数据挖掘十大经典算法

一、C4.5

C4.5 算法是机器学习算法中的一种分类决策树算法,其核心算法是 ID3 算法。

C4.5 算法继承了 ID3 算法的优点，并在以下几方面对 ID3 算法进行了改进：

1) 用信息增益率来选择属性，克服了用信息增益选择属性时偏向选择取值多的属性的不足；

2) 在树构造过程中进行剪枝；

3) 能够完成对连续属性的离散化处理；

4) 能够对不完整数据进行处理。

C4.5 算法有如下优点：产生的分类规则易于理解，准确率较高。其缺点是：在构造树的过程中，需要对数据集进行多次的顺序扫描和排序，因而导致算法的低效。

1、机器学习中，决策树是一个预测模型；他代表的是对象属性与对象值之间的一种映射关系。树中每个节点表示某个对象，而每个分叉路径则代表的某个可能的属性值，而每个叶结点则

对应从根节点到该叶节点所经历的路径所表示的对象的值。决策树仅有单一输出，若欲有复数输出，可以建立独立的决策树以处理不同输出。

2、从数据产生决策树的机器学习技术叫做决策树学习，通俗说就是决策树。

3、决策树学习也是数据挖掘中一个普通的方法。在这里，每个决策树都表述了一种树型结构，他由他的分支来对该类型的对象依靠属性进行分类。每个决策树可以依靠对源数据库的分割进行数据测试。这个过程可以递归式的对树进行修剪。当不能再进行分割或一个单独的类可以被应用于某一分支时，递归过程就完成了。另外，随机森林分类器将许多决策树结合起来以提升分类的正确率。

决策树是如何工作的？

1、决策树一般都是自上而下的来生成的。

2、选择分割的方法有好几种，但是目的都是一致的：对目标类尝试进行最佳的分割。

3、从根到叶子节点都有一条路径，这条路径就是一条—规则

4、决策树可以是二叉的，也可以是多叉的。

对每个节点的衡量：

1) 通过该节点的记录数

2) 如果是叶子节点的话，分类的路径

3) 对叶子节点正确分类的比例。

有些规则的效果可以比其他的一些规则要好。

由于 ID3 算法在实际应用中存在一些问题，于是 Quilan 提出了 C4。5 算法，严格上说 C4。5 只能是 ID3 的一个改进算法。

C4.5 算法继承了 ID3 算法的优点，**并在以下几方面对 ID3 算法进行了改进：**

1) 用信息增益率来选择属性，克服了用信息增益选择属性时偏向选择取值多的属性的不足；

2) 在树构造过程中进行剪枝；

3) 能够完成对连续属性的离散化处理；

4) 能够对不完整数据进行处理。

C4.5 算法有如下优点：产生的分类规则易于理解，准确率较高。其缺点是：在构造树的过程中，需要对数据集进行多次的顺序扫描和排序，因而导致算法的低效。此外，C4.5 只适合于能够驻留于内存的数据集，当训练集大得无法在内存容纳时程序无法运行。 来自搜索的其他内容：

C4.5 算法是机器学习算法中的一种分类决策树算法,其核心算法是 ID3 算法。 分类决策树算法是从大量事例中进行提取分类规则的自上而下的决策树。 决策树的各部分是:

根: 学习的事例集。

枝: 分类的判定条件。

叶: 分好的各个类。

ID3 算法

1、概念提取算法 CLS

1) 初始化参数 C={E},E 包括所有的例子,为根。

2) IF C 中的任一元素 e 同属于同一个决策类则创建一个叶子节点 YES 终止。

ELSE 依启发式标准,选择特征 Fi={V1,V2,V3,. . . Vn}并创建 判定节点

划分 C 为互不相交的 N 个集合 C1,C2,C3,. . . ,Cn；

3) 对任一个 Ci 递归。

2、ID3 算法

1) 随机选择 C 的一个子集 W (窗口)。

2) 调用 CLS 生成 W 的分类树 DT(强调的启发式标准在后)。

3) 顺序扫描 C 搜集 DT 的意外(即由 DT 无法确定的例子)。

4) 组合 W 与已发现的意外,形成新的 W。

5) 重复 2)到 4),直到无例外为止。

启发式标准:

只跟本身与其子树有关,采取信息理论用熵来量度。

熵是选择事件时选择自由度的量度,其计算方法为

P = freq(Cj,S)/|S|;

INFO(S)= - SUM(P*LOG(P)) ;

SUM()函数是求 j 从 1 到 n 和。

Gain(X)=Info(X)-Infox(X);

Infox(X)=SUM((|Ti|/|T|)*Info(X);

