# 机器学习基本算法

目录

[机器学习基本算法 1](#_Toc518638144)

[一、线性回归 1](#_Toc518638145)

[1.基本原理 1](#_Toc518638146)

[2.模型L2正则化：Ridge正则化 1](#_Toc518638147)

[3.模型L1正则化：Lasso正则化 1](#_Toc518638148)

[4.应用场景 2](#_Toc518638149)

[5.具体应用 2](#_Toc518638150)

[二、逻辑回归 2](#_Toc518638151)

[1. 基本原理 2](#_Toc518638152)

[2.二元逻辑回归 2](#_Toc518638153)

[3.K元逻辑回归模型 3](#_Toc518638154)

[4.优点 3](#_Toc518638155)

[5.缺点 3](#_Toc518638156)

[6.应用场景 3](#_Toc518638157)

[7.具体应用 4](#_Toc518638158)

[三、决策树算法 4](#_Toc518638159)

[1.基本原理 4](#_Toc518638160)

[2.决策树ID3算法 5](#_Toc518638161)

[1)信息论基础 5](#_Toc518638162)

[2)具体算法（edited by wtt） 6](#_Toc518638163)

[3)不足之处 6](#_Toc518638164)

[3.决策树C4.5算法 7](#_Toc518638165)

[1)改进之处 7](#_Toc518638166)

[2)算法伪代码 8](#_Toc518638167)

[3)不足之处 8](#_Toc518638168)

[4. CART算法 9](#_Toc518638169)

[1.CART分类树算法(具体未看) 9](#_Toc518638170)

[2. CART回归树算法 9](#_Toc518638171)

[3. CART剪枝算法 9](#_Toc518638172)

[4. CART优点 9](#_Toc518638173)

[5. CART缺点 9](#_Toc518638174)

[5.应用场景 9](#_Toc518638175)

[6.优点 9](#_Toc518638176)

[7.缺点 10](#_Toc518638177)

[四、KNN算法 10](#_Toc518638178)

[1.基本原理 10](#_Toc518638179)

[2.KNN算法三要素 10](#_Toc518638180)

[3.KNN算法蛮力搜索 11](#_Toc518638181)

[4.KNN算法之KD树 12](#_Toc518638182)

[5.KNN算法之球树 12](#_Toc518638183)

[6.KNN算法扩展--限定半径最近邻算法和最近质心算法 12](#_Toc518638184)

[7.优点 12](#_Toc518638185)

[8.缺点 12](#_Toc518638186)

[五、朴素贝叶斯算法 13](#_Toc518638187)

[1.基本原理 13](#_Toc518638188)

[2.算法步骤 13](#_Toc518638189)

[3.应用场景 14](#_Toc518638190)

[4.优点 14](#_Toc518638191)

[5.缺点 14](#_Toc518638192)

[六、支持向量机算法 15](#_Toc518638193)

[1.基本原理 15](#_Toc518638194)

[2.具体算法步骤 15](#_Toc518638195)

[3.优点 15](#_Toc518638196)

[4.缺点 15](#_Toc518638197)

[八、随机森林算法 16](#_Toc518638198)

[1.基本原理 16](#_Toc518638199)

[2.应用场景 16](#_Toc518638200)

[3.优点 16](#_Toc518638201)

[4.缺点 17](#_Toc518638202)

[九、K-means聚类算法 17](#_Toc518638203)

[1.基本原理 17](#_Toc518638204)

[2.具体算法步骤 18](#_Toc518638205)

[3.优点 18](#_Toc518638206)

[4.缺点 18](#_Toc518638207)

[十、Apriori 18](#_Toc518638208)

[十一、神经网络算法 20](#_Toc518638209)

