机器学习入门
A short course of machine learning

李德维
ldw@cumtb.edu.cn infhighdim.github.io

中国矿业大学(北京) 理学院

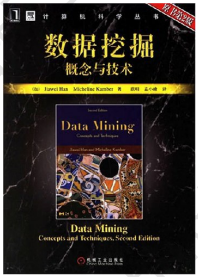
2023.06.14

课程提纲

课程教材

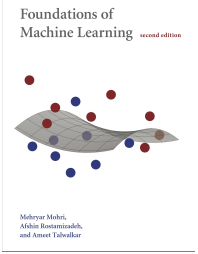
- a 机器学习, 周志华著. 北京: 清华大学出版社, 2016年1月.
- b 数据挖掘概念与技术, (加)Jiawei Han, Micheline Kamber. 机械工业出版社, 2007年3月.
- c 统计学习方法, 李航, 清华大学出版社, 2012年3月.
- d Foundations of Machine Learning. Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar MIT Press, Second Edition, 2018.(https://cs.nyu.edu/ mohri/mlbook/)

(a)



(b)

(c)



(d)

机器学习概述

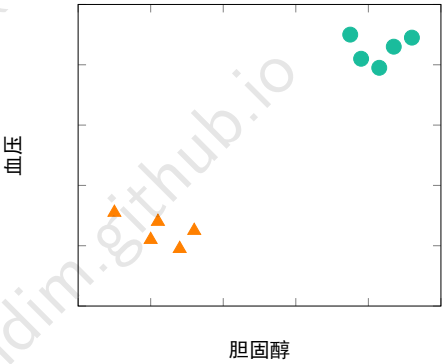
为什么要学机器学习？

色泽	根蒂	敲击声	好瓜
青绿	蜷缩	浊响	是
乌黑	蜷缩	浊响	是
青绿	硬挺	清脆	否
乌黑	稍蜷	沉闷	否

机器学习概述

为什么要学机器学习？

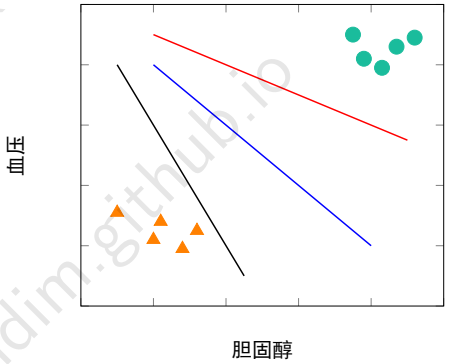
色泽	根蒂	敲击声	好瓜
青绿	蜷缩	浊响	是
乌黑	蜷缩	浊响	是
青绿	硬挺	清脆	否
乌黑	稍蜷	沉闷	否



机器学习概述

为什么要学机器学习？

色泽	根蒂	敲击声	好瓜
青绿	蜷缩	浊响	是
乌黑	蜷缩	浊响	是
青绿	硬挺	清脆	否
乌黑	稍蜷	沉闷	否



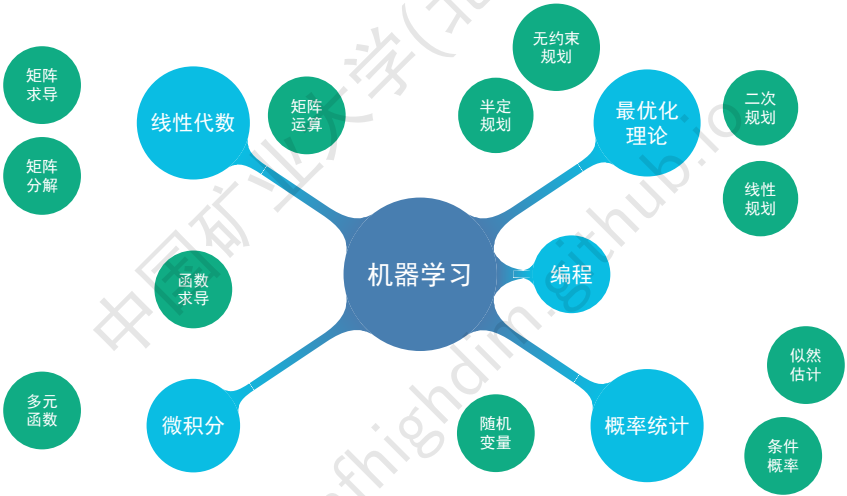
机器学习概述

(a) 人脸识别

(b) 游戏玩家匹配

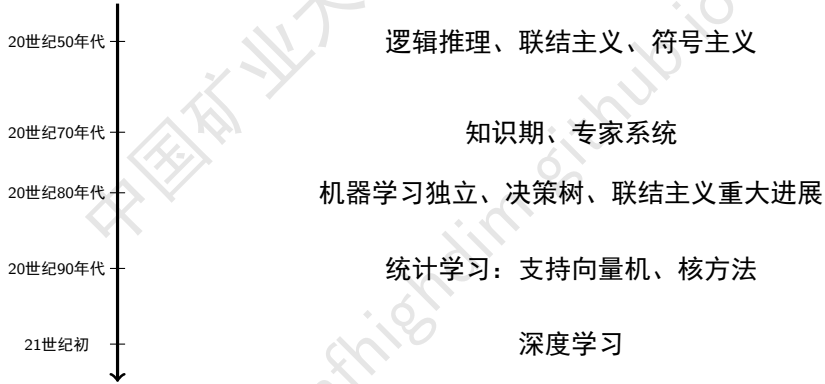
语音识别(微信语音锁)、新闻推荐(今日头条)、无人驾驶(特斯拉)、医学诊断、金融欺诈...

机器学习概述–知识基础

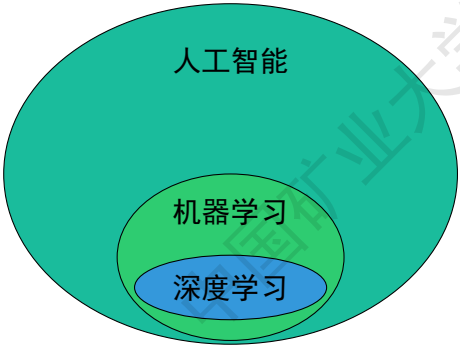


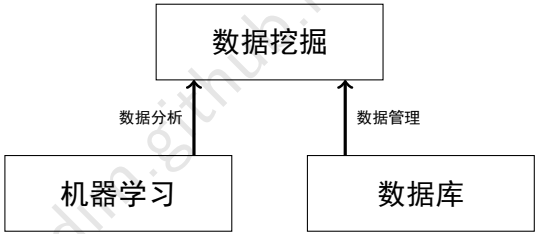
机器学习概述

机器学习 (Machine Learning) 是对研究问题进行模型假设，利用计算机从训练数据中学习得到模型参数，并最终对数据进行预测和分析的一门学科。机器学习涉及多个领域的理论交叉，包括最优化理论、概率论、统计学、逼近论、算法复杂度理论等。



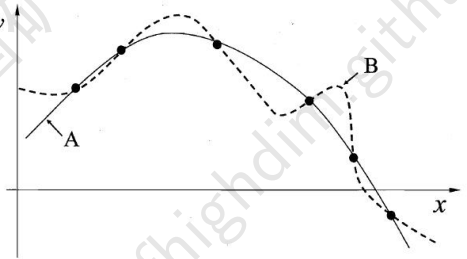
机器学习概述





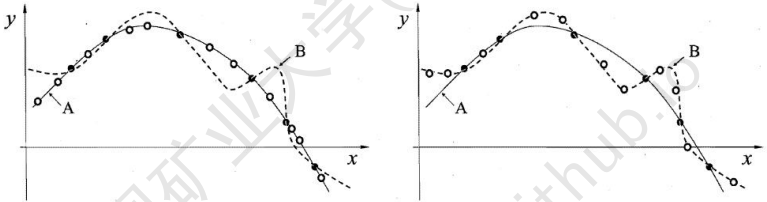
机器学习概述–基本概念

令数据集$D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\} \in R^{m \times n}$，它包含了$m$个数据样本，以及对应的标签信息$y$，每个样本的特征(属性)个数为$n$，也称为样本的维数。标签一般为标量，可以是连续或离散。数据集一般分为两部分，训练集和测试集。从训练数据中学得模型的过程称为"学习"(learning)或"训练"(training)，训练完成后利用测试集对模型进行评估。模型评估的目标是验证学习到的模型能否具有适用新数据的能力，这种能力称为泛化(generalization) 能力。



