we can change it to a dual problem so that (1) we can include kernel functions, (2) we can change hard margin and soft margin SVM problems into the same form, which will will see later, (3) we make the constraint depend only on number of samples (m) and irrelevant to number of dimensions (n).

The relevant preknowledge about prime and dual problems is not discussed in the note. You can easily find online materials about it.

We include Lagrangian multiplier $\alpha_i \geq 0$ and the Lagrangian function can be written as

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{||\boldsymbol{w}||^2}{2} + \sum_{i=1}^{m} \alpha_i (1 - y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b)),$$

where $\boldsymbol{\alpha} = (\alpha_1; \alpha_2; ...; \alpha_m)$,

We have

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i = 0,$$
$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{m} \alpha_i y_i = 0,$$

which is

$$\boldsymbol{w}^* = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i$$
$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

Put the above two results to the Lagrangian function we get:

$$\begin{aligned} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2}(\sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i)^T (\sum_{j=1}^{m} \alpha_j y_j \boldsymbol{x}_j) - \sum_{i=1}^{m} \alpha_i y_i (\sum_{j=1}^{m} \alpha_j y_j \boldsymbol{x}_j)^T \boldsymbol{x}_i + \sum_{i=1}^{m} \alpha_i \\ &= \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i y_i \boldsymbol{x}_i^T \alpha_j y_j \boldsymbol{x}_j - \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i y_i \alpha_j y_j \boldsymbol{x}_j^T \boldsymbol{x}_i + \sum_{i=1}^{m} \alpha_i \\ &= -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j + \sum_{i=1}^{m} \alpha_i \end{aligned}$$

Therefore we get the dual problem:

$$\max_{\boldsymbol{\alpha}} -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^T\boldsymbol{x}_j + \sum_{i=1}^{m}\alpha_i,$$
$$s.t. \sum_{i=1}^{m}\alpha_i y_i = 0$$
$$\alpha_i \geq 0, i = 1, 2, ..., m$$

2.3 Solve the dual problem

This process needs to fit KKT (Karush-Kuhn-Tucker) conditions (we neglect the conditions that partial derivatives of Lagrangian function with respect to the variables we need to optimize equal to zero ($\frac{\partial L}{\partial \boldsymbol{w}} = 0$, $\frac{\partial L}{\partial b} = 0$) because these conditions have been incorporated in the dual problem) :

$$\begin{cases} \alpha_i \geq 0; \\ 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \leq 0; \\ \alpha_i(1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)) = 0 \end{cases} \tag{1}$$

For any sample (x_i, y_i), we have either $\alpha_i = 0$ or $1 - y_i(w^T x_i + b) = 0$. If $\alpha_i = 0$, then the sample will not show up in the dual problem and will not affect $f(x)$ (it is not a supporting vector). If $\alpha_i > 0$, then we have $y_i f(x_i) = 1$, then the sample point is on the boundary (it is a supporting vector).

We can use the KKT conditions to help solve Lagrangian problems or check the validity of the solution we get.

Then we know

$$\begin{aligned} \exists(\boldsymbol{x}_k, y_k) s.t. 1 - y_k(\boldsymbol{w}^T\boldsymbol{x}_k + b) &= 0 \\ y_k(\boldsymbol{w}^T\boldsymbol{x}_k + b) &= 1 \\ y_k^2(\boldsymbol{w}^T\boldsymbol{x}_k + b) &= y_k \\ \boldsymbol{w}^T\boldsymbol{x}_k + b &= y_k \\ b^* &= y_k - \boldsymbol{w}^T\boldsymbol{x}_k \\ b^* &= y_k - \sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i^T\boldsymbol{x}_k \\ b^* &= y_k - \sum_{i\in S}\alpha_i y_i \boldsymbol{x}_i^T\boldsymbol{x}_k \end{aligned} \tag{2}$$

where $S = \{i|\alpha_i > 0, i = 1, 2, ..., m\}$, i.e., the set of subscripts of supporting vectors.

In reality, we usually calculate the mean solution of all supporting vectors, which is

$$b^* = \frac{1}{|S|}\sum_{k\in S}(y_k - \sum_{i\in S}\alpha_i y_i \boldsymbol{x}_i^T\boldsymbol{x}_k) \tag{3}$$

Till now we have

$$\boldsymbol{w}^* = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i$$
$$b^* = y_k - \sum_{i \in S} \alpha_i y_i \boldsymbol{x}_i^T \boldsymbol{x}_k$$

As long as we know α_i's, we can get the hyperplane. In fact, for small dataset, we can calculate the α_i's by directly solving the Lagrangian problem but for larger dataset, we shall use other methods, which will be shown later.

2.4 Example

People always debate whether someone is a great basketball player. Suppose we are doing this job to find a way of classification and we are using # of MVPs and # of championships as features and there are three people being considered (Jordan, James and Justin). We have their information as follows:

Table 1: Player information

Name	**# of MVPs**	**# of championships**	**whether great player**
Jordan	5	6	1
James	4	4	1
Justin	0	0	-1

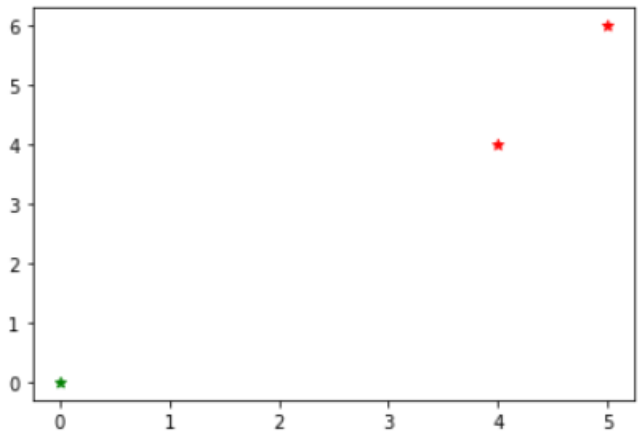

Figure 2: plot of the three players

In this problem we have

$$m = 3, x = \begin{bmatrix} 5 & 4 & 0 \\ 6 & 4 & 0 \end{bmatrix}, y = [11 - 1] \tag{4}$$

The dual problem is therefore:

$$\max_{\boldsymbol{\alpha}} -\frac{1}{2}((5*5+6*6)\alpha_1^2 + (4*5+4*6)\alpha_1\alpha_2 + (4*4+4*4)\alpha_2^2) + \alpha_1 + \alpha_2 + \alpha_3,$$
$$s.t. \alpha_1 + \alpha_2 - \alpha_3 = 0$$
$$\alpha_i \geq 0, i = 1, 2, ..., m$$

That is:

$$\begin{gathered}\max_{\boldsymbol{\alpha}} -\frac{1}{2}(61\alpha_1^2 + 44\alpha_1\alpha_2 + 32\alpha_2^2) + \alpha_1 + \alpha_2 + \alpha_3, \\ s.t. \alpha_1 + \alpha_2 - \alpha_3 = 0 \\ \alpha_i \geq 0, i = 1, 2, ..., m\end{gathered}$$

The KKT conditions are:

$$\begin{cases}\alpha_i \geq 0; \\ 1 - 5w_1 - 6w_2 - b \leq 0; \\ 1 - 4w_1 - 4w_2 - b \leq 0; \\ 1 + b \leq 0; \\ \alpha_1(1 - 5w_1 - 6w_2 - b) = 0; \\ \alpha_2(1 - 4w_1 - 4w_2 - b) = 0; \\ \alpha_3(1 + b) = 0\end{cases} \tag{5}$$

We know

$$\begin{aligned}\boldsymbol{w}^* &= \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i \\ b^* &= y_k - \sum_{i \in S} \alpha_i y_i \boldsymbol{x}_i^T \boldsymbol{x}_k\end{aligned}$$

and through substituting the values of y_i and $\boldsymbol{x}_i$ we get $w_1 = 5\alpha_1 + 4\alpha_2$ and $w_2 = 6\alpha_1 + 4\alpha_2$. In this problem, we know that Justin must be a supporting vector because he is the sole sample in his category (if not, we can always expand the margin until he is a supporting vector). Therefore, To calculate b, we let k=3 in the above expression of b and get $b = -1$.