# 一、线性回归

## 1.基本原理

①拟合函数：或

②损失函数：或

③梯度下降法求解参数，每一轮参数迭代表达式：,为步长

③最小二乘法求解参数，每一轮参数迭代表达式：

④为了防止模型过拟合，进行加入正则化操作。

## 2.模型L2正则化：Ridge正则化

①损失函数：，为常数系数，用来调节线性回归和正则化权重，需要进行调试。为L2的范数。

②梯度下降法求解参数，每一轮参数迭代表达式：,为步长

③最小二乘法求解参数，每一轮参数迭代表达式：，单位矩阵

④优点：不抛弃任何一个变量的情况下，缩小了回归系数，使得模型相对而言比较的稳定

⑤缺点：模型的变量特别多，模型解释性差。

## 3.模型L1正则化：Lasso正则化

①损失函数：，为常数系数，用来调节线性回归和正则化权重，需要进行调试。为L1的范数。

②坐标轴下降法（coordinate descent）（具体算法步骤还未学）

③最小角回归法（ Least Angle Regression， LARS） （具体算法步骤还未学）

④优点：使得一些系数变小，甚至还是一些绝对值较小的系数直接变为0，特别适用于参数数目缩减与参数的选择，用来估计稀疏参数的线性模型。

⑤缺点：损失函数不是连续可导的，梯度下降法和最小二乘法不能使用。

## 4.应用场景

最基本的回归模型，常与其他模型做对比。

## 5.具体应用

1)获取数据

利用UCI大学公开的机器学习数据，下载地址为： <http://archive.ics.uci.edu/ml/machine-learning-databases/00294/>。是一个循环发电场的数据，共有9568个样本数据，每个数据有5列，分别是:AT（温度）, V（压力）, AP（湿度）, RH（压强）, PE（输出电力)。对应PE是样本输出，而AT/V/AP/RH这4个是样本特征，

2. 整理数据

下载后的数据可以发现是一个压缩文件，解压后可以看到里面有一个xlsx文件，我们先用excel把它打开，接着“另存为“”csv格式，保存下来，后面我们就用这个csv来运行线性回归。

实验做完了，但是具体实验步骤未写完

# 二、逻辑回归

## 1. 基本原理

分类模型，离散值，对线性回归的结果做一个在函数g上的转换，可以变化为逻辑回归。

## 2.二元逻辑回归

1)基本原理

函数g一般取sigmoid函数：，该函数有一个非常好的性质，当z趋于正无穷时，g(z)趋于1，而当z趋于负无穷时，g(z)趋于0，非常适合分类概率模型。另外，它还有一个很好的导数性质：

