数据挖掘十大经典算法

国际权威的学术组织 the IEEE International Conference on Data Mining (ICDM) 2006 年 12 月评选出了数据挖掘领域的十大经典算法：C4.5, k-Means, SVM, Apriori, EM, PageRank, AdaBoost, kNN, Naive Bayes, and CART.

不仅仅是选中的十大算法，其实参加评选的 18 种算法，实际上随便拿出一种来都可以称得上是经典算法，它们在数据挖掘领域都产生了极为深远的影响。

1. 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。

2. The k-means algorithm 即 K-Means 算法

k-means algorithm 算法是一个聚类算法，把 n 的对象根据他们的属性分为 k 个分割，k < n。它与处理混合正态分布的最大期望算法很相似,因为他们都试图找到数据中自然聚类的中心。它假设对象属性来自于空间向量，并且目标是使各个群组内部的均方误差总和最小。

3. Support vector machines

支持向量机，英文为 Support Vector Machine，简称 SV 机（论文中一般简称 SVM）。它是一种监督式學習的方法，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。

4. The Apriori algorithm

Apriori 算法是一种最有影响的挖掘布尔关联规则频繁项集的算法。其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。

5. 最大期望(EM)算法

在统计计算中，最大期望（EM，Expectation–Maximization）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。

6. PageRank

PageRank 是 Google 算法的重要内容。2001 年 9 月被授予美国专利，专利人是 Google 创始人之一拉里•佩奇（Larry Page）。因此，PageRank 里的 page 不是指网页，而是指佩奇，即这个

等级方法是以佩奇来命名的。

PageRank 根据网站的外部链接和内部链接的数量和质量俩衡量网站的价值。PageRank 背后的概念是，每个到页面的链接都是对该页面的一次投票，被链接的越多，就意味着被其他网站投票越多。这个就是所谓的“链接流行度”——衡量多少人愿意将他们的网站和你的网站挂钩。PageRank 这个概念引自学术中一篇论文的被引述的频度——即被别人引述的次数越多，一般判断这篇论文的权威性就越高。

7. AdaBoost

Adaboost 是一种迭代算法，其核心思想是针对同一个训练集训练不同的分类器(弱分类器)，然后把这些弱分类器集合起来，构成一个更强的最终分类器 (强分类器)。其算法本身是通过改变数据分布来实现的，它根据每次训练集之中每个样本的分类是否正确，以及上次的总体分类的准确率，来确定每个样本的权值。将修改过权值的新数据集送给下层分类器进行训练，最后将每次训练得到的分类器最后融合起来，作为最后的决策分类器。

8. kNN: k-nearest neighbor classification

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。

9. Naive Bayes

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）。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。

10. CART: 分类与回归树

CART, Classification and Regression Trees。 在分类树下面有两个关键的思想。第一个是关于递归地划分自变量空间的想法；第二个想法是用验证数据进行剪枝。

数据挖掘十大经典算法(1)C4.5

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。

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。

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。

决策树同时也可以依靠计算条件概率来构造。决策树如果依靠数学的计算方法可以取得更加理想的效果。

决策树是如何工作的?

决策树一般都是自上而下的来生成的。

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。

从根到叶子节点都有一条路径，这条路径就是一条“规则”。

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。相信大家对 ID3 算法都很.熟悉了，这里就不做介绍。

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.

§4.3.2 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.

§4.3.3: 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.

§4.3.4: 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.

数据挖掘十大经典算法(2) k-means

k-means algorithm 算法是一个聚类算法，把 n 的对象根据他们的属性分为 k 个分割，k < n。它与处理混合正态分布的最大期望算法很相似,因为他们都试图找到数据中自然聚类的中心。它假设对象属性来自于空间向量，并且目标是使各个群组内部的均方误差总和最小。

假设有 k 个群组 Si, i=1,2,...,k。μi 是群组 Si 内所有元素 xj 的重心，或叫中心点。

k 平均聚类发明于 1956 年， 该算法最常见的形式是采用被称为劳埃德算法(Lloyd algorithm)的迭代式改进探索法。劳埃德算法首先把输入点分成 k 个初始化分组，可以是随机的或者使用一些启发式数据。然后计算每组的中心点，根据中心点的位置把对象分到离它最近的中心，重新确定分组。继续重复不断地计算中心并重新分组，直到收敛，即对象不再改变分组（中心点位置不再改变）。

劳埃德算法和 k 平均通常是紧密联系的，但是在实际应用中，劳埃德算法是解决 k 平均问题的启发式法则，对于某些起始点和重心的组合，劳埃德算法可能实际上收敛于错误的结果。（上面函数中存在的不同的最优解）

虽然存在变异，但是劳埃德算法仍旧保持流行，因为它在实际中收敛非常快。实际上，观察发现迭代次数远远少于点的数量。然而最近，David Arthur 和 Sergei Vassilvitskii 提出存在特定的点集使得 k 平均算法花费超多项式时间达到收敛。

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。
数据挖掘十大经典算法(3) Svm
支持向量机，英文为 Support Vector Machine，简称 SV 机（论文中一般简称 SVM）。它是一种监督式學習的方法，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。

问题定义

设样本属于两个类，用该样本训练 svm 得到的最大间隔超平面。在超平面上的样本点也称为支持向量.
我们考虑以下形式的样本点

其中 ci 为 1 或−1 --用以表示数据点属于哪个类. 是一个 p－(统计学符号), 或 n－(计算机科学符号) 维向量，其每个元素都被缩放到[0,1]或[-1,1].缩放的目的是防止方差大的随机变量主导分类过程.我们可以把这些数据称为“训练数据”，希望我们的支持向量机能够通过一个超平面正确的把他们分开。超平面的数学形式可以写作

根据几何知识，我们知道向量垂直于分类超平面。加入位移 b 的目的是增加间隔.如果没有 b 的话，那超平面将不得不通过原点，限制了这个方法的灵活性。

由于我们要求最大间隔，因此我们需要知道支持向量以及（与最佳超平面）平行的并且离支持向量最近的超平面。我们可以看到这些平行超平面可以由方程族：

来表示。

如果这些训练数据是线性可分的，那就可以找到这样两个超平面，在它们之间没有任何样本点并且这两个超平面之间的距离也最大.通过几何不难得到这两个超平面之间的距离是 2/|w|，因此我们需要最小化 |w|。同时为了使得样本数据点都在超平面的间隔区以外，我们需要保证对于所有的 i 满足其中的一个条件

这两个式子可以写作：

原型
现在寻找最佳超平面这个问题就变成了在(1)这个约束条件下最小化|w|.这是一个二次规劃QP(quadratic programming)最优化中的问题。

更清楚的，它可以表示如下：

最小化 ，满足 。
1/2 这个因子是为了数学上表达的方便加上的。

对偶型(Dual Form)
把原型的分类规则写作对偶型，可以看到分类器其实是一个关于支持向量（即那些在间隔区边缘的训练样本点）的函数。

支持向量机的对偶型如下： 并满足 αi > = 0

软间隔
1995 年, Corinna Cortes 与 Vapnik 提出了一种改进的最大间隔区方法，这种方法可以处理标记错误的样本。如果可区分正负例的超平面不存在，则“软边界”将选择一个超平面尽可能清晰地区分样本，同时使其与分界最清晰的样本的距离最大化。这一成果使术语“支持向量机”（或“SVM”）得到推广。这种方法引入了松驰参数 ξi 以衡量对数据 xi 的误分类度。

。
随后，将目标函数与一个针对非 0ξi 的惩罚函数相加，在增大间距和缩小错误惩罚两大目标之间进行权衡优化。如果惩罚函数是一个线性函数，则等式(3)变形为
数据挖掘十大经典算法(4)Apriori
Apriori 算法是一种最有影响的挖掘布尔关联规则频繁项集的算法。其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。

Apriori 演算法所使用的前置统计量包括了：

最大规则物件数：规则中物件组所包含的最大物件数量
最小支援：规则中物件或是物件组必顸符合的最低案例数
最小信心水准：计算规则所必须符合的最低信心水准门槛
该算法的基本思想是：首先找出所有的频集，这些项集出现的频繁性至少和预定义的最小支持度一样。然后由频集产生强关联规则，这些规则必须满足最小支持度和最小可信度。然后使用第 1 步找到的频集产生期望的规则，产生只包含集合的项的所有规则，其中每一条规则的右部只有一项，这里采用的是中规则的定义。一旦这些规则被生成，那么只有那些大于用户给定的最小可信度的规则才被留下来。为了生成所有频集，使用了递推的方法。
可能产生大量的候选集,以及可能需要重复扫描数据库，是 Apriori 算法的两大缺点。
数据挖掘十大经典算法(5) EM

在统计计算中，最大期望（EM，Expectation–Maximization）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。

最大期望过程说明

我们用 表示能够观察到的不完整的变量值，用 表示无法观察到的变量值，这样 和 一起组成了完整的数据。 可能是实际测量丢失的数据，也可能是能够简化问题的隐藏变量，如果它的值能够知道的话。例如，在混合模型（Mixture Model）中，如果“产生”样本的混合元素成分已知的话最大似然公式将变得更加便利（参见下面的例子）。

