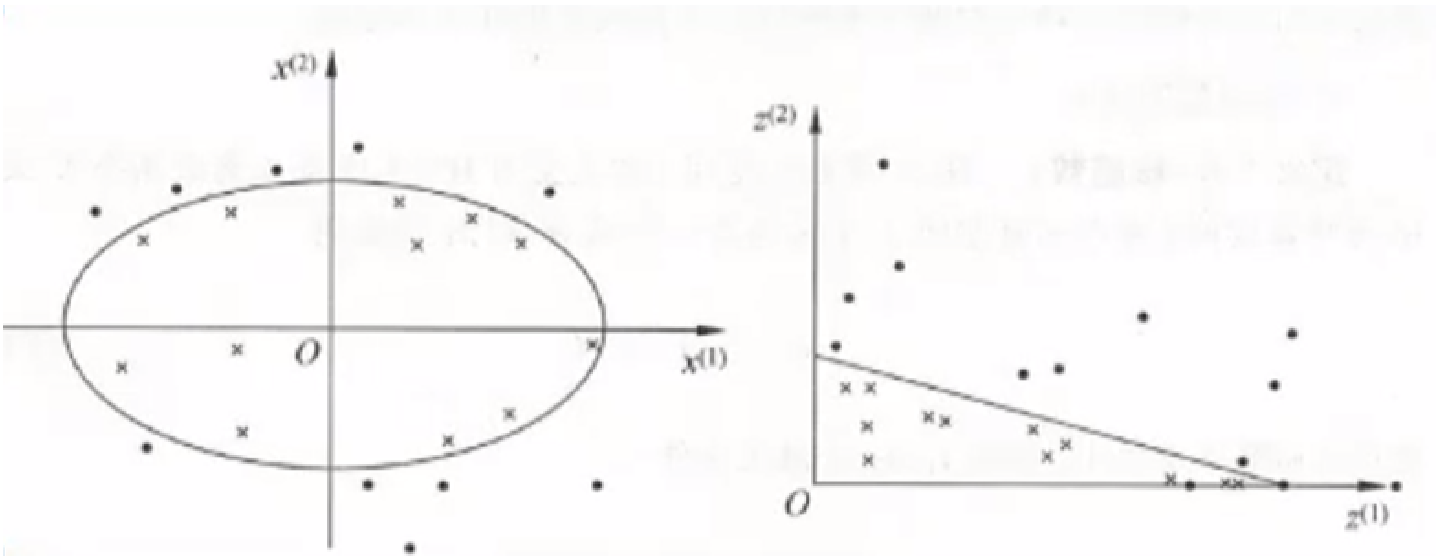
核函数

前言

首先，我们需要简单的做一下引入，什么是核函数。在之前的的局部线性回归推导当中提到过，我们甚至让AI帮我们讲解高斯核推导，但是我们并没有深入了解什么是核函数......简单来说，我们在处理低维模型的时候，往往是非线性的，这时候，我们需要通过函数映射，将低维问题映射到高维，从非线性到线性！（***就是你发现你的模型只能处理线性问题，于是引入核，从而让你的线性模型拥有非线性的处理能力***）

不好理解？没关系，看图（下面这个是例子，看懂最好，看不懂没事，这不是 SVM）：



我们首先看这个椭圆，上过高中的都知道，椭圆的公式

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1 \quad (a>b>0)$$

对于非线性问题，如果直接在二维 (x_1, x_2) 空间用线性分类器（如感知机、线性 SVM），显然无法得到直线分割。

但如果我们构造一个新的特征空间：

$$z_1 = x_1^2, \quad z_2 = x_2^2$$

那么椭圆方程变成：

$$\frac{z_1}{a^2}+\frac{z_2}{b^2}=1$$

而在 (z_1, z_2) 空间中是一个**线性方程**（超平面）。也就是说，通过 **非线性映射**

$$\phi : (x_1, x_2) \mapsto (x_1^2, x_2^2)$$

我们把原本在二维空间非线性可分的问题，变成了在新空间的线性可分问题。

而在 SVM 中，如果直接把每个样本都显式映射到高维空间去计算内积，代价很大（尤其是高维甚至无限维映射），核函数就很好地解决了这个问题。

其关键思想是（K）：**我们不需要显式地去做映射 $\phi(x)$，只需要能直接计算 $\phi(x_i)\cdot\phi(x_j)$ 的结果。**例如，上面映射的内积为：

$$\phi(x)=(x_1^2,x_2^2)\phi(x_i)\cdot\phi(x_j)=x_{i1}^2x_{j1}^2+x_{i2}^2x_{j2}^2$$