2)模型的一般形式

，

其中h(x)>0.5，分类为1；h(x)<0.5，分类为0；h(x)=0.5，无法分类。

3)损失函数：似然函数最大化求解

似然函数：

损失函数：

损失函数优化方法：一般有梯度下降法、坐标轴下降法、牛顿法(不了解)等。

4)正则化

面临过拟合问题，正则化同线性回归模型。

## 3.K元逻辑回归模型

1)模型的一般形式

逻辑回归模型

2)其余类似二元逻辑回归模型

## 4.优点

1.容易检查，大量用于工业问题

2.计算量小，速度快，存储资源低

3.不需要具有相等的方差或正态分布。

4.不假定依赖变量和独立变量之间的线性关系，可以通过L2正则化对多重共线性进行处理

## 5.缺点

1.当特征空间很大时，逻辑回归的性能不是很好；

2.容易欠拟合，分类精度可能不高，不能很好地处理大量多类特征或变量；

## 6.应用场景

1.当需要将响应变量的概率建模为某个其他解释变量的函数时

2.根据一些解释变量，当需要预测分类因变量分为二元响应的两类时

3.流行病学领域，识别疾病的风险因素，并据此制定预防措施。

4.预测候选人是否会赢得或失去政治选举或预测选民是否会投票给某个特定的候选人。

5.将一组单词分类为名词，代词，动词，形容词。

6.天气预报以预测降雨的可能性。

7.风险管理的信用评分系统，以预测账户的违约情况。

## 7.具体应用

实验做完了，但是具体实验步骤未写完

# 三、决策树算法

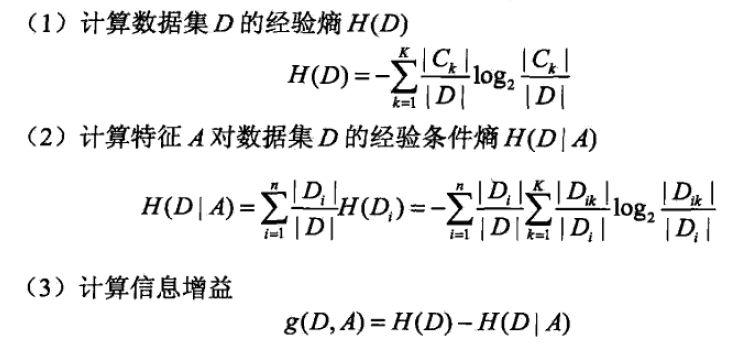
## 1.基本原理

分类算法，也可以作为回归算法

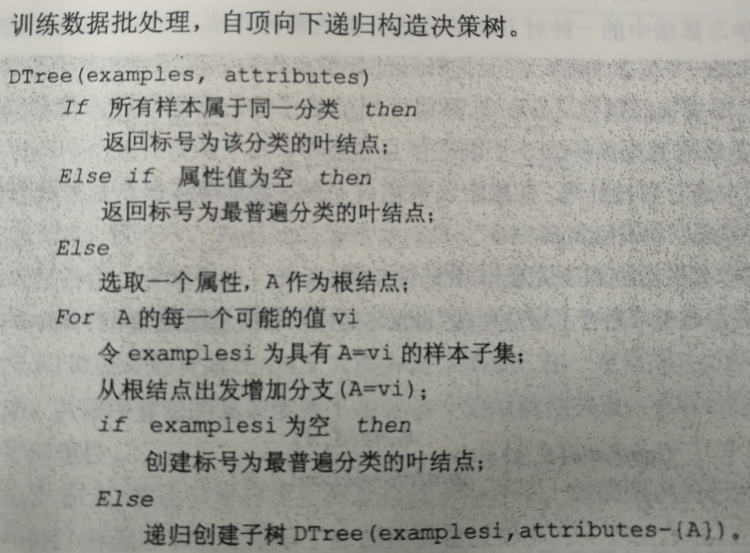
熵是对混乱度的度量

：表示系统总熵，

：属性A的熵，v代表属性A的状态1，2，3，4……v。



算法伪代码：



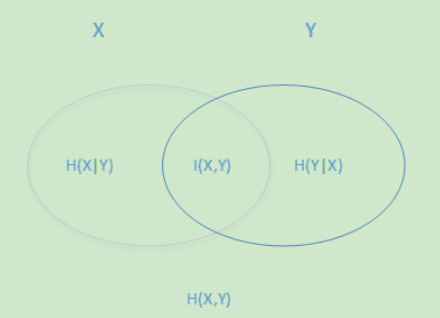
缺点：

1.很容易在训练数据中生成复杂的树结构，造成过拟合（overfitting）。剪枝可以缓解过拟合的负作用，常用方法是限制树的高度、叶子节点中的最少样本数量。

2.学习一棵最优的决策树被认为是NP-Complete问题。实际中的决策树是基于启发式的贪心算法建立的，这种算法不能保证建立全局最优的决策树。Random Forest 引入随机能缓解这个问题

## 2.决策树ID3算法

### 1)信息论基础

左边的椭圆代表熵H(X), 度量了X的不确定性。

右边的椭圆代表熵H(Y)。

中间重合的部分是互信息I(X,Y), 度量了X在知道了Y以后不确定性减少的程度。

左边的椭圆去掉重合部分是条件熵H(X|Y)，度量了在知道Y以后X的不确定性。

右边的椭圆去掉重合部分是条件熵H(Y|X)。

两个椭圆的并就是H(X,Y)。

### 2)具体算法（edited by wtt）

核心思想：以信息增益度量为依据，采用自顶向下的贪婪策略遍历坑的决策树空间，以选择出分裂后信息增益最大的属性。

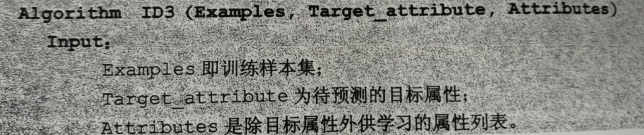
大致流程：

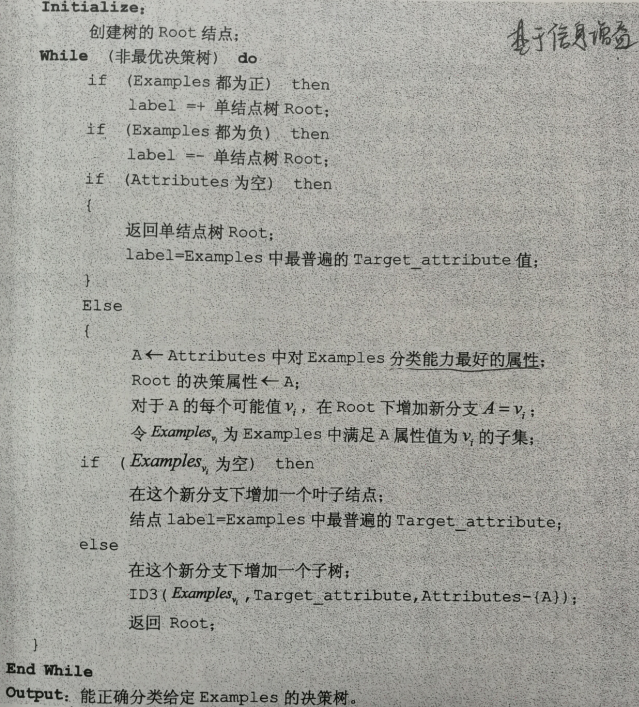
首先，使用统计测试来确定每一个实例属性单独分类新林样本的能力，选择分类能力最好的属性作为树的根节点。