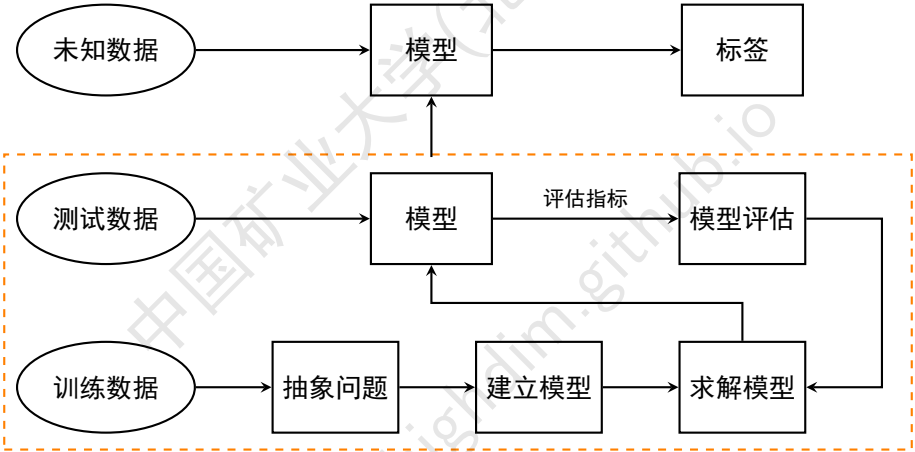
奥卡姆(Occam's razor)剃刀原理: 若有多个假设与观察一致，则选最简单的那个。

“没有免费的午餐”(No Free Lunch Theorem)

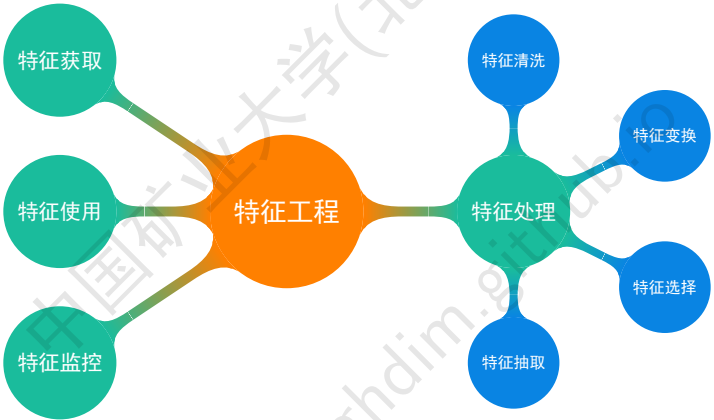


任何模型算法的性能和效果都与具体的问题相关，没有一个算法能在所有任务上都表现最优。

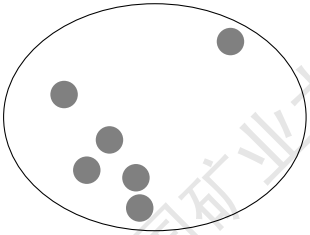
机器学习概述–学习步骤



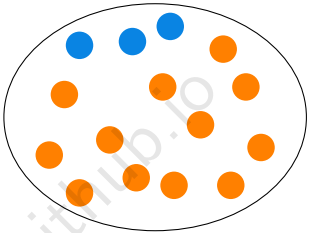
机器学习概述–特征工程



机器学习概述–特征清洗



(a) 离群点：剔除、平滑



(b) 不平衡：上下采样、数据增强

编号	特征1	特征2	特征3
1	5	20.1	3.6
2	4.3	9.5	5.9
3	5	20.1	3.6

(c) 重复值：保留、剔除

编号	特征1	特征2	特征3
1	5	20.1	
2	4.3		5.9
3	5	20.1	3.6

(d) 缺失值：剔除、人工填写、全局常量、属性平均值、同类属性平均值、算法推测

机器学习概述–特征处理

- 聚集：对数据集进行汇总和聚集。例如对于一份销售数据，可以考虑聚集每天的数据，计算月销售额或年销售额；
- 数据泛化: 用高层次概念替换原始数据。数值属性age，可以离散化成young, middle, senior等；
- 规范化：标准化、归一化。将数据按比例缩放，映射到一个标准区间内，比如$[0,1],[-1,1]$；(思考：为什么要做归一化？)

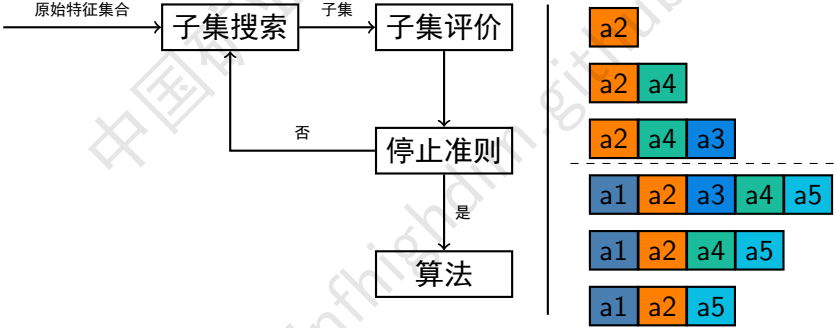
$$\text{max-min}: x' = \frac{x - min}{max - min}; z - score: x' = \frac{x - \hat{X}}{\sigma}; \text{小数定标}: x' = \frac{x}{10^j}$$

非线性归一化、批量归一化(BatchNormalization)

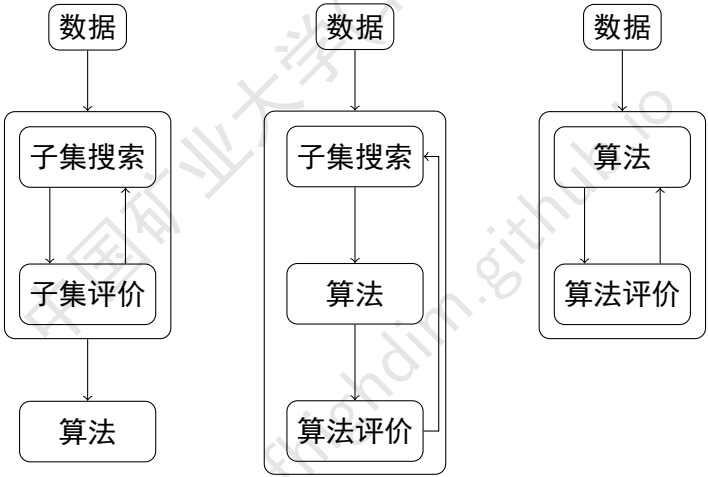
- 特征构造：例如在电商领域，用户行为数据表中每条记录为某个用户的一次浏览行为或一次点击行为，可以由此构造出用户最近一次浏览的时长、用户最近一次登录的点击次数等特征。

机器学习概述–特征选择

- **动机**：相关特征、无关特征
- **优点**：维数灾难–随着维数的增加，计算量呈指数倍增长的一种现象；降低学习任务的难度–更容易学习到好的模型
- **基本思路**：子集搜索–前向、后向、双向；子集评价



机器学习概述–特征选择



(a) 过滤式选择　(b) 包裹式选择　(c) 嵌入式选择

机器学习概述–特征选择

Relief (Relevant Features): 设计了一个"相关统计量"来度量特征的重要性
给定训练集，对于每个样本x_i，分别寻找空间中的同类近邻$x_{i,nh}$和异类近邻$x_{i,nm}$，那么使用如下方式计算特征j的重要性，

$$\delta^j = \sum_i -\text{diff}(x_i^j, x_{i,nh}^j)^2 + \text{diff}(x_i^j, x_{i,nm}^j)^2 \tag{1}$$

其中

$$\text{diff}(a,b) = \begin{cases} \mathbf{1}_{a=b}, & \text{离散} \\ |a-b|, & \text{连续} \end{cases} \tag{2}$$

对基于不同样本得到的估计结果进行平均, 就得到各属性的相关统计量分量, 分量值越大，则对应属性的分类能力就越强。(思考：为什么有效？)

机器学习概述–特征选择

LVW (Las Vegas Wrapper)：使用随机策略来进行子集搜索，并以最终分类器的误差为特征子集评价准则。

输入： 数据集 D;
特征集 A;
学习算法 $\mathfrak{L}$;
停止条件控制参数 T.

过程：

$E = \infty$;
$d = |A|$;
$A^* = A$;
$t = 0$;
while $t < T$ **do**
  随机产生特征子集 A';
  $d' = |A'|$;
  $E' = \text{CrossValidation}(\mathfrak{L}(D^{A'}))$;
  if $(E' < E) \vee ((E' = E) \wedge (d' < d))$ **then**
    $t = 0$;
    $E = E'$;
    $d = d'$;
    $A^* = A'$
  else
    $t = t + 1$
  end if
end while