估计无法观测的数据

让 代表矢量 θ: 定义的参数的全部数据的概率分布（连续情况下）或者概率集聚函数（离散情况下），那么从这个函数就可以得到全部数据的最大似然值，另外，在给定的观察到的数据条件下未知数据的条件分布可以表示为：

数据挖掘十大经典算法(6) PageRank

PageRank 是 Google 算法的重要内容。2001 年 9 月被授予美国专利，专利人是 Google 创始人之一拉里•佩奇（Larry Page）。因此，PageRank 里的 page 不是指网页，而是指佩奇，即这个等级方法是以佩奇来命名的。

Google 的 PageRank 根据网站的外部链接和内部链接的数量和质量俩衡量网站的价值。PageRank 背后的概念是，每个到页面的链接都是对该页面的一次投票，被链接的越多，就意味着被其他网站投票越多。这个就是所谓的“链接流行度”——衡量多少人愿意将他们的网站和你的网站挂钩。PageRank 这个概念引自学术中一篇论文的被引述的频度——即被别人引述的次数越多，一般判断这篇论文的权威性就越高。

Google 有一套自动化方法来计算这些投票。Google 的 PageRank 分值从 0 到 10；PageRank 为 10 表示最佳，但非常少见，类似里氏震级（Richter scale），PageRank 级别也不是线性的，而是按照一种指数刻度。这是一种奇特的数学术语，意思是 PageRank4 不是比 PageRank3 好一级——而可能会好 6 到 7 倍。因此，一个 PageRank5 的网页和 PageRank8 的网页之间的差距会比你可能认为的要大的多。

PageRank 较高的页面的排名往往要比 PageRank 较低的页面高，而这导致了人们对链接的着魔。在整个 SEO 社区，人们忙于争夺、交换甚至销售链接，它是过去几年来人们关注的焦点，以至于 Google 修改了他的系统，并开始放弃某些类型的链接。比如，被人们广泛接受的一条规定，来自缺乏内容的“link farm”（链接工厂）网站的链接将不会提供页面的 PageRank，从 PageRank 较高的页面得到链接但是内容不相关（比如说某个流行的漫画书网站链接到一个叉车规范页面），也不会提供页面的 PageRank。Google 选择降低了 PageRank 对更新频率，以便不鼓励人们不断的对其进行监测。

Google PageRank 一般一年更新四次，所以刚上线的新网站不可能获得 PR 值。你的网站很可能在相当长的时间里面看不到 PR 值的变化，特别是一些新的网站。PR 值暂时没有，这不是什么不好的事情，耐心等待就好了。

为您的网站获取外部链接是一件好事，但是无视其他 SEO 领域的工作而进行急迫的链接建设

就是浪费时间，要时刻保持一个整体思路并记住以下几点：
•Google 的排名算法并不是完全基于外部链接的
•高 PageRank 并不能保证 Google 高排名
•PageRank 值更新的比较慢，今天看到的 PageRank 值可能是三个月前的值
因此我们不鼓励刻意的去追求 PageRank，因为决定排名的因素可以有上百种。尽管如此，PageRank 还是一个用来了解 Google 对您的网站页面如何评价的相当好的指示，建议网站设计者要充分认识 PageRank 在 Google 判断网站质量中的重要作用，从设计前的考虑到后期网站更新都要给予 PageRank 足够的分析，很好的利用。我们要将 PageRank 看作是一种业余爱好而不是一种信仰。

通过对由超过 50,000 万个变量和 20 亿个词汇组成的方程进行计算，PageRank 能够对网页的重要性做出客观的评价。PageRank 并不计算直接链接的数量，而是将从网页 A 指向网页 B 的链接解释为由网页 A 对网页 B 所投的一票。这样，PageRank 会根据网页 B 所收到的投票数量来评估该页的重要性。
此外，PageRank 还会评估每个投票网页的重要性，因为某些网页的投票被认为具有较高的价值，这样，它所链接的网页就能获得较高的价值。重要网页获得的 PageRank（网页排名）较高，从而显示在搜索结果的顶部。Google 技术使用网上反馈的综合信息来确定某个网页的重要性。搜索结果没有人工干预或操纵，这也是为什么 Google 会成为一个广受用户信赖、不受付费排名影响且公正客观的信息来源。
－－－－－－－－－－－－－－－－－－
其实简单说就是民主表决。打个比方，假如我们要找李开复博士，有一百个人举手说自己是李开复。那么谁是真的呢？也许有好几个真的，但即使如此谁又是大家真正想找的呢？:-) 如果大家都说在 Google 公司的那个是真的，那么他就是真的。
在互联网上，如果一个网页被很多其它网页所链接，说明它受到普遍的承认和信赖，那么它的排名就高。这就是 Page Rank 的核心思想。 当然 Google 的 Page Rank 算法实际上要复杂得多。比如说，对来自不同网页的链接对待不同，本身网页排名高的链接更可靠，于是给这些链接予较大的权重。Page Rank 考虑了这个因素，可是现在问题又来了，计算搜索结果的网页排名过程中需要用到网页本身的排名，这不成了先有鸡还是先有蛋的问题了吗？
Google 的两个创始人拉里•佩奇 （Larry Page ）和谢尔盖•布林 (Sergey Brin) 把这个问题变成了一个二维矩阵相乘的问题，并且用迭代的方法解决了这个问题。他们先假定所有网页的排名是相同的，并且根据这个初始值，算出各个网页的第一次迭代排名，然后再根据第一次迭代排名算出第二次的排名。他们两人从理论上证明了不论初始值如何选取，这种算法都保证了网页排名的估计值能收敛到他们的真实值。值得一提的事，这种算法是完全没有任何人工干预的。
理论问题解决了，又遇到实际问题。因为互联网上网页的数量是巨大的，上面提到的二维矩阵从理论上讲有网页数目平方之多个元素。如果我们假定有十亿个网页，那么这个矩阵就有一百亿亿个元素。这样大的矩阵相乘，计算量是非常大的。拉里和谢尔盖两人利用稀疏矩阵计算的技巧，大大的简化了计算量，并实现了这个网页排名算法。今天 Google 的工程师把这个算法移植到并行的计算机中，进一步缩短了计算时间，使网页更新的周期比以前短了许多。
我来 Google 后，拉里 (Larry) 在和我们几个新员工座谈时，讲起他当年和谢尔盖(Sergey) 是怎么想到网页排名算法的。他说："当时我们觉得整个互联网就像一张大的图（Graph），每个网站就像一个节点，而每个网页的链接就像一个弧。我想，互联网可以用一个图或者矩阵描述，我也许可以用这个发现做个博士论文。" 他和谢尔盖就这样发明了 Page Rank 的算法。

网页排名的高明之处在于它把整个互联网当作了一个整体对待。它无意识中符合了系统论的观点。相比之下，以前的信息检索大多把每一个网页当作独立的个体对待，很多人当初只注意了网页内容和查询语句的相关性，忽略了网页之间的关系。

今天，Google 搜索引擎比最初复杂、完善了许多。但是网页排名在 Google 所有算法中依然是至关重要的。在学术界，这个算法被公认为是文献检索中最大的贡献之一，并且被很多大学引入了信息检索课程 (Information Retrieval) 的教程。

如何提高你网页的 PR 值？

什么是 PR 值呢? PR 值全称为 PageRank，PR 是英文 Pagerank 的缩写形式，Pagerank 取自 Google 的创始人 LarryPage，它是 Google 排名运算法则（排名公式）的一部分，Pagerank 是 Google 对网页重要性的评估，是 Google 用来衡量一个网站的好坏的唯一标准。PageRank(网页级别)是 Google 用于评测一个网页“重要性”的一种方法。在揉合了诸如 Title 标识和 Keywords 标识等所有其它因素之后，Google 通过 PageRank 来调整结果，使那些更具“重要性”的网页在搜索结果中另网站排名获得提升，从而提高搜索结果的相关性和质量。PR 值的级别从 1 到 10 级，10 级为满分。PR 值越高说明该网页越受欢迎。Google 把自己的网站的 PR 值定到 10，这说明 Google 这个网站是非常受欢迎的，也可以说这个网站非常重要。Google 大受青睐的另一个原因就是它的网站索引速度。向 Google 提交你的网站直到为 Google 收录，一般只需两个星期。如果你的网站已经为 Google 收录，那么通常 Google 会每月一次遍历和更新(重新索引)你的网站信息。不过对于那些 PR 值 (Pagerank)较高的网站，Google 索引周期会相应的短一些。一个 PR 值为 1 的网站表明这个网站不太具有流行度，而 PR 值为 7 到 10 则表明这个网站非常受欢迎。PR 值最高为 10，一般 PR 值达到 4，就算是一个不错的网站了。那么 PR 值都受那些因素影响呢？下面我们一起来看看。

