# 机器学习 - 支持向量机

# 本章目录

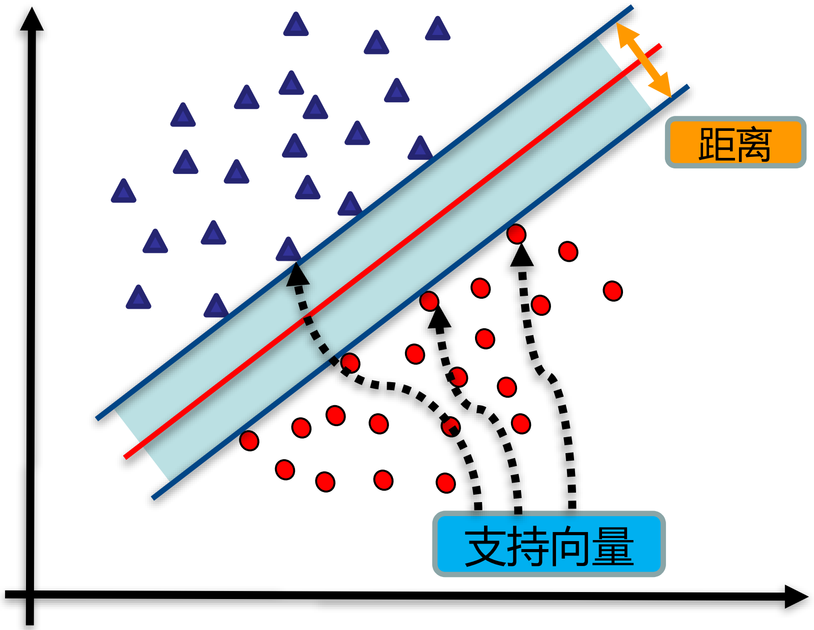
# 1. 支持向量机概述

# 1. 支持向量机概述



**支持向量机（Support Vector Machine, SVM）**是一类按监督学习（supervised learning）方式对数据进行二元分类的广义线性分类器（generalized linear classifier），其决策边界是对学习样本求解的最大边距超平面（maximum-margin hyperplane）。

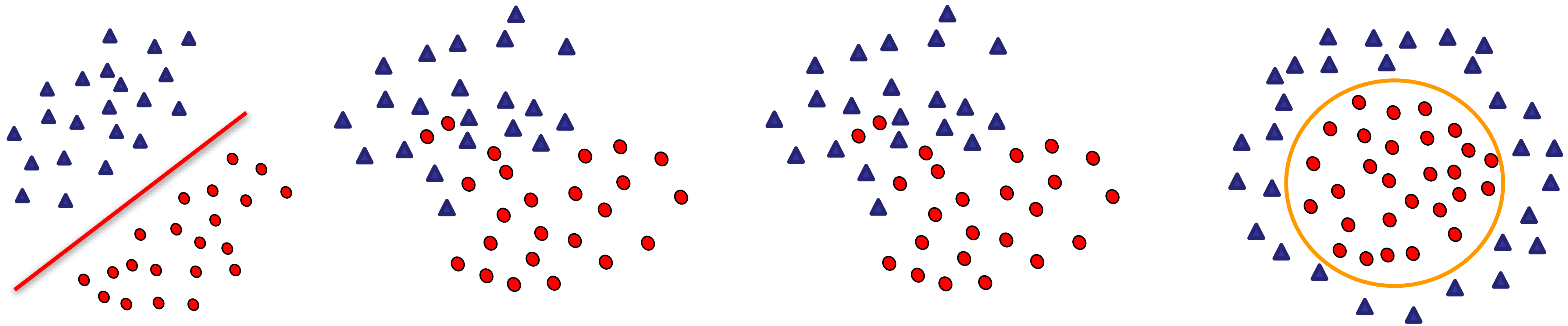
与逻辑回归和神经网络相比，支持向量机，在学习复杂的非线性方程时提供了一种更为清晰，更加强大的方式。