为保证生成的决策树最小,ID3 算法在生成子树时,选取使生成的子树的熵(即 Gain(S))最小的的特征来生成子树。

3、ID3 算法对数据的要求

1） 所有属性必须为离散量。

2）所有的训练例的所有属性必须有一个明确的值。

3） 相同的因素必须得到相同的结论且训练例必须唯一。

C4.5 对 ID3 算法的改进:

1、熵的改进,加上了子树的信息

Split_Infox(X)= - SUM((|T|/|Ti|) *LOG(|Ti|/|T|));

Gain ratio(X)= Gain(X)/Split Infox(X);

2、在输入数据上的改进

1) 因素属性的值可以是连续量,C4。5 对其排序并分成不同的集合后按照 ID3 算法当作离散量进行处理,但结论属性的值必须是离散值。

2）训练例的因素属性值可以是不确定的,以?表示,但结论必须是确定的

3）对已生成的决策树进行裁剪,减小生成树的规模。

二、k-means

术语“k-means”最早是由 James MacQueen 在 1967 年提出的,这一观点可以追溯到 1957 年 Hugo Steinhaus 所提出的想法。1957 年，斯图亚特·劳埃德最先提出这一标准算法，当初是作为一门应用于脉码调制的技术,直到 1982 年，这一算法才在贝尔实验室被正式提出。1965 年， E。W。Forgy 发表了一个本质上是相同的方法，1975 年和 1979 年，Hartigan 和 Wong 分别提出了一个更高效的版本。

算法描述

输入：簇的数目 k；包含 n 个对象的数据集 D。

输出：k 个簇的集合。

方法：

从 D 中任意选择 k 个对象作为初始簇中心；

repeat;

根据簇中对象的均值，将每个对象指派到最相似的簇；

更新簇均值，即计算每个簇中对象的均值；

计算准则函数；

until 准则函数不再发生变化。

算法的性能分析

1、优点

（1）k-平均算法是解决聚类问题的一种经典算法，算法简单、快速。

（2）对处理大数据集，该算法是相对可伸缩的和高效率的，因为它的复杂度大约是 O（nkt），其中 n 是所有对象的数目，k 是簇的数目,t 是迭代的次数。通常 k<<n。这个算法经常以局部最优结束。

（3）算法尝试找出使平方误差函数值最小的 k 个划分。当簇是密集的、球状或团状的，而簇与簇之间区别明显时，它的聚类效果很好。

2、缺点

（1）k-平均方法只有在簇的平均值被定义的情况下才能使用，不适用于某些应用，如涉及有分类属性的数据不适用。

（2）要求用户必须事先给出要生成的簇的数目 k。

（3）对初值敏感，对于不同的初始值，可能会导致不同的聚类结果。

（4）不适合于发现非凸面形状的簇，或者大小差别很大的簇。

（5）对于"噪声"和孤立点数据敏感，少量的该类数据能够对平均值产生极大影响。

算法的改进

针对算法存在的问题，对 K-means 算法提出一些改进：

一是数据预处理，

二是初始聚类中心选择，

三是迭代过程中聚类种子的选择。

1、首先对样本数据进行正规化处理，这样就能防止某些大值属性的数据左右样本间的距离。给定一组含有 n 个数据的数据集，每个数据含有 m 个属性，分别计算每一个属性的均值、标准差对每条数据进行标准化。

3、其次，初始聚类中心的选择对最后的聚类效果有很大的影响，原 K-means 算法是随机选取 k 个数据作为聚类中心，而聚类的结果要是同类间尽可能相似，不同类间尽可能相异，所以初始聚类中心的选取要尽可能做到这一点。采用基于距离和的孤立点定义来进行孤立点的预先筛选，并利用两两数据之间的最大距离在剩余数据集合中寻找初始聚类中心。但对于实际数据，孤立点个数往往不可预知。在选择初始聚类中心时，先将孤立点纳入统计范围，在样本中计算对象两两之间的距离，选出距离最大的两个点作为两个不同类的聚类中心，接着从其余的样本对象中找出已经选出来的所有聚类中心的距离和最大的点为另一个聚类中心，直到选出 k 个聚类中心。这样做就降低了样本输入顺序对初始聚类中心选择的影响。

