6.1 What’s Support Vector Machine?

Support Vector Machine (SVM) is commonly referred to as **a binary classification model**. Its basic model is defined as the **linear classifier with the largest spacing in the feature space**, which can be maximized the spacing using a convex quadratic programming problem.

6.2 Origin of SVM

The origin of SVM is Logistic Regression. Given data points that belong to two different classes, find a linear classifier that divides the data into two classes. But there is a big difference from Logistic Regression, which change labels from y = 0, 1 (LR) to y = -1, 1 (SVM).

$$\theta^T x = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n \tag{1}$$

$$\theta^T x = w^T x + b \tag{2}$$

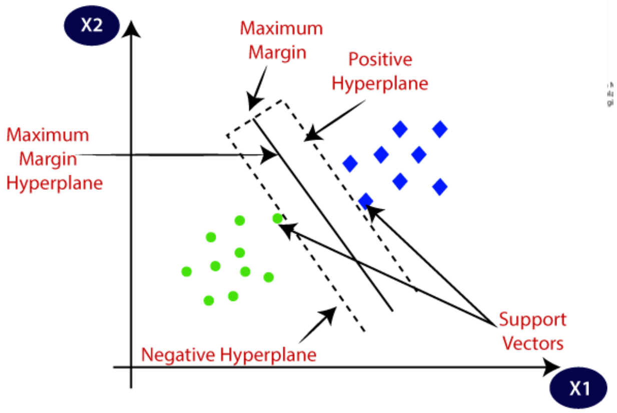
$$h_\theta(x) = g(\theta^T x) = g(w^T x + b) \tag{3}$$

And the sign() function for SVM is:

$$g(z) = \begin{cases} 1, & z > 0 \\ -1, & z < 0 \end{cases} \tag{4}$$

6.3 Details of SVM

As shown in the figure below, there are two different kinds of data on a two-dimensional plane. The two types of data are separated by a line or a hyperplane (n dimension). The data points on one side of the hyperplane correspond to y = 1 and the data points on the other side correspond to y = -1.

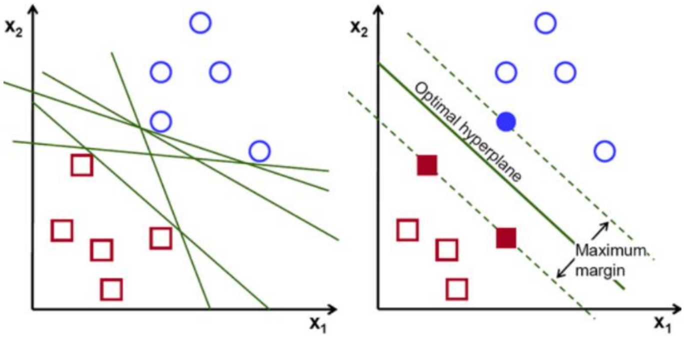


The hyperplane f(x) = $w^Tx + b$ can be represented by the classification function.

- f(x) = 0, x is the data point on the hyperplane;
- f(x) > 0, x is the data point upper the hyperplane corresponding to y = 1;
- f(x) < 0, x is the data point below the hyperplane corresponding to y = -1.

6.3.1 How to Find the Best Hyperplane?

We can lots of hyperplanes to classify different data points, but which one is the best? The answer is that one with **the maximum margin**. Why is it? Let's assume the data points on the both sides are landmines, we can avoid them only if the closest landmine is as far from the hyperplane as possible. And with a maximum margin, we can allow as many people as possible to avoid landmines. Generally, we can use functional margin and **geometric margin as the benchmark for maximum margin.**



6.3.1.1 Functional Margin (FM)

When the hyperplane $w^Tx+b = 0$ is determined, $|w^Tx +b|$ can represent the distance from data point x to hyperplane, and the classification can be judged by observing whether the symbols of w^Tx+b are consistent with those of y. In other words, **the functional distance $y(w^Tx+b)$ is used to indicate the classification correctness and classification confidence**.

$$\begin{aligned} f(x_i) = w^T x_i + b > 0 \;\; &\rightarrow \;\; y_i > 0 \\ f(x_i) = w^T x_i + b < 0 \;\; &\rightarrow \;\; y_i < 0 \end{aligned} \tag{5}$$

