# 支持向量机

在机器学习领域，支持向量机（**S**upport **V**ector **M**achine）是一种监督学习模型，它提供了一种对训练样本分类（classification）问题和回归（regression）问题的有效算法。20世纪末SVM被 Vapnik 提出，它适用于小样本的快速分类，其性能和人工神经网络（ANN）相当，至今 SVM 被广泛应用于经济学，环境科学，工程学等领域。SVM的发现主要是为了找到划分不同样本的决策边界（decision boundary），类似于ID决策树、神经网络等算法，下面我们给予限制条件从简到繁地介绍SVM算法。

## 线性SVM

### 基本思想

首先，我们来考虑这样一个问题：给定训练样本集 $D=\{(x_i,y_i)\}_{1\leq i\leq m}$ ，其中 $y_i\in\{-1,+1\}$ ，我们需要在训练集 $D$ 所在的样本空间内寻找到一个划分超平面，将样本正确地分类。为了介绍线性SVM，我们需要对样本做一些假定和约束：假定 $x_i$ 是二维向量 $x_i=(x_{i1};x_{i2})$ ,同时样本的分布满足线性可分的性质:在 n 维样本空间中，存在一个 n-1 维超平面能够对样本进行正确的划分，对于我们所讨论的问题，即存在一条直线能够将二维的样本进行正确地分类。

那么对于我们无法用观察直接判断线性可分的样本，该如何确定这些样本的线性可分性呢？数学上，存在相关定理：样本集线性可分的充分必要条件是正实例点集所构成的凸集与负实例点集所构成的凸集互不相交。 证明见附录[1] 。至于那些无法线性可分的样本，我们需要用核函数解决，相关内容我们之在下一节介绍。

对于所讨论的问题，该训练样本集在二维空间内的分布如下图所示.

svm1

[图8.1]为样本空间内二分类样本的分布，存在多个划分超平面将样本正确分类，但是根据泛化性能的比较，红色的划分超平面是最优的。

根据[图8.1]事实上存在多个分类直线作为我们的标准，但是经过观察，显然我们需要找到最中间的那条直线作为划分超平面。因为划分超平面需要对样本做到正确分类的同时也需要对未来的预测样本做到最精确地分类，而因为训练集的局限性或者噪声的因素，训练集外的样本离划分超平面的距离会更小，导致了其他划分超平面的失效。故最中间的分类直线是最优选择。同时最优划分超平面又具有这样一个特性，两类样本中离划分超平面最近的那些点与划分超平面之间的距离是相等的，同时这个距离对于最优划分超平面来说是最宽的。

由此我们可以得出：SVM的核心思想是**尽最大努力使划分超平面与分开的两个类别有最大间隔，这样才使得分隔具有更高的可信度。** 而且对于未知的新样本才有很好的分类预测能力（泛化能力）。

### 模型求解

下面求解优化模型：

写出n 维空间中超平面方程：

$\omega\mathbf{x}+\mathbf{b}=\mathbf{0}$

其中 $\omega = (\omega_1; \ldots; \omega_d)$ 维度与 $x$ 相同，在本例中都是二维向量，同时$b$ 为偏置项，距原点平移一段距离。显然我们需要确定超平面的参数 $(\omega, b)$

然后我们要表示出样本点到超平面的距离并且作为求解目标函数

根据点到超平面的距离公式，任意一个样本点 $x$ 到超平面之间的距离：

$r = \frac{|\omega^{\mathbf{T}}\mathbf{x_i}+\mathbf{b}|}{||\omega||}$

为了求解出约束条件：

对每个 $x_i$ 有约束条件：

若超平面能正确分类，则满足

$\omega^{\mathbf{T}}\mathbf{x_i} + \mathbf{b} \geq +\mathbf{1}$，若 $y_i = +1$

$\omega^{\mathbf{T}}\mathbf{x_i} + \mathbf{b} \leq -\mathbf{1}$，若 $y_i = -1$

这里 $|\omega^{\mathbf{T}}\mathbf{x_i} + \mathbf{b}| \geq \mathbf{1}$ 本质上其实和

$\omega^{\mathbf{T}}\mathbf{x_i} + \mathbf{b} \geq \mathbf{0}$，若 $y_i = +1$

$\omega^{\mathbf{T}}\mathbf{x_i} + \mathbf{b} \leq \mathbf{0}$，若 $y_i = -1$

是等价的，特殊地其中满足 $|\omega^{\mathbf{T}}\mathbf{x_i} + \mathbf{b}| = \mathbf{1}$的最近样本点称为支持向量。因为总是存在一组缩放变换，通过同时放大或缩小参数$(\omega, b)$，使得最靠近超平面的点 $\omega^{\mathbf{T}}\mathbf{x_i} + \mathbf{b} = \mathbf{0}$ 经过变换后满足 $|\omega^{\mathbf{T}}\mathbf{x_i} + \mathbf{b}| = \mathbf{1}$，除非超平面上有样本点，而那样违反了我们关于超平面的假设。

