Support Vector Machine

AsukaShiKi

2019 年 1 月 31 日

目录

1 支持向量机

支持向量机（Support Vector Machine，简称 SVM）是一种针对二分类任务设计的分类器，它的理论相对神经网络模型来说更加完备和严密，并且效果显著，结果可预测，是非常值得学习的模型。

本章内容包括有：

间隔与支持向量：如何计算空间中任一点到超平面的距离？什么是间隔？什么是支持向量？支持向量机求解的目标是什么？

对偶问题：求取最大间隔等价于怎样的对偶问题？KKT 条件揭示出支持向量机的的什么性质？如何用 SMO 算法快速求解？为什么 SMO 算法可以快速求解？

核函数：如何处理非线性问题？什么是核函数？为什么需要核函数？有哪些常用的核函数？核函数具有什么性质？

软间隔和正则化：如何应对过拟合问题？软间隔和硬间隔分别指什么？如何求解软间隔支持向量机？0/1 损失函数具有哪些可选的损失替代函数？

支持向量回归：什么是支持向量回归？与传统回归模型有什么不同？支持向量回归的支持向量满足什么条件？

核方法：什么是核方法？什么是表示定理？如何应用？

2 间隔与支持向量

2.1 点到超平面的距离

设$\boldsymbol{\omega}^{\boldsymbol{T}}$x+b=0 是超平面的表达式，则超平面的法向量是$\boldsymbol{\omega}$，证明如下：任取超平面上两点 x_1,x_2, 则两点均满足超平面方程，且两点连线 x_1x_2 所在直线也在超平面上，两点连线的向量形式则为两点向量差 x_1-x_2，则有 $\boldsymbol{\omega}^{\boldsymbol{T}}(x_1\text{-}x_2)=\boldsymbol{\omega}^{\boldsymbol{T}}x_1+b-(\boldsymbol{\omega}^{\boldsymbol{T}}x_2+b)=0-0=0$; 所以得证$\boldsymbol{\omega}$ 垂直于 $\overrightarrow{x_1x_2}$，即$\boldsymbol{\omega}$ 是超平面$\boldsymbol{\omega}^{\boldsymbol{T}}$x+b=0 的法向量。

设有超平面$\boldsymbol{\omega}^{\boldsymbol{T}}$x+b=0，空间中一点 x, 超平面上一点 y, 则点 x 到超平面的距离即为向量 $\overrightarrow{xy}$ 在超平面法向量$\boldsymbol{\omega}$ 方向上的投影长度。则点 x 到超平面距离:

$$\text{dist}^1 = \frac{|\boldsymbol{\omega}^{\boldsymbol{T}}\overrightarrow{xy}|}{||\boldsymbol{\omega}||} = \frac{|\boldsymbol{\omega}^{\boldsymbol{T}}(x-y)|}{||\boldsymbol{\omega}||} = \frac{|\boldsymbol{\omega}^{\boldsymbol{T}}x-\boldsymbol{\omega}^{\boldsymbol{T}}y|}{||\boldsymbol{\omega}||} = \frac{|\boldsymbol{\omega}^{\boldsymbol{T}}x-(0-b)|}{||\boldsymbol{\omega}||}$$

即有：

$$\text{dist} = \frac{|\boldsymbol{\omega}^{\boldsymbol{T}}x+b|}{||\boldsymbol{\omega}||}$$

由上述推导可知，超平面外任一点 x 到超平面$\boldsymbol{\omega}^{\boldsymbol{T}}$x+b=0 的距离都是 $\frac{|\boldsymbol{\omega}^{\boldsymbol{T}}x+b|}{||\boldsymbol{\omega}||}$。

2.2 支持向量

给定一个二分类数据集，正类标记标记为 +1，负类标记为-1；分类学习试图从样本空间找到一个超平面，使得超平面可以将不同类的样本分开。满足要求的超平面可能会有很多，我们需要找到最优的超平面。

在 svm 中，我们试图找到处于两类样本正中间的超平面，这样的超平面对于训练数据局部扰动的容忍性最好，新样本也不容易被误分类；也就是这样的超平面对未见示例的泛化能力最强（如图 1）。

图一中实线即为划分超平面，在线性模型中可以通过方程$\boldsymbol{\omega}^{\boldsymbol{T}}$x+b=0 来表示，在二维样本空间中就是一条直线。$\boldsymbol{\omega}$ 是线性模型的权重向量（又叫投影向量），也是划分超平面的法向量，决定着超平面的方向。偏置项 b 又被称为位移项，决定了超平面和空间原点之间的距离。

假设超平面可以将所有样本正确分类，即对于样本点 (x,y) 有标记 y=+1 的点有$\boldsymbol{\omega}^{\boldsymbol{T}}$x+b >0，对所有标记 y=-1 的点有 $\boldsymbol{\omega}^{\boldsymbol{T}}$x+b<0。

只要存在这样一个超平面，那么我们就可以对参数$\boldsymbol{\omega}$ 和 b 进行适当的放缩，使得有：

$$\boldsymbol{\omega}^{\boldsymbol{T}}\text{x} + \text{b} \geq +1,\text{y} = +1$$

$$\boldsymbol{\omega}^{\boldsymbol{T}}\text{x} + \text{b} \leq \text{-}1;\text{y} = \text{-}1$$