- **classification correctness**: if $y(w^Tx+b) > 0$, the classification is correct, which means the symbol of y always keep same with w^Tx+b; otherwise, it is wrong.
- **classification confidence**: the higher the value of $y(w^Tx+b)$, the greater the classification confidence, and vice versa.

$$\hat{\gamma_i} = y_i(w^T x_i + b) = y_i \cdot f(x_i) \tag{6}$$

w: normal vector of hyperplane
b: intercept
But there's a problem. If W and b shrink or magnify m times at the same time, and the hyperplane doesn't change, but the functional margin changes. In fact, we can add some constraints to normal vector w, leading to **real definition of the distance from a point to the hyperplane**: geometric margin.

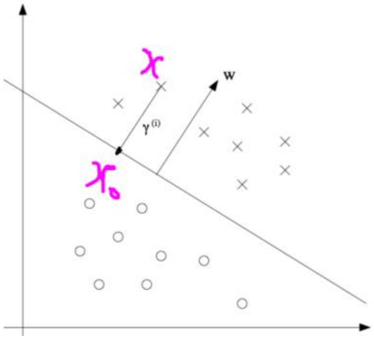
6.3.1.2 Geometric Margin (GM)

Basically, we can call functional margin as the distance from a point to the hyperplane, but we often technically define geometric margin as the real distance from a point to the hyperplane. The geometric margin equals functional margin divided by ||w||. the distance is the numerator of that formula, not normalized.

$$\gamma_i = \frac{y_i(w^T x_i + b)}{||w||} = \frac{y_i \cdot f(x_i)}{||w||} \tag{7}$$

||w||: the length of normal vector.

6.3.1.3 Derivation of FM and GM



Method 1:
In the figure above, there is a hyperplane, a normal vector w, a point x and its project point x_0:

$$x = x_0 + \gamma \frac{w}{||w||} \tag{8}$$

||w||: L2 norm, length of normal; $\frac{w}{||w||}$: unit vector; γ: distance from x to x_0

$$w^T x = w^T x_0 + \gamma \frac{w^T w}{||w||} \tag{9}$$

Because $w^T x_0 + b = 0$ and $w^T w = ||w||^2$, we get:

$$\gamma = \frac{w^T x + b}{||w||} = \frac{f(x)}{||w||} \tag{10}$$

Method 2:

$$\vec{w} \cdot \overrightarrow{x_0 x} = ||w|| \cdot ||x_0 x|| \cdot cos0 = ||w|| \cdot \gamma \tag{11}$$

$$\begin{aligned}\vec{w}\cdot\overrightarrow{x_0x} &= w^1(x^1 - x_0^1) + w^2(x^2 - x_0^2) + \cdots + w^n(x^n - x_0^n)\\ &= (w^1x^1 + w^2x^2 + \cdots + w^nx^n) - (w^1x_0^1 + w^1x_0^2 + \cdots + w^1x_0^n)\\ &= w^Tx - (-b)\\ &= w^Tx + b\end{aligned} \tag{12}$$

Combining Eq. (11) and Eq. (12):

$$||w||\cdot\gamma = w^Tx + b \tag{13}$$

$$\gamma = \frac{w^Tx + b}{||w||} = \frac{f(x)}{||w||} \tag{14}$$

Because $w^Tx + b$ may be positive or negative, therefore we get geometric margin by multiplied by y:

$$|\gamma| = y\cdot\frac{w^Tx + b}{||w||} = \frac{y\cdot f(x)}{||w||} = \frac{\hat{\gamma}}{||w||} \tag{15}$$

6.3.2 Objective Function

Based on the best hyperplane should be with the maximum margin (measured by Geometric Margin), we can consider the objective function as:

$$\max_{\omega,b}\left\{\frac{1}{||w||}\min_i[y_i(w^Tx_i + b)]\right\} \tag{16}$$

There are two points we should take care of:

- $\min_i[y_i(w^Tx_i + b)]$: to find the closest data point (x_i) to hyperplane
- $\mathrm{argmax}_{\omega,b}$: to find w and b which can maximize the distance from x_i to hyperplane