# 1. 支持向量机概述



## 硬间隔、软间隔和非线性 SVM



线性可分　　硬间隔　　软间隔　　线性不可分

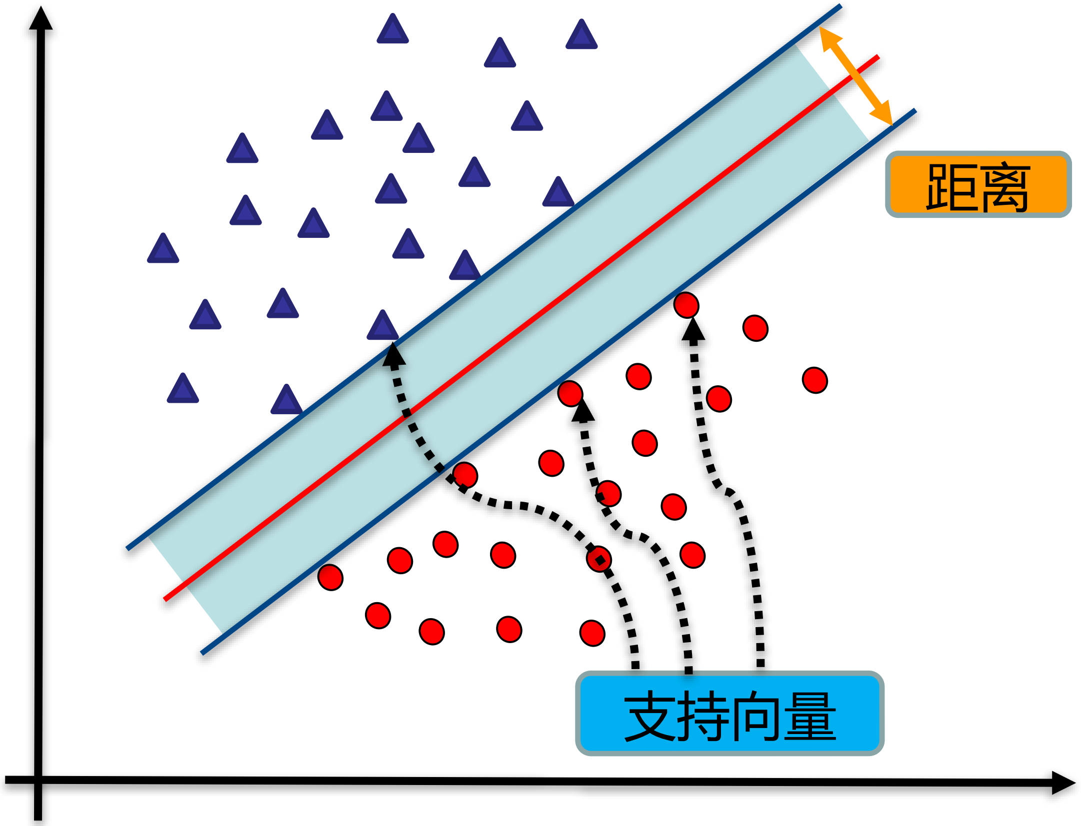
假如数据是完全的线性可分的，那么学习到的模型可以称为硬间隔支持向量机。换个说法，硬间隔指的就是完全分类准确，不能存在分类错误的情况。软间隔，就是允许一定量的样本分类错误。

