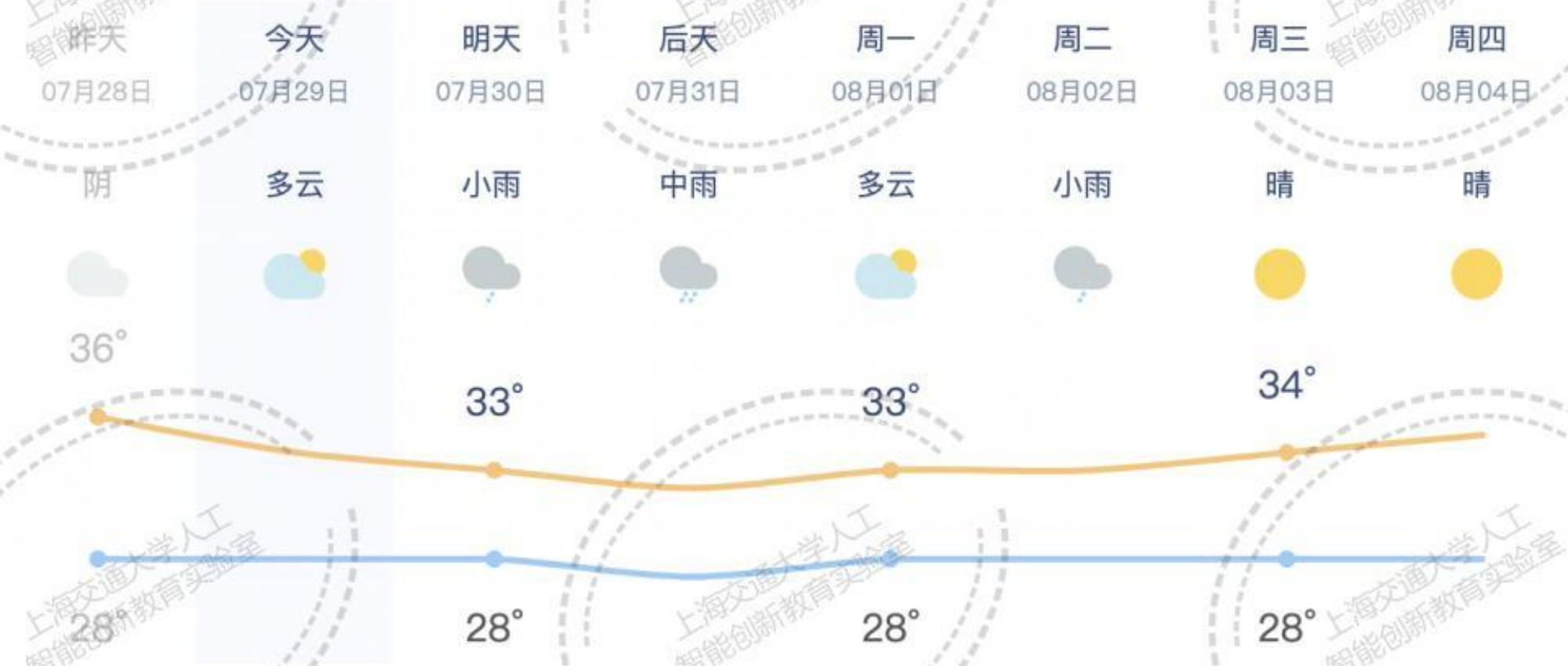
人工智能基础与进阶

线性回归与分类

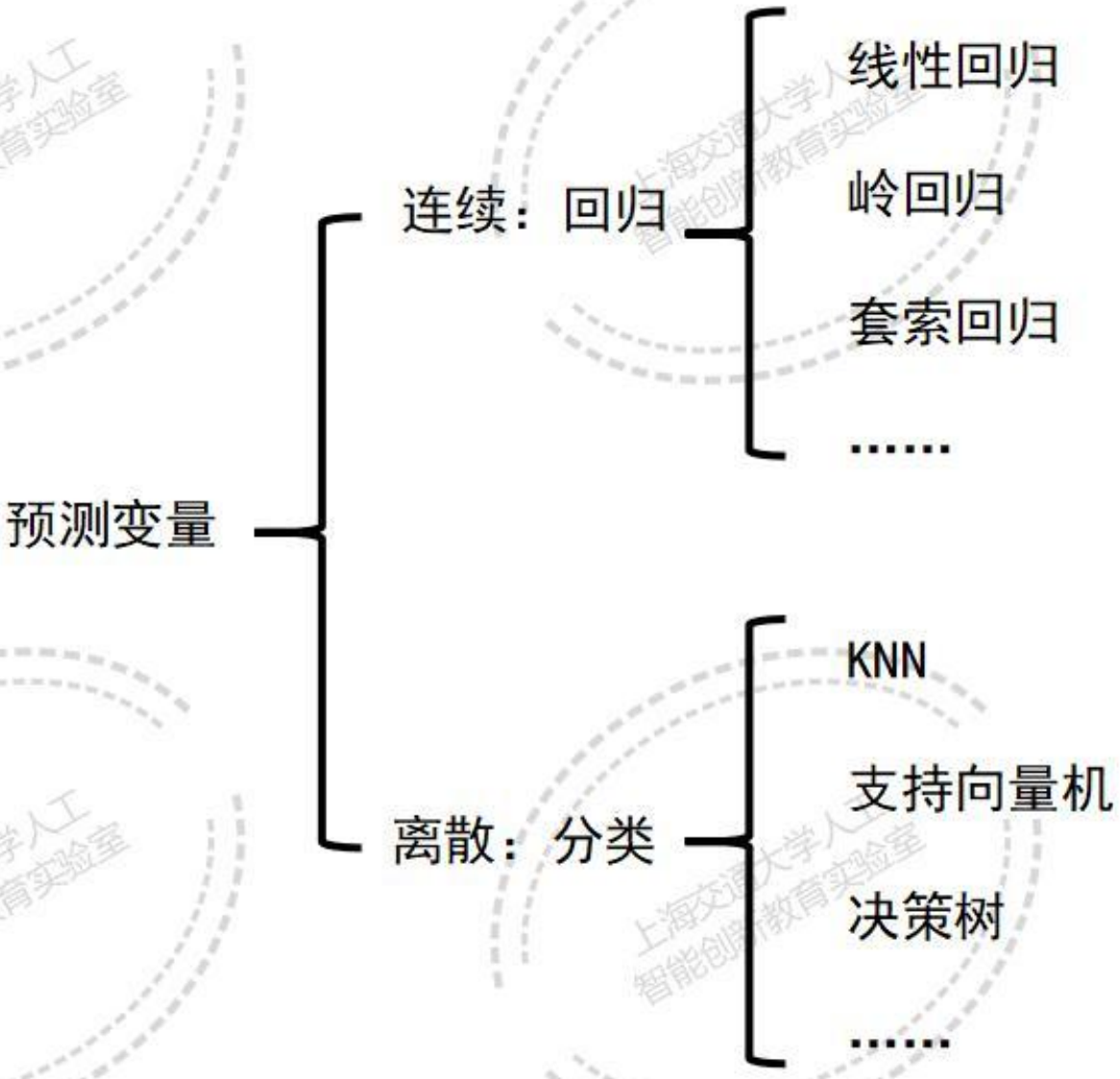
上海交通大学

回归与分类

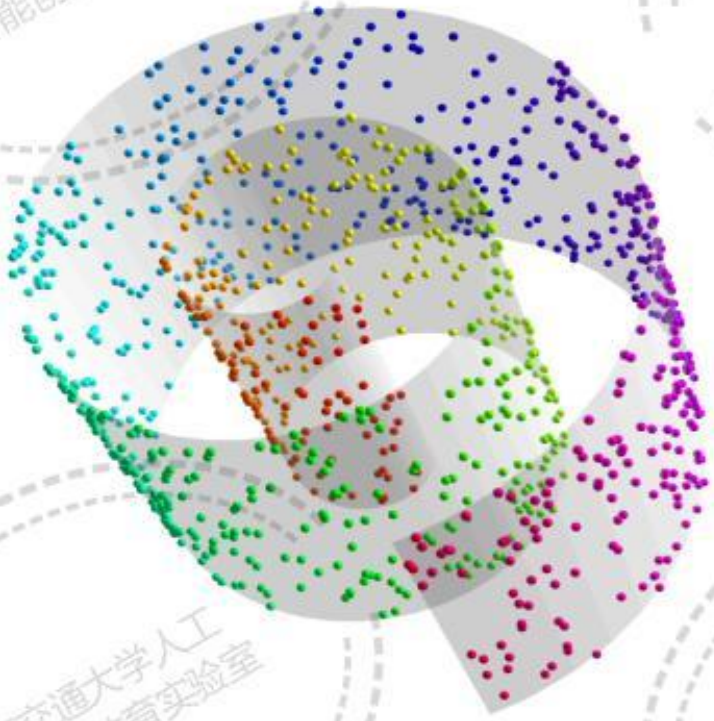
预测气温 vs 预测晴雨



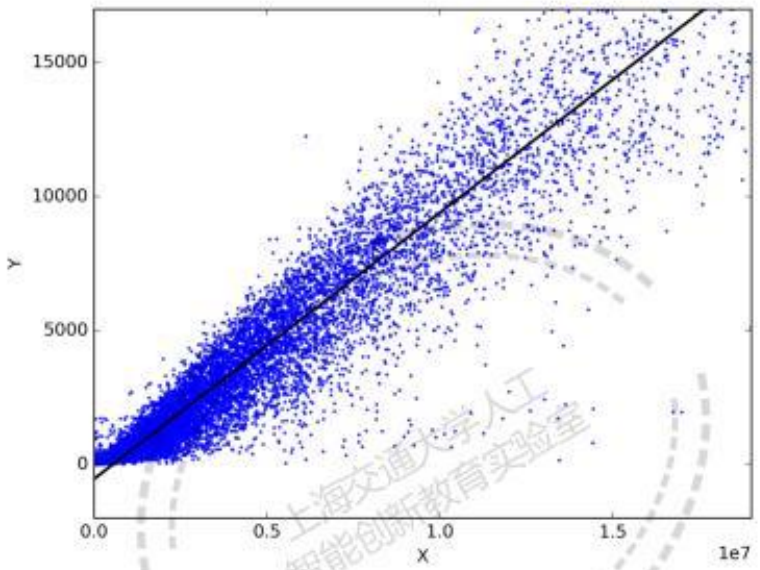
回归与分类



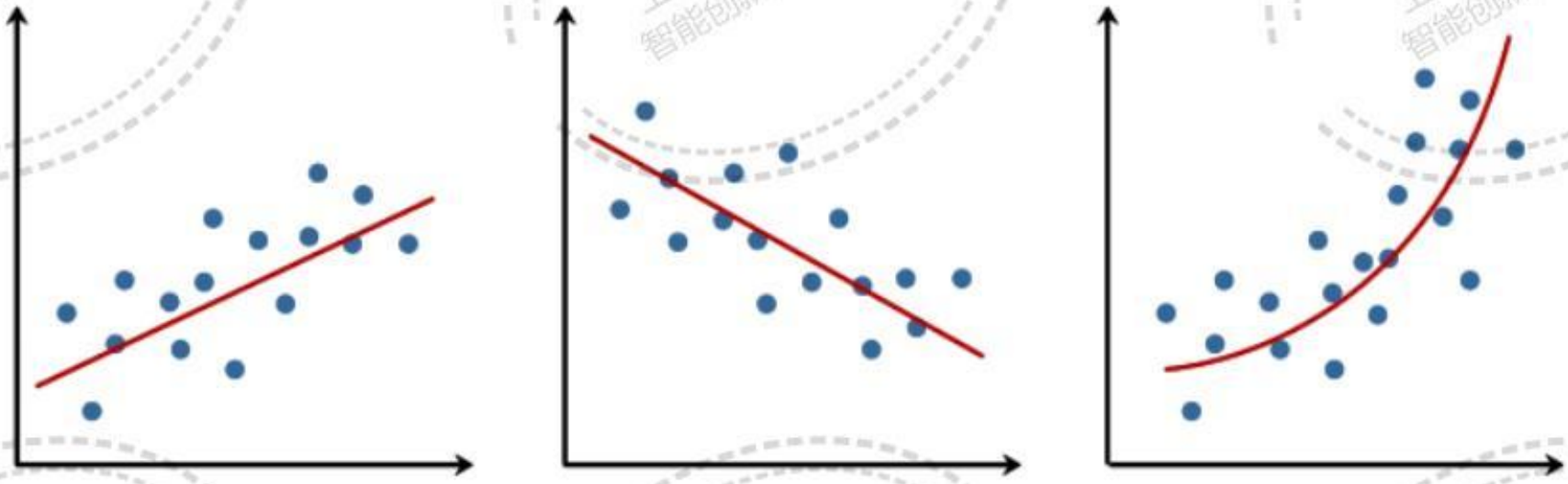
目录 content



第一节 回归



线性回归—采样与回归

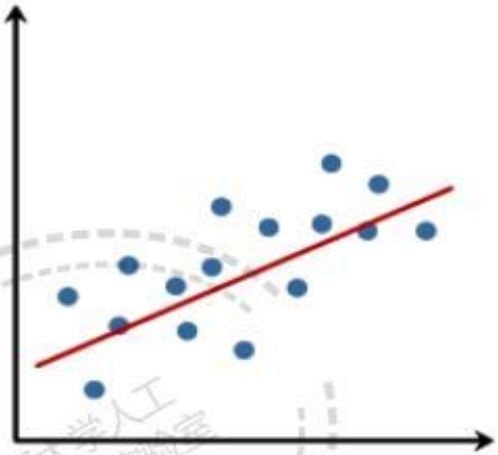


采样点和理想的回归曲线

线性回归—线性回归求解

$f(x)=wx+b$　　一元线性回归表达式

$f(x)=w(1)x(1)+w(2)x(2)+w(3)x(3)+b$　　多元线性回归表达式



线性回归

1. 构建合理的表达式 $f(x)$；

2. 构建合理的准则以确定表达式 $f(x)$ 中的参数的值；

3. 对模型进行求解，最终建立 $f(x)$。

线性回归—线性回归求解

x与y的线性关系：

$$f(x) = wx + b$$ 一元线性表达式

$$f(x) = w(1)x(1) + w(2)x(2)... + w(j)x(j) + b$$ 多元线性表达式

用 $W^T X$ 表示向量间乘法，即 $W^T X = \sum_{j=1}^{n} w(j)x(j)$

b：偏置量（bias）

w：权值，维数与x相同的向量

$$f(x) = \sum_{j=1}^{n} w(j)x(j) + b = W^T X + b$$

构建了x和y之间关系的形式之后，需要确定关系式里的参数。

线性回归—线性回归求解

例如，采样结果如下表所示，假设它们符合一元线性关系 $y = wx$。

	第1组	第2组	第3组	第4组	第5组	第6组
x_i	1	3	3.4	4.1	4.9	5.2
y_i	3.1	9.2	10.1	11.8	14.3	15.8

如果需要w对所有的采样都满足$y_i = wx_i$，那么需要满足 6 个方程:

线性回归—线性回归求解

无法使得w对所有的采样都满足 $y_i = wx_i$，因此尽可能减小误差平方和

	第1组	第2组	第3组	第4组	第5组	第6组
x_i	1	3	3.4	4.1	4.9	5.2
y_i	3.1	9.2	10.1	11.8	14.3	15.8

$$y = 2.971x$$

$$f(x) = ax + b \Rightarrow df/dx = \frac{df}{dx} + \frac{df}{dx} = a + 0$$

$$f(x) = ax^2 + bx + c \Rightarrow df/dx = \frac{df}{dx^2}\frac{dx^2}{dx} + \frac{df}{dx} + \frac{df}{dx} = a \cdot 2x + b + 0$$

$$f(w) = (3.1 - w)^2 \Rightarrow df/dw = \frac{df}{d(3.1-w)}\frac{d(3.1-w)}{dw} = 2(3.1-w)\cdot(-1)$$

