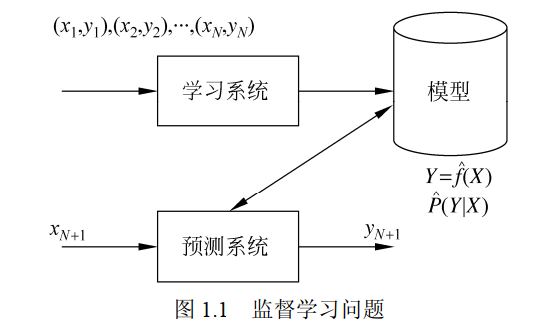
**学习第一天：**

统计学习（ statistical learning）是关于计算机基于数据构建概率统计模型并运用模型对数据进行预测与分析的一门学科。

监督学习利用训练数据集学习一个模型，再用模型对测试样本集进行预测的一种学习方法。



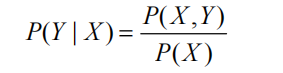
简单的来说监督学习的步骤大体为：

1. 首先给定训练数据集（样本数据）
2. 通过学习（或训练）得到 一个模型，表示为决策函数 Y = f(x)
3. 在预测过程中，预测系统对于给定的测试样本集中的输入进行输出。

监督学习中统计学习方法的三要素：方法＝模型＋策略＋算法

**生成模型和判别模型：**

生成方法由数据学习联合概率分布 P( X,Y)，然后求出条件概率分布P(Y|X )作为预测的模型，即生成模型:

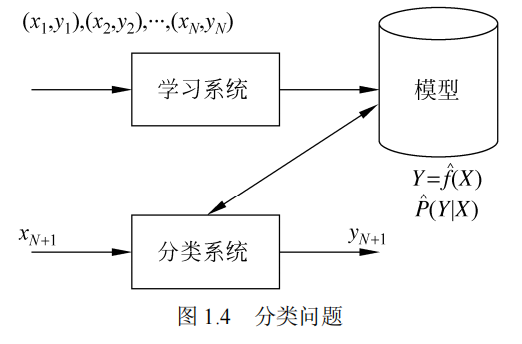
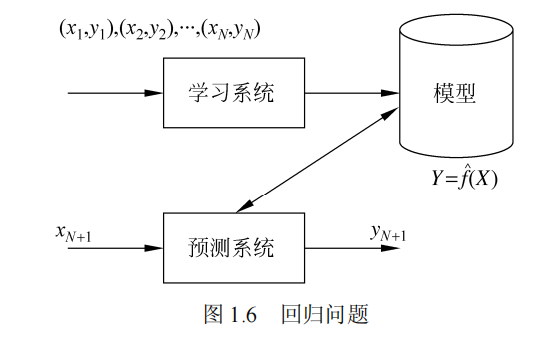


判别方法由数据直接学习决策函数 f ( X)或者条件概率分布 P(Y|X )作为预测的模型，即判别模型．判别方法关心的是对给定的输入X，应该预测什么样的输出Y。典型的判别模型包括：k近邻法、感知机、决策树、逻辑斯谛回归模型、最大熵模型、支持向量机、提升方法和条件随机场等

**学习第二天：**

**《分类问题和回归问题》**

分类是监督学习的一个核心问题．在监督学习中，当输出变量Y 取有限个离散 值时，预测问题便成为分类问题．这时，输入变量 X 可以是离散的，也可以是连续的。

分类问题的评价指标：精确率（precision）与召回率（recall）：

TP——将正类预测为正类数； FN——将正类预测为负类数；

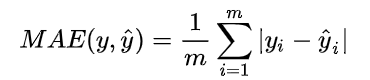
FP——将负类预测为正类数； TN——将负类预测为负类数．

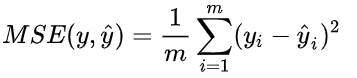
计算表达式为：精确率：P = TP/(TP+FP)，召回率：R= TP/(TP+FN)

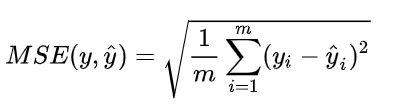
还有F1指标：F1 = 2TP/(2TP+TP+FN)。

常用的分类算法：感知机、朴素贝叶斯法、决策树、逻辑回归模型、支持向量机、GBDT等模型。

在拟合数据的时候，真实数据可能分布在预测值的两侧，所以一般我们使用分类问题的评价指标为：平均绝对误差 MAE,均方误差MSE和均方根误差：







回归问题可以由 著名的最小二乘法（least squares）求解． 许多领域的任务都可以形式化为回归问题，比如，回归可以用于商务领域，作为市场趋势预测、产品质量管理、客户满意度调查、投资风险分析的工具。

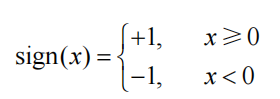
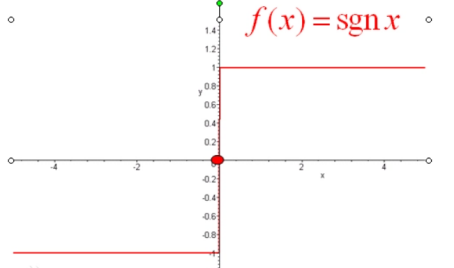
在我们的日常生活中最为常见的一个回归模型就是房价预测模型。

