支持向量机算法原理及实践

www.huawei.com

前言

- 支持向量机属于监督学习类算法，在解决小样本，非线性以及高维识别问题时有很大优势。本章主要从支持向量机的原理，主要特性、原理、应用等角度进行学习。

课程目标

- 学完本课程后，您将能够：
  - 理解支持向量机的原理
  - 掌握支持向量机的主要特征
  - 理解支持向量机应用场景
  - 能将支持向量机应用于相关的简单场景

目录

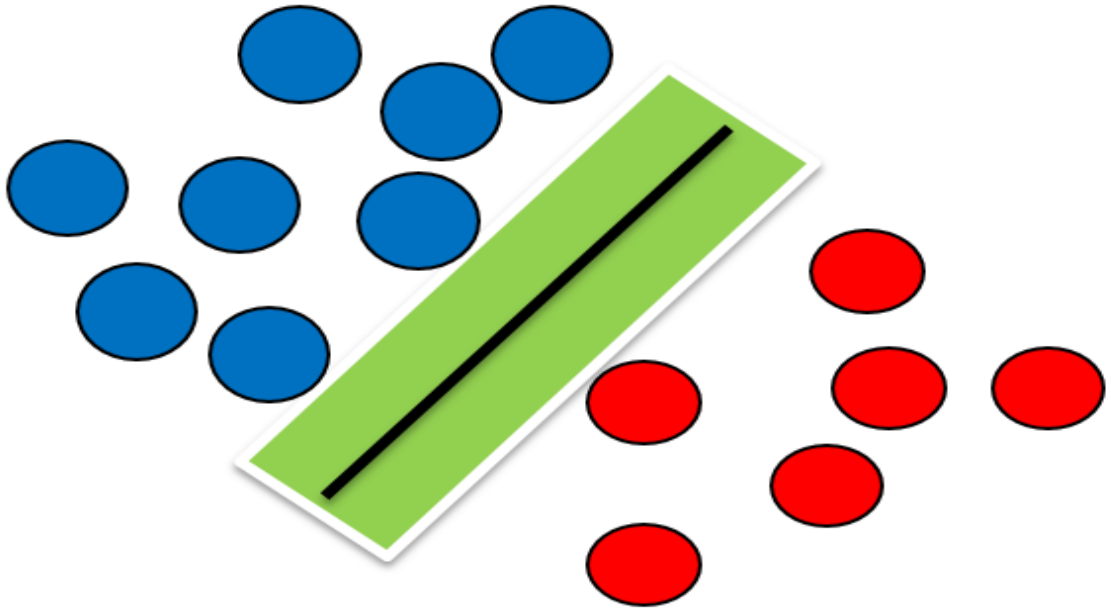
SVM基本概念

- Support Vector Machine （支持向量机）：
  - 支持向量：支持或支撑平面上把两类类别划分开来的超平面的向量点。
  - 机：一个算法
- SVM是基于统计学习理论的一种机器学习方法。简单地说，就是将数据单元表示在多维空间中，然后在这个空间中对数据做划分的算法。
- SVM是建立在统计学习理论的VC维理论和结构风险最小原理基础上的，根据有限的样本信息在模型的复杂性之间寻求最佳折衷，以期获得最好的推广能力（或泛化能力）。所谓VC维是对函数类的一种度量，可以简单的理解为问题的复杂程度，VC维越高，一个问题就越复杂。正是因为SVM关注的是VC维，因此SVM解决问题的时候，和样本的维数是无关的。（甚至样本是上万维的都可以，这使得SVM很适合用来解决文本分类的问题，当然也是因为因为引入了核函数）。

目录

线性分类

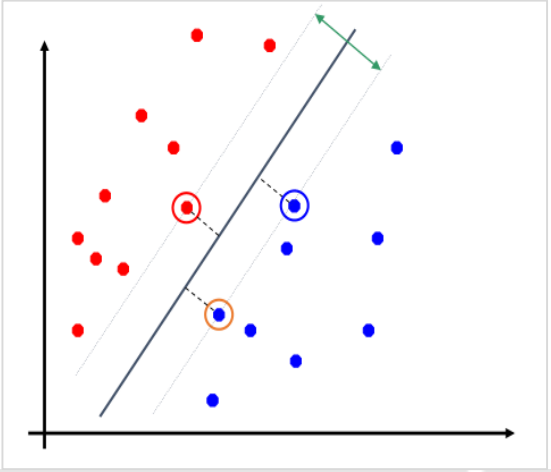
- SVM就是试图把一根棍子放在最佳位置，以达到分类的目的，且让棍的两边有尽可能大的间隙，这个间隙就是球到棍的距离。