第一：网站外部链接的数量和质量

在计算网站排名时，Pagerank 会将网站的外部链接数考虑进去。并不能说一个网站的外部链接数越多其 PR 值就越高，如果这样的话，一个网站尽可能获得最多的外部链接就 OK 了，有这种想法是错误的。Google 对一个网站上的外部链接数的重视程度并不意味着你因此可以不求策略地与任何网站建立连接。这是因为 Google 并不是简单地由计算网站的外部链接数来决定其等级。Google 的 Pagerank 系统不单考虑一个网站的外部链接质量，也会考虑其数量。这个问题看来很有复杂。首先让我们来解释一下什么是阻尼因数(damping factor)。阻尼因素就是当你投票或链接到另外一个站点时所获得的实际 PR 分值。阻尼因数一般是 0.85。当然比起你网站的实际 PR 值，它就显得微不足道了。

现在让我们来看看这个 PR 分值的计算公式：PR(A)=(1- d)+d(PR(t1)/C(t1)+...+PR(tn)/C(tn)) 公式解释：其中 PR(A)表示的是从一个外部链接站点 t1 上，依据 Pagerank?系统给你的网站所增加的 PR 分值；PR(t1)表示该外部链接网站本身的 PR 分值；C(t1)则表示该外部链接站点所拥有的外部链接数量。大家要谨记：一个网站的投票权值只有该网站 PR 分值的 0.85，

那么，是不是说对一个网站而言，它所拥有的较高网站质量和较高 PR 分值的外部链接数量越多就越好呢？错，因为－Google 的 Pagerank 系统不单考虑一个网站的外部链接质量，也会考虑其数量.比方说，对一个有一定 PR 值的网站 X 来说，如果你的网站 Y 是它的唯一一个外部链接，那么 Google 就相信网站 X 将你的网站 Y 视做它最好的一个外部链接，从而会给你的网站 Y 更多的分值。可是，如果网站 X 上已经有 49 个外部链接，那么 Google 就相信网站 X 只是将你的网站视做它第 50 个好的网站。因而你的外部链接站点上的外部链接数越多，你所能够得到的 PR 分值反而会越低，它们呈反比关系。

说它对是因为－一般情况下，一个 PR 分值大于等于 6 的外部链接站点，可显著提升你的 PR 分值。但如果这个外部链接站点已经有 100 个其它的外部链接时，那你能够得到的 PR 分值就几乎为零了。同样，如果一个外部链接站点的 PR 值仅为 2，但你却是它的唯一一个外部链

接，那么你所获得的 PR 值要远远大于那个 PR 值为 6，外部链接数为 100 的网站。
而且这个 0.85 的权值平均分配给其链接的每个外部网站。
第二：Google 在你的网站抓取的页面数
Google 在你的网站抓取的页面数，数目越多，Pagerank 值越高。但通常 Google 并不会主动抓取你的网站的所有页面，尤其是网址里带有“?”的动态链接，Google 不主动，那就要我们主动了，最笨的办法是把网站所有的页面都提交给 Google，但我想没有谁真会这么做，但页面不多的话可以试试。更好的办法是制作一个静态 Html 页面，通常被称作“网站地图”或“网站导航”，它里面包含你要添加的所有网址，然后把这个静态页面提交给 Google。
第三：网站被世界三大知名网站 DMOZ，Yahoo 和 Looksmart 收录
众所周知，Google 的 Pagerank 系统对那些门户网络目录如 DMOZ，Yahoo 和 Looksmart 尤为器重。特别是对 DMOZ。一个网站上的 DMOZ 链接对 Google 的 Pagerank?来说，就好像一块金子一样珍贵。如果你的网站为 ODP 收录，则可有效提升你的页面等级。向 ODP 提交你的站点并为它收录，其实并不是一件难事，只是要多花点时间而已。只要确保你的网站提供了良好的内容，然后在 ODP 合适的目录下点击"增加站点"，按照提示一步步来就 OK 了。至少要保证你的索引页(INDEX PAGE)被收录进去。所以，如果你的网站内容涉及完全不同的几块内容，你可以把每个内容的网页分别向 ODP 提交－不过请记住"欲速则不达"。等到 Google 对其目录更新后，你就能看到你的 PR 值会有什么变化了。如果你的网站为 Yahoo 和 Looksmart 所收录，那么你的 PR 值会得到显著提升。如果你的网站是非商业性质的或几乎完全是非商业性质的内容，那么你可以通过 zeall.com 使你的网站为著名的网络目录 Looksmart 所收录。 Looksmart 也是从 Zeal 网络目录获得非商业搜索列表。
Google PR 值的更新周期是多长时间？
一般情况下 PR 值更新的周期是 2.5～3 个月！最近一次 PR 更新是 2008 年 1 月中旬。
PageRank 相关算法总结：
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。
那么影响 google PageRank 的因素有哪些呢?
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
6google 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
10Google 抓取您网站的页面数量
11 导出链接数量
PageRank 科学排名遏止关键字垃圾
目前，五花八门的网站为争夺网上排名采用恶意点击和输入关键字垃圾的手段来吸引网民的眼球，无论对于互联网企业还是互联网用户，这都不是一个好现象。
为了解决这样的问题，Google 创始人之一拉里.佩奇（Larry Page）发明了一种算法 PageRank，是由搜索引擎根据网页之间相互的超链接进行计算的网页排名。它经常和搜索引擎优化有关。PageRank 系统目前被 Google 用来体现网页的相关性和重要性，以便科学排名，遏止关键字垃圾。
PageRank 这个概念引自一篇学术论文的被媒体转载的频度，一般被转载的次数越多，这篇论文的权威性就越高，价值也就越高。PageRank 是 1998 年在斯坦福大学问世的，2001 年 9 月被授予美国专利。如今它在 Google 所有算法中依然是至关重要的。在学术界，这个算法被公认为是文献检索中最大的贡献之一，并且被很多大学引入了信息检索课程(Information Retrieval) 的教程。
PageRank 通过对由超过 5 亿个变量和 20 亿个词汇组成的方程进行计算，能科学公正地标识网页的等级或重要性。PR 级别为 1 到 10，PR 值越高说明该网页越重要。例如：一个 PR 值为 1 的网站表明这个网站不太具有流行度，而 PR 值为 7 到 10 则表明这个网站极其重要。PageRank 级别不是一般的算术级数，而是按照一种几何级数来划分的。PageRank3 不是比 PageRank2 好一级，而可能会好到数倍。
PageRank 根据网站的外部链接和内部链接的数量和质量来衡量网站的价值。 PageRank 的概念是，每个到页面的链接都是对该页面的一次投票，被链接得越多，就意味着被其他网站投票越多。Google 有一套自动化方法来计算这些投票，但 Google 的排名算法不完全基于外部链接。PageRank 对来自不同网页的链接会区别对待，来自网页本身排名高的链接更受青睐，给这些链接有较大的权重。
同时，Google 不只是看一个网站的投票数量，或者这个网站的外部链接数量。它会对那些投票的网站进行分析。如果这些网站的 PR 值比较高，则其投票的网站可从中受益。因此，Google 的技术专家提醒人们，在建设网站的外部链接时，应尽可能瞄准那些 PR 值高且外部链接数又少的网站。这样的外部链接站点越多，你的 PR 值就会越高，从而使得你的 Google 排名得到显著提升。
PageRank 的另一作用是对关键字垃圾起到巨大的遏制作用。眼下，一些垃圾网站为了提高点击率，用一些与站点内容无关的关键字垃圾壮声威，比如用明星的名字、用公共突发事件称谓等。这些网页的目的或是为了骗取广告点击，或是为了传播病毒。还有一些无赖

式的博客评论也从中搅局，在网上招摇过市，骗取网民的注意力，这也被网络技术人员视为垃圾。

PageRank 目前使用一种基于信任和名誉的算法帮助遏止关键字垃圾，它忽视这些关键字垃圾的存在，以网页相互链接评级别论高低。Google 排名之所以大受追捧，是由于它并非只使用关键字或代理搜索技术，而是将自身建立在高级的网页级别技术基础之上。别的搜索引擎提供给搜索者的是多种渠道值为 8 的网站信息得来的一个粗略的搜索结果，而 Google 提供给它的搜索者的则是它自己产生的高度精确的搜索结果。这就是为什么网站管理员会千方百计去提高自己网站在 Google 的排名了。

PageRank 一般一年更新四次，所以刚上线的新网站不可能获得 PR 值。不过 PR 值暂时没有，并不是什么不好的事情，耐心等待就能得到 Google 的青睐。

数据挖掘十大经典算法(7) AdaBoost

Adaboost 是一种迭代算法，其核心思想是针对同一个训练集训练不同的分类器(弱分类器)，然后把这些弱分类器集合起来，构成一个更强的最终分类器 (强分类器)。其算法本身是通过改变数据分布来实现的，它根据每次训练集之中每个样本的分类是否正确，以及上次的总体分类的准确率，来确定每个样本的权值。将修改过权值的新数据集送给下层分类器进行训练，最后将每次训练得到的分类器最后融合起来，作为最后的决策分类器。使用 adaboost 分类器可以排除一些不必要的训练数据特徵，并将关键放在关键的训练数据上面。