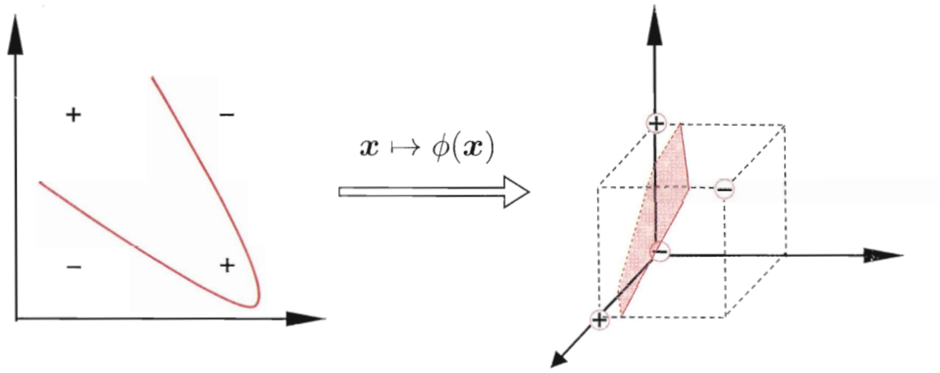
如果换成更一般的二次多项式映射（通式），核函数可以写成：

$$K(x_i,x_j)=(\mathbf{x}_i\cdot\mathbf{x}_j+c)^2$$

这就是**多项式核（Polynomial Kernel）**，它等价于把数据映射到所有二次项的空间，但计算量却只需要原空间的内积运算。

核函数下的 SVM

我们现在回到 SVM，在我们前边的讨论中，我们一直假设训练样本是线性可分的，也就是一定存在一个超平面将训练样本完全正确分类。然而在现实当中这几乎不可能，例如下图这个就不是线性可分的（来自西瓜书）：



对于这样的问题，我们在前言也说过，可以使用核函数隐式地定义了一个从原始特征空间到一个更高维（甚至是无限维）的再生核希尔伯特空间（RKHS）的非线性映射 ϕ 。通过这个映射，样本在 RKHS 中的分布更可能被一个线性超平面所分离（再生希尔伯特空间会在后边进行讲解）。很显然，一般的原始样本空间是有限维度的，即属性有限，那么一定存在一个高维度特征空间使得样本可分。

令 $\phi(\mathbf{x})$ 表示将 $\mathbf{x}$ 映射后的特征向量，于是，映射后的超平面模型表示为：

$$f(\mathbf{x})=\mathbf{w}^\top\phi(\mathbf{x})+b, \quad (6.19)$$

其中的 $\mathbf{w}$ 和 b 不变，依旧是模型参数。于是类似有(6.6)，我们有：

$$\begin{aligned}&\min_{\mathbf{w},\mathbf{b}} && \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{s.t.} && y_i(\mathbf{w}^\top\phi(x_i)+b)\geq 1, \quad i=1,2,...,m.\end{aligned}, \quad (6.20)$$

再对偶化，得到与(6.11)类似的式子：

$$
\begin{aligned}
\max_{\alpha} \quad & \sum_{i=1}^{m} - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \phi(x_i)^\top \phi(x_j) \\
\text{s.t.} \quad & \sum_{i=1}^{m} \alpha_i y_i = 0, \\
& \alpha_i \geq 0, \quad i = 1, 2, ..., m \ .
\end{aligned} \quad , \quad (6.21)
$$

求解(6.21)涉及到计算 $\phi(x_i)^\top \phi(x_j)$ ，这是 x_i 与 x_j 映射到高维特征空间的内积。因为直接计算可能会导致维度过高，甚至无穷维，因此直接计算属于没事找事。我们可以假设这样一个函数：

$$
\kappa(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = \phi(x_i)^\top \phi(x_j), \quad (6.22)
$$

即 x_i 与 x_j 在特征空间的内积等于它们在原始空间中通过函数 $\kappa(\cdot, \cdot)$ （这个 κ 叫做 卡帕kappa）计算的结果。有了这样的函数，我们就可以直接在原始空间的基础上进行计算，不必再去考虑映射到高维空间再去考虑内积这么麻烦（亦称核技巧），于是我们将(6.21)进行重写，得到：

$$
\begin{aligned}
\max_{\alpha} \quad & \sum_{i=1}^{m} - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \kappa(x_i, x_j) \\
\text{s.t.} \quad & \sum_{i=1}^{m} \alpha_i y_i = 0, \\
& \alpha_i \geq 0, \quad i = 1, 2, ..., m \ .
\end{aligned} \quad , \quad (6.23)
$$