目录

3.1 线性SVM

- 在保证决策面方向不变且不会出现错分样本的情况下移动决策面，会在原来的决策面两侧找到两个极限位置（越过该位置就会产生错分现象），如虚线所示。虚线的位置由决策面的方向和距离原决策面最近的几个样本的位置决定。而这两条平行虚线正中间的分界线就是在保持当前决策面方向不变的前提下的最优决策面。两条虚线之间的**垂直距离就是这个最优决策面对应的分类间隔。**
- 显然每一个可能把数据集正确分开的方向都有一个最优决策面（有些方向无论如何移动决策面的位置也不可能将两类样本完全分开），不同方向的最优决策面的分类间 隔通常是不同的，那个具有“最大间隔”的决策面就是SVM要 寻找的最优解。**而这个真正的最优解对应的两侧虚线所穿过的样本点，就是SVM中的支持样本点，称为“支持向量”。**



3.2 目标函数—数学建模

- 求解这个“决策面”的过程，就是最优化。一个最优化问题通常有两个基本的因素：
  - 目标函数：也就是你希望什么东西的什么指标达到最好。
  - 优化对象：你期望通过改变哪些因素来使你的目标函数达到最优。
- 在线性SVM算法中，目标函数显然就是那个“分类间隔”，而优化对象则是决策面。所以要对SVM问题进行数学建模，首先要对上述两个对象（“分类间隔”和“决策面”）进行数学描述。我们先描述决策面。
- N 维空间的超平面方程可以表示为：

$$w^T x + \gamma = 0$$

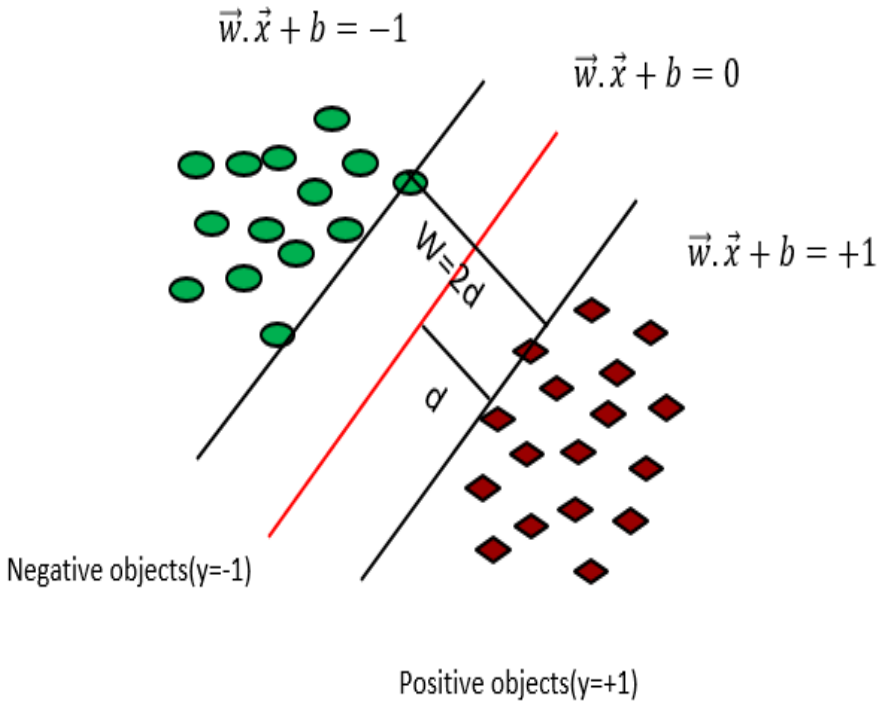
3.2 目标函数—约束条件

- 如图所示，以二维平面为例。要求解的是W的最大化。图中红颜色的圆点标记为1，规定其为正样本；蓝颜色的五星标记为-1，我们人为规定其为负样本。
- 假设决策面正好处于间隔区域的中轴线上，并且相应的支持向量对应的样本点到决策面的距离为d，那么公式进一步写成：

$$\begin{cases} w^T x_i + \gamma \geq 1 & \forall y_i = 1 \\ w^T x_i + \gamma \leq -1 & \forall y_i = -1 \end{cases}$$

