模式识别　之　支持向量机 (SVM)

Author：　Copyright@ 刘泓尊　at THU

Time:　2019.04

Coding：utf-8 based on LaTeX

一、基本概念

为了让您在阅读过程中不至于看得云里雾里，笔者特意将 SVM 相关概念罗列至此，或许这并不全面，如果笔者有表述不清的地方，还请您移步百度百科！

1. **分类**（Classification）(或者叫做分类器)，而支持向量机本身便是一种监督式学习的方法，它广泛的应用于统计分类以及回归分析中。

2. **监督学习**（Supervised Learning）：监督学习是指：利用一组已知类别的样本调整分类器的参数，使其达到所要求性能的过程，也称为监督训练或有教师学习。

3. **凸优化**（Convex Optimization）：对于函数 f(x)，如果 $f''(x) > 0$，我们称这个函数是“凸”的，也就是清华大学小黄书里介绍的“下凸函数”。对于一个凸函数，它的局部极值就是全局最值，也就是说，其局部最优解就是全局最优解。寻找极值的过程，叫做凸优化。（关于非凸优化问题，请参考本小组另一篇科普文《多元函数极值算法浅论》）

4. **支持向量机**（Support Vector Machines）是一种二分类模型，它的目的是寻找一个超平面来对样本进行分割，分割的原则是间隔最大化，最终转化为一个凸二次规划问题来求解。其由简至繁的模型包括：

a. 当训练样本线性可分时，通过硬间隔最大化，学习一个线性可分支持向量机；

b. 当训练样本近似线性可分时，通过软间隔最大化，学习一个线性支持向量机；

c. 当训练样本线性不可分时，通过核技巧和软间隔最大化，学习一个非线性支持向量机

5. **线性可分**（Linear Separability）：如果一个线性函数能够将样本分开，称这些数据样本是线性可分的。那么什么是线性函数呢？其实很简单，在二维空间中就是一条直线，在三维空间中就是一个平面，以此类推，空间维数为 n，那么这样的线性函数就为 n-1 维超平面。

6. **线性不可分**：不满足线性可分条件的样本集。

7. **支持向量**（Support vector）：样本中处在分类器边界的样本点，叫做“支持向量”，可以理解为这些样本点“支撑”起了这个分类边界。

二、历史

SVM 是由模式识别中广义肖像算法 (Generalized Portrait Algorithm) 发展而来的分类器，其早期工作来自前苏联学者 Vladimir N. Vapnik 和 Alexander Y. Lerner 在 1963 年发表的研究。

1964 年，Vapnik 和 Alexey Y. Chervonenkis 对广义肖像算法进行了进一步讨论并建立了硬边距的线性 SVM。此后在二十世纪 70-80 年代，随着模式识别中最大边距决策边界的理论研究、基于松弛变量的规划问题求解技术的出现，和 VC 维（Vapnik–Chervonenkis,VC dimension）的提出，SVM 被逐步理论化并成为统计学习理论的重要部分。

1992 年，Bernhard E. Boser、Isabelle M.Guyon 和 Vapnik 通过核方法首次得到了非线性 SVM。1995 年，Corinna Cortes 和 Vapnik 提出了软边距的非线性 SVM 并将其应用于手写数字识别问题，这份研究在发表后得到了广泛的关注和引用，为其后 SVM 在各领域的应用奠定了基础。

二十世纪 90 年代，SVM 得到快速发展并衍生出一系列改进和扩展算法，包括多分类 SVM、最小二乘 SVM（Least-Square SVM, LS-SVM）、支持向量回归（Support Vector Regression, SVR）、支持向量聚类（Support Vector Clustering）、半监督 SVM（Semi-supervised SVM, S3VM）等，在人脸识别（Face Recognition）、文本分类（Text Categorization）等模式识别（Pattern Recognition）问题中有广泛应用。