Then the KKT conditions become

$$\begin{cases}\alpha_i \geq 0; \\ 2 - 61\alpha_1 - 44\alpha_2 \leq 0; \\ 2 - 44\alpha_1 - 32\alpha_2 \leq 0; \\ \alpha_1(2 - 61\alpha_1 - 44\alpha_2) = 0; \\ \alpha_2(2 - 44\alpha_1 - 32\alpha_2) = 0;\end{cases} \tag{6}$$

We can calculate the possible combination of (α_1, α_2) from the last two equations in the KKT conditions: $(0, \frac{1}{16}), (\frac{2}{61}, 0), (-\frac{3}{2}, \frac{17}{8})$. The corresponding values of the terms we want to maximize are $\frac{1}{16}, \frac{2}{61}, -\frac{139}{2}$ respectively. Therefore, the α_i's we want are $\alpha_1 = 0, \alpha_2 = \frac{1}{16}, \alpha_3 = \frac{1}{16}$. Then we calculate w_1, w_2 and the corresponding hyperplane is:

$$f(x_1, x_2) = \frac{1}{4}x_1 + \frac{1}{4}x_2 - 1 \tag{7}$$

Players who falls below the line are not classified as great basketball player and those who are above the line deserve the title.

3 Kernel Functions

In hard margin SVM, we assume that the sample points are linearly separable. However, in reality, there may not be a hyperplane that can correctly classify the original sample points.

To solve this problem, we can map the original space to a higher dimensional space to make the sample linearly separable in the space. One can prove that if the original space has finite dimensions, then there must exists a higher dimensional space to make the sample linearly separable.

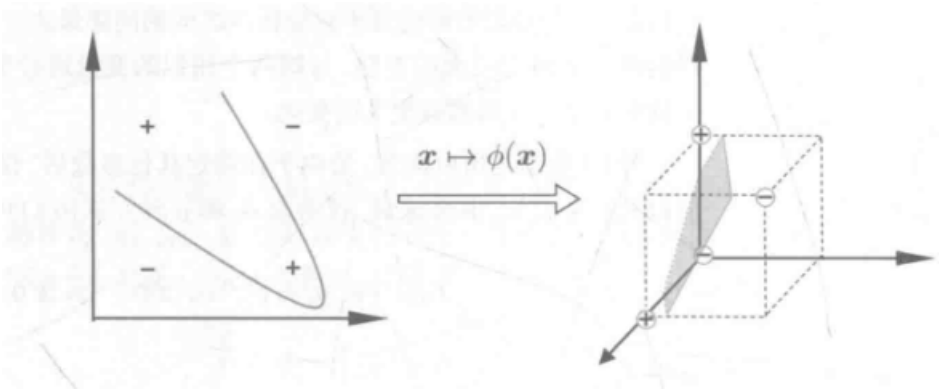

Figure 3: nonlinear mapping

We use $\phi(\boldsymbol{x})$ to represent the eigenvector after mapping $\boldsymbol{x}$. Then the hyperplane in the new space can be represented as:

$$f(x) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + b \tag{8}$$

Similar to the situation in hard margin SVM, we have Lagrangian problem:

$$\begin{gathered} \max_{\boldsymbol{w},b} \frac{||\boldsymbol{w}||^2}{2} \\ s.t. y_i(\boldsymbol{w}^T \phi(\boldsymbol{x}_i) + b) \geq 1 \end{gathered}$$

And the dual problem:

$$\begin{gathered} \max_{\boldsymbol{w},b} -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j) + \sum_{i=1}^{m} \alpha_i, \\ s.t. \sum_{i=1}^{m} \alpha_i y_i = 0 \\ \alpha_i \geq 0, i = 1, 2, ..., m \end{gathered}$$

Solving the dual problem needs calculation of $\phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$. Because the number of dimensions of the new space may be fairly large or even infinite, therefore it is hard to calculate $\phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$ directly. To solve the issue, we can think of a function:

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle = \phi(\boldsymbol{x}_i)^{\mathrm{T}} \phi(\boldsymbol{x}_j) \tag{9}$$

i.e., the inner product of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in the new space equals the result calculated by function $\kappa(,)$ in their original space. Then we won't need to calculate the inner product in the high-dimensional new space and the dual problem can be rewritten as:

$$\begin{gathered}\max_{\boldsymbol{w},b} -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) + \sum_{i=1}^{m}\alpha_i, \\ s.t. \sum_{i=1}^{m}\alpha_i y_i = 0 \\ \alpha_i \geq 0, i = 1,2,...,m\end{gathered}$$

After solving the problem we get:

$$\begin{aligned}f(\boldsymbol{x}) &= \boldsymbol{w}^{*T}\phi(\boldsymbol{x}) + b \\ &= \sum_{i=1}^{m}\alpha_i y_i \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}) + b \\ &= \sum_{i=1}^{m}\alpha_i y_i \kappa(\boldsymbol{x},\boldsymbol{x}_i) + b\end{aligned} \tag{10}$$

$\kappa(.,.)$ is the kernel function.

If we know the precise form of the mapping $\phi(.)$ then we can write $\kappa(.,.)$. However, in reality we may not know the precise form of $\phi(.)$, so does kernel functions exist? What kind of functions can be kernel functions?

Theorem: Let χ be the input space. $\kappa(.,.)$ is a symmetric function defined on $\chi \times \chi$. Then κ is kernel function if and only if for any data $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m\}$, kernel matrix K is always semi-definite:

$$K = \begin{bmatrix} \kappa(\boldsymbol{x}_1,\boldsymbol{x}_1) & \cdots & \kappa(\boldsymbol{x}_1,\boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_1,\boldsymbol{x}_m) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \kappa(\boldsymbol{x}_i,\boldsymbol{x}_1) & \cdots & \kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_i,\boldsymbol{x}_m) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \kappa(\boldsymbol{x}_m,\boldsymbol{x}_1) & \cdots & \kappa(\boldsymbol{x}_m,\boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_m,\boldsymbol{x}_m) \end{bmatrix} \tag{11}$$

This shows that as long as the kernel matrix of a symmetric function is semi-definite, it can be used as kernel function.

The choice of kernel function is crucial to SVM problems. Here are some common kernel functions:

Table 2: Common kernel function

Name	Expression	Parameters
Linear kernel	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) = \boldsymbol{x}_i^T\boldsymbol{x}_j$	
Polynomial kernel	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) = (\boldsymbol{x}_i^T\boldsymbol{x}_j)^d$	$d \geq 1$ is the degree of the polynomial
Gaussian kernel	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) = exp(-\frlac{\|\|\boldsymbol{x}_i-\boldsymbol{x}_j\|\|^2}{2\sigma^2})$	$\sigma > 0$ is the width of the Gaussian kernel
Laplace kernel	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) = exp(-\frac{\|\|\boldsymbol{x}_i-\boldsymbol{x}_j\|\|}{\sigma})$	$\sigma > 0$
Sigmoid kernel	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) = tanh(\beta\boldsymbol{x}_i^T\boldsymbol{x}_j + \theta)$	tanh is hyperbolic tangent function, $\beta > 0, \theta < 0$

Besides, we can use linear combination of functions:

If κ_1 and κ_2 are kernel functions, then for any positive numbers γ_1 and γ_2, the linear combination $\gamma_1\kappa_1 + \gamma_2\kappa_2$ is also kernel function.

If κ_1 and κ_2 are kernel functions, then direct product of the kernel functions $\kappa_1 \otimes \kappa_2(\boldsymbol{x}, \boldsymbol{z}) = \kappa_1(\boldsymbol{x}, \boldsymbol{z})\kappa_2(\boldsymbol{x}, \boldsymbol{z})$ is also kernel function.

If κ_1 is kernel functions, then to any function g(x): $\kappa(\boldsymbol{x}, \boldsymbol{z}) = g(\boldsymbol{x})\kappa_1(\boldsymbol{x}, \boldsymbol{z})g(\boldsymbol{z})$ is kernel function.

4 Soft Margin SVM

4.1 Setting up Lagrangian function

In reality, it tends to be hard to find a suitable kernel function to make sample points completely linearly separable. Even if we find one, it is hard to determine whether it is an over fitted result.

Therefore, we can allow for some errors on some sample points and therefore we include soft margin.

