统计机器学习
Statistical Machine Learning

魏莱

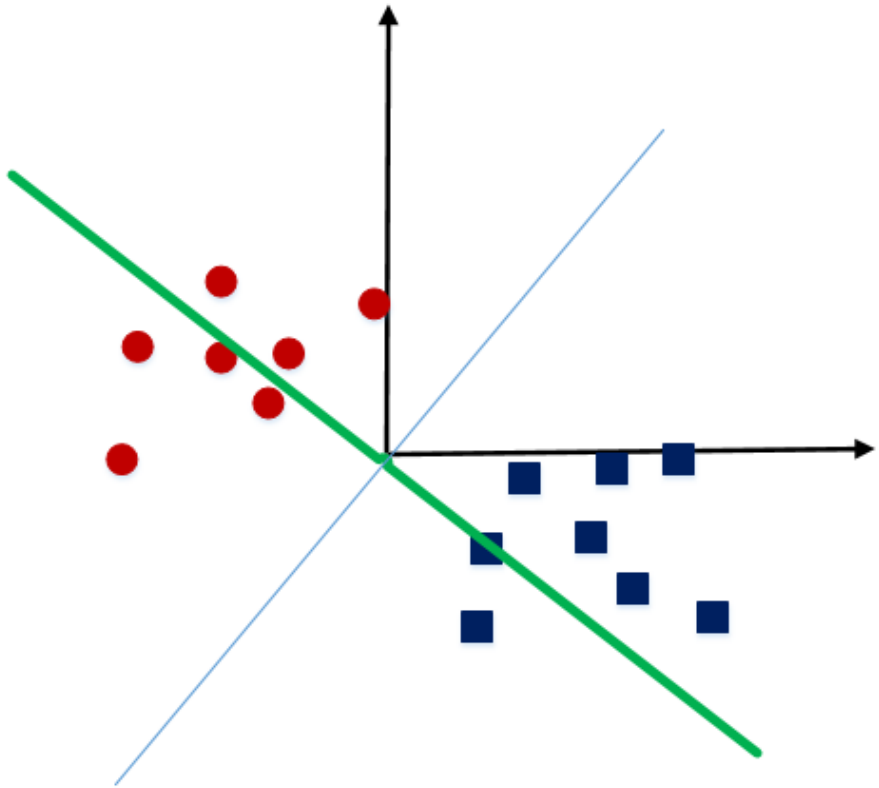
上海海事大学信息工程学院

2019 年 4 月 4 日

第六章：支撑向量机

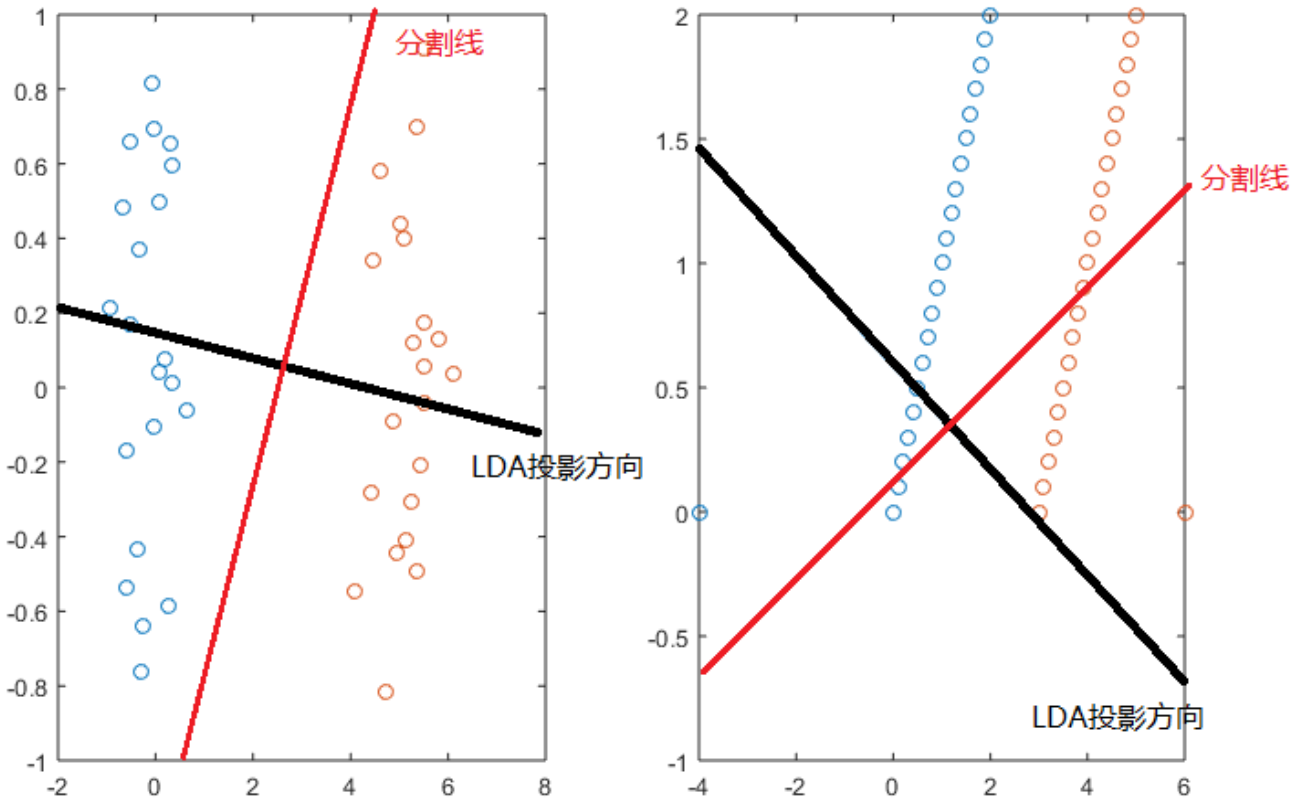
1. 间隔与支撑向量

在线性判别分析中，我们讨论过，如何来找到一个最优的分割线（超平面）来将两类样本分开的问题。



1. 间隔与支撑向量

观察下面两组数据集，通过 LDA 计算得到的投影方向以及由此得到的分割结果：



结论：LDA 适用于满足高斯分布（正态分布）的数据集。那么对于不满足高斯分布的数据集（第二组），如何找到最合理的分割超平面？

1. 间隔与支撑向量

假设数据集 $\mathbf{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n)\}$，考虑二分类问题，令 $y_i \in \{-1, +1\}$。在样本空间中，划分的超平面可以表示为:

$$\mathbf{w}^t\mathbf{x} + b = 0, \tag{1}$$

其中 $\mathbf{w} = (w_1, w_2, \cdots, w_d)^t$ 为超平面的法向量，b 为位移，决定超平面与原点之间的距离。很显然，超平面被 $\mathbf{w}$ 和 b 唯一决定。

样本空间中一数据点 $\mathbf{x}$ 到超平面 $\mathbf{w}^t\mathbf{x}+b=0$ 的距离可以写为：

$$r=\frac{|\mathbf{w}^t\mathbf{x}+b|}{\|\mathbf{w}\|_2}. \tag{2}$$

现在假设超平面 $\mathbf{w}^t\mathbf{x}+b=0$ 能将所有训练样本正确分类，即满足:

$$\begin{cases} \mathbf{w}^t\mathbf{x}_i+b>0, & y_i=+1; \\ \mathbf{w}^t\mathbf{x}_i+b<0, & y_i=-1. \end{cases} \tag{3}$$