三、**预备知识：**

1．Lagrange Multipliers

给定一个目标函数 $f : R_n \to R$，我们希望找到 $x \in R_n$ ，在满足约束条件 $g(x) = 0$ 的前提下，$s.t.f(x)$ 有最小值。这个约束优化问题如下:

$$\min f(x) \quad s.t.g(x) = 0$$

为方便分析，假设 f 对 g 是连续可导函数, 定义 *Lagrangian* 函数 $L(x, \lambda) = f(x) + g(x)$ 其中 λ 为 Lagrange 乘数。Lagrange 乘数法将原来的约束优化问题转化成等价的非约束问题 $\min L(x, \lambda)$ 优化必要条件：

$$\frac{\partial L}{\partial x} = \nabla f + \lambda \nabla g(x) = 0$$

$$\frac{\partial L}{\partial \lambda} = g(x) = 0$$

其中第一个为 stationary equation，第二个为约束条件。

通过求解上述方程，可得 $L(x, \lambda)$ 的驻点（stationary point）xx^* 以及 λ 的值（正负数皆可能）。

2. Karush Kuhn Tucker(KKT)

接下来我们将约束等式 $g(x) = 0$ 推广为 $g(x) \leq 0$。优化问题如下：

$$\min f(x) \quad s.t.g(x) \leq 0$$

约束不等式 $g(x) \leq 0$ 称为 primal feasibility，由此定义可行域（feasible region）$K = R \in R_n : g(x) \leq 0$ 假设 x^* 为满足约束条件的最佳解，分两种情况讨论：

(1)$g(x) \leq 0$, 最佳解位于 K 的内部，称为 interior solution，这时约束条件是无效的；

(2)$g(x) = 0$, 最佳解落在 K 的边界，称为 boundary solution，此时约束条件是有效的。这两种情况的最佳解具有不同的必要条件。

内部解：在约束条件无效的情形下，g(x) 不起作用，约束优化问题退化为无约束优化问题，因此驻点 x^* 满足 $\nabla f = 0$ 且 $\lambda = 0$

边界解：在约束条件有效的情形下，约束不等式变成等式 $g(x) = 0$，这与前面 Lagrange 乘数法的情况相同。我们可以证明驻点 x^* 发生在 $\nabla f \in span \nabla g$，换句话说，存在 使得 $\nabla f = -\lambda g(x)$，但这里 λ 的正负号是尤其意义的。因为我们希望最小化 f，梯度 ∇f 应该指向可行域 K 的内部，但 ∇g 指向可行域 K 的外部（即 $g(x) > 0$ 的区域），因此 $\lambda \geq 0$ 称为对偶可行性。

不论是内部解还是边界解，$\lambda g(x) = 0$ 恒成立，称为互补松弛（complementary slackness）。

综上，最佳解的必要条件包括 Lagrange 函数 $L(x, \lambda)$ 的定常方程式、原始可行性、对偶可行性和互补松弛：

$$\begin{cases} L_x^{'} = \nabla f(x) + \lambda \nabla g(x) \\ g(x) \leq 0 \\ \lambda \geq 0 \\ \lambda g(x) = 0 \end{cases} \tag{1}$$

以上就是 KKT 条件。如果我们要做大化 f(x) 且受限于 $g(x) \geq 0$，那么对偶可行性要改成 $\lambda \leq 0$ 考虑标准约束优化问题

$$\min f(x) \quad s.t. g(x) = 0, j = 1, \cdots, m$$

$$h_k(x)\ 0, k = 1, \cdots, p$$

定义拉格朗日函数

$$L(x, \{\lambda_j\}, \{\mu_k\}) = f(x) + \ m_j = \lambda_j g_j(x) + \ p_k = \mu_k h_k(x)$$

其中 λ_j 是对应 $g_j(x) = 0$ 的拉格朗日乘数，μ_k 是对应 $h_k(x) \leq 0$ 的拉格朗日乘数，KKT 条件为：

$$\begin{cases} L_x^{'} = 0 \\ g_j(x) = 0, j = 1, \cdots, m \\ h_k(x) \leq 0 \\ \mu_k \geq 0 \\ \mu_k h_k(x) = 0, k = 1, \cdots, p \end{cases} \tag{2}$$

四、SVM 分类

1. 线性可分 SVM

给定训练样本集 $D = (x1, y1), (x2, y2), \cdots, (xm, ym)$ 其中 $yi \in -1, +1$, 分类学习最基本的想法就是基于训练集 D 在样本空间中找到一个划分超平面，将不同类别的样本分开。

