SVM

Author：中山大学 17计算机学院 YSY

https://github.com/ysyisyourbrother/SYSU_Notebook

Derivation

$$w^T x + b = 0$$

Many possible separating hyperplanes -> which one to choose?

optimal: maximize the distance between the hyperplane and the closest sample point to endow strong generalization capability. In other words, the goal is to maximize the **margin** which is twice the distance d between the separating hyperplane and the closest sample(support vector).

目标是最大化超平面和样本集中和超平面最近的点的距离

Define

$$d = \max_{1 \le i \le n} \gamma_i = \frac{||w^T x_i + b||}{||w||}$$

Note that the scalar of w, b simultaneously don't influence the result of d.

Hence, we can let $||w^T x' + b|| = 1$ where x' is from the closest sample set.

$$\therefore \quad d = \frac{1}{||w||}, m = \frac{2}{||w||}$$

To maximize m, we can minimize $||w||$. We can construct the first model as below:

$$\begin{aligned} minimize \ & \frac{1}{2}||w||^2 \\ s.t. \quad & for\ y_i = +1, w^T x_i + b \ge +1 \\ & for\ y_i = -1, w^T x_i + b \le -1 \end{aligned}$$

Note that we can write the constraint condition into one equation

$$y_i f(x_i) - 1 = y_i(w^T x_i + b) - 1 \ge 0$$

Also note that the object function is convex, therefore there is only one global optimal.

We can use *Lagrange multipliers* to solve this convex optimize problem.

$$\begin{aligned} minimize \quad & L_P(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i (y_i(x_i^T w + b) - 1) \\ s.t. \quad & a_i \ge 0 \\ & y_i f(x_i) - 1 \ge 0 \end{aligned}$$

This website give a explanation of reason why $a_i \ge 0$ under KKT condition. Compute partial derivative of w and b, we get

$$\frac{\partial L}{\partial w} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i x_i = w \tag{1}$$
$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$$

Introduce dual problem, and substitute w by equation (1)

$$\begin{aligned} maximize \quad & L_D(w, b, \alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \\ s.t. \quad & a_i \geq 0 \\ & \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned}$$

KKT condition provides strong duality property so that the primal problem has the same optimum solution as the dual problem. Thus, under KKT condition, we have

$$\begin{cases} \text{Primal Lagrange KKT constraint} & \Rightarrow & \alpha_i \geq 0 \\ \text{Primal problem constraint} & \Rightarrow & y_i f(x_i) - 1 \geq 0 \\ \text{From derivative of primal problem} & \Rightarrow & \alpha_i (y_i (w^T x_i + b) - 1) = 0 \end{cases}$$

This website illustrates the meaning of $\alpha = 0$ and $y_i(w^T x_i + b) = 0$.

$$\begin{aligned} \text{constraint works, refering to support vectors} &\Rightarrow y_i(w^T x_i + b) - 1 = 0, \alpha_i > 0 \\ \text{constraint does not work,refering the rest vectors} &\Rightarrow y_i(w^T x_i + b) - 1 \neq 0, \alpha_i = 0 \end{aligned}$$

We can see that, the hyperplane is completely defined by support vector.

$$\begin{aligned} w &= \sum_{i=i}^{n} \alpha_i y_i x_i = \sum_{i \in SV} \alpha_i y_i x_i \\ b &= \frac{1}{y_i} - w^T x_i = y_i - w^T x_i \ , \quad i \in SV \end{aligned}$$