聚类中心选好以后，就要进行不断的迭代计算，在 K-means 算法中，是将聚类均值点(类中所有数据的几何中心点)作为新的聚类种子进行新一轮的聚类计算，在这种情况下，新的聚类种子可能偏离真正的数据密集区，从而导致偏差，特别是在有孤立点存在的情况下，有很大的局限性。在选择初始中心点时，由于将孤立点计算在内，所以在迭代过程中要避免孤立点的影响。这里根据聚类种子的计算时，采用簇中那些与第 k-1 轮聚类种子相似度较大的数据，计算他们的均值点作为第 k 轮聚类的种子，相当于将孤立点排除在外，孤立点不参与聚类中心的计算，这样聚类中心就不会因为孤立点的原因而明显偏离数据集中的地方。在计算聚类中心的时候，要运用一定的算法将孤立点排除在计算均值点那些数据之外，这里主要采用类中与聚类种子相似度大于某一阈值的数据组成每个类的一个子集，计算子集中的均值点作为下一轮聚类的聚类种子。为了能让更多的数据参与到聚类中心的计算种去，阈值范围要包含大多数的数据。在第 k-1 轮聚类获得的类，计算该类中所有数据与该类聚类中心的平均距离 S，选择类中与聚类种子相似度大于 2S 的数据组成每个类的一个子集，以此子集的均值点作为第 k 轮聚类的聚类种子。在数据集中无论是否有明显的孤立点存在，两倍的平均距离都能包含大多数的数据。

对孤立点的改进—基于距离法经典 k 均值算法中没有考虑孤立点。所谓孤立点都是基于距离的，是数据 U 集中到 U 中最近邻居的距离最大的对象，换言之，数据集中与其最近邻居的平均距离最大的对象。针对经典 k 均值算法易受孤立点的影响这一问题，基于距离法移除孤立点，具体过程如下:首先扫描一次数据集，计算每一个数据对象与其临近对象的距离，累加求其距离和，并计算出距离和均值。如果某个数据对象的距离和大于距离和均值，则视该点为孤立点。把这个对象从数据集中移除到孤立点集合中，重复直到所有孤立点都找到。最后得到新的数据集就是聚类的初始集合。**对随机选取初始聚类中心的改进** 经典 k 均值算法随机选取 k 个点作为初

始聚类中心进行操作。由于是随机选取，则变化较大，初始点选取不同，获得聚类的结果也不同。并且聚类分析得到的聚类的准确率也不一样。对 k 均值算法的初始聚类中心选择方法—随机法进行改进，其依据是聚类过程中相同聚类中的对象是相似的，相异聚类中的对象是不相似的。因此提出了一种基于数据对象两两间的距离来动态寻找并确定初始聚类中心的思路，具体过程如下:

首先整理移除孤立点后的数据集 U,记录数据个数 y,令 m=1。比较数据集中所有数据对象两两之间的距离。找出距离最近的 2 个数据对象形成集合 Am;比较 Am 中每一个数据对象与数据对象集合 U 中每一个对象的距离,在 U 中找出与 Am 中最近的数据对象,优先吸收到 Am 中,直到 Am 中的数据对象个数到达一定数值,然后令 m=m+1。再从 U 中找到对象两两间距离最近的 2 个数据对象构成 Am,重复上面的过程,直到形成 k 个对象集合。这些集合内部的数据是相似的,而集合间是相异的。 可以看出,这种聚类方法同时满足以下 2 个条件:①每个组至少包含一个数据对象; ②每个数据对象必须属于且仅属于一个组。即数据对象 Xi ∈Ai ,且 U={{A1 ∪A2 ∪…∪Ak} ∪A0} ,且 Ai ∩Aj =Φ。最后对 k 个对象集合分别进行算术平均,形成 k 个初始聚类中心。

近似的 k 平均算法已经被设计用于原始数据子集的计算。 从算法的表现上来说，它并不保证一定得到全局最优解,最终解的质量很大程度上取决于初始化的分组。由于该算法的速度很快，因此常用的一种方法是多次运行 k 平均算法，选择最优解。

k 平均算法的一个缺点是，分组的数目 k 是一个输入参数，不合适的 k 可能返回较差的结果。另外，算法还假设均方误差是计算群组分散度的最佳参数。

三、Svm

支持向量机，英文为 Support Vector Machine，简称 SV 机（论文中一般简称 SVM）。它是一种监督式学习的方法，它广泛的应用于统计分类以及回归分析中。支持向量机属于一般化线性分类器。他们也可以认为是提克洛夫规范化（Tikhonov Regularization）方法的一个特例。这族分类器的特点是他们能够同时最小化经验误差与最大化几何边缘区。因此支持向量机也被称为最大边缘区分类器。在统计计算中，最大期望（EM）算法是在概率（probabilistic）模型中寻找参数最大似然估计的算法，其中概率模型依赖于无法观测的隐藏变量（Latent Variabl）。最大期望经常用在机器学习和计算机视觉的数据集聚（Data Clustering）领域。