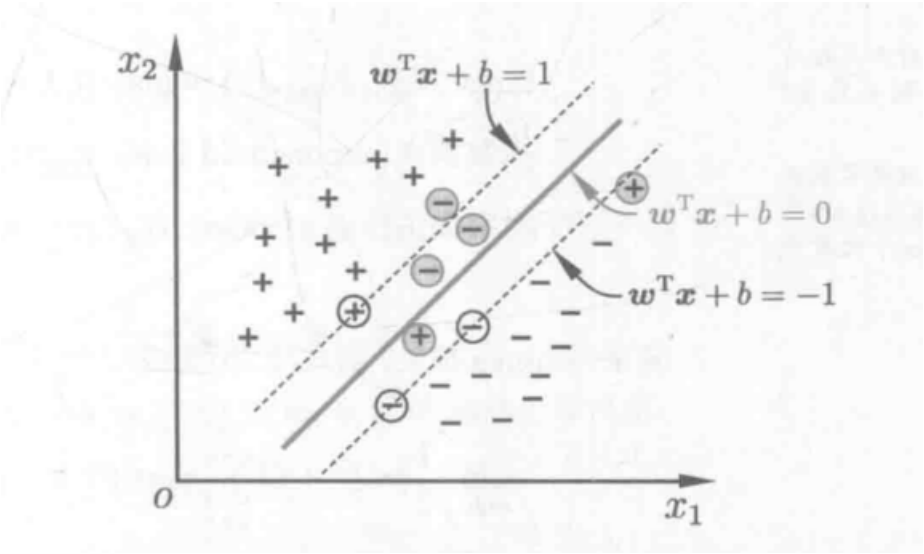

Figure 4: Soft margin

We allow for some samples to violate the constraint:

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$$

At the same time, we want to minimize the number of samples that violate the constraint, therefore the optimization goal can be:

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \ell_{0/1}\left(y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) - 1\right) \tag{12}$$

where $C > 0$ is a constant and $\ell_{0/1}$ is 0/1 loss function.

$$\ell_{0/1}(z) = \begin{cases} 1, & \text{if } z < 0 \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

When C is close to infinity, (12) forces the samples to fit $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$, then the soft margin SVM is equivalent to hard margin SVM.

However, $\ell_{0/1}$ doesn't have good mathematial properties as it is neither convex nor continuous. Hence, we usually use other loss functions to substitute it.

hinge loss function: $\ell_{\text{hinge}}(z) = \max(0, 1 - z)$;

exponential loss function: $\ell_{\exp}(z) = \exp(-z)$;

logistic loss function: $\ell_{\log}(z) = \log(1 + \exp(-z))$.

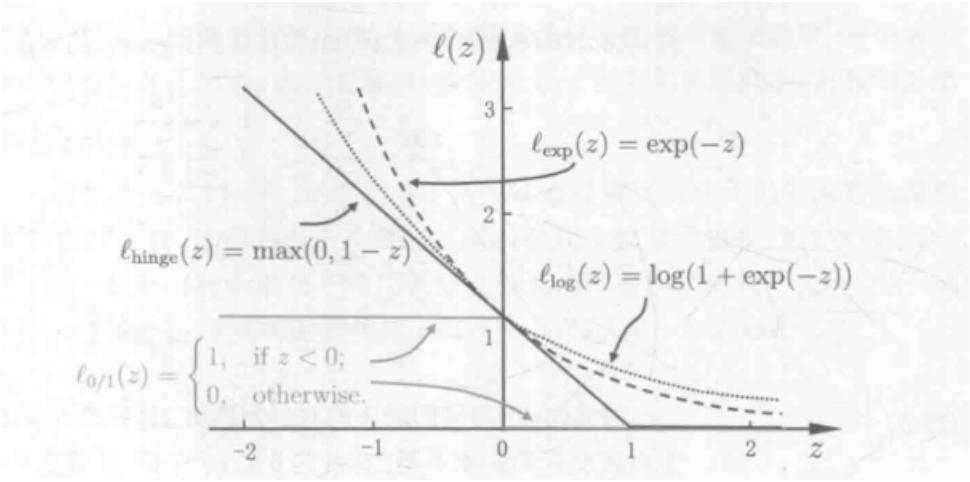

Figure 5: different loss functions

We use hinge loss function as an example. Then (12) becomes

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} max(0, 1 - y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right)) \tag{14}$$

We include slack variables $\xi_i = max(0, 1 - y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right)) \geq 0$, then we can rewrite (14) as:

$$\begin{gathered} \min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \xi_i \\ s.t. y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 1 - \xi_i \\ \xi_i \geq 0, i = 1, 2, ..., m \end{gathered}$$

This is the soft margin SVM. Every sample has a slack variable to represent its degree of violating the constraint.

4.2 Set up the dual problem and solve it

Again, using Lagrangian multiplier, we get

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\mu}) = \frac{||\boldsymbol{w}||^2}{2} + C\sum_{i=1}^{m}\xi_i + \sum_{i=1}^{m}\alpha_i(1-\xi_i - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)) - \sum_{i=1}^{m}\mu_i\xi_i, \tag{15}$$

where $\alpha_i \geq 0$ and $\mu_i \geq 0$ are Lagrangian multipliers.

$$\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{w}} &= \boldsymbol{w} - \sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x_i} = 0, \\
\frac{\partial L}{\partial b} &= -\sum_{i=1}^{m}\alpha_i y_i = 0, \\
\frac{\partial L}{\partial \xi_i} &= C - \alpha_i - \mu_i = 0,
\end{aligned}$$

which is

$$\boldsymbol{w}^* = \sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x_i}, \tag{16}$$

$$\sum_{i=1}^{m}\alpha_i y_i = 0, \tag{17}$$

$$C = \alpha_i + \mu_i. \tag{18}$$

Put these to (6.2) and we find the dual problem:

$$\begin{gathered}
\max_{\boldsymbol{\alpha}} -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \boldsymbol{x_i}^T\boldsymbol{x_j} + \sum_{i=1}^{m}\alpha_i, \\
s.t. \sum_{i=1}^{m}\alpha_i y_i = 0 \\
0 \leq \alpha_i \leq C, i = 1, 2, ..., m
\end{gathered}$$

Rewrite it as

$$\begin{gathered}
\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \boldsymbol{x_i}^T\boldsymbol{x_j} - \sum_{i=1}^{m}\alpha_i, \\
s.t. \sum_{i=1}^{m}\alpha_i y_i = 0 \\
0 \leq \alpha_i \leq C, i = 1, 2, ..., m
\end{gathered}$$

We find that the only difference between this problem and that of hard margin SVM is that the constraint to α_i bacomes $0 \leq \alpha_i \leq C$ instead of $0 \leq \alpha_i$. Therefore, we can use the same method for hard margin SVM to solve the problem (directly solve or include kernel function).

We have KKT conditions:

$$\begin{cases} \alpha_i \geq 0, \mu_i \geq 0, \\ y_i f(\boldsymbol{x_i}) - 1 + \xi_i \geq 0, \\ \alpha_i (y_i f(\boldsymbol{x_i}) - 1 + \xi_i) = 0 \\ \xi_i \geq 0, \mu_i \xi_i = 0 \end{cases} \tag{19}$$

For any sample $(\boldsymbol{x}_i, y_i)$, we have either $\alpha_i = 0$ or $y_i(w^T \boldsymbol{x}_i + b) = 1 - \xi_i$. If $\alpha_i = 0$, then the sample will not influence $f(\boldsymbol{x})$ (it is not a supporting vector). If $\alpha_i > 0$, then we have $y_i f(\boldsymbol{x}_i) = 1 - \xi_i$, then the sample point is a supporting vector. From (18), we know if $\alpha_i < C$, then $\mu_i > 0$ and we have $\xi_i = 0$, i.e., the sample is on the boundary of the maximum margin. If $\alpha_i = C$, then we have $\mu_i = 0$, in which case if $\xi_i \leq 1$, then the sample falls within the maximum margin and if $\xi_i > 1$, then it is classified incorrectly.

