# 1 概述

1. 分类：有无监督、能否增量、基于实例/模型
2. 基本步骤：收集数据-输入数据-数据预处理-训练和测试模型-模型的评估
3. 特征缩放：标准化、归一化
4. 模型评估：混淆矩阵、准确率、精确率、召回率、F1指数
5. 交叉验证

# 2 KNN

1. 距离度量：

   $p=1$，曼哈顿距离

   $p=2$，欧氏距离

   $p=\infty$，切比雪夫距离。各个坐标距离的最大值。原式会变为$\max\limits_{l}|x_i^{(l)}-x_j^{(l)}|$

2. k值选择：

   K值小：单个样本的影响大

   - 优点：近似误差（approximation error）减小
     - 只有与输入实例较近的训练实例才会对预测结果起作用
   - 缺点：估计误差（estimation error）增大
     - 预测结果会对近邻的实例点非常敏感（易受噪声影响）

   K值大：单个样本的影响小

   - 优点：估计误差减小
   - 缺点：近似误差增大

3. KD树：KD树的构造、KD树的搜索

# 朴素贝叶斯

1. 贝叶斯原理

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

   - 先验概率：$P(Y)$
   - 后验概率：$P(Y|X)$
   - 似然度：$P(X|Y)$
   - 边际似然度：$P(X)$
   - 可能性函数：$\frac{P(X|Y)}{P(X)}$
   - 方法：极大似然估计

2. 朴素：假设X,Y独立同分布
3. 预测：$y=\arg\max\limits_{i}P(Y=i)\prod_j P(X^{(j)}=x^{(j)}|Y=i)$
4. 贝叶斯估计：原来的分数上，分子+1，分母+种类的个数

# 线性回归

1. 损失函数：**单样本**预测的错误程度
2. 代价函数：度量**全部样本集**的平均误差
3. 目标函数：代价函数和正则化函数，最终要优化的函数
4. 梯度下降法：

   $J(w)=\frac{1}{2}\sum_{i=1}^{N}(\sum_{k=0}^{n}w_k\cdot x_i^{(k)}-y_i)$

   $\frac{\partial}{\partial w_j}J(w)=\sum_{i=1}^{N}(f(x_i)-y_i)\cdot x^{(j)}$
5. 正规方程：

   $w=(X^TX)^{-1}X^Ty$
6. 正则化：

   L1：套索回归：$\lambda|w|$。稀疏、特征选择。

   L2：岭回归：$\lambda w^2$。抗干扰，适应极端条件。

   模型越复杂，正则化值就越大。

# 感知机

1. 损失函数：函数间隔$\hat{\gamma}_i=y_i(w\cdot x_i+b)$
2. 随机梯度下降：

   $\nabla_w L(w,b)=-y_ix_y$

   $\nabla_b L(w,b)=-y_i$
3. 对偶

# 逻辑回归

1. 模型：$z=\sigma(f(x))=\sigma(w\cdot x+b)=\frac{1}{1+e^{-(w\cdot x+b)}}$
2. 参数估计（极大似然估计）：$\frac{\partial J(w_j)}{\partial w_j}=-\sum_{i=1}^{N}(y_i-\pi(x_i))\cdot x_i^{(j)}$
3. 熵：$H(X)=-\sum_{i=1}^{n}p(x_i)\log(p(x_i))$
4. 交叉熵：$loss=-\sum_{i=1}^{n}y_i\log(\hat{y}_i)$
5. 多类别分类：

- 可以直接处理多个类别：随机森林、朴素贝叶斯
- 严格的二分类器：逻辑回归、感知机、支持向量机
- 拆分策略：
    - 一对其余(OvR)或一对全部(OvA)
    - 一对一(OvO)

# 决策树

1. 条件熵：$H(Y|X)=\sum_{i=1}^{n}p_iH(Y|X=x_i),p_i=P(X=x_i)$
2. 信息增益：总信息熵-条件熵
3. 信息增益比：$g_R(D,A)=\frac{g(D,A)}{H_A(D)}$
4. 基尼系数：

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

$$Gini(D,A)=p_1Gini(D_1)+p_2Gini(D_2)$$

5.

| 算法 | 支持模型 | 树结构 | 特征选择 | 连续值处理 | 缺失值处理 | 剪枝 | 特征属性多次使用 |
|---|---|---|---|---|---|---|---|
| ID3 | 分类 | 多叉树 | 信息增益 | 不支持 | 不支持 | 不支持 | 不支持 |
| C4.5 | 分类 | 多叉树 | 信息增益率 | 支持 | 支持 | 支持 | 不支持 |
| CART | 分类<br>回归 | 二叉树 | 基尼指数<br>均方差 | 支持 | 支持 | 支持 | 支持 |

# SVM

1. 支持向量：离分离超平面最近的点
2. 间隔最大化：支持向量离分离超平面的距离最远
3. 距离度量：几何间隔
4. $$\min_{w,b}\frac{1}{2}||w||^2$$
$$s.t.y_i(w\cdot x_i+b)\geq 1,i=1,2,\cdots,N$$
5. 核函数：解决低维度线性不可分问题

| 名称 | 表达式 | 参数 |
|---|---|---|
| 线性核 | $\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\boldsymbol{x}_i^\top\boldsymbol{x}_j$ | |
| 多项式核 | $\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=(\boldsymbol{x}_i^\top\boldsymbol{x}_j)^d$ | $d\geq 1$为多项式的次数 |
| 高斯核 | $\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp\left(-\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{2\delta^2}\right)$ | $\delta>0$为高斯核的带宽(width) |
| 拉普拉斯核 | $\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp\left(-\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|}{\delta}\right)$ | $\delta>0$ |
| Sigmoid核 | $\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\tanh(\beta\boldsymbol{x}_i^\top\boldsymbol{x}_j+\theta)$ | tanh为双曲正切函数, $\beta>0,\theta<0$ |

# 集成学习

- Bagging：并行。

- Boosting：串行。每次调整样本分布。Adaboost
- Stacking：串+并。上一个模型的输出结果作为一个特征。

# K-means

1. 目标函数：样本和其所属类的中心之间的距离的总和
2. 得到的类别是平坦的、非层次化的
3. 性能评估：
    - **有分类标签**的数据集
        - 使用**兰德指数**（ARI，Adjusted Rand Index）
        - 计算真实标签与聚类标签两种分布相似性之间的相似性，取值范围为[0,1]
        - 1表示最好的结果，即聚类类别和真实类别的分布完全一致
    - **没有分类标签**的数据集
        - 使用**轮廓系数**（Silhouette Coefficient）来度量聚类的质量
        - 轮廓系数同时考虑聚类结果的簇内凝聚度和簇间分离度
        - 取值范围：[-1,1]，轮廓系数越大，聚类效果越好

# PCA

1. 模型的性能会**随着特征的增加先上升后下降。**
2. 找主成分的方向