[1] $||\boldsymbol{\omega}||$ 是 ω 的欧几里得范数

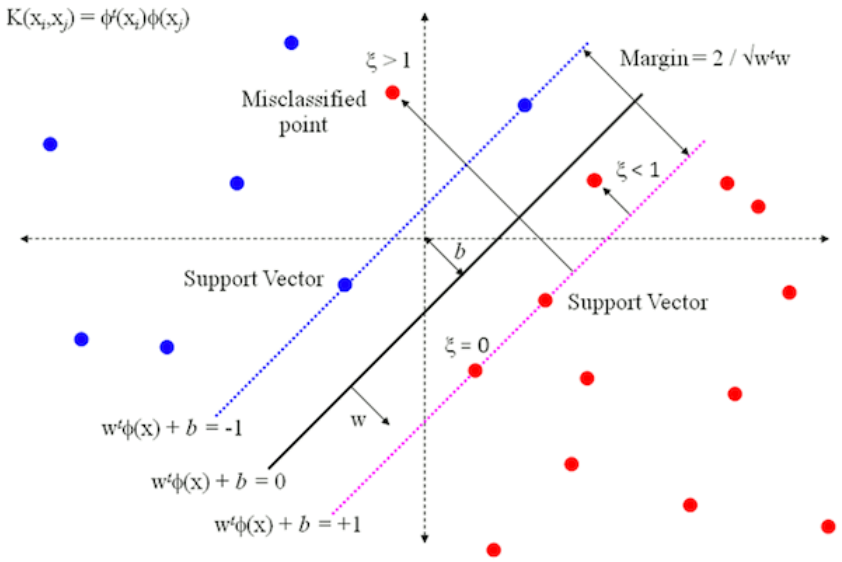


图 1: 对样本空间分类的超平面示例图

而 SVM 中使得上式等号成立的训练样本点就是支持向量 (Support Vector), 它们是距离超平面最近的几个样本点，也即上面图中两条虚线上的点（图中存在比支持向量距离超平面更近的点，这跟软间隔有关，这里先不讨论）。在 SVM 中，我们希望实现的是最大化两类支持向量到超平面的距离之和，那首先就得知道怎么计算距离。怎样计算样本空间中任意数据点到划分超平面的距离我们在上一节中已经讲过。

2.3 间隔

如 2.2 节所述，我们想要实现的是最大化两类支持向量到超平面的距离之和，如图一所示，根据定义我们有：

$$\boldsymbol{\omega}^{\boldsymbol{T}}\mathrm{x} + \mathrm{b} = +1;\mathrm{y} = +1$$

$$\boldsymbol{\omega}^{\boldsymbol{T}}\mathrm{x} + \mathrm{b} = \text{-}1;\mathrm{y} = \text{-}1$$

带入 2.1 节的距离公式可得支持向量到超平面的距离是 $\frac{1}{||\boldsymbol{\omega}||}$。

我们定义**间隔 (margin)** 为两个异类支持向量到超平面的距离之和为：

$$\boldsymbol{\gamma} = 2 * \frac{1}{||\boldsymbol{\omega}||} = \frac{2}{||\boldsymbol{\omega}||}$$

SVM 的目标就是找到具有最大间隔的划分超平面，也即找到是$\boldsymbol{\gamma}$ 最大的参数$\boldsymbol{\omega}$ 和 b 进行适当的放缩，使得有：

$$\max_{\omega,b} \frac{2}{||\boldsymbol{\omega}||}$$

$$s.t. \quad y_i(\omega^T x + b) \geq 1, i = 1, 2, 3, ...m$$

约束部分指的是全部样本都被正确分类，此时标记值（+1 或 −1）乘上预测值（ +1 或 −1）必定是一个 1 的数值。看上去间隔大小只与 ω 有关，但实际上位移项 b 也通过约束影响着 ω 的取值，进而对间隔产生影响。由于最大化 $||\boldsymbol{\omega}||^{-1}$ 等价于最小化 $||\boldsymbol{\omega}||^2$，所以可以重写目标函数为：

$$\min_{\omega,b} \frac{||\boldsymbol{\omega}||^2}{2}$$

$$s.t. \quad y_i(\omega^T x + b) \geq 1, i = 1, 2, 3, ...m(1)$$

我们引入 $\frac{1}{2}$ 的原因是为了求导时可以约去平方项的 2，式 (1) 就是支持向量机的基本型。特别地，我们还有以下定义：

函数间隔：$y_i(\omega^T x + b)$

几何间隔：$\frac{y_i(\omega^T x+b)}{||\boldsymbol{\omega}||^2}$

3 对偶问题

3.1 拉格朗日乘子

式 (1) 是一个带约束的凸二次规划问题 (凸问题意味一定有全局最优解，而不会陷入局部最优解)。本节介绍使用拉格朗日乘子计算最优解的方法。

拉格朗日乘子方法的过程大致可以分为以下几步, 对于待求目标:$min_x f(x), s.t. h(x) = 0, g(x) \leq 0$:

(a). 拉格朗日函数 $L(x, \lambda, \mu) = f(x) + \lambda h(x) + \mu g(x)$

(b). 令 $\frac{\partial L(x,\lambda,\mu)}{\partial x} = 0$, 解出用 λ,μ 表示出的 x

(c). 将 x 带入 $L(x, \lambda, \mu)$ 中得：$\Gamma(\lambda, \mu) = min_x L(x, \lambda, \mu)$

(d). 对偶问题即为：$max_{\lambda,\mu} \Gamma(\lambda, \mu), s.t. \mu \geq 0$

3.2 应用拉格朗日乘子到式 (1)

回到式 (1)

$$\min_{\omega,b} \frac{||\omega||^2}{2}$$

$$s.t. \quad y_i(\omega^T x + b) \geq 1, i = 1, 2, 3, ...m (1)$$

为式 (1) 的每个约束条件添加拉格朗日乘子 $a_i \geq 0$(对应 m 个样本的 m 个约束)，得到该问题的拉格朗日函数：

$$L(\omega, b, a) = \frac{1}{2}||\omega||^2 + \sum_{i=0}^{m} a_i(1 - y_i(\omega^T x_i + b)) \quad (2)$$

其中 $a = (a_1, a_2, a_3, ..., a_m)$，对拉格朗日函数求关于 ω, b 的偏导数，并令偏导为零。

$$\text{偏导数：} \quad \frac{\partial L(\omega,b,a)}{\partial \omega} = \omega - \sum_{i=1}^{m} a_i y_i x_i (3)$$

$$\frac{\partial L(\omega,b,a)}{\partial b} = -\sum_{i=1}^{m} a_i y_i \quad (4)$$

令式 (3)(4) 为 0，得 $\omega = \sum_{i=1}^{m} a_i y_i x_i, \sum_{i=1}^{m} a_i y_i = 0$。