5 Use SMO to calculate α_i's

5.1 Rewrite the problem

We can combine the dual problems we need to solve in hard margin SVM and soft margin SVM as (because hard margin SVM is soft margin SVM with $C = +\infty$, so we can write them in the same form as below):

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j k(\boldsymbol{x_i}, \boldsymbol{x_j}) y_i y_j - \sum_{i=1}^{m} \alpha_i, \tag{20}$$

$$s.t. \sum_{i=1}^{m} \alpha_i y_i = 0$$
$$0 \leq \alpha_i \leq C, i = 1, 2, ..., m$$

where $k(\boldsymbol{x_i}, \boldsymbol{x_j})$ is the kernel function and we write it as k_{ij} in the rest of this part. The problem satisfies the KKT conditions:

$$\begin{cases} \alpha_i \geq 0, \mu_i \geq 0, \\ y_i f(\boldsymbol{x_i}) - 1 + \xi_i \geq 0, \\ \alpha_i (y_i f(\boldsymbol{x_i}) - 1 + \xi_i) = 0 \\ \xi_i \geq 0, \mu_i \xi_i = 0 \end{cases} \tag{21}$$

We can directly solve it because the expression we need to maximize is a quadratic one. However, the complexity of the problem is proportional to the number of samples. When the data set is large, training the SVM requires solving a large-scale quadratic programming (QP) problem and it costs much time. Because of it, we use SMO (Sequential Minimal Optimization) to get α_i's. The general idea of the SMO algorithm is to decompose this large QP problem into a series of small QP sub-problems as much as possible; then in the inner loop, these small QP problems are solved analytically, rather than numerical optimization, thus reducing computation time. Specifically, we fix all α_i's except two each

time and get the optimized values of the two α_i's. Then we choose other two α_i's to start a new round of optimization and such process continues on until the result becomes consistent.

We define the expression we want to minimize as ϕ. Then we have

$$\begin{aligned}\phi(\alpha_1,\alpha_2) =& \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j k_{ij}\alpha_i\alpha_j - \sum_{j=1}^{m}\alpha_i \\ =& \frac{1}{2}y_1^2 k_{11}\alpha_1^2 + \frac{1}{2}y_2^2 k_{22}\alpha_2^2 + y_1y_2k_{12}\alpha_1\alpha_2 + c_1y_1\alpha_1 + c_2y_2\alpha_2 - \alpha_1 - \alpha_2 \\ & + \text{ Const} \\ =& \frac{1}{2}k_{11}\alpha_1^2 + \frac{1}{2}k_{22}\alpha_2^2 + y_1y_2k_{12}\alpha_1\alpha_2 + c_1y_1\alpha_1 + c_2y_2\alpha_2 - \alpha_1 - \alpha_2 + \text{ Const}\end{aligned}$$

where $c_1 = \sum_{i=3}^{m} y_i\alpha_i k_{1i}$ and $c_2 = \sum_{i=3}^{m} y_i\alpha_i k_{2i}$, where α_i's will be changed to optimized α_i's in second, third, ... round of optimization. Therefore we write them as:

$$\begin{aligned}c_1 &= \sum_{j=3}^{m} y_j k_{1j}\alpha_j^{\text{old}} = d_1^{\text{old}} - b^{\text{old}} - y_1k_{11}\alpha_1^{\text{old}} - y_2k_{21}\alpha_2^{\text{old}} \\ c_2 &= \sum_{j=3}^{m} y_j k_{2j}\alpha_j^{\text{old}} = d_2^{\text{old}} - b^{\text{old}} - y_1k_{12}\alpha_1^{\text{old}} - y_2k_{22}\alpha_2^{\text{old}}\end{aligned} \tag{22}$$

where α_i^{old} and b^{old} is the optimized α_i and b after the last round of optimization,$\mathrm{d}_i^{old} = \sum_{j=1}^{m} y_j\alpha_j k_{ji} + b^{old}$, which can be viewed as the distance of x_i to the hyperplane.

"Const" represents the part of the expression which is irrelevant to α_1 and α_2 and can be neglected when optimizing.

Then the problem is changed to:

$$\begin{aligned}\min_{\alpha 1,\alpha_2}\phi(\alpha_1,\alpha_2) = \frac{1}{2}k_{11}\alpha_1^2 + \frac{1}{2}k_{22}\alpha_2^2 + y_1y_2k_{12}\alpha_1\alpha_2 - (\alpha_1+\alpha_2) + c_1y_1\alpha_1 + c_2y_2\alpha_2 \\ s.t. \alpha_1y_1 + \alpha_2y_2 = -\sum_{i=3}^{m} y_i\alpha_i^{old} = z \\ 0 \le \alpha_i \le C, \quad i = 1,2,3\ldots,m\end{aligned}$$

where z is calculated using α's optimized value:

$$z = -\sum_{i=3}^{m} y_i\alpha_i^{\text{old}} = y_1\alpha_1^{\text{old}} + y_2\alpha_2^{\text{old}} \tag{23}$$

5.2 Single variable optimization without constraint

With $y_i^2 = 1$ and $\alpha_1y_1 + \alpha_2y_2 = z$ we have

$$\alpha_1 = \frac{z - y_2\alpha_2}{y_1} = y_1(z - y_2\alpha_2) \tag{24}$$

Substitute it into the problem to remove α_1 we get:

$$\phi(\alpha_2) = \frac{1}{2}k_{11}\left(z - y_2\alpha_2\right)^2 + \frac{1}{2}k_{22}\alpha_2^2 + y_2k_{12}\left(z - y_2\alpha_2\right)\alpha_2 + c_1\left(z - y_2\alpha_2\right) + c_2y_2\alpha_2 - y_1\left(z - y_2\alpha_2\right) - \alpha_2 \tag{25}$$

As it is quadratic, we can get the derivative:

$$\begin{aligned}\frac{d\phi}{d\alpha_2} &= -y_2k_{11}\left(z - y_2\alpha_2\right) + k_{22}\alpha_2 + y_2k_{12}\left(z - 2y_2\alpha_2\right) - c_1y_2 + c_2y_2 + y_1y_2 - 1 \\ &= \left(k_{11} + k_{22} - 2k_{12}\right)\alpha_2 - y_2k_{11}z + y_2k_{12}z - c_1y_2 + c_2y_2 + y_1y_2 - 1\end{aligned} \tag{26}$$

Then we have

$$\begin{aligned}\frac{\partial\phi}{\partial\alpha_2} = 0 &\Rightarrow \left(k_{11} + k_{22} - 2k_{12}\right)\alpha_2 = y_2k_{11}z - y_2k_{12}z + c_1y_2 - c_2y_2 - y_1y_2 + 1 \\ &\Rightarrow \left(k_{11} + k_{22} - 2k_{12}\right)\alpha_2 = y_2\left(k_{11}z - k_{12}z + c_1 - c_2 - y_1 + y_2\right)\end{aligned} \tag{27}$$

Putting (22) and (23) into (27) we get

$$\begin{aligned}\left(k_{11} + k_{22} - 2k_{12}\right)\alpha_2 =& y_2\big[k_{11}\left(y_1\alpha_1^{\text{old}} + y_2\alpha_2^{\text{old}}\right) - k_{12}\left(y_1\alpha_1^{\text{old}} + y_2\alpha_2^{\text{old}}\right) \\ &+ \left(d_1^{\text{old}} - b^{\text{old}} - y_1k_{11}\alpha_1^{\text{old}} - y_2k_{12}\alpha_2^{\text{old}}\right) \\ &- \left(d_2^{\text{old}} - b^{\text{old}} - y_1k_{12}\alpha_1^{\text{old}} - y_2k_{22}\alpha_2^{\text{old}}\right) - y_1 + y_2\big] \\ =& y_2\left[y_2k_{11}\alpha_2^{\text{old}} - 2y_2k_{12}\alpha_2^{\text{old}} + y_2k_{22}\alpha_2^{\text{old}} + \left(d_1^{\text{old}} - y_1\right) - \left(d_2^{\text{old}} - y_2\right)\right] \\ =& \left(k_{11} + k_{22} - 2k_{12}\right)\alpha_2^{\text{old}} + y_2\left(d_1^{\text{old}} - y_1\right) - y_2\left(d_2^{\text{old}} - y_2\right)\end{aligned} \tag{28}$$

We denote $\eta = k_{11} + k_{22} - 2k_{12}$, which is usually positive. We also denote $E_i = d_i - y_i$ (E_i is denoted as "error" by Platt (1998)). Then we get

$$\alpha_2^{\text{unclipped}} = \alpha_2^{\text{old}} + \frac{y_2\left(E_1 - E_2\right)}{\eta} \tag{29}$$

This is the optimized result for α_2 after this round of optimization without considering the constraint that $0 \le \alpha_1, \alpha_2 \le C$ (therefore"unclipped").