由此可以假设，存在正实数 δ，使得下式成立：

$$\begin{cases} \mathbf{w}^t\mathbf{x}_i + b > +\delta, & y_i = +1; \\ \mathbf{w}^t\mathbf{x}_i + b < -\delta, & y_i = -1. \end{cases} \tag{4}$$

于是，可以想象存在两个超平面 $\mathbf{w}^t\mathbf{x} + b = \pm\delta$ 将数据样本分割开。令 $\tilde{\mathbf{w}} = \mathbf{w}/\delta, \tilde{b} = b/\delta$，则上式又变成

$$\begin{cases} \tilde{\mathbf{w}}^t\mathbf{x}_i + \tilde{b} > +1, & y_i = +1; \\ \tilde{\mathbf{w}}^t\mathbf{x}_i + \tilde{b} < -1, & y_i = -1. \end{cases} \tag{5}$$

为了书写的方便，我们再将 $\tilde{\mathbf{w}}, \tilde{b}$，写成 $\mathbf{w}, b$。所以：

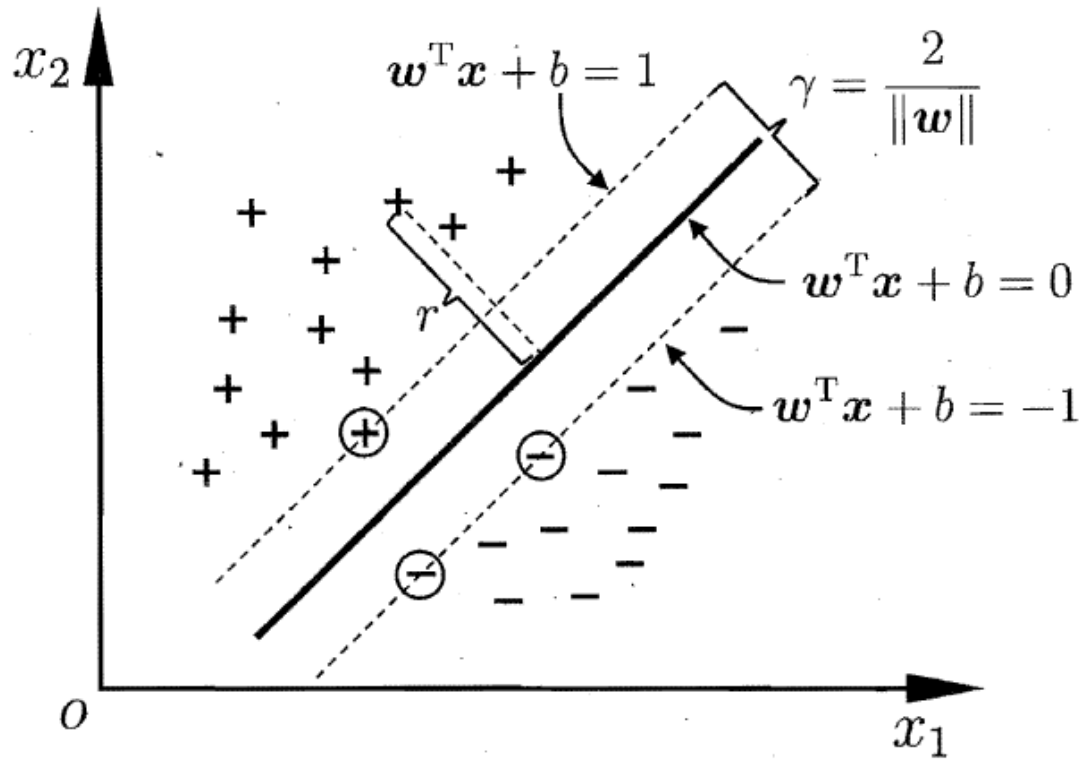
$$\begin{cases} \mathbf{w}^t\mathbf{x}_i + b > +1, & y_i = +1; \\ \mathbf{w}^t\mathbf{x}_i + b < -1, & y_i = -1. \end{cases} \tag{6}$$

1. 间隔与支撑向量

由此，可以计算两个超平面之间的距离为：

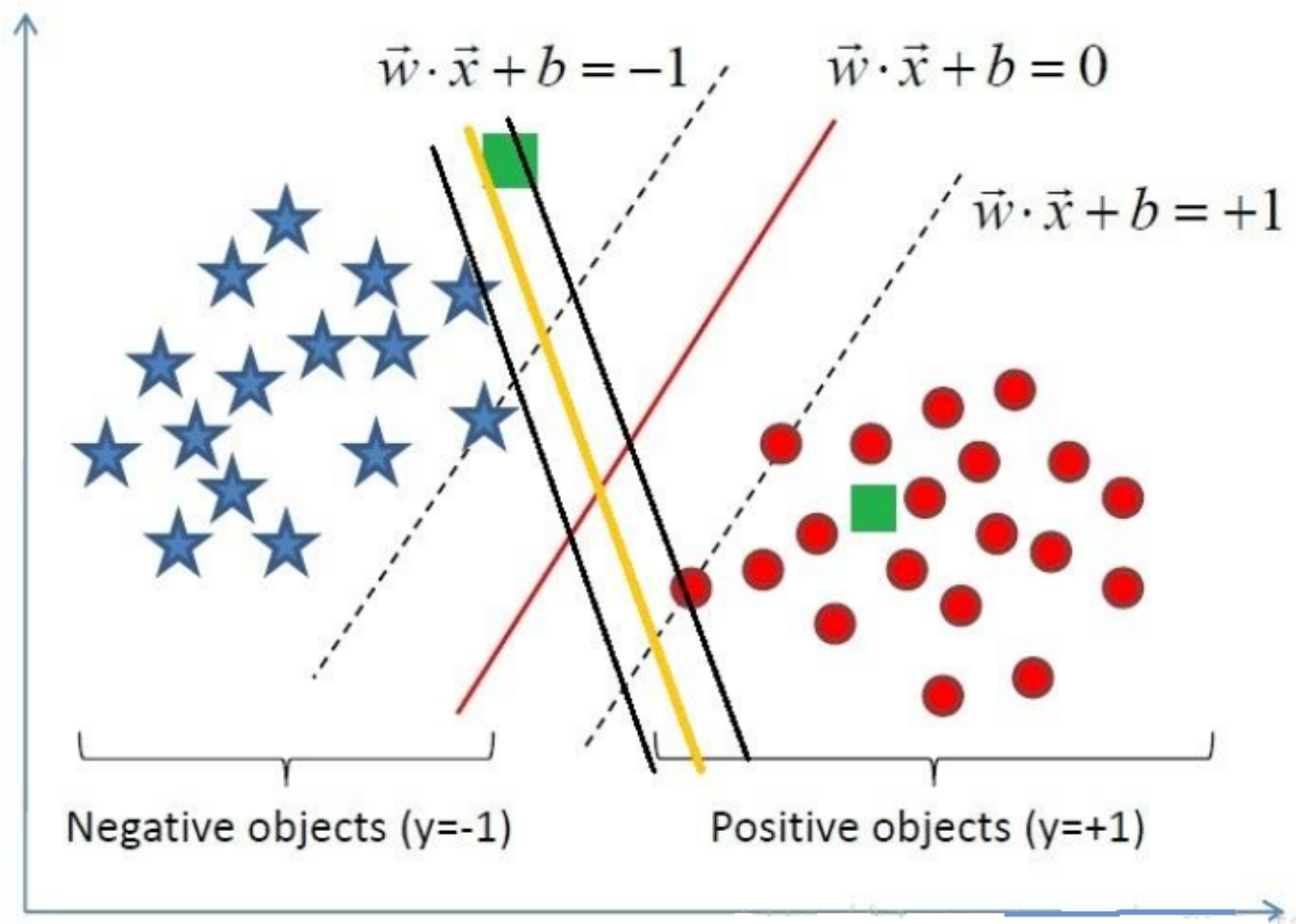
$$\gamma = \frac{2}{\|\mathbf{w}\|_2}, \tag{7}$$

