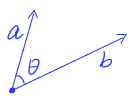
SVM

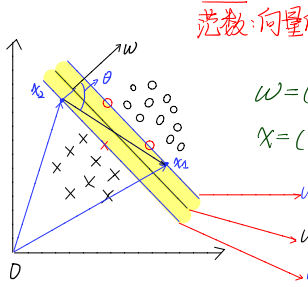
(1) Linear Separable Condition



内积: $(1,2,3)\cdot(4,5,6)=1\times4+2\times5+3\times6=32$

$a\cdot b=\|a\|\cdot\|b\|\cdot\cos\theta$

范数: 向量的长度



$w=(w_1,w_2,\cdots w_n)$ 平面法向量

$x=(x_1,x_2,\cdots,x_n)$

$wx+b=1$ 支持平面 support plane

$wx+b=0$ 分离平面 separation plane

$wx+b=-1$ 支持平面 support plane.

×,○,◎ 支持向量 support vector

Compute the distance between 2 support plane

$\begin{cases} w\cdot x_1+b=1 \ (1) \\ w\cdot x_2+b=-1 \ (2) \end{cases}$

(1)-(2) $\Rightarrow w(x_1-x_2)=2 \Rightarrow \|w\|\cdot\|x_1-x_2\|\cos\theta=2 \Rightarrow d=\frac{2}{\|w\|}$

间隔要求最大 max

Decision function $\begin{cases} w\cdot x_i+b\geq1, & y_i=1 \\ \text{上支持平面以上} \\ \text{下支持平面以下} \\ w\cdot x_i+b\leq-1, & y_i=-1 \end{cases}$ Constraint

(1) Convex Optimization Problem

$\min\limits_{w}\frac{\|w\|^2}{2}$, subject to 受限于 $y_i(w\cdot x_i+b)\geq1$, $i=1,\cdots,N$

约束 Constraint

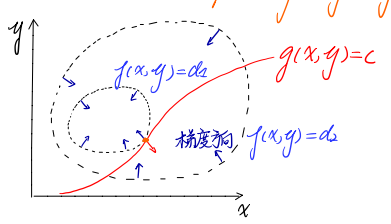
(1) Lagrange Multiplier Method

$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{N}\alpha_i(y_i(w^T\cdot x_i+b)-1)$

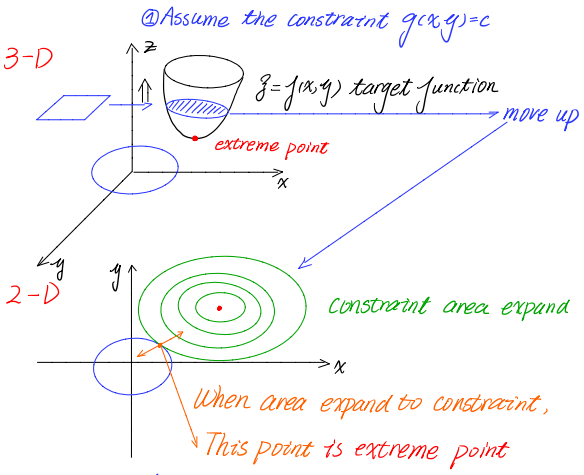
$\frac{\partial L(w,b,\alpha)}{\partial w}=0 \Rightarrow w=\sum_{i=1}^{N}\alpha_i y_i x_i$

$\frac{\partial L(w,b,\alpha)}{\partial b}=0 \Rightarrow \sum_{i=1}^{N}\alpha_i y_i=0$

Principle of lagrange



①Assume the constraint $g(x,y)=c$



$\min f(x,y)$, s.t. $g(x,y)=c$

$L(x,y)=f(x,y)+\lambda(g(x,y)-c)$

Lagrange multiplier

$\nabla L(x,y)=\nabla(f+\lambda(g-c))$ 目标函数扩展

$=\nabla f+\lambda\nabla g=0$ 与约束首次接触的点,即为极值点

$\Rightarrow \nabla f=-\lambda\nabla g$

(2) KKT最优化: Lagrange multiplier method 的推广

Since for lagrange multiplier method,

$\min\limits_{w}\frac{\|w\|^2}{2}$, subject to 受限于 $y_i(w\cdot x_i+b)=1$, $i=1,\cdots,N$ ↑must be equal

So, introduce KKT 解带有不等号约束的凸优化问题

$\min f(x)$ s.t. 受制于 $h_j(x)=0$, $j=0,1,\cdots,p$; $g_k(x)\leq0$, $k=0,\cdots,q$; $x\in X\subset R^n$

Can convert it into

1. $h_j(x)=0$, $j=0,\cdots,p$; $g_k(x)\leq0$, $k=0,\cdots,q$

2. $\nabla f(x)+\sum_{j=1}^{p}\lambda_j\nabla h_j(x)+\sum_{k=1}^{q}\mu_k\nabla g_k(x)=0$