5.3 Clip $\alpha_2^{\text{unclipped}}$

Notice that the optimized result should satisfy $0 \le \alpha_1, \alpha_2 \le C$, so we need to clip our result.

We have $y_1\alpha_1 + y_2\alpha_2 = z$ and $y_1\alpha_1^{old} + y_2\alpha_2^{old} = z$. Therefore

$$\begin{aligned}&y_1\alpha_1 + y_2\alpha_2 = y_1\alpha_1^{\text{old}} + y_2\alpha_2^{\text{old}} \\ &\Rightarrow y_1\alpha_1 = y_1\alpha_1^{\text{old}} + y_2\left(\alpha_2^{\text{old}} - \alpha_2\right) \\ &\Rightarrow \alpha_1 = \alpha_1^{\text{old}} + y_1y_2\left(\alpha_2^{\text{old}} - \alpha_2\right)\end{aligned} \tag{30}$$

1. When $y_1 = y_2$, we have $y_1y_2 = 1$, then from (30) we have $\alpha_1 = \alpha_1^{\text{old}} + \alpha_2^{\text{old}} - \alpha_2$, so

$$0 \le \alpha_1 \le C \Leftrightarrow \alpha_1^{\text{old}} + \alpha_2^{\text{old}} - C \le \alpha_2 \le \alpha_1^{\text{old}} + \alpha_2^{\text{old}}$$

We also have $0 \leq \alpha_2 \leq C$, so

$$\max\left(0, \alpha_1^{\text{old}} + \alpha_2^{\text{old}} - C\right) \leq \alpha_2 \leq \min\left(C, \alpha_1^{\text{old}} + \alpha_2^{\text{old}}\right)$$

2. When $y_1 \neq y_2$, we have $y_1 y_2 = -1$, then from (30) we have $\alpha_1 = \alpha_1^{\text{old}} - \alpha_2^{\text{old}} + \alpha_2$, so

$$0 \leq \alpha_1 \leq C \Leftrightarrow \alpha_2^{\text{old}} - \alpha_1^{\text{old}} \leq \alpha_2 \leq \alpha_2^{\text{old}} - \alpha_1^{\text{old}} + C$$

We have $0 \leq \alpha_2 \leq C$, so

$$\max\left(0, \alpha_2^{\text{old}} - \alpha_1^{\text{old}}\right) \leq \alpha_2 \leq \min\left(C, \alpha_2^{\text{old}} - \alpha_1^{\text{old}} + C\right)$$

Therefore the clipping method for $\alpha_2^{\text{unclipped}}$ is:

$$\alpha_2^{\text{new}} = \begin{cases} H, \text{ if } \alpha_2^{\text{unclipped}} \geq H \\ \alpha_2^{\text{unclipped}}, \text{ if } L < \alpha_2^{\text{unclipped}} < H \\ L, \text{ if } \alpha_2^{\text{unclipped}} \leq L \end{cases}$$

where $L = \max\left(0, \alpha_1^{\text{old}} + \alpha_2^{\text{old}} - C\right), H = \min\left(C, \alpha_1^{\text{old}} + \alpha_2^{\text{old}}\right)$ if $y_1 y_2 = 1$; $L = \max\left(0, \alpha_2^{\text{old}} - \alpha_1^{\text{old}}\right), H = \min\left(C, \alpha_2^{\text{old}} - \alpha_1^{\text{old}} + C\right)$ if $y_1 y_2 = -1$.

(We choose $\alpha_2^{\text{new}} = H$ if $\alpha_2^{\text{unclipped}} \geq H$ because among all the values between L and H, H is the value that can minimize the quadratic objective function. We choose $\alpha_2^{\text{new}} = L$ if $\alpha_2^{\text{unclipped}} \leq L$ because among all the values between L and H, L is the value that can minimize the quadratic objective function.)

5.4 Calculate α_1^{new}

From equation (30) we calculate α_1^{new} :

$$\alpha_1^{\text{new}} = \alpha_1^{\text{old}} + y_1 y_2 \left(\alpha_2^{\text{old}} - \alpha_2^{\text{new}}\right)$$

The above 3 steps obtain the optimal solution $\alpha_1^{new}, \alpha_2^{new}$ of the QP subproblem with two variables by analytical method, saving computational time compared to numerical optimization methods. This round of optimization updates $\alpha_1^{\text{old}}, \alpha_2^{\text{old}}$ to $\alpha_1^{\text{new}}, \alpha_2^{\text{new}}$.

5.5 Updating b

(1) If $0 < \alpha_1^{\text{new}} < C$

Using KKT condition we have $y_1 d_1 = y_1 \left(\sum_{j=1}^{m} y_j k_{j1} \alpha_j + b\right) = 1$ to update b, we now have

$$b_1^{\text{new}} = y_1 - \sum_{j=3}^{m} y_j k_{j1} \alpha_j^{\text{old}} - y_1 k_{11} \alpha_1^{\text{new}} - y_2 k_{21} \alpha_2^{\text{new}}$$

and because $E_1^{old} = d_1^{old} - y_1 = \sum_{j=1}^{m} y_j k_{j1} \alpha_j^{\text{old}} + b^{\text{old}} - y_1$, substitute it to the above equation

$$\begin{aligned} b_1^{\text{new}} &= -E_1^{\text{old}} + y_1 k_{11} \alpha_1^{\text{old}} + y_2 k_{21} \alpha_2^{\text{old}} - y_1 k_{11} \alpha_1^{\text{new}} - y_2 k_{21} \alpha_2^{\text{new}} + b^{\text{old}} \\ &= -E_1^{\text{old}} - y_1 k_{11} \left(\alpha_1^{\text{new}} - \alpha_1^{\text{old}}\right) - y_2 k_{21} \left(\alpha_2^{\text{new}} - \alpha_2^{\text{old}}\right) + b^{\text{old}} \end{aligned}$$

After updating b, we have $(\boldsymbol{x}_1, y_1)$ satisfying KKT conditions. And at the same time, $(\boldsymbol{x}_2, y_2)$ also satisfies KKT conditions (we don't prove this here), so we only need to update b like this if $0 < \alpha_1^{\text{new}} < C$.

(2) If $0 < \alpha_2^{\text{new}} < C$

We update b according to $y_2 d_2 = y_2 \left(\sum_{j=1}^{m} y_j \alpha_j k_{j2} + b \right) = 1$. Likewise, we have

$$b_2^{\text{new}} = -E_2^{\text{old}} - y_1 k_{12} \left(\alpha_1^{\text{new}} - \alpha_1^{\text{old}} \right) - y_2 k_{22} \left(\alpha_2^{\text{new}} - \alpha_2^{\text{old}} \right) + b^{\text{old}}$$

(3) $0 < \alpha_1^{\text{new}} < C$ and $0 < \alpha_2^{\text{new}} < C$

Then the calculated $b_1^{new} = b_2^{new}$.

(4) If $\alpha_1^{\text{new}}, \alpha_1^{\text{new}} \in 0, C$ and $L \neq H$:

Then values between b_1^{new} and b_2^{new} can all satisfy KKT conditions (we don't prove it here) and we let $b^{new} = \frac{1}{2}(b_1^{new} + b_2^{new})$.

(5) If $\alpha_1^{\text{new}}, \alpha_1^{\text{new}} \in 0, C$ and $L = H$:

We skip this step of the subquestion.