然后，为根节点属性的每一个可能的值产生一个分支，把训练样本分配到适当的分支之下；重复该过程，用每个分支结点关联的训练样本选取在改点被测试的最佳属性；从而形成对合格决策树的贪婪搜索。

从上述流程可以看出，ID3无回溯，属于局部最优，而非全局最优。

算法伪代码：





### 3)不足之处

1.ID3没有考虑连续特征、缺失值、过拟合等问题

2.采用信息增益大的特征优先建立决策树的节点。但在相同条件下，取值比较多的特征比取值少的特征信息增益大。

## 3.决策树C4.5算法

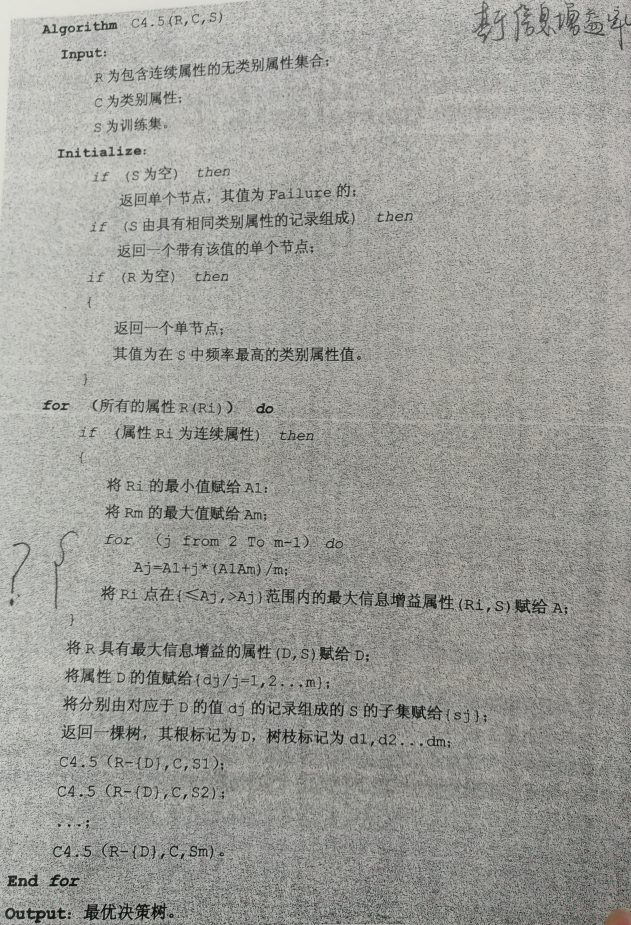
### 1)改进之处

ID3采用的信息增益度量存在一个缺点，它一般会优先选择有较多属性值的Feature,因为属性值多的Feature会有相对较大的信息增益?(信息增益反映的给定一个条件以后不确定性减少的程度,必然是分得越细的数据集确定性更高,也就是条件熵越小,信息增益越大).为了避免这个不足C4.5中是用信息增益比率(gain ratio)来作为选择分支的准则。信息增益比率通过引入一个被称作分裂信息(Split information)的项来惩罚取值较多的Feature。除此之外，C4.5还弥补了ID3中不能处理特征属性值连续的问题。但是，对连续属性值需要扫描排序，会使C4.5性能下降。

C4.5算法中的信息增益率融合了增益度量Gain(S,A)和分裂信息度量SplitInformation(S,A)来共同定义：

分裂信息用于衡量属性分裂数据的广度和均匀程度，定义为：

### 2)算法伪代码



### 3)不足之处

1.决策树算法非常容易过拟合，必须要进行剪枝。剪枝的算法有非常多，C4.5的剪枝方法主要是两种，一种是预剪枝，即在生成决策树的时候就决定是否剪枝。另一个是后剪枝，即先生成决策树，再通过交叉验证来剪枝。

2.生成的是多叉树，并且使用了熵模型，有大量的耗时的对数运算,如果是连续值还有大量的排序运算，算法效率不高。

3.只能用于分类

## 4. CART算法

### 1.CART分类树算法(具体未看)

1)最优特征选择方法

2)对连续特征和离散特征处理的改进

3)具体过程

### 2. CART回归树算法

### 3. CART剪枝算法

### 4. CART优点

### 5. CART缺点

## 5.应用场景

1.财务中期权定价

2.遥感(基于决策树的模式识别的应用领域)