5.11 支持向量机

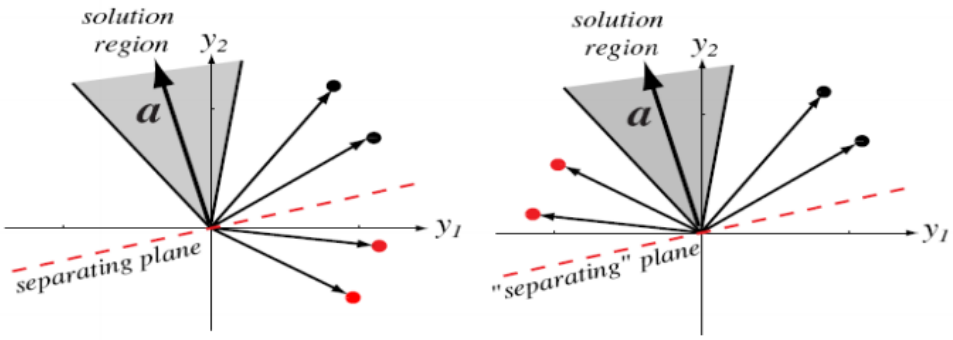
Support Vector Machines （SVM）

$$\mathbf{y}_k = \varphi(\mathbf{x}_k), \quad z_k = \pm 1$$

$$g(\mathbf{y}) = \mathbf{a}^t\mathbf{y}, \quad \mathbf{y} \text{ 属于增量空间}$$

分隔超平面保证

$$z_k g(\mathbf{y}_k) \geq 0, \quad k = 1, \cdots, n$$



$\mathbf{y}$到超平面的距离：$\dfrac{|g(\mathbf{y})|}{\|\mathbf{a}\|}$

正值边沿裕度 b,

$$\frac{z_k g(\mathbf{y}_k)}{\|\mathbf{a}\|} \geq b, k = 1, \cdots, n$$

等价形式：

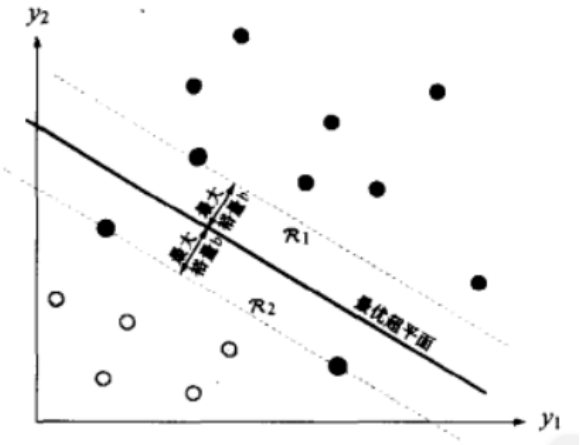
$$\frac{z_k(g(\mathbf{y}_k) - b)}{\|\mathbf{a}\|} \geq 0$$

目标: 找到一个 $\mathbf{a}$ 使得 ：

$$\min_{\mathbf{a}} \|\mathbf{a}\|^2 \text{ subject to } z_k(g(\mathbf{y}_k) - b) \geq 0, k = 1, \cdots, n$$

SVM的目标是要找到一个最小的a，满足对样本点带裕量的分离。可以直接固定b的值为1，然后最大化点和超平面的距离，这样我们只需要最小化a，而不需要考虑多一个变量b。

图 5-19　训练一个找最优超平面的支持向量机。这个最优超平面是到最近的训练模式的距离为最大的平面。支持向量是那些(最近的)模式，到超平面的距离为 b。图中有 3 个支持向量，都标明为实心点



一个支持向量机可通过选择当前被分类得最差的模式来进行训练。在训练的时候，这样的一个模式通常在判决边界错误的一边，并且离判决边界最远。当训练结束的时候，**这样的模式就是支持向量**，他们最终会被优化到超平面附近。支持向量是那些定义最优分割平面的点，也是最难被正确分类的点。

然而发现最差模式的计算是非常费时的，每次更新都要搜索所有训练集来寻找被分的最差的模式。

SVM的训练

第一步是将数据映射到更高维的空间上去，可以结合一些经验知识来设计，映射后的空间维数可以是任意高的。

我们将有约束的权向量极小化问题转换为无约束的拉格朗日待定因子问题。由约束$z_k g(y_k) \geq 1$(偏置)和最小化a的目标构造拉格朗日

$$L(\mathbf{a}, \boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{a}||^2 - \sum_{k=1}^{n} \alpha_k [z_k \mathbf{a}^t \mathbf{y}_k - 1]$$

并寻找使得L极小的权向量a和使它极大的待定因子α_k

并寻找使 $L()$极小的权向量 $\mathbf{a}$ 和使它极大的待定因子 $\alpha_k \geqslant 0$。式(108)的最后一项表达的是将样本正确分类的目标。可以证明使用 Kuhn-Tucker 构造法(习题 33)可将这个最优化的形式修改为极大化

$$L(\boldsymbol{\alpha}) = \sum_{k=1}^{n} \alpha_i - \frac{1}{2} \sum_{k,j}^{n} \alpha_k \alpha_j z_k z_j \mathbf{y}_j^t \mathbf{y}_k \qquad (109)$$