其中a，b，c为常数

通过求导可以找到函数的**极小值点**（导数为0的点）

$3.1 = w \times 1$

$9.2 = w \times 3$

$10.1 = w \times 3.4$

$11.8 = w \times 4.1$

$14.3 = w \times 4.9$

$15.8 = w \times 5.2$

$$\min_w \{ (3.1 - w)^2 + (9.2 - 3w)^2 + \ldots + (15.8 - 5.2w)^2 \}$$

求导

$$-2(3.1 - w) - 6(9.2 - 3w) - \ldots - 10.4(15.8 - 5.2w) = 0$$

解得 $w = 2.971$

线性回归—线性回归求解

如果存在偏置b，需要计算偏导数 $y = wx + b$

	第1组	第2组	第3组	第4组	第5组	第6组
x_i	1	3	3.4	4.1	4.9	5.2
y_i	3.1	9.2	10.1	11.8	14.3	15.8

$$\min_{w,b} \quad (3.1-(w+b))^2+(9.2-(3w+b))^2+\ldots+(15.8-(5.2w+b))^2$$

求偏导

$$-2\,(3.1-(w+b))-6(9.2-(3w+b))-\ldots-10.4(15.8-(5.2w+b))=0$$

$$-2\,(3.1-(w+b))-2(9.2-(3w+b))-\ldots-2(15.8-(5.2w+b))=0$$

$$w = 2.931$$
$$b = 0.167$$

$$\Rightarrow \quad y = 2.931x + 0.167$$

线性回归—高维线性回归

在n维空间我们希望利用m个采样$\{x_i, y_i\}_{i=1}^m$来建立x和y之间的关系。

以房价模型为例，假设影响房屋价格y的因素有：

- 房屋面积
- 房间数量
- 楼层数
- 房屋年龄

我们可以得到如下所示的多元线性回归表达式：

$$f(x) = w(1)x(1) + \ldots + w(4)x(4) + b$$

写成向量表达式为：

$$f(x) = W^T X + b$$

线性回归—高维线性回归

由于噪声等干扰因素的存在，观测到的采样 y_i 是包含噪声的，即如果真实的函数关系为 $f(x) = \bar{w}^T x + \bar{b}$，那么采样值为

$$y_i = \bar{w}^T x_i + \bar{b} + \varepsilon$$

这里ε为随机噪声，可以假设其服从均值为0，方差为 σ^2 的正态分布：

$$\mathbf{Prob}(\varepsilon) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{\varepsilon^2}{2\sigma^2}\right)$$

这样的噪声 ε 被称为**高斯噪声**。

y_i 和 $w^T x_i + \mathbf{b}$ 之间存在的偏差 $y_i - (w^T x_i + \mathbf{b})$ 被称为残差。我们寻找的“最优”参数应当是使得总的误差值“最小”。

最小二乘

为使得误差最小，就需要对误差进行衡量：如利用残差的平方和，即 $\sum_{i=1}^{m}(y_i-(\boldsymbol{w}^T\boldsymbol{x}_i+\boldsymbol{b}))^2$ 或者“均方误差”（即在前式基础上再除以总的样本数）对拟合的效果进行衡量。

$$\min_{\boldsymbol{w},\boldsymbol{b}} \quad \sum_{i=1}^{m}(y_i-(\boldsymbol{w}^T\boldsymbol{x}_i+\boldsymbol{b}))^2$$

其含义是寻找出一对 $\boldsymbol{w}$、$\boldsymbol{b}$ 使得误差平方和最小（注意，对于优化问题的目标函数，乘以一个正的常数并不影响求解的结果，因此，“误差平方和”或“均方误差”对应的目标函数没有区别）。因为是通过极小化误差平方和来求解回归问题，因此被称为“**最小二乘**方法”（**least squares**）。

最小二乘

为求解最小二乘问题，需要将目标函数对参数分别进行求导，得到**最优解**需要满足的条件，即

$$\frac{\partial \sum_{i=1}^{m}(y_i-(\boldsymbol{w}^T\boldsymbol{x}_i+b))^2}{\partial \boldsymbol{w}}=2\sum_{i=1}^{m}(y_i-(\boldsymbol{w}^T\boldsymbol{x}_i+b))\boldsymbol{x}_i=0$$

$$\frac{\partial \sum_{i=1}^{m}(y_i-(\boldsymbol{w}^T\boldsymbol{x}_i+b))^2}{\partial b}=2\sum_{i=1}^{m}(y_i-(\boldsymbol{w}^T\boldsymbol{x}_i+b))=0$$

由此得到了包含 $m+1$ 个未知数的 $m+1$ 个线性方程，通过求解这个**线性方程组**（又被称为“**正则方程**”normal equation），就可以得到最小二乘问题的解。

设正则方程的解是 $\boldsymbol{w}^*$，b^*，则它们是问题的最优解，即 $f(\boldsymbol{x})=\boldsymbol{w}^{*T}\boldsymbol{x}+b^*$ 所给出的回归直线，在所有直线中具有最小的误差平方和。

最小二乘

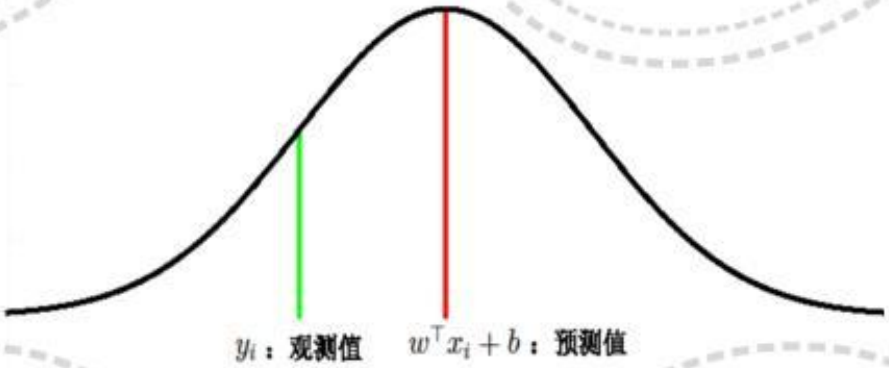
对于给定的参数 w 和 b，输入 x_i，观测到 y_i 的条件概率为：

$$Prob(y_i|x_i, w, b)$$

$$= Prob(w^T x_i + b + \varepsilon | x_i, w, b)$$

$$= \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(y_i - (w^T x_i + b))^2}{2\sigma^2}\right)$$

同时观测到 m 个采样 $\{x_i, y_i\}_{i=1}^m$ 的概率为 $\prod_{i=1}^m \mathbf{Prob}(y_i|x_i, w, b)$。为方便讨论，对其取对数，将乘法转变为加法可以得到：



给定参数和输入时，观测到 y_i 的条件概率

$$log\left(\prod_{i=1}^m \mathbf{Prob}(y_i|x_i, w, b)\right)$$

$$= \sum_{i=1}^m log \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(y_i - (w^T x_i + b))^2}{2\sigma^2}\right) - l\left(\prod_{i=1}^m \mathbf{rob}(y_i|x_i, w, b)\right)$$

$$= -\left(\sum_{i=1}^m log\sqrt{2\pi} + log\,\sigma + \frac{(y_i - (w^T x_i + b))^2}{2\sigma^2}\right)$$

最小二乘—非线性问题求解

最小二乘方法是用于求解**线性回归问题**的，能够通过优化得到线性参数 $\boldsymbol{w}$ 和 $\boldsymbol{b}$。

如果选用非线性基函数 $\boldsymbol{\phi(x)}: \boldsymbol{R^n \mapsto R^d}$，那么，最小二乘方法也可以用以构建非线性的回归函数： $\boldsymbol{y_i = w^T \phi(x_i) + b + \varepsilon}$，相应的回归问题可以写为：

$$\min_{w,b} \sum_{i=1}^{m} \left(y_i - \left(w^T \phi(x_i) + b\right)\right)^2$$

常用的基函数包括**线性函数**、**多项式函数**、**高斯核函数**等。例如，2阶多项式基函数包含：

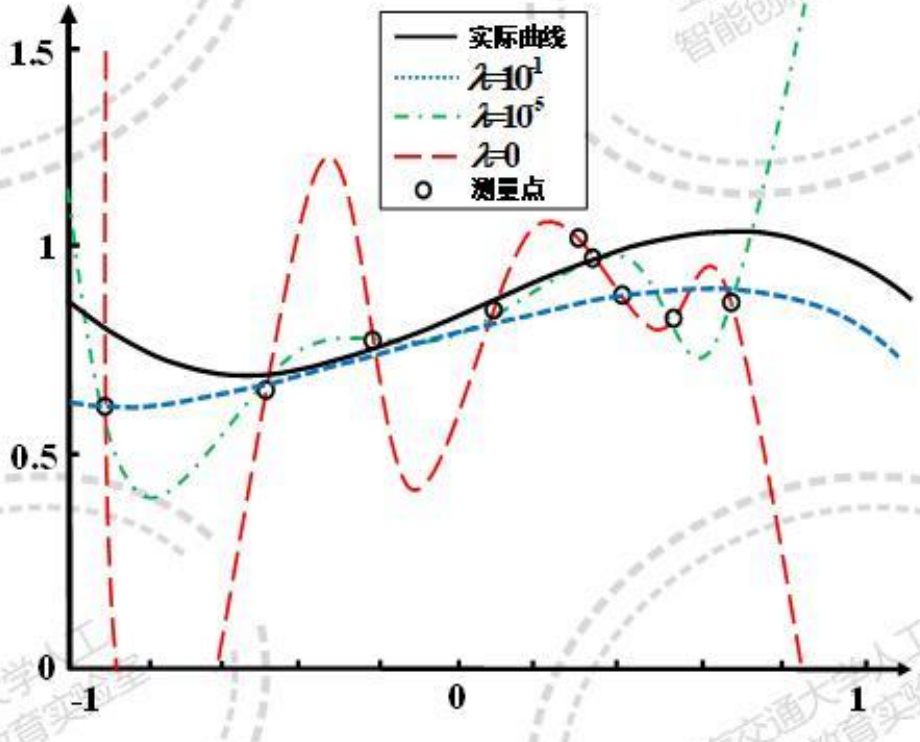
- 所有分量：$x(1), x(2), \ldots, x(n)$
- 各分量的平方项 $x(1)^2, x(2)^2, \ldots, x(n)^2$
- 各分量的二阶交叉项 $x(1)x(2), x(1)x(3), \ldots, x(n-1)x(n)$。

最小二乘—非线性问题求解

过拟合现象：虽然利用最小二乘得到的多项式在每一个采样点上都拟合得很好，但在非采样点上的准确度却不高。

在采样有噪声的情况下，过拟合带来的危害更大。

过拟合的实质是采样误差和扩展误差之间的不一致性。一般来说，使用的模型越复杂，两者的不一致性越大。



使用5阶多项式基函数的学习结果

岭回归

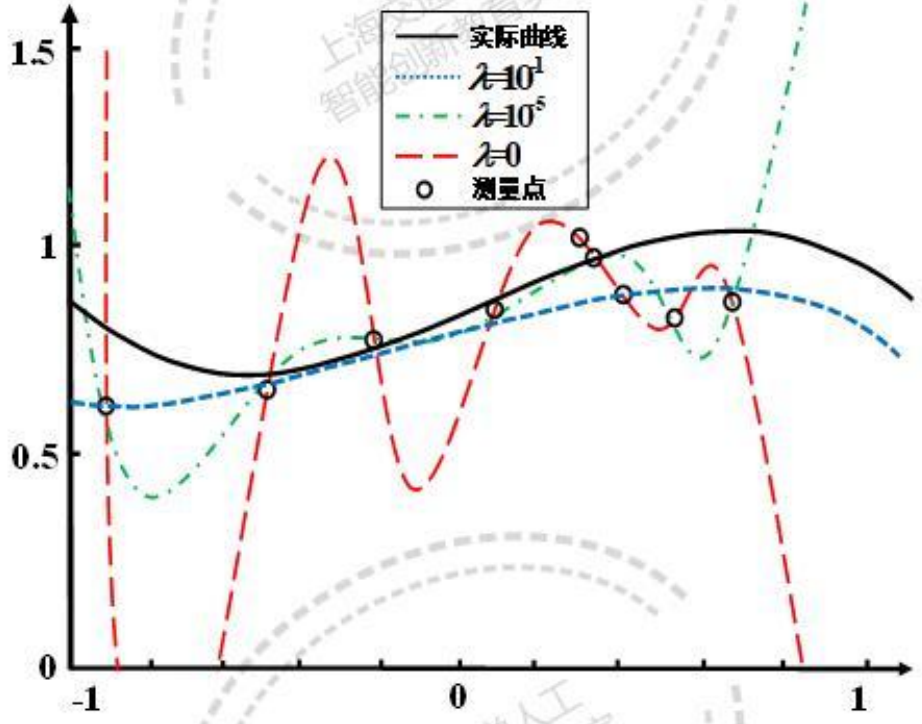
在 $\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i)+\boldsymbol{b}$ 中，$\boldsymbol{w}$ 越大，函数的变化范围和剧烈程度就越大。可以用 $\|\boldsymbol{w}\|_2^2=\sum\left(\boldsymbol{w}(\boldsymbol{j})\right)^2$ 衡量模型的复杂程度，进而将最小二乘法修改为：

$$\min_{\boldsymbol{w},\boldsymbol{b}} \quad \boldsymbol{\lambda}\|\boldsymbol{w}\|_2^2+\sum_{\boldsymbol{i}=1}^{\boldsymbol{m}}\left(\boldsymbol{y}_i-\left(\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}_i)+\boldsymbol{b}\right)\right)^2$$

其中，$\|\boldsymbol{w}\|_2^2$ 被称为“**正则化项**”（regularization term）用以调节模型的复杂性；$\boldsymbol{\lambda}$ 为正则化参数，用以在模型准确度和复杂性之间取得平衡。

修改后的最小二乘方法又被称为“**岭回归**”（ridge regression）。

岭回归



$$\min_{w,b} \quad \lambda\|w\|_2^2 + \sum_{i=1}^{m} \left(y_i - (w^T\phi(x_i) + b)\right)^2$$

合理的正则化参数可以更好地帮助建模

套索回归

“**套索回归**”（LASSO regression）与岭回归类似，不同之处在于岭回归使用正则化项 $\|w\|_2^2 = \sum(w(j))^2$ 来衡量模型的复杂程度，而**套索回归**则使用 $\|w\|_1 = \sum|w(j)|$ 来调节模型的复杂性。

套索回归也有一个正则化参数λ，用来调节模型的准确度和复杂性。当λ=0时，由下式可知，套索回归又变为原始的最小二乘。

$$\min_{w,b} \quad \lambda\|w\|_1 + \sum_{i=1}^{m}(y_i - (w^T\phi(x_i) + b))^2$$

二分类和多分类

分类

?

