6. Kernels

这里简单地介绍一下核函数，受限于所学，以下的内容可能存在着错误或者理解上的偏差。

我们知道，Logistic Regression 的决策边界 (decision boundary) 是线性的，如果我们希望得到非线性的决策边界，我们可以通过 feature mapping 的方式，把数据从低维映射到高维。这里我们用 ϕ 来表示 feature mapping。

注意到，我们在之前的讨论中，我们将算法以内积的形式写出，即 $< x^{(i)}, x^{(j)} >$。如果我们想要得到非线性的决策边界，就需要先通过 ϕ 把 x 从低维空间映射到高维空间得到 $\phi(x)$，然后计算 $< \phi(x^{(i)}), \phi(x^{(j)}) >$。可以看到，这样的计算是非常耗时的，特别是当我们映射到的高维空间的维度非常高的情况下。因此，我们就希望找到一种方式把上述步骤合为一步。具体地说，我们希望我们可以不用显示地表示 $\phi(x)$，就能够得到 $< \phi(x^{(i)}), \phi(x^{(j)}) >$ 的计算结果，从而降低计算复杂度。

先来看一个例子，假设我们对 $x \in \mathcal{R}^d$ 进行二次多项式转换 (2nd order polynomial transform) 即：

$$\phi_2(x) = (1, x_1, \cdots, x_d, x_1^2, x_1x_2, \cdots, x_1x_d, x_2x_1, x_2^2, \cdots, x_2x_d, \cdots, x_d^2)^T$$

然后我们计算 $< \phi_2(x), \phi_2(x') >$ (为了避免符号的混乱，接下来用 x 和 x' 代替 $x^{(i)}$ 和 $x^{(j)}$)，即 $\phi_2(x)^T\phi_2(x')$

$$\begin{aligned}
\phi_2(x)^T\phi_2(x') &= 1 + \sum_{i=1}^{d} x_i x_i' + \sum_{i=1}^{d}\sum_{j=1}^{d} x_i x_j x_i' x_j' \\
&= 1 + \sum_{i=1}^{d} x_i x_i' + \sum_{i=1}^{d} x_i x_i' \sum_{j=1}^{d} x_j x_j' \\
&= 1 + x^T x' + (x^T x')(x^T x')
\end{aligned}$$

于是，我们发现，通过计算 $1 + x^Tx' + (x^Tx')(x^Tx')$ 就可以得到 $\phi_2(x)^T\phi_2(x')$ 的结果。而且这样的计算方式，不需要我们具体地写出 $\phi(x)$，然后再计算内积。因此，这样的方法的计算复杂度更低。其实 $K_{\phi_2}(x, x') = 1 + x^Tx' + (x^Tx')(x^Tx') = \phi_2(x)^T\phi_2(x')$ 就是核函数。

由此，我们给出核函数的定义：

设 $\mathcal{X}$ 是输入空间（欧式空间 $\mathcal{R}^n$ 的子集或离散集合），又设 $\mathcal{H}$ 为特征空间（希尔伯特空间），如果存在一个从 $\mathcal{X}$ 到 $\mathcal{H}$ 的映射