# 1. 支持向量机概述



**算法思想**

找到集合边缘上的若干数据（称为支持向量（Support Vector）），用这些点找出一个平面（称为决策面），使得支持向量到该平面的距离最大。



# 1. 支持向量机概述



## 背景知识

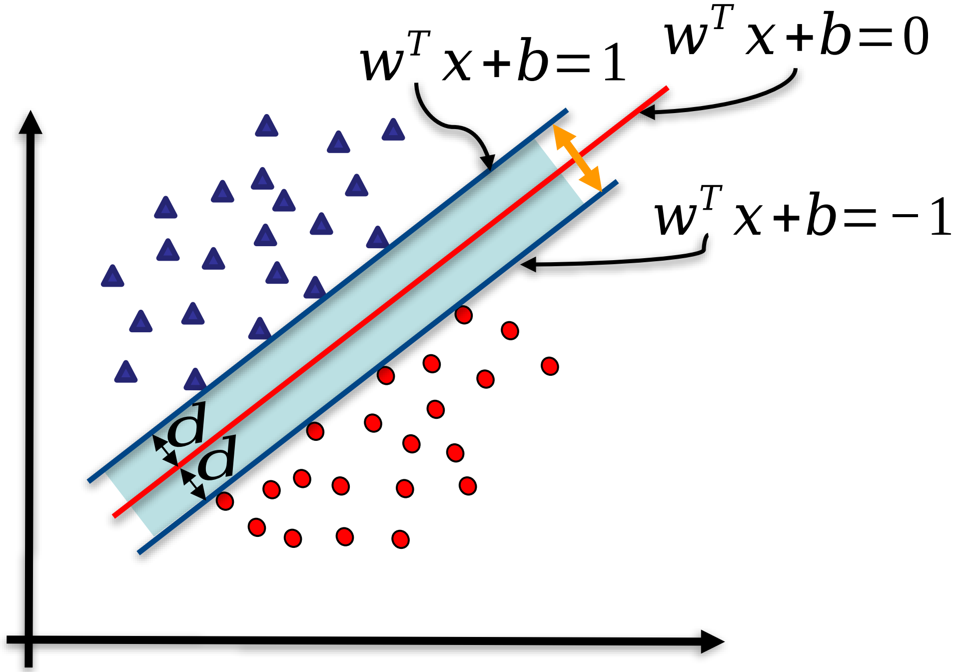
任意超平面可以用下面这个线性方程来描述：

二维空间点 到直线 的距离公式是：

扩展到 维空间后，点 到超平面

的距离为：

其中




如图所示，根据支持向量的定义我们知道，支持向量到超平面的距离为 ，其他点到超平面的距离大于 。每个支持向量到超平面的距离可以写为：

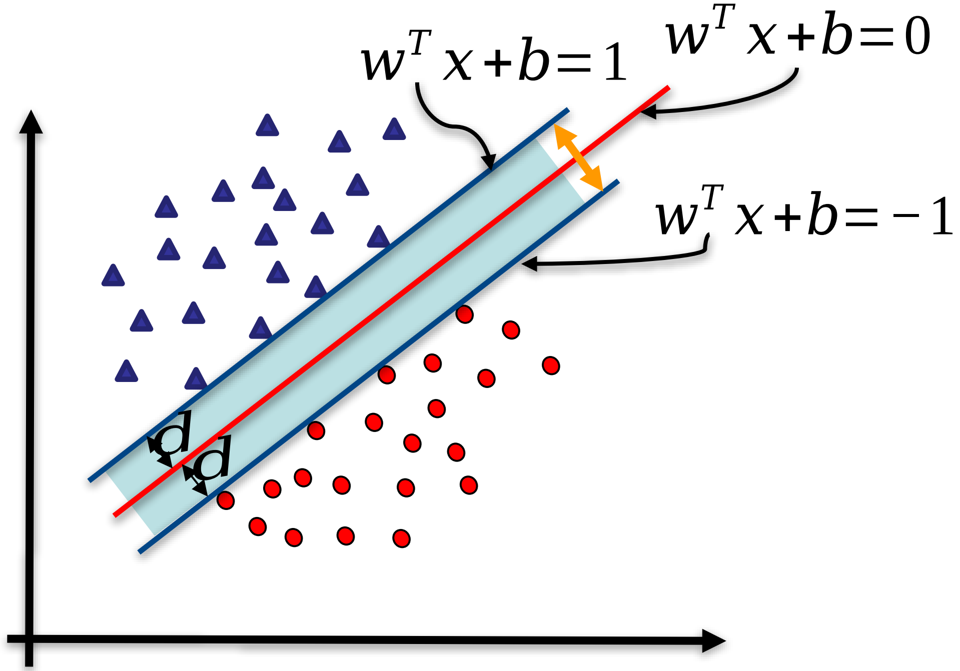
# 1. 支持向量机概述



## 背景知识

如图所示，根据支持向量的定义我们知道，支持向量到超平面的距离为 ，其他点到超平面的距离大于 。

于是我们有这样的一个公式：故：




我们暂且令为 1（之所以令它等于 1，是为了方便推导和优化，且这样做对目标函数的优化没有影响），