同时对任意 $y_i$ 等价于

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

因为无论对于 $y_i = -1$还是 $+1$ 上式都成立。

我们的目标是写出两类样本距超平面的最近样本点之间的最大间隔：

$d = \frac{2}{||\omega||}$

同时 $max_{\omega,b}\frac{2}{||\omega||}$ 等价于 $min_{\omega,b}\frac{1}{2}||\omega||^2$ 这里做的变换纯粹是为了数学处理的便利，利于之后的拉格朗日乘数法的求导。于是我们得到了SVM的基本模型

$min_{\omega,b}\frac{1}{2}||\omega||^2$

s.t. $y_i(\omega^{\mathbf{T}}\mathbf{x_i} + \mathbf{b}) \geq \mathbf{1}$ 其中 $i = 1, 2, 3...m$

## 对偶问题

求解(1)式有两种常用方式，第一种是利用现成的凸二次规划问题的优化库直接求解，第二种方法是进行转化求解对偶问题。这里我们运用拉格朗日乘数法求解，从而我们得到了对偶问题（dual problem）：

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

其中 $\alpha = (\alpha_1; \alpha_2; \ldots \alpha_m)$，对 $(\omega, b)$求偏导等于0得

$\frac{\partial L}{\partial \omega} = 0 \rightarrow \omega = \sum_{\mathbf{i=1}}^{\mathbf{m}}\alpha_{\mathbf{i}}\mathbf{y_i}\mathbf{x_i}$

$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{m}\alpha_i y_i = 0$

这里涉及到矩阵求导，相关概念见附录。

把（a）带入 $L(\alpha, \omega, \mathbf{b})$中，同时（b）作为约束条件得到对偶问题：

$max_{\alpha} \sum_{i=1}^{m} \alpha_i - \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x_i^T x_j}$

s.t. $\sum_{i=1}^{m} \alpha_i y_i = 0$

$\alpha_i \geq 0$

原问题中存在不等式约束，因此对偶问题应该满足KKT条件（具体参见附录），满足：

$\alpha_i >= 0$

$y_i(\omega^T x_i + b) - 1 >= 0$

$\alpha_i [y_i(\omega^T x_i + b) - 1] = 0$

对任意 $i$ 成立

若 $\alpha_i > 0$，则 $y_i(\omega^T x_i + b) - 1 = 0$ 说明 $x_i$ 是支持向量，位于最大间隔边界；若 $\alpha_i = 0$，则 $y_i(\omega^T x_i + b) - 1 \geq 0$ 同时该样本点对我们习得的模型$f(x) = \sum_{i=1}^{m} \alpha_i y_i x_i^T x + b$是没有贡献和影响的。因此我们可以得到：支持向量机模型参数只和支持向量有关。

那么如何求解对偶问题呢？我们有许多方法，首先这是一个二次规划问题，因此我们能调用现成的库函数进行计算；但是我们有更快速的方法，有许多计算机科学家提出了更高效的算法解决这个问题，下面我们利用著名的SMO算法求解对偶问题。先介绍SMO算法的基本思想：选择 $\alpha_i, \alpha_j$同时固定其他$\alpha$分量的值，然后根据目标函数最大化原则更新 $\alpha_i, \alpha_j$ 的值，不断重复这个过程直到目标函数收敛。

关于如何选取 $\alpha_i, \alpha_j$ 是基于原则：实验证明只要选择的 $\alpha_i, \alpha_j$ 中有一个不满足KKT条件，目标函数值就会在迭代过程中增大。因此第一个变量应该选择与KKT条件相背程度最大的变量。第二个变量应该选择让目标函数增加量最大的变量，因为穷举可能代价很大，SMO选择与第一个变量所对应的样本间隔最大的样本点所对应的变量 $\alpha_j$ 作为第二个更新变量，因为样本距离越大，相似度越小，目标函数值的变化量也会越大。

简单地描述推导过程：

因为 $\sum_{i=1}^{m} \alpha_i y_i = 0$

$\alpha_i y_i + \alpha_j y_j = \lambda$ 其中 $\lambda$ 是个常数，且有 $\alpha_i \geq 0, \alpha_j \geq 0$