是不是猫？→是（y=1）/否（y=-1）

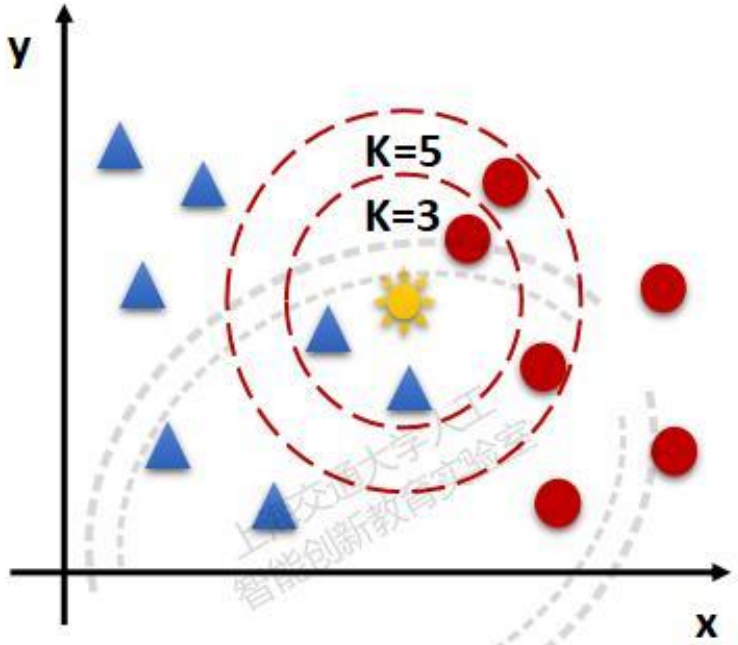
二分类

多个二分类器

是猫（y=1）、狗（y=2）还是兔子（y=3）？

多分类

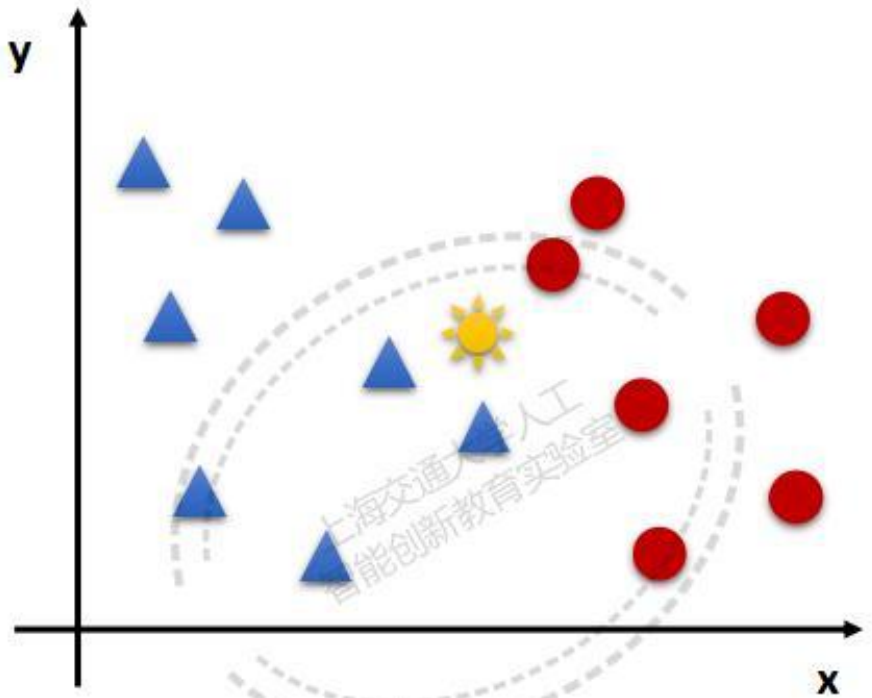
第二节 KNN



KNN

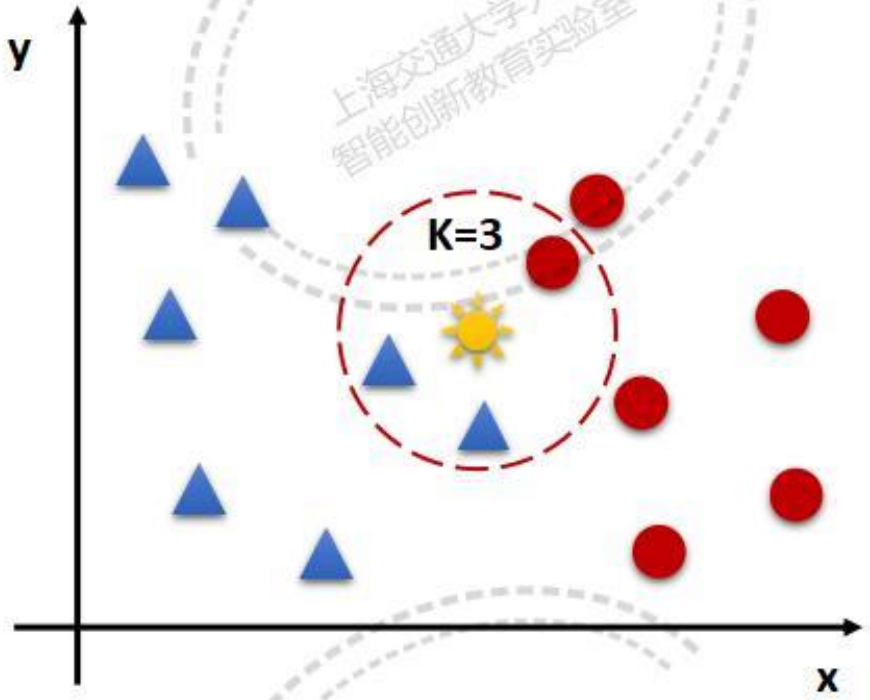
K最邻近（KNN，K-NearestNeighbor）分类算法是数据挖掘分类技术中最简单的方法之一。

KNN中的K，指的是K最近邻，也就是K个最近的邻居的意思，即每个样本都可以用它最接近的K个邻近值来代表。



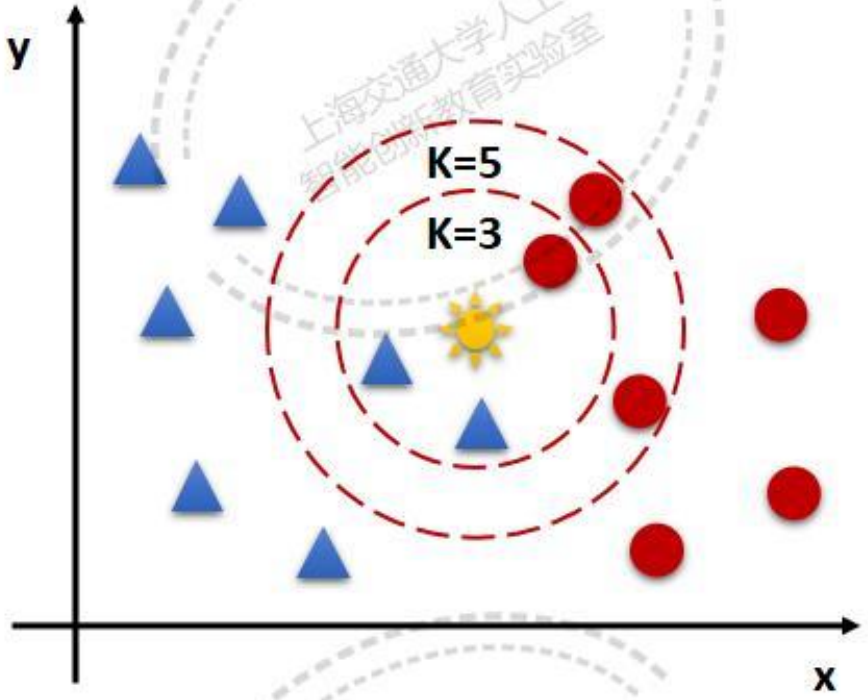
如何确定新样本点属于哪一类（▲或者●）？

KNN



K=3: 2 1

KNN



K=3: 2 1

K=5: 2 3

KNN-距离计算

度量空间中点距离，有好几种度量方式，比如常见的曼哈顿距离，欧式距离等。

KNN算法常用的是**欧式距离**，在二维平面中即两点之间的直线距离，计算公式如下：

$$\rho = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

若要拓展到多维空间，则公式变为：

$$d(x,y) := \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

KNN-算法步骤

总体来说，KNN分类算法包括以下4个步骤：

- 准备数据，对数据进行预处理 。
- 计算测试样本点（也就是待分类点）到其他每个样本点的距离。
- 对每个距离进行排序，然后选择出距离最小的K个点 。
- 对K个点所属的类别进行比较，根据少数服从多数的原则，将测试样本点归入在K个点中占比最高的那一类。

KNN-算法举例

电影名称	打斗镜头	拥抱镜头	电影类型
谍影重重	57	2	武打片
叶问3	65	2	武打片
我的特工爷爷	21	4	武打片
奔爱	4	46	情感片
夜孔雀	8	39	情感片
代理情人	2	38	情感片
这个杀手不太冷	49	6	?

表格中6个样本（电影）分别给出其特征（打斗镜头、拥抱镜头）和标签（电影类型）信息，现在给定一个新的样本（这个杀手不太冷），我们想知道这部电影的类型。由于是2维数据，我们可以用平面直角坐标系表示。

KNN-算法举例

KNN-算法举例

$$D_{\text{谍影重重}} = \sqrt{(49-57)^2+(6-2)^2} \approx 8.94$$

$$D_{\text{叶问 3}} = \sqrt{(49-65)^2+(6-2)^2} \approx 16.49$$

$$D_{\text{我的特工爷爷}} = \sqrt{(49-21)^2+(6-4)^2} \approx 28.07$$

$$D_{\text{奔爱}} = \sqrt{(49-4)^2+(6-46)^2} \approx 60.21$$

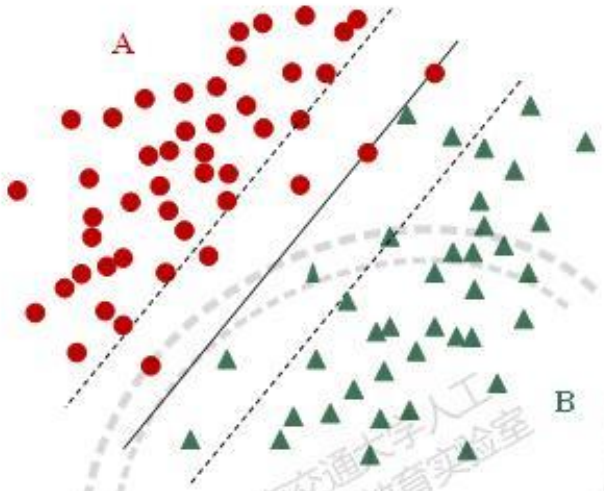
$$D_{\text{夜孔雀}} = \sqrt{(49-8)^2+(6-39)^2} \approx 52.63$$

$$D_{\text{代理情人}} = \sqrt{(49-2)^2+(6-38)^2} \approx 56.86$$

电影名称	与未知电影距离
谍影重重	8.94
叶问3	16.49
我的特工爷爷	28.07
夜孔雀	52.63
代理情人	56.86
奔爱	60.21

- 当K=3时，前三个样本出现最多的电影类型是武打片，因此《这个杀手不太冷》样本也应该归为武打片。
- 当K=5时，前5个样本出现最多的电影类型也是武打片，因此样本也属于武打片。

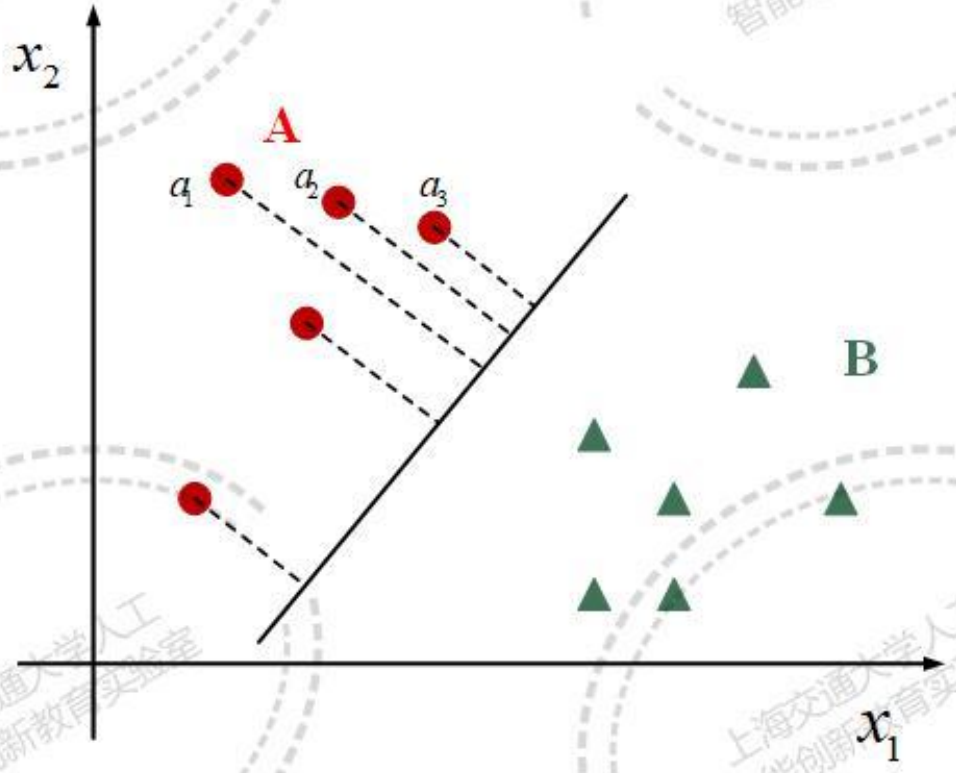
第三节 支持向量机



支持向量机

考察在二分类问题中的线性分类方法，给定采样点 $\{x_i, y_i\}_{i=1}^m$，其中 $x_i \in R^n$, $y_i \in \{-1, +1\}$，试图构建一个线性函数 $f(x) = w^T x + b$ 使得$f(x)$的符号与其标号相同，即利用 $f(x) = 0$作为分类面，在二维空间中，这样的分类面就是一条直线。

