机器学习讲义(L11-C)：支持向量机与核方法

授课教师：王贝伦 / 助教：张嘉琦，黄旭，谈笑，徐浩卿

在上一节中我们简单介绍了在支持向量机中使用核方法——**软间隔支持向量机**——的相关内容。在这一节，我们将详细说明有关核方法和支持向量机的更多内容。

1 软间隔支持向量机

我们先回顾一下软间隔支持向量机。

对于线性可分的数据，硬间隔支持向量机就可以达到很好的效果。然而对于线性不可分的数据，我们需要利用映射函数 $\Phi(\cdot)$ 将数据映射到高维空间中，从而使数据线性可分（如图1所示）。

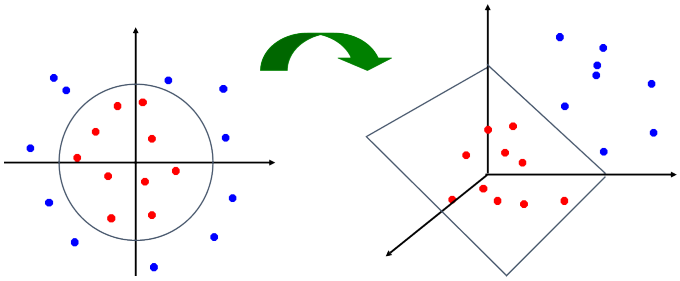

Figure 1: 将线性不可分数据映射到高维空间。

我们定义核函数为

$$K(x,z)=\Phi(x)^T\Phi(z) \tag{1}$$

实际使用中，高斯核比较常用，即 $K(x,z)=\exp(-\frac{||x-z||_2^2}{2\sigma^2})$。

考虑硬间隔分类器的对偶形式，我们有

$$\begin{aligned}\max_{\alpha}\quad & \sum_{i=1}^{N}\alpha_2=\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j x_i^T x_j \\ \text{s.t.}\quad & \alpha_i\geq 0 \\ & \sum_{i=1}^{N}\alpha_i y_i=0,\ \text{for } i=1,2,\cdots,N.\end{aligned} \tag{2}$$

通过核函数将数据内积直接映射到高维空间，我们就可以用尽量少的时间成本将数据转化为线性可分状态。

$$\begin{aligned}\max_{\alpha}\quad & \sum_{i=1}^{N}\alpha_2=\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i,x_j) \\ \text{s.t.}\quad & \alpha_i\geq 0 \\ & \sum_{i=1}^{N}\alpha_i y_i=0,\ \text{for } i=1,2,\cdots,N.\end{aligned} \tag{3}$$

2 核方法的性质

2.1 有效性

我们之前只说核方法能够将数据再高维空间变为线性可分。但是我们需要将数据映射到纬度为多少的空间时，数据才线性可分？对于任意的数据集，是否都存在一个空间使得数据映射之后线性可分？要回答这些问题，我们需要解释这样一个概念：**VC 维(Vapnik-Chervonenkis dimension)**。VC 维是一种被用来衡量研究对象（数据集与学习模型）学习能力的指标。VC 维的解释比较复杂，不属于这门课程的讲授范围，有兴趣可以参考引用材料 [1,2,3]。这里我们只介绍 VC 维在支持向量机中的使用。

定义：分类算法的 VC 维

假设我们有一个拥有参数 θ 的分类模型 f。如果存在一个 θ，使得 f 能够无错误地分类所有数据点 $(x_1,x_2,\cdots,x_n)$，我们就称模型 f **分散 (shatter)** 了数据集 $(x_1,\cdots,x_n)$。一个分类模型的 VC 维被定义为这个模型能分散的数据的数量。

我们来看一个简单的例子。假设我们在二维空间中有一个线性分类算法，即分类模型 f 需要在二维空间中学习到一条线性的分类边界。如图所示，在二维空间，对于任意的三个点，我们永远能找到一条直线来完美地区分不同类的数据。然而当我们有三个点时，有时我们找不到这样一条完美的分界线。那么对于一个二维空间的线性分类器，它的 VC 维就等于 3.

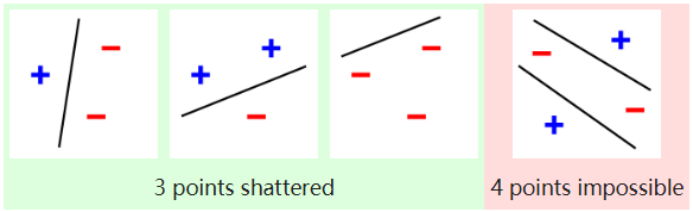

Figure 2: VC 维为 3 的线性分类器示例

一般来说，对于一个 N 维空间中的线性分类器，它的 VC 维是 $N+1$，即分类器能完美分散 $N+1$ 个数据点。那么这和我们的软间隔支持向量机有什么关系？我们之前说过，软间隔支持向量机的核心思想就是将 n 个数据点映射到高维空间中以变得线性可分。我们实际上需要的就是一个在高维空间 $\mathbb{R}^M$ 中 VC 维等于 n 的支持向量机。这也说明空间的维度应该是 $M=n+1$。通过不同的核函数，我们一般都能将数据转换到一个 $n+1$ 维的空间中。这就证明了软间隔支持向量机的有效性。

