Support Vector Machine

School of Mathematical Sciences
Fudan University
Zhou Yuan
15210180092

目录

1 支撑向量机Support Vector Machine

支撑向量机是一种分类模型，它的基本模型是定义在样本空间上的间隔最大的分类器，间隔最大使它有别于感知机；支撑向量机还包括核技巧，这是它称为实质上的非线性分类器。支撑向量机的学习策略就是讲最大化，可以形式化为一个求解凸二次规划的问题，也等于正则化的合页损失函数的最小化问题。支撑向量机的学习算法是求解凸二次规划的最优化算法。

支撑向量机学习方法包含构建由简至繁的模型：线性可分支撑向量机、线性支撑向量机及非线性支撑向量机。简单模型是复杂模型的基础，也是复杂模型的特殊情况。当训练数据线性可分时，通过硬间隔最大化，学习一个线性的分类器，即线性可分支撑向量机，又称为硬间隔支撑向量机；当训练数据近似线性可分时，通过软间隔最大化，也学习一个线性的分类器，即线性支撑向量机，又称为软间隔支撑向量机；当训练数据线性不可分时，通过使用核技巧及软间隔最大化，学习非线性支撑向量机。

当输入空间为欧式空间或离散集合、样本空间为希尔伯特空间时，核函数表示将输入从输入空间映射到特征空间得到的特征向量之间的内积。通过使用核函数可以学习非线性支撑向量机，等价于隐式地在高维的特征空间中学习线性支撑向量机，这样的方法称为核技巧。核方法是比支撑向量机更一般的机器学习方法。

1.1 线性可分支撑向量机

考虑一个二值分类问题，假设输入空间与特征空间为两个不同的空间。输入空间为欧式空间或离散集合，特征空间为欧式空间或希尔伯特空间。线性可分的支撑向量机、线性支撑向量机假设这两个空间的元素一一对应，并将输入空间中的输入映射为特征空间中的特征向量。非线性支撑向量机利用一个从输入空间到特征空间的非线性的映射将输入映射为特征向量。所以，输入都由输入空间转换到特征空间，支撑向量机的学习是在特征空间进行的。

假定给定一个特征空间上的训练数据集

$$T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$$

其中，$x_i \in \mathcal{X} = R^n, y_i \in \mathcal{Y} = \{+1, -1\}, i = 1, 2, \ldots, N$，$x_i$为第$i$个特征向量，$y_i$为$x_i$的类标记，当$y_i = +1$时，称$x_i$为正实例；当$y_i = -1$时，称$x_i$为负实例，$(x_i, y_i)$称为样本点。学习的目标是在特征空间找到一个分离超平面，能将样本分到不同的类。分离超平面对应于方程$w \cdot x + b = 0$，它的法向量w和截距b决定，可用(w, b)来表示。分离超平面将特征空间划分为两个部分，一部分是正类，一部分

是负类。法向量指向的一侧是正类，另一侧是负类。

一般的，当训练数据及现行可分时，存在误区多个分离超平面可将两类数据正确分离开。感知机利用误分类最小的策略，求得分离超平面，不过这是有无穷多个解，而线性可分支持向量机利用间隔最大化求最优分离超平面，解是唯一的。

给定线性可分训练数据集，通过间隔最大化或等价地求解相应的凸二次规划问题得到的分离超平面为$w^* \cdot x + b^* = 0$，以及相应的分类决策函数

$$f(x) = sign(w^* \cdot x + b^*) \tag{1}$$

称为线性可分支撑向量机。

一般来说，一个点距离超平面的远近可以表示分类预测的确信程度。在超平面$w \cdot x + b = 0$确定的情况下,超平面关于样本点的几何间隔$r_i = y_i(\frac{w}{||w||} \cdot x_i + \frac{b}{||w||})$是样本点到超平面的带符号的距离，当样本点被正确分类时，符号为正，距离为样本点到超平面的距离。间隔最大化的直观解释为：对训练数据集找到几何间隔最大的超平面意味着以充分大的确信度对训练数据进行分类。也就是说，不仅将正负样本点分开，而且对最难分的样本点（离超平面最近的点）也有最够大的确信度将它们分开。这样的超平面应该对未知的样本点有很好的分类预测能力。