目前，对 adaboost 算法的研究以及应用大多集中于分类问题，同时近年也出 现了一些在回归问题上的应用。就其应用 adaboost 系列主要解决了：两类问题、多类单标签问题、多类多标签问题、大类单标签问题，回归问题。它用全部的训练样本进行学习。

该算法其实是一个简单的弱分类算法提升过程，这个过程通过不断的训练，可以提高对数据的分类能力。整个过程如下所示：

1. 先通过对 N 个训练样本的学习得到第一个弱分类器 ；
2. 将 分错的样本和其他的新数据一起构成一个新的 N 个的训练样本，通过对这个样本的学习得到第二个弱分类器；
3. 将 和 都分错了的样本加上其他的新样本构成另一个新的 N 个的训练样本，通过对这个样本的学习得到第三个弱分类器；
4. 最终经过提升的强分类器 。即某个数据被分为哪一类要通过 ， ……的多数表决。

2.3 Adaboost(Adaptive Boosting)算法

对于 boosting 算法，存在两个问题：

1. 如何调整训练集，使得在训练集上训练的弱分类器得以进行；
2. 如何将训练得到的各个弱分类器联合起来形成强分类器。

针对以上两个问题，adaboost 算法进行了调整：

1. 使用加权后选取的训练数据代替随机选取的训练样本，这样将训练的焦点集中在比较难分的训练数据样本上；
2. 将弱分类器联合起来，使用加权的投票机制代替平均投票机制。让分类效果好的弱分类器具有较大的权重，而分类效果差的分类器具有较小的权重。

Adaboost 算法是 Freund 和 Schapire 根据在线分配算法提出的，他们详细分析了 Adaboost 算法错误率 的上界，以及为了使强分类器 达到错误率，算法所需要的最多迭代次数等相关问题。与 Boosting 算法不同的是，adaboost 算法不需要预先知道弱学习算法学习正确率的下限即弱分类器的误差，并且最后得到的强分类器的分类精度依赖于所有弱分类器的分类精度，这样可以深入挖掘弱分类器算法的能力。

Adaboost 算法中不同的训练集是通过调整每个样本对应的权重来实现的。开始时，每个样本对应的权重是相同的，即 其中 n 为样本个数，在此样本分布下训练出一弱分类器 。对于分

类错误的样本，加大其对应的权重；而对于分类正确的样本，降低其权重，这样分错的样本就被突出出来，从而得到一个新的样本分布。在新的样本分布下，再次对弱分类器进行训练，得到弱分类器。依次类推，经过 T 次循环，得到 T 个弱分类器，把这 T 个弱分类器按一定的权重叠加（boost）起来，得到最终想要的强分类器。

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。

数据挖掘十大经典算法(8) kNN

邻近算法

KNN 算法的决策过程 k-Nearest Neighbor algorithm

左图中，绿色圆要被决定赋予哪个类，是红色三角形还是蓝色四方形？如果 K=3，由于红色三角形所占比例为 2/3，绿色圆将被赋予红色三角形那个类，如果 K=5，由于蓝色四方形比例为 3/5，因此绿色圆被赋予蓝色四方形类。

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。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。

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。

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。

数据挖掘十大经典算法(9) Naive Bayes

贝叶斯分类器

贝叶斯分类器的分类原理是通过某对象的先验概率，利用贝叶斯公式计算出其后验概率，即该对象属于某一类的概率，选择具有最大后验概率的类作为该对象所属的类。目前研究较多的贝叶斯分类器主要有四种，分别是：Naive Bayes、TAN、BAN 和 GBN。

贝叶斯网络是一个带有概率注释的有向无环图，图中的每一个结点均表示一个随机变量,图中两结点间若存在着一条弧，则表示这两结点相对应的随机变量是概率相依的，反之则说明这两个随机变量是条件独立的。网络中任意一个结点 X 均有一个相应的条件概率表(Conditional Probability Table，CPT)，用以表示结点 X 在其父结点取各可能值时的条件概率。若结点 X 无父结点,则 X 的 CPT 为其先验概率分布。贝叶斯网络的结构及各结点的 CPT 定义了网络中各变量的概率分布。

贝叶斯分类器是用于分类的贝叶斯网络。该网络中应包含类结点 C，其中 C 的取值来自于类集合(c1 , c2 , ... , cm)，还包含一组结点 X = (X1 , X2 , ... , Xn)，表示用于分类的特征。对于贝叶斯网络分类器，若某一待分类的样本 D，其分类特征值为 x = (x1 , x2 , ... , x n) ，则样本 D 属于类别 ci 的概率 P(C = ci | X1 = x1 , X2 = x 2 , ... , Xn = x n) ，(i = 1 ,2 , ... , m) 应满足下式：

P(C = ci | X = x) = Max{ P(C = c1 | X = x) , P(C = c2 | X = x) , ... , P(C = cm | X = x) }

而由贝叶斯公式：

P(C = ci | X = x) = P(X = x | C = ci) * P(C = ci) / P(X = x)

其中，P(C = ci) 可由领域专家的经验得到,而 P(X = x | C = ci) 和 P(X = x) 的计算则较困难。应用贝叶斯网络分类器进行分类主要分成两阶段。第一阶段是贝叶斯网络分类器的学习，即从样本数据中构造分类器，包括结构学习和 CPT 学习；第二阶段是贝叶斯网络分类器的推理，即计算类结点的条件概率，对分类数据进行分类。这两个阶段的时间复杂性均取决于特征值间的依赖程度，甚至可以是 NP 完全问题，因而在实际应用中，往往需要对贝叶斯网络分类器进行简化。根据对特征值间不同关联程度的假设，可以得出各种贝叶斯分类器，Naive Bayes、TAN、BAN、GBN 就是其中较典型、研究较深入的贝叶斯分类器。

朴素贝叶斯

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。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。

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。
朴素贝叶斯模型：

Vmap=arg max P(Vj | a1,a2...an)
Vj 属于 V 集合
其中 Vmap 是给定一个 example,得到的最可能的目标值.
其中 a1...an 是这个 example 里面的属性.
这里面,Vmap 目标值,就是后面计算得出的概率最大的一个.所以用 max 来表示

贝叶斯公式应用到 P(Vj | a1,a2...an)中.
可得到 Vmap= arg max P(a1,a2...an | Vj) P(Vj) / P (a1,a2...an)
又因为朴素贝叶斯分类器默认 a1...an 他们互相独立的.
所以 P(a1,a2...an)对于结果没有用处. [因为所有的概率都要除同一个东西之后再比较大小,最后结果也似乎影响不大]
可得到 Vmap= arg max P(a1,a2...an | Vj) P(Vj)
然后
"朴素贝叶斯分类器基于一个简单的假定：给定目标值时属性之间相互条件独立。换言之。该假定说明给定实力的目标值情况下。观察到联合的 a1,a2...an 的概率正好是对每个单独属性的概率乘积： P(a1,a2...an | Vj) = Π i P(ai| Vj)
....
朴素贝叶斯分类器：Vnb =arg max P(Vj) Π i P (ai | Vj)
"
Vnb = arg max P (Vj)
此处 Vj (yes | no)，对应天气的例子。
数据挖掘十大经典算法(10) CART
如果一个人必须去选择在很大范围的情形下性能都好的、同时不需要应用开发者付出很多的努力并且易于被终端用户理解的分类技术的话，那么 Brieman, Friedman, Olshen 和 Stone（1984）提出的分类树方法是一个强有力的竞争者。我们将首先讨论这个分类的过程，然后在后续的节中我们将展示这个过程是如何被用来预测连续的因变量。Brieman 等人用来实现这些过程的程序被称为分类和回归树（CART, Classification and Regression Trees）方法。

分类树
在分类树下面有两个关键的思想。第一个是关于递归地划分自变量空间的想法；第二个想法是用验证数据进行剪枝。

递归划分

让我们用变量 y 表示因变量（分类变量），用 x1, x2, x3,...,xp 表示自变量。通过递归的方式把关于变量 x 的 p 维空间划分为不重叠的矩形。这个划分是以递归方式完成的。首先，一个自变量被选择，比如 xi 和 xi 的一个值 si，比方说选择 si 把 p 维空间为两部分：一部分是 p 维的超矩形，其中包含的点都满足 xi<=si，另一个 p 维超矩形包含所 xi>si。接着，这两部分中的一个部分通过选择一个变量和该变量的划分值以相似的方式被划分。这导致了三个矩形区

域（从这里往后我们把超矩形都说成矩形）。随着这个过程的持续，我们得到的矩形越来越小。这个想法是把整个 x 空间划分为矩形，其中的每个小矩形都尽可能是同构的或“纯”的。“纯”的意思是（矩形）所包含的点都属于同一类。我们认为包含的点都只属于一个类（当然，这不总是可能的，因为经常存在一些属于不同类的点，但这些点的自变量有完全相同的值）。

数据挖掘十大经典算法

国际权威的学术组织 the IEEE International Conference on Data Mining (ICDM) 2006 年 12 月评选出了数据挖掘领域的十大经典算法：C4.5, k-Means, SVM, Apriori, EM, PageRank, AdaBoost, kNN, Naive Bayes, and CART.