$$\phi(x) : \mathcal{X} \rightarrow \mathcal{H}$$

使得对所有的 $x, z \in \mathcal{X}$，函数 $K(x, z)$ 满足条件

$$K(x, z) =< \phi(x), \phi(z) >$$

则称 $K(x, z)$ 为核函数。

接下来，我们介绍两个常用的核函数，一个是多项式核，另一个是高斯核。

6.1 Polynomial Kernel

对上面举例的二次多项式转换，我们可以进行以下推广：

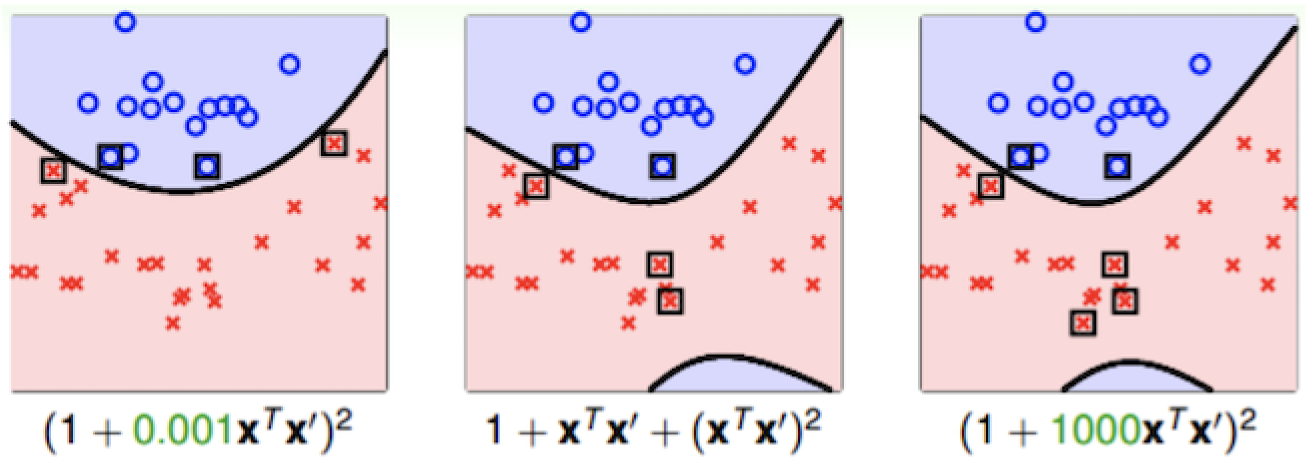
$$\phi_2(x)=(1,x_1,\cdots,x_d,x_1^2,\cdots,x_d^2)\Leftrightarrow K_{\phi_2}(x,x')=1+x^Tx'+(x^Tx')^2$$
$$\phi_2(x)=(1,\sqrt{2}x_1,\cdots,\sqrt{2}x_d,x_1^2,\cdots,x_d^2)\Leftrightarrow K_{\phi_2}(x,x')=1+2x^Tx'+(x^Tx')^2$$
$$\phi_2(x)=(1,\sqrt{2\gamma}x_1,\cdots,\sqrt{2\gamma}x_d,x_1^2,\cdots,x_d^2)\Leftrightarrow K_{\phi_2}(x,x')=1+2\gamma x^Tx'+\gamma^2(x^Tx')^2$$

总结一下，即

$$K_2(x,x')=(1+\gamma x^Tx')^2\quad with\,\gamma>0$$

我们看到，不同的 γ 对应不同的 $\phi_2(x)$。那么这些 $\phi_2(x)$ 存在着什么样的区别呢?

我们看到它们最后计算的结果是不一样的，也就是说它们定义了不同的内积。在一个内积空间中，不同的内积就代表不同的距离。而不同的距离就意味着有不同的几何特性，这样计算出来的间隔 (margin) 就不一样。采用不同的转换，虽然在看似同样的空间，但是会得到可能不同的边界。具体如下图所示：



我们还可以对上式接着进行推广：

$$\begin{aligned}K_2(x,x')&=(\zeta+\gamma x^Tx')^2\quad with\,\gamma>0,\zeta\geq 0\\K_3(x,x')&=(\zeta+\gamma x^Tx')^3\quad with\,\gamma>0,\zeta\geq 0\\&\vdots\\K_Q(x,x')&=(\zeta+\gamma x^Tx')^Q\quad with\,\gamma>0,\zeta\geq 0\end{aligned}$$

最终得到多项式核，我们可以看到，线性核其实是多项式核的特殊情况。

$$
\begin{aligned}
K_1(x, x') &= (0 + 1 \cdot x^T x')^1 \\
&\vdots \\
K_Q(x, x') &= (\zeta + \gamma x^T x')^Q \quad with\, \gamma > 0, \zeta \geq 0
\end{aligned}
$$