注意到 VC 维的定义需要分类器“无误差”地分类样本，但在实际使用中我们可以接受一些错误的存在。所以我们可以将数据映射到稍低一些的维度中。另外我们可以看到，一个模型的 VC 维只与模型的复杂度以及数据样本数量有关。也就是说数据本身的特征数量以及数据分布并不会影响一个模型的学习能力。

2.2 高效性

我们之前说过，核函数的作用就是让我们在低维空间中直接计算两个数据样本在高维空间中的内积。这样时间复杂度会大幅度地降低。我们以二次核函数 $K(x,z)=(1+x^Tz)^2$ 为例，假设两个数据 $x,z\in\mathbb{R}^p$，二次核函数对应的映射函数 $\Phi(x)$ 会将数据 $x=[x_1,x_2,\cdots,x_p]^T$ 映射到 $\mathbb{R}^{p^2}$ 空间中：

$$\Phi(x) = \left[1, \sqrt{2}x_1, \cdots, \sqrt{2}x_p, x_1^2, \cdots, x_2^p, \sqrt{2}x_1x_2, \cdots, \sqrt{2}x_{p-1}x_p\right]^T \tag{4}$$

在新的空间中计算两个向量的内积 $\Phi(x)^T\Phi(z)$ 的时间复杂度为 $O(p^2)$。而通过核函数 $K(x,z)$，我们只需要花费 $O(p)$ 的时间成本就能计算出高维空间的向量内积。二次核函数是一种比较简单的情况，我们考虑一般的多项式核函数 $K(x,z) = (1 + x^Tz)^d$。在公式3中，如果我们在高维空间中直接计算内积，需要的时间复杂度为 $O(p^dn^2)$。而利用核函数，我们运算的时间复杂度只有 $O(pn^2)$。在变量数量很多的情况下，核函数能节省大量的计算时间。

3 选择核函数

支持向量机，尤其是软间隔支持向量机的性能很大程度上取决于核函数的选择。可选择的核函数种类非常多，对于不同的应用与目的，我们需要选择不同的核函数。一般来说，当我们不知道选择用什么核函数时，多项式核或者 RBF 核是一个比较合理的选择。

4 支持向量机的预测过程

从公式3中我们可以看到，拉格朗日乘子 α 只与每个样本有关 $(\alpha_i, \alpha_j$ 对于 $1 \le i, j \le n)$。因为 θ 是通过 α 计算的，θ 也与样本有关。注意到只有支持向量对应的 $\alpha \neq 0$，即支持向量对应的 θ 会对预测产生效果。给定一个新的数据 x_{new}，我们预测它的样本标签的过程为

$$\begin{aligned} \widehat{y}_{\text{new}} &= \text{sign}(\theta^T x_{\text{new}} + b) \\ &= \text{sign}(\sum_{i=1}^{n} \alpha_i y_i x_i^T x_{\text{new}} + b) \\ &= \text{sign}(\sum_{i \in \text{SV}} \alpha_i y_i x_i^T x_{\text{new}} + b) \end{aligned} \tag{5}$$

其中 SV 表示所有支持向量的集合。我们只需要使用支持向量就可以预测出一个新样本的标签。

5 支持向量机的有效性

我们已经基本介绍完了支持向量机的相关知识。但是仍然存在一个问题：当我们用核函数将数据映射到高维空间中时，我们的变量数量会变得非常大，但为什么我们不考虑支持向量机可能存在过拟合的情况？最主要的原因是在支持向量机中，我们要预测的参数个数与样本数量有关，而不是与变量数量有关。在公式5中我们看到 α 被分配到每一个样本上。而且只有支持向量的 α 不等于 0，这也说明我们只有很少数量的参数 α 需要预测。还有一个原因就是，支持向量机的优化问题关注最大化间隔。这可以被看成是一种正则项，对获得较小间隔的分类器进行惩罚。因此，支持向量机在某种程度上是通过最大化间隔来降低过拟合性的。

引用

[1] Vapnik–Chervonenkis dimension: https://en.wikipedia.org/wiki/Vapnik%E2%80%93Chervonenkis_dimension.

[2] Definition of VC Dimension: https://www.coursera.org/lecture/ntumlone-mathematicalfoundations/definition-of-vc-dimension-AnYJ6

[3] VC Dimension: https://beader.me/mlnotebook/section2/vc-dimension-one.html