In order to simplify Eq. (16), we generally set $\hat{\gamma} = y(w^Tx + b) = 1$, so $y_i(w^Tx_i + b) \geq 1$, which means $\min_i[y_i(w^Tx_i + b)] = 1$, therefore we get a new objective function with a limited condition:

$$\max_{\omega,b}\left\{\frac{1}{||w||}\right\}, \qquad s.t.\, y_i(w^Tx_i + b) \geq 1 \tag{17}$$

Converting the problem with solving maximum value in Eq. (17) to solve minimum value in Eq. (18):

$$\min_{\omega,b}\left\{\frac{1}{2}||w||^2\right\}, \qquad s.t.\, y_i(w^Tx_i + b) \geq 1 \tag{18}$$

6.3.3 Solve with Lagrange Multiplier Method

Build Lagrange function:

$$L(w,b,\alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n}\alpha_i[y_i(w^Tx_i + b) - 1] \tag{19}$$

Because we introduce Lagrange multiplier α, our objective function should be optimized as below:

$$\min_{\omega,b}\max_{\alpha\geq 0} L(w,b,\alpha) \tag{20}$$

Dual problem:

$$\min_{\omega,b}\max_{\alpha\geq 0} L(w,b,\alpha) \rightarrow \max_{\alpha\geq 0}\min_{\omega,b} L(w,b,\alpha) \tag{21}$$

Partial derivative of Lagrange function:

1) min{L(w, b , α)} based on w and b:

$$\begin{cases} \frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i \\ \frac{\partial L}{\partial b} = 0 \rightarrow 0 = \sum_{i=1}^{n} \alpha_i y_i x_i \end{cases} \tag{22}$$

Plug Eq. (22) into Eq. (19):

$$\begin{aligned} L(w,b,\alpha) &= \frac{1}{2}w^T w - \sum_{i=1}^{n} \alpha_i y_i w^T x_i + \sum_{i=1}^{n} \alpha_i y_i b + \sum_{i=1}^{n} \alpha_i \\ &= \frac{1}{2}\left(\sum_{i=1}^{n} \alpha_i y_i x_i\right)^T \sum_{i=1}^{n} \alpha_i y_i x_i - \sum_{i=1}^{n} \alpha_i y_i \left(\sum_{i=1}^{n} \alpha_i y_i x_i\right)^T x_i + \sum_{i=1}^{n} \alpha_i y_i b + \sum_{i=1}^{n} \alpha_i \\ &= -\frac{1}{2}\left(\sum_{i=1}^{n} \alpha_i y_i x_i\right)^T \sum_{i=1}^{n} \alpha_i y_i x_i + \sum_{i=1}^{n} \alpha_i \\ &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \end{aligned} \tag{23}$$

2) max{L(w, b , α)} based on α:

$$\max_{\alpha\geq 0} L(w,b,\alpha) = \max_{\alpha\geq 0} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{24}$$

$$\begin{cases} s(x,\alpha) = \min_{\alpha\geq 0} \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^{n} \alpha_i \\ s.t. \sum_{i=1}^{n} \alpha_i y_i = 0, \ \ \alpha_i \geq 0 \end{cases} \tag{25}$$

$$\begin{cases} \frac{\partial L}{\partial \alpha_1} = 0 \\ \frac{\partial L}{\partial \alpha_2} = 0 \\ \vdots \\ \frac{\partial L}{\partial \alpha_n} = 0 \end{cases} \tag{26}$$

3) Compute w and b based on α from 2):

$$\begin{cases} w^* = \sum_{i=1}^{n} \alpha_i y_i x_i \\ b^* = y - w^* x = -\dfrac{[\max(\mathrm{i}): y_i = -1]\, {w^*}^T x_i + [\min(i): y_i = 1]\, {w^*}^T x_i}{2} \end{cases} \tag{27}$$

Note:

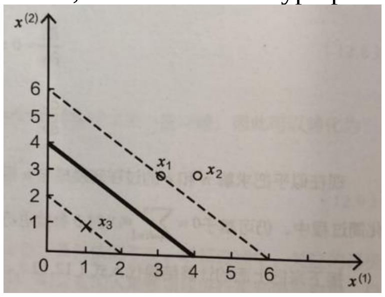
- We just use support vector when we compute b because **the Lagrange multipliers for non-support vector are always 0**. In addition, when we use the hyperplane to predict some data point, we can also just use the support vector.
- Why Lagrange multipliers for non-support vector = 0? Because the objective function, we want to maximize it:

$$\max_{\alpha \geq 0} L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i [y_i(w^T x_i + b) - 1]$$

  For support vector, the yellow part = 0, but non-support vector, it > 0, if we also want to maximize the whole formula ($\alpha_i \geq 0$), so we have to make $\alpha_i = 0$.
- That's all for SVM when data points are in **a linear separable situation**.