6.2 Gaussian Kernel

既然核函数可以让我们不需要写出 $\phi(x)$，就可以得到 $\phi(x)^T\phi(x')$ 的计算结果。那么我们能否找到一个核函数，对应的是无穷维度的特征映射 ϕ 呢？其中一个就是我们要介绍的高斯核，高斯核函数的定义如下：

$$
K(x, x') = exp\left(-\gamma\|x - x'\|^2\right) \quad with\, \gamma > 0
$$

然后，我们可以证明（这里假设 $\gamma = 1$）：

$$
\begin{aligned}
K(x, x') &= exp\left(-\|x - x'\|^2\right) \\
&= exp\left(-(x - x')^T(x - x')\right) \\
&= exp\left(-x^T x - x'^T x' + 2x^T x'\right) \\
&= exp\left(-\|x\|^2\right) exp\left(-\|x'\|^2\right) exp\left(2x^T x'\right)
\end{aligned}
$$

这里根据泰勒公式：

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

可得：

$$
\begin{aligned}
K(x, x') &= exp\left(-\|x\|^2\right) exp\left(-\|x'\|^2\right) exp\left(2x^T x'\right) \\
&= exp\left(-\|x\|^2\right) exp\left(-\|x'\|^2\right) \left(\sum_{n=0}^{\infty} \frac{(2x^T x')^n}{n!}\right)
\end{aligned}
$$

这里我们可以看到 $exp\left(-\|x\|^2\right) exp\left(-\|x'\|^2\right)$ 为常数。而 $\sum_{n=0}^{\infty} \frac{(2x^T x')^n}{n!}$ 其实是对无穷个不同维度的多项式核求和。

如果 $x, x' \in \mathcal{R}$，上式可以写成:

$$
\begin{aligned}
K(x, x') &= exp\left(-(x)^2\right) exp\left(-(x')^2\right) \left(\sum_{n=0}^{\infty} \frac{(2xx')^n}{n!}\right) \\
&= \sum_{n=0}^{\infty} \left(exp\left(-(x)^2\right) exp\left(-(x')^2\right) \sqrt{\frac{2^n}{n!}} \sqrt{\frac{2^n}{n!}} (x)^n (x')^n\right) \\
&= \phi(x)^T \phi(x')
\end{aligned}
$$

其中，

$$\phi(x) = exp\left(-x^2\right) \cdot \left(1, \sqrt{\frac{2}{1!}}x, \sqrt{\frac{2^2}{2!}}x^2, \cdots\right)$$

可以看到，高斯核对应了一个无穷维度的特征映射 ϕ。

我们换个角度来进行理解，高斯核其实是对 x 和 x' 相似度 (similarity) 很好的度量，当 x 与 x' 距离越近时，函数值越接近1；当 x 与 x' 距离越远时，函数值越接近0。（其实向量之间的内积 (inner product) 就是对两个向量相似度的度量，而核函数是在隐式地计算两个向量在高维空间的内积，也可以看做是度量两个向量的相似度。）