输出： 特征子集 A^*

机器学习概述–特征选择

给定数据集D, 考虑线性回归模型

岭回归

$$\min_w \sum_{i=1}^{m}(y_i - w^\top x_i)^2 + \lambda||w||^2$$

LASSO

$$\min_w \sum_{i=1}^{m}(y_i - w^\top x_i)^2 + \lambda||w||$$

哪一种可以做特征选择？

机器学习概述–特征选择

给定数据集D, 考虑线性回归模型

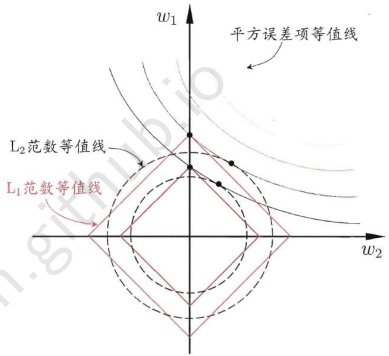
岭回归

$$\min_w \sum_{i=1}^{m}(y_i - w^\top x_i)^2 + \lambda||w||^2$$

LASSO

$$\min_w \sum_{i=1}^{m}(y_i - w^\top x_i)^2 + \lambda||w||$$

哪一种可以做特征选择？

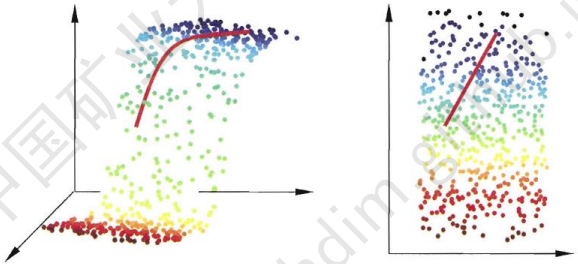


机器学习概述–特征抽取

降维：高维空间，低维映射，学习难度降低，学习效果不变甚至更好

学习映射矩阵W，获得

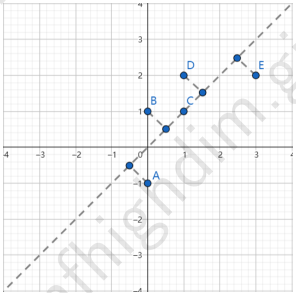
$$z = Wx$$



机器学习概述–特征抽取

主成分分析(Principal Component Analysis，简称PCA)：将n维原始特征映射到k维($k < n$)上，称这k维特征为主成分。其主要目标是将特征维度变小，同时尽量减少信息损失。
PCA将原始样本点投影到理想的超平面上：

- **最大可分性**: 样本点在这个超平面上的投影能尽可能分开;
- 最近重构性: 样本点到这个超平面的距离都足够近.



机器学习概述–特征抽取

下面从最大可分性角度推导。给定m个样本点$x'_1, x'_2, \cdots, x'_m$，首先进行中心化，令$x_i = x'_i - \frac{1}{m}\sum_{i=1}^{m} x'_i$，那么有

$$\mu_x = \sum_{i=1}^{m} x_i = 0 \tag{3}$$

为了将原始样本投影到低维空间，计算样本x_i与单位方向向量w的内积$y_i = x_i^\top w$，投影后的方差为

$$Var(y) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \mu_y)^2 = \frac{1}{m}\sum_{i=1}^{m}(y_i - \frac{1}{m}\sum_{i=1}^{m} x_i w)^2 = \frac{1}{m}\sum_{i=1}^{m} y_i^2 \tag{4}$$

$$Var(y) = \frac{1}{m}\sum_{i=1}^{m}(x_i^\top w)^2 = w^\top(\frac{1}{m}\sum_{i=1}^{m}x_i x_i^\top)w = w^\top \Sigma w \tag{5}$$

基于最大可分性，我们构建如下最优化问题

$$\max_{w} \quad w^\top \Sigma w \tag{6}$$

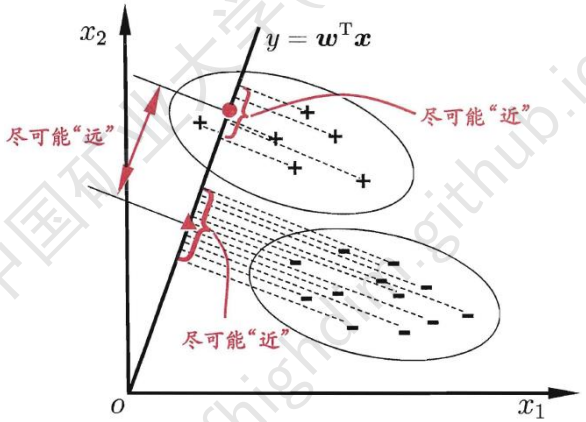
$$s.t. \quad w^\top w = 1 \tag{7}$$

对上式构建拉格朗日函数$L(w,\lambda) = w^\top \Sigma w + \lambda(1 - w^\top w)$，令$\nabla_w L = 0$可得

$$\Sigma w = \lambda w \tag{8}$$

将上式带入到方差可得$Var(y) = \lambda$.

线性判别分析(Linear Discriminant Analysis，简称LDA)



定义类内散度矩阵

$$S_w = \Sigma_0 + \Sigma_1 = \sum_{x \in X_0} (x - \mu_0)(x - \mu_0)^\top + \sum_{x \in X_1} (x - \mu_1)(x - \mu_1)^\top$$

和类间散度矩阵

$$S_b = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^\top$$

可以构建如下最优化问题

$$\max \frac{w^\top S_b w}{w^\top S_w w}$$

可以等价转化为如下最优化问题

$$\min_w \quad -w^\top S_b w \tag{9}$$

$$s.t. \quad w^\top S_w w = 1 \tag{10}$$

机器学习概述–特征抽取

利用拉格朗日乘子法，有

$$S_b w = \lambda S_w w$$

由于$S_b w$方向恒为$\mu_0 - \mu_1$，不妨令$S_b w = \lambda(\mu_0 - \mu_1)$, 那么有

$$w = S_w^{-1}(\mu_0 - \mu_1)$$

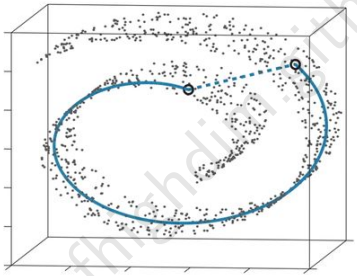
考虑到数值解的稳定性，一般会先对S_w进行奇异值分解，然后再求逆矩阵。
LDA也可以推广至多分类情形。

机器学习概述–特征抽取

- 多维标度分析(MDS): 降维前后能够保持距离关系不变

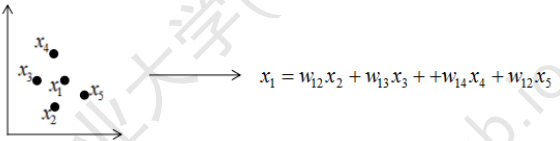
$$\min_{w} \sum_{ij} (||Wx_i - Wx_j|| - d(x_i, x_j))^2$$

- 等距特征映射ISOMAP：引入测地距离



机器学习概述–特征抽取

- 局部线性嵌入LLE：保持局部线性关系



- 拉普拉斯特征映射LE：基于图构建邻接矩阵，降维后仍能保持原有的数据结构信息

$$\min \sum_{ij} ||z_i - z_j||^2 w_{ij}$$

其中$w_{ij} = \exp(-\frac{||x_i - x_j||^2}{\sigma^2})$

机器学习概述–特征抽取

- t-分布随机近邻嵌入tsne：利用概率分布定义距离关系

$$p_{ij} = \frac{exp(-||x_i - x_j||^2)/2\sigma^2}{\sum_{k \neq l} exp(-||x_k - x_l||^2)/2\sigma^2}$$