其被称为“间隔”（margin）。下图显示了两个超平面以及由此得到的分割超平面



1. 间隔与支撑向量

而在某些问题中，我们可以找到几组这样的超平面，如下图所示。那么哪一组是最好的？



很显然，间隔最大的哪一组应该是最好的，由此计算得到的分割超平面也是最优的。那么如何找到间隔最大的划分超平面？可以定义如下问题：

$$\begin{aligned} &\max_{\mathbf{w},b} && \frac{2}{\|\mathbf{w}\|_2} \\ &s.t. && y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1, i = 1, 2, \cdots, n \end{aligned} \tag{8}$$

很显然，上式问题与下式问题等价:

$$\begin{aligned} &\min_{\mathbf{w},b} && \frac{1}{2}\|\mathbf{w}\|_2^2 \\ &s.t. && y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1, i = 1, 2, \cdots, n \end{aligned} \tag{9}$$

而上式问题，就是著名的**支撑向量机（support vector machine，SVM）**的基本型。

通过求解上式模型中的 $\mathbf{w}, b$，就可以获得分割超平面。式（9）实际上是一个二次凸规划问题 (convex quadratic programming)。可以直接用一些优化软件计算包来求解 (CVX，http://cvxr.com/cvx/)，但该问题可以有更为高效的求解方法。

2. 对偶问题

首先，带约束的优化问题的通用解法为 Lagrange 乘子法。因此对式（9），每条约束添加 Lagrange 乘子 $a_i \geq 0$，则可以得到 Lagrange 函数：

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{n} a_i\big(1 - y_i(\mathbf{w}^t\mathbf{x}_i + b)\big) \tag{10}$$

对 $\mathbf{w}, b$ 求偏导，并令导数等于 0，则可以得到：

$$\mathbf{w} = \sum_{i=1}^{n} a_i y_i \mathbf{x}_i \tag{11}$$

$$0 = \sum_{i=1}^{n} a_i y_i \tag{12}$$

将（11）式代入（10），可以将 $\mathbf{w}, b$ 消去，再考虑（12）式的约束，就得到如下问题：

$$\begin{aligned} \max_{\mathbf{a}} \quad & \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j \\ s.t. \quad & \sum_{i=1}^{n} a_i y_i = 0 \\ & a_i \geq 0, i = 1, 2, \cdots, n \end{aligned} \tag{13}$$

该问题称为原支撑向量机问题的对偶问题（dual problem）。现在假设 $\mathbf{a}$ 已经求解得到，那么由于 $\mathbf{w} = \sum_{i=1}^{n} a_i y_i \mathbf{x}_i$，因此可以计算得到 $\mathbf{w}$。如何求解 b，暂时先放一下。

而上式求解出的 a_i 实际上是原问题的 Lagrange 乘子，每一个 a_i 对应着训练样本 $(\mathbf{x}_i, y_i)$。再观察原问题的约束，即 $y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1$。因此，把原问题以及对偶问题的所有约束项合在一起，可以得到:

$$\begin{cases} a_i \geq 0 \\ y_i f(\mathbf{x}_i) - 1 \geq 0 \end{cases}$$

此外，由于 a_i 为对偶问题的解，同时 $\mathbf{w}, b$ 为原问题的解，因此还满足 $a_i(y_i f(\mathbf{x}_i) - 1) = 0$。这三个约束一起称为 **KKT(Karush-Kuhn-Tucker) 条件**。

于是对任意训练样本 $(\mathbf{x}_i, y_i)$，总有 $a_i = 0$ 或者 $y_i f(\mathbf{x}_i) = 1$。

1. 如果 $a_i = 0$，则因为 $\mathbf{w} = \sum_{i=1}^{n} a_i y_i \mathbf{x}_i$，所以样本 $(\mathbf{x}_i, y_i)$ 对计算分割超平面没有影响；

2. 如果 $y_i f(\mathbf{x}_i) = 1 \Rightarrow \mathbf{w}^t \mathbf{x}_i - b = 1(/-1)$，说明样本出现在分割边界上。该样本对计算分割超平面有影响。这样的数据样本称为**支撑向量（support vectors）**。

上面两点说明，在支撑向量机计算过程中，大部分训练样本都不需要，最终模型仅与支撑向量有关。

最后再回到求解 b 的问题上来，由于对任意支撑向量有 $y_i(\mathbf{w}^t\mathbf{x}_i + b) = 1$，由此可以得到，$b = y_i - \mathbf{w}^t\mathbf{x}_i$。为了增强鲁棒性，常常将所有支撑向量计算得到的 b 的均值记为最优值，即 $b = \sum_{i\in\mathbf{S}} \frac{1}{|\mathbf{S}|} \sum_{i\in\mathbf{S}}(y_i - \mathbf{w}^t\mathbf{x}_i)$。这里 $\mathbf{S} \subseteq \mathbf{D}$ 为原数据集中所有支撑向量组成的子集。

3. 序贯最小化算法 (Sequential Minimal Optimization，SMO)

由上述推理可以发现，只要求解出对偶问题中的 $\mathbf{a}$，那么支撑向量机就可以唯一确定。现在回到问题（13）来讨论求解该问题。求解支撑向量机原问题的对偶问题，有很多种解法，其中最为著名的就是 **SMO 算法**。

3. 序贯最小化算法 (Sequential Minimal Optimization，SMO)

1. SMO 基本思路：
   a. 选择一个 $\mathbf{a}$ 中的两个分量 a_i, a_j，固定除他们以外的所有分量；
   b. 利用梯度下降法更新 a_i, a_j，转 a。

3. 序贯最小化算法 (Sequential Minimal Optimization，SMO)