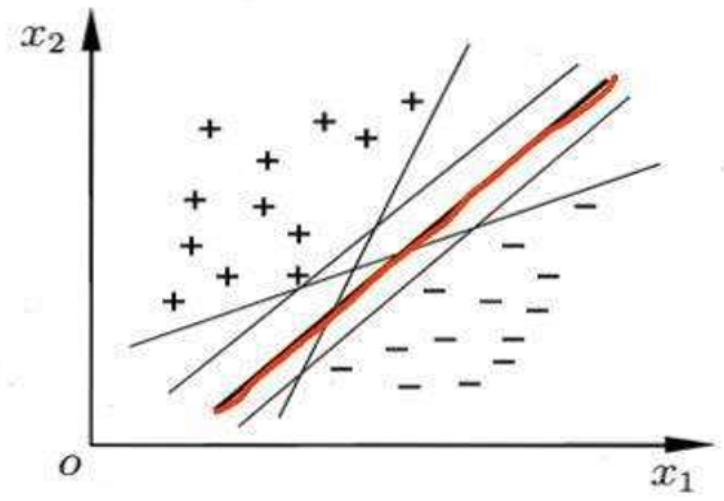

图 1: 存在多个划分超平面将样本分开

如上图，显然样本是线性可分的，但是很显然不只有这一条直线可以将样本分开，而是有无数条，我们所说的线性可分支持向量机就对应着能将数据正确划分并且间隔最大的直线。即下图：

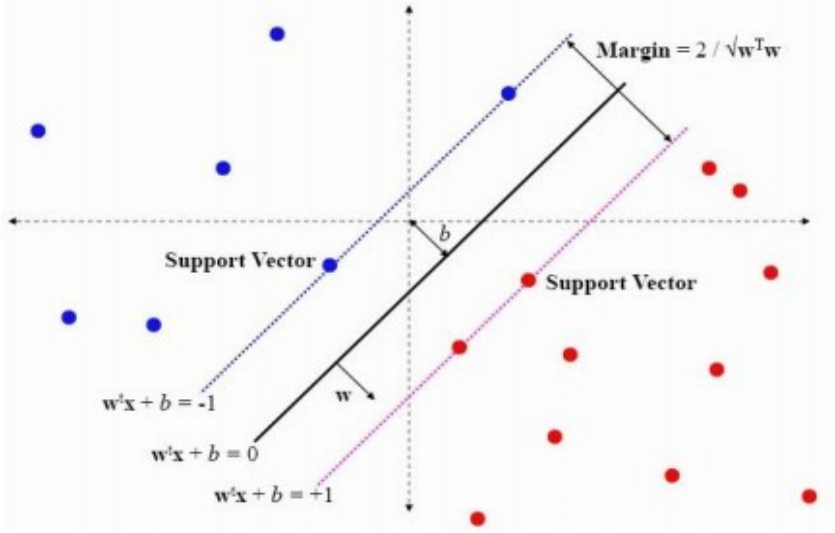

图 2: SVM 的超平面示例

任意一个 n 维空间的 n-1 维超平面都可以写成：$w^T + b = 0$ 的形式。其中 $\vec{w}$ 为平面的法向量

下面我们开始计算间隔，其实间隔就等于两个异类支持向量的差在 $\vec{w}$ 上的投影，即：

$$\gamma = \frac{(\overrightarrow{x_+} - \overrightarrow{x_-}) \cdot w^T}{\|w\|} \tag{3}$$

其中 $\overrightarrow{x_+}$ 和 $\overrightarrow{x_-}$ 分别表示两个正负支持向量.