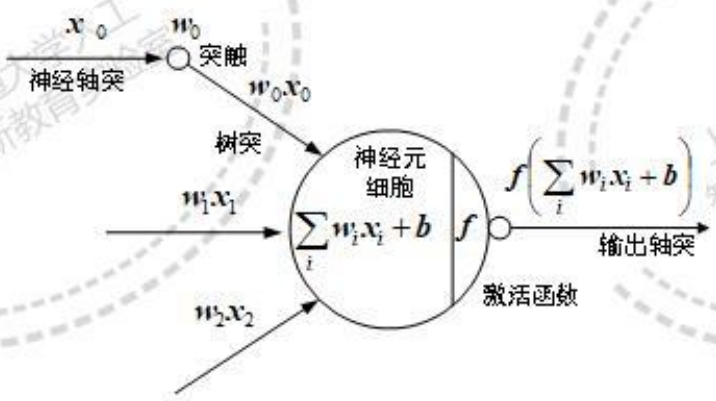
同样的数据可能存在多条这样的分类直线，那么它们之中存不存对分类最有利的直线呢？

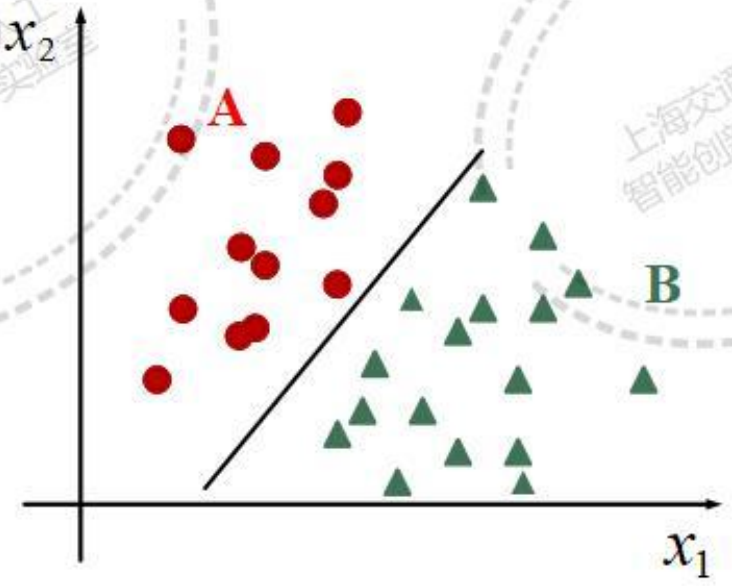


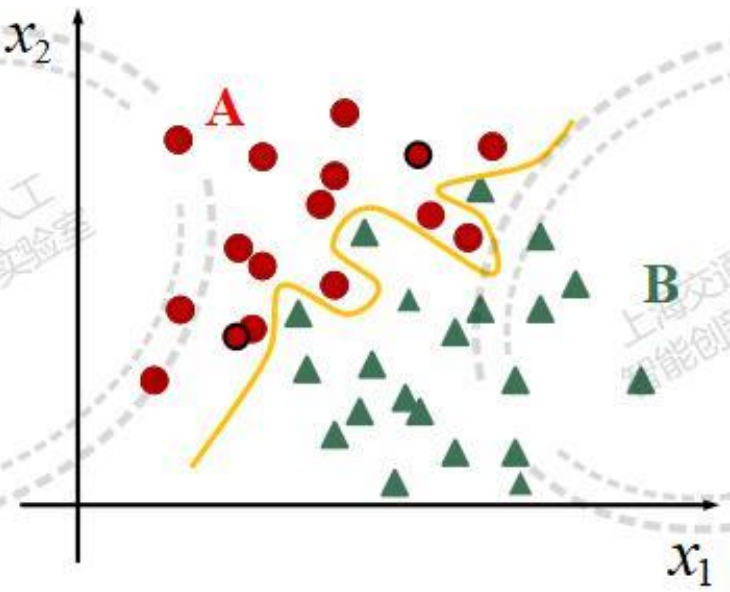
支持向量机—从感知机说起

感知机是二分类的线性模型，其输入是实例的特征向量，输出的是事例的类别，分别是+1和−1，属于判别模型。

假设训练数据集是线性可分的，感知机学习的目标是求得一个能够将训练数据集**正实例点和负实例点完全正确分开的分离超平面**。如果是非线性可分的数据，则最后无法获得超平面







支持向量机—从感知机说起

点到线的距离

直线方程 $Ax + By + C = 0$

点P的坐标是$(x_0^1,\ x_0^2)$到直线的距离为

$$d_0 = \frac{Ax_0^1 + Bx_0^2 + C}{\sqrt{A^2 + B^2}}$$

对于高维的数据，点到超平面的距离为

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

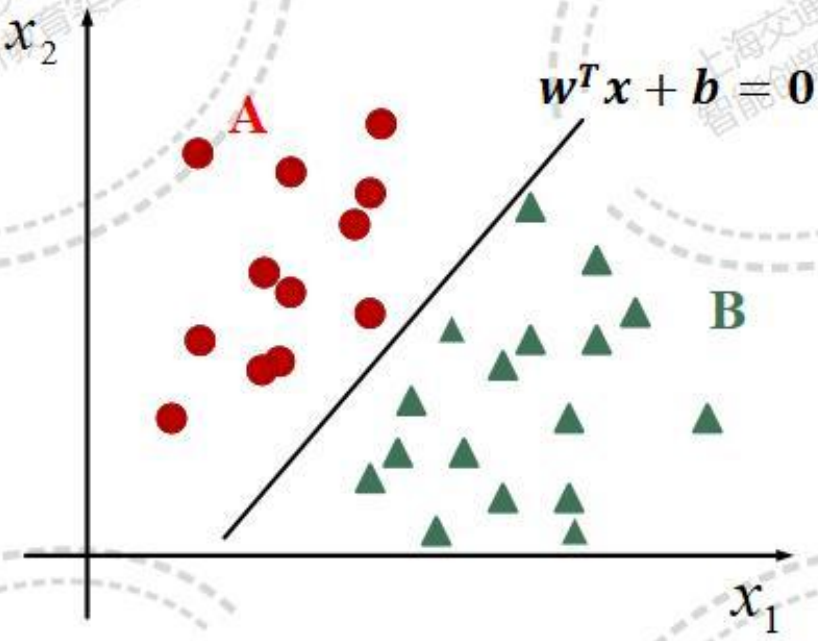
其中，$w = (w_0,\ w_1,\ w_2,\ \cdots,\ w_n)$

$$X = (x_0,\ x_1,\ x_2,\ \cdots,\ x_n)$$

感知机从输入空间到输出空间的模型如下：

$$f(x) = sign(w^T x + b)$$

$$sign(x) = \begin{cases} -1 & x \le 0 \\ 1 & x > 0 \end{cases}$$



需要注意：

对于感知机，线性分类问题

样本为$X = (x_0,\ x_1,\ x_2,\ \cdots,\ x_n)$，

其中，如一个样本点 $x_0 = (x_0^1,\ x_0^2)$；

对于直线方程有：$w = (A, B)$；

通常写作：$w = (w_0,\ w_1)$

支持向量机—从感知机说起

首先定义对于样本 $(x_i,\ y_i)$, 其中y_i样本类别属性：

如果$\frac{w^T x_i+b}{\|w\|} > 0$, 则记$y_i = +1$

如果$\frac{w^T x_i+b}{\|w\|} < 0$, 则记$y_i = -1$

这样，判断分类是否正确时，可知：

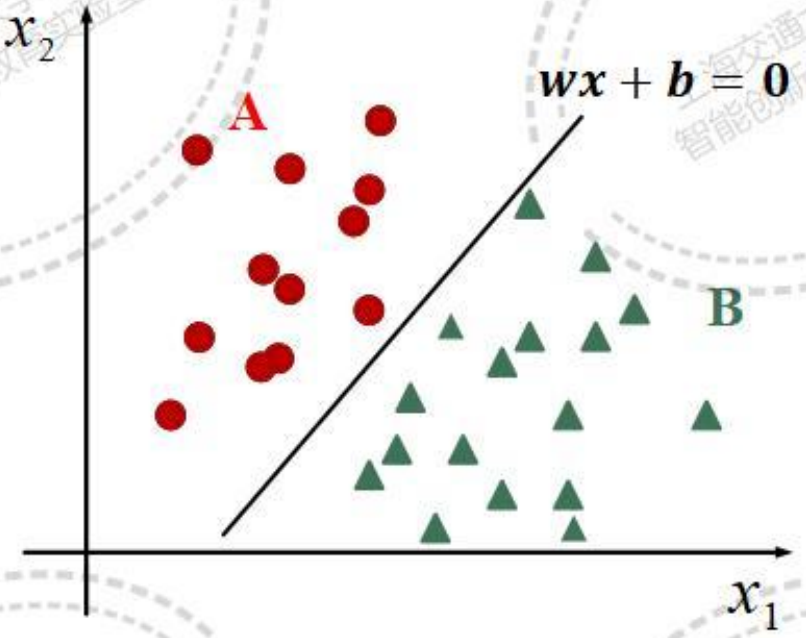
正确分类的样本满足 $y_i \frac{w^T x_i+b}{\|w\|} > 0$

错误分类的样本会是 $y_i \frac{w^T x_i+b}{\|w\|} < 0$

期望使误分类的所有样本，到超平面的距离之和最小。所以损失函数定义如下：

$$L(w,\ b) = -\frac{1}{\|w\|}\sum_{x_i \in M} y_i(w^T x + b),$$

其中M集合是误分类点的集合



需要注意：

对于感知机，线性分类问题

样本为$X = (x_0,\ x_1,\ x_2,\ \cdots,\ x_n)$,

其中，如一个样本点 $x_0 = (x_0^1,\ x_0^2)$;

对于训练样本点都会赋予一个类别信息$y_i \in [-1, +1]$。

支持向量机—从感知机说起

感知机模型的学习算法：

输入：训练集

$$\mathrm{T}=(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N);\ y_i\in[-1,+1],\ \text{学习率}\ \gamma(0<\gamma<1)$$

输出：w，b；感知机模型 $f(x)=sign(wx+b)$

1. 赋初值：w_0，b_0;

2. 选取数据点：(x_i,y_i);

3. 判断该数据点是否为当前模型的误分类点，

即判断若 $y_i(wx+b)<0$ 则更新

$$w=w+\gamma y_i x_i$$

$$\mathrm{b}=b+\gamma y_i$$

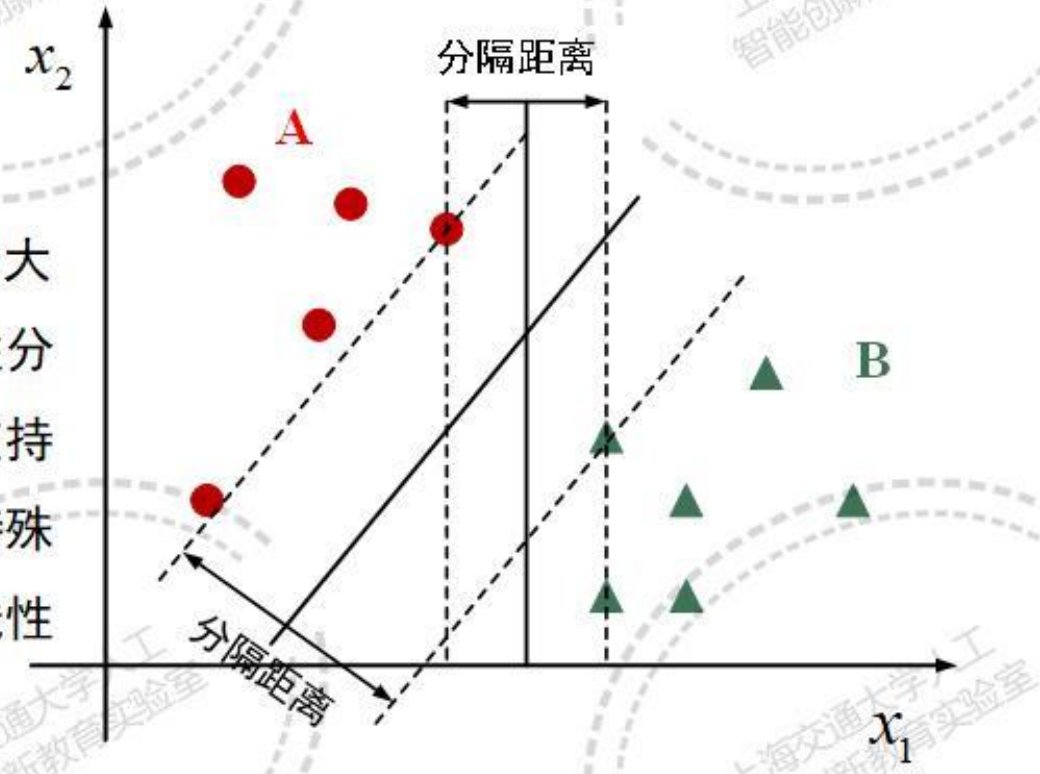
4. 转到2，指导训练集中没有误分类点。

支持向量机—线性分类器

支持向量机

（support vector machine， SVM）

就是一种在特征空间上使分类间隔最大的分类器，它对两个类别进行分类。线性分类器是分类器中的一种，类似地，线性支持向量机也是支持向量机中的一种。若无特殊说明，我们这里说的支持向量机指的是线性支持向量机。



支持向量机—线性分类器

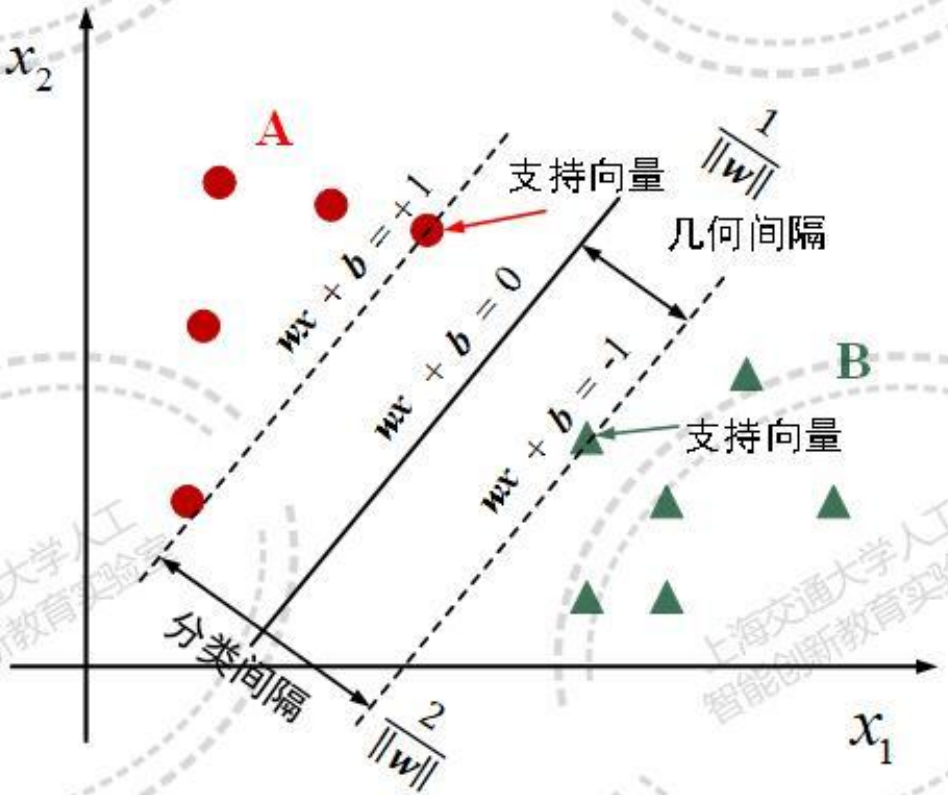
离分类直线最近的这些样本点对分类间隔大小非常“重要”，起着关键作用，这些样本点被称为支持向量。

沿用感知机部分的问题描述与符号表示，如图

几何间隔

$$\frac{|w_1x_1 + w_2x_2 + \mathbf{b}|}{\sqrt{w_1^2 + w_2^2}}$$

$$= \mathbf{y} \times \frac{w_1x_1 + w_2x_2 + b}{\sqrt{w_1^2 + w_2^2}} = \frac{1}{\|w\|}$$