$$q_{ij} = \frac{exp(-||z_i - z_j||^2)}{\sum_{k \neq l} exp(-||z_k - z_l||^2)}$$

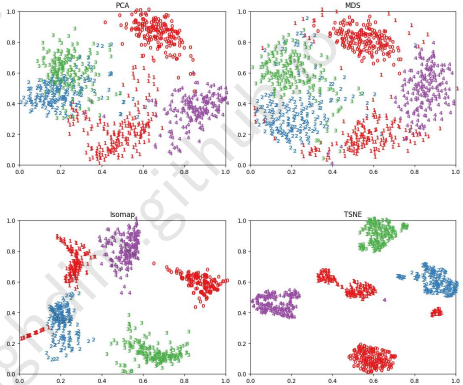
目标函数

$$C = KL(P||Q) = \sum_{ij} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

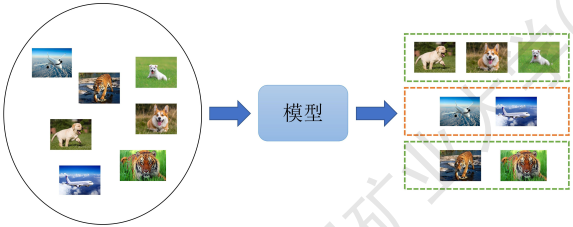
机器学习概述–特征抽取

手写字体集合

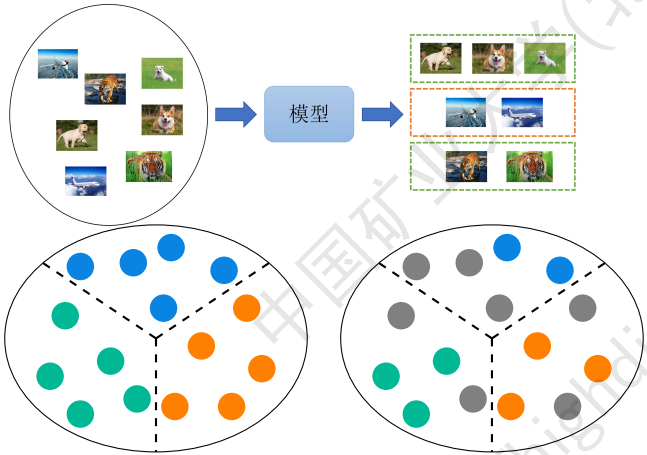
降维方法对比



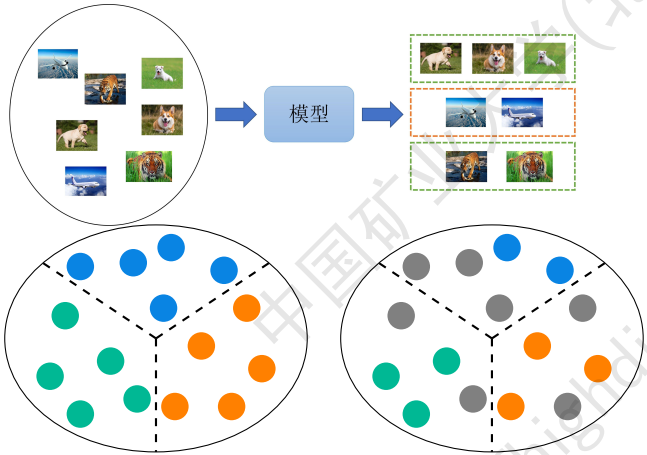
机器学习概述–分类问题



机器学习概述–分类问题

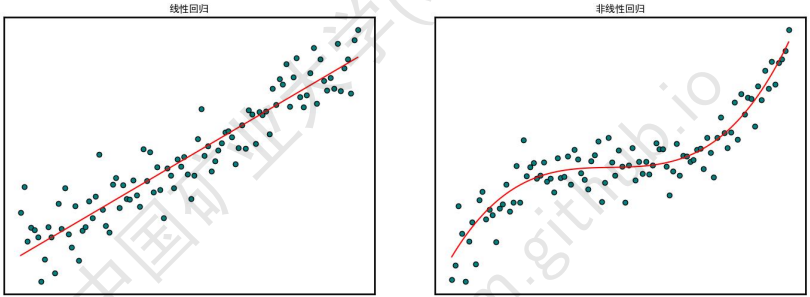


机器学习概述–分类问题



- 医疗诊断：患者生理指标，是否患病
- 金融欺诈：客户信息、还款历史，是否违约
- 人脸识别：面部信息，谁
- 市场营销：消费者行为，是否目标客户

机器学习概述–回归



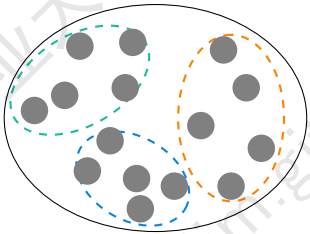
- 金融领域：历史股票价格，未来股票价格
- 医疗领域：患者信息、用药情况，血药浓度
- 人脸识别：面部信息，用户年龄

机器学习概述–回归问题评估指标

- MAE
- MSE
- RMSE
- MAPE
- R2 score
- Adjusted R2 score

机器学习概述–聚类

聚类学习是按照某种特定标准(如距离等)把一个数据集划分为不同的类或簇（子集），使得同一个簇内的数据对象的相似性尽可能大，不在同一个簇中的数据对象的差异性也尽可能地大（即聚类后同一类的数据尽可能聚集到一起，不同类数据尽量分离）。



- 社交网络：用户特征，社交圈子
- 电子商务：客户信息，分群
- 新闻分类：新闻文本，新闻类型

机器学习概述–聚类问题评估指标

评估标准：类内相似度高，类间相似度低

外部评价指标

- 纯度
- 归一化互信息(Normalized Mutual Information, NMI)
- 兰德指数(Rand index, RI)
- 调整兰德系数(Adjusted Rand index, ARI)
- R2 score
- Adjusted R2 score

内部评价指标

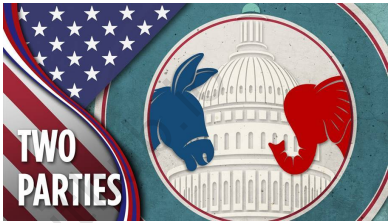
- 轮廓系数(Silhouette Coefficient)
- Calinski-Harabaz指数

机器学习概述–综合案例

2004年3月，在美国的自动驾驶车比赛，斯坦福大学机器学习专家S. Thrun的小组研制的参赛车用6小时53分钟成功走完了132英里赛程获得冠军。感知：语音识别、目标识别、物体追踪；预测：车辆行人的行为预测。

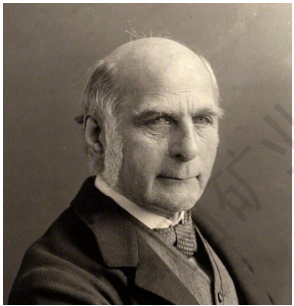
机器学习概述–综合案例

2004年3月，在美国的自动驾驶车比赛，斯坦福大学机器学习专家S. Thrun的小组研制的参赛车用6小时53分钟成功走完了132英里赛程获得冠军。感知：语音识别、目标识别、物体追踪；预测：车辆行人的行为预测。



2012美国大选，奥巴马麾下有一支机器学习团队，他们对各类选情数据进行分析，为奥巴马成功竞选提供了有力支持。涉及选民分类、选民的偏好画像。

经典算法–Logistic回归



- 1078对父、子身高的散点图
- 身材高大的父辈的孩子要矮些，而身材矮小的父辈的孩子要高些
- 遗传现象–身高趋于一般、“退化到平庸”

线性回归：
对于一个回归数据集$D = \{(x_i, y_i)\}_{i=1}^{m}$，线性回归旨在学习一个线性模型，通过对特征信息线性组合来预测y，即

$$\begin{aligned} & f(x) \\ = \quad & w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b \\ = \quad & w^\top x + b \end{aligned}$$

最小二乘法：

$$\min_{w,b} E = \sum_{i=1}^{m} (f(x_i) - y_i)^2$$

经典算法–Logistic回归

为了使表达形式更加简洁，将原始问题转换为向量形式。令$\hat{\boldsymbol{w}} = (w_1, \cdots, w_n, b)$, $\hat{\boldsymbol{x}}_i = (x_{i1}, \cdots, x_{in}, 1)$, 那么有

$$f(\boldsymbol{x}_i) = \boldsymbol{w}^\top \boldsymbol{x}_i + b = \hat{\boldsymbol{w}}^\top \hat{\boldsymbol{x}}_i{}^\top$$

令

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1^\top & 1 \\ \boldsymbol{x}_2^\top & 1 \\ \cdots & \\ \boldsymbol{x}_n^\top & 1 \end{pmatrix}$$

目标函数(思考：如何求解？)

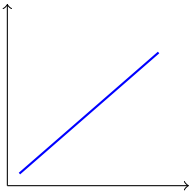
$$E = \sum_{i=1}^{m} (y_i - \hat{\boldsymbol{w}}^\top \hat{\boldsymbol{x}}_i{}^\top)^2 = (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}})^\top (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}})$$

经典算法–Logistic回归

分类问题：对于一个数据集$D = \{(x_i, y_i)\}_{i=1}^m$，标签$y$是离散、标量，机器学习旨在寻找一个决策函数$f(x)$，使得对任意$x$，都能预测出其对应的标签$y$。如果$y$有两种取值，则为二分类问题，多于两种取值则为多分类问题。

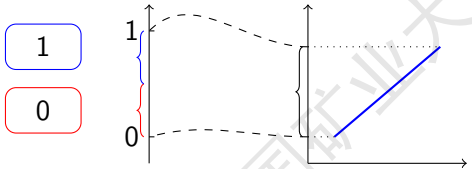
对于一个二分类数据集，即标签$y \in \{0, 1\}$，是否可以借鉴回归的思想？

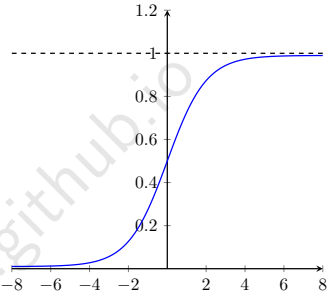
$$\left.\begin{matrix}0\\1\end{matrix}\right\} = y = \quad ? \quad = f(x) = w^\top x + b =$$