将两个方程合并，我们可以简写为：$y(w^T x+b)\geq 1$

至此我们就可以得到最大间隔超平面的上下两个超平面 $d=¿w^T x+b\vee\frac{¿}{¿\vee w\vee¿}¿$

# 2. 线性可分支持向量机

# 2. 线性可分支持向量机



**背景知识**

点到面的距离公式

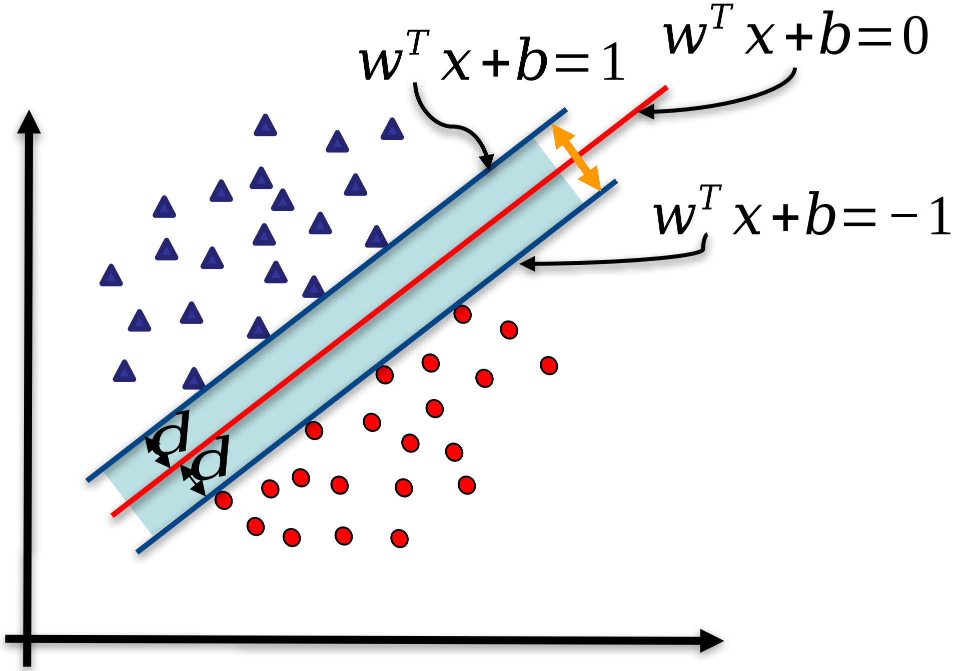
$$d=\frac{|Ax_0+By_0+Cz_0+D|}{\sqrt{A^2+B^2+C^2}}$$

$$y(w^T x+b)\geq 1$$

$$d=¿w^T x+b∨\frac{¿}{¿∨w∨¿}¿$$

$$y(w^T x+b)¿=w¿^T x+b∨¿$$

支持向量机的最终目的是最大化



# 2. 支持向量机求解



**函数间隔**：

**几何间隔**：，当数据被正确分类时，几何间隔就是点到超平面的距离

为了求几何间隔最大，SVM 基本问题可以转化为求解 :( 为几何间隔，为函数间隔 )