超平面(w, b)关于训练数据集T的几何间隔为超平面关于T中所有样本点(x_i, y_i)的几何间隔的最小值$r = \min_{i=1,\ldots,N} r_i$。最大间隔分离超平面问题可以表示为下面的约束问题：

$$\begin{aligned} \max_{w,b} \quad & r \\ s.t. \quad & y_i(\tfrac{w}{||w||} \cdot x_i + \tfrac{b}{||w||}) \geq r, \quad i = 1, 2, \ldots, N \ . \end{aligned}$$

可以将上述优化问题等价改写为：

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}||w||^2 \\ s.t. \quad & y_i(w \cdot x_i + b) - 1 \geq 0, \quad i = 1, 2, \ldots, N \ . \end{aligned}$$

在线性可分的情况下，训练数据集中的样本点与分离超平面距离最近的样本点成为支持向量。在决定分离超平面的时候，只有支持向量起作用，而其他样本点不起作用。如果移动支持向量将改变所求的解；但是如果在间隔边界以外移动其他的样本点，甚至去掉这些点，则解释不会改变的。由于支持向量机在确定分离超平面中起着决定性的作用，所以将这种分类模型称为支撑向量机。支持向量的个数一把很少，所以支持向量机一般由很少的“重要的”训练样本确定。

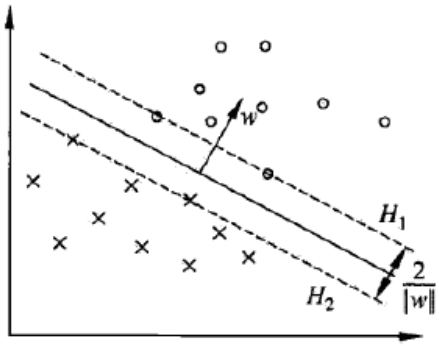

图 1: 支持向量

如上图，在H_1, H_2上的点就是支持向量，注意到H_1, H_2平行，并且没有实例点落在它们中间。在H_1, H_2之间形成一条长带，分离超平面与它们平行且位于它们中央，长带的宽度等于$\frac{2}{||w||}$。

将上面的约束问题构建拉格朗日函数，引入拉格朗日乘子$\alpha_i \geq 0, i = 1, \ldots, N$，得到的拉格朗日函数为：

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

利用对偶性，可以得到原优化问题的对偶问题：

$$\begin{aligned} \max_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N} \alpha_i \\ s.t. \quad & \textstyle\sum_{i=1}^{N} \alpha_i y_i = 0, \quad i = 1, 2, \ldots, N \ . \end{aligned}$$

此时，分离超平面可以写成：

$$\sum_{i=1}^{N} \alpha_i^* y_i (x \cdot x_i) + b^* = 0 \tag{3}$$

分类决策函数可以写为：

$$f(x) = sign(\sum_{i=1}^{N} \alpha_i^* y_i (x \cdot x_i) + B^*) \tag{4}$$

可以看出分类决策函数只依赖于输入x和训练样本输入的内积。

1.2 线性支撑向量机

线性可分问题的支撑向量机学习方法，对线性不可分训练数据是不适用的。通常情况是，训练数据中有一些特异点，将这些特异点去除后，剩下的大部分样本点组成的集合是线性可分的。

线性不可分意味着某些样本点不能满足函数间隔大于等于1的约束条件。为了解决这个问题，对每个样本点引进一个松弛变量$\xi_i \geq 0$，使函数间隔加上松弛变量大于等于1，这样，新的优化问题是：

$$\begin{aligned} \max_{w,b,\xi} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\xi_i \\ s.t. \quad & y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \ldots, N \\ & \xi_i \geq 0, \qquad\quad i = 1, 2, \ldots, N \end{aligned} .$$