因为 $\overrightarrow{x_+}$ 和 $\overrightarrow{x_-}$ 满足 $y_i(w^T x_i + b) = 1$ 即在超平面 $w^T x + b = 0$ 确定的情况下，$\left|w^T x + b\right|$ 能够相对的表示点 x 到距离超平面的远近，而 $(w^T x + b)$ 的符号与类标记 y 的符号是否一致表示分类是否正确，所以，可以用量 $y(w^T x + b)$ 的正负性来判定或表示分类的正确性和确信度。展开来写就是:

$$\begin{cases} +1 * (w^T x + b) = 1, y_i = +1 \\ -1 * (w^T x + b) = 1, y_i = -1 \end{cases} \tag{4}$$

很显然，由于这些 supporting vector 刚好在边界上，所以它们满足 $y(w^T x + b) = 1$ 对于所有不是支持向量的点，也就是在“阵地后方”的点，则显然有 $y(w^T x + b) > 1$ 至于为什么选取 1 和-1 作为分类界限，大概起源于 logistic 回归，但是笔者并不想多花时间写这个回归模型，有兴趣的同学自己去找博客吧哈哈哈。但是可以肯定的是，这里取任何互为相反数的常数 c 都可以，因为在空间中总可以通过一种仿射变换使得 c “归一化”，实际上也就是给样本数据乘一个缩放系数。推出：

$$\begin{cases} w^T x_+ = 1 - b \\ w^T x_- = -1 - b \end{cases} \tag{5}$$

代入公式 (3) 中可以得到

$$\gamma = \frac{(1 - b) + (1 + b)}{\|w\|} = \frac{2}{\|w\|} \tag{6}$$

对一个数据点进行分类，当它的间隔值越大的时候，分类的可信度越大。对于一个包含 n 个点的数据集，我们可以很自然地定义它的间隔值为所有这 n 个点的间隔值中最小的那个。于是，为了使得分类的可信度高，我们希望所选择的超平面能够最大化这个间隔值。至此，我们求得了间隔，SVM 的思想是使得间隔最大化，也就是：

$$\max_{w,b} \frac{2}{\|w\|}, s.t. y_i(w^T x_i + b) \geq 1 \quad (i = 1, 2, \cdots, m) \tag{7}$$

显然，最大化 $\frac{2}{\|w\|}$ 于最小化 $\|w\|$，为了计算方便，将公式（6）转化成如下：

$$\min_{w,b} \frac{1}{2}\|w\|^2, s.t. y_i(w^T x_i + b) \geq 1 \quad (i = 1, 2, \cdots, m) \tag{8}$$

公式 (8) 即为支持向量机的基本型。

公式 (8) 本身是一个凸二次规划问题. 我们采用基于拉格朗日乘子法的对偶算法进行计算。对公式 (8) 使用拉格朗日乘子法得到其对偶问题，该问题的拉格朗日函数可以写为：

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 + \sum_{i=1}^{m} \alpha_i (1 - y_i(w^T x_i + b)) \tag{9}$$

公式 (9) 分别对 w 和 b 求偏导：

$$\begin{cases} \frac{\partial L}{\partial w} = w - \sum\limits_{i=1}^{m} \alpha_i y_i x_i \\ \frac{\partial L}{\partial b} = \sum\limits_{i=1}^{m} \alpha_i y_i \end{cases} \tag{10}$$

令其分别为 0，可以得到：

$$\begin{cases} w = \sum\limits_{i=1}^{m} \alpha_i y_i x_i \\ \sum\limits_{i=1}^{m} \alpha_i y_i = 0 \end{cases} \tag{11}$$

将公式 (10)(11) 代入公式 (9)，可得：

$$\begin{aligned} L(w, b, \alpha) &= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i x_j \\ s.t. \sum_{i=1}^{m} \alpha_i y_i &= 0, \alpha_i \geq 0, i = 1, 2, \cdots, m \end{aligned} \tag{12}$$

此时，原问题就转化为以下仅关于 α 的问题：

$$\begin{aligned} &\max_{\alpha} \left(\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i x_j \right) \\ &s.t. \sum_{i=1}^{m} \alpha_i y_i = 0, \alpha_i \geq 0, i = 1, 2, \cdots, m \end{aligned} \tag{13}$$

解出 α 之后，根据公式 (11) 可以求得 w ，进而求得 b，可以得到模型：

上述过程的 KKT 条件为：

$$\begin{cases} \alpha_i \geq 0 \\ y_i f(x_i) - 1 \geq 0 \\ \alpha_i (y_i f(x_i) - 1) = 0 \end{cases} \tag{14}$$

