R语言:计算各种距离

原创 2016年11月13日 23:40:20

标签：r语言 (http://so.csdn.net/so/search/s.do?q=r语言&t=blog) / 距离 (http://so.csdn.net/so/search/s.do?q=距离&t=blog) / distance (http://so.csdn.net/so/search/s.do?q=distance&t=blog)

9563

> 本文系改编的，利用R语言来计算各种距离

- MATLAB 版本的
  http://blog.csdn.net/sinat_26917383/article/details/52101425
  (http://blog.csdn.net/sinat_26917383/article/details/52101425)

- PYTHON版本的
  http://book.2cto.com/201511/58274.html (http://book.2cto.com/201511/58274.html)

==========

在做分类时常常需要估算不同样本之间的相似性(Similarity Measurement),这时通常采用的方法就是计算样本间”距离”(Distance)。采用什么样的方法计算距离是很讲究，甚至关系到分类的正确与否。

本文的目的就是对常用的相似性度量做一个总结。

本文目录：

==========

1、欧式距离(Euclidean Distance)

欧式距离是最易于理解的一种距离计算方法，源自欧式空间中两点间的距离公式。

两个n维向量a与b间的欧式距离：

$$d = \sqrt[2]{(a-b)^T(a-b)}$$

用R语言计算距离主要是dist函数。若X是一个M×N的矩阵，则dist(X)将X矩阵M行的每一行作为一个N维向量，然后计算这M个向量两两间的距离。

```
> aa=matrix(rnorm(15,0,1),c(3,5))
> aa
           [,1]       [,2]       [,3]       [,4]       [,5]
[1,] -1.6486207 -0.2149357 -0.02125219  0.0211059 -2.4320995
[2,] -0.2600026 -1.0145245 -0.24380395 -1.4597659 -0.8684985
[3,]  0.3500116  1.0524999  0.67703932  4.0102187  0.5309405
> dist(aa,p=2)
         1        2
2 2.693503
3 5.548077 6.113250
```

第一个行与第二行的距离为2.693503；第二行与第三行的距离为6.113250；第一行与第三行的距离为5.548077

2、曼哈顿距离(Manhattan Distance)

从名字就可以猜出这种距离的计算方法了。想象你在曼哈顿要从一个十字路口开车到另外一个十字路口，驾驶距离是两点间的直线距离吗？显然不是，除非你能穿越大楼。实际驾驶距离就是这个“曼哈顿距离”。而这也是曼哈顿距离名称的来源， 曼哈顿距离也称为城市街区距离(City Block distance)。

两个n维向量a(a1;a2;...;an)与 b(b1;b2;...;bn)间的曼哈顿距离

$$d = \sum_{k=1}^{n} |a_k - b_k|$$

R语言计算曼哈顿距离

```
> aa=matrix(rnorm(15,0,1),c(3,5))
>
>
> dist(aa,"manhattan")
         1        2
2 7.888601
3 5.944147 5.029586
```

第一行与第二行的距离为7.888601，第二行与第三行的距离为5.029586。第一行与第二行的距离为5.944147。

3、切比雪夫距离（Chebyshev Distance）

国际象棋玩过么？国王走一步能够移动到相邻的8个方格中的任意一个。那么国王从格子(x1,y1)走到格子(x2,y2)最少需要多少步？自己走走试试。你会发现最少步数总是max(|x2-x1|,|y2-y1|)步 。有一种类似的一种距离度量方法叫切比雪夫距离。
两个n维向量a(a1;a2;...;an)与 b(b1;b2;...;bn)间的曼哈顿距离

$$d = max_k |a_k - b_k|$$

或

$$d = lim_{p\to\infty} (\sum_{k=1}^{n} |a_k - b_k|^p)^{1/p}$$

R语言代码：

```
> aa=matrix(rnorm(15,0,1),c(3,5))
> aa
           [,1]       [,2]       [,3]       [,4]       [,5]
[1,] 0.3186289  0.8924295 -1.4619398  2.0500080 -0.9411515
[2,] 0.1582820  0.9655279 -0.9702412 -0.8561219  0.2322826
[3,] 0.7457046 -2.2780280 -0.7043906 -0.1458262  1.0166551
> dist(aa,"maximum")
         1        2
2 2.906130
3 3.170458 3.243556
```

4、闵可夫斯基距离(Minkowski Distance)

闵可夫斯基距离不是一种距离，而是一组距离的定义

（1）闵可夫斯基距离的定义

$a(a1;a2;\ldots;an)$ $b(b1;b2;\ldots;bn)$

两个n维变量 与 间的闵可夫斯基距离的定义为：

$$d=\sqrt[p]{\sum_{k=1}^{n}|a_k-b_k|^p}$$

其中p为一个变参数

$p=1$

- 当 时，就是曼哈顿距离;