3.银行对贷款申请人分类

4.Gerber产品公司使用决策树机器学习算法来决定他们是否应继续使用塑料PVC（聚氯乙烯）在他们的产品。

5.Rush大学医学中心开发了一个名为Guardian的工具，它使用决策树机器学习算法来识别有风险的患者和疾病趋势。

## 6.优点

1.简单直观，生成的决策树很直观。

2.基本不需要预处理，不需要提前归一化，处理缺失值。

4.既可以处理离散值也可以处理连续值。很多算法只是专注于离散值或者连续值。

5.可以处理多维度输出的分类问题。

6.相比于神经网络之类的黑盒分类模型，决策树在逻辑上可以得到很好的解释

7.可以交叉验证的剪枝来选择模型，从而提高泛化能力。

8.对于异常点的容错能力好，健壮性高。

## 7.缺点

1.决策树算法非常容易过拟合，导致泛化能力不强。可以通过设置节点最少样本数量和限制决策树深度来改进。

2.决策树会因为样本发生一点点的改动，就会导致树结构的剧烈改变。这个可以通过集成学习之类的方法解决。

3.寻找最优的决策树是一个NP难的问题，我们一般是通过启发式方法，容易陷入局部最优。可以通过集成学习之类的方法来改善。

4.有些比较复杂的关系，决策树很难学习，比如异或。这个就没有办法了，一般这种关系可以换神经网络分类方法来解决。

5.如果某些特征的样本比例过大，生成决策树容易偏向于这些特征。这个可以通过调节样本权重来改善。

# 四、KNN算法

## 1.基本原理

KNN方法既可以做分类，也可以做回归。KNN做回归和分类的主要区别在于最后做预测时候的决策方式不同。KNN做分类预测时，一般是选择多数表决法，即训练集里和预测的样本特征最近的K个样本，预测为里面有最多类别数的类别。而KNN做回归时，一般是选择平均法，即最近的K个样本的样本输出的平均值作为回归预测值。

## 2.KNN算法三要素

KNN算法主要考虑三个重要的要素：分类决策规则，k值的选取，距离度量的方式。

对于分类决策规则，一般都是使用前面提到的多数表决法。

对于k值的选择，没有一个固定的经验，一般根据样本的分布，选择一个较小的值，可以通过交叉验证选择一个合适的k值。

选择较小的k值，就相当于用较小的领域中的训练实例进行预测，训练误差会减小，只有与输入实例较近或相似的训练实例才会对预测结果起作用，与此同时带来的问题是泛化误差会增大，换句话说，K值的减小就意味着整体模型变得复杂，容易发生过拟合；

选择较大的k值，就相当于用较大领域中的训练实例进行预测，其优点是可以减少泛化误差，但缺点是训练误差会增大。这时候，与输入实例较远（不相似的）训练实例也会对预测器作用，使预测发生错误，且K值的增大就意味着整体的模型变得简单。

对于距离的度量，我们有很多的距离度量方式，但是最常用的是欧式距离，即对于两个n维向量x和y，两者的欧式距离定义为：

算法伪代码：



## 3.KNN算法蛮力搜索

要找到k个最近的邻居来做预测，只需要计算预测样本和所有训练集中的样本的距离，然后计算出最小的k个距离即可，接着多数表决，很容易做出预测。这个方法的确简单直接，在样本量少，样本特征少的时候有效。但是在实际运用中很多时候用不上，因为我们经常碰到样本的特征数有上千以上，样本量有几十万以上，如果我们这要去预测少量的测试集样本，算法的时间效率很成问题。因此，这个方法我们一般称之为蛮力实现。比较适合于少量样本的简单模型的时候用。

## 4.KNN算法之KD树

## 5.KNN算法之球树

## 6.KNN算法扩展--限定半径最近邻算法和最近质心算法

1)限定半径最近邻算法

样本中某系类别的样本非常的少，甚至少于K，导致稀有类别样本在找K个最近邻的时候，会把距离其实较远的其他样本考虑进来，而导致预测不准确。为了解决这个问题，限定最近邻的一个最大距离，只在一个距离范围内搜索所有的最近邻，这避免了上述问题。这个距离称为限定半径。