如果我们在 SVM 中使用高斯核，当我们要进行预测时，我们计算：

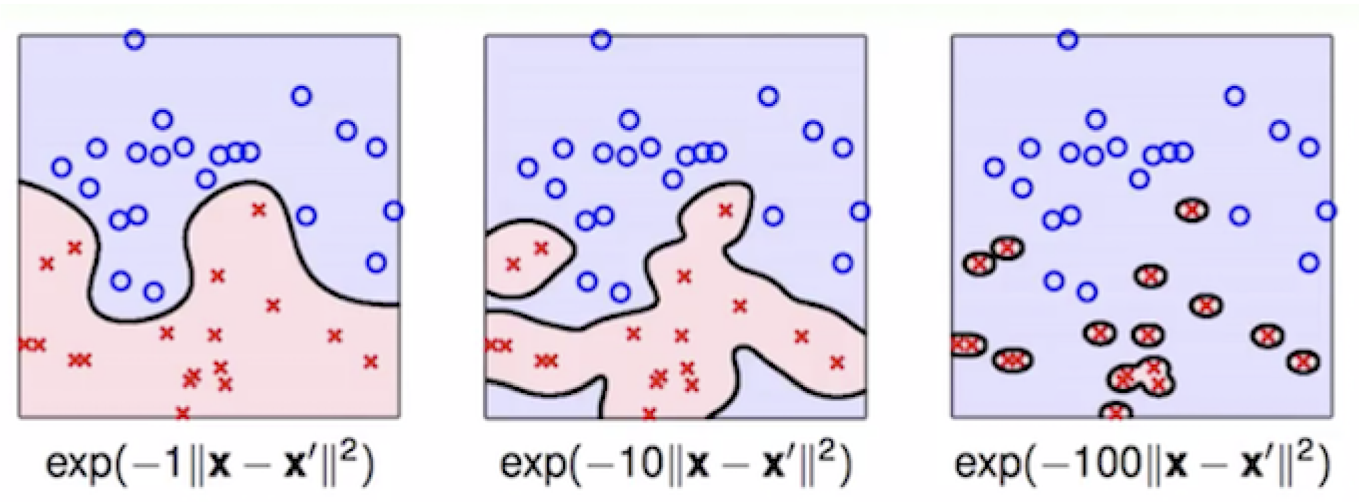
$$w^T x + b = \sum_{i=1}^{m} \alpha_i y^{(i)} K(x, x^{(i)}) + b$$

因为除了支持向量外，其他训练样本对应的 $\alpha_i = 0$。所以我们可以把 $\sum_{i=1}^{m}$ 写成 $\sum_{SV}$ 表示只对支持向量进行求和，SV 指 support vector。所以

$$\begin{aligned} w^T x + b &= \sum_{SV} \alpha_i y^{(i)} K(x, x^{(i)}) + b \\ &= \sum_{SV} \alpha_i y^{(i)} exp\left(-\gamma \|x - x^{(i)}\|^2\right) \end{aligned}$$

我们看到上式其实就是在求中心在支持向量的高斯核的线性组合。基于这个特性，也有人称高斯核为 Radial Basis Function (RBF) kernel。

接下来，我们来看看不同 γ 取值的效果：



我们也可以把高斯核函数写成如下形式：

$$K(x, x') = exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

我们可以看到，当 $\gamma = 100$ 时，每个图标 x 就像一个小岛一样。因为 γ 变大的时候就相当于标准差变小了，我们的高斯核函数就变“尖“了。而由于最后的结果是”尖尖“的高斯核函数的线性组合，就得到了图中的结果。所以，γ 的取值需要仔细地进行选择，否则有可能会过拟合。

最后，我们可以看到当 $\gamma \to \infty$ 时，高斯核函数就会变成：

$$K(x, x^{'}) = 1\{x = x^{'}\}$$

当 $x \neq x'$ 时，$K(x, x') = 0$；当 $x = x'$ 时，$K(x, x') = 1$。此时，高斯核函数就是确定输入的 x 是否等于某个支持向量。

6.3 Mercer's Theorem

那么给定某个函数 K，我们怎么样才能确定这个函数是一个有效的核(valid kernel)呢？换句话说，我们怎么样才能确定存在着某一个特征映射 ϕ，使得对于所有的 x 和 z，都有 $K(x, z) = \phi(x)^T\phi(z)$？

假设 K 是一个有效的核，对应着某种特征映射 ϕ。考虑某个有 m 个点的有限集合 $\{x^{(1)}, \cdots, x^{(m)}\}$（这个集合不一定是训练集）。然后设 K 为 $m \times m$ 的矩阵，其第 (i, j) 个值 $K_{ij} = K(x^{(i)}, x^{(j)})$，我们称矩阵 K 为核矩阵(kernel matrix)。这里对符号 K 进行了重复使用，既指代核函数 $K(x, z)$，也指代核矩阵 K，因为这两者有非常明显的关系。