# 2. 支持向量机求解



① 转化为凸函数：

先令，方便计算（参照衡量，不影响评价结果）

再将转化成求解凸函数， 1/2 是为了求导之后方便计算。

# 2. 支持向量机求解



② 用拉格朗日乘子法和 KKT 条件求解最优值：

整合成：

其中为拉格朗日乘子

推导：

根据 Karush-Kuhn-Tucker （ KKT ）条件：

# 2. 支持向量机求解



代入

# 2. 支持向量机求解



再把 max 问题转成 min 问题：

添加负号

得到最优解

解出后，代入超平面模型也就是：

，可得，

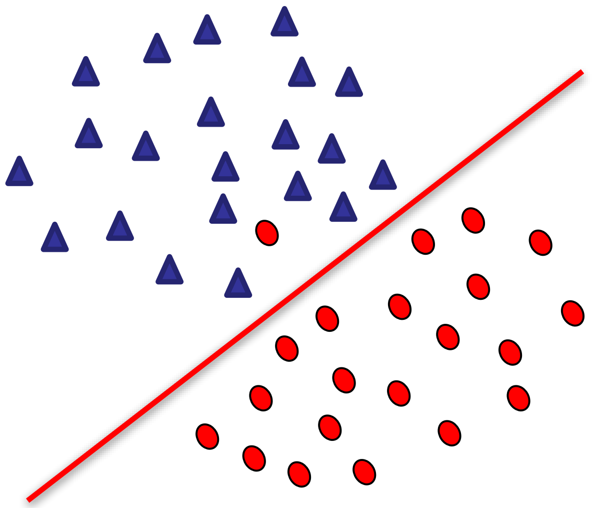
以上为 SVM 对偶问题的对偶形式

# 3. 线性支持向量机

# 3. 线性支持向量机



若数据线性不可分，则可以引入松弛变量，使函数间隔加上松弛变量大于等于1 ，则目标函数：




对偶问题：

$$s.t.\ C \geq \alpha_i \geq 0, i=1,2,\ldots,m \sum_{i=1}^{m} \alpha_i y_i = 0$$

为惩罚参数， 值越大 ，对分类的惩罚越大。跟线性可分求解的思路一致，同样这里先用拉格朗日乘子法得到拉格朗日函数，再求其对偶问题。

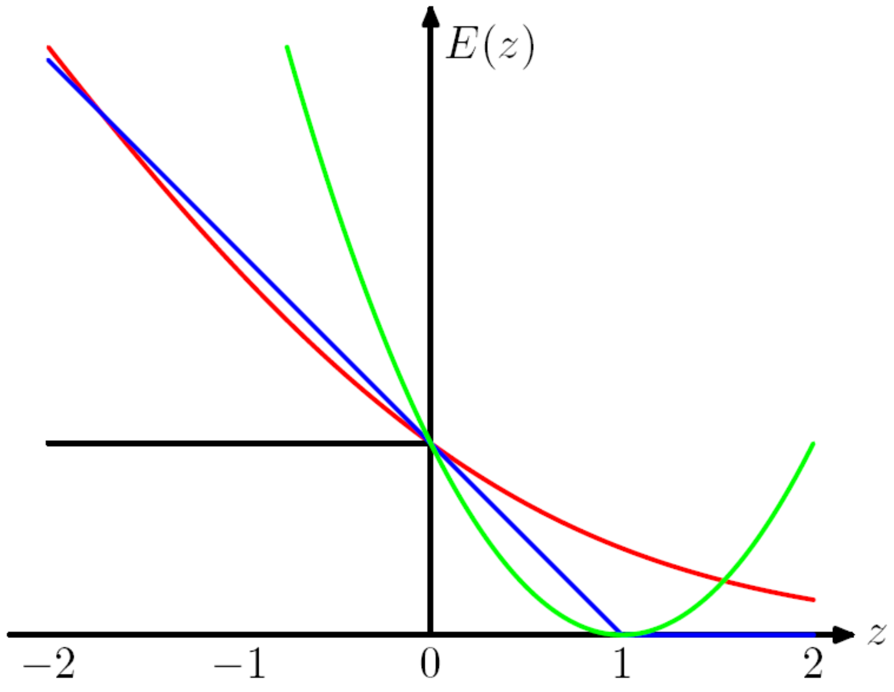
# 3. 线性支持向量机



绿色的线为 square loss
蓝色的线为 hinge loss
红的的线为负 log loss

为 " 松弛变量 "