不仅仅是选中的十大算法，其实参加评选的 18 种算法，实际上随便拿出一种来都可以称得上是经典算法，它们在数据挖掘领域都产生了极为深远的影响。

1. 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。

2. The k-means algorithm 即 K-Means 算法

k-means algorithm 算法是一个聚类算法，把 n 的对象根据他们的属性分为 k 个分割，k < n。它与处理混合正态分布的最大期望算法很相似,因为他们都试图找到数据中自然聚类的中心。它假设对象属性来自于空间向量，并且目标是使各个群组内部的均方误差总和最小。

3. Support vector machines

支持向量机，英文为 Support Vector Machine，简称 SV 机（论文中一般简称 SVM）。它是一种监督式學習的方法，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。

4. The Apriori algorithm

Apriori 算法是一种最有影响的挖掘布尔关联规则频繁项集的算法。其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。

5. 最大期望(EM)算法

在统计计算中，最大期望（EM，Expectation–Maximization）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。

6. PageRank

PageRank 是 Google 算法的重要内容。2001 年 9 月被授予美国专利，专利人是 Google 创始人之一拉里•佩奇（Larry Page）。因此，PageRank 里的 page 不是指网页，而是指佩奇，即这个等级方法是以佩奇来命名的。

PageRank 根据网站的外部链接和内部链接的数量和质量俩衡量网站的价值。PageRank 背后的概念是，每个到页面的链接都是对该页面的一次投票，被链接的越多，就意味着被其他网站投票越多。这个就是所谓的“链接流行度”——衡量多少人愿意将他们的网站和你的网站挂钩。PageRank 这个概念引自学术中一篇论文的被引述的频度——即被别人引述的次数越多，一般判断这篇论文的权威性就越高。

7. AdaBoost

Adaboost 是一种迭代算法，其核心思想是针对同一个训练集训练不同的分类器(弱分类器)，然后把这些弱分类器集合起来，构成一个更强的最终分类器 (强分类器)。其算法本身是通过改变数据分布来实现的，它根据每次训练集之中每个样本的分类是否正确，以及上次的总体分类的准确率，来确定每个样本的权值。将修改过权值的新数据集送给下层分类器进行训练，最后将每次训练得到的分类器最后融合起来，作为最后的决策分类器。

8. kNN: k-nearest neighbor classification

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。

9. Naive Bayes

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）。 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。

10. CART: 分类与回归树

CART, Classification and Regression Trees。 在分类树下面有两个关键的思想。第一个是关于递归地划分自变量空间的想法；第二个想法是用验证数据进行剪枝。
数据挖掘十大经典算法(1)C4.5
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。

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。

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。

决策树同时也可以依靠计算条件概率来构造。决策树如果依靠数学的计算方法可以取得更加理想的效果。

决策树是如何工作的
决策树一般都是自上而下的来生成的。
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。
从根到叶子节点都有一条路径，这条路径就是一条“规则”。
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。相信大家对 ID3 算法都很.熟悉了，这里就不做介绍。

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.

§4.3.2 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.

§4.3.3: 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.

§4.3.4: 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.

数据挖掘十大经典算法(2) k-means

k-means algorithm 算法是一个聚类算法，把 n 的对象根据他们的属性分为 k 个分割，k < n。它与处理混合正态分布的最大期望算法很相似,因为他们都试图找到数据中自然聚类的中心。它假设对象属性来自于空间向量，并且目标是使各个群组内部的均方误差总和最小。

假设有 k 个群组 Si, i=1,2,...,k。μi 是群组 Si 内所有元素 xj 的重心，或叫中心点。

k 平均聚类发明于 1956 年， 该算法最常见的形式是采用被称为劳埃德算法(Lloyd algorithm)的迭代式改进探索法。劳埃德算法首先把输入点分成 k 个初始化分组，可以是随机的或者使用一些启发式数据。然后计算每组的中心点，根据中心点的位置把对象分到离它最近的中心，重新确定分组。继续重复不断地计算中心并重新分组，直到收敛，即对象不再改变分组（中心点位置不再改变）。

劳埃德算法和 k 平均通常是紧密联系的，但是在实际应用中，劳埃德算法是解决 k 平均问题的启发式法则，对于某些起始点和重心的组合，劳埃德算法可能实际上收敛于错误的结果。（上面函数中存在的不同的最优解）

虽然存在变异，但是劳埃德算法仍旧保持流行，因为它在实际中收敛非常快。实际上，观察发现迭代次数远远少于点的数量。然而最近，David Arthur 和 Sergei Vassilvitskii 提出存在特定的点集使得 k 平均算法花费超多项式时间达到收敛。

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。

数据挖掘十大经典算法(3) Svm

支持向量机，英文为 Support Vector Machine，简称 SV 机（论文中一般简称 SVM）。它是一种監督式學習的方法，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。

问题定义

设样本属于两个类，用该样本训练 svm 得到的最大间隔超平面。在超平面上的样本点也称为支持向量.
我们考虑以下形式的样本点

其中 ci 为 1 或−1 --用以表示数据点属于哪个类. 是一个 p − (统计学符号), 或 n − (计算机科学符号) 维向量，其每个元素都被缩放到[0,1]或[-1,1].缩放的目的是防止方差大的随机变量主

导分类过程.我们可以把这些数据称为“训练数据”，希望我们的支持向量机能够通过一个超平面正确的把他们分开。超平面的数学形式可以写作

根据几何知识，我们知道向量垂直于分类超平面。加入位移 b 的目的是增加间隔.如果没有 b 的话，那超平面将不得不通过原点，限制了这个方法的灵活性。

由于我们要求最大间隔，因此我们需要知道支持向量以及（与最佳超平面）平行的并且离支持向量最近的超平面。我们可以看到这些平行超平面可以由方程族：

来表示。

如果这些训练数据是线性可分的，那就可以找到这样两个超平面，在它们之间没有任何样本点并且这两个超平面之间的距离也最大.通过几何不难得到这两个超平面之间的距离是 2/|w|，因此我们需要最小化 |w|。同时为了使得样本数据点都在超平面的间隔区以外，我们需要保证对于所有的 i 满足其中的一个条件

这两个式子可以写作：

原型
现在寻找最佳超平面这个问题就变成了在(1)这个约束条件下最小化|w|.这是一个二次規劃 QP(quadratic programming)最优化中的问题。

更清楚的，它可以表示如下：

最小化 ，满足 。
1/2 这个因子是为了数学上表达的方便加上的。

对偶型(Dual Form)
把原型的分类规则写作对偶型，可以看到分类器其实是一个关于支持向量（即那些在间隔区边缘的训练样本点）的函数。

支持向量机的对偶型如下： 并满足 αi > = 0

软间隔
1995 年, Corinna Cortes 与 Vapnik 提出了一种改进的最大间隔区方法，这种方法可以处理标

记错误的样本。如果可区分正负例的超平面不存在，则“软边界”将选择一个超平面尽可能清晰地区分样本，同时使其与分界最清晰的样本的距离最大化。这一成果使术语“支持向量机”（或“SVM”）得到推广。这种方法引入了松驰参数 ξi 以衡量对数据 xi 的误分类度。

。
随后，将目标函数与一个针对非 0ξi 的惩罚函数相加，在增大间距和缩小错误惩罚两大目标之间进行权衡优化。如果惩罚函数是一个线性函数，则等式(3)变形为
数据挖掘十大经典算法(4)Apriori
Apriori 算法是一种最有影响的挖掘布尔关联规则频繁项集的算法。其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。

Apriori 演算法所使用的前置统计量包括了：

最大规则物件数：规则中物件组所包含的最大物件数量
最小支援：规则中物件或是物件组必顸符合的最低案例数
最小信心水准：计算规则所必须符合的最低信心水准门槛
该算法的基本思想是：首先找出所有的频集，这些项集出现的频繁性至少和预定义的最小支持度一样。然后由频集产生强关联规则，这些规则必须满足最小支持度和最小可信度。然后使用第 1 步找到的频集产生期望的规则，产生只包含集合的项的所有规则，其中每一条规则的右部只有一项，这里采用的是中规则的定义。一旦这些规则被生成，那么只有那些大于用户给定的最小可信度的规则才被留下来。为了生成所有频集，使用了递推的方法。
可能产生大量的候选集,以及可能需要重复扫描数据库，是 Apriori 算法的两大缺点。
数据挖掘十大经典算法(5) EM
在统计计算中，最大期望（EM，Expectation–Maximization）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。

最大期望过程说明
我们用 表示能够观察到的不完整的变量值，用 表示无法观察到的变量值，这样 和 一起组成了完整的数据。 可能是实际测量丢失的数据，也可能是能够简化问题的隐藏变量，如果它的值能够知道的话。例如，在混合模型（Mixture Model）中，如果“产生”样本的混合元素成分已知的话最大似然公式将变得更加便利（参见下面的例子）。