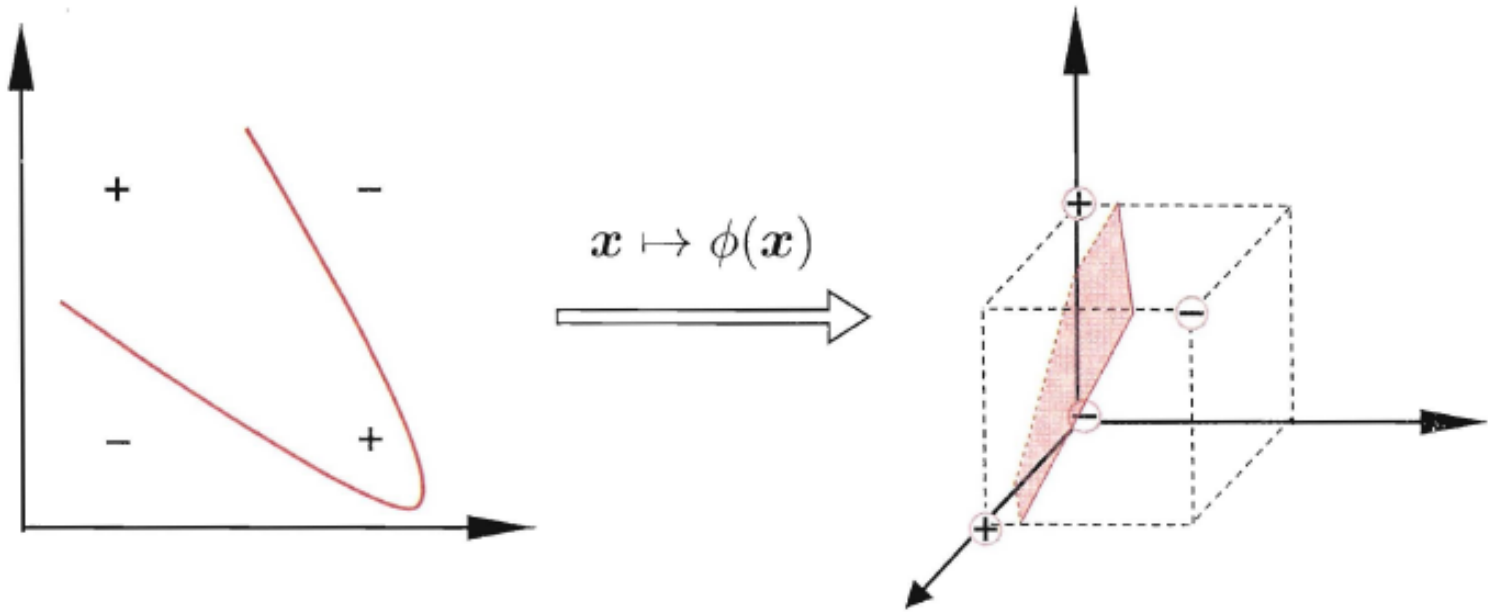
2. SMO 变量选择策略：
   a. 由于最终所有计算得到的 a_i 都会满足 KKT 条件，因此如果存在某个 a_i 不满足 KKT 条件，那么目标函数 $\sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j$ 会最快衰减。因此，应该选择违背 KKT 条件的那些变量先更新。
   b. 比较各变量所对应的目标函数值减幅的复杂度过高，因此 SMO 采用了一个启发式: 使选取的两变量所对应样本之间的间隔最大。一种直观的解释是，这样的两个变量有很大的差别，与对两个相似的变量进行更新相比，对它们进行更新会带给目标函数值更大的变化。

3. 序贯最小化算法 (Sequential Minimal Optimization，SMO)

3. SMO 效率：
   a. 固定其他变量后优化 a_i, a_j，实际上是将原约束项 $\sum_{i=1}^{n} a_i y_i = 0$ 转变成 $a_i y_i + a_j y_j = c$，这里 $c = -\sum_{k \neq i,j} a_k y_k$。而进一步，$a_j = c y_j - a_i y_i y_j$
   b. 于是式（13）变成关于 a_i 的一个二次规划问题，存在闭式解，因此算法效率较高。

4. 核函数

前面谈论假设训练样本是线性可分的，即存在一个划分超平面能将训练样本正确分类。然而在现实任务中，原始样本空间内也许并不存在一个能正确划分两类样本的超平面，如下左图：



回忆我们曾经在线性模型中提到过的，**低维空间中的非线性问题可以转变成高维空间中的线性问题**。因此解决上面问题的一个方法就是将样本从原始空间映射到一个更高维的特征空间，使得样本在这个特征空间内线性可分。那么如何找到这个高维空间？假设 $\phi(\mathbf{x})$ 为样本 $\mathbf{x}$ 在高维空间中的映射，由此，高维空间中的划分超平面可以表示成：

$$f(\mathbf{x}) = \mathbf{w}^t \phi(\mathbf{x}) + b \tag{14}$$

依据前面的推导，可以得知高维空间中的支撑向量机模型为

$$\begin{aligned} &\min_{\mathbf{w},b} && \frac{1}{2}\|\mathbf{w}\|_2^2 \\ &s.t. && y_i(\mathbf{w}^t\phi(\mathbf{x}_i)+b) \geq 1, i=1,2,\cdots,n \end{aligned} \tag{15}$$

相应的对偶问题为：

$$\begin{aligned} &\max_{\mathbf{a}} && \sum_{i=1}^n a_i - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n a_i a_j y_i y_j \phi(\mathbf{x}_i)^t\phi(\mathbf{x}_j) \\ &s.t. && \sum_{i=1}^n a_i y_i = 0 \\ & && a_i \geq 0, i=1,2,\cdots,n \end{aligned} \tag{16}$$

上式中 $\phi(\mathbf{x}_i)^t\phi(\mathbf{x}_j)$ 称为高维空间中样本 $\mathbf{x}_i, \mathbf{x}_j$ 的内积。

4. 核函数

直接计算 $\phi(\mathbf{x}_i)^t\phi(\mathbf{x}_j)$ 非常困难，因为高维空间可能维数非常高（甚至是无穷维的），为了避开这个障碍，可以设想样本之间的内积运算由某个函数定义，即：

$$\phi(\mathbf{x}_i)^t\phi(\mathbf{x}_j) = \langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle = k(\mathbf{x}_i, \mathbf{x}_j) \tag{17}$$

其中，$k(\cdot,\cdot)$ 称为**核函数（kernel function）**。

有了这样的核函数，我们可以不用去显示的定义映射函数：ϕ。同时样本 $\mathbf{x}_i, \mathbf{x}_j$ 在高维空间中的内积可以直接由核函数计算得到。于是，式（16）可以转变成:

$$\begin{aligned} \max_{\mathbf{a}} \quad & \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\ s.t. \quad & \sum_{i=1}^{n} a_i y_i = 0 \\ & a_i \geq 0, i = 1, 2, \cdots, n \end{aligned} \tag{18}$$

最后计算得到分割超平面 $f(\mathbf{x}) = \sum_{i=1}^{n} a_i y_i k(\mathbf{x}_i, \mathbf{x}_j) + b$。这里表面出模型最优解可通过训练样本的核函数展开，这一展式亦称**“支持向量展式”(support vector expansion)**。

4. 核函数

那么，核函数 $k(\cdot,\cdot)$ 又具有什么样的形式呢？核函数是否一定存在呢？

Mercer 定理：任何半正定的函数都可以作为核函数。也就是说只要一个对称函数所对应的核矩阵半正定，它就能作为核函数使用。

假设数据集 $\mathbf{D}=\{\mathbf{x}_1,\cdots,\mathbf{x}_n\}$，所谓的核矩阵即满足下式的矩阵：

$$\mathbf{K}=\begin{pmatrix} k(\mathbf{x}_1,\mathbf{x}_1) & \cdots & k(\mathbf{x}_1,\mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n,\mathbf{x}_1) & \cdots & k(\mathbf{x}_n,\mathbf{x}_n) \end{pmatrix}$$

4. 核函数

事实上，对于一个半正定核矩阵，总能找到一个与之对应的映射。换言之，任何一个核函数都隐式地定义了一个称为“**再生核希尔伯特空间**”**(Reproducing Kernel Hilbert Space, RKHS)**的特征空间。

通过前面的讨论可知，我们希望样本在特征空间内线性可分，因此特征空间的好坏对支持向量机的性能至关重要. 需注意的是，在不知道特征映射的形式时，我们并不知道什么样的核函数是合适的，而核函数也仅是隐式地走义了这个特征空间。于是，“核函数选择”成为支持向量机的最大变数。**若核函数选择不合适，则意味着将样本映射到了一个不合适的特征空间，很可能导致 SVM 性能不佳。**