最大期望算法经过两个步骤交替进行计算：

第一步是计算期望（E），也就是将隐藏变量象能够观测到的一样包含在内从而计算最大似然的期望值；另外一步是最大化（M），也就是最大化在 E 步上找到的最大似然的期望值从

而计算参数的最大似然估计。M 步上找到的参数然后用于另外一个 E 步计算，这个过程不断交替进行。

Vapnik 等人在多年研究统计学习理论基础上对线性分类器提出了另一种设计最佳准则。其原理也从线性可分说起，然后扩展到线性不可分的情况。甚至扩展到使用非线性函数中去，这种分类器被称为支持向量机(Support Vector Machine,简称 SVM)。支持向量机的提出有很深的理论背景。支持向量机方法是在近年来提出的一种新方法。

SVM 的主要思想可以概括为两点：

(1) 它是针对线性可分情况进行分析，对于线性不可分的情况，通过使用非线性映射算法将低维输入空间线性不可分的样本转化为高维特征空间使其线性可分，从而使得高维特征空间采用线性算法对样本的非线性特征进行线性分析成为可能；

(2) 它基于结构风险最小化理论之上在特征空间中建构最优分割超平面，使得学习器得到全局最优化，并且在整个样本空间的期望风险以某个概率满足一定上界。 在学习这种方法时，首先要弄清楚这种方法考虑问题的特点，这就要从线性可分的最简单情况讨论起，在没有弄懂其原理之前，不要急于学习线性不可分等较复杂的情况，支持向量机。在设计时，需要用到条件极值问题的求解，因此需用拉格朗日乘子理论，但对多数人来说，以前学到的或常用的是约束条件为等式表示的方式，但在此要用到以不等式作为必须满足的条件，此时只要了解拉格朗日理论的有关结论就行。

介绍

支持向量机将向量映射到一个更高维的空间里，在这个空间里建立有一个最大间隔超平面。在分开数据的超平面的两边建有两个互相平行的超平面。分隔超平面使两个平行超平面的距离最大化。假定平行超平面间的距离或差距越大，分类器的总误差越小。一个极好的指南是 C。J。C Burges 的《模式识别支持向量机指南》。van der Walt 和 Barnard 将支持向量机和其他分类器进行了比较。

动机

有很多个分类器(超平面）可以把数据分开，但是只有一个能够达到最大分割。我们通常希望分类的过程是一个机器学习的过程。这些数据点并不需要是中的点，而可以是任意(统计学符号)中或者(计算机科学符号)的点。我们希望能够把这些点通过一个 n-1 维的超平面分开，通常这个被称为线性分类器。有很多分类器都符合这个要求，但是我们还希望找到分类最佳的平面，即使得属于两个不同类的数据点间隔最大的那个面，该面亦称为最大间隔超平面。如果我们能够找到这个面，那么这个分类器就称为最大间隔分类器。

四、Apriori

Apriori 算法是种最有影响的挖掘布尔关联规则频繁项集的算法。它的核心是基于两阶段频集思想的递推算法。该关联规则在分类上属于单维、单层、布尔关联规则。在这里，所有支持度大于最小支持度的项集称为频繁项集(简称频集)，也常称为最大项目集。

在 Apriori 算法中，寻找最大项目集(频繁项集)的基本思想是：算法需要对数据集进行多步处理。第一步，简单统计所有含一个元素项目集出现的频数，并找出那些不小于最小支持度的项目集，即一维最大项目集。从第二步开始循环处理直到再没有最大项目集生成。循环过程是：第 k 步中，根据第 k-1 步生成的(k-1)维最大项目集产生 k 维侯选项目集，然后对数据库进行搜索，得到侯选项目集的项集支持度，与最小支持度进行比较，从而找到 k 维最大项目集。

从算法的运行过程，我们可以看出该 Apriori 算法的优点：简单、易理解、数据要求低，然而我们也可以看到 Apriori 算法的缺点：

(1)在每一步产生侯选项目集时循环产生的组合过多，没有排除不应该参与组合的元素；

(2)每次计算项集的支持度时，都对数据库 D 中的全部记录进行了一遍扫描比较，如果是一个大型的数据库的话，这种扫描比较会大大增加计算机系统的 I/O 开销。而这种代价是随着数据库的记录的增加呈现出几何级数的增加。因此人们开始寻求更好性能的算法，如 F-P 算法。

五、EM

最大期望算法（Expectation-maximization algorithm，又译期望最大化算法）在统计中被用于寻找，依赖于不可观察的隐性变量的概率模型中，参数的最大似然估计。