- 则约束条件为：

$$y_i(w^T x_i + \gamma) \geq 1 \ \forall x_i$$



3.2 目标函数的确定

- 目标函数：

$$d = \frac{|w^T x+\gamma|}{\|w\|}$$

- 目标函数进一步简化（取临界线，则分子为1）：

$$d = \frac{1}{\|w\|}$$

- 因为上述函数依然为非凸函数，所以对上述函数进行转换，转换成为一个凸函数。因为求解d的最大化问题，实际是$\|w\|$的最小化问题。进而 $\|w\|$ 的最小化问题等效于：

$$min\frac{1}{2}\|w\|^2$$

- 最终的目标函数和约束条件放在一起进行描述(s.t. subject to)：

$$min\frac{1}{2}\|w\|^2$$

$$\text{s.t. } y_i(w^T x_i +\gamma) \geq 1, i = 1,2,\dots,n$$

3.3 拉格朗日函数优化

- **使用拉格朗日函数优化的目的：**它将约束条件放到目标函数中，从而将有约束优化问题转换为无约束优化问题。
- 将有约束的原始目标函数转换为无约束的新构造的拉格朗日目标函数公式如下：

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

- 其中α_i是拉格朗日乘子，α_i 大于等于0，是构造新目标函数时引入的系数变量（我们自己设置）。令：

$$\theta(w) = \max_{\alpha_i \geq 0} L(w,b,\alpha)$$

- 当样本点不满足约束条件时，即在可行解区域外：

$$y_i(w^T x_i + b) < 1$$

- 将α_i设置为正无穷，此时$\theta(w)$显然也是正无穷。
- 当样本点满足约束条件时，即在可行解区域内：

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

- 此时，显然$\theta(w)$为原目标函数本身。将上述两种情况结合一下就是新的目标函数。

3.3.1 拉格朗日对偶

- **新的目标函数为：**

$$\theta(w) = \begin{cases} \frac{1}{2}\|w\|^2 & x\epsilon可行区域 \\ +\infty & x\epsilon非可行区域 \end{cases}$$

- 问题就变成了求新目标函数的最小值，即：

$$\min_{w,b} \theta(w) = \max_{\alpha_i \geq 0} \min_{w,b} L(w,b,\alpha) = p$$

- 拉格朗日对偶优化： 新目标函数先求最大值，再求最小值。首先就要面对带有需求求解的参数w和b的方程，而α_i又是不等式约束，这个求解过程不好做。所以需要使用拉格朗日函数对偶性，将最小和最大的位置交换一下，这样就变成了：

$$\min_{w,b} \max_{\alpha_i \geq 0} L(w,b,\alpha) = d$$

- 交换以后的这个新的问题最优值用d来表示。而且d≤p，而我们真正需要关心的问题是d=p的时候， 即，拉格朗日对偶处理之后，我们的解没有改变。

3.3.2 KKT条件

- **若想得到d=p，需要满足的条件是：**
  - 凸函数：我们已在前面进行了凸优化
  - KKT条件
- **KKT(Karush-Kuhn-Tucker)条件的最优值条件必须满足以下条件：**
  - 条件一：经过拉格朗日函数处理之后的新目标函数 L(w, b, α)对α求导为零（即保证原目标函数 $\frac{1}{2}\|w\|^2$ 与限制条件$\sum_{i=1}^{n}\alpha_i(y_i(w^T x_i$ +b)-1) 必须有交点，也即是必须要有解）
  - 条件二：h(x) = 0
  - 条件三：α*g(x) ≥ 0

3.3.2 KKT条件 (续)

- 第一步：首先固定α，要让L(w, b, α)关于w和b最小化，分别对w和b求偏导数，令其等于0， 即：
  - $L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n}\alpha_i(y_i(w^T x_i + b)-1)$
  - $\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n}\alpha_i y_i x_i$
  - $\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n}\alpha_i y_i = 0$
  - 将上述结果带回$L(w,b,\alpha)$，即求得内侧的最大值化，化简得：

$$L(w,b,\alpha)=\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j$$

3.3.3 SMO优化

- 第二步：现在内侧的最大值求解完成，再求解外侧的最小值，从上式可以得到:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i \ -\frac{1}{2} \sum_{i=1}^{n} \alpha_i \, \alpha_j y_i y_j x_i^T x_j$$