对应的对偶优化问题是：

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N}\alpha_i \\ s.t. \quad & \textstyle\sum_{i=1}^{N}\alpha_i y_i = 0, \quad i = 1, 2, \ldots, N \\ & \alpha_i \leq C, \quad i = 1, 2, \ldots, N \\ & \alpha_i \geq 0, \quad i = 1, 2, \ldots, N \end{aligned} .$$

分离超平面可以写成：

$$\sum_{i=1}^{N}\alpha_i^* y_i (x \cdot x_i) + b^* = 0 \tag{5}$$

分类决策函数可以写成

$$f(x) = sign(\sum_{i=1}^{N}\alpha_i^* y_i (x \cdot x_i) + b^*) \tag{6}$$

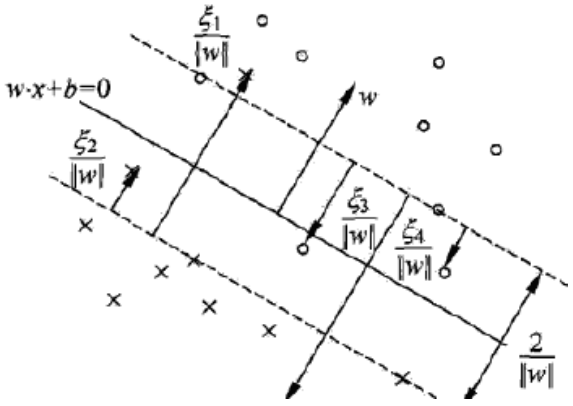

图 2: 软间隔的支持向量

如上图，软间隔的支持向量x_i或者在间隔边界上，或者在间隔边界与分离超平面之间，或者在分离超平面误分的一侧。

1. 若$\alpha_i^* < C$，则$\xi_i = 0$, 支持向量恰好落在间隔边界上；

2. 若$\alpha_i^* = C, 0 < \xi_i < 1$，则分类正确，$x_i$在间隔边界与分离超平面之间；

3. 若$\alpha_i^* = C, \xi_i = 1$，则$x_i$在分离超平面上；

4. 若$\alpha_i^* = C, \xi_i > 1$，则$x_i$位于分离超平面误分类的一侧。

1.3 非线性支持向量机与核函数

非线性分类问题是指通过非线性模型才能很好地进行分类的问题。一般来说，对给定的一个训练样本集$T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$，其中，实例$x_i$属于输入空间，$x_i \in \mathcal{X} = R^n$，对应的标记有两类$y_i \in \mathcal{Y} = \{-1, +1\}, i = 1, 2, \ldots, N$。如果能用$R^n$中的一个超曲面将正负实例正确分开，则称这个问题为非线性可分问题。

用线性分类方法求解非线性分类问题分为两步：

- 首先使用一个变换将原空间的数据映射到新空间；
- 然后在新空间里用线性分类学习的方法从训练数据中学习分类模型

这也就是所谓的核技巧。核技巧应用到支持向量机，其基本想法就是通过一个非线性变换将输入空间对应于一个特征空间，使得在输入空间中的超曲面模型对应于特征空间中的超平面模型。这样，分类问题的学习任务通过在特征空间中求解线性支持向量机就可以完成。

如果存在一个映射是从输入空间$\mathcal{X}$到特征空间$\mathcal{H}$ 的映射：$\phi(x) : \mathcal{X} \to \mathcal{H}$，使得所有的$x, z \in \mathcal{X}$，函数$K(x, z)$满足条件：

$$K(x, z) = \phi(x) \cdot \phi(z) \tag{SVM.2}$$

则称函数$K(x, z)$为核函数，$\phi(x)$为映射函数，$\phi(x) \cdot \phi(z)$ 表示内积。核技巧的想法是，在学习与预测中只定义核函数$K(x, z)$，而不是显式地定义映射函数ϕ。通常，直接计算函数$K(x, z)$比较容易，而通过$\phi(x), \phi(z)$ 计算函数$K(x, z)$并不容易。注意，ϕ是输入空间R^n 到特征空间$\mathcal{H}$的映射，特征空间$\mathcal{H}$一般是高维的，甚至是无穷维的。对于给定的核函数$K(x, z)$，特征空间和映射函数的取法可以是不唯一的，可以取不容的特征空间，即便是在同一特征空间里也可以取不同的映射。