在统计计算中，最大期望（EM）算法是在概率模型中寻找参数最大似然估计或者最大后验估计的算法，其中概率模型依赖于无法观测的隐藏变量（Latent Variable）。最大期望经常用在机器学习和计算机视觉的数据聚类（Data Clustering）领域。最大期望算法经过两个步骤交替进行计算，第一步是计算期望（E），利用对隐藏变量的现有估计值，计算其最大似然估计值；第二步是最大化（M），最大化在 E 步上求得的最大似然值来计算参数的值。M 步上找到的参数估计值被用于下一个 E 步计算中，这个过程不断交替进行。M 是一个在已知部分相关变量的情况下，估计未知变量的迭代技术。

EM 的算法流程如下：

1、初始化分布参数

2、重复直到收敛：

3、E 步骤：估计未知参数的期望值，给出当前的参数估计。

4、M 步骤：重新估计分布参数，以使得数据的似然性最大，给出未知变量的期望估计。

六、PageRank

PageRank，网页排名，又称网页级别、Google 左侧排名或佩奇排名，是一种由搜索引擎根据网页之间相互的超链接计算的技术，而作为网页排名的要素之一，以 Google 公司创办人拉里·佩奇（Larry Page）之姓来命名。Google 用它来体现网页的相关性和重要性，在搜索引擎优化操作中是经常被用来评估网页优化的成效因素之一。Google 的创始人拉里·佩奇和谢尔盖·布林于 1998 年在斯坦福大学发明了这项技术。

PageRank 通过网络浩瀚的超链接关系来确定一个页面的等级。Google 把从 A 页面到 B 页面的链接解释为 A 页面给 B 页面投票，Google 根据投票来源（甚至来源的来源，即链接到 A 页面的页面）和投票目标的等级来决定新的等级。简单的说，一个高等级的页面可以使其他低等级页面的等级提升。

PageRank 让链接来"投票"，一个页面的“得票数”由所有链向它的页面的重要性来决定，到一个页面的超链接相当于对该页投一票。一个页面的 PageRank 是由所有链向它的页面（“链入页面”）的重要性经过递归算法得到的。一个有较多链入的页面会有较高的等级，相反如果一个页面没有任何链入页面，那么它没有等级。

2005 年初，Google 为网页链接推出一项新属性 nofollow，使得网站管理员和网志作者可以做出一些 Google 不计票的链接，也就是说这些链接不算作"投票"。nofollow 的设置可以抵制垃圾评论。

Google 工具条上的 PageRank 指标从 0 到 10。它似乎是一个对数标度算法，细节未知。

PageRank 是 Google 的商标，其技术亦已经申请专利。PageRank 算法中的点击算法是由 Jon Kleinberg 提出的。

PageRank 算法

1、PageRank

基本思想：如果网页 T 存在一个指向网页 A 的连接，则表明 T 的所有者认为 A 比较重要，从而把 T 的一部分重要性得分赋予 A。这个重要性得分值为：PR（T）/C(T)

其中 PR（T）为 T 的 PageRank 值，C(T)为 T 的出链数，则 A 的 PageRank 值为一系列类似于 T 的页面重要性得分值的累加。

优点：是一个与查询无关的静态算法，所有网页的 PageRank 值通过离线计算获得；有效减少在线查询时的计算量，极大降低了查询响应时间。

不足：人们的查询具有主题特征，PageRank 忽略了主题相关性，导致结果的相关性和主题性降低；另外，PageRank 有很严重的对新网页的歧视。

2、Topic-Sensitive PageRank（主题敏感的 PageRank）

基本思想:针对 PageRank 对主题的忽略而提出。核心思想:通过离线计算出一个 PageRank 向量集合，该集合中的每一个向量与某一主题相关，即计算某个页面关于不同主题的得分。主要分为两个阶段：主题相关的 PageRank 向量集合的计算和在线查询时主题的确定。

优点：根据用户的查询请求和相关上下文判断用户查询相关的主题（用户的兴趣）返回查询结果准确性高。

不足：没有利用主题的相关性来提高链接得分的准确性。

3、Hilltop

基本思想：与 PageRank 的不同之处：仅考虑专家页面的链接。主要包括两个步骤：专家页面搜索和目标页面排序。

优点：相关性强，结果准确。

不足：专家页面的搜索和确定对算法起关键作用，专家页面的质量决定了算法的准确性，而专家页面的质量和公平性难以保证；忽略了大量非专家页面的影响，不能反应整个 Internet 的民意；当没有足够的专家页面存在时，返回空，所以 Hilltop 适合对于查询排序进行求精。

影响 google PageRank 的因素有?