如果 K 是一个有效的核，那么就有:

$$K_{ij} = K(x^{(i)}, x^{(j)}) = \phi(x^{(i)})^T\phi(x^{(j)}) = \phi(x^{(j)})^T\phi(x^{(i)}) = K(x^{(j)}, x^{(i)}) = K_{ji}$$

这就说明 K 是一个对称矩阵。此外，设 $\phi_k(x)$ 表示向量 $\phi(x)$ 的第 k 个坐标值(k-th coordinate)。我们发现对任意的向量 z，我们有：

$$\begin{aligned} z^TKz &= \sum_i \sum_j z_i K_{ij} z_j \\ &= \sum_i \sum_j z_i \phi(x^{(i)})^T \phi(x^{(j)}) z_j \\ &= \sum_i \sum_j z_i \sum_k \phi_k(x^{(i)}) \phi_k(x^{(j)}) z_j \\ &= \sum_k \sum_i \sum_j z_i \phi_k(x^{(i)}) \phi_k(x^{(j)}) z_j \\ &= \sum_k \left(\sum_i z_i \phi_k(x^{(i)}) \right)^2 \\ &\geq 0 \end{aligned}$$

由于 z 是任意的，这说明矩阵 K 是半正定(positive semi-definite)矩阵。

这样，我们就证明了，如果 K 是一个有效的核，那么对应的核矩阵 $K \in \mathcal{R}^{m \times m}$ 是一个半正定矩阵。实际上，这不仅仅是必要条件，也是一个充分条件。我们也称一个有效的核为 Mercer kernel。

接下来，我们给出 Mercer 定理，很多教材对该定理的描述要更复杂一些，里面牵涉到 L^2 函数，但如果输入属性(input attributes)只在实数域 $\mathcal{R}^n$ 取值，那么这里给出的表述也是等价的。

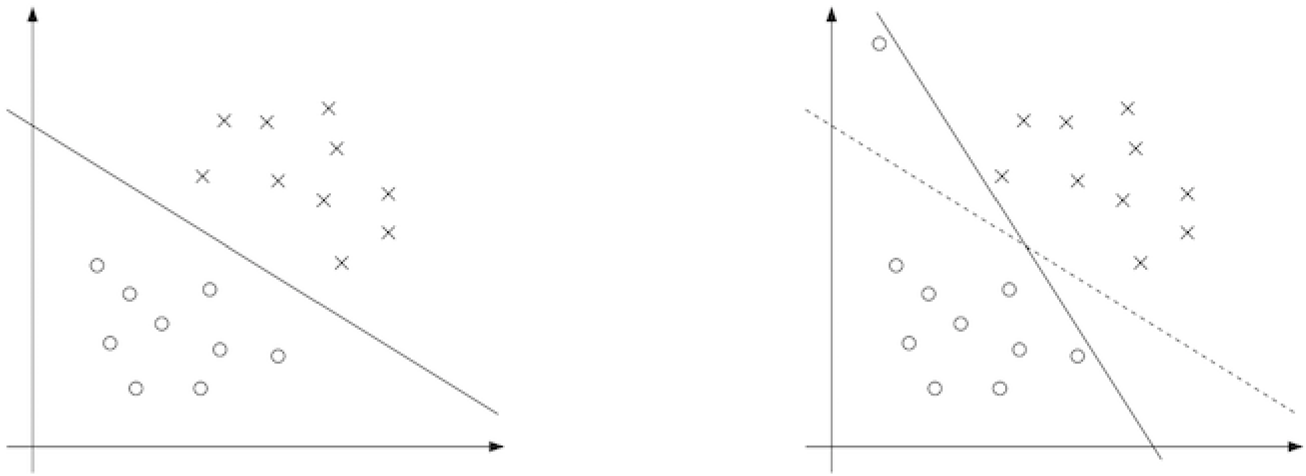
Theorem (Mercer): 给定函数 $K: \mathcal{R}^n \times \mathcal{R}^n \rightarrow \mathcal{R}$。要使 K 是一个有效的核(Mercer kernel)，其充分必要条件为：对任意的 $\{x^{(1)}, \cdots, x^{(m)}\}(m < \infty)$，都有对应的核矩阵为半正定矩阵。