5.6 Updating d_i and E_i

$$\begin{aligned} d_i^{new} &= \sum_{j=1}^{m} y_j k_{ji} \alpha_j^{new} + b^{new} \\ &= \sum_{j \in S} y_j k_{ji} \alpha_j^{new} + b^{new} \end{aligned}$$

According to the definition, $E_i = d_i - y_i$, we have

$$E_i^{new} = d_i^{new} - y_i = \sum_{j \in S} y_j k_{ji} \alpha_j^{new} + b^{new} - y_i$$

Platt 1999 gives another formula:

We have

$$E_i^{new} = d_i^{new} - y_i = \sum_{j=3}^{m} y_j k_{ji} \alpha_j^{\text{old}} + y_1 k_{1i} \alpha_1^{new} + y_2 k_{2i} \alpha_2^{new} + b^{new} - y_i$$

and

$$\begin{aligned} E_i^{\text{old}} = d_i^{\text{old}} - y_i &= \sum_{j=1}^{m} y_j k_{ji} \alpha_j^{\text{old}} + b^{old} - y_i \\ &= \sum_{j=3}^{m} y_j k_{ji} \alpha_j^{\text{old}} + y_1 k_{1i} \alpha_1^{\text{old}} + y_2 k_{2i} \alpha_2^{\text{old}} + b^{\text{old}} - y_i \end{aligned}$$

Therefore we have

$$E_i^{\text{new}} = E_i^{\text{old}} + y_1 k_{1i} \left(\alpha_1^{\text{new}} - \alpha_1^{\text{old}} \right) + y_2 k_{2i} \left(\alpha_2^{\text{new}} - \alpha_2^{\text{old}} \right) + b^{\text{new}} - b^{\text{old}} \tag{31}$$

This formula has an edge that we can save time for calculation if we store all samples' E_i's.

5.7 Stopping condition

According to Platt (1999), The stopping condition of SMO is:

$$\begin{aligned} \alpha_i = 0 &\Rightarrow y_i d_i \geq 1 \\ 0 < \alpha_i < C &\Rightarrow y_i d_i = 1 \\ \alpha_i = C &\Rightarrow y_i d_i \leq 1 \end{aligned} \tag{32}$$

which can be derived from the KKT condition (21).

5.8 SMO process conclusion

(1) Initialize $\boldsymbol{\alpha} = \mathbf{0}, b = 0$ and give accuracy ϵ, usually 10^{-3} to 10^{-2}.

(2) Select α_1 and α_2 (the way to select appropriate α_i's is not discussed in this note, you can refer to Platt 1998 for more, but the general rule is that you choose the two α_i's that violate the KKT conditions most), and solve the QP subproblem; update α_1 and α_2; then update b and E_i.

(3) See if the stopping condition is met within ϵ. If met, then go to (4). If not, then go to (2).

(4) Get the best solution to the problem.

5.9 Example

We solve the same problem in section 2.4, but we use SMO method this time.

We first choose α_2 and α_3 to update.

We have $k_{11} = 5^2+6^2 = 61, k_{12} = k_{21} = 5*4+6*4 = 44, k_{22} = 32, k_{13} = k_{31} = k_{23} = k_{32} = k_{33} = 0$.

We initialize $\alpha_1 = \alpha_2 = \alpha_3 = 0, b^{old} = 0$. Then

$$d_1 = \sum_{j=1}^{m} y_j \alpha_j k_{j1} + b^{old} = 0$$
$$d_2 = \sum_{j=1}^{m} y_j \alpha_j k_{j2} + b^{old} = 0$$
$$d_3 = \sum_{j=1}^{m} y_j \alpha_j k_{j3} + b^{old} = 0$$
$$\eta = k_{22} + k_{33} - 2k_{23} = 32$$
$$E_1 = d_1 - y_1 = 0 - 1 = -1$$
$$E_2 = d_2 - y_2 = 0 - 1 = -1$$
$$E_3 = d_3 - y_3 = 0 - (-1) = 1$$
$$\alpha_3 = \alpha_3^{old} + \frac{y_3(E_2 - E_3)}{\eta} = \frac{-1(-1-1)}{32} = \frac{1}{16}$$
$$\alpha_2 = \alpha_2^{old} + y_1 y_2(\alpha_2^{old} - \alpha_2^{new}) = 0 + (-1)(0 - \frac{1}{16}) = \frac{1}{16}$$
$$b_2^{new} = -E_2 - y_2 k_{22}(\alpha_2^{new} - \alpha_2^{old}) - y_3 k_{32}(\alpha_3^{new} - \alpha_3^{old}) + b^{old} = -(-1) - 1 * 32 * (\frac{1}{16} - 0) -$$
$$(-1) * 0 * (\frac{1}{16} - 0) + 0 = -1$$
$$E_2^{new} = E_2^{old} + y_2 k_{22}(\alpha_2^{new} - \alpha_2^{old}) + y_3 k_{32}(\alpha_2^{new} - \alpha_2^{old}) + b_2^{new} - b^{old} = -1 + 32 * \frac{1}{16} - 0 * \frac{1}{16} - 1 = 0$$
$$E_3^{new} = E_2^{old} + y_2 k_{23}(\alpha_2^{new} - \alpha_2^{old}) + y_3 k_{33}(\alpha_3^{new} - \alpha_3^{old}) + b_2 - b^{old} = 1 + 0 + 0 - 1 + 0 = 0$$

We check the KKT conditions:

$$d_1^{new} = \sum_{j=1}^{m} y_j \alpha_j^{new} k_{j1} + b_2^{new} = y_2 \alpha_2^{new} k_{21} + b_2^{new} = 1 * \frac{1}{16} * 44 - 1 = \frac{7}{4}$$
$$d_2^{new} = \sum_{j=1}^{m} y_j \alpha_j^{new} k_{j2} + b_2^{new} = y_2 \alpha_2^{new} k_{22} + b_2^{new} = 1 * \frac{1}{16} * 32 - 1 = 1$$
$$d_3^{new} = \sum_{j=1}^{m} y_j \alpha_j^{new} k_{j3} + b_2^{new} = b_2^{new} = -1$$

Then we have

$$y_1 d_1^{new} = \frac{7}{4} > 1, \alpha_1 = 0$$
$$y_2 d_2^{new} = 1, \alpha_2 > 0$$
$$y_3 d_3^{new} = 1, \alpha_3 > 0$$

This satisfies the KKT conditions. Therefore we shall stop and conclude that $\alpha_1 = 0, \alpha_2 = \alpha_3 = \frac{1}{16}$, which is the same as the result in Section 2.4.

6 Strengths and limitations of SVM

6.1 Strengths

1. It can be applied to nonlinear classification.

2. With only a few supporting vectors deciding the result, most useless samples can be removed and this increases efficiency.

6.2 Limitations

1. It is not efficient on large numbers of samples as SVM solves quadratic programming problems. There are improvements like SMO method.

2. SVM can't deal with multi-classification problems. This can be dealt with by using several SVMs in one problem.

3. There is no one-for-all methods for nonlinear classification (choosing kernel functions).

SVR

1 Overview of SVR

SVR is a way of regression, but it differs from traditional regression in that it allows a difference between $f(\boldsymbol{x})$ and y to at most ϵ and will not penalize the difference within ϵ. Only when the difference between $f(\boldsymbol{x})$ and y is larger than ϵ will we calculate the loss. In the graph below, the sample within the margin is recognized as being predicted correctly.

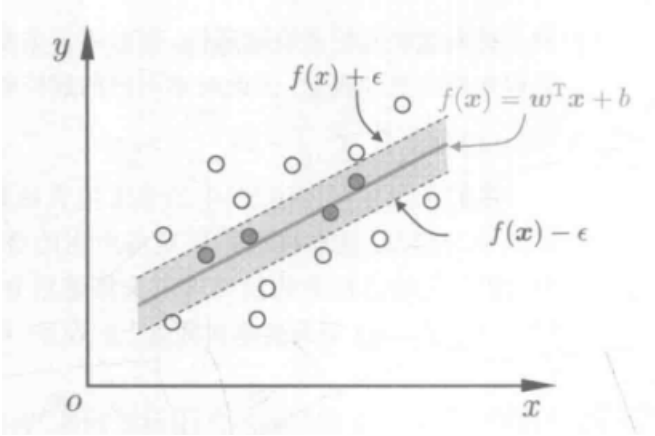

Figure 6: SVR

The problem: We are given a sample set $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n)\}, y_i \in R$ and asked to find a hyperplane $\boldsymbol{w}^T\boldsymbol{x} + b = 0$ where $||\boldsymbol{w}|| = \sqrt{w_1^2 + w_2^2 + ... + w_n^2}$ so that $f(\boldsymbol{x})$ and y are as close as possible.