分类间隔

分类间隔是支持向量几何间隔的两倍 $\frac{2}{\|w\|}$ 。

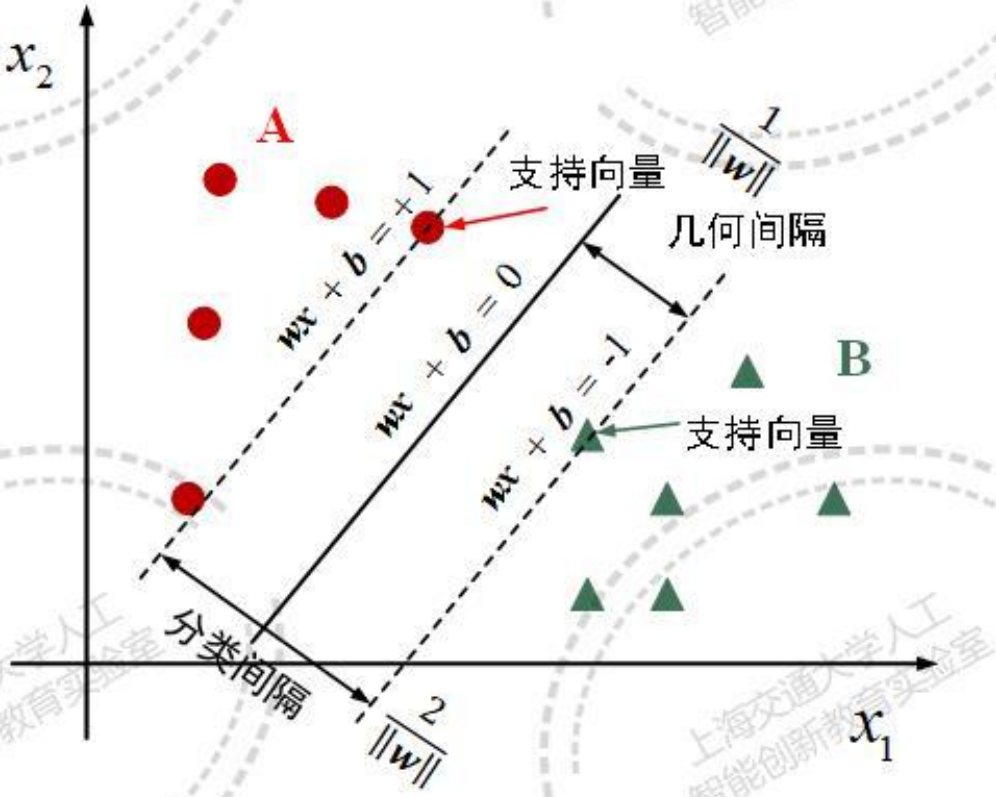
支持向量机—线性分类器损失函数

为了能从训练样本集中得到支持向量机的模型，目标就是让分类间隔最大。

这个间隔就是$\frac{1}{\|w\|}$，即 $\max_{w,b}\left\{\frac{1}{\|w\|}\right\}$，使得 $y_i \times (w^T x + b) \geq 1$

也就是：$\min_{w,b} \frac{1}{2}\|w\|^2$，

使得 $y_i \times (w^T x + b) - 1 \geq 0$



支持向量机—线性分类器软间隔分类

通常情况下的训练集中都会存在一些异常点，而这些异常点会导致训练集线性不可分，但除去这些异常点之后，剩下的样本就是线性可分的，而上面讲到的硬间隔最大化是无法处理线性不可分的问题，线性不可分意味着有些样本点的函数间隔是不能满足大于等于 1 的约束条件。

“硬”间隔分类条件：$\min_{w,b} \frac{1}{2}\|w\|^2$，**使得** $y_i \times (w^T x + b) - 1 \geq 0$

软间隔优化目标变为：

$$\min_{w,b} \quad \frac{1}{2}\|w\|_2^2 + C\sum_i L(y_i(w^T x_i + b))$$

其中C为常数，L为hinge损失函数：

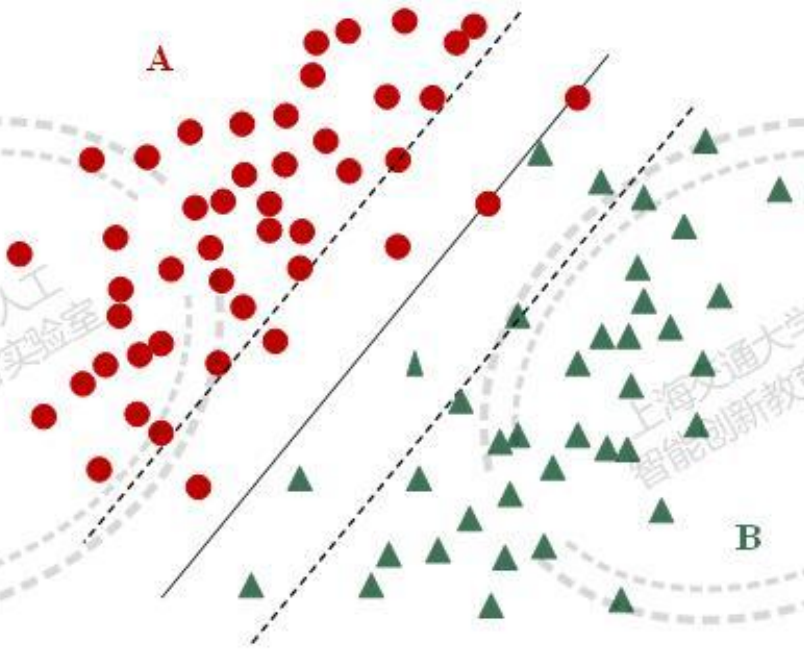
$$L(u) = \max\{0,\ 1-u\}$$

上面的软间隔优化目标函数等价于：

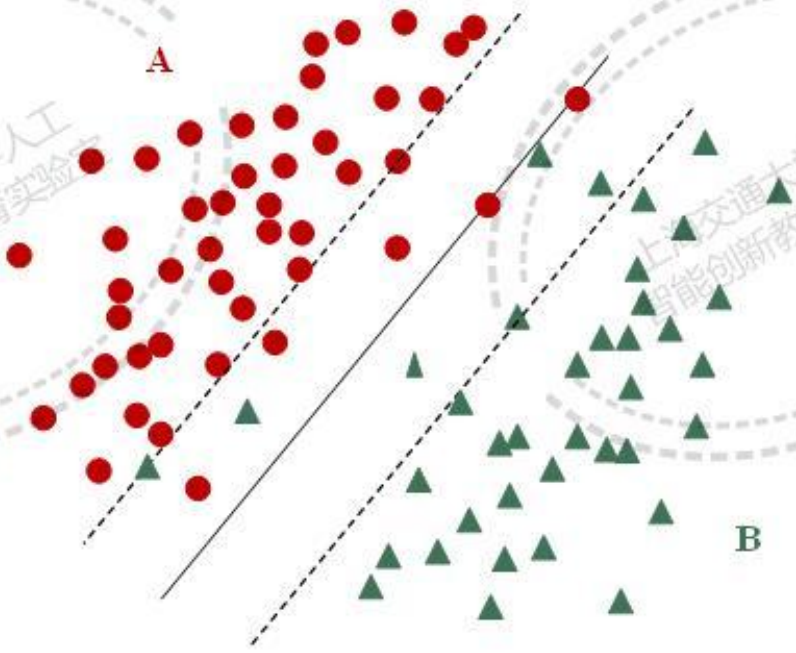
$$\min_{w,b} \quad \frac{1}{2}\|w\|_2^2 + C\sum_i \rho_i$$

使得 $\quad y_i(w^T x_i + b) \geq 1 - \rho_i$

$$\rho_i \geq 0$$



支持向量机—软间隔分类



- 对分错的点，给予惩罚

$$\min_{w,b} \quad \frac{1}{2}\|w\|_2^2 + C\sum_i L(y_i(w^T x_i + b))$$

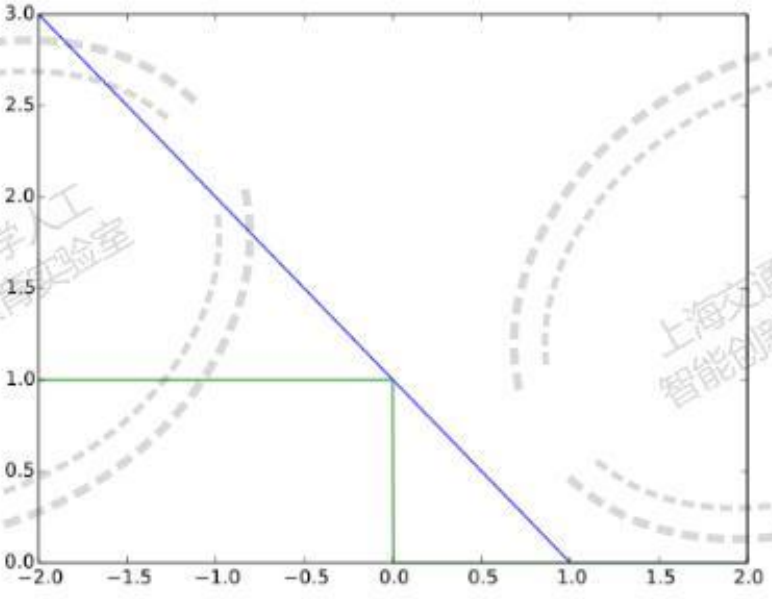
- 链接罚函数（hinge loss）

$$L(u) = \max\{0,\ 1-u\}$$

$$\min_{w,b} \quad \frac{1}{2}\|w\|_2^2 + C\sum_i \rho_i$$

$$\text{s.t.} \quad y_i(w^T x_i + b) \geq 1 - \rho_i$$

$$\rho_i \geq 0$$

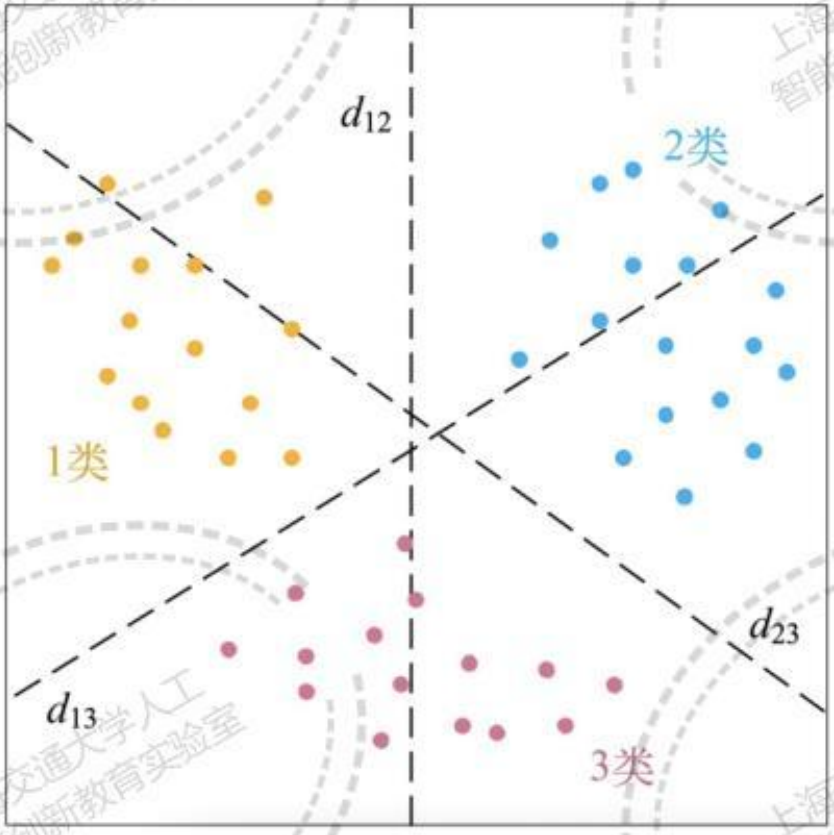


支持向量机—从二分类到多分类

- 方案1：1对其余方法(one-against-all)
  - 训练类A与其他，类B与其他，类C与其他的二分类器

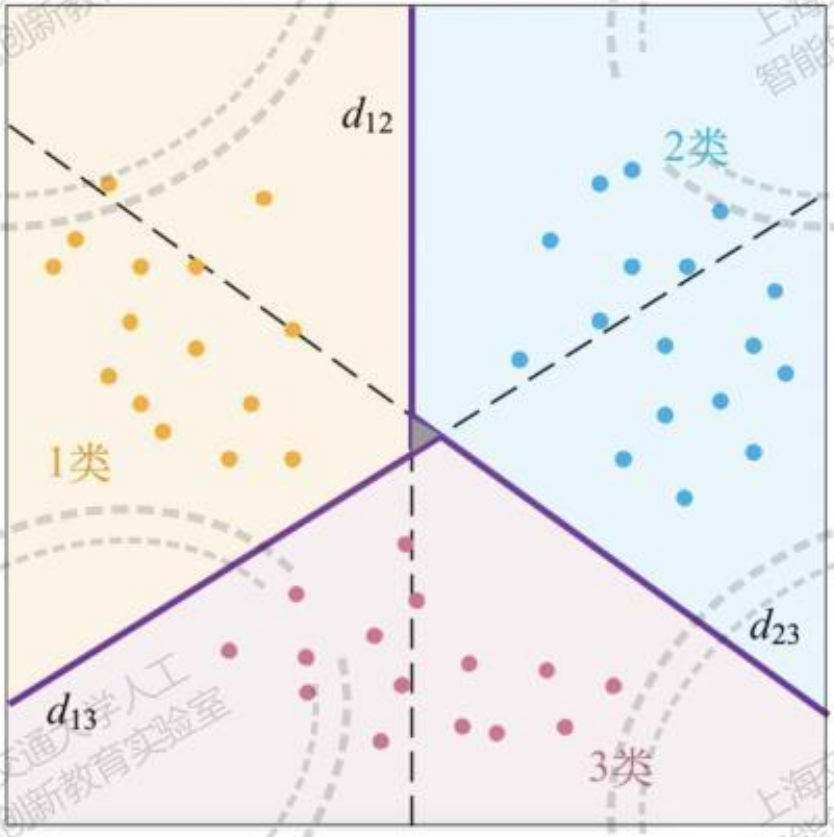
- 方案2：1对1方法（one-against-one）
  - 训练类A与B，类B与C，类C与A的二分类器
  - 投票选出类别
    - 每个二分类SVM根据其决策函数对新数据有一个预测(投票)，以 i 类和 j 类之间的二分类SVM为例，若对新数据的预测为 i 类，则 i 类得票加1；否则 j 类得票加1；
    - 最终得票最多的类别就是对新数据的预测；
    - 若出现平票的情况，(虽然可能不是一个好方法)，简单地选择索引较小的那个类别作为对新数据的分类。

支持向量机—举例：1对1



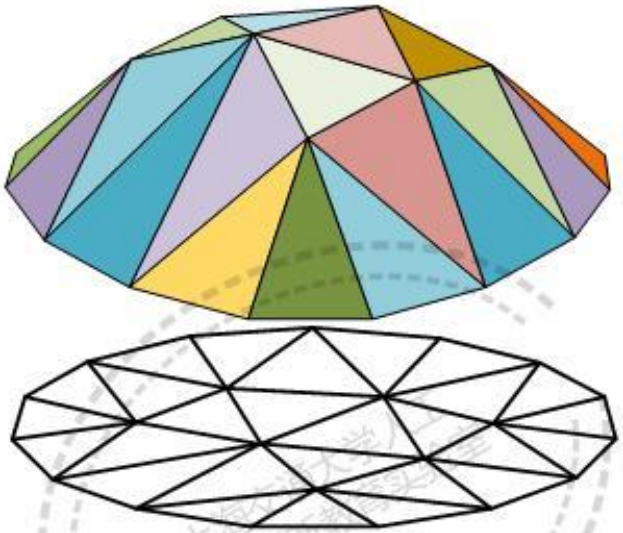
- 虚线 d_{12} 表示1类和2类数据之间的决策边界
- 虚线 d_{13} 表示1类和3类之间的决策边界
- 虚线 d_{23} 表示2类和3类之间的决策边界