$p=2$

- 当 时，就是欧式距离;

$p\to\infty$

- 当 时，就是切比雪夫距离；

（2）闵可夫斯基距离的缺点

闵可夫斯基距离，包含曼哈顿距离、欧式距离和切比雪夫距离都存在明显的缺点。

举个例子：二维样本(身高,体重)，其中身高范围是150~190，体重范围是50~60，有三个样本：a(180,50)，b(190,50)，c(180,60)。那么a与b之间的闵氏距离（无论是曼哈顿距离、欧氏距离或切比雪夫距离）等于a与c之间的闵氏距离，但是身高的10cm真的等价于体重的10kg么？因此用闵氏距离来衡量这些样本间的相似度很有问题。

简单说来，闵氏距离的缺点主要有两个：(1)将各个分量的量纲(scale)，也就是“单位”当作相同的看待了。(2)没有考虑各个分量的分布（期望，方差等)可能是不同的。

dist函数默认p=2

R语言代码：

```
> aa=matrix(rnorm(15,0,1),c(3,5))
> aa
            [,1]       [,2]       [,3]       [,4]        [,5]
[1,] -1.0308810 -0.8312748  2.155180 -0.03742647 -0.009221875
[2,]  0.4809107  0.2089795  1.036577 -0.18443966 -0.739891640
[3,]  0.2201239  0.9085380 -2.424723 -1.41154591  0.310477668
> dist(aa,"minkowski")
         1        2
2 2.274732
3 5.249560 3.891922
```

5、标准化欧氏距离 (Standardized Euclidean distance)

(1)标准欧氏距离的定义

标准化欧氏距离是针对简单欧氏距离的缺点而作的一种改进方案。标准欧氏距离的思路：既然数据各

维分量的分布不一样，好吧！那我先将各个分量都“标准化”到均值、方差相等吧。均值和方差标准化到多少呢？这里先复习点统计学知识吧，假设样本集X的均值(mean)为m，标准差(standard deviation)为s，那么X的“标准化变量”表示为：

而且标准化变量的数学期望为0，方差为1。因此样本集的标准化过程(standardization)用公式描述就是：

$$x^* = \frac{x - \mu}{\delta}$$

标准化后的值 = (标准化前的值 - 分量的均值) /分量的标准差

经过简单的推导就可以得到两个n维向量a(a1,a2,...,an)与 b(b1,b2,...,bn)间的标准化欧氏距离的公式：

如果将方差的倒数看成是一个权重，这个公式可以看成是一种加权欧氏距离(Weighted Euclidean distance)。

$$d = \sqrt{\sum_{k=1}^{n}(\frac{a_k - b_k}{\delta_k})^2}$$

R语言代码：

```
> aa=matrix(rnorm(15,0,1),c(3,5))
> aa
           [,1]       [,2]       [,3]       [,4]       [,5]
[1,] 0.7233675 -2.2366853  0.2925270  0.145778 2.21037802
[2,] 0.6626326  1.1180383 -0.9277047 -1.632137 0.05656014
[3,] 1.7862817  0.2219847 -1.3089391  1.317257 1.44481446
> aa1 = scale(t(aa), center=T,scale=T)
> aa1
            [,1]       [,2]       [,3]
[1,]  0.30986940  0.7139633  0.8662527
[2,] -1.53828705  1.1167893 -0.3723890
[3,]  0.04086725 -0.6927588 -1.5846052
[4,] -0.05075789 -1.3158598  0.4948694
[5,]  1.23830828  0.1778660  0.5958721
attr(,"scaled:center")
[1]  0.2270730 -0.1445221  0.6922797
attr(,"scaled:scale")
[1] 1.601625 1.130527 1.262913
> aaa <- matrix(rep(0,9),3,3)
> aaa
     [,1] [,2] [,3]
[1,]    0    0    0
[2,]    0    0    0
[3,]    0    0    0
> bb <-c(1,1,1,1,1)#方差
> bb
[1] 1 1 1 1 1
> for (i in 1:3)
+     for (j in 1:3)
+         if (i<j)
+             aaa[i,j] <- sqrt(sum(((aa1[,j] - aa1[,i])/bb)^2))
> aaa
     [,1]     [,2]     [,3]
[1,]    0 3.236657 2.240865
[2,]    0 0.000000 2.547490
[3,]    0 0.000000 0.000000
```

6、马氏距离(Mahalanobis Distance)