即 hinge 损失函数。每一个样本都有一个对应的松弛变量，表征该样本不满足约束的程度。




损失函数 hinge loss

# 3. 线性支持向量机



求解原始最优化问题的解和，得到线性支持向量机，其分离超平面为

分类决策函数为：

线性可分支持向量机的解唯一，但不唯一。对偶问题是

# 3. 线性支持向量机



解出后，代入超平面模型：

可得

其中 ：$0<\alpha_i^*<C$

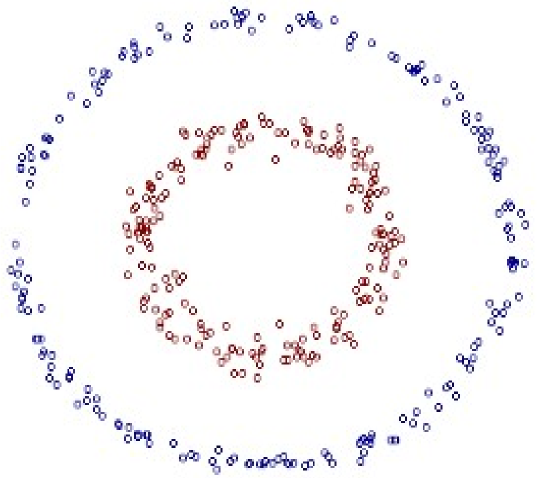
# 4. 线性不可分支持向量机

# 4. 线性不可分支持向量机

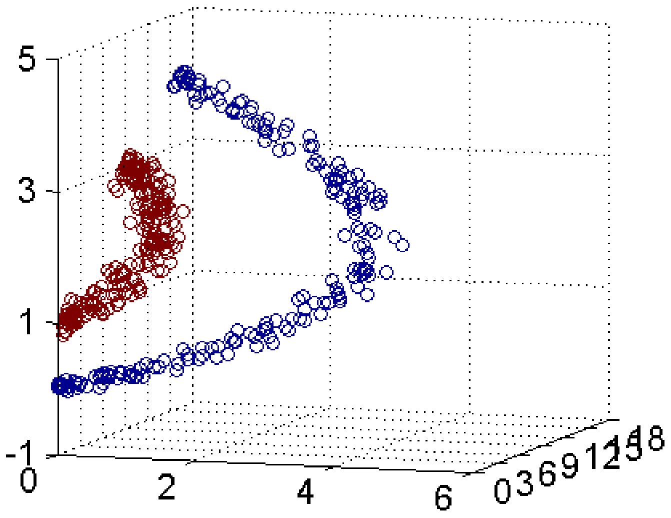


## 核技巧

在低维空间计算获得高维空间的计算结果，满足高维，才能在高维下线性可分。 我们需要引入一个新的概念：**核函数。它可以将样本从原始空间映射到一个更高维的特质空间中，使得样本在新的空间中线性可分**。这样我们就可以使用原来的推导来进行计算，只是所有的推导是在新的空间，而不是在原来的空间中进行，即用核函数来替换当中的内积。