1 与 pr 高的网站做链接:

2 内容质量高的网站链接

3 加入搜索引擎分类目录

4 加入免费开源目录

5 你的链接出现在流量大、知名度高、频繁更新的重要网站上

6 google 对 DPF 格式的文件比较看重。

7 安装 Google 工具条

8 域名和 tilte 标题出现关键词与 meta 标签等

9 反向连接数量和反向连接的等级

10 Google 抓取您网站的页面数量

11 导出链接数量

七、AdaBoost

AdaBoost，是英文"Adaptive Boosting"（自适应增强）的缩写，是一种机器学习方法，由 Yoav Freund 和 Robert Schapire 提出。

AdaBoost 方法的自适应在于：前一个分类器分错的样本会被用来训练下一个分类器。

AdaBoost 方法对于噪声数据和异常数据很敏感。但在一些问题中，AdaBoost 方法相对于大多数其它学习算法而言，不会很容易出现过拟合现象。

AdaBoost 方法中使用的分类器可能很弱（比如出现很大错误率），但只要它的分类效果比随机好一点（比如两类问题分类错误率略小于 0。5），就能够改善最终得到的模型。而错误率高于随机分类器的弱分类器也是有用的，因为在最终得到的多个分类器的线性组合中，可以给它们赋予负系数，同样也能提升分类效果。

AdaBoost 方法是一种迭代算法，在每一轮中加入一个新的弱分类器，直到达到某个预定的足够小的错误率。每一个训练样本都被赋予一个权重，表明它被某个分类器选入训练集的概率。如果某个样本点已经被准确地分类，那么在构造下一个训练集中，它被选中的概率就被降低；相反，如果某个样本点没有被准确地分类，那么它的权重就得到提高。通过这样的方式，AdaBoost 方法能“聚焦于”那些较难分（更富信息）的样本上。

在具体实现上，最初令每个样本的权重都相等，对于第 k 次迭代操作，我们就根据这些权重来选取样本点，进而训练分类器 Ck。然后就根据这个分类器，来提高被它分错的的样本的权重，并降低被正确分类的样本权重。然后，权重更新过的样本集被用于训练下一个分类器 Ck[2]。整个训练过程如此迭代地进行下去。

Adaboost 算法的具体步骤如下：

1、给定训练样本集,其中,分别对应于正例样本和负例样本；为训练的最大循环次数；

2、初始化样本权重，即为训练样本的初始概率分布；

3、第一次迭代：

(1) 训练样本的概率分布下，训练弱分类器：

(2) 计算弱分类器的错误率：

(3) 选取，使得最小

(4) 更新样本权重：

(5) 最终得到的强分类器：

Adaboost 算法是经过调整的 Boosting 算法，其能够对弱学习得到的弱分类器的错误进行适应性调整。上述算法中迭代了次的主循环，每一次循环根据当前的权重分布对样本 x 定一个分布 P，然后对这个分布下的样本使用若学习算法得到一个错误率为的弱分类器，对于这个算法定义的弱学习算法，对所有的，都有，而这个错误率的上限并不需要事先知道，实际上。每一次迭代，都要对权重进行更新。更新的规则是：减小弱分类器分类效果较好的数据的概率，增大弱分类器分类效果较差的数据的概率。最终的分类器是个弱分类器的加权平均。

八、KNN

1、K 最近邻(k-Nearest Neighbor，KNN)分类算法，是一个理论上比较成熟的方法，也是最简单的机器学习算法之一。该方法的思路是:如果一个样本在特征空间中的 k 个最相似(即特征空间中最邻近)的样本中的大多数属于某一个类别，则该样本也属于这个类别。

2、KNN 算法中，所选择的邻居都是已经正确分类的对象。该方法在定类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。KNN 方法虽然从原理上也依赖于极限定理，但在类别决策时，只与极少量的相邻样本有关。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。

3、KNN 算法不仅可以用于分类，还可以用于回归。通过找出一个样本的 k 个最近邻居，将这些邻居的属性的平均值赋给该样本，就可以得到该样本的属性。更有用的方法是将不同距离的邻居对该样本产生的影响给予不同的权值(weight)，如权值与距离成正比。

4、该算法在分类时有个主要的不足是，当样本不平衡时，如一个类的样本容量很大，而其他类样本容量很小时，有可能导致当输入一个新样本时，该样本的 K 个邻居中大容量类的样本占多数。因此可以采用权值的方法（和该样本距离小的邻居权值大）来改进。该方法的另一个不足之处是计算量较大，因为对每一个待分类的文本都要计算它到全体已知样本的距离，才能求得它的 K 个最近邻点。目前常用的解决方法是事先对已知样本点进行剪辑，事先去除对分类作用不大的样本。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。