将式 (3) 带入式 (1) 消去 ω, 再由约束条件式 (4) 消去 b，得到式 (1) 的对偶问题 (附推导过程)：

$$\begin{aligned} L(\omega, b, a) &= \frac{1}{2}\omega^T\omega + \sum_{i=0}^{m} a_i - \sum_{i=0}^{m} a_i y_i \omega^T x_i - \sum_{i=0}^{m} a_i y_i b \\ &= \frac{1}{2}\omega^T\omega + \sum_{i=0}^{m} a_i - \omega^T(\sum_{i=0}^{m} a_i y_i x_i) - b(\sum_{i=0}^{m} a_i y_i) \end{aligned}$$

$$L(\omega,b,a)=\frac{1}{2}\omega^T\omega+\sum_{i=0}^{m}a_i-\omega^T\omega-0$$
$$=\sum_{i=0}^{m}a_i-\frac{1}{2}\omega\omega^T$$

将 $\omega=\sum_{i=1}^{m}a_iy_ix_i$ 带入上式可得式 (1) 的对偶问题：

$$\max_a\sum_{i=1}^{m}a_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}a_ia_jy_iy_jx_i^Tx_j$$

$$s.t.\sum_{i=1}^{m}a_iy_i=0,a_i\geq 0,i=1,2,3,....,m \quad (5)$$

解出 max 值 a_i 后，可以求出 ω,b 得到模型：

$$f(x)=\omega^Tx+b=\sum_{i=1}^{m}a_iy_ix_i^Tx_i+b \quad (6)$$

我们在式 (5) 解出的 max 值 a_i 是式 (2) 的拉格朗日乘子，对应训练样本 (x_i,y_i), 由于式 (1) 有不等式约束，所以求解过程需要满足 KKT 条件：

$$\begin{cases}a_i\geq 0\\ y_if(x_i)-1\geq 0\\ a_i(y_if(x_i)-1)=0\end{cases}$$

这个 KKT 条件说明了，对任何一个样本 x_i 来说，要么对应的拉格朗日乘子 a_i 为 0，此时样本 x_i 对式 (6) 毫无贡献，不会影响到模型；要么函数间隔 $y_if(x_i)=1$，此时样本 x_i 位于最大间隔边界上，是一个支持向量。它揭示了 SVM 的一个重要性质：最终模型只与支持向量有关，因此训练完成后，大部分的训练样本都不需保留。

3.3 SMO 算法

对偶问题式 (5) 是二次规划问题，为避免通用二次规划算法求解是问题规模正比于样本数的问题，我们使用高效的 SMO 算法。

初始化参数 a 后，SMO 算法重复下面两个步骤直至收敛：

1. 选取一对需要更新的变量 a_i 和 a_j。
2. 固定 a_i 和 a_j 之外的参数，求解对偶问题式 (5) 来更新 a_i 和 a_j。

3.3.1 选取 a_i 和 a_j

首先我们要知道，只要我们选取的 a_i,a_j 有一个不满足 KKT 条件，那么更新后的目标函数的值就会变大，而且违背 KKT 条件的程度越大，更新后的目标函数值增幅越大。

由此，SMO 算法先选取一个违背 KKT 条件程度最大的变量 a_i, 再选取一个使目标函数增长最快的变量 a_j，但由于找出 a_j 的开销较大，所以 SMO 算法采用了一个启发式算法，使选取的两变量对应的样本之间间隔最大，这样的两个变量差别很大，与选取两个相似的变量相比，这种方法能为目标函数带来更大的变化，从而更快的搜索到全局最大值。

由于每次迭代中，SMO 算法只选取两个参数，其他参数固定，所以效率较高。由此，可以将式 (5) 的约束重写为：

$$a_i y_i + a_j y_j = c, a_i \geq 0, a_j \geq 0 \quad (7)$$

其中，$c = -\sum_{k \neq i,j} a_k y_k$ 看作是固定的常数。

3.3.2 计算 ω 和 b

利用式 (7), 可以将 a_j 从式 (5) 中消去，考虑 $a_i \geq 0$ 约束，可以直接算出 a_i, a_j。使用 SMO 算法计算出最优解之后，我们关注的是如何推出 ω 和 b ，从而得到最终模型。获得 ω 很简单，直接用式 (3)=0 就可以了。而位移项 b 则可以通过支持向量导出，因为对于任一支持向量 (x_s, y_s)，都有函数间隔等于 1，所以有：

$$y_s f(x) = y_s(\sum_{i \in S} a_i y_i x_i^T x_s + b) = 1 \quad (8)$$

这里的 S 是所有支持向量的下标集 (事实上，用所有样本的下标也行，不过非支持向量的拉格朗日乘子等于 0，对求和没贡献，这一点前面已经提到了)。理论上，我们只要选取任意一个支持向量代入式 (8) 就可以把 b 算出来了。但实际任务中往往采用一种更鲁棒的做法：用所有支持向量求解的平均值：

$$b = \frac{1}{|S|} \sum_{s \in S} (\frac{1}{y_s} - \sum_{i \in S} a_i y_i x_i^T x_s)$$

4 核函数

4.1 处理非线性划分

在现实任务中，我们更常遇到的是在原始样本空间中非线性可分的问题。对这样的问题，一种常用的思路是将样本从原始空间映射到一个更高维的特征空间，使得样本在该特征空间中线性可分。幸运的是，只要原始空间是有限维的 (也即属性数目有限)，那就必然存在一个高维特征空间使样本线性可分, 举例如下，二维平面上若干样本点呈如下分布：