我们分析一下，对于任意的训练样本 (xi, yi)，若 $\alpha_i = 0$ 则其不会在公式（13）中的求和项中出现，也就是说，它不影响模型的训练；若 $\alpha_i > 0$ 则 $y_i f(x_i) - 1 = 0$，也就是 $y_i f(x_i) = 1$，即该样本一定在边界上，是一个支持向量。这里显示出了支持向量机的重要特征：当训练完成后，大部分样本都不需要保留，最终模型只与支持向量有关。至此，我们分析完了线性可分样本的原理

2. 线性不可分 SVM

对于非线性问题，线性可分支持向量机并不能有效解决，要使用非线性模型才能很好地分类。先看一个例子，如下图，很显然使用直线并不能将两类样本分开，但是可以使用一条椭圆曲线（非线性模型）将它们分开。非线性问题往往不好求解，所以希望能用解线性分类问题的方法求解，因此可以采用非线性变换，将非线性问题变换成线性问题。

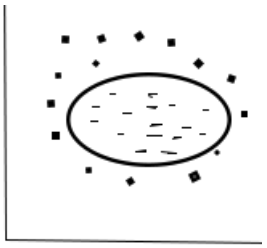

图 3: 线性不可分样本示例

对于非线性的情况，SVM 的处理方法是选择一个核函数，通过将数据映射到高维空间，来解决在原始空间中线性不可分的问题。由于核函数的优良品质，这样的非线性扩展在计算量上并没有比原来复杂多少，这一点是非常难得的。当然，这要归功于核方法——除了 SVM 之外，任何将计算表示为数据点的内积的方法，都可以使用核方法进行非线性扩展。令 ϕ(x) 表示将 x 映射后的特征向量，于是在特征空间中，划分超平面所对应的的模型可表示为：