算法分类过程如下：

1、首先我们事先定下 k 值（就是指 k 近邻方法的 k 的大小，代表对于一个待分类的数据点，我们要寻找几个它的邻居）。这边为了说明问题，我们取两个 k 值，分别为 3 和 9；

2、根据事先确定的距离度量公式（如：欧氏距离），得出待分类数据点和所有已知类别的样本点中，距离最近的 k 个样本。

3、统计这 k 个样本点中，各个类别的数量。根据 k 个样本中，数量最多的样本是什么类别，我们就把这个数据点定为什么类别。训练样本是多维特征空间向量，其中每个训练样本带有一个类别标签。算法的训练阶段只包含存储的特征向量和训练样本的标签。 在分类阶段，k 是一个用户定义的常数。一个没有类别标签的向量 （查询或测试点）将被归类为最接近该点的 K 个样本点中最频繁使用的一类。一般情况下，将欧氏距离作为距离度量，但是这是只适用于连续变量。在文本分类这种非连续变量情况下，另一个度量——重叠度量（或海明距离）可以用来作为度量。

通常情况下，如果运用一些特殊的算法来计算度量的话，K 近邻分类精度可显著提高，如运用大边缘最近邻法或者近邻成分分析法。“多数表决”分类的一个缺点是出现频率较多的样本将会主导测试点的预测结果，那是因为他们比较大可能出现在测试点的 K 邻域而测试点的属性又是通过 K 领域内的样本计算出来的。解决这个缺点的方法之一是在进行分类时将样本到测试点的距离考虑进去。

K 值的选择

如何选择一个最佳的 K 值取决于数据。一般情况下，在分类时较大的 K 值能够减小噪声的影响。但会使类别之间的界限变得模糊。一个较好的 K 值能通过各种启发式技术来获取，比如，交叉验证。

噪声和非相关性特征向量的存在会使 K 近邻算法的准确性减小。对于选择特征向量进行分类已经作了很多研究。一个普遍的做法是利用进化算法优化功能扩展[3]，还有一种较普遍的方法是利用训练样本的互信息进行选择特征。

K 近邻算法也适用于连续变量估计，比如适用反距离加权平均多个 K 近邻点确定测试点的值。该算法的功能有：

1、从目标区域抽样计算欧式或马氏距离；

2、在交叉验证后的 RMSE 基础上选择启发式最优的 K 邻域；

3、计算多元 k-最近邻居的距离倒数加权平均。

九、Naive Baye

贝叶斯分类的基础是概率推理，就是在各种条件的存在不确定，仅知其出现概率的情况下，如何完成推理和决策任务。概率推理是与确定性推理相对应的。而朴素贝叶斯分类器是基于独立假设的，即假设样本每个特征与其他特征都不相关。举个例子，如果一种水果其具有红，圆，直径大概 4 英寸等特征，该水果可以被判定为是苹果。

尽管这些特征相互依赖或者有些特征由其他特征决定，然而朴素贝叶斯分类器认为这些属性在判定该水果是否为苹果的概率分布上独立的。朴素贝叶斯分类器依靠精确的自然概率模型，在有监督学习的样本集中能获取得非常好的分类效果。在许多实际应用中，朴素贝叶斯模型参数估计使用最大似然估计方法，换而言之朴素贝叶斯模型能工作并没有用到贝叶斯概率或者任何贝叶斯模型。

尽管是带着这些朴素思想和过于简单化的假设，但朴素贝叶斯分类器在很多复杂的现实情形中仍能够取得相当好的效果。2004 年，一篇分析贝叶斯分类器问题的文章揭示了朴素贝叶斯分类器取得看上去不可思议的分类效果的若干理论上的原因。尽管如此，2006 年有一篇文章详细比较了各种分类方法，发现更新的方法（如 boosted trees 和随机森林）的性能超

过了贝叶斯分类器。朴素贝叶斯分类器的一个优势在于只需要根据少量的训练数据估计出必要的参数（变量的均值和方差）。由于变量独立假设，只需要估计各个变量的方法，而不需要确定整个协方差矩阵。

两种分类模型：

分类是将一个未知样本分到几个预先已知类的过程。数据分类问题的解决是一个两步过程：第一步,建立一个模型，描述预先的数据集或概念集。通过分析由属性描述的样本（或实例，对象等）来构造模型。假定每一个样本都有一个预先定义的类，由一个被称为类标签的属性确定。为建立模型而被分析的数据元组形成训练数据集，该步也称作有指导的学习。在众多的分类模型中，应用最为广泛的两种分类模型是：**决策树模型(Decision Tree Model)和朴素贝叶斯模型（Naive Bayesian Model，NBC）；决策树模型通过构造树来解决分类问题。**