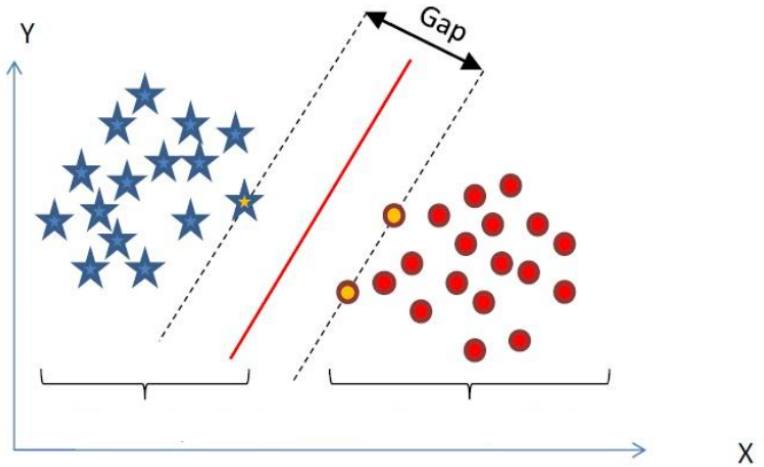
几种常用核函数

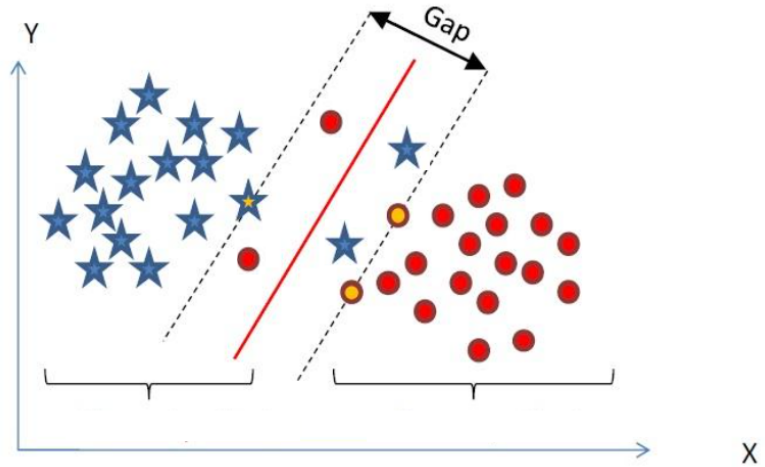
名称	表达式	参数
多项式核	$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^t \mathbf{x}_j)^d$	$d \geq 1$
高斯核	$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\delta^2})$	$\delta > 0$ 为带宽（bandwidth）
拉普拉斯核	$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{2\delta^2})$	$\delta > 0$
Sigmoid 核	$k(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta \mathbf{x}_i^t \mathbf{x}_j + \theta)$	tanh 为双曲正切函数，$\beta > 0, \theta < 0$

此外通过核函数的线性组合、内积运算也能得到新的核函数，特别的：若 k 为核函数，对于任意函数 $g(\mathbf{x})$，$\tilde{k}(\mathbf{x}, \mathbf{z}) = g(\mathbf{x}) k(\mathbf{x}, \mathbf{z}) g(\mathbf{z})$ 也是核函数。

5. 软间隔

在前面讨论的过程中，我们都是假设不同类的数据样本能够完全被分割开来（原空间或者映射后的特征空间），但对于有些数据集，由于噪音的存在，不同类样本可能并不能够完全被超平面分割开来。观察下面两幅图，右侧图中数据集有很少量的样本不能被超平面分隔开，但对于大多数样本分割超平面都是合适的。





5. 软间隔

对于这样的问题，首先，并不能找到完全能够将不同类样本分割开来的超平面；其次，如果通过多个超平面组合来进行分割，那么多个超平面组合得到的模型将可能过于复杂，即过分的适合现有数据样本，对将来可能出现的样本并不太适合（过拟合）。因此容忍一定的样本被错分，可能使得求得的超平面具有更好的泛化性能。

5. 软间隔

那么如何“容忍一定的样本被错分”？考虑前面 SVM 中，分割准确性的约束：$y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1$。如果有部分样本被错分了，那么大于等于号将不一定对每个样本都成立。但我们可以对每一个样本设定一个参数 $\varepsilon_i > 0$，使得 $y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1 - \varepsilon_i$ 成立。这个 ε_i 可以表示错分的程度，称为**松弛变量（slack variables）**。

另外，我们当然希望错分的样本越少越好，错分的程度越低越好，因此 SVM 模型可以重新改写成：

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^n \varepsilon_i \\ s.t. \quad & y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1 - \varepsilon_i \\ & \varepsilon_i \geq 0, i = 1, 2, \cdots, n \end{aligned} \tag{19}$$

这里，$C > 0$ 是一个参数，这就是常用的“软间隔支持向量机”。

同样通过 Lagrange 乘子法，可以得到软间隔支持向量机的 Lagrange 函数：

$$L(\mathbf{w}, b, \mathbf{a}, \varepsilon, \mu) = \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\varepsilon_i + \sum_{i=1}^{n} a_i(1-\varepsilon_i - y_i(\mathbf{w}^t\mathbf{x}_i + b)) - \sum_{i=1}^{n}\mu_i\varepsilon_i \tag{20}$$

$a_i \geq 0, \mu_i \geq 0$ 为 Lagrange 乘子。上式分别对 $\mathbf{w}, b, \varepsilon_i$ 求导，可得：

$$\begin{aligned}
\mathbf{w} &= \textstyle\sum_{i=1}^{n} a_i y_i \mathbf{x}_i \\
0 &= \textstyle\sum_{i=1}^{n} a_i y_i \\
C &= a_i + \mu_i
\end{aligned}$$

将上面求得的导数代入 Lagrange 函数，得到对偶问题

$$\begin{aligned} \max_{\mathbf{a}} \quad & \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j y_i y_j \mathbf{x}_i^t \mathbf{x}_j \\ s.t. \quad & \sum_{i=1}^{n} a_i y_i = 0 \\ & 0 \leq a_i \leq C, i = 1, 2, \cdots, n \end{aligned} \tag{21}$$

综合约束项，最后得到 KKT 条件

$$\begin{cases} 0 \leq a_i \leq C, \mu_i \geq 0, \\ y_i f(\mathbf{x}_i) - 1 + \varepsilon_i \geq 0, \\ a_i (y_i f(\mathbf{x}_i) - 1 + \varepsilon_i) = 0, \\ \varepsilon_i \geq 0, \mu_i \varepsilon_i = 0 \end{cases}$$

分析：

1. 对于任意样本总有 $a_i = 0$ 或者 $y_i f(\mathbf{x}_i) = 1 + \varepsilon_i$;，若 $a_i = 0$，则该样本对计算分割超平面无影响，否则该样本为支撑向量；
2. 若 $a_i > 0$，如果有 $\varepsilon_i = 0$，则该样本在最大间隔边界上；如果有 $0 < \varepsilon_i \leq 1$，则该样本在最大间隔内部；若 $\varepsilon_i > 1$，则该样本被错分。

6. 支持向量回归

在上面讨论过程中，我们说 ε_i 代表了错分的程度，满足 $y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1 - \varepsilon_i$。如果，令 $\varepsilon_i = g(1 - y_i(\mathbf{w}^t\mathbf{x}_i + b))$，$g(\cdot)$ 为某种损失函数。则软间隔 SVM 的目标函数可以写成