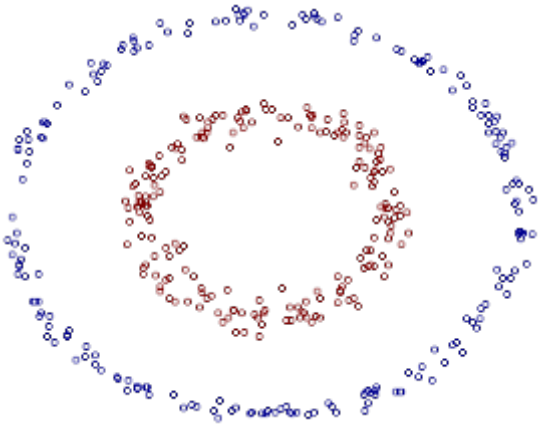


图 2: 二维平面样本点分布图

此时划分两类样本，需要一个非线性的圆形曲线，假设原始空间中两个属性分别是 x, y, 如果我们做一个映射，把样本点都映射到一个三维特征空间，维度取值分别为 x^2, y^2, y, 则得到下面的分布：

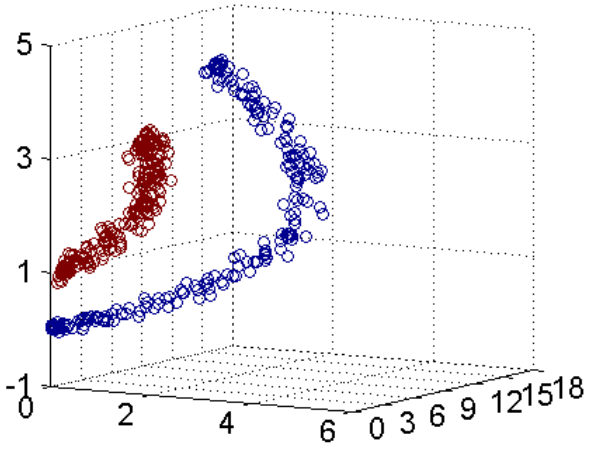


图 3: 三维映射图

映射过后，可以看见我们只需要一个线性超平面就可以将两类样本点分开。

4.2 核函数定义

在上一节例子中，我们将每个样本对应的二维特征向量 x 映射为一个三维的特征向量，假设我们用 $\phi(x)$ 来表示映射所得特征向量。则在映射的高维特征空间中，用于划分的线性超平面可以表示为：

$$f(x) = \omega^T \phi(x) + b$$

类似式 (1)，可以得到此时的目标函数为：

$$\min_{\omega,b} \frac{1}{2}||\omega||^2 \quad (9)$$

$$s.t. \quad y_i(\omega^T \phi(x) + b) \geq 1, i = 1, 2, 3, ..., m$$

对应的对偶问题为:

$$\max_a \sum_{i=1}^{m} a_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} a_i a_j y_i y_j \phi(x_i)^T \phi(x_j) \quad (10)$$

$$s.t. \quad \sum_{i=1}^{m} a_i y_i = 0, a_i \geq 0, i = 1, 2, ..., m$$

式 (10) 中的 $\phi(x_i)^T \phi(x_i = j)$ 是向量 x_i, x_j 映射到高维空间后的内积，由于特征空间维数可能会很高，所以直接计算映射后的特征向量的内积很困难，如果映射后的特征空间是无限维的，则根本无法计算。

为了解决这样的问题，引入了核函数；假设输入空间是二维的，每个样本点有两个属性 x, y，存在映射将每个样本点映射到三维空间：

$$\phi(x) = \phi(x, y) = (x^2, \sqrt{2}xy, y^2)$$

给定原始空间中的两个样本点 $V_1 = (x_1, y_1)$ 和 $V_2 = (x_2, y_2)$, 则他们映射到高维特征空间后的内积可以写作：

$$\begin{aligned}
\phi(V_1)^T \phi(V_2) &=< \phi(V_1), \phi(V_2) > \\
&=< (x_1^2, \sqrt{2}x_1 y_1, y_1^2), (x_2^2, \sqrt{2}x_2 y_2, y_2^2) > \\
&= x_1^2 x_2^2 + 2x_1 x_2 y_1 y_2 + y_1^2 y_2^2 \\
&= (x_1 x_2 + y_1 y_2)^2 \\
&=< V_1, V_2 >^2 \\
&= \kappa(V_1, V_2)
\end{aligned}$$

从上面的例子中可以看出高维特征空间中两个点的内积，可以写成一个关于原始空间中两个点的函数 $\kappa(,;,)$, 这个就是核函数，上面的例子用的就是多项式核，多项式次数 d 取 2。

4.3 为何需要核函数

这里的例子为了计算方便，映射的空间维数依然很低，所以稍微解释一下为什么需要核函数？假设原始空间是二维的，那么对于两个属性 x 和 y，取一阶二阶的组合只有 5 个 (即 x^2, y^2, x, y, xy)。但当原始空间是三维的时候，仍然取一阶二阶，组合就多达 19 个了 $(x, y, z, xy, xz, yz, xyz, x^2y, x^2z, y^2x, y^2z, z^2x, z^2y, x^2yz, xy^2z, xyz^2, x^2y^2z, x^2yz^2, xy^2z^2)$。随着原始空间维数增长，新空间的维数是呈爆炸性上升的。何况现实中我们遇到的问题的原始空间往往本来就已经是高维的，如果再进行映射，新特征空间的维度是难以想象的。

然而有了核函数，我们就可以在原始空间中通过函数 $\kappa(.;.)$ 计算 (这称为核技巧 (kernel trick))，而不必直接计算高维甚至无穷维特征空间中的内积。

使用核函数后，对偶问题式 (10) 可以重写为：

$$\begin{aligned} &\max_{a} \sum_{i=1}^{m} a_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} a_i a_j y_i y_j \kappa(x_i; x_j) \\ &s.t. \sum_{i=1}^{m} a_i y_i = 0, a_i \geq 0, i = 1, 2, 3, ..., m \quad (11) \end{aligned}$$

求解后得到的模型可以表示为：

$$\begin{aligned} f(x) &= \omega^T \phi(x) + b \\ &= \sum_{i=1}^{m} a_i y_i \phi(x_i)^T \phi(x) + b \\ &= \sum_{i=1}^{m} a_i y_i \kappa(x_i; x) + b \end{aligned}$$

这条式子表明了模型最优解可通过训练样本的核函数展开，称为支持向量展式 (support vector expansion)。在需要对新样本进行预测时，我们无需把新样本映射到高维（甚至无限维）空间，而是可以利用保存下来的训练样本 (支持向量) 和核函数 κ 进行求解。