1、首先利用训练数据集来构造一棵决策树，一旦树建立起来，它就可为未知样本产生一个分类。在分类问题中使用决策树模型有很多的优点，决策树便于使用，而且高效；根据决策树可以很容易地构造出规则，而规则通常易于解释和理解；决策树可很好地扩展到大型数据库中，同时它的大小独立于数据库的大小；决策树模型的另外一大优点就是可以对有许多属性的数据集构造决策树。决策树模型也有一些缺点，比如处理缺失数据时的困难，过度拟合问题的出现，以及忽略数据集中属性之间的相关性等。

2、和决策树模型相比，朴素贝叶斯模型发源于古典数学理论，有着坚实的数学基础，以及稳定的分类效率。同时，NBC 模型所需估计的参数很少，对缺失数据不太敏感，算法也比较简单。理论上，NBC 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 NBC 模型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，这给 NBC 模型的正确分类带来了一定影响。在属性个数比较多或者属性之间相关性较大时，NBC 模型的分类效率比不上决策树模型。而在属性相关性较小时，NBC 模型的性能最为良好。

贝叶斯分类器特点

1、 需要知道先验概率

先验概率是计算后验概率的基础。在传统的概率理论中，先验概率可以由大量的重复实验所获得的各类样本出现的频率来近似获得，其基础是“大数定律”，这一思想称为“频率主义”。而在称为“贝叶斯主义”的数理统计学派中，他们认为时间是单向的，许多事件的发生不具有可重复性，因此先验概率只能根据对置信度的主观判定来给出，也可以说由“信仰”来确定。

2、按照获得的信息对先验概率进行修正

在没有获得任何信息的时候，如果要进行分类判别，只能依据各类存在的先验概率，将样本

划分到先验概率大的一类中。而在获得了更多关于样本特征的信息后，可以依照贝叶斯公式对先验概率进行修正，得到后验概率，提高分类决策的准确性和置信度。

3、分类决策存在错误率

由于贝叶斯分类是在样本取得某特征值时对它属于各类的概率进行推测，并无法获得样本真实的类别归属情况，所以分类决策一定存在错误率，即使错误率很低，分类错误的情况也可能发生。

十、CART

分类回归树(CART,Classification And Regression Tree)也属于一种决策树，分类回归树是一棵二叉树，且每个非叶子节点都有两个孩子，所以对于第一棵子树其叶子节点数比非叶子节点数多 1。

决策树生长的核心是确定决策树的分枝准则。

1、 如何从众多的属性变量中选择一个当前的最佳分支变量；

也就是选择能使异质性下降最快的变量。

异质性的度量：GINI、TWOING、least squared deviation。

前两种主要针对分类型变量，LSD 针对连续性变量。

代理划分、加权划分、先验概率

2、 如何从分支变量的众多取值中找到一个当前的最佳分割点（分割阈值）。

(1) 分割阈值：

A、数值型变量——对记录的值从小到大排序，计算每个值作为临界点产生的子节点的异质性统计量。能够使异质性减小程度最大的临界值便是最佳的划分点。

B、分类型变量——列出划分为两个子集的所有可能组合，计算每种组合下生成子节点的异质性。同样，找到使异质性减小程度最大的组合作为最佳划分点。

在决策树的每一个节点上我们可以按任一个属性的任一个值进行划分。按哪种划分最好呢？有 3 个标准可以用来衡量划分的好坏：GINI 指数、双化指数、有序双化指数。

终止条件：

一个节点产生左右孩子后，递归地对左右孩子进行划分即可产生分类回归树。这里的终止条件是什么？什么时候节点就可以停止分裂了？

满足以下一个即停止生长。

（1） 节点达到完全纯性；

（2） 数树的深度达到用户指定的深度；

（3） 节点中样本的个数少于用户指定的个数；

（4） 异质性指标下降的最大幅度小于用户指定的幅度。

剪枝

当分类回归树划分得太细时，会对噪声数据产生过拟合作用。因此我们要通过剪枝来解决。剪枝又分为前剪枝和后剪枝：前剪枝是指在构造树的过程中就知道哪些节点可以剪掉，于是干脆不对这些节点进行分裂，在 N 皇后问题和背包问题中用的都是前剪枝，上面的 x 2 方法也可以认为是一种前剪枝；后剪枝是指构造出完整的决策树之后再来考查哪些子树可以剪掉。

在分类回归树中可以使用的后剪枝方法有多种，比如：代价复杂性剪枝、最小误差剪枝、悲观误差剪枝等等。这里我们只介绍代价复杂性剪枝法。

预测

回归树——预测值为叶节点目标变量的加权均值

分类树——某叶节点预测的分类值应是造成错判损失最小的分类值。