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n} g(1 - y_i(\mathbf{w}^t\mathbf{x}_i + b)) \tag{22}$$

按照前面讨论，$\varepsilon_i = 1 - y_i(\mathbf{w}^t\mathbf{x}_i + b) > 0$ 的样本有可能被错分（如果有 $0 < \varepsilon_i \leq 1$，则该样本在最大间隔内部；若 $\varepsilon_i > 1$，则该样本被错分）。则可以定义:

$$g(z) = \begin{cases} 1, & z > 0; \\ 0, & \text{otherwize.} \end{cases} \tag{23}$$

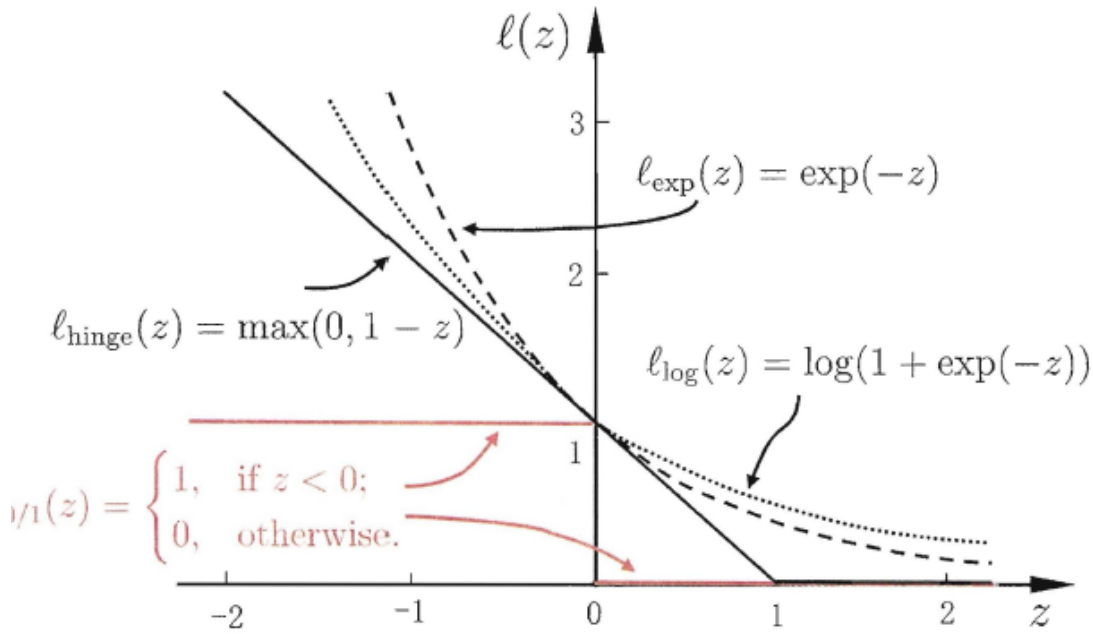
6. 支持向量回归

上式定义的 $g(\cdot)$ 非凸、非连续，数学性质不太好。常常用一些**替代损失 (surrogate loss)**函数来代替，如:

hinge 损失 $g_{hinge}(z) = \max(0, 1 - z)$;

指数损失 $g_{exp}(z) = \exp(-z)$

对数几率损失 $g_{log}(z) = \log(1 + \exp(-z))$



6. 支持向量回归

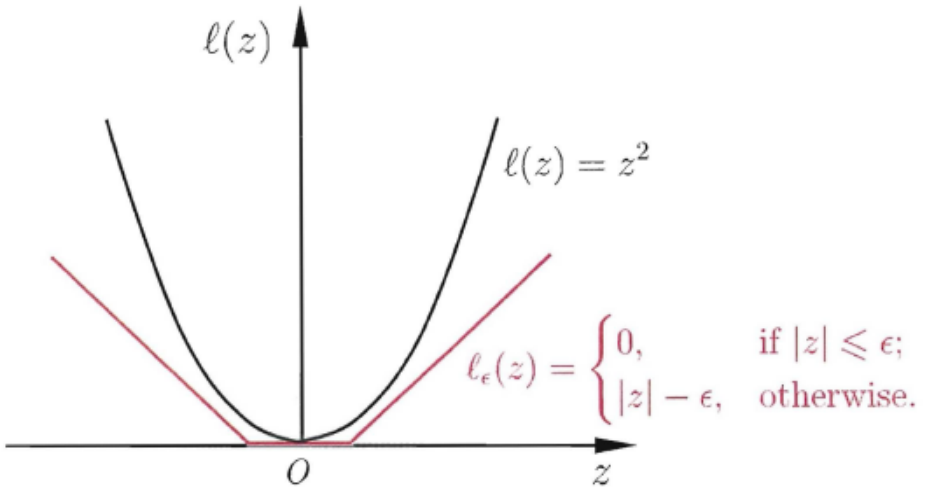
有了损失函数，我们可以从损失角度来重新给出软间隔 SVM 的目标函数，即：

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n} g(f(\mathbf{x}_i) - y_i)) \tag{24}$$

考虑上式与线性模型之间的关系。

定义新的 $\epsilon-$ 不敏感损失函数：

$$g(z) = \begin{cases} 0, & \text{if } |z| < \epsilon; \\ |z| - \epsilon, & \text{otherwise.} \end{cases}$$

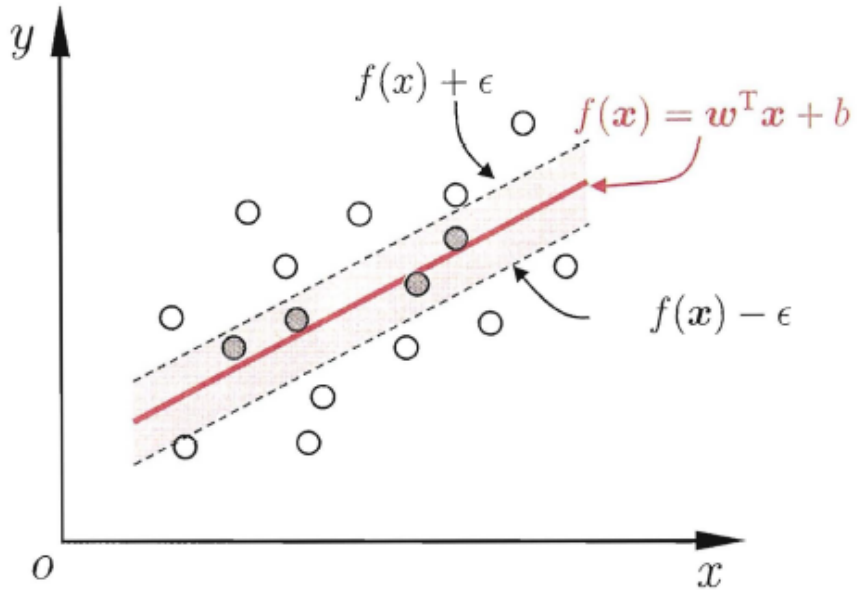


6. 支持向量回归

再引入松弛变量 $\varepsilon_i, \tilde{\varepsilon}_i$，最后可以得到

$$\begin{aligned} \min_{\mathbf{w},b,\varepsilon_i,\tilde{\varepsilon}_i} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}(\varepsilon_i + \tilde{\varepsilon}_i) \\ s.t. \quad & f(\mathbf{x}_i) - y_i \leq \epsilon + \varepsilon_i \\ & y_i - f(\mathbf{x}_i) \leq \epsilon + \tilde{\varepsilon}_i \\ & \varepsilon_i \geq 0, \tilde{\varepsilon}_i \geq 0, i = 1, 2, \cdots, n \end{aligned} \tag{25}$$