注意，核函数本身不等于映射; 它只是一个与计算两个数据点映射到高维空间之后的内积等价的函数。当我们发现数据在原始空间线性不可分时，会有把数据映射到高维空间来实现线性可分的想法，比方说引入原有属性的幂或者原有属性之间的乘积作为新的维度。假设我们把数据点都映射到了一个维数很高甚至无穷维的特征空间，而模型求解和预测的过程需要用到映射后两个数据点的内积，这时直接计算就没辙了。但我们又幸运地发现，原来高维空间中两点的内积在数值上等于原始空间通过某个核函数算出的函数值，无需先映射再求值，就很好地解决了计算的问题了。

4.3.1 核函数的性质

核函数定理：给定一个输入空间 χ, 函数 $\kappa(.;.)$ 是定义在 $\chi \times \chi$ 上的对称函数。当且仅当对于任意数据集 $D = x_1, x_2, x_3, ..., x_m$, 对应的核矩阵都是半正定矩阵的时候，$\kappa$ 是核函数。核矩阵是一个规模为 $m \times m$ 的函数矩阵，每个元素都是一个函数，比如第 i 行 j 列的元素是 $\kappa(x_i, x_j)$。也即是说，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，简称 RKHS)”的特征空间。做映射的初衷是希望样本在新特征空间上线性可分，新特征空间的好坏直接决定了支持向量机的性能，但是我们并不知道怎样的核函数是合适的。一般来说有以下几种常用核函数:

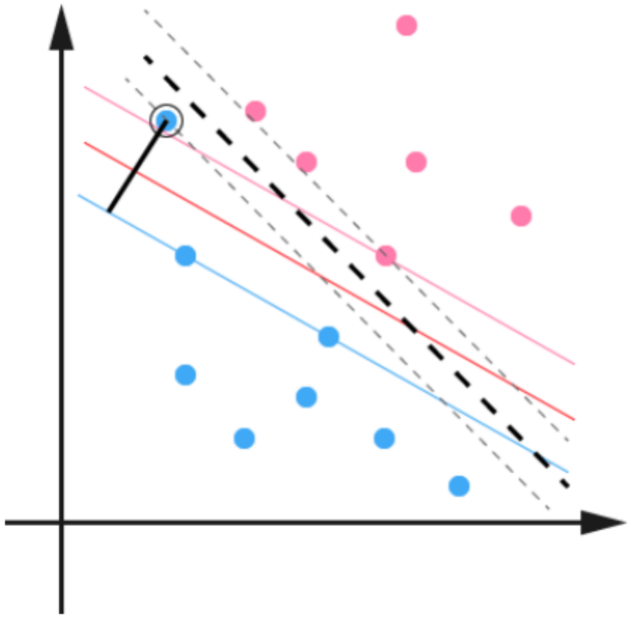
名称	表达式	参数
线性核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$	-
多项式核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j)^d$	$d \geq 1$为多项式的次数，d=1时退化为线性核
高斯核（亦称RBF核）	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})$	$\sigma > 0$ 为高斯核的带宽（width）
拉普拉斯核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\sigma})$	$\sigma > 0$
Sigmoid核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta \mathbf{x}_i^T \mathbf{x}_j + \theta)$	$tanh$ 为双曲正切函数，$\beta > 0, \theta < 0$

特别地，文本数据一般用线性核，情况不明可尝试高斯核。除了这些常用的核函数，要产生核函数还可以使用组合的方式:

1. 若 κ_1 和 κ_2 都是核函数，则 $a\kappa_1 + b\kappa_2$ 也是核函数，其中 $a > 0, b > 0$。
2. 若 κ_1 和 κ_2 都是核函数，则其直积 $\kappa_1 \times \kappa_2(x, z) = \kappa_1(x, z)\kappa_2(x, z)$ 也是核函数。
3. 若 κ_1 都是核函数，则对于任意函数 $g(x)$,$\kappa(x, z) = g(x)\kappa_1(x, z)g(z)$。

5 软间隔与正则化

上一节中，通过利用核函数映射来解决非线性可分的问题，但现实中很难找到合适的核函数，即使某个核函数能令训练集在新特征空间中线性可分，也难保这不是过拟合造成的结果。比方说上面这张图，黑色虚线是此时的划分超平面，最大间隔很小。但事实上，黑色圆



圈圈起的蓝点是一个 outlier，可能是噪声的原因，它偏离了正确的分布。而训练模型时，我们并没有考虑这一点，这就导致把训练样本中的 outlier 当成数据的真实分布拟合了，也即过拟合。

但当我们允许这个 outlier 被误分类时，得到的划分超平面可能就如图中深红色线所示，此时的最大间隔更大，预测新样本时误分类的概率也会降低很多。

在实际任务中，outlier 的情况可能更加严重。比方说，如果图中的 outlier 再往右上移动一些距离的话，我们甚至会无法构造出一个能将数据划分开的超平面。

缓解该问题的一个思路就是允许支持向量机在一些样本上出错，为此，引入软间隔 (soft margin) 的概念。软间隔是相对于硬间隔 (hard margin) 的一个概念，硬间隔要求所有样本都必须划分正确，也即约束：

$$y_i(\omega^T x_i + b) \geq 1$$

软间隔则允许某些样本不满足约束（根据约束条件的不同，有可能某些样本出现在间隔内，甚至被误分类）。此时目标函数可以重写为：

$$\min_{\omega,b} \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{m} l_{0/1}(y_i(\omega^T x + b) - 1) \quad (12)$$

其中 $l_{0/1}$ 是 0/1 损失函数:

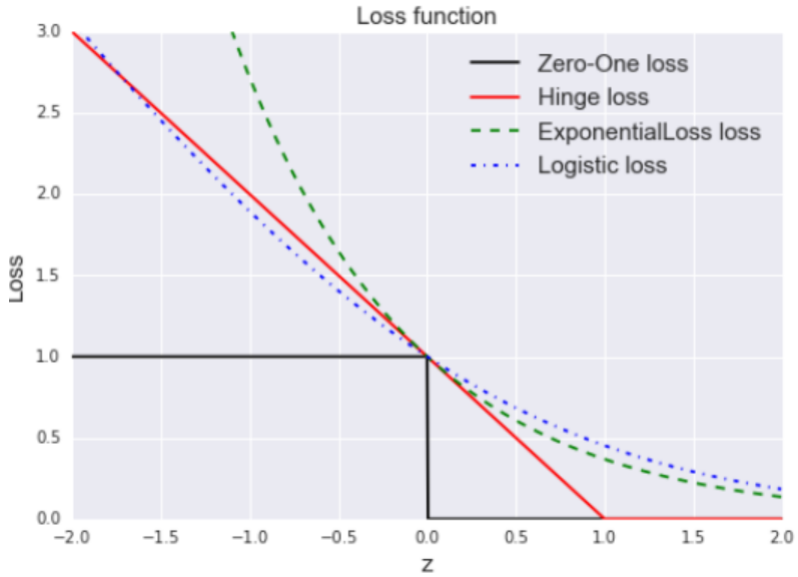
$$l_{0/1}(z) = \begin{cases} 1, if z < 0 \\ 0, otherwise. \end{cases}$$

它的含义很简单：如果分类正确，那么函数间隔必定大于等于 1，此时损失为 0；如果分类错误，那么函数间隔必定小于等于-1，此时损失为 1。而 C 则是一个大于 0 的常数，当 C 趋于无穷大时，式 (12) 等效于带约束的式 (1)，因为此时对误分类的惩罚无限大，也即要求全部样本分类正确。当 C 取有限值时，允许某些样本分类错误。由于 0/1 损失函数是一个非凸不连续函数，所以式 (12) 难以求解，于是在实际任务中，我们采用一些凸的连续函数来取替它，这样的函数就称为替代损失 (surrogate loss) 函数。

最常用的有以下三种：

hinge 损失：$l_{hinge}(z) = max(, 1 - z)$
指数损失：$l_{exp}(z) = exp(-z)$
对率损失：$l_{log}(z) = log(1 + exp(-z))$

不妨作图观察比较一下这些损失函数:



这里有个问题是，书中提到对率损失中 log 指 ln，也即底数为自然对数，但这种情况下对率损失在 $z = 0$ 处不为 1，而是 0.693。但是书中的插图里，对率损失经过 (0,1) 点，此时底数应为 2，上面的插图就是按底数为 2 计算的。实际任务中最常用的是 hinge 损失，这里就以 hinge 损失为例，替代 0/1 损失函数，此时目标函数式 (12) 可以重写为：

$$\min_{\omega,b} \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{m} max(0, 1 - y_i(\omega^T x + b)) \quad (13)$$

引入松弛变量 (slack variables) ，可以把式 (13) 重写为：

$$\min_{\omega,b} \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{T}\xi_i \quad (14)$$

$$s.t. \quad y_i(w^T x + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, 2, 3, ..., m$$

该式描述的就是软间隔支持向量机，其中每个样本都对应着一个松弛变量，用以表示该样本误分类的程度，松弛变量的值越大，程度越高。

5.1 软间隔支持向量机

式 (14) 仍然是一个二次规划问题，我们仍用前面的方法求解:

1. 通过拉格朗日乘子法把 m 个约束转换 m 个拉格朗日乘子，得到该问题的拉格朗日函数。

2. 分别对 $\omega.b,\xi$ 求偏导，代入拉格朗日函数得到对偶问题。

3. 使用 SMO 算法求解对偶问题，解出所有样本对应的拉格朗日乘子。

4. 需要进行新样本预测时，使用支持向量及其对应的拉格朗日乘子进行求解。

故有式 (14) 的拉格朗日函数：

$$L(\omega, b, a, \xi, \mu) = \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{m}\xi_i + \sum_{i=1}^{m} a_i(1 - \xi_i - y_i(\omega^T x_i + b)) - \sum_{i=1}^{m}\mu_i\xi_i \quad (15)$$

其中 $a_i \geq 0, \mu_i \geq 0$ 是拉格朗日乘子。

令 $L(\omega, b, a, \xi, \mu)$ 对 ω, b, ξ_i 的偏导为 0，可得：

$$\omega = \sum_{i=1}^{m} a_i y_i x_i$$

$$0 = \sum_{i=1}^{m} a_i y_i$$

$$C = a_i + \mu_i$$

将上述三式代入式 (15) 可得式 (14) 的对偶问题：

$$\max_a \sum_{i=1}^{m} a_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} a_i a_j y_i y_j x_i^T x_j \quad (16)$$
$$s.t. \quad \sum_{i=1}^{m} a_i y_i = 0, 0 \le a_i \le C, i = 1, 2, 3, ..., m$$

除去约束条件外，我们可以看出，软间隔支持向量机的对偶问题式 (16) 和硬间隔支持向量机的对偶问题式 (5) 几乎完全一致。同样的，由于式 (14) 的约束条件是不等式约束，所以求解过程要求满足 KKT 条件：

$$\begin{cases} a_i \ge 0 \\ \mu_i \ge 0 \\ y_i f(x_i) - 1 + \xi_i \ge 0 \\ a_i(y_i f(x_i) - 1 + \xi_i) = 0 \\ \xi_i \ge 0 \\ \mu_i \xi_i \ge 0 \end{cases}$$

KKT 条件可以理解为：

1. 对任意训练样本，要么对应的拉格朗日乘子 $a_i = 0$；要么函数间隔等于 1 和对应的松弛变量之差 $(y_i(\omega^T x + b) = 1 - \xi_i)$。