经典算法–Logistic回归

期望找到一个连续可微函数，将$f(x)$映射到$[0,1]$。





Sigmoid函数

$$S(x) = \frac{1}{1+e^{-x}}$$

经典算法–Logistic回归

使用如下函数来预测标签

$$f(x) = \frac{1}{1 + e^{-(w^\top x + b)}}$$

一个事件的几率（odds）是指该事件发生的概率与该事件不发生的概率的比值．如果事件发生的概率是p，那么该事件的几率是$\frac{p}{1-p}$。所以对数几率

$$\ln \frac{f(x)}{1 - f(x)} = w^\top x + b$$

对数几率回归–logistic regression

- 无需事先假设数据分布
- 近似概率预测
- Sigmoid函数任意阶可导

经典算法–Logistic回归

如果将$f(x)$看作是$y=1$的概率估计，那么有

$$\ln \frac{P(y=1|x)}{P(y=0|x)} = w^\top x + b$$

那么有

$$P(y=1|x) = \frac{e^{w^\top x+b}}{1+e^{w^\top x+b}} = \pi(x)$$

$$P(y=0|x) = \frac{1}{1+e^{w^\top x+b}} = 1-\pi(x)$$

利用极大似然法估计参数模型，似然函数为

$$\prod_{i=1}^{m} [\pi(x_i)]^{y_i} [1-\pi(x_i)]^{1-y_i}$$

经典算法–Logistic回归

目标函数

$$
\begin{aligned}
L &= -\sum_{i=1}^{m}[y_i \ln \pi(x_i) + (1-y_i)\ln(1-\pi(x_i))] \\
&= -\sum_{i=1}^{m}\left[y_i \ln \frac{\pi(x_i)}{1-\pi(x_i)} + \ln(1-\pi(x_i))\right] \\
&= -\sum_{i=1}^{m}[y_i(w^\top x_i + b) - \ln(1+\exp(w^\top x_i + b))] \\
&= \sum_{i=1}^{m}[-y_i(\hat{w}^\top \hat{x}_i) + \ln(1+\exp(\hat{w}^\top \hat{x}_i))]
\end{aligned}
$$

可以利用梯度下降法、牛顿法求解。

经典算法–Logistic回归

牛顿法

$$h'(x^{(k+1)}) = h'(x^{(k)}) + h''(x^{(k)})(x^{(k+1)} - x^{(k)}) = 0$$

本例中的迭代

$$\hat{w}^{(k+1)} = \hat{w}^{(k)} - (\frac{\partial^2 L}{\partial \hat{w} \partial \hat{w}^\top})^{-1} \frac{\partial L}{\partial \hat{w}}$$

其中

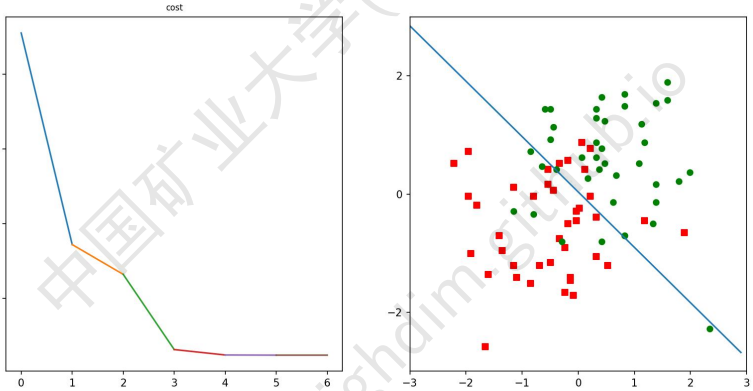
$$\frac{\partial L}{\partial \hat{w}} = -\sum_{i=1}^{m} \hat{x}_i (y_i - \pi(x_i))$$

$$\frac{\partial L}{\partial \hat{w} \partial \hat{w}^\top} = \sum_{i=1}^{m} \hat{x}_i \hat{x}_i^\top \pi(x_i)(1 - \pi(x_i))$$

(思考：如何推广到多分类？)

经典算法–Logistic回归

实例



机器学习概述–分类问题评估指标

对于一个二分类问题，定义如下指标

- 真正(True Positive, TP)：被模型预测为正的正样本；
- 假正(False Positive, FP)：被模型预测为正的负样本；
- 假负(False Negative, FN)：被模型预测为负的正样本；
- 真负(True Negative, TN)：被模型预测为负的负样本；

计算比例

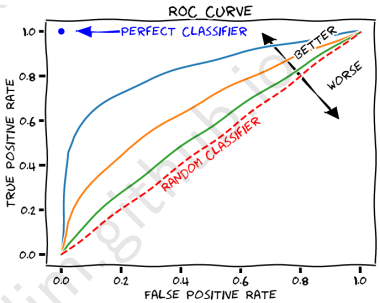
- 真正率$=\frac{TP}{TP+FN}$
- 假正率$=\frac{FP}{TN+FP}$
- 假负率$=\frac{FN}{TP+FN}$
- 真负率$=\frac{TN}{TN+FP}$

机器学习概述–分类问题评估指标

常见的指标

- 准确率(Accuracy)$=\frac{TP+TN}{TP+FP+FN+TN}$
- 精度(Precision)$=\frac{TP}{TP+FP}$
- 召回率(Recall)$=\frac{TP}{TP+FN}$
- $F1=\frac{2\times Precision\times Recall}{Precision+Recall}$
- $F_\beta=\frac{(1+\beta^2)\times Precision\times Recall}{\beta^2\times Precision+Recall}$
- P-R曲线

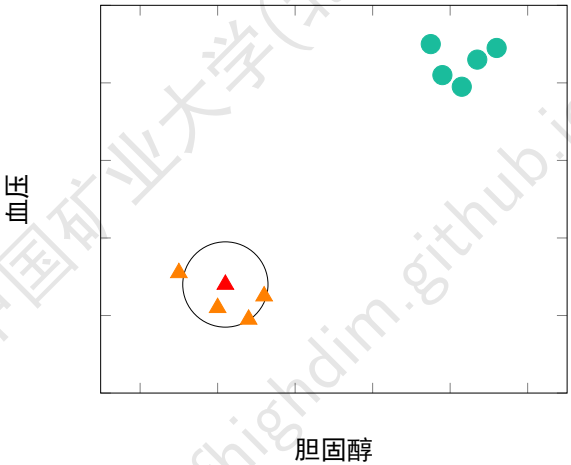
ROC(接受者操作特征曲线), AUC



排序关系

思考：为什么不用评估指标做损失函数？

经典算法–k近邻



经典算法–k近邻

k近邻: 给定测试样本, 基于某种距离度量找出训练集中与其最靠近的k个训练样本，然后基于这k个"邻居"的信息来进行预测。这k个样本的多数属于某个类，就把该样本分为这个类。

模型三要素：距离度量、k值和分类决策规则。

优点：

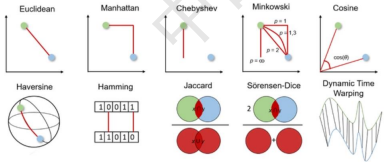
- 算法简单、直观
- 可用于分类和回归
- 更适用于类域的交叉或重叠较多的分类样本集

缺点：

- 时间复杂度和空间复杂度高
- 训练样本不平衡时，对稀有类别的预测准确率低

经典算法–k近邻

距离度量: 欧氏距离$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$、曼哈顿距离$d = \sum_{i=1}^{n}|x_i - y_i|$、切比雪夫距离$d = \max_i |x_i - y_i|$、闵可夫斯基距离$d = \sqrt[p]{\sum_{i=1}^{n}(x_i - y_i)^p}$、余弦距离



例子：已知二维空间的3个点, $x_1 = (1,1)$, $x_2 = (5,1)$, $x_3 = (4,4)$, 试求在p取不同值时，L_p 距离下x_1的最近邻点.

解:

$$L_p(x_1, x_2) = 4$$

$$L_1(x_1, x_3) = 6$$

$$L_2(x_1, x_3) = 4.24$$

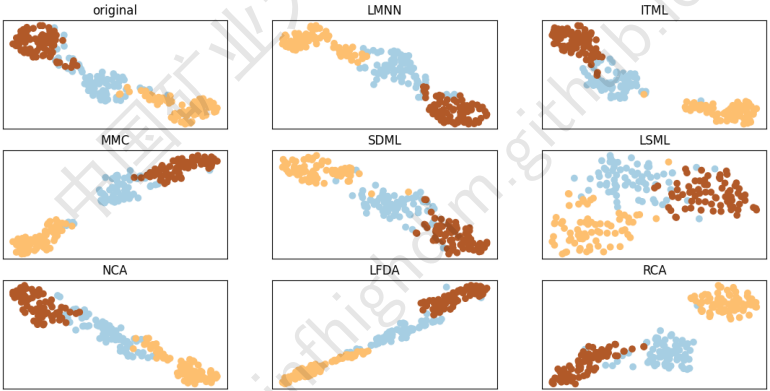
$$L_3(x_1, x_3) = 3.78$$

$$L_4(x_1, x_3) = 3.57$$

经典算法–k近邻

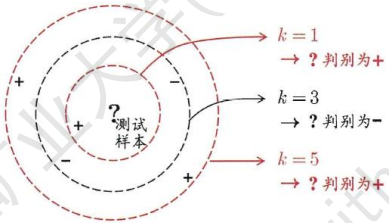
学习距离度量M，类内距离和类间距离分别为

$$d_S = \sum_{y_i=y_j} (x_i - x_j)^\top M(x_i - x_j), d_D = \sum_{y_i \neq y_j} (x_i - x_j)^\top M(x_i - x_j)$$



经典算法–k近邻

k值



近似误差、估计误差
k值大小与模型复杂度的关系

经典算法–k近邻

分类决策规则–多数表决：如果分类的损失函数为0-1损失函数，分类函数为

$$f : R^n \to \{c_1, c_2, \cdots, c_K\}$$

那么误分类的概率为

$$P(y \neq f(x)) = 1 - P(y = f(x))$$

对于给定的样本x，其最近邻的k个样本构成集合$N_k(x)$，如果涵盖$N_k(x)$的区域类别是c_j，那么误分类率是

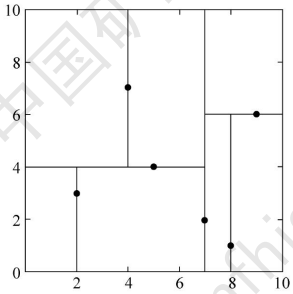
$$\frac{1}{k} \sum_{x_i \in N_k(x)} I(y_i \neq c_j) = 1 - \frac{1}{k} \sum_{x_i \in N_k(x)} I(y_i = c_j)$$

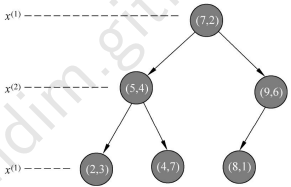
可以看到只有当$\frac{1}{k} \sum_{x_i \in N_k(x)} I(y_i = c_j)$最大时，才能使误分类率最小即经验风险最小。

经典算法–k近邻

kd树构建

- 对训练数据进行快速k近邻搜索
- 对k维空间中的实例点进行存储以便对其进行快速检索的树形数据结构.
- 不断地用垂直于坐标轴的超平面将k维空间切分，构成一系列的k维超矩形区域.





