最近邻居和聚集（Nearest Neighbor and Clustering）

距离近：在一些重要的属性上比较相似

聚集（clustering）：是把相似的记录放在一起。

用途

聚集

让用户在较高的层次上观察数据库。常被用来做商业上的*顾客分片*（segmentation）。

找到不能与其他记录集合在一起的记录，做例外分析。

最近邻居

预测，距离相近的对象通常他们的预测值也相似，因此只要知道一个对象的预测值，就可以用他来预测他的邻居的值。

分数卡

基本思想

一般来说一个数据库没有一种最好的分类方法。聚集要在类中对象的相似程度和类的数目之间

找到一个最佳的结合点。

N维空间和距离

变量（字段）的个数作为空间的维数。

基本的距离定义有两种：*Manhatan距离* $\sum |a-b|$、*欧氏距离* $(\sum (a-b)^2)^{1/2}$

决定变量权重的方法：

1. 按照实际问题中各个变量对预测值的影响程度
2. 用进化的办法，修改各个变量的权重，看是否能提高预测的准确率。

在文本挖掘中：1 用单词出现频率的倒数；2 按照各个单词对要检索内容的相关程度

怎样计算两个类的距离：

1. 单连通方法（single-link method）：取两个类中最近记录的距离为类的距离。此种方法可以生成细长的蛇形类，不适于应用在典型的一堆堆记录集合在一起的情况。
2. 完全连通方法（complete-link method）：取两个类中最远记录的距离为类的距离。同第1中方法相反，此种方法易生成很小的记录都聚集在一起的类。
3. 平均连通方法（group-average link）：计算两个类中所有记录对的距离平均值。效果介于1、2种算法之间。
4. Ward方法（Ward’s method）：计算两个类中所有记录的距离的和。易于用在生成类层次的情况，对例外的数据（outliers）很敏感，很难应用于生成蛇形类。

聚集的分类和算法流程

分层的聚集（hierarchy）：生成一个从小到大的聚集层次树。用户可以自由剪切这棵树，得到对数据的不同划分方法

合并（Agglomerative）：从下到上，最初每个记录都是一类，逐步合并，直到合并成一个大类。

1. 令数据库中的每一条记录都是一个类

2. 把距离最近的类合并

3. 重复2直到只含唯一的一个类为止

分割（Divisive）：从上到下，一开始所有记录属于一个大类，逐步分割每一个类，直到不能分割为止。因选择分割哪一个类需要很大的计算量，此种方法很少使用。

1. 令数据库中的所有记录都属于一个类

2. 在所有的类中找到一个类中数据相似性最小的一个类，把他一分为二

3. 重复2直到每个类中记录的个数都是1或达到一个预先设定的阈值，或类的个数已达到预先设定的最大个数

不分层聚集：速度更快，但需要用户在使用前设定一些参数，如类的个数、同一类中记录的最大距离。有时要反复修改参数才能得到一种满意的分类方法。

一次通过法（single-pass methods）：只扫描数据库一次，就可完成分类。

1. 从数据库中读取一条记录，判断他距哪个类的距离最近

2. 如果即使到最近的类的距离比我们设定的距离相比还远，那么建立一个新的类，把此记录放到此类中去。

3. 如果数据库中还有记录转1。

问题：数据库中记录的输入顺序和类内最大距离的设定，对分类的结果影响很大。

再分配法（reallocation methods）：要把一条记录从一个类拿出来重新分到另一个类。

1. 预先设定想要把数据分成类的个数

2. 为每个类随机选取一条数据，作为类的中心或“ 种子 ”

3. 一次读取数据库中的每一条记录，将其归到距离最近的类。

4. 重新计算各个类的中心

5. 重复3-4，直到类的中心不再变化或变化很小

问题：用户设定的类的个数很难与实际数据中存在的类的个数正好相符

最近邻居用于预测

方法

1. 找到数据库中距离最近记录，将此记录的值作为新记录的预测值
2. 找到最近的K个记录，用这K个记录按其到新记录的距离作为权重，综合得到新记录的预测值。

缺点

1. 模型太大，预测时要使用整个历史数据库
2. 没有正规的用于防止overfitting的方法（formal way）

模型的改进：删除用于预言的历史数据库中多余的数据，以得到数据量小而且准确度高的数据。

1. 合并相似的记录，用一条记录（称为原型）代替相似的几条记录。要在不降低预测准确率的前提下。
2. 只保留一组相似数据中的“ 边界 ”数据（称为“ 哨兵 ”），去掉“ 边界 ”内部的无用数据。

发展方向

1. 应用算法到新的领域
2. 改进输入变量权重的计算方法，和如何减小用于预测的历史数据的大小
3. 根据历史记录自动计算变量的权重