**学习第三天：**

**《感知机》**

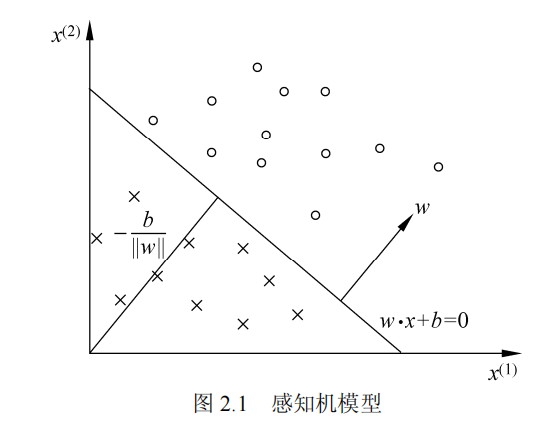
假设输入空间是，输出空间是  ．输入x表示实例的特征向量，对应于输入空间（也即特征空间）的点；输出y表示实例的真实类别．那么由输入空间到输出空间的函数：f(x)=sign(w\*x + b)成为感知机。

sign为符号函数，函数的具体形式如下左图所示，函数图像如后边所示：

感知机是一种线性分类模型，属于判别模型．

观察函数感知器内部函数可以得知，其中包含一个线性分类的模型。感知器通过构建一个分类超平面用于来完成分类的工作。



所以感知机的训练最为重要的是在上图中找寻到一个合适的分离超平面，这也是训练这个感知机最为关键的所在。这种训练也就是确定感知机模型参数 w , b ，需要确定一个学习策略，即定义（经验）损失函数并将损失函数极小化。

通过设置损失函数使用梯度下降算法来训练感知机算法是最为常见的方法。

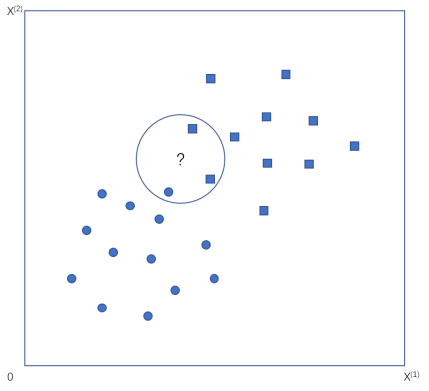
训练算法直观上有如下解释：当一个实例点被误分类，即位于分离超平面 的错误一侧时，则调整 w , b 的值，使分离超平面向该误分类点的一侧移动，以减少该误分类点与超平面间的距离，直至超平面越过该误分类点使其被正确分类。

**学习第四天：**

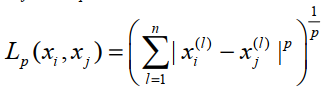
**《K近邻法》**

K近邻算法是一个很典型和常见的分类额回归算法。

k 近邻法假设给定一个训练数据集，其中的实例类别已定（即各个样本的标签是确定下来的）。分类时，对新的实例，根据其 k 个最近邻的训练实例的类别，通过多数表决等方式进行预测。有少数人服从多数人的那么一种感觉，因此k 近邻法不具有显式的学习过程。



在二分类的应用中，假设我们想判断一个样本是属于什么类别。那么我们要做的就是判断距离这个样本点最近距离的点的距离。所以这里就涉及到距离的计算：



存在几种特殊的请款：

1. 当p=1的时候，这种距离为哈曼度距离，即一种绝对值距离。
2. 当p=2的时候，这种距离为欧氏距离也是我们最为常见的距离。
3. 当p=∞的时候，这种距离为各个坐标距离的最大值距离。

K值选择：

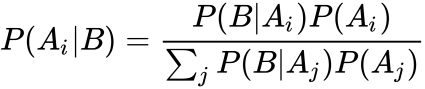
1. 过小的话学习到更多的知识，容易过拟合。对异常点变得十分的敏感
2. 过大的话模型整体变得简单，学习到的误差会变得很大。

**学习第五天：**

**《朴素贝叶斯法》**

朴素贝叶斯法是基于贝叶斯定理与特征条件独立假设的分类方法，是一种生成模型。

贝叶斯法则是关于随机事件A和B的[条件概率](https://baike.baidu.com/item/%E6%9D%A1%E4%BB%B6%E6%A6%82%E7%8E%87" \t "https://baike.baidu.com/item/%E8%B4%9D%E5%8F%B6%E6%96%AF%E5%85%AC%E5%BC%8F/_blank)和[边缘概率](https://baike.baidu.com/item/%E8%BE%B9%E7%BC%98%E6%A6%82%E7%8E%87" \t "https://baike.baidu.com/item/%E8%B4%9D%E5%8F%B6%E6%96%AF%E5%85%AC%E5%BC%8F/_blank)的：