经典算法–k近邻

kd树搜索：

- 二叉树搜索比较待查询节点和分裂节点的分裂维的值，（小于等于就进入左子树分支，大于就进入右子树分支直到叶子结点）
- 顺着“搜索路径”找到最近邻的近似点
- 回溯搜索路径，并判断搜索路径上的结点的其他子结点空间中是否可能有距离查询点更近的数据点，如果有可能，则需要跳到其他子结点空间中去搜索
- 重复这个过程直到搜索路径为空

例子：查找点(2, 4.5)的最近邻点

- 搜索路径中的结点为(7,2),(5,4),(4,7), 最近邻为(4,7), 距离为3.202；
- 回溯至(5,4)，距离为3.04，更新为最近邻。以(2,4.5)为圆心，以3.202为半径画一个圆，它与超平面y=4相交，所以需要跳到(5,4)的左子空间去搜索。搜索路径中的结点为(7,2),(2, 3)；
- 回溯至(2,3)，(2,3)是叶子节点，距离为1.5，最近邻更新为(2,3)，距离更新为1.5；
- 回溯至(7,2)，距离大于1.5, 以(2,4.5)为圆心，以1.5为半径画圆, 不和超平面x=7相交, 搜索结束。

经典算法–朴素贝叶斯

机器学习所要实现的是基于有限的训练样本集尽可能准确地估计出后验概率$P(y|x)$. 大体来说，主要有两种策略:

- 给定x, 可通过直接建模$P(y|x)$来预测y，这样得到的是"判别式模型" (discriminative models);
- 先对联合概率分布$P(x,y)$建模，然后再由此获得$P(y|x)$，这样得到的是"生成式模型"(generative models)。

对于生成式模型，利用贝叶斯定理有

$$P(c|x) = \frac{P(x,c)}{P(x)} = \frac{P(c)P(x|c)}{P(x)}$$

$$P(Y = c_k|X = x) = \frac{P(X = x|Y = c_k)P(Y = c_k)}{\sum_k P(X = x|Y = c_k)P(Y = c_k)}$$

经典算法–朴素贝叶斯

条件概率分布$P(X=x|Y=c_k)$有指数级数量的参数，其估计实际是不可行的.
朴素贝叶斯–条件独立性的假设：

$$\begin{aligned} P(Y=c_k|X=x) &= P(X^{(1)}=x^{(1)},\cdots,X^{(n)}=x^{(n)}|Y=c_k) \\ &= \prod_{j=1}^{n} P(X^{(j)}=x^{(j)}|Y=c_k) \end{aligned}$$

那么后验概率计算公式为

$$P(Y=c_k|X=x)=\frac{P(Y=c_k)\prod_j P(X^{(j)}=x^{(j)}|Y=c_k)}{\sum_k P(Y=c_k)\prod_j P(X^{(j)}=x^{(j)}|Y=c_k)}$$

经典算法–朴素贝叶斯

后验概率最大化准则：假设选择0-1损失函数

$$L(Y, f(X)) = \begin{cases} 1, Y \neq f(X) \\ 0, Y = f(X) \end{cases}$$

条件期望风险

$$R_{exp}(f) = E\left[\sum_{k=1}^{K} P(c_k|X)L(c_k, f(X))\right]$$

为了使期望风险最小化

$$\begin{aligned} f(x) &= \arg\min_{y\in\mathcal{Y}} \sum_{k=1}^{K} P(c_k|X=x)L(c_k, y) \\ &= \arg\min_{y\in\mathcal{Y}} \sum_{k=1}^{K} P(y \neq c_k|X=x) = \arg\max_{y\in\mathcal{Y}} \sum_{k=1}^{K} P(y=c_k|X=x) \end{aligned}$$

经典算法–朴素贝叶斯

朴素贝叶斯利用极大似然估计来学习模型参数
首先估计先验概率

$$P(Y=c_k)=\frac{\sum_{i=1}^{m} I(y_i=c_k)}{m}$$

设第j个特征的可能取值集合为$\{a_{j1},a_{j2},\cdots,a_{jS_j}\}$，那么条件概率的估计为

$$P(X^{(j)}=a_{jl}|Y=c_k)=\frac{\sum_{i=1}^{m} I(x_i^{(j)}=a_{jl},y_i=c_k)}{\sum_{i=1}^{m} I(y_i=c_k)}$$

经典算法–朴素贝叶斯

利用如下数据学习一个朴素贝叶斯分类器并确定$x=(2,S)$的标签y。

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$X^{(1)}$	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3
$X^{(2)}$	S	M	M	S	S	S	M	M	L	L	L	M	M	L	L
Y	–1	–1	1	1	–1	–1	–1	1	1	1	1	1	1	1	–1

解：

$$P(Y=1), P(Y=-1), P(X^{(1)}=1|Y=1), P(X^{(1)}=2|Y=1), P(X^{(1)}=3|Y=1)$$

$$P(X^{(2)}=S|Y=1), P(X^{(2)}=M|Y=1), P(X^{(2)}=L|Y=1)$$

$$P(X^{(1)}=1|Y=-1), P(X^{(1)}=2|Y=-1), P(X^{(1)}=3|Y=-1)$$

$$P(X^{(2)}=S|Y=-1), P(X^{(2)}=M|Y=-1), P(X^{(2)}=L|Y=-1)$$

经典算法–朴素贝叶斯

特殊情况下可能会出现某些要估计的概率为0，考虑使用拉普拉斯平滑。

$$P(X^{(j)} = a_{jl}|Y = c_k) = \frac{\sum_{i=1}^{m} I(x_i^{(j)} = a_{jl}, y_i = c_k) + \lambda}{\sum_{i=1}^{m} I(y_i = c_k) + S_j\lambda}$$

$$P(Y = c_k) = \frac{\sum_{i=1}^{m} I(y_i = c_k) + \lambda}{m + k\lambda}$$

经典算法–朴素贝叶斯

实例: 基于上个例子的数据，使用拉普拉斯平滑学习一个朴素贝叶斯分类器

$$P(Y=1)=\frac{10}{17},\quad P(Y=-1)=\frac{7}{17}$$

$$P(X^{(1)}=1\,|\,Y=1)=\frac{3}{12},\quad P(X^{(1)}=2\,|\,Y=1)=\frac{4}{12},\quad P(X^{(1)}=3\,|\,Y=1)=\frac{5}{12}$$

$$P(X^{(2)}=S\,|\,Y=1)=\frac{2}{12},\quad P(X^{(2)}=M\,|\,Y=1)=\frac{5}{12},\quad P(X^{(2)}=L\,|\,Y=1)=\frac{5}{12}$$

$$P(X^{(1)}=1\,|\,Y=-1)=\frac{4}{9},\quad P(X^{(1)}=2\,|\,Y=-1)=\frac{3}{9},\quad P(X^{(1)}=3\,|\,Y=-1)=\frac{2}{9}$$

$$P(X^{(2)}=S\,|\,Y=-1)=\frac{4}{9},\quad P(X^{(2)}=M\,|\,Y=-1)=\frac{3}{9},\quad P(X^{(2)}=L\,|\,Y=-1)=\frac{2}{9}$$

对于给定的 $x=(2,S)^{\mathrm{T}}$ 计算：

$$P(Y=1)P(X^{(1)}=2\,|\,Y=1)P(X^{(2)}=S\,|\,Y=1)=\frac{10}{17}\cdot\frac{4}{12}\cdot\frac{2}{12}=\frac{5}{153}=0.0327$$

$$P(Y=-1)P(X^{(1)}=2\,|\,Y=-1)P(X^{(2)}=S\,|\,Y=-1)=\frac{7}{17}\cdot\frac{3}{9}\cdot\frac{4}{9}=\frac{28}{459}=0.0610$$

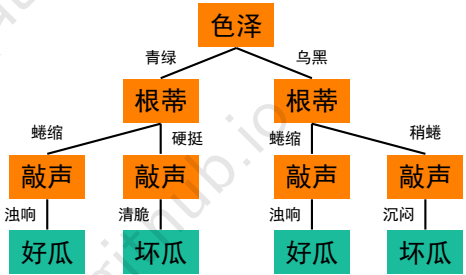
由于 $P(Y=-1)P(X^{(1)}=2\,|\,Y=-1)P(X^{(2)}=S\,|\,Y=-1)$ 最大，所以 $y=-1$.