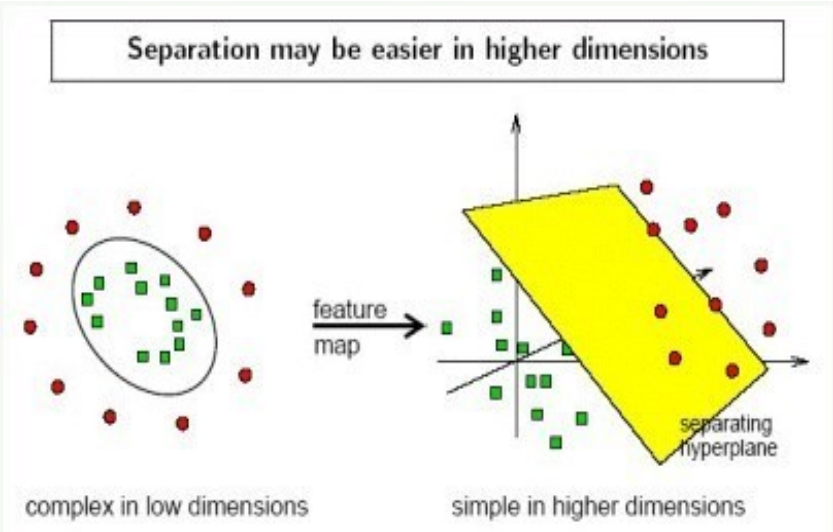

图 4: 利用核方法将数据映射到高维空间

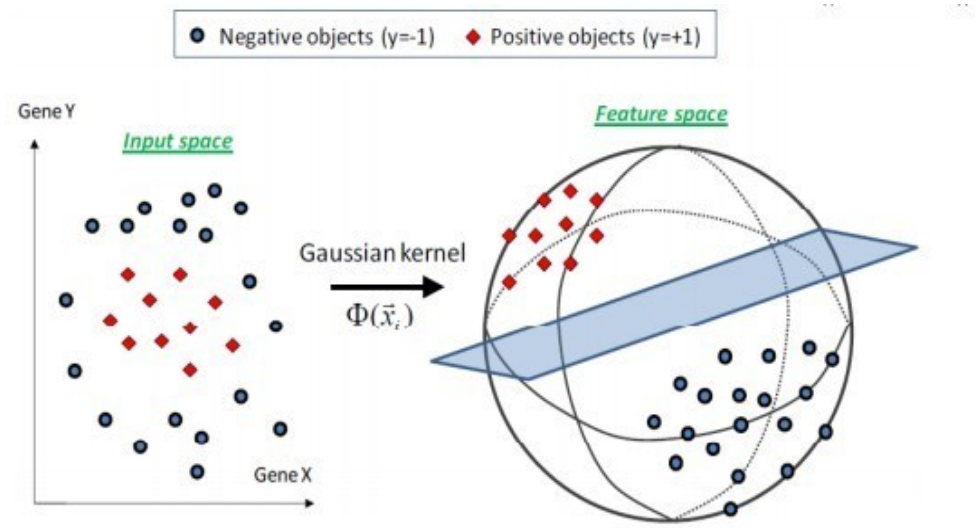

图 5: 使用 SIGMOD 函数的映射示例

非线性可分问题的算法可以用下图简要说明：

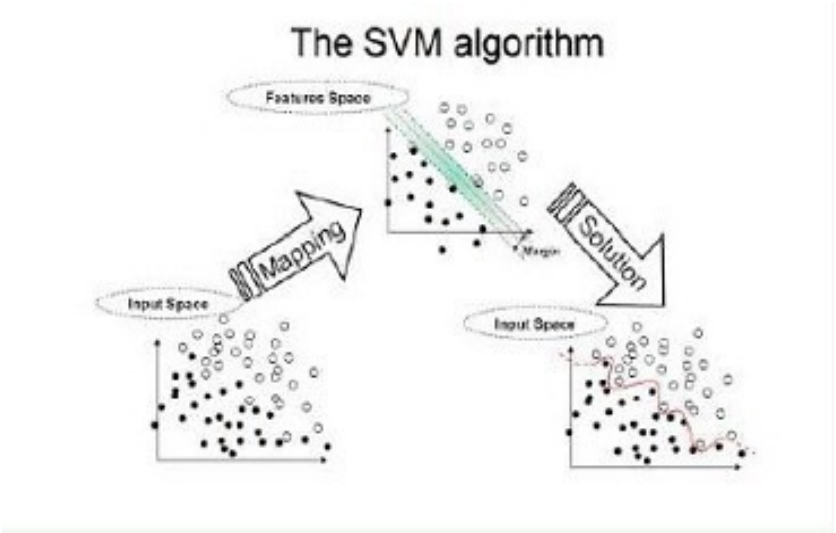

图 6: SVM 核方法示例

而下文我们将具体介绍的核函数则提供了此种问题的解决途径，从下文你将看到，核函数通过把数据映射到高维空间来增加第一节所述的线性学习器的能力，使得线性学习器对偶空间的表达方式让分类操作更具灵活性和可操作性。因为训练样例一般是不会独立出现的，它们总是以成对样例的内积形式出现，而用对偶形式表示学习器的优势在为在该表示中可调参数的个数不依赖输入属性的个数，通过使用恰当的核函数来替代内积，可以隐式得将非线性的训练数据映射到高维空间，而不增加可调参数的个数 (当然，前提是核函数能够计算对应着两个输入特征向量的内积)

有最小化函数 (注意这里 $\phi(x)$ 代替了之前的 x):

$$\min_{w,b} \frac{1}{2}\|w\|^2, s.t. y_i(w^T\phi(x_i)+b) \geq 1 \quad (i=1,2,\cdots,m) \tag{15}$$

其对偶问题为：

$$\begin{aligned} \max_{\alpha} &\left(\sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \phi(x_i)\phi(x_j)\right) \\ s.t. &\sum_{i=1}^{m}\alpha_i y_i = 0, \alpha_i \geq 0, i=1,2,\cdots,m \end{aligned} \tag{16}$$