2)最近质心算法

首先把样本按输出类别归类。对于第 L类的CL个样本。它会对CL个样本的n维特征中每一维特征求平均值，最终该类别所有维度的n个平均值形成所谓的质心点。对于样本中的所有出现的类别，每个类别会最终得到一个质心点。当我们做预测时，需要比较预测样本和这些质心的距离，最小的距离对于的质心类别即为预测的类别。通常用在文本分类处理上。

## 7.优点

1）理论成熟，思想简单，既可以用来做分类也可以用来做回归

2）可用于非线性分类

3）训练时间复杂度比支持向量机之类的算法低，仅为O(n)

4）和朴素贝叶斯之类的算法比，对数据没有假设，准确度高，对异常点不敏感

5）由于KNN方法主要靠周围有限的邻近的样本，而不是靠判别类域的方法来确定所属类别的，因此对于类域的交叉或重叠较多的待分样本集来说，KNN方法较其他方法更为适合

6）该算法比较适用于样本容量比较大的类域的自动分类，而那些样本容量较小的类域采用这种算法比较容易产生误分

## 8.缺点

1）计算量大，尤其是特征数非常多的时候

2）样本不平衡的时候，对稀有类别的预测准确率低

3）KD树，球树之类的模型建立需要大量的内存

4）使用懒散学习方法，基本上不学习，导致预测时速度比起逻辑回归之类的算法慢

5）相比决策树模型，KNN模型可解释性不强

# 五、朴素贝叶斯算法

## 1.基本原理

生成式模型，用贝叶斯概率定理来建立，通过计算概率来进行分类。

对于未知类别的样本X，朴素贝叶斯分类将预测X属于具有最高后验概率的类，

根据贝叶斯定理有：

由于*P(X)*对于所有类为常数，只需要最大化即可。而：

对于某个样本,它所在的类别为：

## 2.算法步骤

**朴素贝叶斯分类算法：**

输入：训练数据集S

输出：各个类别的先验概率,各个类的后验概率

for(S中每个训练样本)

{

统计类别的计数;

for(每个描述属性值)

统计类别的描述属性值的计数

}

for(每个类别c)

{

;

for(每个描述属性)

for(每个描述属性值)

for(每个a1,a2，…,am)

}

## 3.应用场景

1.情绪分析(Facebook等社交网络)

2.文本分类(google使用文档分类来索引文档)

3.新闻分类

4.邮件分类(google mail使用邮件分类区分垃圾邮件和非垃圾邮件)

5.疾病预测

## 4.优点

1）朴素贝叶斯模型发源于古典数学理论，有稳定的分类效率。

2）对小规模的数据表现很好，能个处理多分类任务，适合增量式训练，尤其是数据量超出内存时，可以一批批的去增量训练。

3）对缺失数据不太敏感，算法也比较简单，常用于文本分类。

## 5.缺点

1）朴素贝叶斯模型与其他分类方法相比具有最小的误差率，但是实际上并非总是如此，这是因为朴素贝叶斯模型给定输出类别的情况下,假设属性之间相互独立，这个假设在实际应用中往往是不成立的，在属性个数比较多或者属性之间相关性较大时，分类效果不好。而在属性相关性较小时，朴素贝叶斯性能最为良好。对于这一点，有半朴素贝叶斯之类的算法通过考虑部分关联性适度改进。