$\lambda_j\neq0$, $\mu_k\geq0$, $\mu_k g_k(x)=0$

(3) Take result of gradient from lagrang method into original function ⇒ Dual problem

对偶问题

Finally ⇒ $L(w,b,\alpha)=\frac{1}{2}\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j(x_i^T\cdot x_j)$

变量变成了 $\alpha_i\alpha_j$ 数据已知

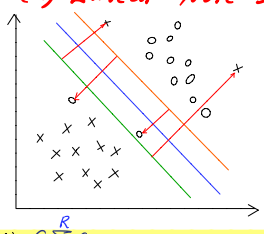
s.t $\alpha_i\geq0$, $i=1,\cdots,n$

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

Assume $\alpha=\{\alpha_i\}$, $i=1,\cdots,n$; is solution of, exist $\alpha_j>0$

$w=\sum_{i=1}^{N}\alpha_i y_i x_i$; $b=y_j-\sum_{i=1}^{N}\alpha_i y_i(x_i\cdot x_j)$

(2) Linear Non-separable



$\min\frac{1}{2}\|w^2\|+C\sum_{i=1}^{R}\varepsilon_i$ Slack 越小越好

$y_i(w^Tx_i+b)\geq 1-\varepsilon_i$, $\varepsilon_i\geq0$

木槿色线,绿色线 性分类平面近

(1) $C\sum_{i=1}^{R}\varepsilon_i$ is a penalty item based on linear separable condition.

1, When x_i in right side, $\varepsilon=0$, $R=Size(X)$, C is assigned by user, 表示对分错的点加入多少的惩罚。

2. When C is large, 错分点↓; When C is small, 错分点↑

⇓ 会有 overfitting ⇓ 模型又太正确

(2) Dual problem in linear non-separable condition

$\max\limits_{\alpha}\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j$ s.t. $C\geq\alpha_{i,j}\geq0$, $i=1,\cdots,n$ (C为一个边界)

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

(3) SMO Algorithm (Sequential Minimal optimization) 解 lagrange ⇒ dual

1. Update 2 multipliers each time, recursion ⇒ get result.

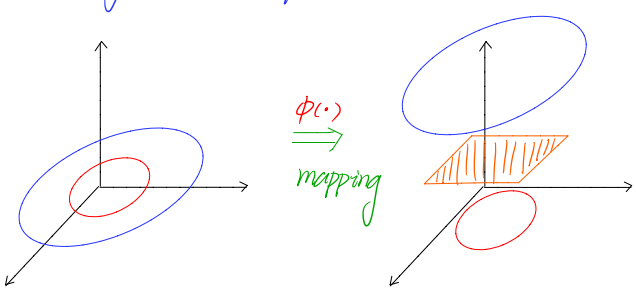
Repeat till convergence{

1. Select some pair α_i,α_j to update next (using a heuristic that tries to pick the two that will allow us to make the biggest progress towards the global maximum)

2. Reoptimize $W(\alpha)$ with respect to α_i,α_j, while holding all the other α_k's fixed }

(4) Non-linear Condition —— Dimension transform

Convert linear non-separable data into linear separable data in a high dimension space.



$x_i=\{x_i^{(1)},x_i^{(2)},\cdots,x_i^{(n)}\}$ ⟹ $\phi(x_i)$

$y_i=\{-1,0,1\}$ ⟹ y_i (φ(·))

Using mapped sample vectors do inner product

$\max\limits_{\alpha}\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i x_j$ ⟹ $\max\limits_{\alpha}\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\phi(x_i)^T\phi(x_j)$

s.t. $C\geq\alpha_{i,j}\geq0$, $i=1,\cdots,n$; $\sum_{i=1}^{n}\alpha_i y_i=0$

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

If original space is n-dimensional, mapping to n^2-dimensional space. Time complexity from $O(n)\rightarrow O(n^2)$

(5) Kernel Function

① Example: Assume x,z are N-dimensional veriable.

Using Kernel function $K(x,z)=(x^Tz)^2$

Expand it: $K(x^T,z)=(x^Tz)^2=\left(\sum_{i=1}^{n}x_i^Tz_i\right)\left(\sum_{j=1}^{n}x_jz_j\right)=\sum_{i=1}^{n}\sum_{j=1}^{n}x_ix_jz_iz_j$

$=\sum_{i=1}^{n}\sum_{j=1}^{n}(x_ix_j)(z_iz_j)=\phi(x)^T\phi(z)$

So, we can can compute the inner product of x,z, then $(xz)^2$, decreasing the time complexity.

② If for any mapping function, we can find a kernel function, making it possible that we could compute inner product in the mapped high dimensional space, into the low level dimensional space.

we know $\phi(\cdot)$, and we can find the $K(\cdot)$

In fact, we can find a kernel function first, for example, the problem in 3-D space is non-linear. Need to map the problem into a high-D space, making it a linear problem. We can replace the inner product of high level dimension with the value computed by the kernel function.