若要对公式 (16) 求解，会涉及到计算 $\phi(xi)^T\phi(xj)$，这是样本 x_i 和 x_j 映射到特征空间之后的内积，由于特征空间的维数可能很高，甚至是无穷维，因此直接计算 $\phi(xi)^T\phi(xj)$ 通常是困难的，于是想到这样一个函数：

$$\kappa(x_i, x_j) = \phi(x_i)^T\phi(x_j) \tag{17}$$

举个简单直接点的例子，如果不是用核技术，就会先计算线性映射 $\phi(x_1)$ 和 $\phi(x_2)$, 然后计算这两个特征的内积，使用了核技术之后，先把 $\phi(x_1)$ 和 $\phi(x_2)$ 的通用表达式子：$\langle\phi(x_1)\ \phi(x_2)\rangle = \kappa(\langle x_1, x_2\rangle)$ 计算出来，注意到这里的 $\langle\ \ ,\ \ \rangle$ 表示内积，$\kappa(\ \ ,\ \ )$ 就是对应的核函数，这个表达往往非常简单，所以计算非常方便。

于是公式 (16) 写成如下：

$$\begin{aligned} \max_{\alpha} &\left(\sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \kappa(x_i, x_j)\right) \\ s.t. &\sum_{i=1}^{m}\alpha_i y_i = 0, \alpha_i \geq 0, i=1,2,\cdots,m \end{aligned} \tag{18}$$

求解后得到：

$$\begin{aligned} f(x) &= w^T\phi(x) + b \\ &= \sum_{i=1}^{m}\alpha_i y_i \phi(x_i)^T\phi(x_j) + b \\ &= \sum_{i=1}^{m}\alpha_i y_i \kappa(x_i, x_j) + b \end{aligned} \tag{19}$$

这里的函数 $\kappa(x_i, x_j)$ 就是核函数，在实际应用中，通常人们会从一些常用的核函数里选择（根据样本数据的不同，选择不同的参数，实际上就得到了不同的核函数），下面给出常用的核函数：

1. 线性核：

$$\kappa(x_i, x_j) = x_i^T x_j$$

2. 多项式核（d 是多项式的次数，d=1 退化为线性核）：

$$\kappa(x_i, x_j) = (x_i^T x_j)^d$$

3. 高斯核（σ>0）：

$$\kappa(x_i, x_j) = exp(-\frac{(\|x_i - x_j\|)^2}{2\sigma^2})$$

4. 拉普拉斯核（σ>0）：

$$\kappa(x_i, x_j) = exp(-\frac{\|x_i - x_j\|}{\sigma})$$

5.sigmiod 核（β>0,θ>0）：

$$\kappa(x_i, x_j) = tanh(\beta x_i^T x_j + \theta)$$

此外，核函数也可以通过组合得到，在此不再赘述。

核函数的本质上面说了这么一大堆，读者可能还是没明白核函数到底是个什么东西？我再简要概括下，即以下三点：

1. 实际中，我们会经常遇到线性不可分的样例，此时，我们的常用做法是把样例特征映射到高维空间中去，映射到高维空间后，相关特征便被分开了，也就达到了分类的目的。

2. 但进一步，如果凡是遇到线性不可分的样例，一律映射到高维空间，那么这个维度大小是会高到可怕的（甚至是无穷维）。

3. 此时，核函数就隆重登场了，核函数的价值在于，它虽然也是将特征进行从低维到高维的转换，但核函数优势在于：它事先在低维上进行计算，而将实质上的分类效果表现在了高维上，也就如上文所说的避免了直接在高维空间中的复杂计算。