和给出训练数据的约束条件

$$\sum_{k=1}^{n} z_k \alpha_k = 0 \qquad \alpha_k \geqslant 0, \qquad k = 1, \ldots, n \qquad (110)$$

虽然这些方程可用经典二次规划来求解，但还设计了许多别的方案(见参考文献)。

Kuhn-Tucker 定理

$$\min_{\mathbf{a}} L(\mathbf{a}, b, \boldsymbol{\alpha})$$

$$L(\mathbf{a}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{a}\|^2 - \sum_{k=1}^{n} \alpha_k \left[z_k (\mathbf{a}^t \mathbf{y}_k - b) \right] \quad (*)$$

$$\frac{\partial L(\mathbf{a}, b, \boldsymbol{\alpha})}{\partial \mathbf{a}} = 0, \quad \frac{\partial L(\mathbf{a}, b, \boldsymbol{\alpha})}{\partial b} = 0$$

$$\mathbf{a} = \sum_{k=1}^{n} \alpha_k z_k \mathbf{y}_k, \quad \sum_{k=1}^{n} \alpha_k z_k = 0$$

代入公式（*），且b=1，得

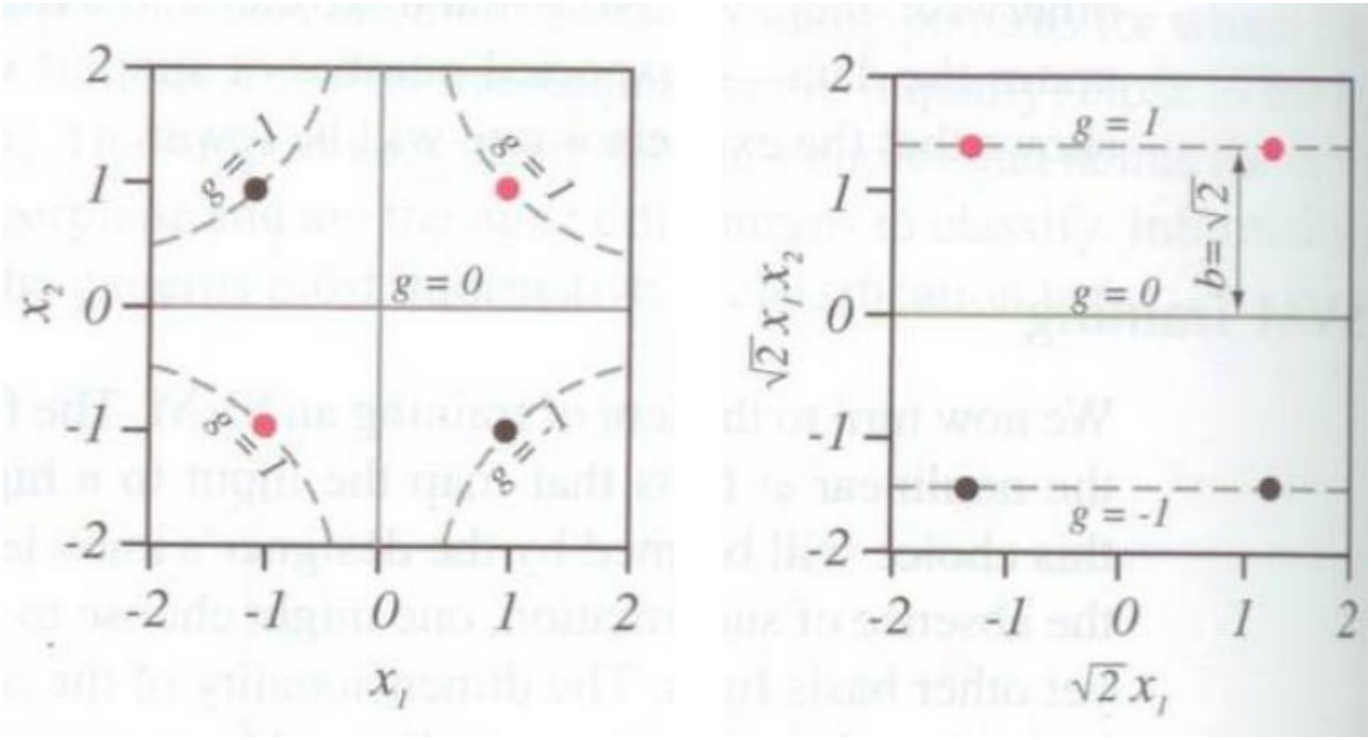
$$L(\boldsymbol{\alpha}) = \sum_{k=1}^{n} \alpha_k - \frac{1}{2} \sum_{k,j}^{n} \alpha_k \alpha_j z_k z_j \mathbf{y}_j^t \mathbf{y}_k$$