(1)马氏距离定义

有M个样本向量X1~Xm，协方差矩阵记为S，均值记为向量μ，则其中样本向量Xi到u的马氏距离表示为：

$$d(X_i) = \sqrt{(X_i - u)^T S^{-1} (X_i - u)}$$

而其中向量Xi与Xj之间的马氏距离定义为：

$$d(X_i, X_j) = \sqrt{(X_i - X_j)^T S^{-1} (X_i - X_j)}$$

若协方差矩阵是单位矩阵（各个样本向量之间独立同分布）,则公式就成了：

$$d(X_i, X_j) = \sqrt{(X_i - X_j)^T (X_i - X_j)}$$

也就是欧氏距离了。
若协方差矩阵是对角矩阵，公式变成了标准化欧氏距离。

(2)马氏距离的优缺点：量纲无关，排除变量之间的相关性的干扰。

R语言代码：

```
mashi <-function(a,b)
{
#a,b均为向量
return (((a-b)%*% t(t(a-b))) / cov(a,b))
}
```

例子：

```
> a=rnorm(5,0,1)
> b=rnorm(5,1,1)
> a
[1] -1.2162212  0.3688722  0.3144903  0.5182250  0.4402706
> b
[1] 0.07437722 1.29657555 1.97632344 0.51883332 0.26438674
> mashi(a,b)
         [,1]
[1,] 20.39844
```

7、夹角余弦(Cosine)
有没有搞错，又不是学几何，怎么扯到夹角余弦了？各位看官稍安勿躁。几何中夹角余弦可用来衡量两个向量方向的差异，机器学习中借用这一概念来衡量样本向量之间的差异。
(1)在二维空间中向量A(x1,y1)与向量B(x2,y2)的夹角余弦公式：

$$cos(\theta) = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + x_2^2}\sqrt{x_2^2 + y_2^2}}$$

(2) 两个n维样本点a(a1;a2;...;an)与 b(b1;b2;...;bn)的夹角余弦

$$cos(\theta) = \frac{a^T b}{|a||b|}$$

夹角余弦取值范围为[-1,1]。夹角余弦越大表示两个向量的夹角越小，夹角余弦越小表示两向量的夹角越大。当两个向量的方向重合时夹角余弦取最大值1，当两个向量的方向完全相反夹角余弦取最小值-1。
夹角余弦的具体应用可以参阅参考文献[1]。

R语言代码：

```
> aa=matrix(rnorm(15,0,1),c(3,5))
> aa
          [,1]       [,2]       [,3]       [,4]       [,5]
[1,]  1.390935  0.2061215 -0.4412572 -0.1490162 -0.6332618
[2,] -1.404099  1.7485971  1.0966853  0.7876016  1.0543667
[3,]  1.571527 -0.5391710  0.1622600  0.6927980 -1.1825320
> bb <- matrix(rep(0,9),3,3)
> bb
     [,1] [,2] [,3]
[1,]    0    0    0
[2,]    0    0    0
[3,]    0    0    0
> for (i in 1:3)
+   for (j in 1:3)
+     if (i < j)
+       bb[i,j] = sum(t(aa[i,])*aa[j,])/sqrt((sum(aa[i,]^2))*sum(aa[j,]^2))
> bb
     [,1]       [,2]      [,3]
[1,]    0 -0.6294542 0.7612659
[2,]    0  0.0000000 -0.6025365
[3,]    0  0.0000000 0.0000000
```

8、汉明距离(Hamming distance)

(1)汉明距离的定义

两个等长字符串s1与s2之间的汉明距离定义为将其中一个变为另外一个所需要作的最小替换次数。例如字符串“1111”与“1001”之间的汉明距离为2。

应用：信息编码（为了增强容错性，应使得编码间的最小汉明距离尽可能大）。

```
> library(e1071)
> x <- c(1, 0, 0)
> y <- c(1, 0, 1)
> hamming.distance(x, y)
[1] 1
```

9、杰卡德相似系数(Jaccard similarity coefficient)

(1) 杰卡德相似系数

两个集合A和B的交集元素在A，B的并集中所占的比例，称为两个集合的杰卡德相似系数，用符号J(A,B)表示。

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

杰卡德相似系数是衡量两个集合的相似度一种指标。

(2) 杰卡德距离

与杰卡德相似系数相反的概念是杰卡德距离(Jaccard distance)。杰卡德距离可用如下公式表示：

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

杰卡德距离用两个集合中不同元素占所有元素的比例来衡量两个集合的区分度。

(3) 杰卡德相似系数与杰卡德距离的应用

可将杰卡德相似系数用在衡量样本的相似度上。

样本A与样本B是两个n维向量，而且所有维度的取值都是0或1。例如：A(0111)和B(1011)。我们将样本看成是一个集合，1表示集合包含该元素，0表示集合不包含该元素。