还想再通俗一点吗??? 好!!! 那么我们看看下面的例子。

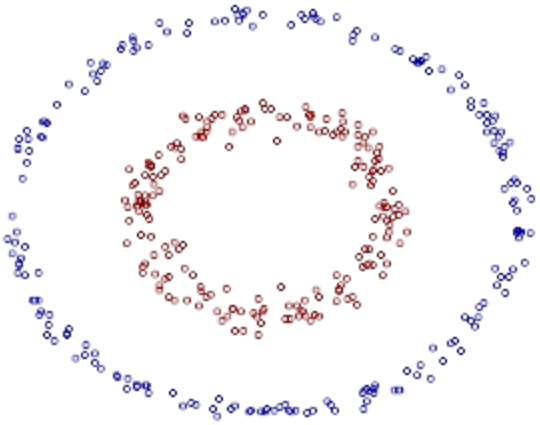

图 7: 一个线性不可分的例子

如果用 x_1 和 x_2 来表示这个二维平面的两个坐标的话，我们知道一条二次曲线（圆圈是二次曲线的一种特殊情况）的方程可以写作这样的形式：

$$a_1x_1 + a_2x_1^2 + a_3x_2 + a_4x_2^2 + a_5x_1x_2 + a_6 = 0 \tag{20}$$

注意上面的形式,如果我们构造另外一个五维的空间,其中五个坐标的值分别为 $(x_1, x_1^2, x_2, x_2^2, x_1x_2) = (z_1, z_2, z_3, z_4, z_5)$ ，那么显然，上面的方程在新的坐标系下可以写作：

$$\sum_{i=1}^{5} a_i z_i + a_6 = 0 \tag{21}$$

关于新的坐标 $(z_1, z_2, z_3, z_4, z_5)$，这正是一个超平面的方程！也就是说，如果我们做一个映射 f，将 $(x_1, x_1^2, x_2, x_2^2, x_1x_2)$ 按照上面的规则映射为 $(z_1, z_2, z_3, z_4, z_5)$，那么在新的空间中原来的数据将变成线性可分的，从而使用之前我们推导的线性分类算法就可以进行处理了。这正是 Kernel 方法处理非线性问题的基本思想。

但是!!!

其实刚才的方法稍想一下就会发现有问题：在最初的例子里，我们对一个二维空间做映射，选择的新空间是原始空间的所有一阶和二阶的组合，得到了五个维度；如果原始空间是三维，那么我们会得到 19 维的新空间，这个数目是呈爆炸性增长的，这给的计算带来了非常大的困难，而且如果遇到无穷维的情况，就根本无从计算了。所以就需要 Kernel 出马了

设两个向量 $x_1 = (\eta_1, \eta_2)^T$ 和 $x_2 = (\xi_1, \xi_2)^T$，而 $\phi(\cdot)$ 即是到前面说的五维空间的映射，因此映射过后的内积为：

$$\langle \phi(x_1), \phi(x_2) \rangle = \eta_1\xi_1 + \eta_1^2\xi_1^2 + \eta_2\xi_2 + \eta_2^2\xi_2^2 + \eta_1\eta_2\xi_1\xi_2 \tag{22}$$

另外，我们又注意到：

$$(\langle x_1, x_2 \rangle + 1)^2 = 2\eta_1\xi_1 + \eta_1^2\xi_1^2 + 2\eta_2\xi_2 + \eta_2^2\xi_2^2 + 2\eta_1\eta_2\xi_1\xi_2 + 1 \tag{23}$$

二者有很多相似的地方，实际上，我们只要把某几个维度线性缩放一下，然后再加上一个常数维度，具体来说，上面这个式子的计算结果实际上和映射

$$\phi(x_1, x_2) = (\sqrt{2}x_1, x_1^2, \sqrt{2}x_2, x_2^2, \sqrt{2}x_1x_2, 1)^T \tag{24}$$

之后的内积 $\langle (\phi(x_1), \phi(x_2)) \rangle$ 的结果是相等的，那么区别在于什么地方呢？

1. 一个是映射到高维空间中，然后再根据内积的公式进行计算；2. 而另一个则直接在原来的低维空间中进行计算，而不需要显式地写出映射后的结果。

刚才提到的映射的维度爆炸，在前一种方法已经无法计算的情况下，后一种方法却依旧能从容处理，甚至是无穷维度的情况也没有问题。

计算两个向量在隐式映射过后的空间中的内积的函数叫做核函数 (Kernel Function). 即, 核函数能简化映射空间中的内积运算——刚好“碰巧”的是，在我们的 SVM 里需要计算的地方数据向量总是以内积的形式出现的。

至此，关于 SVM 的两类问题：线性可分支持向量机与硬间隔最大化，非线性支持向量机与核函数，介绍完毕。

五、应用特点

对于 SVM 而言，它并没有对原始数据的分布做任何的假设，这表明 SVM 模型对数据分布的要求低，那么其适用性自然就会更广一些。如果我们事先对数据的分布没有任何的先验信息，即，不知道是什么分布，那么 SVM 无疑是比较好的选择。