$$\text{s.t. } \alpha_i \geq 0, i = 1,2, \dots, n$$

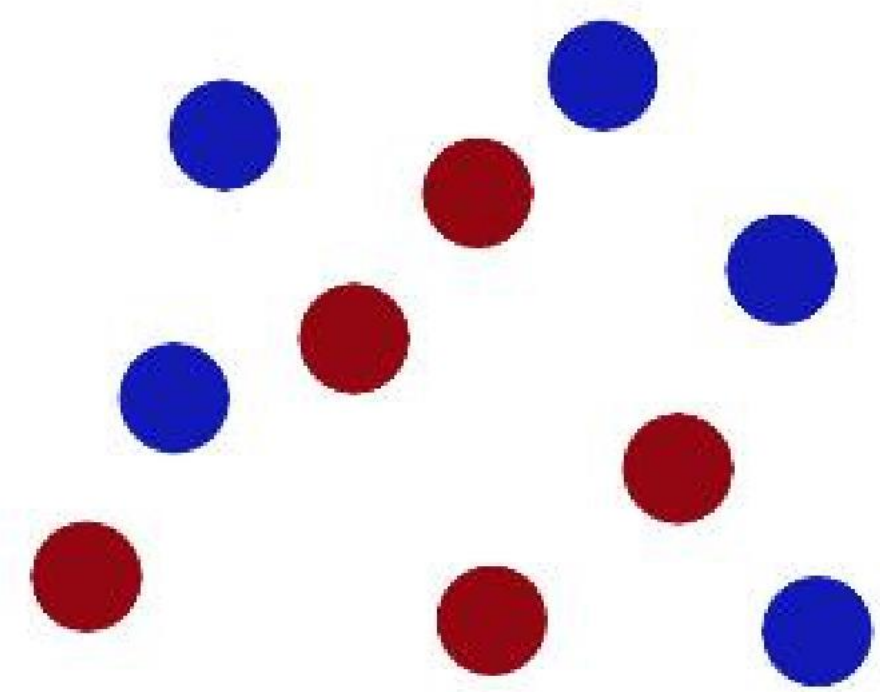
$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

- 第三步：优化问题已经化简到上式的形式，对于这个问题，有更高效地优化算法，即序列最小优化(SMO)算法。通过这个优化算法能得到α，再根据α，就可以解出w和b，进而求得最初的目的：找到超平面，即“决策平面”。

目录

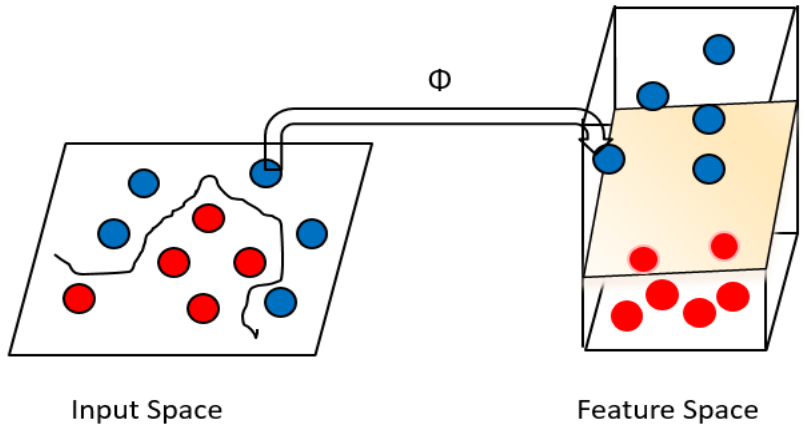
非线性分类

- 假如球放为如下情况，如何分类：



非线性分类 (续)