2 Setting up and Solving SVR problem

2.1 Setting up SVR problem

Then the SVR problem is:

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \ell_\epsilon \left(f\left(\boldsymbol{x}_i\right) - y_i\right),$$

where C is regularization constant, ℓ_ϵ is ϵ-insensitive loss function shown in figure 6.

$$\ell_\epsilon(z) = \begin{cases} 0, & \text{if } |z| \leq \epsilon \\ |z| - \epsilon, & \text{otherwise} \end{cases}$$

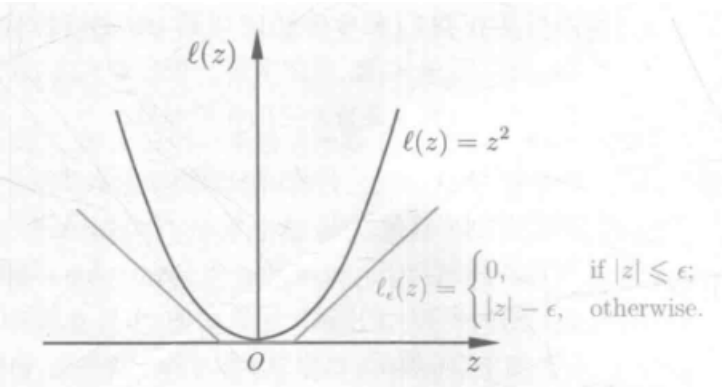

Figure 7: ϵ-insensitive loss function

We have the term $\frac{1}{2}\|\boldsymbol{w}\|^2$ in the expression we want to minimize to facilitate the following calculation and this form can be seen as L2 regularization.

We include slack variables ξ_i and $\hat{\xi}_i$ ($\xi_i = max\{f(\boldsymbol{x}_i) - y_i - \epsilon, 0\}, \hat{\xi}_i = max\{y_i - f(\boldsymbol{x}_i) - \epsilon, 0\}$) and rewrite as:

$$\begin{aligned} &\min_{\boldsymbol{w},b,\xi_i,\hat{\xi}_i} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\left(\xi_i + \hat{\xi}_i\right) \\ &\text{s.t. } f(\boldsymbol{x}_i) - y_i \leq \epsilon + \xi_i, \\ &y_i - f(\boldsymbol{x}_i) \leq \epsilon + \hat{\xi}_i, \\ &\xi_i \geq 0, \hat{\xi}_i \geq 0, i = 1, 2, \ldots, m. \end{aligned}$$

We include Lagrangian multiplier $\mu_i \geq 0, \hat{\mu}_i \geq 0, \alpha_i \geq 0, \hat{\alpha}_i \geq 0$, using Lagrangian multiplier method we have the Lagrangian function:

$$\begin{aligned} &L(\boldsymbol{w}, b, \boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}}, \boldsymbol{\xi}, \hat{\boldsymbol{\xi}}, \boldsymbol{\mu}, \hat{\boldsymbol{\mu}}) \\ &= \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\left(\xi_i + \hat{\xi}_i\right) - \sum_{i=1}^{m}\mu_i\xi_i - \sum_{i=1}^{m}\hat{\mu}_i\hat{\xi}_i \\ &+ \sum_{i=1}^{m}\alpha_i\left(f(\boldsymbol{x}_i) - y_i - \epsilon - \xi_i\right) + \sum_{i=1}^{m}\hat{\alpha}_i\left(y_i - f(\boldsymbol{x}_i) - \epsilon - \hat{\xi}_i\right). \end{aligned}$$

We insert $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$ and make the partial derivatives of $L(\boldsymbol{w}, b, \boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}}, \boldsymbol{\xi}, \hat{\boldsymbol{\xi}}, \boldsymbol{\mu}, \hat{\boldsymbol{\mu}})$ to $\boldsymbol{w}, b, \xi_i \quad \hat{\xi}_i$ equal 0 we have

$$\boldsymbol{w} = \sum_{i=1}^{m}(\hat{\alpha}_i - \alpha_i)\boldsymbol{x}_i \tag{33}$$

$$0 = \sum_{i=1}^{m}(\hat{\alpha}_i - \alpha_i) \tag{34}$$

$$C = \alpha_i + \mu_i \tag{35}$$

$$C = \hat{\alpha}_i + \hat{\mu}_i \tag{36}$$

We insert the result to the Lagrangian function, and get the dual problem of SVR.

$$\begin{aligned}
&\max_{\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}}} \sum_{i=1}^{m} y_i \left(\hat{\alpha}_i - \alpha_i\right) - \epsilon \left(\hat{\alpha}_i + \alpha_i\right) \\
&-\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \left(\hat{\alpha}_i - \alpha_i\right) \left(\hat{\alpha}_j - \alpha_j\right) \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j \\
&\text{s.t.} \ \sum_{i=1}^{m} \left(\hat{\alpha}_i - \alpha_i\right) = 0, \\
&0 \leq \alpha_i, \hat{\alpha}_i \leq C.
\end{aligned}$$

This should satisfy the KKT conditions. That is:

$$\begin{cases}
\alpha_i \left(f\left(x_i\right) - y_i - \epsilon - \xi_i\right) = 0 \\
\hat{\alpha}_i \left(y_i - f\left(x_i\right) - \epsilon - \hat{\xi}_i\right) = 0 \\
\alpha_i \hat{\alpha}_i = 0, \xi_i \hat{\xi}_i = 0 \\
\left(C - \alpha_i\right) \xi_i = 0, \left(C - \hat{\alpha}_i\right) \hat{\xi}_i = 0
\end{cases}$$

We can know that if a sample is above the upper boundary, $\hat{\xi}_i = y_i - f\left(x_i\right) - \epsilon > 0$, therefore $\hat{\alpha}_i = C$ because $\left(C - \hat{\alpha}_i\right) \hat{\xi}_i = 0$. And We have $f(x_i) < y_i$, so $f(x_i) - y_i - \epsilon < 0$, so $\xi_i = max\{f(x_i) - y_i - \epsilon, 0\} = 0$. Then we also have $f(x_i) - y_i - \epsilon - \xi_i < 0$, therefore $\alpha_i = 0$ because $\alpha_i \left(f\left(x_i\right) - y_i - \epsilon - \xi_i\right) = 0$.

We can conduct similar analysis for other cases and we have the following relationship between values of α's and ξ's and position of sample.

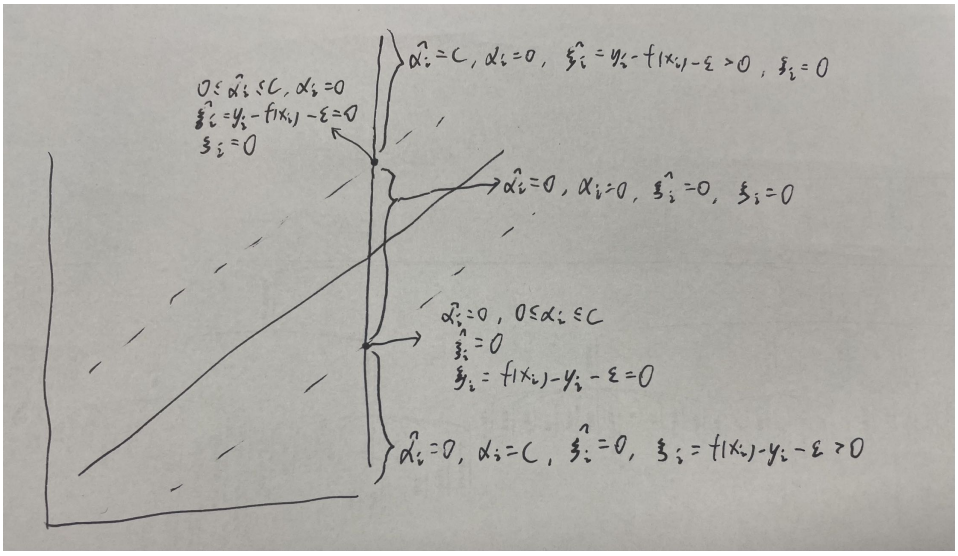

Figure 8: values of α's and ξ's and corresponding position of samples

2.2 Solving SVR problem

We insert (33) into $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$ and the solution to SVR is like

$$f(x) = \sum_{i=1}^{m} (\hat{\alpha}_i - \alpha_i)\, \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x} + b. \tag{37}$$

Those samples which make $\hat{\alpha}_i - \alpha_i \neq 0$ in (37) are the supporting vectors and they fall outside the -margin.