6.3.3 Example for SVM

Assume there are only three data points including positive sample X$_1$(3, 3), X$_2$(4, 3) and negative sample X$_3$(1, 1), which are shown below, what is the best hyperplane?



Solution:

Given positive sample X$_1$(3, 3), X$_2$(4, 3) ⟶ y$_1$ = 1, y$_2$ = 1 and negative sample X$_3$(1, 1) ⟶ y$_3$ = 1

1) Build optimization model using Eq. (25)

$$\begin{cases} \min\limits_{\alpha \geq 0} \dfrac{1}{2} \sum\limits_{i,j=1}^{3} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum\limits_{i=1}^{n} \alpha_i \\ s.t.\ \ \alpha_1 + \alpha_2 - \alpha_3 = 0, \ \ \alpha_i \geq 0 \end{cases}$$

2) Plug data points into optimization model:

$$\begin{aligned} s(x, \alpha) &= \min_{\alpha \geq 0} \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^{n} \alpha_i \\ &= \frac{1}{2}(18\alpha_1^2 + 25\alpha_2^2 + 2\alpha_3^2 + 42\alpha_1\alpha_2 - 12\alpha_2\alpha_3 - 14\alpha_2\alpha_3) - \alpha_1 - \alpha_2 - \alpha_1 \\ &= 4\alpha_1^2 + \frac{13}{2}\alpha_2^2 + 10\alpha_1\alpha_2 - 2\alpha_1 - 2\alpha_2 \end{aligned}$$

3) Compute partial derivative for $\boldsymbol{\alpha}$

$$\begin{cases} \frac{\partial L}{\partial \alpha_1} = 0 \rightarrow 8\alpha_1 + 10\alpha_2 - 2 = 0 \\ \frac{\partial L}{\partial \alpha_2} = 0 \rightarrow 13\alpha_2 + 10\alpha_1 - 2 = 0 \end{cases}$$

We can compute $\alpha_1 = 1.5, \alpha_2 = -1$, but they are not satisfying $\alpha_i \geq 0$, which mean the feasible α_i should be on the boundary $\alpha_i = 0$. Therefore, $\alpha_1 = 0, \alpha_2 = 2/13$, or $\alpha_1 = 0.25, \alpha_2 = 0$.

Plug $\alpha_1 = 0, \alpha_2 = 2/13$ or $\alpha_1 = 0.25, \alpha_2 = 0$ into $s(x, \alpha) = 4\alpha_1^2 + \frac{13}{2}\alpha_2^2 + 10\alpha_1\alpha_2 - 2\alpha_1 - 2\alpha_2$ in step 2) separately.

$$s\left(0, \frac{2}{13}\right) = 0.11 \; and \; s(0.25, 0) = -0.25$$

Therefore, we can get $\alpha_1 = 0.25, \alpha_2 = 0$, now $\alpha_3 = \alpha_1 + \alpha_2 = 0.25$

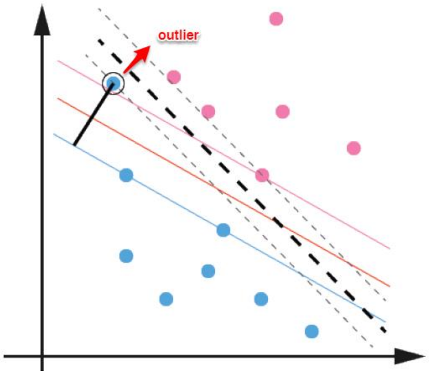
4) Compute w and b

$$\begin{cases} w^* = \sum_{i=1}^{n} \alpha_i y_i x_i = 0.25 \times 1 \times (3,3) + 0 \times 1 \times (4,3) + 0.25 \times (-1) \times (1,1) = (0.5, 0.5) \\ b^* = y - w^* x = -\frac{(0.5, 0.5)(1, 1) + \; (0.5, 0.5)(3, 3)}{2} = -2 \end{cases}$$

Finally, the best hyperplane is $0.5\alpha_1 + 0.5\alpha_2 - 2 = 0$.