- 桌子一拍，求飞到空中。然后，用一张纸插到两种球的中间。

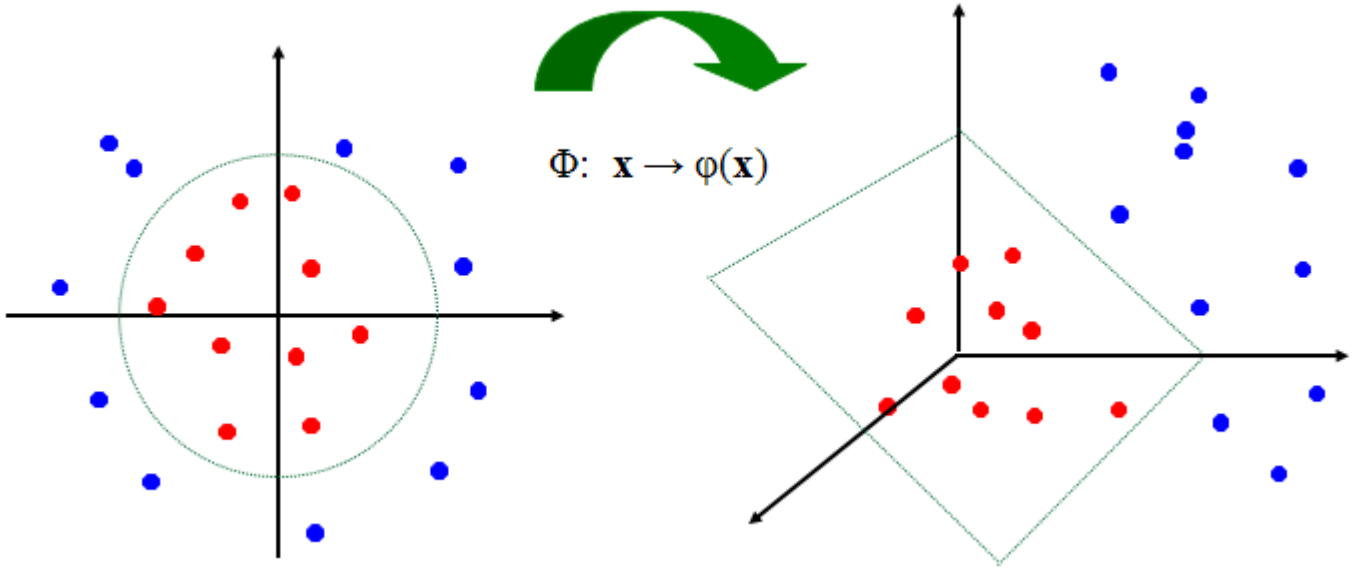


- 球叫做data，把棍子叫做分类器(classifier)，找到最大间隙的窍门(trick)叫做最优化(optimization)，拍桌子叫做核函数(kernel)，那张纸叫做超平面(hyperplane)。

目录

5.1 非线性SVM

- 对于以上所述的SVM，处理能力还是较弱，仅仅能处理线性可分的数据。如果数据线性不可分的时候，我们就将低维的数据映射向更高的维度，以此使数据重新线性可分。这转化的关键便是核函数。



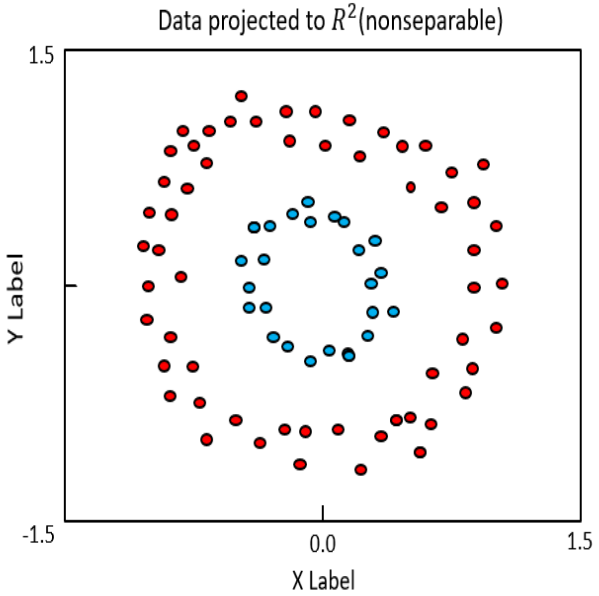
5.2 映射关系

- 如右图为两个半径不同的圆圈加上少量的噪音得到，所以一个理想的分界应该是一个圆圈而不是一条直线。如果用X_1和X_2来表示这个二维平面的两个坐标的话则方程可以写作：

$$a_1x_1+a_2x_1^2+a_3x_2+a_4x_2^2+a_5x_1x_2+a_6=0$$

- 根据上述形式，如果我们构造另外一个所谓的空间，其中五个坐标的值分别为：

$$z_1=x_1,\ z_2=x_1^2,\ z_3=x_2,\ z_4=x_2^2,\ z_5=x_1x_2$$



5.2 映射关系 (续)

- 将R^2空间映射到R^5:

$$(x_1, x_2) \rightarrow (z_1, z_2, z_3, z_4, z_5)$$

- 那么显然上面的方程在新的坐标系下可以写作:

$$\sum_{i=1}^{5} a_i Z_i + a_6 = 0$$

- 此时，总能找到一个超平面$w^T Z + b = 0$

$$w^T = \{a_1, a_2, a_3, a_4, a_5\}^T, b = a_6$$

- 将低维X按照上面的规则映射为高维Z，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倒的线性分类算法就可以进行处理了，这也正是核函数处理非线性问题的基本思想。