p ：样本A与B都是1的维度的个数

q ：样本A是1，样本B是0的维度的个数

r ：样本A是0，样本B是1的维度的个数
s ：样本A与B都是0的维度的个数
那么样本A与B的杰卡德相似系数可以表示为：
这里p+q+r可理解为A与B的并集的元素个数，而p是A与B的交集的元素个数。
而样本A与B的杰卡德距离表示为：

$$J = \frac{p}{p+q+r}$$

R语言代码：

```
library(proxy)
> x <- matrix(sample(c(FALSE, TRUE), 8, rep = TRUE), ncol = 2)
> x
      [,1]  [,2]
[1,]  TRUE  TRUE
[2,] FALSE  TRUE
[3,] FALSE FALSE
[4,] FALSE FALSE
> dist(x, method = "Jaccard")
    1   2   3
2 0.5
3 1.0 1.0
4 1.0 1.0 0.0
```

10、相关系数 (Correlation coefficient)与相关距离(Correlation distance)

(1) 相关系数的定义

$$\rho_{XY} = \frac{cov(X,Y)}{\sqrt{d(X)}\sqrt{d(X)}} = \frac{E((X-EX)(Y-EY))}{\sqrt{d(x)}\sqrt{d(X)}}$$

相关系数是衡量随机变量X与Y相关程度的一种方法，相关系数的取值范围是[-1,1]。
相关系数的绝对值越大，则表明X与Y相关度越高。
当X与Y线性相关时，相关系数取值为1（正线性相关）或-1（负线性相关）。

(2)相关距离的定义:

$$D_{XY} = 1 - \rho_{XY}$$

R语言代码：

```
> aa=matrix(rnorm(15,0,1),c(3,5))
> aa
           [,1]       [,2]        [,3]        [,4]       [,5]
[1,] -0.5186859  0.8688277 -0.60667129 -0.93180447 -1.4665178
[2,]  0.1623851  0.4467074 -0.80715445 -0.34559516  0.1938283
[3,] -0.8910159 -0.5494911  0.00393534 -0.04257953  0.3308673
> 1-cor(t(aa))
          [,1]      [,2]     [,3]
[1,] 0.0000000 0.6291852 1.637603
[2,] 0.6291852 0.0000000 1.404476
[3,] 1.6376026 1.4044762 0.000000
```

11、信息熵(Information Entropy)
信息熵并不属于一种相似性度量。那为什么放在这篇文章中啊？这个。。。我也不知道。 (ﾍ▽ﾍ)
信息熵是衡量分布的混乱程度或分散程度的一种度量。分布越分散(或者说分布越平均)，信息熵就越大。分

布越有序（或者说分布越集中），信息熵就越小。
计算给定的样本集X的信息熵的公式：

$$entropy = -\sum_{i=1}^{C} p_i log_2(p_i)$$

参数的含义：
C：样本集X的分类数
pi：X中第i类元素出现的概率
信息熵越大表明样本集S分类越分散，信息熵越小则表明样本集X分类越集中。。当S中C个分类出现的概率一样大时（都是1/C），信息熵取最大值log2(C)。当X只有一个分类时，信息熵取最小值0

```
test.entropy <- function(d){

print(d)
res <- 0
for(i in 1:length(d))
{
  if(d[i]!=0)
        res <- res + d[i]*log(d[i])
    }
return (-res)
}
```

12、kl散度

刻画两个分布的差异。

$$D(spec1||spec2) = \sum(spec1 * log(\frac{spec1}{spec2}))$$

```
library(seewave)
data(tico)
str(tico)
tico1 <- spec(tico, at=0.65, plot=FALSE)
tico2 <- spec(tico, at=1.1, plot=FALSE)
kl.dist(tico1, tico2)    # log2 (binary logarithm)
kl.dist(tico1, tico2, base=exp(1))  # ln (natural logarithm)
```

13、兰式距离

$$d_{ij} = \sum_{k=1}^{n} \frac{|x_{ik} - x_{jk}|}{x_{ik} + x_{jk}}$$

与马氏距离一样，兰氏距离对数据的量纲不敏感。不过兰氏距离假定变量之间相互独立，没有考虑变量之间的相关性。

R语言代码：

```
> aa=matrix(rnorm(15,0,1),c(3,5))
> aa
            [,1]          [,2]       [,3]       [,4]       [,5]
[1,]  0.02289905 -0.007154829 -1.1331360  0.7498863  1.1254641
[2,] -1.06508101 -0.316642339  0.7597450  0.3327373 -1.4923720
[3,] -0.67681654  0.188728888 -0.6868684 -0.4417319  0.4783747
> dist(aa, method = "canberra")
          1         2
2 14.589357
3  6.664461 33.073080
```