From the KKT conditions we know that for every sample $(\boldsymbol{x}_i, y_i)$ we have $(C - \alpha_i)\xi_i = 0$ and $\alpha_i \left(f\left(x_i\right) - y_i - \epsilon - \xi_i\right) = 0$. So after we get α_i, if $0 < \alpha_k < C$, then it must be that $\xi_k = 0$, then we have $f(\boldsymbol{x}_k) = y_k + \epsilon$ and by using (37) we have

$$b = y_k + \epsilon - \sum_{i=1}^{m} (\hat{\alpha}_i - \alpha_i)\, \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_k.$$

which is equivalent to

$$b = y_k + \epsilon - \sum_{i \in S} (\hat{\alpha}_i - \alpha_i)\, \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_k.$$

where $S = \{i | \hat{\alpha}_i - \alpha_i \neq 0, i = 1, 2, ...m\}$ is the set of subscripts of supporting vectors.

Again, in reality, we usually select all samples satisfying $0 < \alpha_i < C$, calculate the b's one by one and take the average.

2.3 Adding kernel function

If we consider $\phi(x)$, the x after mapping, then (33) will be like

$$\boldsymbol{w} = \sum_{i=1}^{m} (\hat{\alpha}_i - \alpha_i)\, \phi\left(\boldsymbol{x}_i\right). \tag{38}$$

Putting (38) into (8) then SVR can be represented as

$$f(x) = \sum_{i=1}^{m} (\hat{\alpha}_i - \alpha_i)\, \kappa(\boldsymbol{x}, \boldsymbol{x}_i) + b \tag{39}$$

where $\kappa(\boldsymbol{x_i}, \boldsymbol{x}_j) = \phi(\boldsymbol{x_i})^T \phi(\boldsymbol{x}_j)$ is kernel function.

Finally, we can again use methods like SMO to calculate $\hat{\alpha}_i$ and α_i.

3 Example

We use the same dataset as the example before:

Table 3: Player information

Name	**# of MVPs**	**# of championships**
Jordan	5	6
James	4	4
Justin	0	0

We want to know whether greater personal ability (we use of MVPs to represent personal ability) can help a player win more championships. The dual problem is

$$\begin{aligned}
&\max_{\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}}} \sum_{i=1}^{m} y_i \left(\hat{\alpha}_i - \alpha_i\right) - \epsilon \left(\hat{\alpha}_i + \alpha_i\right) \\
&-\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \left(\hat{\alpha}_i - \alpha_i\right) \left(\hat{\alpha}_j - \alpha_j\right) \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j \\
&\text{s.t. } \sum_{i=1}^{m} \left(\hat{\alpha}_i - \alpha_i\right) = 0, \\
&0 \leq \alpha_i, \hat{\alpha}_i \leq C.
\end{aligned}$$

We assume $\epsilon = 0.1$ and $C = 1$ (which are default values in the SVR function in Python sklearn library). And we have $x_1 = 5, x_2 = 4, x_3 = 0, y_1 = 6, y_2 = 4, y_3 = 0$. Then the problem is

$$\begin{aligned}
&\max_{\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}}} 6(\hat{\alpha_1} - \alpha_1) - 0.1(\hat{\alpha_1} + \alpha_1) + 4(\hat{\alpha_2} - \alpha_2) - 0.1(\hat{\alpha_2} + \alpha_2) - 0.1(\hat{\alpha_3} + \alpha_3) \\
&-\frac{1}{2}[25(\hat{\alpha_1} - \alpha_1)^2 + 40(\hat{\alpha_1} - \alpha_1)(\hat{\alpha_2} - \alpha_2) + 16(\hat{\alpha_2} - \alpha_2)^2] \\
&\text{s.t. } \hat{\alpha_1} - \alpha_1 + \hat{\alpha_2} - \alpha_2 + \hat{\alpha_3} - \alpha_3 = 0, \\
&0 \leq \alpha_i, \hat{\alpha_i} \leq 1, i = 1, 2, 3.
\end{aligned}$$

We can set the Lagrangian function:

$$\begin{aligned}
L(\boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}}, \mu, \boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) = &-\frac{1}{2}[25(\hat{\alpha_1}-\alpha_1)^2+40(\hat{\alpha_1}-\alpha_1)(\hat{\alpha_2}-\alpha_2)+16(\hat{\alpha_2}-\alpha_2)^2]+5.9\hat{\alpha_1}-6.1\alpha_1+3.9\hat{\alpha_2}-4.1\alpha_2-0.1\hat{\alpha_3}-0.1\alpha_3 \\
&+ \mu(\hat{\alpha_1} - \alpha_1 + \hat{\alpha_2} - \alpha_2 + \hat{\alpha_3} - \alpha_3) - \textstyle\sum_{i=1}^{3} \left(\lambda_{i1}\alpha_i + \lambda_{i2}(1 - \alpha_i)\right) - \sum_{i=1}^{3} \left(\hat{\lambda_{i1}}\hat{\alpha_i} + \hat{\lambda_{i2}}(1 - \hat{\alpha_i})\right)
\end{aligned}$$

where $\mu, \boldsymbol{\lambda} = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \\ \lambda_{31} & \lambda_{32} \end{bmatrix}$ $\hat{\boldsymbol{\lambda}} = \begin{bmatrix} \hat{\lambda_{11}} & \hat{\lambda_{12}} \\ \hat{\lambda_{21}} & \hat{\lambda_{22}} \\ \hat{\lambda_{31}} & \hat{\lambda_{32}} \end{bmatrix}$ are Lagrangian multipliers.

We have 19 variables and we can get the following 19 equations:

$$\begin{cases}
\frac{\partial L}{\partial \hat{\alpha_i}} = 0, i = 1, 2, 3 \\
\frac{\partial L}{\partial \alpha_i} = 0, i = 1, 2, 3 \\
\frac{\partial L}{\partial \mu} = 0 \\
\lambda_{i1}\alpha_i = 0, i = 1, 2, 3 \\
\lambda_{i2}(1 - \alpha_i) = 0, i = 1, 2, 3 \\
\hat{\lambda_{i1}}\hat{\alpha_i} = 0, i = 1, 2, 3 \\
\hat{\lambda_{i2}}(1 - \hat{\alpha_i}) = 0, i = 1, 2, 3
\end{cases}$$

This will be a very complex equation set to solve so we instead use Excel solver and find that the optimal solution is $\hat{\alpha_1} = 1, \alpha_1 = 0, \hat{\alpha_2} = 0, \alpha_2 = 1, \hat{\alpha_3} = 0, \alpha_3 = 0$.

Then we know that sample 1 should be above or at the upper boundary. Sample 2 should be below or at the lower boundary. Sample 3 should be at or within the boundary.

Then $w = \sum_{i=1}^{m} (\hat{\alpha}_i - \alpha_i) \boldsymbol{x}_i = 1$

With the slope of the hyperplane equal to 1, we find that Sample 2 and Sample 3 must be both at the same relative position with respect to the hyperplane and boundaries because the slope of the line connecting Sample 2 and Sample 3 is also 1. Then we know that Sample 2 and Sample 3 must be both at lower boundary. And in this case $b = 0.1$.

Therefore the hyperplane is $f(x) = x + 0.1$. We find that higher personal ability can help a player win more championships.

4 Strengths and limitations

4.1 Strengths

1. It is insensitive to outliers, and therefore it usually predicts better than linear regression model.

2. Its implementation is easy.

4.2 Limitations

1. Time complexity of SVR can be very large and therefore not suitable for large datasets.

2. SVR can perform poorly on datasets with a larger number of features than samples. (same as linear regression)

Reference

Platt, J.C. (1998) 'Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines', *MIT Press*, Boston.

Platt, J.C. (1999) 'Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods'. *Advances in Large Margin Classifiers*, 10(3), pp. 61–74.

Zhihua, Z. (2016) *Machine Learning*. Tsinghua University Press.

Russell, S.J. and Norvig, P. (2009) *Artificial Intelligence: A Modern Approach 3rd Edition*. Prentice Hall.