线性不可分





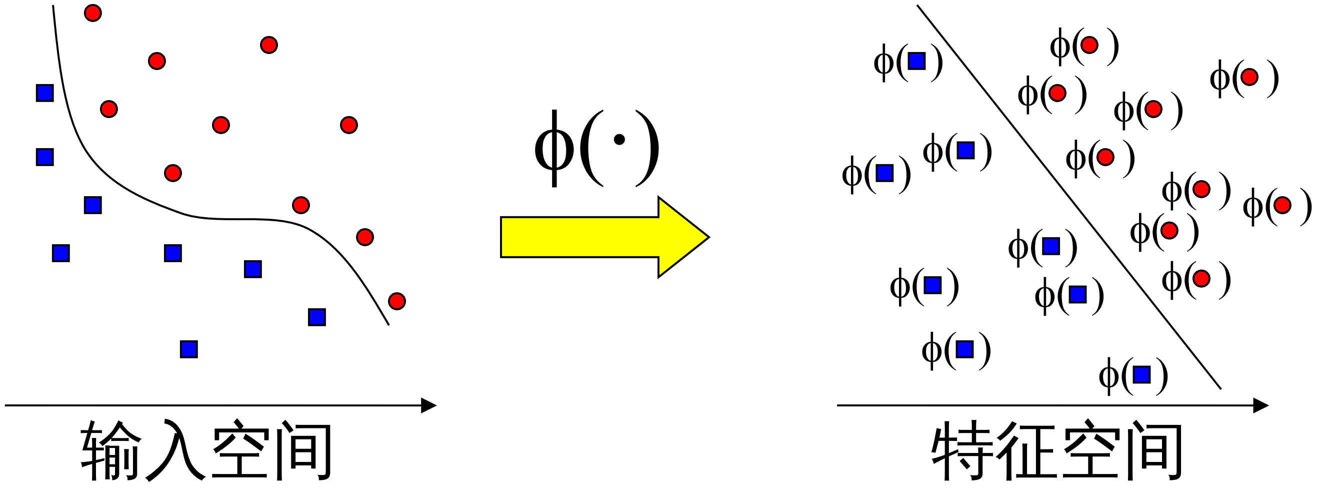
高维下线性可分

# 4. 线性不可分支持向量机



## 核技巧

用核函数来替换原来的内积。




即通过一个非线性转换后的两个样本间的内积。具体地，是一个核函数，或正定核，意味着存在一个从输入空间到特征空间的映射，对于任意空间输入的有：

# 4. 线性不可分支持向量机



在线性支持向量机学习的对偶问题中，用核函数替代内积，求解得到的就是非线性支持向量机

# 4. 线性不可分支持向量机



常用核函数有：

**线性核函数**

**多项式核函数**

**高斯核函数**

这三个常用的核函数中，只有高斯核函数是需要调参的。

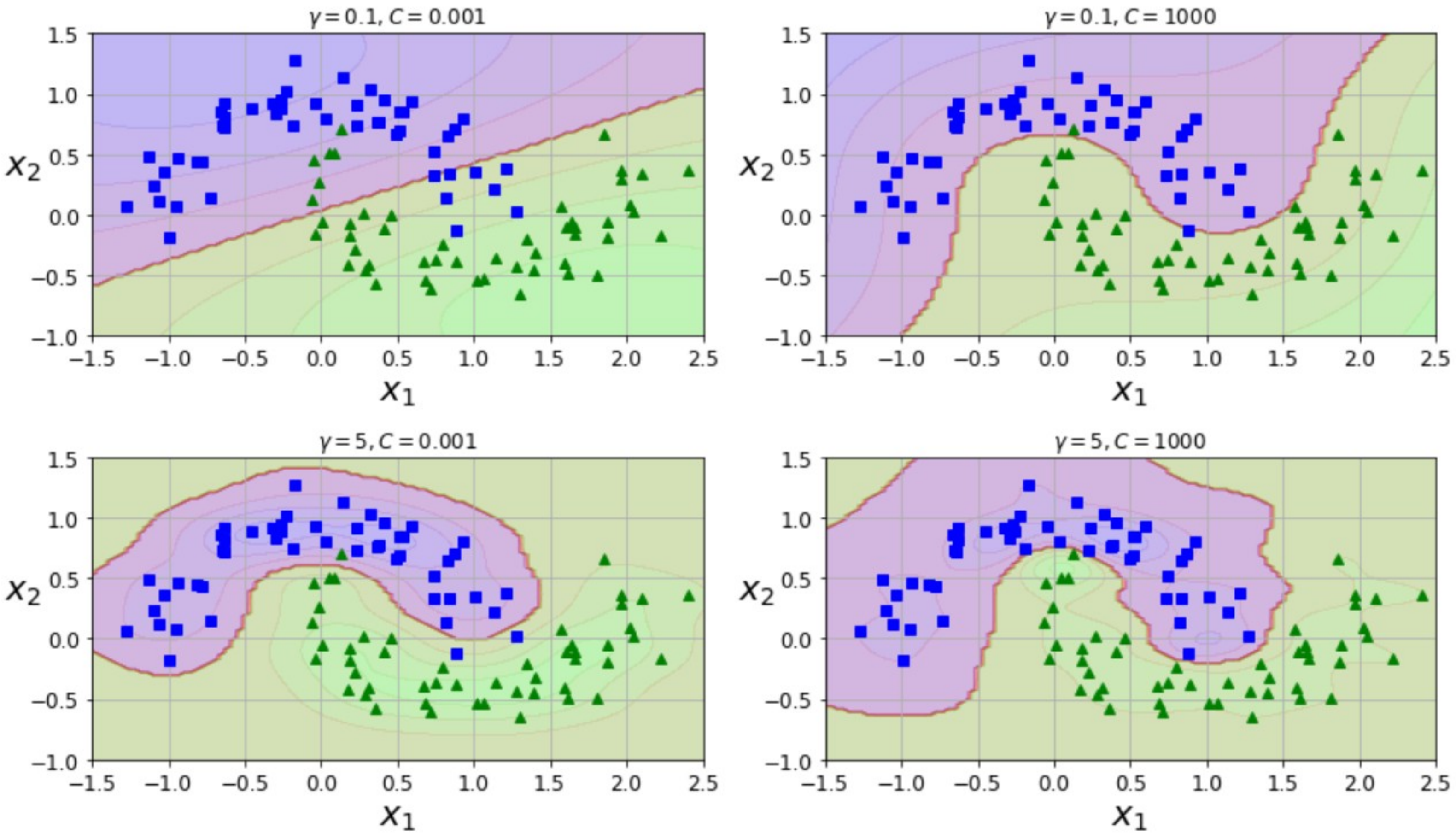
# 4. 线性不可分支持向量机



**SVM 的超参数**

越大，支持向量越少，值越小，支持向量越多。

其中 C 是惩罚系数，即对误差的宽容度。 C 越高，说明越不能容忍出现误差，容易过拟合。 C 越小，容易欠拟合。



# 总结



**下面是一些 SVM 普遍使用的准则：**

为特征数，为训练样本数。

(1) 如果相较于而言，要大许多，即训练集数据量不够支持我们训练一个复杂的非线性模型，我们选用逻辑回归模型或者不带核函数的支持向量机。

(2) 如果较小，而且大小中等，例如在 1-1000 之间，而在 10-10000 之间，使用高斯核函数的支持向量机。

(3) 如果较小，而较大，例如在 1-1000 之间，而大于 50000 ，则使用支持向量机会非常慢，解决方案是创造、增加更多的特征，然后使用逻辑回归或不带核函数的支持向量机。

# 参考文献



[1] CORTES C, VAPNIK V. Support-vector networks[J]. Machine learning, 1995, 20(3): 273–297.

[2] Andrew Ng. Machine Learning[EB/OL]. Stanford University,2014. https://www.coursera.org/course/ml

[3] 李航 . 统计学习方法 [M]. 清华大学出版社 ,2019.

[4] Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning[M]. Springer, New York, NY, 2001.

[5] CHRISTOPHER M. BISHOP. Pattern Recognition and Machine Learning[M]. Springer,2006.

[6] Stephen Boyd, Lieven Vandenberghe, Convex Optimization[M]. Cambridge University Press, 2004.

[7] PLATT J C. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines[J]. 1998: 22.