$$\text{Maximizing L, 满足:} \sum_{k=1}^{n} \alpha_k z_k = 0$$

上式中的SUM符号相当于两层求和

XOR问题的SVM

使用SVM的方法我们预处理他们的特征，将他们映射到一个更高维的空间中，在这个空间中他们线性可分。我们使用：$1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2$ 这里的根号2是为了规范化。



$$\omega_1 : \mathbf{x}_1 = \begin{pmatrix} 1 & 1 \end{pmatrix}^t, \mathbf{x}_3 = \begin{pmatrix} -1 & -1 \end{pmatrix}^t$$

$$\omega_2 : \mathbf{x}_2 = \begin{pmatrix} 1 & -1 \end{pmatrix}^t, \mathbf{x}_4 = \begin{pmatrix} -1 & 1 \end{pmatrix}^t$$

$$\varphi \text{ functions } : 1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2$$

$$\max_{\boldsymbol{\alpha}} \sum_{k=1}^{4} \alpha_k - \frac{1}{2} \sum_{k,j} \alpha_k \alpha_j z_k z_j \mathbf{y}_j^t \mathbf{y}_k$$

$$\text{subject to } \alpha_1 - \alpha_2 + \alpha_3 - \alpha_4 = 0, \text{and } \alpha_k \geq 0$$

$$\text{解} : \alpha_k^* = 1/8, k = 1, \cdots, 4$$

这四个训练样本都是支 持向量

$$\mathbf{a}^* = \sum_{k=1}^{4} \alpha_k^* z_k \mathbf{y}_k$$

$$\text{最终判别函数} : g(\mathbf{x}) = x_1 x_2$$

$$\text{判定超平面} : g = 0$$

$$\text{裕度} : b^* = 1/\|\mathbf{a}^*\| = \sqrt{2}$$

VC dimension

In order to solve the problem where sample are not linearly separable, we introduce the concept of VC dimension. The VC dimension of a class of function $\{f_i\}$ is the maximum number of points that can be separated into two classes in all possible ways by $\{f_i\}$.

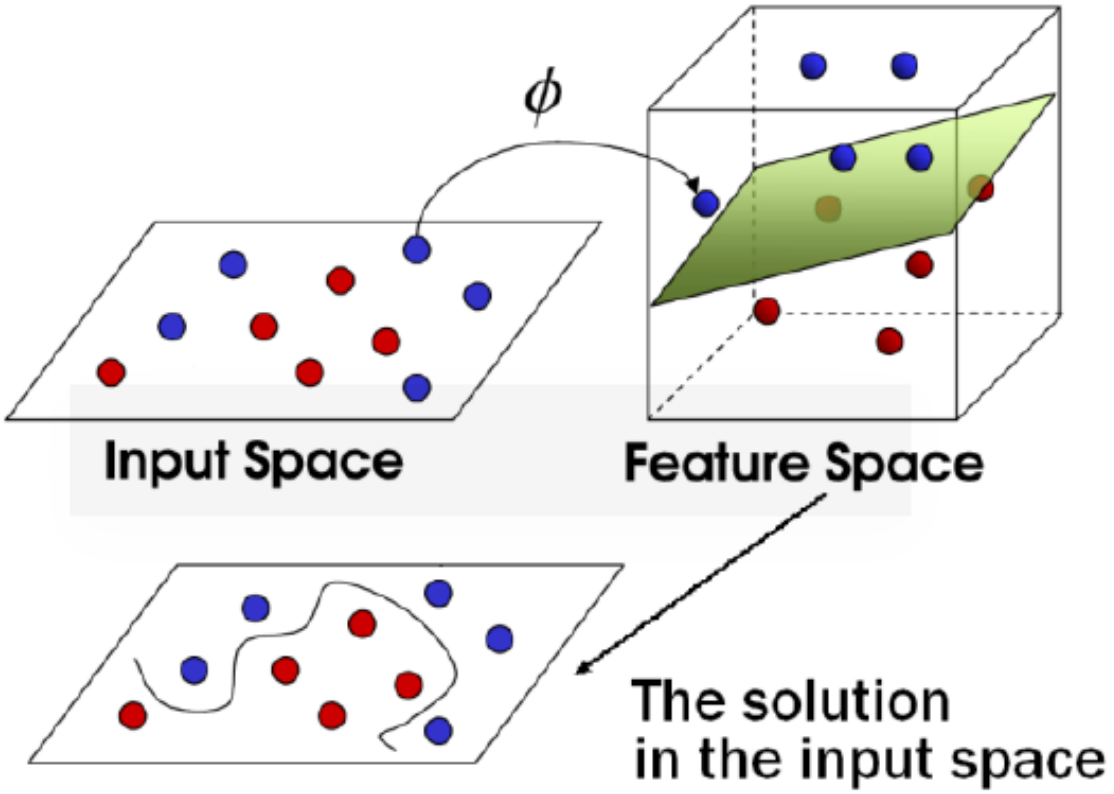
When dimension = 2, we can only separate 3 points.