决策树

色泽	根蒂	敲击声	好瓜
青绿	蜷缩	浊响	是
乌黑	蜷缩	浊响	是
青绿	硬挺	清脆	否
乌黑	稍蜷	沉闷	否

决策树

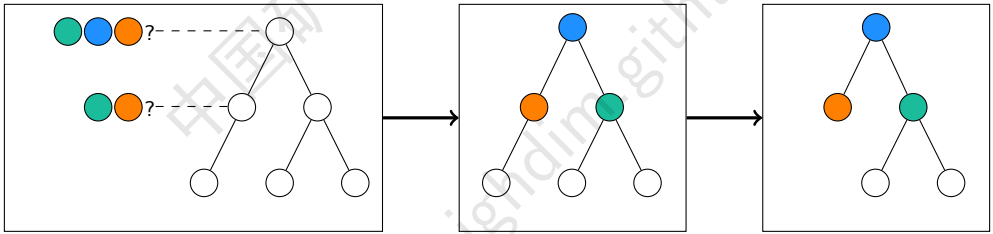
色泽	根蒂	敲击声	好瓜
青绿	蜷缩	浊响	是
乌黑	蜷缩	浊响	是
青绿	硬挺	清脆	否
乌黑	稍蜷	沉闷	否



决策树

定义：分类决策树模型是一种描述对实例进行分类的树形结构。决策树由结点（node）和有向边（directed edge）组成。结点有两种类型：内部结点（internal node）和叶结点（leaf node）. 内部结点表示一个特征或属性，叶结点表示一个类。

算法步骤：特征选择、决策树的生成和决策树的修剪。



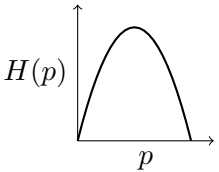
决策树

熵：随机变量不确定性。设X是一个取有限个值的离散随机变量，其概率分布为$P(X=x_i)=p_i$，则随机变量X的熵定义为

$$H(X)=H(p)=-\sum_{i=1}^{n}p_i\log p_i$$

熵越大，随机变量的不确定性就越大。

熵$H(p)$随概率p变化的曲线



条件熵：$H(Y|X)$在已知随机变量X的条件下随机变量Y的不确定性。X给定条件下Y的条件概率分布的熵对X的数学期望

$$H(Y|X)=\sum_{i=1}^{n}p_iH(Y|X=x_i)$$

当熵和条件熵中的概率由数据估计(特别是极大似然估计)得到时，所对应的熵与条件熵分别称为**经验熵**(empirical entropy)和**经验条件熵**(empirical conditional entropy)。

决策树

信息增益(information gain): 得知特征Y的信息而使得类Y的信息的不确定性减少的程度。特征A对训练数据集D的信息增益$g(D,A)$，定义为集合D的经验熵$H(D)$与特征A给定条件下D的经验条件熵$H(D|A)$之差

$$g(D,A) = H(D|A) - H(D)$$

- 经验熵$H(D)$表示对数据集D进行分类的不确定性;
- 经验条件熵$H(D|A)$表示在特征A给定的条件下对数据集D进行分类的不确定性;
- 特征A使得对数据集D的分类的不确定性减少的程度。

信息增益比: 在训练数据集的经验熵大的时候，信息增益值会偏大。

$$g_R(D,A) = \frac{g(D,A)}{H(D)}$$

决策树

特征选择：决策树的分支结点所包含的样本尽可能属于同一类别，即结点的"纯度"(purity)越来越高。

利用信息增益来做特征选择：首先计算数据集的经验熵

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} \log_2 \frac{|C_k|}{|D|}$$

然后计算特征A对数据集D的经验条件熵

$$H(D|A) = \sum_{i=1}^{m} \frac{|D_i|}{|D|} H(D_i) = -\sum_{i=1}^{m} \frac{|D_i|}{|D|} \sum_{k=1}^{K} \frac{|D_{ik}|}{|D_i|} \log_2 \frac{|D_{ik}|}{|D_i|}$$

决策树

特征选择–信贷数据集

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

$$H(D) = -\frac{9}{15}\log_2\frac{9}{15} - \frac{6}{15}\log_2\frac{6}{15} = 0.971$$

分别以A_1, A_2, A_3, A_4表示年龄、有工作、有自己的房子和信贷情况4个特征

$$\begin{aligned} g(D, A_1) &= H(D) - \left[\frac{5}{15}H(D_1) + \frac{5}{15}H(D_2) + \frac{5}{15}H(D_3)\right] \\ &= 0.083 \end{aligned}$$

$$g(D, A_2) = H(D) - \left[\frac{5}{15}H(D_1) + \frac{10}{15}H(D_2)\right]$$

$$= 0.324$$

$$g(D, A_3) = 0.420$$

$$g(D, A_4) = 0.363$$

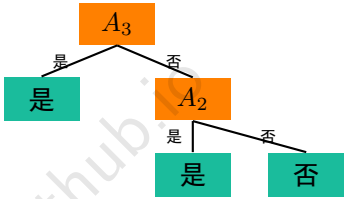
决策树

ID3：在决策树各个结点上应用信息增益准则选择特征，递归地构建决策树。

- 特征A_3(有自己的房子)的信息增益值最大，将训练数据集D划分为两个子集D_1(A_3取值为“是”)和D_2(A_3取值为“否”)；
- 对D_2则需从特征A_1（年龄），A_2（有工作）和A_4（信贷情况）中选择新的特征．分别计算信息增益

$$g(D_2, A_1) = 0.251, g(D_2, A_2) = 0.918, g(D_2, A_4) = 0.474$$

- A_2将训练数据集D划分为两个子集

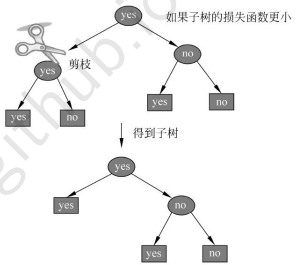


C4.5：使用信息增益比来选择特征。

决策树

过拟合(overfitting)：对训练数据的分类很准确，但对未知的测试数据的分类却没有那么准确。

剪枝(pruning): 极小化决策树整体的损失函数(loss function)或代价函数(cost function)



- 预剪枝(prepruning): 生成过程中剪枝
- 后剪枝(postpruning)：生成完成后剪枝

决策树–CART分类树

基尼指数(Breiman, 1984): 分类问题中，假设有K个类，样本点属于第k类的概率为p_k, 则概率分布的基尼指数定义为

$$Gini(p) = \sum_{k=1}^{K} p_k(1-p_k) = 1 - \sum_{k=1}^{K} p_k^2$$

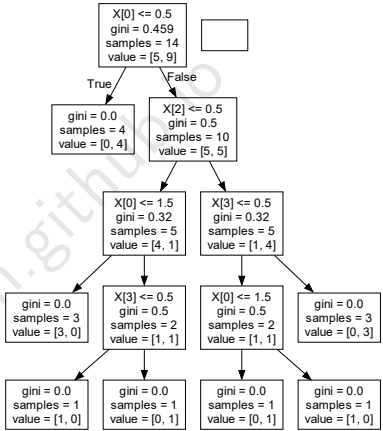
基尼指数$Gini(D)$表示集合D的不确定性，基尼指数$Gini(D,A)$表示经$A=a$分割后集合D的不确定性。基尼指数值越大，样本集合的不确定性也就越大。

算法步骤：

- 计算现有特征对该数据集的基尼指数;
- 选择基尼指数最小的特征及其对应的切分点作为最优特征与最优切分点;
- 依次递归，生成二叉决策树。

决策树–CART分类树

年龄	收入	学生	信用	买电脑
$<= 30$	高	否	一般	否
$<= 30$	高	否	好	否
$31 - 40$	高	否	一般	是
> 40	中	否	一般	是
> 40	低	是	一般	是
> 40	低	是	好	否
$31 - 40$	低	是	好	是
$<= 30$	中	否	一般	否
$<= 30$	低	是	一般	是
> 40	中	是	一般	是
$<= 30$	中	是	好	是
$31 - 40$	中	否	好	是
$31 - 40$	高	是	一般	是
> 40	中	否	好	否



决策树–CART回归树

对于回归问题，构建回归树，基于平方误差最小化的原则寻找最优切分变量和切分点。
(1)对于第j个变量$x^{(j)}$和取值s，定义两个区域

$$R_1(j,s)=\{x|x^{(j)\leq s}\}, R_2(j,s)=\{x|x^{(j)\geq s}\}$$

(2)然后通过求解如下最优化问题来获得(j,s)