由超平面模型得到：

$$
\begin{aligned}
f(\mathbf{x}) &= \mathbf{w}^\top \phi(\mathbf{x}) + b \\
&= \sum_{i=1}^{m} \alpha_i y_i \phi(x_i)^\top \phi(x) + b \\
&= \sum_{i=1}^{m} \alpha_i y_i \kappa(x, x_i) + b
\end{aligned} \quad , \quad (6.24)
$$

这里的函数 $\kappa(\cdot, \cdot)$ 就是“核函数”(kernel function)。式(6.24)显示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)。

这也是我们在 对偶问题 的 SMO 算法中使用的核函数。还记得么？我们在“核函数代换”那部分让各位直接“令” $x_i^T \cdot x_j = \kappa_{ij}$ ，其实这个就是核函数。

其余求解超平面方程和 SMO 算法一致！

关于核函数的一些解释

这部分同样来自西瓜书，也是我之前没关注的地方、解决合适的核函数、什么是再生希尔伯特空间...

显然，我们若已经知道了映射 $\phi(\cdot)$ 的具体形式，则可以很好地写出kappa核函数。但事实是我们通常不知道 $\phi(\cdot)$ ，怎么怎么确定 $\phi(\cdot)$ 函数就成了一个问题，或者说合适的核函数是否一定存在呢？什么样的函数能做核函数呢？

关于这部分可以参考这篇2002年的综述：Learning with kernels: support vector machines, regularization, optimization, and beyond

参考上述综述，有定理：令 χ 为输入空间， $\kappa(\cdot,\cdot)$ 是定义在 $\chi \times \chi$ 上的对称函数，则 κ 是核函数当且仅当对于任意数据 $D=\{x_1,x_2,...,x_m\}$ ，“核矩阵”(kernel matrix) $\mathbf{K}$ 总是半正定的：

$$\mathbf{K}=\begin{bmatrix}\kappa(x_1,x_1) & \cdots & \kappa(x_1,x_j) & \cdots & \kappa(x_1,x_m)\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ \kappa(x_i,x_1) & \cdots & \kappa(x_i,x_j) & \cdots & \kappa(x_i,x_m)\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ \kappa(x_m,x_1) & \cdots & \kappa(x_m,x_j) & \cdots & \kappa(x_m,x_m)\end{bmatrix}.$$

上述定理表示了，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。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 ϕ 。换言之，任何一个 kappa 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, 简称 RKHS)的特征空间。

关于大卫·希尔伯特：这位大佬可谓是一统了数学、物理学和计算机科学。他的学生外尔和冯·诺依曼分别奠定了广义相对论的数学基础和计算机科学与量子力学基础、他的发小闵可夫斯基奠定了狭义相对论的数学基础...可以说人脉这一块了./

结论

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定义了这个特征空间。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，很可能导致性能不佳。

所以我们列出了几个常用的核函数：

1. 线性核： $\kappa(x_i,x_j)=x_i^\top x_j$
2. 多项式核： $\kappa(x_i,x_j)=(x_i^\top,x_j)^d$ ， $d\geq 1$ 为多项式次数
3. 高斯核： $\kappa(x_i,x_j)=exp(-\frac{\|x_i-x_j\|^2}{2\sigma^2})$ ， $\sigma>0$ 为高斯核的带宽
4. 拉普拉斯核： $\kappa(x_i,x_j)=exp(-\frac{\|x_i-x_j\|}{\sigma})$ ， $\sigma>0$
5. Sigmoid 核： $\kappa(x_i,x_j)=tanh(\beta x_i^\top x_j+\theta)$ ， $tanh$ 为双曲正切函数， $\beta>0,\theta<0$

除此以外，我们还可以将函数组合起来，如：

- 核函数 κ_1 和 κ_2 ，则任意正数 γ_1 和 γ_2 有线性组合：

$$\gamma_1\kappa_1+\gamma_2\kappa_2, \quad (6.25)$$

- 核函数 κ_1 和 κ_2 有直积：

$$\kappa_1\otimes\kappa_2(\mathbf{x},\mathbf{z})=\kappa_1(\mathbf{x},\mathbf{z})\kappa_2(\mathbf{x},\mathbf{z}), \quad (6.26)$$

- 核函数 κ_1 ，则有任意函数 $g(\mathbf{x})$ ：

$$\kappa(\mathbf{x},\mathbf{z})=g(\mathbf{x})\kappa_1(\mathbf{x},\mathbf{z})g(\mathbf{z}), \quad (6.27)$$

这三个同样是核函数（~~再生希尔伯特空间的再生希尔伯特空间（bushi~~