常用的相似度计算方法原理及实现

在数据分析和数据挖掘以及搜索引擎中，我们经常需要知道个体间差异的大小，进而评价个体的相似性和类别。常见的比如数据分析中比如相关分析，数据挖掘中的分类聚类（K-Means等）算法，搜索引擎进行物品推荐时。

相似度就是比较两个事物的相似性。一般通过计算事物的特征之间的距离，如果距离小，那么相似度大；如果距离大，那么相似度小。比如两种水果，将从颜色，大小，维生素含量等特征进行比较相似性。

问题定义：有两个对象X,Y,都包含N维特征，X=(x1,x2,x3,……..,xn),Y=(y1,y2,y3,……..,yn),计算X和Y的相似性。常用的有五种方法，如下。

1.1 欧氏距离 Eucledian Distance

欧氏距离是最常用的距离计算公式，即连接两个点的线段的长度，衡量多维空间中两个点之间的绝对距离，计算简单，使用勾股定理在笛卡尔坐标计算距离，当数据很稠密并且连续时，这是一种很好的计算方式。缺点：欧式距离受到尺度影响，这意味着所计算的距离会根据特征维度的单位不同发生倾斜，比如对身高（cm）和体重（kg）两个单位不同的指标使用欧式距离会使结果不准确。因此在使用欧式距离度量之前，需要对数据进行归一化处理，欧式距离在不同尺度具有风险。

此外，随着数据维数的增加，欧氏距离的作用也就越小。这与维数灾难（curse of dimensionality）有关。 用例：当你拥有低维数据且向量的大小非常重要时，欧式距离的效果非常好。如果在低维数据上使用欧式距离，则如 k-NN 和 HDBSCAN 之类的方法可达到开箱即用的效果。

（euclidean metric）（也称欧氏距离）是一个通常采用的距离定义，指在m维空间中两个点之间的真实距离，或者向量的自然长度（即该点到原点的距离），在二维和三维空间中的欧氏距离就是两点之间的实际距离。欧几里得距离是数据上的直观体现，但在处理一些受主观影响很大的评分数据时，效果则不太明显。

2、曼哈顿距离（Manhattan Distance）

Manhattan distance = |x1 – x2| + |y1 – y2|，p1at (x1, y1) and p2 at (x2, y2).

代码：

3、明可夫斯基距离（Minkowski distance）

明氏距离是欧氏距离的推广，是对多个距离度量公式的概括性的表述，看看下图

公式：

从公式我们可以看出，

当p==1,“明可夫斯基距离”变成“曼哈顿距离”

当p==2,“明可夫斯基距离”变成“欧几里得距离”

当p==∞,“明可夫斯基距离”变成“切比雪夫距离”

代码：

4、（余弦相似度）Cosine Similarity

余弦相似度用向量空间中两个向量夹角的余弦值作为衡量两个个体间差异的大小。相比距离度量，余弦相似度更加注重两个向量在方向上的差异，而非距离或长度上。

代码：

5、Jaccard Similarity

Jaccard系数主要用于计算符号度量或布尔值度量的个体间的相似度，因为个体的特征属性都是由符号度量或者布尔值标识，因此无法衡量差异具 体值的大小，只能获得“是否相同”这个结果，所以Jaccard系数只关心个体间共同具有的特征是否一致这个问题。

对于上面两个对象A和B,我们用Jaccard计算它的相似性，公式如下

首先计算出A和B的交（A ∩ B），以及A和B的并 （A ∪ B）:

然后利用公式进行计算:

代码：

六、皮尔森相关系数(Pearson Correlation Coefficient)

又称相关相似性，通过Peason相关系数来度量两个用户的相似性。计算时，首先找到两个用户共同评分过的项目集，然后计算这两个向量的相关系数。

公式：

实现汇总：

参考资料

1、Implementing the five most popular Similarity Measures in Python

2、相似度方法总结