6.4 SVM with Soft Margin

The above SVM can be considered as SVM with strictly correct classification, but sometimes this classification is not good because it's easy to be overfitting. For example, there are some uncertain outliers of data in real life, **if we want to classify data with outliers, the output is a decision boundary with a small margin, but if we classify data without outliers, the output will be a decision boundary with a big margin.** Therefore, if we ignore some outliers, the model will be so good that that avoid overfitting.



Generally, we introduce a slack variable ξ_i to decrease the requirement of model allow a little bit error:

$$y_i(w^T x_i + b) \geq 1 - \xi_i \tag{28}$$

Then the objective function becomes the following equation with a regularization term:

$$\min\left(\frac{1}{2}||w||^2+C\sum_{i=1}^{n}\xi_i\right)\qquad \xi_i\geq 0,\qquad i=0,1,\dots,n \tag{29}$$

Note:

- If C is close to positive infinity, we must make $\xi_i = 0$ to minimize Eq. (29), which means nothing changing with Eq. (18);
- If C is close to negative infinity, we can be allowed to make ξ_i a little bigger, which means this model is with a bigger Error Tolerance being caused by ignoring some outliers.

The Lagrange function becomes:

$$L(w,b,\alpha,\xi,\mu)=\frac{1}{2}||w||^2+C\sum_{i=1}^{n}\xi_i-\sum_{i=1}^{n}\alpha_i[y_i(w^Tx_i+b)-1+\xi_i]-\sum_{i=1}^{n}u_i\xi_i \tag{30}$$

Partial derivative of Lagrange function

1) min{L(w, b , α, ξ, μ)} based on w, b and ξ_i:

$$\begin{cases}\dfrac{\partial L}{\partial w}=0\rightarrow w=\sum_{i=1}^{n}\alpha_i y_i x_i\\ \dfrac{\partial L}{\partial b}=0\rightarrow 0=\sum_{i=1}^{n}\alpha_i y_i x_i\\ \dfrac{\partial L}{\partial \xi_i}=0\rightarrow C-\alpha_i-\xi_i=0\end{cases} \tag{31}$$

2) Plug Eq. (31) into Eq. (30) and max{L(w, b , α, ξ, μ)} based on α:

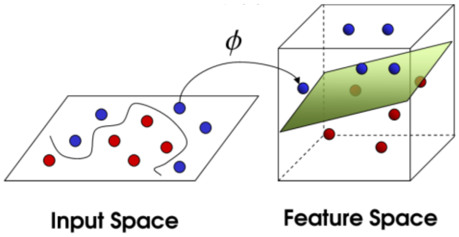
$$\max_{\alpha\geq 0}(w,b,\alpha,\xi,\mu)=\max_{\alpha\geq 0}\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j \tag{32}$$

$$\begin{cases}s(x,\alpha)=\min\limits_{\alpha\geq 0}\dfrac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j-\sum_{i=1}^{n}\alpha_i\\ s.t.\ \sum_{i=1}^{n}\alpha_i y_i=0,\ \ 0\leq\alpha_i\leq C\end{cases} \tag{33}$$

6.5 Kernel Function

6.5.1 Dimension Transformation

The above situations are linear separable, but what if the data set is linear inseparable? What we should do is transfer data from low dimension to high dimension.



Let’s say there are two data points: x = (x_1, x_2, x_3), y = (y_1, y_2, y_3) in 3D space, supposing we want to map them into a 9 D space. The mapping function is:

$$F(x) = (x_1x_1, x_1x_2, x_1x_3, x_2x_1, x_2x_2, x_2x_3, x_3x_1, x_3x_2, x_3x_3) \tag{34}$$

For example, x = (1, 2, 3), y = (4, 5, 6), we can get: F(x) = (1, 2, 3, 2, 4, 6, 3, 6, 9), F(y) = (16, 20, 24, 20, 25, 30, 24, 30, 36), and we get the cross product is <F(x), F(y)> = 1024.