代入目标函数得到关于 $\alpha_i$ 的单变量二次规划问题且存在闭式解，因此可以得到一组 $\alpha_i, \alpha_j$.不断迭代后可求出 $\omega$ 的值。

对于任意支持向量 $(x_k, y_k)$ 满足：

$y_k(\sum_{i=1}^{m} \alpha_i y_i x_i^T x_k + b) = 1$

都能求出 $b$ 的值，但是对所有支持向量求出的 $b$ 的平均值更加鲁棒。

# 核函数

我们现在去除训练样本集是线性可分的假设，来重新研究样本集的这样一种分布：

svm2

图为线性不可分的样本在原始空间内的分布情况

显然我们无法在平面内找到一个划分直线，即无法在n维样本空间内找到一个n-1维超平面将二分类样本正确划分，这样的问题就是线性不可分问题。

但是我们并不是真的就束手无策，想到试图找到一个映射 $\Phi$ 将样本从原始空间映射到更高维度的特征空间，保证样本能够在高维空间内是线性可分的。数学上可证明，对于一个有限维空间内的样本，一定能够找到一个高维空间使得样本在该空间内线性可分。

将 $x$ 改写为 $\Phi(x)$ 同时模型也改写为

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

对于新的对偶问题，只需要将 $x$ 改写为 $\Phi(x)$

$$\max_\alpha \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \Phi(x_i)^T \Phi(x_j)$$

约束条件保持不变。

对于如何计算 $\Phi(x_i)^T \Phi(x_j)$ 我们使用的方法是在原始样本空间中找到一个映射$K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$ 来代替直接计算，因为当特征空间的维数很高的时候，计算代价十分巨大，有时甚至是无穷维的，因此显然函数 $K(x_i, x_j)$ 更高效。此处的映射 $K$ 就是**核函数**。

将目标函数改写：

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

原约束条件保持不变。

最终得到的模型改写：

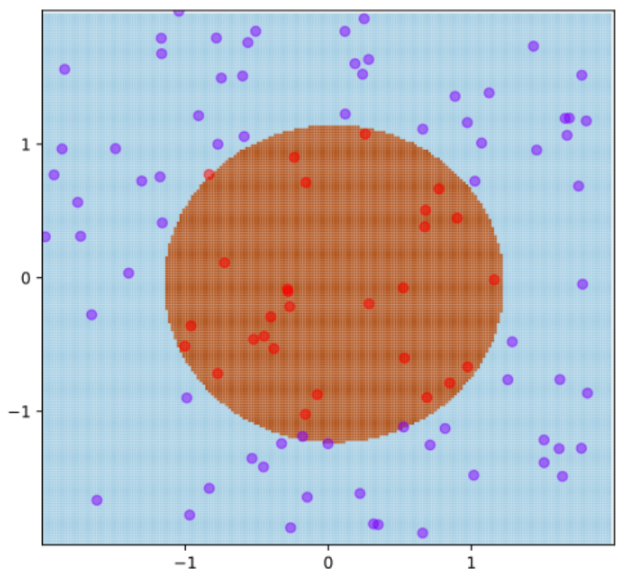
$$f(x) = \sum_{i=1}^m \alpha_i y_i K(x, x_i) + b$$

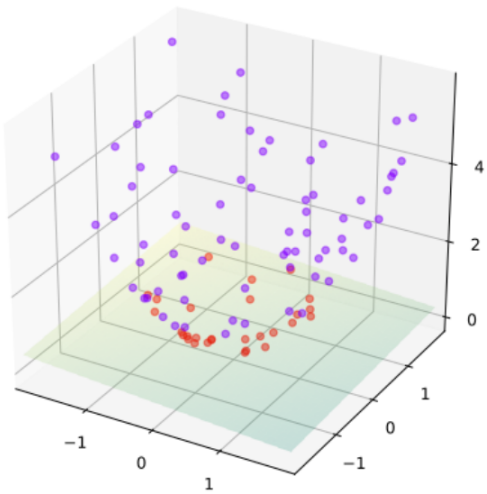
这里给出三种常用的核函数

d多项式： $K(x_i, x_j) = (1 + x_i^T x_j)^d$

径向基：$K(x_i, x_j) = exp(-\gamma ||x_i - x_j||^2)$

神经网络：$K(x_i, x_j) = tanh(\alpha x_i^T x_j + \beta)$



举例：用映射$\Phi:(x_1,x_2)\rightarrow(x_1,x_2,x_1^2+x_2^2)$作为 SVM 从原始空间到高维特征空间的映射

# 决策树模型

决策树是一种常用的回归和分类模型。以分类问题为例，决策树算法是一种基于树形结构的决策过程：基于样本的每一个特征取值对实例进行分类。例如，当我们看到一只“猫”，我们的大脑在识别“这是一只猫还是别的动物？”的过程中会经过这个判断流程（子决策）：如”它有腿吗？“，有腿的话又会判断”它有几只腿？“等等，最后我们会根据我们判断的一些特征最终做出决策：”这是一只猫“。
ID3