We can also treat that 3 points can be separated in two dimension space. When there are 4 points, we can not separated in 2 dimension space.

Theorem The VC dimension of the set of oriented hyperplanes in R^n is $n + 1$.

Thus, when there are finite set of points, we can always find a hyperplanes after mapping the data to higher dimension.

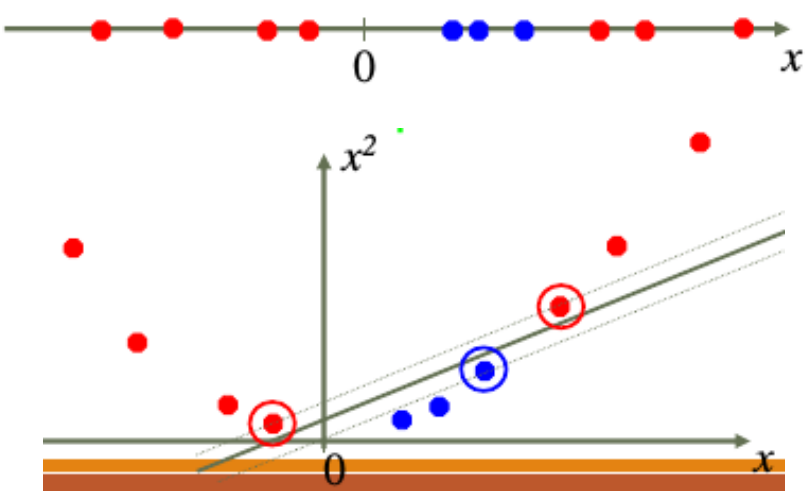
e.g.



kernel trick

We define a mapping function $\varphi(x)$ to a space of sufficiently high dimension.