上式就是**支持向量回归 (Support Vector Regression，SVR)**。



按照前面的方法，引入 Lagrange 乘子 $\mu_i \geq 0, \tilde{\mu}_i \geq 0, a_i \geq 0, \tilde{a}_i \geq 0$，可以的得到上式的 Lagrange 函数

$$\begin{aligned} & L(\mathbf{w}, b, \mathbf{a}, \tilde{\mathbf{a}}, \varepsilon, \tilde{\varepsilon}, \mu, \tilde{\mu}) \\ & = \tfrac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^n(\varepsilon_i + \tilde{\varepsilon}_i) - \sum_{i=1}^n \mu_i\varepsilon_i - \sum_{i=1}^n \tilde{\mu}_i\tilde{\varepsilon}_i \\ & + \sum_{i=1}^n a_i(f(\mathbf{x}_i - y_i - \epsilon - \varepsilon_i) + \sum_{i=1}^n \tilde{a}_i(f(\mathbf{x}_i - y_i - \epsilon - \tilde{\varepsilon}_i)) \end{aligned} \tag{26}$$

上式分别对 $\mathbf{w}, b, \varepsilon_i, \tilde{\varepsilon}_i$ 求导，可得

$$\begin{aligned} \mathbf{w} &= \sum_{i=1}^n(a_i - \tilde{a}_i)\mathbf{x}_i \quad & 0 &= \sum_{i=1}^n(a_i - \tilde{a}_i) \\ C &= a_i + \mu_i & C &= \tilde{a}_i + \tilde{\mu}_i \end{aligned} \tag{27}$$

将等式（27）完全代入等式（26），可得 SVR 的对偶问题：

$$\begin{aligned} \max_{\mathbf{a},\tilde{\mathbf{a}}} \quad & \sum_{i=1}^{n} y_i(\tilde{a}_i - a_i) - \epsilon((a_i + \tilde{a}_i)) \\ & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\tilde{a}_i - a_i)(\tilde{a}_j - a_j)\mathbf{x}_i^t\mathbf{x}_j \\ s.t. \quad & \sum_{i=1}^{n}(\tilde{a}_i - a_i) = 0 \\ & 0 \le a_i, \tilde{a}_i \le C \end{aligned} \tag{28}$$

上述过程中，求得的最优解需满足 KKT 条件，即要求:

$$\begin{cases} a_i(f(\mathbf{x}_i - y_i - \epsilon - \varepsilon_i) = 0, \\ \tilde{a}_i(f(\mathbf{x}_i - y_i - \epsilon - \tilde{\varepsilon}_i) = 0, \\ a_i\tilde{a}_i = 0, \varepsilon_i\tilde{\varepsilon}_i = 0 \\ (C - a_i)\varepsilon_i = 0, (C - \tilde{a}_i)\tilde{\varepsilon}_i = 0 \end{cases} \tag{29}$$

分析：

1. 当且仅当，$f(\mathbf{x}_i) - y_i - \epsilon - \varepsilon_i = 0$，$a_i$ 可以取非零值，而当且仅当，$f(\mathbf{x}_i) - y_i - \epsilon - \tilde{\varepsilon}_i = 0$，$\tilde{a}_i$ 可以取非零值，而且 $f(\mathbf{x}_i) - y_i - \epsilon - \varepsilon_i = 0$ 和 $f(\mathbf{x}_i) - y_i - \epsilon - \tilde{\varepsilon}_i = 0$ 不能同时成立；
2. 而如果 $f(\mathbf{x}_i) - y_i - \epsilon - \varepsilon_i = 0$ 或者 $f(\mathbf{x}_i) - y_i - \epsilon - \tilde{\varepsilon}_i = 0$，说明样本在 ϵ 间隔之外，因为 $\varepsilon_i, \tilde{\varepsilon}_i \geq 0$，所以 $f(\mathbf{x}_i) - y_i - \epsilon \geq 0$；
3. 根据计算结果 $\mathbf{w} = \sum_{i=1}^{n}(a_i - \tilde{a}_i)\mathbf{x}_i$，可知只有在 ϵ 间隔之外的样本才对 SVR 计算得到的超平面有影响。

最后将 $\mathbf{w}=\sum_{i=1}^{n}(a_i-\tilde{a}_i)\mathbf{x}_i$ 代入 $f(\mathbf{x})=\mathbf{w}^t\mathbf{x}+b$，可得 SVR 的解形式为：

$$f(\mathbf{x})=\sum_{i=1}^{n}(a_i-\tilde{a}_i)\mathbf{x}_i^t\mathbf{x}+b \tag{30}$$

此外，当样本落在 ϵ 间隔之外，则 $0<a_i,\tilde{a}_i<C$，而根据 KKT 条件,$(C-a_i)\varepsilon_i=0,(C-\tilde{a}_i)\tilde{\varepsilon}_i=0$，则可知 $\varepsilon_i=\tilde{\varepsilon}_i=0$，于是可得 $f(\mathbf{x}_i)-y_i-\epsilon=0$，因此，

$$b=y_i+\epsilon-\mathbf{w}^t\mathbf{x}_i=y_i+\epsilon-\sum_{i=1}^{n}(a_i-\tilde{a}_i)\mathbf{x}_i^t\mathbf{x} \tag{31}$$

根据前面 SVM 中计算鲁棒性位移 b 的方法，可知在 SVR 中，可以将 ϵ 间隔之外的多个样本计算的位移均值作为最优的 b。

观察 SVR 的解 $f(\mathbf{x}) = \sum_{i=1}^{n}(a_i - \tilde{a}_i)\mathbf{x}_i^t\mathbf{x} + b$，可以发现其中具有样本内积 $\mathbf{x}_i^t\mathbf{x}$，根据前面的核化 SVM 方法，可以很容易得到核化的 SVR 解，即：

$$f(\mathbf{x}) = \sum_{i=1}^{n}(a_i - \tilde{a}_i)k(\mathbf{x}_i, \mathbf{x}) + b \tag{32}$$

7. LIBSVM

基于 Matlab 的高效 SVM 软件包：

LIBSVM–A Library for Support Vector Machines

https://www.csie.ntu.edu.tw/~cjlin/libsvm/

可以发现前面的核化 SVM（Kernel SVM）以及核化 SVR（Kernel SVR）的解具有类似的形式，

KSVM 解：$f(\mathbf{x}) = \sum_{i=1}^{n} a_i y_i k(\mathbf{x}_i, \mathbf{x}_j) + b$

KSVR 解：$f(\mathbf{x}) = \sum_{i=1}^{n} (a_i - \tilde{a}_i) k(\mathbf{x}_i, \mathbf{x}) + b$

分析 SVM 和 SVR 的目标函数，其目标函数实际上也具有相同的形式：

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n} g(f(\mathbf{x}_i), y_i) \tag{33}$$