当然，核技巧(kernel trick)用法远远不仅限于 SVM 算法。它在其他方面有着广泛的运用。

7. Regularization and the non-separable case

到目前为止，我们对 SVM 的推导都基于一个假设，就是我们假定训练集是线性可分的。虽然通过特征映射 ϕ 将数据映射到高维空间可以增加数据线性可分的概率（这个结论来自于 Cover 定理）。但是我们还是不能保证数据一直都是线性可分的。

同时由于数据中往往存在着噪音，对于这种偏离正常位置很远的点，我们称之为 outlier。在我们原来的 SVM 模型中，outlier 的存在可能会造成很大的影响。因为我们的超平面是由少数的支持向量决定的，如果这些支持向量里存在 outlier 的话，就会使得超平面出现显著偏移，还导致分类器的 margin 小了很多。如下图所示：



现在考虑 outlier 问题，约束条件变成了：

$$y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \quad i = 1, \cdots, m$$

其中，$\xi_i \geq 0$ 称为松弛变量(slack variable)，对应数据点允许偏离的函数间隔的量。当然，如果我们允许 ξ_i 任意大的话，那任意的超平面都是符合条件的了。所以，我们在原来的目标函数后面加上一项，使得这些 ξ_i 的总和也要最小：

$$\min_{w,b,\xi} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$$

其中 C 是一个超参数(hyper parameter)，用于控制目标函数中两项（”寻找 margin 最大的超平面“和“保证数据点偏差量最小”）之间的权重。注意，ξ 是需要优化的变量，而 C 是事先确定好的常量。因此，

$$
\begin{aligned}
\min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \\
s.t. \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \quad i = 1, \cdots, m \\
& \xi_i \geq 0, \quad i = 1, \cdots, m
\end{aligned}
$$

所以，我们得到新的拉格朗日函数如下：

$$
L(w,b,\xi,\alpha,\mu) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left(y^{(i)}(w^T x^{(i)} + b) - 1 + \xi_i\right) - \sum_{i=1}^{m}\mu_i\xi_i
$$

其中，$\alpha_i \geq 0, \mu_i \geq 0$

首先，我们求 $L(w,b,\xi,\alpha,\mu)$ 对 w, b, ξ 的极小，由

$$
\begin{aligned}
\nabla_w L(w,b,\xi,\alpha,\mu) &= w - \sum_{i=1}^{m}\alpha_i y^{(i)} x^{(i)} = 0 \\
\nabla_b L(w,b,\xi,\alpha,\mu) &= -\sum_{i=1}^{m}\alpha_i y^{(i)} = 0 \\
\nabla_{\xi_i} L(w,b,\xi,\alpha,\mu) &= C - \alpha_i - \mu_i = 0
\end{aligned}
$$

得

$$
\begin{aligned}
w = \sum_{i=1}^{m}\alpha_i y^{(i)} x^{(i)} \\
\sum_{i=1}^{m}\alpha_i y^{(i)} = 0 \\
C - \alpha_i - \mu_i = 0
\end{aligned}
$$

将上面三个式子代入 $L(w,b,\xi,\alpha,\mu)$ 得：

$$
\begin{aligned}
\min_{w,b,\xi} L(w,b,\xi,\alpha,\mu) &= \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left(y^{(i)}(w^T x^{(i)} + b) - 1 + \xi_i\right) - \sum_{i=1}^{m}\mu_i\xi_i \\
&= \frac{1}{2}w^T w - w^T\sum_{i=1}^{m}\alpha_i y^{(i)} x^{(i)} - b\sum_{i=1}^{m}\alpha_i y^{(i)} + \sum_{i=1}^{m}\alpha_i \\
&- \sum_{i=1}^{m}\alpha_i\xi_i - \sum_{i=1}^{m}\mu_i\xi_i + \sum_{i=1}^{m}C\xi_i
\end{aligned}
$$