e.g. $\varphi(x) = (x, x^2)$

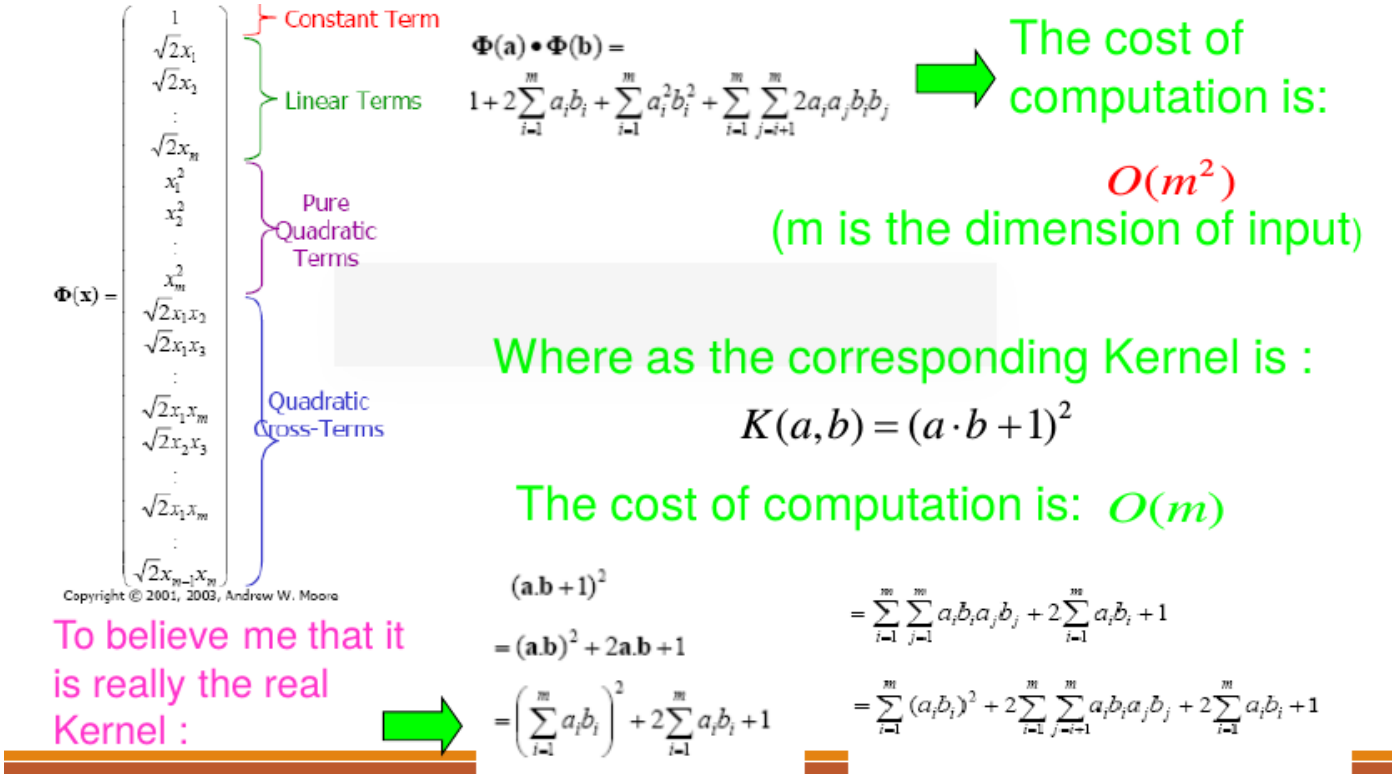


Optimization problem change into

$$maximize\ \ L_D(w,b,\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{i=1}^{n}\alpha_i\alpha_j y_i y_j \varphi(x_i)^T\varphi(x_j)$$

Decision function will be $g(x) = f(\varphi(x)) = sign(w^T\varphi(x)+b) = sign(\sum_{i=1}^{n}\alpha_i y_i \varphi(x_i^t)\varphi(x)+b)$

Compute in high dimension is computationally expensive. Nevertheless, define kernel function $K(x_i,x_j) = \varphi(x_i)^T\varphi(x_j)$ and it's unnecessary to perform computations in high dimensional space explicitly.



the Gram matrix

K=

K(1,1)	K(1,2)	K(1,3)	…	K(1,m)
K(2,1)	K(2,2)	K(2,3)	…	K(2,m)
…	…	…	…	…
K(m,1)	K(m,2)	K(m,3)	…	K(m,m)

Mercer's Theorem:

A function $K(x_i,x_j)$ is a kernel iff the kernel matrix is symmetric positive semi-definite

A function $K(x_i,x_j)$ is a kernel iff for any $g(u)$ such that $\int g(u)^2 du$ is finite then $\int K(u,v)g(u)g(v)dudv \geq 0$

- Polynomial kernel: $K(x_i,x_j) = (x_i^T x_j + 1)^p$
- Gaussian Radial Basis Function: $K(x_i,x_j) = e^{-\frac{1}{2\sigma^2}||x_i - x_j||^2}$ data is lifted to infinite dimension(consider

$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$)

- Sigmoidal: $K(x_i, x_j) = \tanh\left(kx_i^T x_j - \delta\right)$. Note that it is not a kernel for every k and δ

Note that SVM model using a sigmoid kernel function is equivalent to a two-layer, feed-forward neural network.

There are many ways to create kernels

- $K(a, b) = cK_1(a, b)$
- $K(a, b) = K_1(a, b)K_2(a, b)$
- $K(a, b) = K_3(\varphi(a), \varphi(b))$
- $K(a, b) = a^T Bb$ where B is a positive semi-definite $d \times d$ matrix.