2. 如果一个样本的拉格朗日乘子 $a_i = 0$，则它对模型没有任何影响，不需要保留。

3. 如果一个样本的拉格朗日乘子大于 0，则它是支持向量; 若拉格朗日乘子 a_i 又小于 C，按照式 (15) 有 $\mu_i > 0$，因此松弛变量 $\xi_i = 0$，此时函数间隔为 1，样本落在最大间隔边界上。若拉格朗日乘子 a_i 等于 C，按照式 (15) 有 $\mu_i = 0$，因此松弛变量 $\xi_i > 0$。若 $\xi_i < 1$，则样本落在间隔内，但依然被正确分类。若 $\xi_i > 1$，则样本落在另一个类的间隔外，被错误分类。

下图就展示了一个典型的软间隔支持向量机。图中就有一些异常点，这些点有的虽然在虚线与超平面之间 $(0 < y_i(\omega^T + b) < 1)$，但也能被正确分类 (x_3)。有的点落到了超平面的另一侧，就会被误分类 $(x_4\ x_5)$。特别地，在 R.Collobert. 的论文 Large Scale Machine Learning 中提到，常数 C 一般取训练集大小的倒数 $(\frac{1}{m})$。

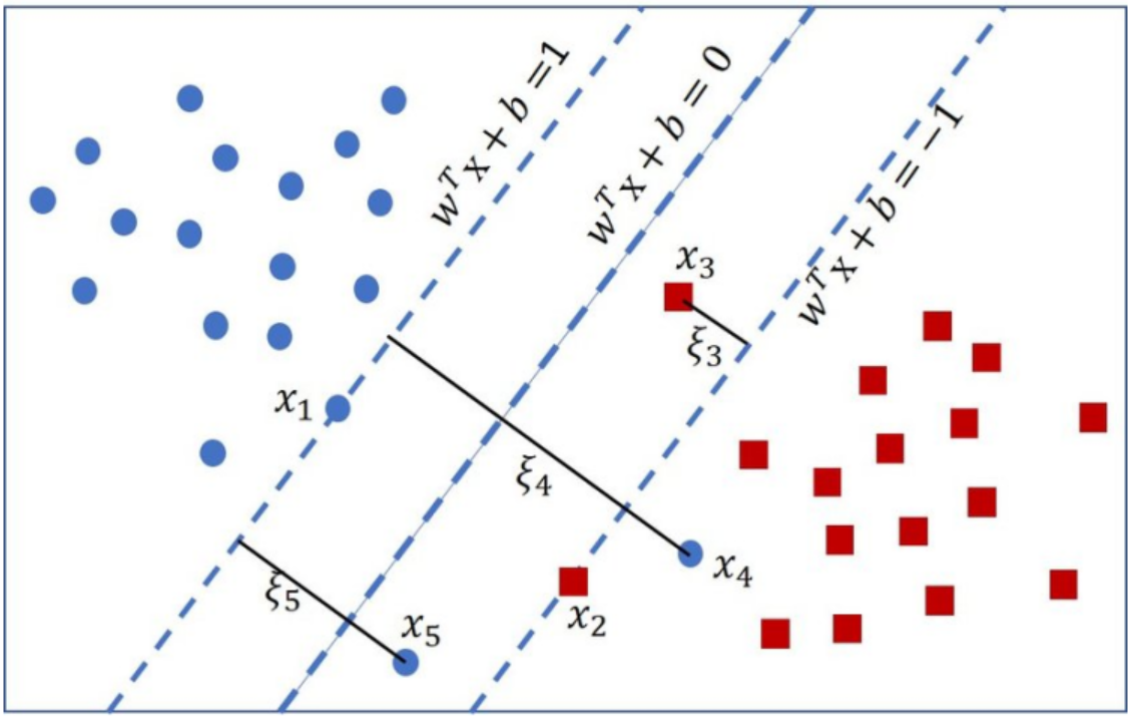


图 4: 软间隔支持向量机

5.2 支持向量机和逻辑回归的联系与区别

上面用的是 hinge 损失，不过我们也提到了还有其他一些替代损失函数，事实上，使用对率损失时，SVM 得到的模型和 LR 是非常类似的。

支持向量机和逻辑回归的相同点：

1. 都是线性分类器，模型求解出一个划分超平面；

2. 两种方法都可以增加不同的正则化项；

3. 通常来说性能相当。

支持向量机和逻辑回归的不同点：

1.LR 使用对率损失，SVM 一般用 hinge 损失；

2. 在 LR 的模型求解过程中，每个训练样本都对划分超平面有影响，影响力随着与超平面的距离增大而减小，所以说 LR 的解受训练数据本身的分布影响；SVM 的模型只与占训练数据少部分的支持向量有关，所以说，SVM 不直接依赖数据分布，所得的划分超平面不受某一类点的影响；

3. 如果数据类别不平衡比较严重，LR 需要先做相应处理再训练，SVM 则不用；

4.SVM 依赖于数据表达的距离测度，需要先把数据标准化，LR 则不用 (但实际任务中可能会为了方便选择优化过程的初始值而进行标准化)。如果数据的距离测度不明确（特别是高维数据），那么最大间隔可能就变得没有意义；

5.LR 的输出有概率意义，SVM 的输出则没有；

6.LR 可以直接用于多分类任务，SVM 则需要进行扩展；

7.LR 使用的对率损失是光滑的单调递减函数，无法导出支持向量，解依赖于所有样本，因此预测开销较大；SVM 使用的 hinge 损失有“零区域”，因此解具有稀疏性（书中没有具体说明这句话的意思，但按我的理解是解出的拉格朗日乘子 a 具有稀疏性，而不是权重向量 ω），从而不需用到所有训练样本。

在实际运用中，LR 更常用于大规模数据集，速度较快；SVM 适用于规模小，维度高的数据集。

在 Andrew NG 的课里讲到过：

1. 如果 Feature 的数量很大，跟样本数量差不多，这时候选用 LR 或者是 Linear Kernel 的 SVM；

2. 如果 Feature 的数量比较小，样本数量一般，不算大也不算小，选用 SVM+Gaussian Kernel；

3. 如果 Feature 的数量比较小，而样本数量很多，需要手工添加一些 feature 变成第一种情况。

5.3 正则化

事实上，无论使用何种损失函数，SVM 的目标函数都可以描述为以下形式：

$$\min_f \Omega(f) + C \sum_{i=1}^{m} l(f(x_i), y_i) \quad (17)$$

在 SVM 中第一项用于描述划分超平面的“间隔”的大小，第二项用于描述在训练集上的误差。

更一般地，第一项称为结构风险 (structural risk)，用来描述模型的性质。第二项称为经验风险 (empirical risk)，用来描述模型与训练数据的契合程度。参数 C 用于权衡这两种风险。