估计无法观测的数据
让 代表矢量 θ: 定义的参数的全部数据的概率分布（连续情况下）或者概率集聚函数（离散情况下），那么从这个函数就可以得到全部数据的最大似然值，另外，在给定的观察到的数据条件下未知数据的条件分布可以表示为：
数据挖掘十大经典算法(6) PageRank

PageRank 是 Google 算法的重要内容。2001 年 9 月被授予美国专利，专利人是 Google 创始人之一拉里•佩奇（Larry Page）。因此，PageRank 里的 page 不是指网页，而是指佩奇，即这个等级方法是以佩奇来命名的。

Google 的 PageRank 根据网站的外部链接和内部链接的数量和质量俩衡量网站的价值。PageRank 背后的概念是，每个到页面的链接都是对该页面的一次投票，被链接的越多，就意味着被其他网站投票越多。这个就是所谓的“链接流行度”——衡量多少人愿意将他们的网站和你的网站挂钩。PageRank 这个概念引自学术中一篇论文的被引述的频度——即被别人引述的次数越多，一般判断这篇论文的权威性就越高。

Google 有一套自动化方法来计算这些投票。Google 的 PageRank 分值从 0 到 10；PageRank 为 10 表示最佳，但非常少见，类似里氏震级（Richter scale），PageRank 级别也不是线性的，而是按照一种指数刻度。这是一种奇特的数学术语，意思是 PageRank4 不是比 PageRank3 好一级——而可能会好 6 到 7 倍。因此，一个 PageRank5 的网页和 PageRank8 的网页之间的差距会比你可能认为的要大的多。

PageRank 较高的页面的排名往往要比 PageRank 较低的页面高，而这导致了人们对链接的着魔。在整个 SEO 社区，人们忙于争夺、交换甚至销售链接，它是过去几年来人们关注的焦点，以至于 Google 修改了他的系统，并开始放弃某些类型的链接。比如，被人们广泛接受的一条规定，来自缺乏内容的“link farm”（链接工厂）网站的链接将不会提供页面的 PageRank，从 PageRank 较高的页面得到链接但是内容不相关（比如说某个流行的漫画书网站链接到一个叉车规范页面），也不会提供页面的 PageRank。Google 选择降低了 PageRank 对更新频率，以便不鼓励人们不断的对其进行监测。

Google PageRank 一般一年更新四次，所以刚上线的新网站不可能获得 PR 值。你的网站很可能在相当长的时间里面看不到 PR 值的变化，特别是一些新的网站。PR 值暂时没有，这不是什么不好的事情，耐心等待就好了。

为您的网站获取外部链接是一件好事，但是无视其他 SEO 领域的工作而进行急迫的链接建设就是浪费时间，要时刻保持一个整体思路并记住以下几点：

•Google 的排名算法并不是完全基于外部链接的

•高 PageRank 并不能保证 Google 高排名

•PageRank 值更新的比较慢，今天看到的 PageRank 值可能是三个月前的值

因此我们不鼓励刻意的去追求 PageRank，因为决定排名的因素可以有上百种。尽管如此，PageRank 还是一个用来了解 Google 对您的网站页面如何评价的相当好的指示，建议网站设计者要充分认识 PageRank 在 Google 判断网站质量中的重要作用，从设计前的考虑到后期网站更新都要给予 PageRank 足够的分析，很好的利用。我们要将 PageRank 看作是一种业余爱好而不是一种信仰。

通过对由超过 50,000 万个变量和 20 亿个词汇组成的方程进行计算，PageRank 能够对网页的重要性做出客观的评价。PageRank 并不计算直接链接的数量，而是将从网页 A 指向网页 B 的链接解释为由网页 A 对网页 B 所投的一票。这样，PageRank 会根据网页 B 所收到的投票数量来评估该页的重要性。

此外，PageRank 还会评估每个投票网页的重要性，因为某些网页的投票被认为具有较高的价值，这样，它所链接的网页就能获得较高的价值。重要网页获得的 PageRank（网页排名）较高，从而显示在搜索结果的顶部。Google 技术使用网上反馈的综合信息来确定某个网页的重要性。搜索结果没有人工干预或操纵，这也是为什么 Google 会成为一个广受用户信赖、不受付费排名影响且公正客观的信息来源。

————————————————

其实简单说就是民主表决。打个比方，假如我们要找李开复博士，有一百个人举手说自己是李开复。那么谁是真的呢？也许有好几个真的，但即使如此谁又是大家真正想找的呢？:-) 如果大家都说在 Google 公司的那个是真的，那么他就是真的。

在互联网上，如果一个网页被很多其它网页所链接，说明它受到普遍的承认和信赖，那么它的排名就高。这就是 Page Rank 的核心思想。 当然 Google 的 Page Rank 算法实际上要复杂得多。比如说，对来自不同网页的链接对待不同，本身网页排名高的链接更可靠，于是给这些链接予较大的权重。Page Rank 考虑了这个因素，可是现在问题又来了，计算搜索结果的网页排名过程中需要用到网页本身的排名，这不成了先有鸡还是先有蛋的问题了吗？

Google 的两个创始人拉里•佩奇 （Larry Page ）和谢尔盖•布林 (Sergey Brin) 把这个问题变成了一个二维矩阵相乘的问题，并且用迭代的方法解决了这个问题。他们先假定所有网页的排名是相同的，并且根据这个初始值，算出各个网页的第一次迭代排名，然后再根据第一次迭代排名算出第二次的排名。他们两人从理论上证明了不论初始值如何选取，这种算法都保证了网页排名的估计值能收敛到他们的真实值。值得一提的事，这种算法是完全没有任何人工干预的。

理论问题解决了，又遇到实际问题。因为互联网上网页的数量是巨大的，上面提到的二维矩阵从理论上讲有网页数目平方之多个元素。如果我们假定有十亿个网页，那么这个矩阵就有一百亿亿个元素。这样大的矩阵相乘，计算量是非常大的。拉里和谢尔盖两人利用稀疏矩阵计算的技巧，大大的简化了计算量，并实现了这个网页排名算法。今天 Google 的工程师把这个算法移植到并行的计算机中，进一步缩短了计算时间，使网页更新的周期比以前短了许多。

我来 Google 后，拉里 (Larry) 在和我们几个新员工座谈时，讲起他当年和谢尔盖(Sergey) 是怎么想到网页排名算法的。他说："当时我们觉得整个互联网就像一张大的图（Graph)，每个网站就像一个节点，而每个网页的链接就像一个弧。我想，互联网可以用一个图或者矩阵描述，我也许可以用这个发现做个博士论文。" 他和谢尔盖就这样发明了 Page Rank 的算法。网页排名的高明之处在于它把整个互联网当作了一个整体对待。它无意识中符合了系统论的观点。相比之下，以前的信息检索大多把每一个网页当作独立的个体对待，很多人当初只注意了网页内容和查询语句的相关性，忽略了网页之间的关系。

今天，Google 搜索引擎比最初复杂、完善了许多。但是网页排名在 Google 所有算法中依然是至关重要的。在学术界，这个算法被公认为是文献检索中最大的贡献之一，并且被很多大学引入了信息检索课程 (Information Retrieval) 的教程。

如何提高你网页的 PR 值？

什么是 PR 值呢？PR 值全称为 PageRank，PR 是英文 Pagerank 的缩写形式，Pagerank 取自 Google 的创始人 LarryPage，它是 Google 排名运算法则（排名公式）的一部分，Pagerank 是 Google 对网页重要性的评估，是 Google 用来衡量一个网站的好坏的唯一标准。PageRank(网页级别)是 Google 用于评测一个网页“重要性”的一种方法。在揉合了诸如 Title 标识和 Keywords 标识等所有其它因素之后，Google 通过 PageRank 来调整结果，使那些更具“重要性”的网页在搜索结果中另网站排名获得提升，从而提高搜索结果的相关性和质量。 PR 值的级别从 1 到 10 级，10 级为满分。PR 值越高说明该网页越受欢迎。Google 把自己的网站的 PR 值定到 10，这说明 Google 这个网站是非常受欢迎的，也可以说这个网站非常重要。Google 大受青睐的另一个原因就是它的网站索引速度。向 Google 提交你的网站直到为 Google 收录，一般只需两个星期。如果你的网站已经为 Google 收录，那么通常 Google 会每月一次遍历和更新(重新索引)你的网站信息。不过对于那些 PR 值 (Pagerank)较高的网站，Google 索引周期会相应的短一些。一个 PR 值为 1 的网站表明这个网站不太具有流行度，而 PR 值为 7 到 10 则表明这个网站非常受欢迎。PR 值最高为 10，一般 PR 值达到 4，就算是一个不错的网站了。