2）需要知道先验概率，且先验概率很多时候取决于假设，假设的模型可以有很多种，因此在某些时候会由于假设的先验模型的原因导致预测效果不佳。

3）由于是通过先验和数据来决定后验的概率从而决定分类，所以分类决策存在一定的错误率。

4）对输入数据的表达形式很敏感。

# 六、支持向量机算法

## 1.基本原理

1.通过找到将训练集划分成类的线/超平面，从而将数据分类

2.存在很多这样的线/超平面，找到边际最大化(各类之间的距离)的线/超平面

3.包括线性SVM和非线性SVM

## 2.具体算法步骤

## 3.优点

1) 解决高维特征的分类问题和回归问题很有效,在特征维度大于样本数时依然有很好的效果。

2) 仅仅使用一部分支持向量来做超平面的决策，无需依赖全部数据。

3) 有大量的核函数可以使用，从而可以很灵活的来解决各种非线性的分类回归问题。

4)样本量不是海量数据的时候，分类准确率高，泛化能力强。

## 4.缺点

1) 如果特征维度远远大于样本数，则SVM表现一般。

2) SVM在样本量非常大，核函数映射维度非常高时，计算量过大，不太适合使用。

3）非线性问题的核函数的选择没有通用标准，难以选择一个合适的核函数。

4）SVM对缺失数据敏感。

# 八、随机森林算法

## 1.基本原理

首先随机选取不同的特征(feature)和训练样本(training sample)，生成大量的决策树，然后综合这些决策树的结果来进行最终的分类。

随机森林在现实分析中被大量使用，它相对于决策树，在准确性上有了很大的提升，同时一定程度上改善了决策树容易被攻击的特点。

数据维度相对低（几十维），同时对准确性有较高要求时。因为不需要很多参数调整就可以达到不错的效果，基本上不知道用什么方法的时候都可以先试一下随机森林。

## 2.应用场景

1.银行预测贷款申请人是否是高风险

2.汽车行业预测机械部件是否故障

3.医疗行业预测患者是否发展成慢性疾病

4.回归预测社交媒体份额和绩效分数的平均数

5.预测语音识别软件中的模式

6.图像和文本分类

## 3.优点

1） 训练可以高度并行化，对于大数据时代的大样本训练速度有优势。个人觉得这是的最主要的优点。

2） 由于可以随机选择决策树节点划分特征，这样在样本特征维度很高的时候，仍然能高效的训练模型。

3） 在训练后，可以给出各个特征对于输出的重要性

4） 由于采用了随机采样，训练出的模型的方差小，泛化能力强。

5） 相对于Boosting系列的Adaboost和GBDT， RF实现比较简单。

6） 对部分特征缺失不敏感。

## 4.缺点

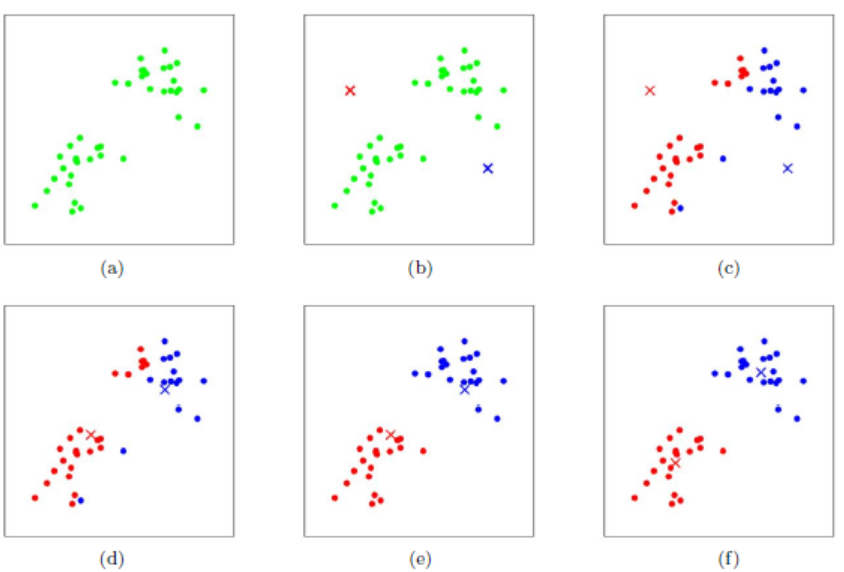
1）在某些噪音比较大的样本集上，RF模型容易陷入过拟合。

2) 取值划分比较多的特征容易对RF的决策产生更大的影响，从而影响拟合的模型的效果。

# 九、K-means聚类算法

## 1.基本原理

对于给定的样本集，按照样本之间的距离大小，将样本集划分为K个簇。让簇内的点尽量紧密的连在一起，而让簇间的距离尽量的大。



图a表达了初始的数据集，假设k=2。

图b中，随机选择了两个k类所对应的类别质心，即图中的红色质心和蓝色质心，然后分别求样本中所有点到这两个质心的距离，并标记每个样本的类别为和该样本距离最小的质心的类别，