若 $g(z) = 0, z = y_i f(\mathbf{x}_i)$，则上式为经典 SVM 目标函数；

若 $g(z) = \max(0, 1-z), z = y_i f(\mathbf{x}_i)$，则上式为软间隔 SVM；

若 $g(z) = \ln(1+e^{-z}), z = y_i f(\mathbf{x}_i)$，则上式为对数几率回归；

若 $g(z)$ 为 $\epsilon-$ 不敏感函数，$z = y_i - f(\mathbf{x}_i)$，则上式为 SVR 目标函数；

若 $g(z) = z^2$，$z = y_i - f(\mathbf{x}_i)$，则上式为正则线性回归模型；

分析目标函数，

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n} g(f(\mathbf{x}_i), y_i) \tag{34}$$

其包含两项，后面这一项是用来表述训练集上的误差，前面一项用来描述划分超平面的“间隔”大小或者模型的复杂性，把上式写成更一般的形式：

$$\min_{f} \Omega(f) + C\sum_{i=1}^{n} g(f(\mathbf{x}_i), y_i) \tag{35}$$

其中第一项是模型 f 的函数，用于描述模型 f 的某些性质，也称为**结构风险 (structural risk)**；后面一项是用来衡量模型与训练数据的契合程度，也叫做“**经验风险”(empirical risk)**。

从经验风险最小化的角度来看，$\Omega(f)$ 表述了我们希望获得具有何种性质的模型 (例如希望获得复杂度较小的模型)，这为引入领域知识和用户意图提供了途径; 另一方面，该信息有助于削减假设空间，从而降低了最小化训练误差的过拟合风险. 从这个角度来说，上式也称为“正则化”(regularization) 问题，$\Omega(f)$ 称为正则化项，C 则称为正则化常数。

再回到 KSVM 以及 KSVR 算法上来，其解都具有 $f(\mathbf{x}) = \sum_{i=1}^{n} \beta_i k(\mathbf{x}, \mathbf{x}_i)$ 的形式。

而通过分析其目标函数可以得到更一般的问题 $\min_f \Omega(f) + C\sum_{i=1}^{n} g(f(\mathbf{x}_i), y_i)$。实际上，根据**表示定理**，该问题的解总可以写成 $f(\mathbf{x}) = \sum_{i=1}^{n} \beta_i k(\mathbf{x}, \mathbf{x}_i)$ 的形式，只要：

1. Ω 为关于 f 的某个范数的在 $[0, +\infty)$ 上的单调递增函数；
2. $g(z)$ 为任意非负的损失函数。

9. 核线性判别分析，Kernel LDA

原 LDA 目标函数为：

$$\max_{\mathbf{w}} J(\mathbf{w}) = \frac{\mathbf{w}^t \mathbf{S}_b \mathbf{w}}{\mathbf{w}^t \mathbf{S}_w \mathbf{w}} \tag{36}$$

现在考虑核方法。假设存在映射 ϕ 将原数据样本 $\mathbf{x}$ 映射到更高维的空间得到 $\phi(\mathbf{x})$，在该空间中构造 LDA 目标函数。则有：

$$\max_{\mathbf{w}} J(\mathbf{w}) = \frac{\mathbf{w}^t \mathbf{S}_b^{\phi} \mathbf{w}}{\mathbf{w}^t \mathbf{S}_w^{\phi} \mathbf{w}} \tag{37}$$

这里 $\mathbf{S}_b^{\phi}, \mathbf{S}_w^{\phi}$ 为高维空间中的类间及类内离散度矩阵。

9. 核线性判别分析，Kernel LDA

考虑二分类问题，假设数据集 $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$ 有两类样本，分别记为 C_1, C_2，样本数量分别为 n_1, n_2。则对于第 i 类，其均值以及两个离散度矩阵分别为：

$$\begin{aligned} \mu_i^\phi &= \frac{1}{n_i} \sum_{\mathbf{x} \in C_i} \phi(\mathbf{x}) \\ \mathbf{S}_b^\phi &= (\mu_1^\phi - \mu_0^\phi)(\mu_1^\phi - \mu_0^\phi)^t \\ \mathbf{S}_w^\phi &= \sum_{i=0}^{1} \sum_{\mathbf{x} \in C_i} (\phi(\mathbf{x}) - \mu_i^\phi)(\phi(\mathbf{x}) - \mu_i^\phi)^t \end{aligned} \tag{38}$$

然后思考，如果将 $J(\mathbf{w})$ 作为损失函数，令 $\Omega \equiv 0$，那么可知上式问题的解，即高维核空间中进行投影得到的向量 $h(\mathbf{x}) = \sum_{i=1}^n \beta_i k(\mathbf{x}, \mathbf{x}_i) = \sum_{i=1}^n \beta_i \phi(\mathbf{x})^t \phi(\mathbf{x}_i) = \mathbf{w}^t \phi(\mathbf{x})$。因此，$\mathbf{w} = \sum_{i=1}^n \beta_i \phi(\mathbf{x}_i)$。

最后将 $\mathbf{w}=\sum_{i=1}^{n}\beta_i\phi(\mathbf{x}_i)$，代入公式（37），可以得到

$$\max_{\beta} J(\beta)=\frac{\beta^t\mathbf{M}_b^{\phi}\beta}{\beta^t\mathbf{M}_w^{\phi}\beta} \tag{39}$$

其中 $\mathbf{M}_b=(\tilde{\mu}_0-\tilde{\mu}_1)(\tilde{\mu}_0-\tilde{\mu}_1)^t, \mathbf{M}_w=\mathbf{K}\mathbf{K}^t-\sum_{i=0}^{1} n_i\tilde{\mu}_i\tilde{\mu}_i^t$，而 $\tilde{\mu}_0=\frac{1}{n_0}\mathbf{K}\mathbf{1}_0, \tilde{\mu}_1=\frac{1}{n_0}\mathbf{K}\mathbf{1}_1$，$\mathbf{K}$ 为核矩阵，$\mathbf{1}_i$ 为 n 维向量，当且仅当 $\mathbf{x}_j\in C_i$，则 $\mathbf{1}_i$ 的第 j 个分量为 0。

最后按照经典 LDA 的方法，可以求出 β，由此也可以求得 $h(\mathbf{x})$。

公式（10）推导：

$$\mathcal{L}(\mathrm{w},\mathrm{b},\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{m}\alpha_i\left[y^{(i)}\left(w^Tx^{(i)}+b\right)-1\right]$$

$$=\frac{1}{2}\mathrm{w}^T\mathrm{w}-\sum_{i=1}^{m}\alpha_i y^{(i)}w^Tx^{(i)}-\sum_{i=1}^{m}\alpha_i y^{(i)}b+\sum_{i=1}^{m}\alpha_i$$

$$=\frac{1}{2}\mathrm{w}^T\sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)}-\sum_{i=1}^{m}\alpha_i y^{(i)}w^Tx^{(i)}-\sum_{i=1}^{m}\alpha_i y^{(i)}b+\sum_{i=1}^{m}\alpha_i$$

$$=\frac{1}{2}\mathrm{w}^T\sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)}-w^T\sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)}-\sum_{i=1}^{m}\alpha_i y^{(i)}b+\sum_{i=1}^{m}\alpha_i$$

$$=-\frac{1}{2}\mathrm{w}^T\sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)}-\sum_{i=1}^{m}\alpha_i y^{(i)}b+\sum_{i=1}^{m}\alpha_i$$

$$=-\frac{1}{2}\mathrm{w}^T\sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)}-b\sum_{i=1}^{m}\alpha_i y^{(i)}+\sum_{i=1}^{m}\alpha_i$$

$$=-\frac{1}{2}\left(\sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)}\right)^T\sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)}-b\sum_{i=1}^{m}\alpha_i y^{(i)}+\sum_{i=1}^{m}\alpha_i$$

Thanks !