经过映射函数ϕ 将原来的输入空间变换到一个新的特征空间，将输入空间中的内积$x_i \cdot x_j$ 变换为特征空间中的内积$\phi(x) \cdot \phi(z)$，在新的特征空间里从训练样本中学习线性支持向量机。当映射函数是非线性函数时，学习到的含有核函数的支持向量机是非线性的分类器。也就是说，在核函数$K(x, z)$给定的条件下，可以利

用解线性分类问题的方法求解非线性分类问题的支持向量机。学习是隐式的在特征空间中进行的，不需要显式的定义特征空间和映射函数。这样的技巧就是核技巧，它是巧妙地利用线性分类学习方法与核函数解决非线性问题的技术。在实际应用中，往往依赖领域知识直接选择核函数，核函数的选择的有效性需要通过实验验证。

此时对偶问题的目标函数变为：

$$W(\alpha)=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i,x_j)-\sum_{i=1}^{N}\alpha_i \tag{7}$$

同样，分类决策函数中的内积也可以用核函数来代替：

$$f(x)=sign(\sum_{i=1}^{N}\alpha_i^* y_i\phi(x_i)\cdot\phi(x)+b^*)=sign(\sum_{i=1}^{N}\alpha_i^* y_i K(x_i,x)+b^*) \tag{8}$$

常用的核函数有：

1. 多项式核函数

$$K(x,z)=(x\cdot z+1)^p \tag{9}$$

对应的支撑向量机是一个p次多项式分类器，分类决策函数为：

$$f(X)=sign(\sum_{i=1}^{N}\alpha_i^* y_i(x_i\cdot x+1)^p+b^*) \tag{10}$$

2. 高斯核函数

$$K(x,z)=exp(-\frac{||x-z||^2}{2\sigma^2}) \tag{11}$$

对应的支撑向量机是高斯径向基函数分类器，分类决策函数是：

$$f(x)=sign(\sum_{i=1}^{N}\alpha_i^* y_i exp(-\frac{||x-z||^2}{2\sigma^2})+b^*) \tag{12}$$

2 Convergent-cross mapping,CCD

算法的主要思想是：

输入：时间序列X, Y

输出：X, Y之间的因果关系。

1. 对于时间序列$X(t)$用特定的方法选定重构的维数m与时间延迟tau；
2. 对时间序列$X(t)$按照上述选出的维数与时间延迟进行重构；
3. 对重构后的系统中，每一个点$(X(t^*), X(t^* - \tau), \ldots, X(t^* - m\tau))$找出其对应最近邻的$N$个点；
4. 用这N个点对应的下标时间t，找出$Y(t)$序列对应的N个点；
5. 用$Y(t)$序列中找出的这N个点对$Y(t^*)$做出估计，例如$Y(t^*) = \frac{1}{N}\sum_{i=1}^{N} Y(t_i)$，得到$Y(t)$序列的估计$\hat{(Y)}(t)$。
6. 根据不同时间t，求出$Y(t), \hat{(Y)}(t)$的相关系数序列$\rho_{XtoY}(t)$。
7. 用同样的方法，得到$X(t)$的序列的估计$\hat{(X)}(t)$，求出相关系数序列$\rho_{YtoX}(t)$。
8. 判断因果的方法：例如$\rho_{XtoY}(t)$序列的相关系数较大，且主要占据上方，呈收敛状态，则序列$X(t)$是序列$Y(t)$的结果，序列$Y(t)$是序列$X(t)$的原因。