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right]$$

(3)计算输出值

$$\hat{c}_m=\frac{1}{N_m}\sum_{x_i\in R_m(j,s)}y_i, m=1,2$$

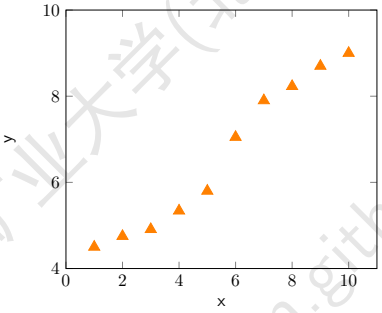
决策树–CART回归树

(4)重复如上步骤，直至满足停止条件(树的深度、叶子结点数量)；
(5) 将输入空间划分为M个区域$R_1, R_2, \cdots, R_M$, 最终决策树为

$$f(x) = \sum_{m=1}^{M} \hat{c}_m I(x \in R_m)$$

最小二乘回归树(least squares regression tree)

决策树–CART回归树

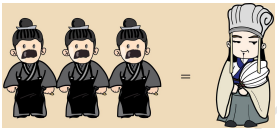


步骤：对特征x，取$s=1$，先求两个区域，再分别求得区域均值，然后计算方差，依次计算$s=2,\cdots,10$

s	1	2	3	4	5	6	7	8	9	10
c_1	4.50	4.63	4.72	4.88	5.06	5.39	5.75	5.18	6.35	6.62
c_2	6.85	7.12	7.43	7.78	8.18	8.46	8.64	8.85	9.00	0
v(s)	22.65	17.7	12.19	7.38	3.36	5.07	10.05	15.18	21.33	27.63

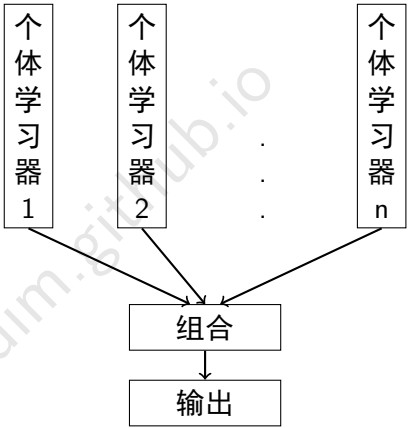
随机森林

集成学习：构建并结合多个学习器来完成学习任务



基本步骤

- 学习若干个不同的个体学习器
- 利用某种策略将多个学习器进行组合



随机森林

集成学习的效果

	测试例$_1$	测试例$_2$	测试例$_3$
h_1	√	√	×
h_2	×	√	√
h_3	√	×	√
集成	√	√	√

(a) 集成提升性能

	测试例$_1$	测试例$_2$	测试例$_3$
h_1	√	√	×
h_2	√	√	×
h_3	√	√	×
集成	√	√	×

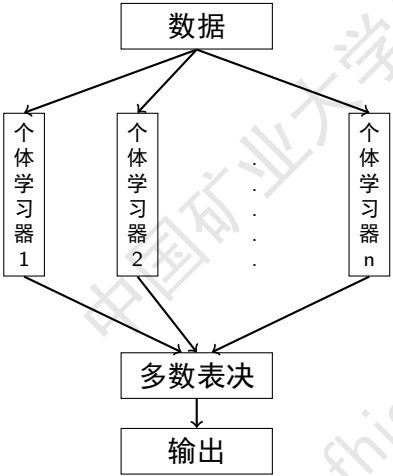
(b) 集成不起作用

	测试例$_1$	测试例$_2$	测试例$_3$
h_1	√	×	×
h_2	×	√	×
h_3	×	×	√
集成	×	×	×

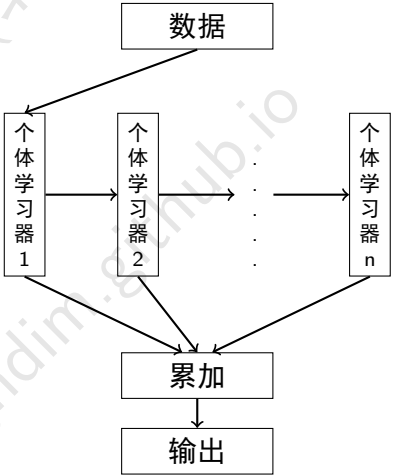
(c) 集成起负作用

个体学习器应该好而不同：既要有准确性又要有多样性

随机森林



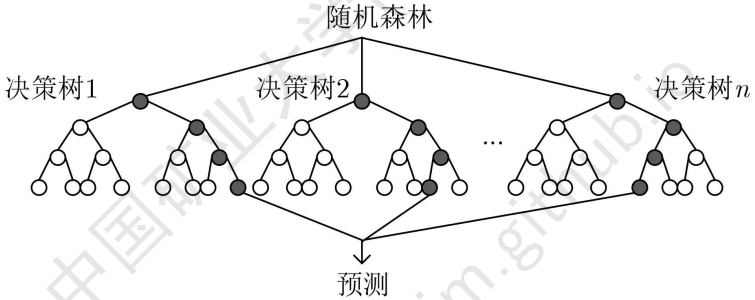
(r) bagging



(s) boosting

随机森林

随机森林：一种由多个决策树构成的集成算法，不同决策树之间没有关联



随机性: 样本随机、特征随机

- 样本：自助采样法(bootstrap sampling);
- 特征：从特征集合中随机选取k个特征.

随机森林

优点

- 采用了集成算法，精度优于大多数单模型算法;
- 样本和特征随机性的引入降低了过拟合风险;
- 训练过程中能检测特征重要性，是常见的特征筛选方法;
- 每棵树可以同时生成，并行效率高，训练速度快.

缺点

- 在某些噪音较大的分类或回归问题上会过拟合;
- 对于有不同取值的属性的数据，取值划分较多的属性会对随机森林产生更大的影响.

梯度提升树

对于一般的boosting算法，以树作为基分类器，约定$f_t(x)$表示第t轮的模型，$h_t(x)$表示第t轮学习得到的决策树，迭代步骤

$$f_t(x) = f_{t-1}(x) + h_t(x)$$

损失函数定义为

$$L(f_t(x), y) = L(f_{t-1}(x) + h_t(x), y)$$

最终的模型可以表示为

$$f_t(x) = \sum_{t=1}^{T} h_t(x)$$

思考：如何确定$h_t(x)$？

梯度提升树

Freidman提出了用损失函数的负梯度来拟合本轮损失的近似值，进而拟合一个CART回归树。

$$r_{t,i} = -\frac{\partial L(f(x_i), y)}{\partial f(x_i)}$$

在第t轮迭代中，要拟合的数据为$(x_i, r_{t,i})$，获得第t棵回归树$h_t(x)$

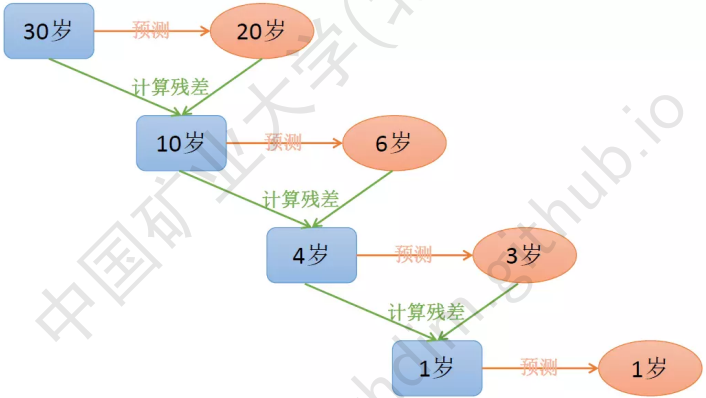
$$h_t(x) = \sum_{j=1}^{J} c_{t,j} I(x \in R_{t,j})$$

本轮的模型为

$$f_t(x) = f_{t-1}(x) + \sum_{j=1}^{J} c_{t,j} I(x \in R_{t,j})$$

定义: 以CART回归树作为基学习器的boosting算法。

梯度提升树



梯度提升树

优点

- 采用基于负梯度的Boosting集成手段
- 适用于回归和分类任务
- 预测精度比随机森林高
- 对异常值的鲁棒性强(采用了Huber损失和分位数损失)

缺点

- 串行方式的模型训练，难并行，造成计算开销
- 不适合高维稀疏离散特征

梯度提升树

随机森林与梯度提升树对比

区别点	随机森林	GBDT
集成方式	bagging	boosting
决策树类型	分类树、回归树	CART回归树
结合方式	并行生成	顺序生成
优化指标	方差优化	残差优化
训练样本	有放回抽样	全样本
表决方式	多数表决	累加之和

Xgboost

Xgboost(eXtreme Gradient Boost)是分布式梯度提升决策树 (GBDT) 机器学习库

与gbdt区别

- 二阶导数信息
- 特征采样
- 分裂节点寻找的近似算法
- Shrinkage思想

相关资料：
https://xgboost.readthedocs.io/en/stable/
import xgboost as xgb
xgb.XGBClassifier()
xgb.XGBRegressor()

实例

谢谢观看!