例子：上图为简化的决策过程

决策树的优点是：分类速度快。学习时，利用训练数据根据最优的属性划分建立决策树；预测时，利用预测的样本的所有特征进行判断，最终根据叶节点的分类进行划分类别。决策树通常有三个主要操作：选择最优属性划分，生成树，剪枝。现在的决策树模型主要是基于20世纪末提出的ID3，C4.5，CART算法。

## 决策树算法原理和框架

通常的决策树包含一个根节点，若干个内部节点和叶节点。根节点包含了所有样本集；内部节点对应了一个属性的测试条件；叶节点对应了样本的最终分类，即决策结果。从根节点到任意一个叶节点的通路代表了一个判断序列，根据从而根据这个判断序列能够决策出样本在该判断序列下的分类。决策树模型的最终目的是生成一个对未知样本的决策能力强，泛化能力强的决策树。

下面我们给出生成一颗决策树的基本流程：

约定 训练样本集 $D=\{(x_i,y_i)\}_{0\leq i\leq m}$特征属性集 $A=\{a_i\}_{0\leq i\leq m}$

```
BuildTree(D,A)
begin
1 node= new treenode();
/* 情况1 */
2 if (D中样本类别c相同) then
3   node 被标记为c类的叶节点;
4 end if
/* 情况2 */
5 if (A为空 or D在A上取值相同) then
6   node 被标记为c类的叶节点 其中c为D中类别众数;
7 end if
/*
按照不同的标准找到最优划分属性a
*/
8 计算最优划分特征 a;
9 for (a的一个取值ai) do
10  node 生成子节点 childnode;
11  Di表示D在属性a中取值为ai的样本集;
     /* 情况3 */
12  if (Di为空) then
13    childnode 被标记为c类的叶节点 其中c为D中类别众数;
14    return
```

```
  else
    childnode=BuildTree(Di,A-a)中treenode;
  end if
 end for
end
```

我们可以看出整个决策树是递归结构建立的，通过反复找到最优划分属性$a$来标记当前节点所属的类别$c$.

这里说明一下三个递归终止条件：情况1：样本是同一种类$c$，就把当前节点标记为这个类$c$。情况2：没有属性可以选择 或 属性的取值对分类无影响，就标记为所有类别中出现频数最高的类$c$，表示无法划分。情况3：当前节点包含的属性$a$上取值为$a_i$样本集合为空，表示不能划分。

# 最优划分属性

最优划分属性是建立决策树的关键，同时也关乎决策树对未来数据分类预测的泛化能力的强弱。我们希望随着划分次数的增加，当前节点所含样本类别的纯度获得显著提升，从而达到对样本集的分类能力。

## 信息熵和信息增益

信息熵（information entropy）是用来度量集合纯度的指标，信息熵越大，纯度越低。对于当前的样本集合$D$，信息熵可以表示为

$$E(D)=-\sum_{k=1}^{K}\alpha_k log_2 \alpha_k$$

其中 $K$ 表示样本集合 $D$ 的类别总数，$\alpha_k$ 为类别为 $k$ 的样本数占样本总数 $|D|$ 的比例。同时我们有 $0 \leq E(D) \leq log_2 K$

信息增益（information gain）是用来度量集合 $D$ 以属性 $a$ 来划分后的信息熵的增量，即划分后纯度的提升程度的指标。对于划分标准属性 $a=\{a_i\}_{1\leq i\leq N}$ ，在 $a$ 上取值为 $a_i$ 的样本集合为 $D_i$ ，则样本集合 $D$ 以 $a$ 来划分后的信息增益为

$$G(D,a)=E(D)-\sum_{i=1}^{N}\frac{|D_i|}{|D|}E(D_i)$$

其中 $\frac{|D_i|}{|D|}$ 为取值为 $a_i$ 的样本数所占样本总数的比例。可以看到信息增益越大则划分后的样本纯度提升就越大，因此找到最优划分属性的过程，就是计算出 $A$ 中最大信息增益对应的属性 $a$ 的过程，就是 $a^*=arg_{a\in A}maxGain(D,a)$ ，其中 ID3 决策树就是用信息增益来选择最优划分属性的。