那么 PR 值都受那些因素影响呢？下面我们一起来看看。

第一：网站外部链接的数量和质量

在计算网站排名时，Pagerank 会将网站的外部链接数考虑进去。并不能说一个网站的外部链接数越多其 PR 值就越高，如果这样的话，一个网站尽可能获得最多的外部链接就 OK 了，有这种想法是错误的。Google 对一个网站上的外部链接数的重视程度并不意味着你因此可以不求策略地与任何网站建立连接。这是因为 Google 并不是简单地由计算网站的外部链接数来决定其等级。Google 的 Pagerank 系统不单考虑一个网站的外部链接质量，也会考虑其数量。这个问题看来很有复杂。首先让我们来解释一下什么是阻尼因数(damping factor)。阻尼因素就是当你投票或链接到另外一个站点时所获得的实际 PR 分值。阻尼因数一般是 0.85。当然比起你网站的实际 PR 值，它就显得微不足道了。

现在让我们来看看这个 PR 分值的计算公式：PR(A)=(1- d)+d(PR(t1)/C(t1)+...+PR(tn)/C(tn)) 公式解释：其中 PR(A)表示的是从一个外部链接站点 t1 上，依据 Pagerank?系统给你的网站所增加的 PR 分值；PR(t1)表示该外部链接网站本身的 PR 分值；C(t1)则表示该外部链接站点所拥有的外部链接数量。大家要谨记：一个网站的投票权值只有该网站 PR 分值的 0.85，

那么，是不是说对一个网站而言，它所拥有的较高网站质量和较高 PR 分值的外部链接数量越多就越好呢？错，因为－Google 的 Pagerank 系统不单考虑一个网站的外部链接质量，也会考虑其数量.比方说，对一个有一定 PR 值的网站 X 来说，如果你的网站 Y 是它的唯一一个外部链接，那么 Google 就相信网站 X 将你的网站 Y 视做它最好的一个外部链接，从而会给你的网站 Y 更多的分值。可是，如果网站 X 上已经有 49 个外部链接，那么 Google 就相信网站 X 只是将你的网站视做它第 50 个好的网站。因而你的外部链接站点上的外部链接数越多，你所能够得到的 PR 分值反而会越低，它们呈反比关系。

说它对是因为－一般情况下，一个 PR 分值大于等于 6 的外部链接站点，可显著提升你的 PR 分值。但如果这个外部链接站点已经有 100 个其它的外部链接时，那你能够得到的 PR 分值就几乎为零了。同样，如果一个外部链接站点的 PR 值仅为 2，但你却是它的唯一一个外部链接，那么你所获得的 PR 值要远远大于那个 PR 值为 6，外部链接数为 100 的网站。

而且这个 0.85 的权值平均分配给其链接的每个外部网站。

第二：Google 在你的网站抓取的页面数

Google 在你的网站抓取的页面数，数目越多，Pagerank 值越高。但通常 Google 并不会主动抓取你的网站的所有页面，尤其是网址里带有“?”的动态链接，Google 不主动，那就要我们主动了，最笨的办法是把网站所有的页面都提交给 Google，但我想没有谁真会这么做，但页面不多的话可以试试。更好的办法是制作一个静态 Html 页面，通常被称作“网站地图”或“网站导航”，它里面包含你要添加的所有网址，然后把这个静态页面提交给 Google。

第三：网站被世界三大知名网站 DMOZ，Yahoo 和 Looksmart 收录

众所周知，Google 的 Pagerank 系统对那些门户网络目录如 DMOZ，Yahoo 和 Looksmart 尤为器重。特别是对 DMOZ。一个网站上的 DMOZ 链接对 Google 的 Pagerank?来说，就好像一块金子一样珍贵。如果你的网站为 ODP 收录，则可有效提升你的页面等级。向 ODP 提交你的站点并为它收录，其实并不是一件难事，只是要多花点时间而已。只要确保你的网站提供了良好的内容，然后在 ODP 合适的目录下点击"增加站点"，按照提示一步步来就 OK 了。至少要保证你的索引页(INDEX PAGE)被收录进去。所以，如果你的网站内容涉及完全不同的几块内容，你可以把每个内容的网页分别向 ODP 提交－不过请记住"欲速则不达"。等到 Google 对其目录更新后，你就能看到你的 PR 值会有什么变化了。如果你的网站为 Yahoo 和 Looksmart 所收录，那么你的 PR 值会得到显著提升。如果你的网站是非商业性质的或几乎完全是非商业性质的内容，那么你可以通过 zeall.com 使你的网站为著名的网络目录 Looksmart 所收录。 Looksmart 也是从 Zeal 网络目录获得非商业搜索列表。

Google PR 值的更新周期是多长时间？
一般情况下 PR 值更新的周期是 2.5～3 个月！最近一次 PR 更新是 2008 年 1 月中旬。
PageRank 相关算法总结：
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。
那么影响 google PageRank 的因素有哪些呢?
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
6google 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
10Google 抓取您网站的页面数量
11 导出链接数量
PageRank 科学排名遏止关键字垃圾
目前，五花八门的网站为争夺网上排名采用恶意点击和输入关键字垃圾的手段来吸引网民的眼球，无论对于互联网企业还是互联网用户，这都不是一个好现象。
为了解决这样的问题，Google 创始人之一拉里.佩奇（Larry Page）发明了一种算法 PageRank，是由搜索引擎根据网页之间相互的超链接进行计算的网页排名。它经常和搜索引擎优化有关。PageRank 系统目前被 Google 用来体现网页的相关性和重要性，以便科学排名，遏止关键字

垃圾。

PageRank 这个概念引自一篇学术论文的被媒体转载的频度，一般被转载的次数越多，这篇论文的权威性就越高，价值也就越高。PageRank 是 1998 年在斯坦福大学问世的，2001 年 9 月被授予美国专利。如今它在 Google 所有算法中依然是至关重要的。在学术界，这个算法被公认为是文献检索中最大的贡献之一，并且被很多大学引入了信息检索课程(Information Retrieval) 的教程。

PageRank 通过对由超过 5 亿个变量和 20 亿个词汇组成的方程进行计算，能科学公正地标识网页的等级或重要性。PR 级别为 1 到 10，PR 值越高说明该网页越重要。例如：一个 PR 值为 1 的网站表明这个网站不太具有流行度，而 PR 值为 7 到 10 则表明这个网站极其重要。PageRank 级别不是一般的算术级数，而是按照一种几何级数来划分的。PageRank3 不是比 PageRank2 好一级，而可能会好到数倍。

PageRank 根据网站的外部链接和内部链接的数量和质量来衡量网站的价值。 PageRank 的概念是，每个到页面的链接都是对该页面的一次投票，被链接得越多，就意味着被其他网站投票越多。Google 有一套自动化方法来计算这些投票，但 Google 的排名算法不完全基于外部链接。PageRank 对来自不同网页的链接会区别对待，来自网页本身排名高的链接更受青睐，给这些链接有较大的权重。

同时，Google 不只是看一个网站的投票数量，或者这个网站的外部链接数量。它会对那些投票的网站进行分析。如果这些网站的 PR 值比较高，则其投票的网站可从中受益。因此，Google 的技术专家提醒人们，在建设网站的外部链接时，应尽可能瞄准那些 PR 值高且外部链接数又少的网站。这样的外部链接站点越多，你的 PR 值就会越高，从而使得你的 Google 排名得到显著提升。

PageRank 的另一作用是对关键字垃圾起到巨大的遏制作用。眼下，一些垃圾网站为了提高点击率，用一些与站点内容无关的关键字垃圾壮声威，比如用明星的名字、用公共突发事件称谓等。这些网页的目的或是为了骗取广告点击，或是为了传播病毒。还有一些无赖式的博客评论也从中搅局，在网上招摇过市，骗取网民的注意力，这也被网络技术人员视为垃圾。

PageRank 目前使用一种基于信任和名誉的算法帮助遏止关键字垃圾，它忽视这些关键字垃圾的存在，以网页相互链接评级别论高低。Google 排名之所以大受追捧，是由于它并非只使用关键字或代理搜索技术，而是将自身建立在高级的网页级别技术基础之上。别的搜索引擎提供给搜索者的是多种渠道值为 8 的网站信息得来的一个粗略的搜索结果，而 Google 提供给它的搜索者的则是它自己产生的高度精确的搜索结果。这就是为什么网站管理员会千方百计去提高自己网站在 Google 的排名了。

PageRank 一般一年更新四次，所以刚上线的新网站不可能获得 PR 值。不过 PR 值暂时没有，并不是什么不好的事情，耐心等待就能得到 Google 的青睐。

数据挖掘十大经典算法(7) AdaBoost

Adaboost 是一种迭代算法，其核心思想是针对同一个训练集训练不同的分类器(弱分类器)，然后把这些弱分类器集合起来，构成一个更强的最终分类器 (强分类器)。其算法本身是通过改变数据分布来实现的，它根据每次训练集之中每个样本的分类是否正确，以及上次的总体分类的准确率，来确定每个样本的权值。将修改过权值的新数据集送给下层分类器进行训练，最后将每次训练得到的分类器最后融合起来，作为最后的决策分类器。使用 adaboost 分类器可以排除一些不必要的训练数据特徵，并将关键放在关键的训练数据上面。