其中P(A|B)是在B发生的情况下A发生的可能性。A1,A2...An是完备事件组。

P(A)是A的[先验概率](https://baike.baidu.com/item/%E5%85%88%E9%AA%8C%E6%A6%82%E7%8E%87" \t "https://baike.baidu.com/item/%E8%B4%9D%E5%8F%B6%E6%96%AF%E5%85%AC%E5%BC%8F/_blank)或边缘概率。之所以称为"先验"是因为它不考虑任何B方面的因素。

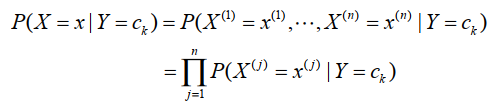
P(A|B)是已知B发生后A的[条件概率](https://baike.baidu.com/item/%E6%9D%A1%E4%BB%B6%E6%A6%82%E7%8E%87" \t "https://baike.baidu.com/item/%E8%B4%9D%E5%8F%B6%E6%96%AF%E5%85%AC%E5%BC%8F/_blank)，也由于得自B的取值而被称作A的[后验概率](https://baike.baidu.com/item/%E5%90%8E%E9%AA%8C%E6%A6%82%E7%8E%87" \t "https://baike.baidu.com/item/%E8%B4%9D%E5%8F%B6%E6%96%AF%E5%85%AC%E5%BC%8F/_blank)。

P(B|A)是已知A发生后B的条件概率，也由于得自A的取值而被称作B的后验概率。

P(B)是B的先验概率或边缘概率

对于给定的训练数据集，首先基于特征条件独立假设学习输入/输出的联合概率分布；然后基于此模型，对给定的输入x ，利用贝叶斯定理求出后验概率最大的输出y。

朴素贝叶斯法对条件概率分布作了条件独立性的假设：



这一假设使朴素贝叶斯法变得简单，但有时会牺牲一定的分类准确率。

朴素贝叶斯的参数估计方法：

1. 极大似然估计
2. 贝叶斯估计

**学习第六天：**

**《决策树》**

决策树是一种基本的分类与回归方法。

决策树模型呈树形结构，在分类问题中，表示基于特征对实例进行分类的过程．它可以认为是 if-then 规则的集合，也可以认为是定义在特征空间与类空间上的条件概率分布．

决策树分类优点：

1. 其主要优点是模型具有可读性，分类速度快
2. 可以进行一定的可视化，比较直观，可解释性比较强。

常用的决策树算法：

1. ID3：信息增益，递归构建决策树的思想。
2. C4.5：增益比
3. CART:基尼指数

**ID3算法**

具体步骤为：

1. 从根结点开始，对结点计算所有可能的特征的信息增益，选择信息增益最大的特征作为结点的特征。
2. 由该特征的不同取值建立子结点。
3. 再对子结点递归地调用以上方法，构建决策树。
4. 直到所有特征的信息增益均很小或没有特征可以选择为止。

C4.5 算法与 ID3 算法相似， C4.5 算法对 ID3 算法进行了改进． C4.5 在生成的过程中用信息增益比来选择特征。

**决策树的剪枝：**

ID3在进行特征点的分割的时候值不断递归的，直到所有的特征的信息增益均最小的时候。这就会造成过拟合，因此需要进行一定的策略调整来减少过拟合的可能性，提升决策树的性能。

**剪枝策略：**

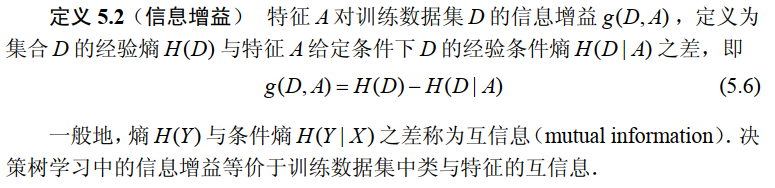
预剪枝：在决策树生成过程中，对每个节点在划分之前先进行估计，若当前节点的划分不能带来决策树泛化性能的提升，则停止划分，并将当前节点标记为叶节点。

后剪枝：先从训练集中生成一课完整的决策树，然后自底向上对非叶子节点进行考察，若将该节点对应的子树替换为叶子结点能带来决策树泛化性能的提升，则将该子树替换为叶节点。

**学习第七天：**

**《决策树算法对比》**

**一、ID3算法**



**缺点：**

（1）ID3没有考虑连续特征，比如长度，密度都是连续值，无法在ID3运用。这大大限制了ID3的用途。（2）对倾向于可取值数目较多的属性。（3）ID3算法对于缺失值的情况没有做考虑。（4）没有考虑过拟合的问题

1. **C4.5算法**

**缺点：**

（1）可能的算法效率降低。

（2）C4.5算法只适用于能够驻留于内存的数据集，当训练集大于内存容量时，程序则无法运行。（3）C4.5生成的是多叉树，即一个父节点可以有多个节点。多节点的计算效率低下。

**优点：**

1. 属性变量可以是连续型的
2. **CART算法**

基于Gini指数进行划分的一种决策树，即可以做回归也可以做分类。

所以有回归树的生成和分类树的生成两种方式。

回归树：最小二乘回归树算法

分类树：基于基尼指数的生成算法

改进了对于连续特征和离散特征的处理，提升了计算效率。



来源csdn博客文章