图c中，经过计算样本和红色质心和蓝色质心的距离，得到了所有样本点的第一轮迭代后的类别。

此时对当前标记为红色和蓝色的点分别求其新的质心，如图d所示，新的红色质心和蓝色质心的位置已经发生了变动。

图e和图f重复了我们在图c和图d的过程，即将所有点的类别标记为距离最近的质心的类别并求新的质心。最终我们得到的两个类别如图f。

在实际K-Mean算法中，我们一般会多次运行图c和图d，才能达到最终的比较优的类别。

## 2.具体算法步骤

## 3.优点

1）原理比较简单，实现也是很容易，收敛速度快。

2）聚类效果较优。

3）算法的可解释度比较强。

4）主要需要调参的参数仅仅是簇数k。

## 4.缺点

1）K值的选取不好把握

2）对于不是凸的数据集比较难收敛

3）如果各隐含类别的数据不平衡，比如各隐含类别的数据量严重失衡，或者各隐含类别的方差不同，则聚类效果不佳。

4） 采用迭代方法，得到的结果只是局部最优。

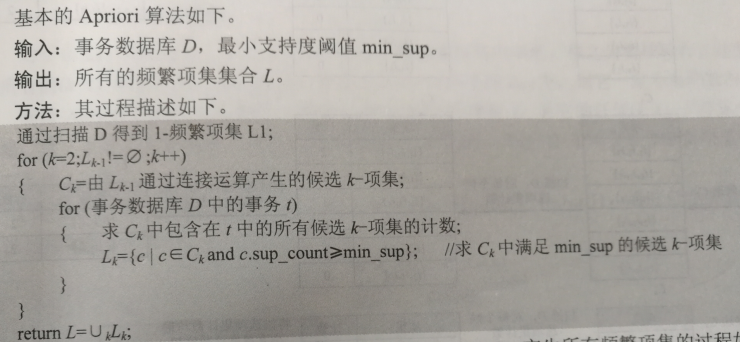
5） 对噪音和异常点比较的敏感。

# 十、Apriori

1.基本原理:

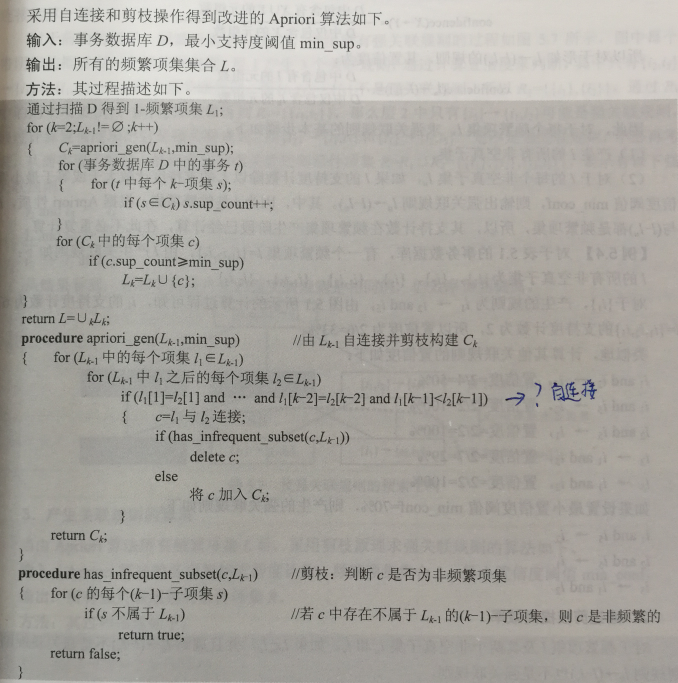
关联分析是一类非常有用的数据挖掘方法，能从数据中挖掘出潜在的关联关系。比如，在著名的购物篮事务（market basket transactions）问题中, 关联分析则被用来找出此类规则：顾客在买了某种商品时也会买另一种商品。在上述例子中，大部分都知道关联规则：{Diapers} → {Beer}；即顾客在买完尿布之后通常会买啤酒。后来通过调查分析，原来妻子嘱咐丈夫给孩子买尿布时，丈夫在买完尿布后通常会买自己喜欢的啤酒。

基本Apriori算法：

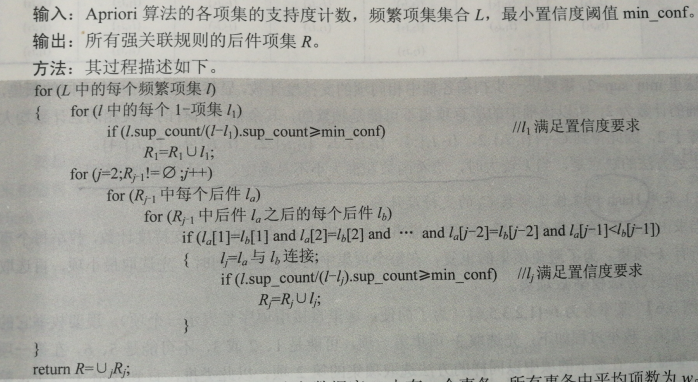


改进的Apriori算法：

采用自连接并增加了剪枝操作



当由Apriori算法所有频繁项集，采用剪枝原理求强关联规则的算法如下：



# 十一、神经网络算法

前馈神经网络用于分类。

前馈神经网络最常用的算法是误差反向传播（Back Propagation，BP）算法。

算法伪代码：