支持向量机—举例：1对1

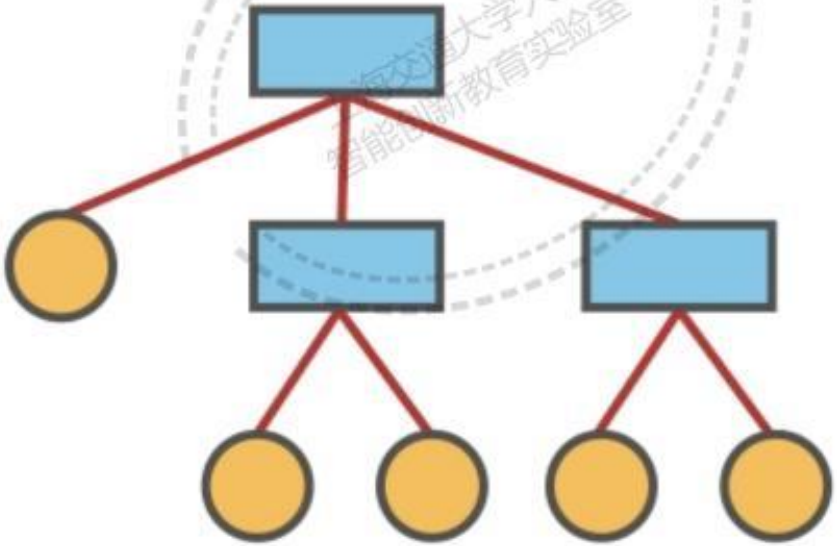


- 若新数据位于橙色区域，则其对1类的得票为2票，会被预测为1类
- 同理，位于蓝色和红色区域的数据会分别被预测为2类和3类
- 中心灰色区域，对三个类的得票均为1票

第四节 决策树



决策树



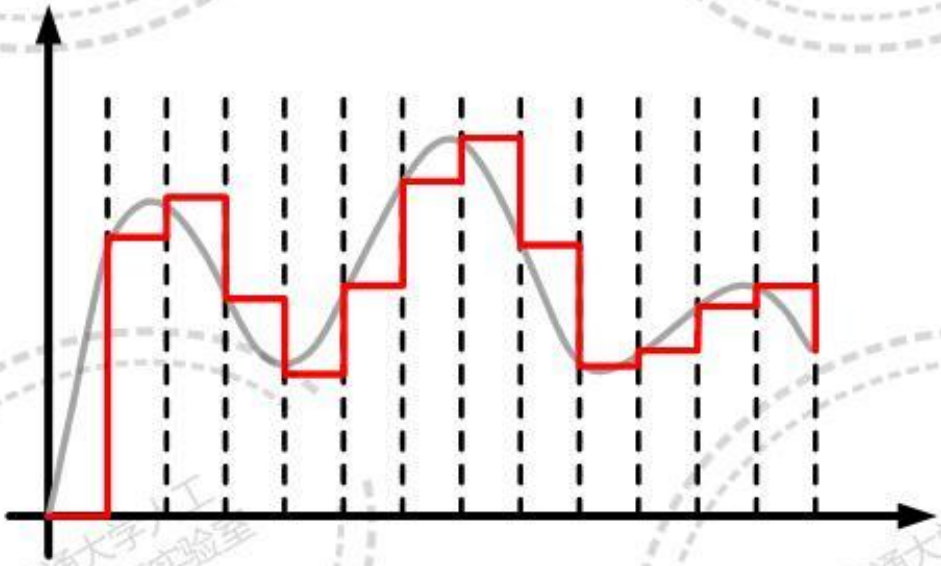
决策树学习算法以树形结构建立模型，类似于流程图，模型本身包含一系列逻辑决策，带有表明根据某一特性做出决定的决策节点。决策树从根节点开始，由叶节点结束，节点引出的分枝表示可做出的选择，叶结点表示遵循决策组合的结果。

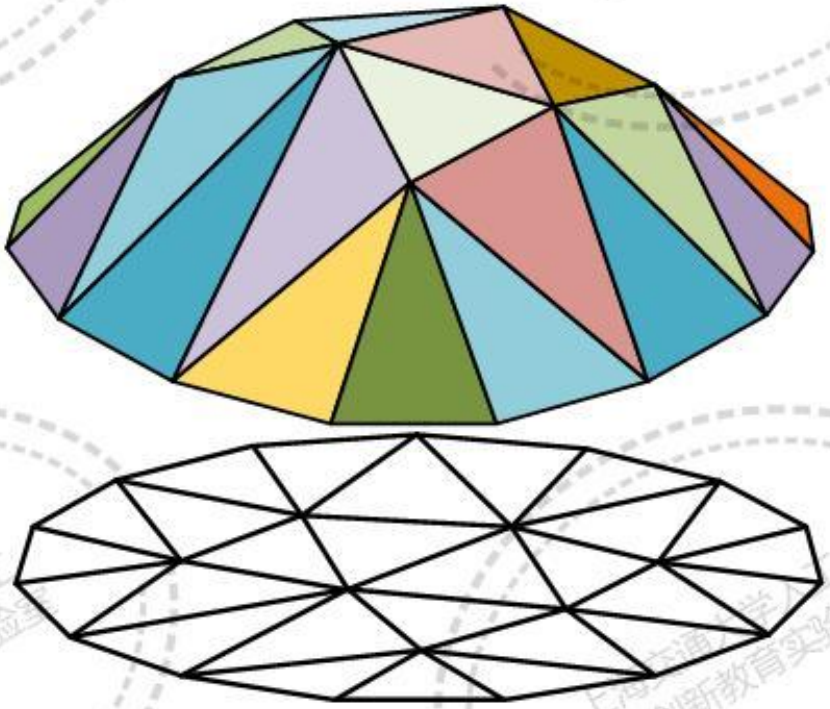
决策树是最广泛使用的机器学习技术之一，它几乎可以用于任何类型的数据建模，同时具有无与伦比的性能。

决策树

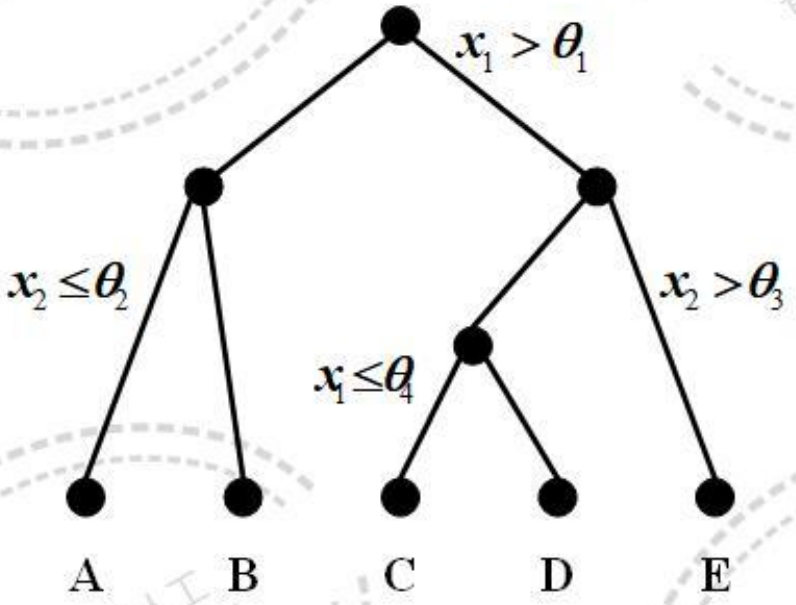
将区域划分为足够小的部分

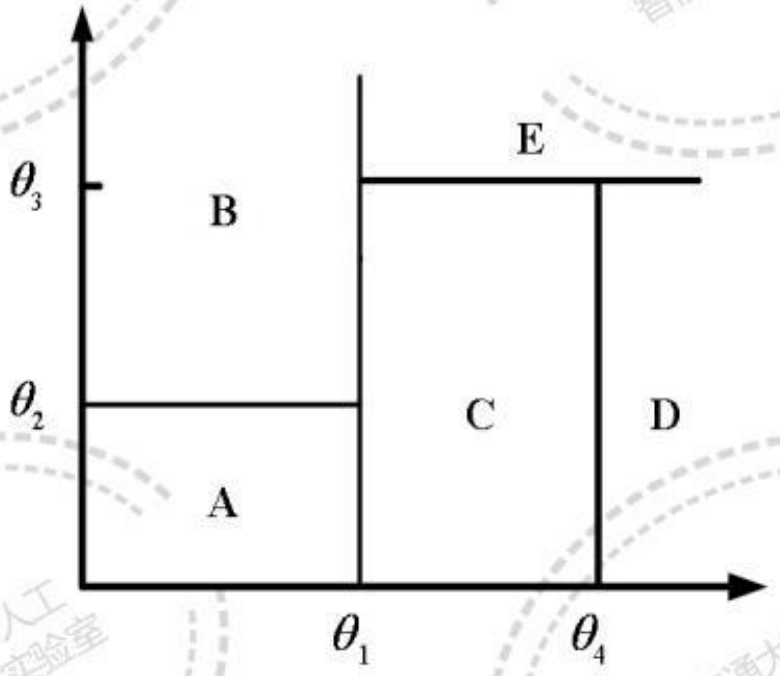
在每个子区域内使用线性或者常值函数





决策树—决策树与子区域

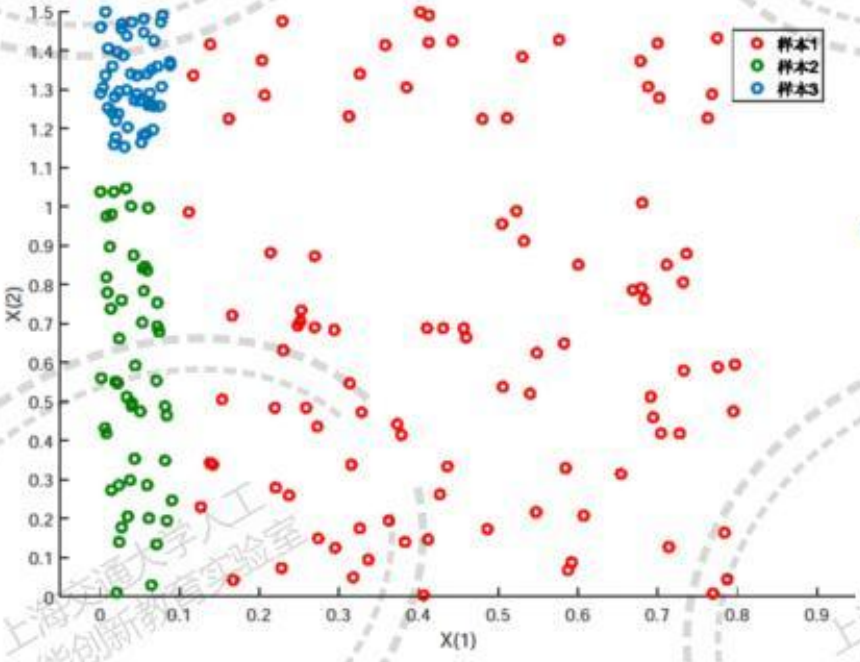

$x_1 > \theta_1$
$x_2 \le \theta_2$
$x_2 > \theta_3$
$x_1 \le \theta_4$
A
B
C
D
E

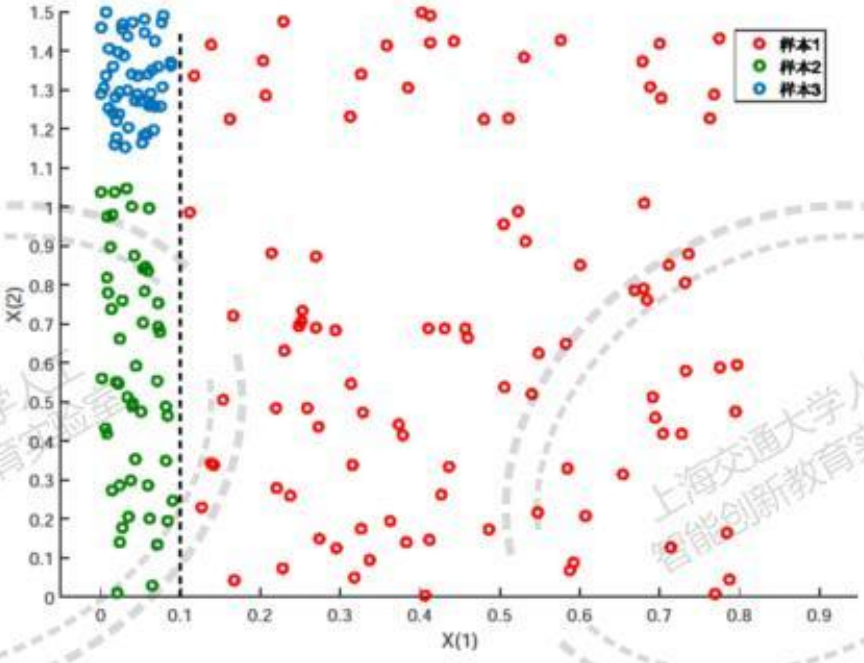

θ_3
θ_2
B
E
C
D
A
θ_1
θ_4

决策树—决策树的构建

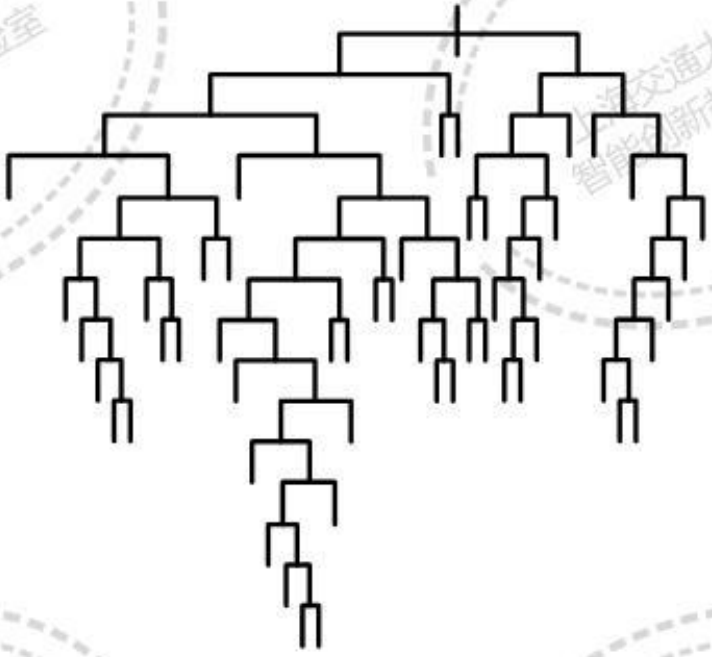
基尼不纯度: $-\sum_i P(c_i)\log P(c_i)$

第 i 类占的比例

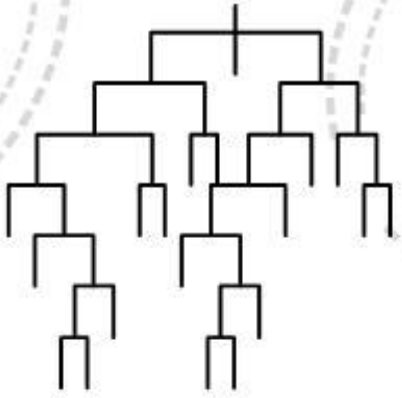




决策树—决策树的“剪枝”