目前，对 adaboost 算法的研究以及应用大多集中于分类问题，同时近年也出 现了一些在回归问题上的应用。就其应用 adaboost 系列主要解决了: 两类问题、 多类单标签问题、多类多标签问题、大类单标签问题，回归问题。它用全部的训练样本进行学习。

该算法其实是一个简单的弱分类算法提升过程，这个过程通过不断的训练，可以提高对数据的分类能力。整个过程如下所示：

1. 先通过对 N 个训练样本的学习得到第一个弱分类器 ；
2. 将 分错的样本和其他的新数据一起构成一个新的 N 个的训练样本，通过对这个样本的学习得到第二个弱分类器；
3. 将 和 都分错了的样本加上其他的新样本构成另一个新的 N 个的训练样本，通过对这个样本的学习得到第三个弱分类器；
4. 最终经过提升的强分类器 。即某个数据被分为哪一类要通过 ， ……的多数表决。

2.3 Adaboost(Adaptive Boosting)算法

对于 boosting 算法，存在两个问题：

1. 如何调整训练集，使得在训练集上训练的弱分类器得以进行；
2. 如何将训练得到的各个弱分类器联合起来形成强分类器。

针对以上两个问题，adaboost 算法进行了调整：

1. 使用加权后选取的训练数据代替随机选取的训练样本，这样将训练的焦点集中在比较难分的训练数据样本上；
2. 将弱分类器联合起来，使用加权的投票机制代替平均投票机制。让分类效果好的弱分类器具有较大的权重，而分类效果差的分类器具有较小的权重。

Adaboost 算法是 Freund 和 Schapire 根据在线分配算法提出的，他们详细分析了 Adaboost 算法错误率 的上界，以及为了使强分类器 达到错误率，算法所需要的最多迭代次数等相关问题。与 Boosting 算法不同的是，adaboost 算法不需要预先知道弱学习算法学习正确率的下限即弱分类器的误差，并且最后得到的强分类器的分类精度依赖于所有弱分类器的分类精度，这样可以深入挖掘弱分类器算法的能力。

Adaboost 算法中不同的训练集是通过调整每个样本对应的权重来实现的。开始时，每个样本对应的权重是相同的，即 其中 n 为样本个数，在此样本分布下训练出一弱分类器 。对于分类错误的样本，加大其对应的权重；而对于分类正确的样本，降低其权重，这样分错的样本就被突出出来，从而得到一个新的样本分布。在新的样本分布下，再次对弱分类器进行训练，得到弱分类器。依次类推，经过 T 次循环，得到 T 个弱分类器，把这 T 个弱分类器按一定的权重叠加（boost）起来，得到最终想要的强分类器。

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。

数据挖掘十大经典算法(8) kNN

邻近算法

KNN 算法的决策过程 k-Nearest Neighbor algorithm

左图中，绿色圆要被决定赋予哪个类，是红色三角形还是蓝色四方形？如果 K=3，由于红色三角形所占比例为 2/3，绿色圆将被赋予红色三角形那个类，如果 K=5，由于蓝色四方形比例为 3/5，因此绿色圆被赋予蓝色四方形类。

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。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。

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。

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。

数据挖掘十大经典算法(9) Naive Bayes

贝叶斯分类器

贝叶斯分类器的分类原理是通过某对象的先验概率，利用贝叶斯公式计算出其后验概率，即该对象属于某一类的概率，选择具有最大后验概率的类作为该对象所属的类。目前研究较多的贝叶斯分类器主要有四种，分别是：Naive Bayes、TAN、BAN 和 GBN。

贝叶斯网络是一个带有概率注释的有向无环图，图中的每一个结点均表示一个随机变量,图中两结点间若存在着一条弧，则表示这两结点相对应的随机变量是概率相依的，反之则说明这两个随机变量是条件独立的。网络中任意一个结点 X 均有一个相应的条件概率表(Conditional Probability Table，CPT)，用以表示结点 X 在其父结点取各可能值时的条件概率。若结点 X 无父结点,则 X 的 CPT 为其先验概率分布。贝叶斯网络的结构及各结点的 CPT 定义了网络中各变量的概率分布。

贝叶斯分类器是用于分类的贝叶斯网络。该网络中应包含类结点 C，其中 C 的取值来自于类集合(c1 , c2 , ... , cm)，还包含一组结点 X = (X1 , X2 , ... , Xn)，表示用于分类的特征。对于贝叶斯网络分类器，若某一待分类的样本 D，其分类特征值为 x = (x1 , x2 , ... , x n) ，则样本 D 属于类别 ci 的概率 P(C = ci | X1 = x1 , X2 = x 2 , ... , Xn = x n) ，(i = 1 ,2 , ... , m) 应满足下式：

P(C = ci | X = x) = Max{ P(C = c1 | X = x) , P(C = c2 | X = x) , ... , P(C = cm | X = x) }

而由贝叶斯公式：

P(C = ci | X = x) = P(X = x | C = ci) * P(C = ci) / P(X = x)

其中，P(C = ci) 可由领域专家的经验得到,而 P(X = x | C = ci) 和 P(X = x) 的计算则较困难。应用贝叶斯网络分类器进行分类主要分成两阶段。第一阶段是贝叶斯网络分类器的学习，即

从样本数据中构造分类器，包括结构学习和 CPT 学习；第二阶段是贝叶斯网络分类器的推理，即计算类结点的条件概率，对分类数据进行分类。这两个阶段的时间复杂性均取决于特征值间的依赖程度，甚至可以是 NP 完全问题，因而在实际应用中，往往需要对贝叶斯网络分类器进行简化。根据对特征值间不同关联程度的假设，可以得出各种贝叶斯分类器，Naive Bayes、TAN、BAN、GBN 就是其中较典型、研究较深入的贝叶斯分类器。

朴素贝叶斯

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。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。

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。

朴素贝叶斯模型：

Vmap=arg max P(Vj | a1,a2...an)

Vj 属于 V 集合

其中 Vmap 是给定一个 example,得到的最可能的目标值.

其中 a1...an 是这个 example 里面的属性.

这里面,Vmap 目标值,就是后面计算得出的概率最大的一个.所以用 max 来表示

贝叶斯公式应用到 P(Vj | a1,a2...an)中.

可得到 Vmap= arg max P(a1,a2...an | Vj) P(Vj) / P (a1,a2...an)

又因为朴素贝叶斯分类器默认 a1...an 他们互相独立的.

所以 P(a1,a2...an)对于结果没有用处. [因为所有的概率都要除同一个东西之后再比较大小,最后结果也似乎影响不大]

可得到 Vmap= arg max P(a1,a2...an | Vj) P(Vj)

然后

"朴素贝叶斯分类器基于一个简单的假定：给定目标值时属性之间相互条件独立。换言之。该假定说明给定实力的目标值情况下。观察到联合的 a1,a2...an 的概率正好是对每个单独属性的概率乘积： P(a1,a2...an | Vj) = Π i P(ai| Vj)

....

朴素贝叶斯分类器：Vnb =arg max P(Vj) Π i P (ai | Vj)
"
Vnb = arg max P (Vj)
此处 Vj (yes | no)，对应天气的例子。
数据挖掘十大经典算法(10) CART
如果一个人必须去选择在很大范围的情形下性能都好的、同时不需要应用开发者付出很多的努力并且易于被终端用户理解的分类技术的话，那么 Brieman, Friedman, Olshen 和 Stone（1984）提出的分类树方法是一个强有力的竞争者。我们将首先讨论这个分类的过程，然后在后续的节中我们将展示这个过程是如何被用来预测连续的因变量。Brieman 等人用来实现这些过程的程序被称为分类和回归树（CART, Classification and Regression Trees）方法。

分类树
在分类树下面有两个关键的思想。第一个是关于递归地划分自变量空间的想法；第二个想法是用验证数据进行剪枝。

递归划分
让我们用变量 y 表示因变量（分类变量），用 x1, x2, x3,...,xp 表示自变量。通过递归的方式把关于变量 x 的 p 维空间划分为不重叠的矩形。这个划分是以递归方式完成的。首先，一个自变量被选择，比如 xi 和 xi 的一个值 si，比方说选择 si 把 p 维空间为两部分：一部分是 p 维的超矩形，其中包含的点都满足 xi<=si，另一个 p 维超矩形包含所有的点满足 xi>si。接着，这两部分中的一个部分通过选择一个变量和该变量的划分值以相似的方式被划分。这导致了三个矩形区域（从这里往后我们把超矩形都说成矩形）。随着这个过程的持续，我们得到的矩形越来越小。这个想法是把整个 x 空间划分为矩形，其中的每个小矩形都尽可能是同构的或“纯”的。“纯” 的意思是（矩形）所包含的点都属于同一类。我们认为包含的点都只属于一个类（当然，这不总是可能的，因为经常存在一些属于不同类的点，但这些点的自变量有完全相同的值）。
有的点满足