If we compute cross product in this way, it will cost so much time. But what if we compute cross product in low dimension, then mapping the results into high dimension space?

$$K(x, y) = (< x, y >)^2 = 1024 \tag{35}$$

The result is same. We can get the conclusion:

$$K(x, y) = (< x, y >)^2 = < F(x), F(y) > \tag{36}$$

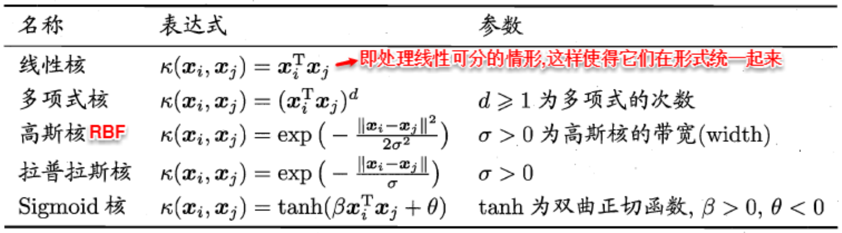
Now the question would be

$$\begin{cases} s(x, \alpha) = \min\limits_{\alpha \geq 0} \dfrac{1}{2} \displaystyle\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{K(x_i, x_j)} - \sum_{i=1}^{n} \alpha_i \\ s.t. \displaystyle\sum_{i=1}^{n} \alpha_i y_i = 0, \ \ \alpha_i \geq 0, \ \ i = 1,2,3, \dots, n \end{cases} \tag{37}$$

And the classification function would be

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i \, \boldsymbol{K(x_i, x)} + b \tag{38}$$

6.5.2 Common Kernel Function



名称	表达式	参数
线性核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j$ → 即处理线性可分的情形,这样使得它们在形式统一起来	
多项式核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j)^d$	$d \geqslant 1$ 为多项式的次数
高斯核RBF	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right)$	$\sigma > 0$ 为高斯核的带宽(width)
拉普拉斯核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\sigma}\right)$	$\sigma > 0$
Sigmoid 核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\beta \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j + \theta)$	tanh 为双曲正切函数, $\beta > 0$, $\theta < 0$

In a word, the essence of Kernel function is to compute in a low dimension space, and show the output in a high dimension space.

6.6 Pros and Cons

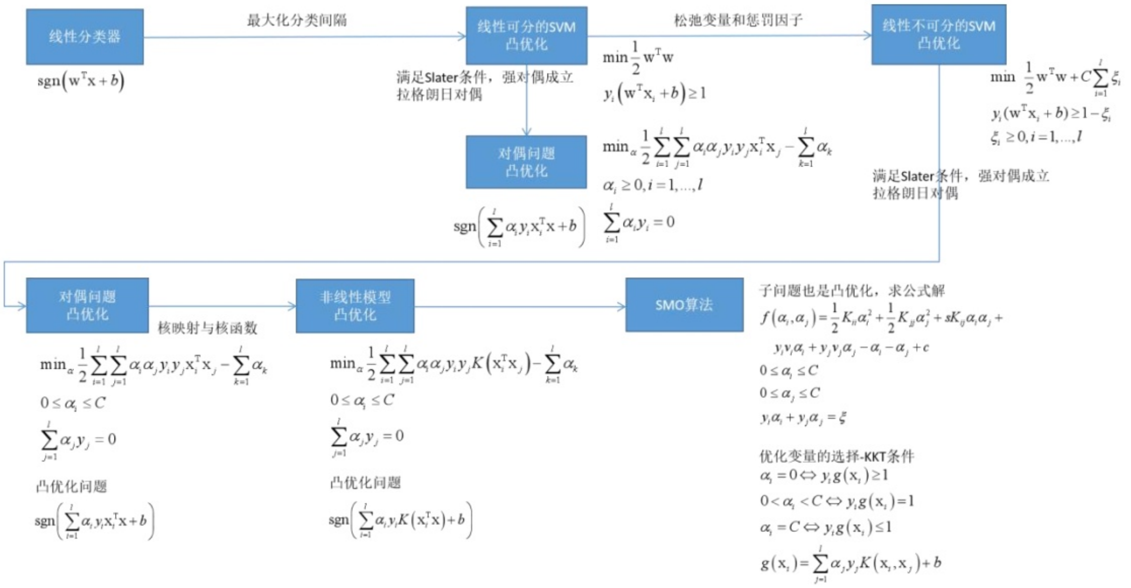
Pros:

- It can solve the problem of high dimension space with large features.
- Solve machine learning problems with small samples;
- Can handle the interaction of nonlinear features;
- No local minimum problem; (as opposed to algorithms such as neural networks)
- You don't have to rely on the whole data;
- Strong generalization ability;

Cons:

- When there are many samples, the efficiency is not very high;
- There is no general solution to nonlinear problems, sometimes it is difficult to find a suitable kernel;
- Conventional SVM only supports dichotomies;
- Sensitive to missing data;

Summary


线性分类器
$\mathrm{sgn}(\mathrm{w}^{\mathrm{T}}\mathrm{x}+b)$
最大化分类间隔
线性可分的SVM 凸优化
$\min \frac{1}{2}\mathrm{w}^{\mathrm{T}}\mathrm{w}$
$y_i(\mathrm{w}^{\mathrm{T}}\mathrm{x}_i+b)\geq 1$
松弛变量和惩罚因子
线性不可分的SVM 凸优化
$\min \frac{1}{2}\mathrm{w}^{\mathrm{T}}\mathrm{w}+C\sum_{i=1}^{l}\xi_i$
$y_i(\mathrm{w}^{\mathrm{T}}\mathrm{x}_i+b)\geq 1-\xi_i$
$\xi_i\geq 0, i=1,\dots,l$
满足Slater条件，强对偶成立 拉格朗日对偶
对偶问题 凸优化
$\min_\alpha \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \mathrm{x}_i^{\mathrm{T}}\mathrm{x}_j-\sum_{k=1}^{l}\alpha_k$
$\alpha_i\geq 0, i=1,\dots,l$
$\sum_{i=1}^{l}\alpha_i y_i=0$
$\mathrm{sgn}\left(\sum_{i=1}^{l}\alpha_i y_i \mathrm{x}_i^{\mathrm{T}}\mathrm{x}+b\right)$
满足Slater条件，强对偶成立 拉格朗日对偶
对偶问题 凸优化
核映射与核函数
$\min_\alpha \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \mathrm{x}_i^{\mathrm{T}}\mathrm{x}_j-\sum_{k=1}^{l}\alpha_k$
$0\leq\alpha_i\leq C$
$\sum_{j=1}^{l}\alpha_j y_j=0$
凸优化问题
$\mathrm{sgn}\left(\sum_{i=1}^{l}\alpha_i y_i \mathrm{x}_i^{\mathrm{T}}\mathrm{x}+b\right)$
非线性模型 凸优化
$\min_\alpha \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(\mathrm{x}_i^{\mathrm{T}}\mathrm{x}_j)-\sum_{k=1}^{l}\alpha_k$
$0\leq\alpha_i\leq C$
$\sum_{j=1}^{l}\alpha_j y_j=0$
凸优化问题
$\mathrm{sgn}\left(\sum_{i=1}^{l}\alpha_i y_i K(\mathrm{x}_i^{\mathrm{T}}\mathrm{x})+b\right)$
SMO算法
子问题也是凸优化，求公式解
$f(\alpha_i,\alpha_j)=\frac{1}{2}K_{ii}\alpha_i^2+\frac{1}{2}K_{jj}\alpha_j^2+sK_{ij}\alpha_i\alpha_j+$
$y_i v_i\alpha_i+y_j v_j\alpha_j-\alpha_i-\alpha_j+c$
$0\leq\alpha_i\leq C$
$0\leq\alpha_j\leq C$
$y_i\alpha_i+y_j\alpha_j=\xi$
优化变量的选择-KKT条件
$\alpha_i=0\Leftrightarrow y_i g(\mathrm{x}_i)\geq 1$
$0<\alpha_i<C\Leftrightarrow y_i g(\mathrm{x}_i)=1$
$\alpha_i=C\Leftrightarrow y_i g(\mathrm{x}_i)\leq 1$
$g(\mathrm{x}_i)=\sum_{j=1}^{l}\alpha_j y_j K(\mathrm{x}_i,\mathrm{x}_j)+b$