不纯度 = 0



不纯度 = 0.1

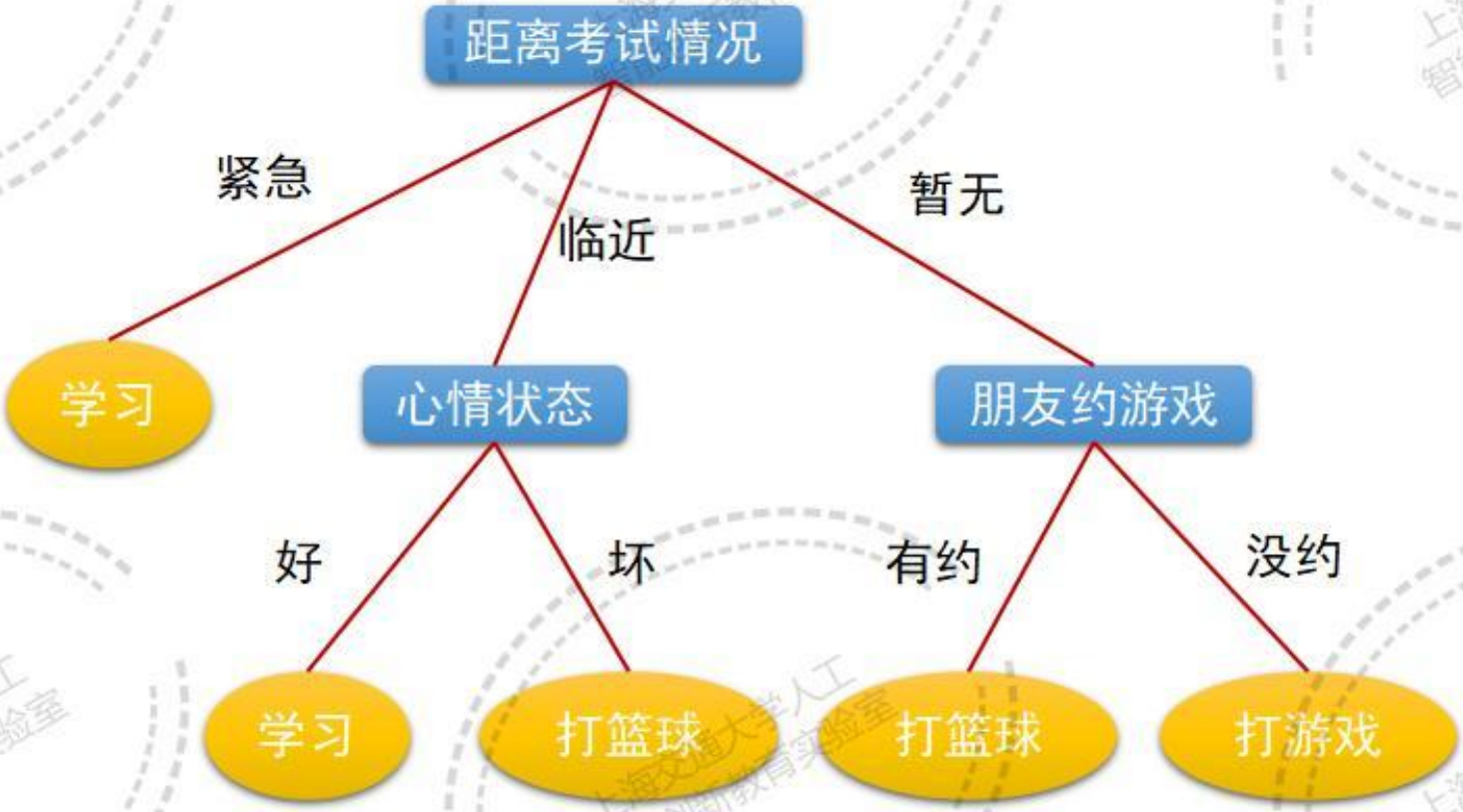
The Simpler, The Better
“越简单的模型越好”

决策树—放学后活动选择

决策树—放学后活动选择

编号	距离考试情况	心情	朋友约游戏	放学后活动
1	紧急	好	有约	学习
2	紧急	好	没约	学习
3	紧急	坏	有约	学习
4	紧急	坏	没约	学习
5	临近	好	有约	学习
6	临近	好	没约	学习
7	临近	坏	有约	打篮球
8	临近	坏	没约	打篮球
9	暂无	好	有约	打游戏
10	暂无	好	没约	打篮球
11	暂无	坏	有约	打游戏
12	暂无	坏	没约	打篮球

决策树—放学后活动选择



决策树—熵和信息增益

样本数据的**熵**表示分类值之间混杂的程度，最小值0表示的样本是完全同质的，而1表示样本凌乱的程度最大。熵的定义为：

$$\text{Entropy}\ (S) = \sum_{i=1}^{c} -p_i \lg_2(p_i)$$

在熵的公式中，对于给定的数据分割 (S)，常数 c 代表类的个数，p_i 代表第 i 类样本占整个数据集样本的比例。

决策树算法使用熵值来计算在每一个可能特征上的划分所引起的同质性（均匀性）变化，该变化称为**信息增益**。对于特征 (F)，信息增益等于划分前数据(S_1)的信息熵减去依该特征划分后的数据(S_2)的信息熵。

$$\text{InfoGain}(F) = \text{Entropy}(S_1) - \text{Entropy}(S_2)$$

经过一次分割后数据被划分到多个分区中，因此计算需要考虑所有分区熵的总和，这可以根据样本落入各个分区中的比例来加权计算。

决策树—ID3算法

决策树算法有多种类型，本章介绍的**ID3算法**主要针对属性（特征）选择问题，是决策树学习方法中，最具影响和最为典型的算法，该方法使用信息增益选择测试属性。

ID3算法的主要思想是：

- $Gain（S，A）$是属性A在集合S上的信息增益
- $Gain(S，A) = Entropy(S) - Entropy(S，A)$
- $Gain(S，A)$越大，说明选择的测试属性对分类提供的信息越多

决策树—实例

以下用常见的例子来说明ID3算法的过程，例如电器用品公司统计数笔问卷的信息，想分析数据内在的关系是否能知道什么属性的购买者会买计算机。

也就是寻找出购买计算机的人群属性。

编号	计数	年龄	收入	学生	信誉	归类：买计算机？
1	64	青	高	否	良	不买
2	64	青	高	否	优	不买
3	128	中	高	否	良	买
4	60	老	中	否	良	买
5	64	老	低	是	良	买
6	64	老	低	是	优	不买
7	64	中	低	是	优	买
8	128	青	中	否	良	不买
9	64	青	低	是	良	买
10	132	老	中	是	良	买
11	64	青	中	是	优	买
12	32	中	中	否	优	买
13	32	中	高	是	良	买
14	63	老	中	否	优	不买