注意到，$\sum_{i=1}^{m}\alpha_i y^{(i)} = 0$ 和 $C = \alpha_i + \mu_i$，所以

$$
\begin{aligned}
\min_{w,b,\xi} L(w,b,\xi,\alpha,\mu) &= \frac{1}{2}w^T w - w^T \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} + \sum_{i=1}^{m} \alpha_i \\
&= \frac{1}{2}w^T \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} - w^T \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} + \sum_{i=1}^{m} \alpha_i \\
&= -\frac{1}{2}w^T \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} + \sum_{i=1}^{m} \alpha_i \\
&= -\frac{1}{2}\left(\sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}\right)^T \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} + \sum_{i=1}^{m} \alpha_i \\
&= -\frac{1}{2} \sum_{i=1}^{m} \alpha_i y^{(i)} (x^{(i)})^T \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} + \sum_{i=1}^{m} \alpha_i \\
&= -\frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} < x^{(i)}, x^{(j)} > + \sum_{i=1}^{m} \alpha_i
\end{aligned}
$$

再对 $\min_{w,b,\xi} L(w,b,\xi,\alpha,\mu)$ 求 α 的极大，得到对偶问题：

$$
\begin{aligned}
\max_{\alpha} \quad & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} < x^{(i)}, x^{(j)} > \\
s.t. \quad & \sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \\
& C - \alpha_i - \mu_i = 0 \\
& \alpha_i \geq 0 \\
& \mu_i \geq 0, \quad i = 1, \cdots, m
\end{aligned}
$$

利用等式约束 $C - \alpha_i - \mu_i = 0$ 消去 μ_i，从而只留下 α_i，于是得到：

$$
\begin{aligned}
\max_{\alpha} \quad & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} < x^{(i)}, x^{(j)} > \\
s.t. \quad & \sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \\
& 0 \leq \alpha_i \leq C, \quad i = 1, \cdots, m
\end{aligned}
$$

和之前相同，我们可以用 $\sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}$ 来表示 w。但对于 b^* 来说，我们不能直接用之前的公式，这里 b 的解不唯一，它存在于一个区间。

最后，我们来看一下 KKT 对偶互补条件（可以用来测试 SMO 算法的收敛性）：

$$
\begin{aligned}
\alpha_i^* \left(y^{(i)}(w^{*T} x^{(i)} + b^*) - 1 + \xi_i^*\right) &= 0 \\
\mu_i^* \xi_i^* &= 0
\end{aligned}
$$

因为 $\mu_i^* = C - \alpha_i^*$，因此

$$
\begin{aligned}
\alpha_i^* \left(y^{(i)}(w^{*T}x^{(i)} + b^*) - 1 + \xi_i^* \right) &= 0 \\
(C - \alpha_i^*)\xi_i^* &= 0
\end{aligned}
$$

所以，我们可以得到（注意 $\xi_i \geq 0$）：

$$
\begin{aligned}
\alpha_i^* &= 0 \quad &\Longleftrightarrow \quad & y^{(i)}(w^{*T}x^{(i)} + b) \geq 1 \\
\alpha_i^* &= C \quad &\Longleftrightarrow \quad & y^{(i)}(w^{*T}x^{(i)} + b) \leq 1 \\
0 < \alpha_i^* &< C \quad &\Longleftrightarrow \quad & y^{(i)}(w^{*T}x^{(i)} + b) = 1
\end{aligned}
$$

接下来，我们介绍 SMO 算法来求解该问题。

参考资料：

1. 吴恩达，cs229 讲义
2. 林轩田，机器学习技法
3. 李航，《统计学习方法》
4. July、pluskid，《支持向量机通俗导论（理解 SVM 的三层境界）》

In [ ]:

```

```