5.3 核函数

- 由于从输入空间到特征空间的这种映射会使得维度发生爆炸似地增长，这给映射过程中的内积的计算带来了很大地困难，而且如果遇到无穷维的情况就根本无法计算。而且如果先将数据从低维映射到高维后，再计算两数据的内积，计算量会非常大，因此核函数就此被引入用来解决SVM分类的非线性问题。
- **核函数：**基本作用是接受两个低维空间里的向量，能够计算出经过某个变换后在高维空间里的向量的内积。因此只需要在输入空间内就可以进行特征空间的内机。
- 通过上述描述，我们知道要想构造核函数，需要明确输入空间内数据的分布情况，我们并不知道自己所处理的数据的具体分布，故一般很难构造出完全符合输入空间的核函数。因此常用几种常用的核函数来代替构造核函数。

5.3 核函数（续）

- 常用的核函数一般有：
  - 线性核函数(Linear Kernel):

$$K(X, X_i) = (\mathrm{X}.\ X_i)$$

  - 多项式核函数(Polynomial Kernel)：

$K(X, X_i)=(s(\mathrm{X}.\ X_i) + c)^d$, 其中s, c, d为参数

  - 径向基（高斯）核函数(Radical basis function Kernel)：

$K(X, X_i)=\exp(-\gamma|x - X_i|^2)$, 其中γ为参数

  - Sigmoid核函数：

$K(X, X_i)=\tanh(s(\mathrm{X}.\ X_i)+c)$, 其中s,c为参数

5.3 核函数 (续)

- **线性核函数：**主要用于**线性可分**的情况，我们可以看到特征空间到输入空间的维度是一样的，但是其参数减少速度快，对于线性可分数据，其分类效果很理想且效率更高。因此我们通常首先使用线性核函数来做分类，如果不行再换用其他核函数。
- **多项式核函数：**多项式核函数可以实现将低维的输入空间映射到高维的特征空间，但是多项式核函数的参数多，当多项式的阶数比较高的时候，核矩阵的元素值将趋于无穷大或者无穷小，计算复杂度是会大到无法计算（线性核函数可以看作多项式核函数的一种）。

5.3 核函数 (续)

- **高斯核函数：**在常用的核函数中，使用最广泛的就是RBF核，无论低维、高维、小样本、大样本等情况，RBF核都适用，具有较宽的收敛域，是较理想的分类依据函数。
- **Sigmoid核函数：**采用Sigmoid核函数，支持向量机实现得就是一种多层神经网络。

5.3 核函数（续）

- 核函数的引入避免了“维数灾难”，大大减小了计算量。而输入空间的维数n对核函数矩阵无影响，因此，核函数方法可以有效处理高维输入。
- 无需知道非线性变换函数Φ的形式和参数。
- 核函数的形式和参数的变化会隐式地改变从输入空间到特征空间的映射，进而对特征空间的性质产生影响，最终改变各种核函数方法的性能。
- 核函数方法可以和不同的算法相结合，形成多种不同的基于核函数技术的方法，且这两部分的设计可以单独进行，并可以为不同的应用选择不同的核函数和算法。

本章小结

- 本章节主要介绍了支持向量机算法，分别从线性可分问题和线性不可分问题两个角度来讲解SVM的算法原理。
- **线性可分：**
  - 求解使得超平面具有最大内间间隔的w^T，b参数。
  - 将问题转化为对偶问题进行快速求解。
- **线性不可分：**
  - 将数据空间映射到高维空间，使原本线性不可分变为线性可分。
  - 引入核函数，简化映射空间中的内积运算。它避开了直接在高维空间中进行计算，而表现形式却等价于高维空间。
  - 不同的样本结构与不同的核函数结合，达到很好地分割效果。

思考题

1. 常用的核函数有？（　　）

   A. 线性核函数

   B. 多项式核函数

   C. 高斯核函数

   D. Sigmoid核函数

2. 支持向量机使用的场景？（　）

   A. 多分类

   B. 二分类

   C. 回归

   D. 聚类

学习推荐

- 华为Support案例库
  - https://support.huawei.com/carrierindex/zh/anony/index.html
- ICT人才交流学习社区
  - http://cn.hiclc.com/