前面学习的模型大多都是在最小化经验风险的基础上，再考虑结构风险（避免过拟合）。SVM 却是从最小化结构风险来展开的。从最小化经验风险的角度来看，表述了我们希望得到具有何种性质的模型（例如复杂度较小的模型），为引入领域知识和用户意图提供了路径（比方说贝叶斯估计中的先验概率）。

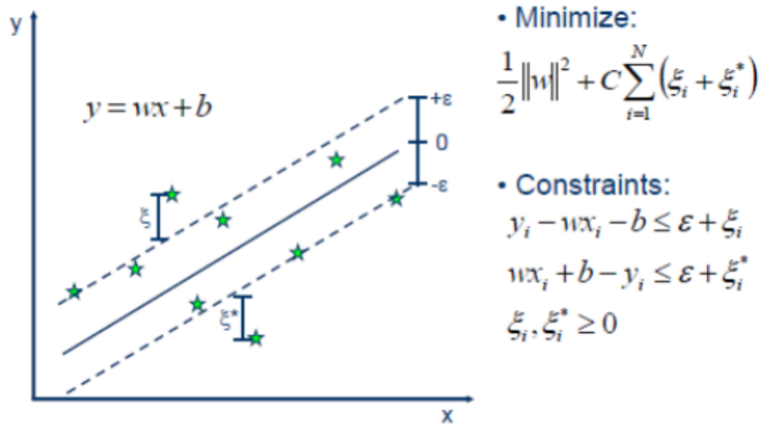
另一方面，$\Omega(f)$ 还可以帮我们削减假设空间，从而降低模型过拟合的风险。从这个角度来看，可以称 $\Omega(f)$ 为正则化（regularization）项，C 为正则化常数。正则化可以看作一种罚函数法，即对不希望出现的结果施以惩罚，从而使优化过程趋向于期望的目标。

L_p 范数是常用的正则化项，其中 L_2 范数 $||\omega||^2$ 倾向于 ω 的分量取值尽量稠密，即非零分量个数尽量多；L_0 范 $||\omega||_0$ 和范数 $||\omega||_1$ 则倾向于 ω 的分量取值尽量稀疏，即非零分量个数尽量少。

6 支持向量回归

同样是利用线性模型 $f(x) = \omega^T x + b$ 来预测，回归问题希望预测值和真实值 y 尽可能相近，而不是像分类任务那样，旨在令不同类的预测值可以被划分开。

传统的回归模型计算损失时直接取真实值和预测值的差，支持向量回归 (Support Vector Regression，简称 SVR) 则不然。SVR 假设我们能容忍最多有 ϵ 的偏差，只有当真实值和预测值之间相差超出了 ϵ 时才计算损失。



如图所示，以 SVR 拟合出的直线为中心，两边各构建出一个宽度为 ϵ 的地带，落在这个宽度为 2ϵ 的间隔带内的点都被认为是预测正确的。

因此，问题可以形式化为目标函数：

$$\min_{\omega,b} \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{m} l_\epsilon(f(x_i) - y_i)$$

其中 C 为正则化常数，l_ϵ 称为 $\epsilon-$ 不敏感损失 (insensitive loss) 函数。定义如下：

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

引入松弛变量 ξ_i 和 ξ_j，分别表示间隔带两侧的松弛程度，它们可以设定为不同的值。此时，目标函数式 (18) 可以重写为：

$$\min_{\omega,b} \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{m}(\xi_i + \hat{\xi}_i) \quad (19)$$

$$s.t. f(x_i) - y_i \leq \epsilon + \xi_i, y_i - f(x_i) \leq \epsilon + \xi_i, \xi_i \geq 0,$$

$$\hat{\xi}_i \geq 0, i = 1, 2, ..., m$$

注意这里有四组 m 个约束条件，所以对应地有四组拉格朗日乘子。

接下来就是用拉格朗日乘子法获得问题对应的拉格朗日函数，然后求偏导再代回拉格朗日函数，得到对偶问题。然后使用 SMO 算法求解拉格朗日乘子，最后得到模型，这里不一一详述了。

特别地，SVR 中同样有支持向量的概念，解具有稀疏性，所以训练好模型后不需保留所有训练样本。此外，SVR 同样可以通过引入核函数来获得拟合非线性分布数据的能力。

7 核方法

给定训练样本 $(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)$，若不考虑偏移项 b, 无论是 SVM 还是 SVR, 模型总可以表示为核函数 $\kappa(x, x_i)$ 的线性组合。不仅如此，我们有定理：

令 $\mathbb{H}$ 为核函数 κ 的再生希尔伯特空间，$||h||_{\mathbb{H}}$ 表示 $\mathbb{H}$ 空间中关于 h 的范数，对于任意单调递增函数 $\Omega : [0, \infty] \to \mathbb{R}$ 和任意非负损失的函数 $l : \mathbb{R}^m \to [0, \infty]$, 优化问题：

$$\min_{h \in \mathbb{H}} F(h) = \Omega(||h||_{\mathbb{H}}) + l(h(x_1, h(x_2, ..., h(x_m)))) \quad (20)$$

的解总可以写为：

$$h^x(x) = \sum_{i=1}^{m} a_i \kappa(x, x_i)$$

这个定理表明，对于形如式 (20)，旨在最小化损失和正则化项之和的优化问题，解都可以表示为核函数的线性组合。基于核函数的学习方法，统称为核方法 (kernal methods)。最常见的就是通过核化 (引入核函数)，将线性学习器扩展为非线性学习器。这不仅限于 SVM，事实上 LR 和 LDA 也都可以采用核函数，只是 SVM 使用 hinge 损失，解具有稀疏性所以用得更多。