Soft margin

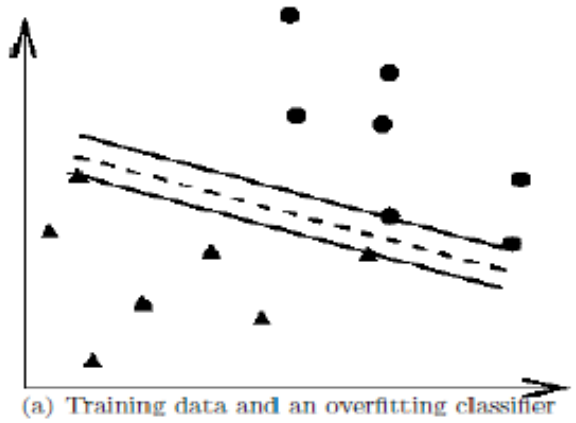
Sometimes, the points are linearly separable but the margin is low. Soft margin can release this problem.

Use slack variable $\xi_1, \xi_2, \ldots, \xi_n$ for each sample. Change the constrains to

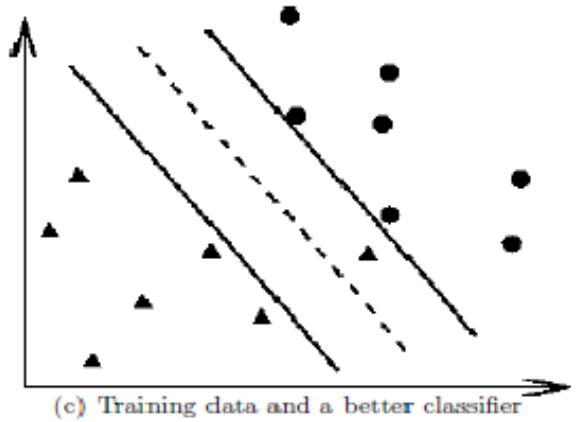
$$y_i(w^T x_i + b) \geq 1 - \xi_i \quad \forall i \quad \xi_i \geq 0$$

If $0 < \xi_i < 1$, then sample i is on the correct side of the hyperplane but within the region of the region.

If $\xi_i > 1$, then sample i is on the wrong side of the hyperplane.



(a) Training data and an overfitting classifier



(c) Training data and a better classifier

Then, optimization problem becomes

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i \\ s.t. \quad & y_i(w^t x_i + b) \geq 1 - \xi_i, \quad \forall i \end{aligned}$$

C is a hyper parameter to control the penalty of term $\sum_{i=1}^{n}\xi_i$. Large C means less training samples that are not in ideal position. It may lead to over-fitting. Small C means more more training samples that are not in ideal position. It may lead to under-fitting.

The dual optimization problem becomes

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j \\ s.t. \quad & 0 \leq \alpha_i \leq C, \forall i \\ & \sum_{i=1}^{n}\alpha_i y_i = 0 \end{aligned}$$

Under KKT condition

$$\begin{aligned} \alpha_i = 0 (\text{constraint does not work}) &\Rightarrow y_i(w^T x_i + b) > 1 \\ 0 < \alpha_i < C (\text{constraint of support vector}) &\Rightarrow y_i(w_i^T + b) = 1 \\ \alpha_i = C (\text{points with slack variable } \xi_i > 0) &\Rightarrow y_i(w^T x_i + b) < 1 \end{aligned}$$

Combine **kernel trick** and **soft margin** and use cross-validation, we can find a good SVM model for binary classification task.

Multi-class classification

- one vs. rest(ovr)
- one vs. one(ovo)
- many vs. many(MvM)

weakness

- Don't know which dimension that we can map the data to a space in which they are linearly separable. Higher dimension just increases the likelihood that they are linear separable.(Gaussian Radial Basis can solve this problem since it map the data into infinite dimension space but may lead to over-fitting)
- Kernel matrix is almost diagonal. The points are orthogonal and only similar to itself. Thats to say, our kernel function is not really adequate since it does not generalize good over the date.

symptoms of over-fitting

- low margin
- large number of support vectors

advantages

- Bases on a strong and nice theory
- Training is relatively easy(no local optimal, training time does not depend on dimensionality of feature space, only on fixed input space)
- generally avoids over-fitting
- Almost superior when data set is small.