决策树—第一步：计算决策属性的熵

决策属性是“买计算机？”，该属性分为两类：买/不买。

先算出买与不买的概率，再算出决策属性的熵。 具体步骤如下：

1. 统计买计算机与不买计算机的人数

买的人数为 $S_1 = 640$； 不买的人数为 $S_2 = 383$；总人数为1023；

买的概率为 $P_1 = \frac{640}{1023} = 0.6256$， $P_2 = \frac{383}{1023} = 0.3744$；

信息熵为 $E(S_1, S_2) = E(640,1023) = -P_1 log_2 P_1 - P_2 log_2 P_2 = 0.9540$

决策树—第二步：计算条件属性的熵

2. 计算不同属性的信息增益：

条件属性有4个：年龄、收入、学生、信誉

分别计算各个属性的信息增益

(1) 年龄属性：青年、中年、老年

青年的熵：

买的人数为 $S_1 = 128$；不买的人数为 $S_2 = 256$；总人数为384；

买的概率为 $P_1 = \frac{128}{384} = 0.3333$， $P_2 = \frac{256}{384} = 0.6667$；

信息熵为 $E(S_1, S_2) = E(128, 256) = -P_1 log_2 P_1 - P_2 log_2 P_2 = 0.9183$

中年的熵：

买的人数为 $S_1 = 256$；不买的人数为 $S_2 = 0$；总人数为256；

买的概率为 $P_1 = \frac{256}{256} = 1$， $P_2 = \frac{0}{384} = 0$；

信息熵为 $E(S_1, S_2) = E(256, 0) = -P_1 log_2 P_1 - P_2 log_2 P_2 = 0$

决策树—第二步：计算条件属性的熵

2. 计算不同属性的信息增益：

条件属性有4个：年龄、收入、学生、信誉

分别计算各个属性的信息增益

首先计算年龄的信息增益

（1）年龄属性：青年、中年、老年

老年的熵：

买的人数为 $S_1 = 256$；不买的人数为 $S_2 = 127$；总人数为383；

买的概率为 $P_1 = \frac{256}{383} = 0.6684$，$P_2 = \frac{127}{383} = 0.3316$；

信息熵为 $E(S_1, S_2) = E(256, 127) = -P_1 log_2 P_1 - P_2 log_2 P_2 = 0.9166$

（2）计算年龄的平均信息熵期望

青年人数占比：384/1023=0. 3754；中年人数占比：256/1023=0. 2502；

老年人数占比：383/1023=0. 3744；

平均信息熵期望 = 0. 3754*0. 9183+0. 2502*0+0. 3744*0. 9166=0. 6879

决策树—第二步：计算条件属性的熵

2. 计算不同属性的信息增益：

条件属性有4个：年龄、收入、学生、信誉

分别计算各个属性的信息增益

（3）计算年龄属性的信息增益

Gain(年龄信息增益) = 0. 9540 − 0. 6879 = 0. 2661

其次，计算收入属性的信息增益

（1）收入属性：高收入、中收入、低收入

高收入的熵：

买的人数为 $S_1 = 160$；不买的人数为 $S_2 = 128$；总人数为288；

买的概率为 $P_1 = \frac{160}{288} = 0.5556$，$P_2 = \frac{128}{288} = 0.4444$；

信息熵为 $E(S_1, S_2) = E(160,\ 128) = -P_1 log_2 P_1 - P_2 log_2 P_2 = 0.9911$

决策树—第二步：计算条件属性的熵

2. 计算不同属性的信息增益：

条件属性有4个：年龄、收入、学生、信誉

其次，计算收入属性的信息增益

（1）收入属性：高收入、中收入、低收入

中收入的熵：

买的人数为 $S_1 = 288$；不买的人数为 $S_2 = 191$；总人数为479；

买的概率为 $P_1 = \frac{288}{479} = 0.6013$， $P_2 = \frac{191}{479} = 0.3987$；

信息熵为 $E(S_1, S_2) = E(288, 191) = -P_1 log_2 P_1 - P_2 log_2 P_2 = 0.9702$

低收入的熵：

买的人数为 $S_1 = 192$；不买的人数为 $S_2 = 64$；总人数为256；

买的概率为 $P_1 = \frac{192}{256} = 0.7500$， $P_2 = \frac{64}{256} = 0.2500$；

信息熵为 $E(S_1, S_2) = E(192, 64) = -P_1 log_2 P_1 - P_2 log_2 P_2 = 0.8113$

决策树—第二步：计算条件属性的熵

2. 计算不同属性的信息增益：

条件属性有4个：年龄、收入、学生、信誉

分别计算各个属性的信息增益

其次，计算收入属性的信息增益

（1）收入属性：高收入、中收入、低收入

（2）计算收入的平均信息熵期望

高收入占288/1023=0. 2815；中收入占479/1023=0. 4682；

低收入占256/1023=0. 2502。

平均信息期望 = 0. 2815*0. 9911+0. 4682*0. 9702+0. 2502*0. 8113=0. 9362

（3）计算年龄属性的信息增益

Gain(收入信息增益) = 0. 9540 − 0. 9362 = 0. 0178

决策树—第二步：计算条件属性的熵

2. 计算不同属性的信息增益：

条件属性有4个：年龄、收入、学生、信誉

<u>第三，计算学生属性的信息增益</u>

（1）学生属性：学生、非学生

学生的熵：

买的人数为 $S_1 = 420$；不买的人数为 $S_2 = 64$；总人数为484；

买的概率为 $P_1 = \frac{420}{484} = 0.8678$，$P_2 = \frac{64}{484} = 0.1322$；

信息熵为 $E(S_1, S_2) = E(420, 64) = -P_1 log_2 P_1 - P_2 log_2 P_2 = 0.5634$

非学生的熵：

买的人数为 $S_1 = 220$；不买的人数为 $S_2 = 319$；总人数为539；

买的概率为 $P_1 = \frac{220}{539} = 0.4082$，$P_2 = \frac{319}{539} = 0.5918$

信息熵为 $E(S_1, S_2) = E(220, 319) = -P_1 log_2 P_1 - P_2 log_2 P_2 = 0.9755$

决策树—第二步：计算条件属性的熵

2. 计算不同属性的信息增益：

条件属性有4个：年龄、收入、学生、信誉

第三，计算学生属性的信息增益

（1）学生属性：学生、非学生

（2）计算学生属性的平均信息熵期望

学生占 484/1023=0.4731；非学生占 539/1023=0.5269；

平均信息期望 0.4731*0.5634+0.5269*0.9755=0.7805

（3）计算学生属性的信息增益

Gain(学生信息增益) = 0.9540 − 0.7805 = 0.1735

决策树—第二步：计算条件属性的熵

2. 计算不同属性的信息增益：

条件属性有4个：年龄、收入、学生、信誉

第四，计算信誉属性的信息增益

（1）信誉属性：优、良

信誉优的熵：

买的人数为 S_1 =160；不买的人数为 $S_2 = 191$；总人数为351；

买的概率为 $P_1 = \frac{160}{351} = 0.4558$， $P_2 = \frac{191}{351} = 0.5442$；

信息熵为 $E(S_1, S_2) = E(160, 191) = -P_1 log_2 P_1 - P_2 log_2 P_2$=0.9944

信誉良的熵：

买的人数为 S_1 =480；不买的人数为 $S_2 = 192$；总人数为672；

买的概率为 $P_1 = \frac{480}{672} = 0.7143$， $P_2 = \frac{192}{672} = 0.2857$

信息熵为 $E(S_1, S_2) = E(480, 192) = -P_1 log_2 P_1 - P_2 log_2 P_2 = 0.8631$

决策树—第二步：计算条件属性的熵

2. 计算不同属性的信息增益：

条件属性有4个：年龄、收入、学生、信誉

第三，计算信誉属性的信息增益

（1）信誉属性：优、良

（2）计算信誉属性的平均信息熵期望

信誉优占 351/1023=0.3431；信誉良占 672/1023=0.6569；

平均信息期望 = 0.6569*0.8631+0.3431*0.9944=0.9052

（3）计算信誉属性的信息增益

Gain(信誉信息增益) = 0.9540 − 0.9052 = 0.0488

决策树—第三步：选择属性

比较四个属性的信息增益，

年龄：0.2661

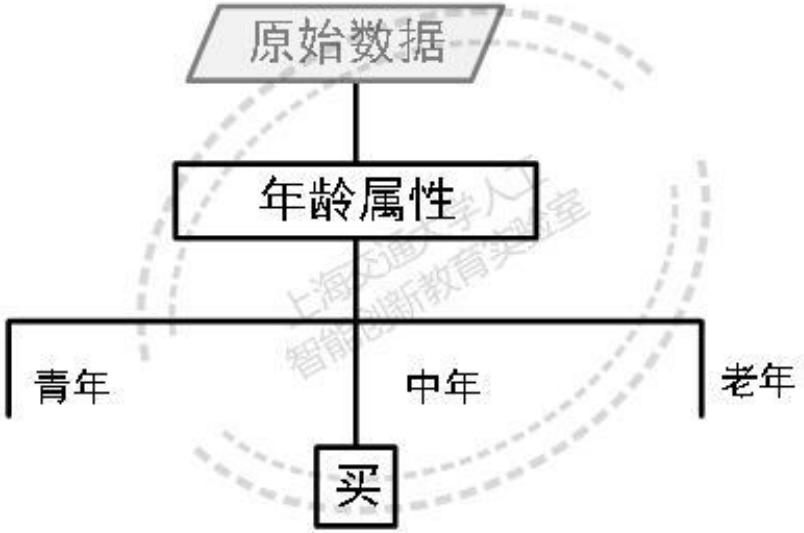
收入：0.0178

学生：0.1735

信誉：0.0488

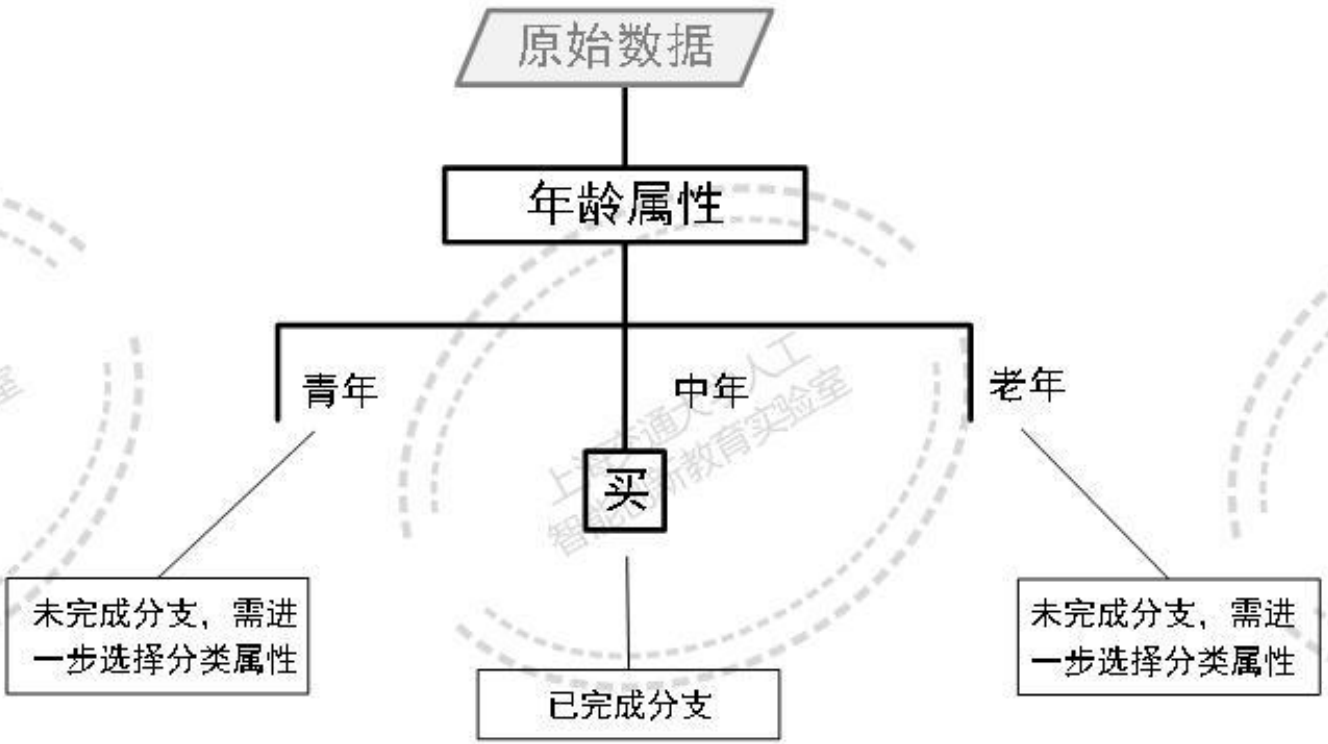
年龄属性获得的信息增益最大

选择值最大的为决策树分枝的特征（属性）：年龄



决策树—第四步：继续分枝

根据还没有分类完成的数据继续重复第一步到第三步直到所有分枝都为同一类才停止计算。

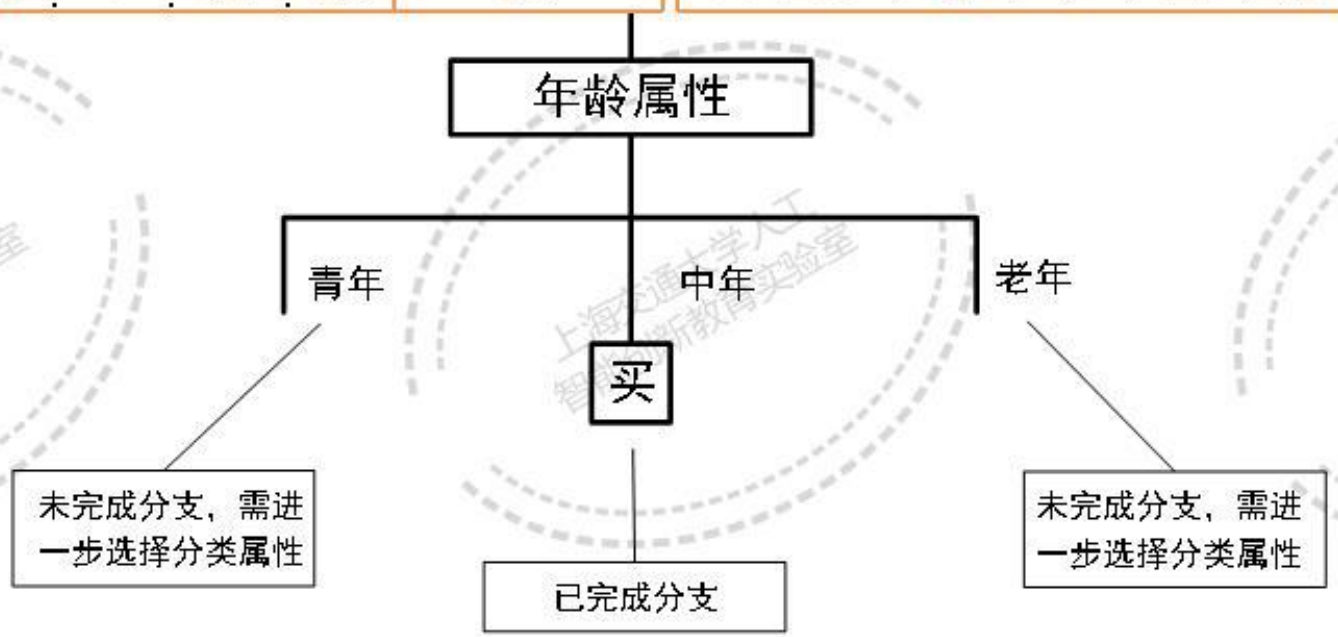


决策树—第四步：继续分枝

青年和老年分支通过剩下三个属性来确定选择哪个属性来继续向下分支

编号	计数	~~年龄~~	收入	学生	信誉	归类：买计算机？
1	64	青	高	否	良	不买
2	64	青	高	否	优	不买
8	128	青	中	否	良	不买
9	64	青	低	是	良	买
11	64	青	中	是	优	买

编号	计数	~~年龄~~	收入	学生	信誉	归类：买计算机？
4	60	老	中	否	良	买
5	64	老	低	是	良	买
6	64	老	低	是	优	不买
10	132	老	中	是	良	买
14	63	老	中	否	优	不买



决策树—第四步：继续分枝

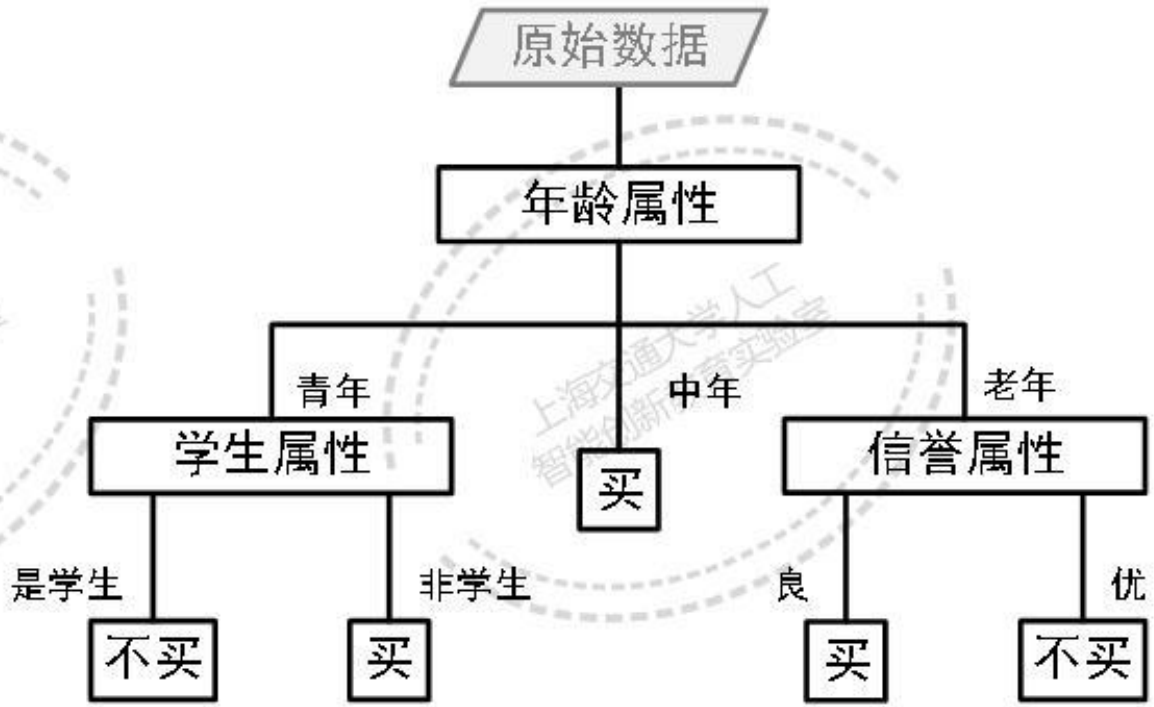
青年和老年分支通过剩下三个属性来确定选择哪个属性来继续向下分支，步骤仍然与前面相同。

在青年属性分支获得如下结果：

G(学生属性信息增益)＞ G(收入属性信息增益)＞ G(信誉属性信息增益)

在老年属性分支获得如下结果：

G(信誉属性信息增益)＞ G(收入属性信息增益)＞ G(学生属性信息增益)



决策树—连续型数据和离散型数据

- 离散型数据

  学历、职业、缴费方式、客户流失

- 连续型数据

  年龄、在网时长、费用变化率

年龄	学历	职业	缴费方式	在网时长/年	费用变化率/%	客户流失
58	大学	公务员	托收	13	10.00	NO
47	高中	工人	营业厅缴费	9	42.00	NO
26	研究生	公务员	充值卡	2	63.00	YES
28	大学	公务员	营业厅缴费	5	2.91	NO
32	初中	工人	营业厅缴费	3	2.30	NO
42	高中	无业人员	充值卡	2	100	YES
68	初中	无业人员	营业厅缴费	9	2.30	NO

决策树—连续型数据离散化

- 连续型数据离散化
  - 年龄

    年龄≤40——青年；40＜年龄≤50——中年；年龄＞50——老年
  - 在网时长

    在网时长≤5——H1；在网时长＞5——H2
  - 费用变化率：F1、F2、F3

    费用变化率≤30%——F1；30＜费用变化率≤99%——F2；

    费用变化率=100%——F3

年龄	学历	职业	缴费方式	在网时长/年	费用变化率/%	客户流失
58	大学	公务员	托收	13	10.00	NO
47	高中	工人	营业厅缴费	9	42.00	NO
26	研究生	公务员	充值卡	2	63.00	YES
28	大学	公务员	营业厅缴费	5	2.91	NO
32	初中	工人	营业厅缴费	3	2.30	NO
42	高中	无业人员	充值卡	2	100	YES
68	初中	无业人员	营业厅缴费	9	2.30	NO

决策树—连续型数据离散化

- 连续型数据离散化
  - 年龄：青年、中年、老年
  - 在网时长：H1、H2
  - 费用变化率：F1、F2、F3

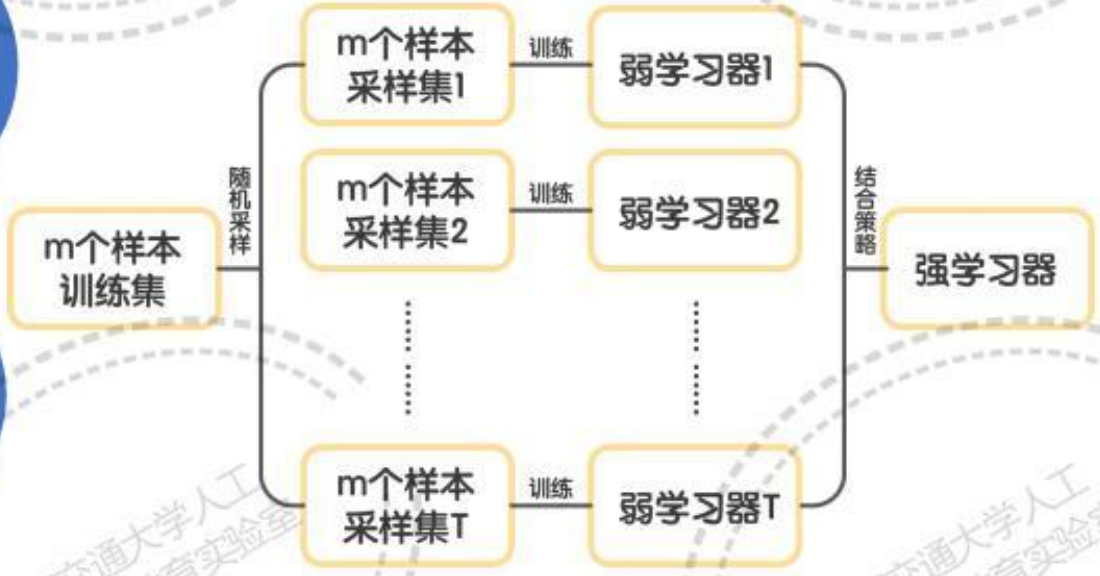
年龄	学历	职业	缴费方式	在网时长/年	费用变化率/%	客户流失
老年	大学	公务员	托收	H2	F1	NO
中年	高中	工人	营业厅缴费	H2	F2	NO
青年	研究生	公务员	充值卡	H1	F2	YES
青年	大学	公务员	营业厅缴费	H1	F1	NO
青年	初中	工人	营业厅缴费	H1	F1	NO
中年	高中	无业人员	充值卡	H1	F3	YES
老年	初中	无业人员	营业厅缴费	H2	F1	NO

随机森林-从集成学习说起

能从弱的学习机得到强的学习机，让“三个臭皮匠”顶得上“诸葛亮” ？

集成学习通过将多个学习器进行结合，通常可获得比单一学习器显著优越的泛化性能。

随机森林—基于Bagging



随机森林

随机森林（random forest，RF）是Bagging的一个扩展变体，它在以决策树为基学习器的基础上，进一步在决策树的训练过程中引入了随机属性选择。

具体来说，对基决策树的每个节点，从该节点的属性集合d 中随机选择一个包含k个属性的子集，然后再从这个子集中选择一个最优属性用于划分。这里的参数k控制了随机性的引人程度：

- 若令k=d，则基决策树的构建与传统决策树相同；
- 若令k=1，则是随机选择一个属性用于划分。