汉明距离

1. 汉明码

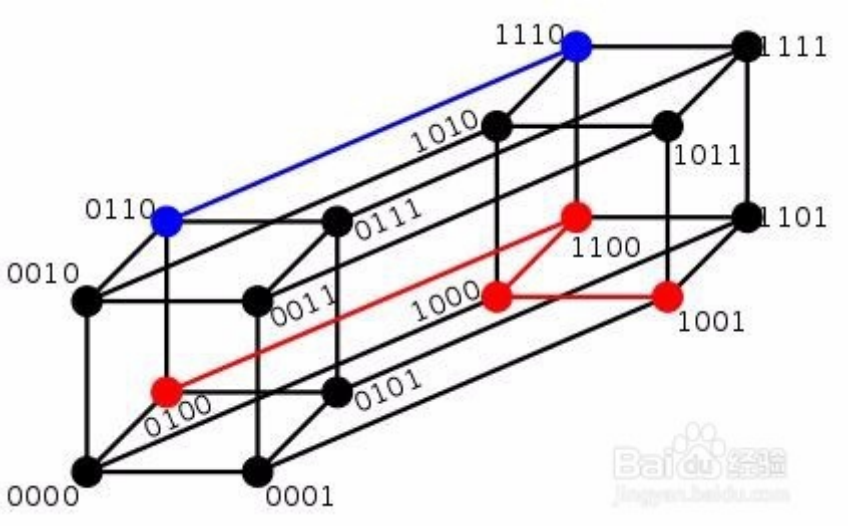
汉明码(Hamming Code)，是在电信领域的一种线性调试码，汉明码在传输的消息流中插入**验证码**，当计算机存储或移动数据时，可能会产生数据位错误，以**侦测**并**更正**单一比特错误。

2. 汉明距离

两个等长字符串之间的汉明距离是两个字符串对应位置的不同字符的个数。换句话说，它就是将一个字符串变换成另外一个字符串所需要替换的字符个数。即：A xor B。

例如：A=0110 与 B=1110 A xor B=1 A和B的汉明距离为1。

A=0100 与 B=1001 A xor B=3 A和B的汉明距离为3。



3.汉明重量

汉明重量是字符串相对于同样长度的零字符串的汉明距离，也就是说，它是字符串中非零的元素个数:对于二进制字符串来说，就是 1 的个数，所以 11101 的汉明重量是 4。

4.最小汉明距离

在一个码组集合中，任意两个码字之间对应位上码元取值不同的位的数目定义为这两个码字之间的汉明距离。
例如：(00)与(01)的距离是1，(110)和(101)的距离是2。在一个码组集合中，任意两个编码之间汉明距离的最小值称为这个码组的最小汉明距离。**最小汉明距离越大，码组越具有抗干扰能力越强。**

5.汉明距离的应用

汉明距离更多的用于信号处理，表明一个信号变成另一个信号需要的最小操作(替换位)，实际中就是比较两个比特串有多少个位不一样，简洁的操作时就是两个比特串进行异或之后包含1的个数。汉明距在图像处理领域也有这广泛的应用，是**比较二进制图像**非常有效的手段。其在包括**信息论、编码理论、密码学**等领域都有应用。

6. 利用*Python numpy*计算汉明码

```
from numpy import *
matV = mat([1,1,1,1],[1,0,0,1])
smstr = nonzero(matV[0]-matV[1])
print smstr
```

*Jaccard*相似数

1.*Jaccard*系数的定义

给定两个集合A,B，*Jaccard*系数定义为**A与B交集的大小与A与B并集的大小的比值**，定义如下：

$$J(A,B)=\frac{|A\cap B|}{|A\cup B|}=\frac{|A\cap B|}{|A|+|B|-|A\cap B|}$$

当集合A，B都为空时，J(A,B)定义为1。

与*Jaccard*系数相关的指标叫做*Jaccard* 距离，用于描述集合之间的不相似度。*Jaccard* 距离越大，样本相似度越低。公式定义如下：

$$d_j(A,B)=1-J(A,B)=\frac{|A\cup B|-|A\cap B|}{|A\cup B|}=\frac{A\Delta B}{|A\cup B|}$$

其中对参差（symmetric difference）

$A\Delta B=|A\cup B|-|A\cap B|$。

性质

$$J(A,B)\in[0,1]$$

2.*Jaccard*系数主要的应用的场景有

1.过滤相似度很高的新闻，或者网页去重。

2.考试防作弊系统。

3.论文查重系统。

3.相似系数应用的简单理解

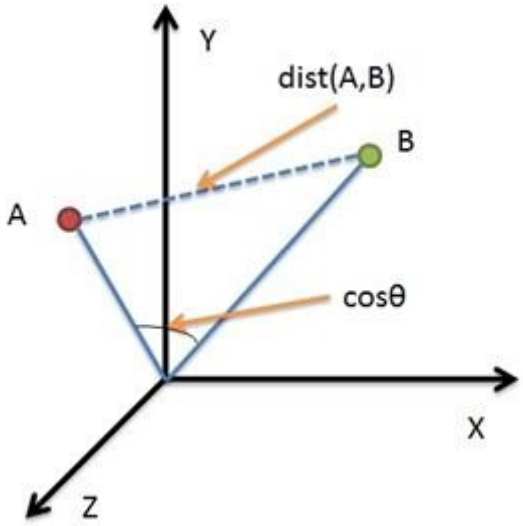
相似系数可以用于筛选和查询，普及一个知识，我们所有的信息都是通过二进制码储存的。相似系数有很多：Cosine 相似度，欧几里得相似度，*Jaccard*相似数{相似度计算的算法总结}

举例：

- **Cosine 相似度和欧几里得距离的异同**

相同：基于直角坐标系实现的

不同：



	Item 101	Item 102	Item 103	Correlation with User 1
User 1	5.0	3.0	2.5	1.000
User 2	2.0	2.5	5.0	0.796
User 3	2.5	-	-	1.000
User 4	5.0	-	3.0	0.997
User 5	4.0	3.0	2.0	0.995

余弦相似度的特点：

1. 对用户的绝对的数值不敏感
2. 计算时不考虑用户之间的共同评分项数量，即使仅仅有极少相同评分项，也有可能获得很大的相似度结果，例如上表中的uer3与user1.
3. 只要各个评分项之间越趋向于对应成比例，而不论数值差异如何，则相似度越趋近于1.000.

根据欧氏距离和余弦相似度各自的计算方式和衡量特征，分别适用于不同的数据分析模型：

欧氏距离能够体现个体数值特征的绝对差异，所以更多的用于需要从维度的数值大小中体现差异的分析，如使用用户行为指标分析用户价值的相似度或差异；

余弦相似度更多的是从方向上区分差异，而对绝对的数值不敏感，更多的用于使用用户对内容评分来区分用兴趣的相似度和差异，同时修正了用户间可能存在的度量标准不统一的问题

余弦相似度更适合于这样一类数据的挖掘工作：

1. 计算结果对用户数据绝对值不敏感，例如在描述用户的兴趣、喜好、或用于情感分析时。
2. 用户数据中的评分值其实是用户主观的评分结果，换言之，每个用户的评价标准是不一致的，有一些对于“好的”界定标准更为苛刻，而另一些则对于“好”、“不好”的界定则更为宽容。这种情况下，用余弦相似度来计算用户之间的相似度或差异，可以**弱化度量标准不统一这一因素。**

> ### *Jaccard*相似数
>
> 处理二元变量，速度快效率高

4.各种相似度计算的python实现