不用找到 $\phi(\cdot)$, 只需要在低维空间找出一个合理的核函数,并在数学上证明:一定存在 $\phi(\cdot)$

可以把低维空间中非线性映射成为高维空间的线性问题。

低维空间计算核函数的值可代替高维空间的内积。

③ Necessary and sufficient condition of kernel function —— Mercer Theorem

If function K is a mapping from $R^n\times R^n\rightarrow R$ (2 n-dimensional vector mapped to real number). If K is a mercer kernel function then it if for training sample $\{x_i\}$ $i=1,\cdots,N$, it's corresponding kernel function matrix is symmetric semi-positive.

① Kernel function matrix $\begin{bmatrix} & & \\ & k_{ij} & \\ & & \end{bmatrix}$ $k_{ij}=K(x_i,x_j)$

② Positive define matrix: Assume M is a n-th order matrix, for any non-zero vector z. $zMz>0$, calling M a positive defined matrix

③ Semi-positive define matrix: Assume M is a n-th order matrix, for any non-zero vector z. $zMz\geq0$, calling M a positive defined matrix

④ Commonly used kernel function

多项式 1. $K(x_i,x_j)=(x_ix_j+1)^k$

高斯 RBF 2. $K(x_i,x_j)=\exp(-\frac{\|x_i-x_j\|^2}{2\sigma^2})$

S型 3. $K(x_i,x_j)=\tanh(kx_i\cdot x_j-\delta)$

逼近 Approch

(6) Using Kernel function in lagrange function

① maximize $L(\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j\phi(x_i)\phi(x_j)$

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

⇓

$\max\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i,x_j)$ s.t. $0\leq\alpha_i\leq C$; $\sum\alpha_i y_i=0$

低维空间中计算: 比如将x映射为α_i, 可用SMO向导出.

② Using kernel function to do classification

$w^Tx+b=\left(\sum_{i=1}^{m}\alpha_i y_i x_i\right)^Tx+b=\sum_{i=1}^{m}\alpha_i y_i K(x_i,y_i)+b$

(7) Algorithm

Input: Linear separable training data set

$D=\{(x_i,y_i)\}$, $i=1,\cdots,N$; $x_i\in\chi=R^n$; $y_i\in\gamma=\{+1,-1\}$ $i=1,\cdots N$

Output: Maximum margin separation hyperplane

① Linear separable svm — margin maximum

step(1): $\min\limits_{w,b}\frac{1}{2}\|w\|^2$; s.t $y_i(w\cdot x_i+b)-1\geq0$, $i=1,\cdots,N$

compute optimal solution w^*, b^*

step(2) Hyperplane $w^*x+b^*=0$

Decision function $f(x)=sign(w^*x+b^*)$

② Linear separable SVM

Step(1) $\min\limits_{\alpha}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)-\sum_{i=1}^{N}\alpha_i$

s.t. $\sum_{i=1}^{N}\alpha_i y_i=0$; $\alpha_i\geq0$, $i=1,\cdots,N$

compute the optimal solution $\alpha^*=(\alpha_i^*)^T$, $i=1,\cdots,N$

Step(2) Compute $w^*=\sum_{i=1}^{N}\alpha_i^* y_i x_i$

choose any $\alpha_j^*>0$ in α^*, and compute

$b^*=y_j-\sum_{i=1}^{N}\alpha_i^* y_i(x_i\cdot x_j)$

Step(3) Hyperplane $w^*x+b=0$

Decision function $f(x)=sign(w^*x+b)$

③ Linear SVM

step(1) Choose penalty parameter C, construct convex quadratic programming problem

$\min\limits_{\alpha}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)-\sum_{i=1}^{N}\alpha_i$; s.t. $\sum_{i=1}^{N}\alpha_i y_i=0$

$0\leq\alpha_i\leq C$, $i=1,\cdots,N$

Solution $\alpha^*=(\alpha_1^*,\cdots,\alpha_N^*)^T$

Step(2) Compute $w^*=\sum_{i=1}^{N}\alpha_i^*$, choose any $\alpha_j^*\in(0,C)$, compute

$b=y_j-\sum_{i=1}^{N}y_i\alpha_i^*(x_i\cdot x_j)$

Step(3) Hyperplane $w^*x+b=0$

Decision function $f(x)=sign(w^*x+b)$

④ Nonlinear SVM

Step(1) choose kernel function $K(x,z)$ and penalty C

$\min\limits_{\alpha}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i,x_j)-\sum_{i=1}^{N}\alpha_i$

s.t. $\sum_{i=1}^{N}\alpha_i y_i=0$, $\alpha_i\in(0,c)$, $i=1,\cdots N$

get optimal $\alpha^*=(\alpha_1^*,\cdots,\alpha_N^*)^T$

Step(2) find any α_j^* from α^*, $\alpha_j^*\in(0,c)$

$b^*=y_j-\sum_{i=1}^{N}\alpha_i^* y_i K(x_i\cdot x_j)$

Step(3) Decision function

$f(x)=sign(\sum_{i=1}^{N}\alpha_i^* y_i K(x\cdot x_i)+b^*)$