MULTIVARIATE DATA ANALYSIS FOR ABUNDANCE AND PRESENCE ABSENCE DATA

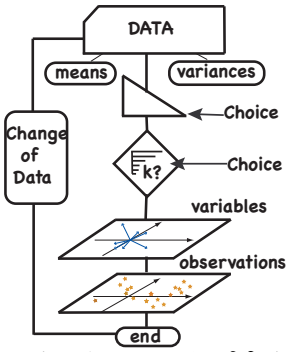
Susan Holmes ©
http://www-stat.stanford.edu/~susan/

Bio-X and Statistics, Stanford University

SISMID 14, June 2011

One table methods: PCA, MDS, PCoA, CA, ...

.



All based on the principle of finding the largest axis of inertia/variability.

New Variables/coordinates from old or distances

Best Projection Directions??

Metric Multidimensional Scaling

Given a distance matrix (or its square) how do we find the points in Euclidean space whose distances are given by this matrix?
Can we always find such a map?
Schoenberg (1935) but also Borschadt 1866.
Think of towns, whose road distances are known for whom we want to reconstruct a map.

Decomposition of Distances

If we started with original data in $\mathbb{R}^p$ that are not centered: Y, apply the centering matrix

$$X = HY, \qquad \text{with } H = (I - \frac{1}{n}\mathbf{1}\mathbf{1}'), \text{and } \mathbf{1}' = (1, 1, 1 \ldots, 1)$$

Call $B = XX'$, if $D^{(2)}$ is the matrix of squared distances between rows of X in the euclidean coordinates, we can show that

$$-\frac{1}{2}HD^{(2)}H = B$$

We can go backwards from a matrix D to X by taking the eigendecomposition of B in much the same way that PCA provides the best rank r approximation for data by taking the singular value decomposition of X, or the eigendecomposition of XX'.

$$X^{(r)} = US^{(r)}V' \text{ with } S^{(r)} = \begin{pmatrix} s_1 & 0 & 0 & 0 & \ldots \\ 0 & s_2 & 0 & 0 & \ldots \\ 0 & 0 & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & s_r & \ldots \end{pmatrix}$$

Multidimensional Scaling also called PCoA

Simple classical multidimensional scaling.

- Square D elementwise $D^{(2)} = D_2$.
- Compute $\frac{-1}{2}HD_2H = B$.
- Diagonalize B to find the principal coordinates.

Important: What D to use.

Distances, Similarities, Dissimilarities

Distances:

- Euclidean
- Chisquare

$$\text{Chisquare}(exp, obs) = \sum_j \frac{(exp_j - obs_j)^2}{exp_j}$$

- Hamming/L1
- DNA distances (dist.dna in ape)

Similarity Indices:

- Confusion (cognitive psychology).
- Matching coefficient

$$\frac{\text{nb of matching attrs}}{\text{nb of attrs}} = \frac{f_{11} + f_{00}}{f_{11} + f_{00} + f_{10} + f_{01}}$$

- Jaccard Similarity Index.

$$\frac{J}{A + B - J} = \frac{a}{a + b + c}$$

See versions in vegan and ade4.

Example of Distances

```
SMC = function(p,q) {
   # Compute M01,M10,M11,M00
   M01 = sum((p == 0) & (q == 1))
   M10 = sum((p == 1) & (q == 0))
   M00 = sum((p == 0) & (q == 0))
   M11 = sum((p == 1) & (q == 1))
   return((M11+M00)/(M01+M10+M00+M11))
}
# function for computing Jaccard coefficient
JC = function(p,q) {
   # Compute M01,M10,M11,M00
   M01 = sum((p == 0) & (q == 1))
   M10 = sum((p == 1) & (q == 0))
   M11 = sum((p == 1) & (q == 1))
   return(M11/(M01+M10+M11))
}
```

Example of Distances

```
cos.sim = function(p,q) {
   return(sum(p*q) / sqrt(sum(p^2)*sum(q^2)))
}
d1 = c(3,2,0,5,0,0,0,2,0,0)
d2 = c(1,0,0,0,0,0,0,1,0,2)
print(cos.sim(d1,d2))
[1] 0.3149704
 p=c(rep(0,6),rep(1,4))
 q=c(rep(0,6),1,0,0,1)
 q
 [1] 0 0 0 0 0 0 1 0 0 1
 print(JC(p,q))
[1] 0.5
 print(SMC(p,q))
[1] 0.8
```

Example of Distances: functions in R

```
help(dist)
dist(x, method = "euclidean", diag = FALSE, upper = FALSE,
as.dist(m, diag = FALSE, upper = FALSE)
method  the distance measure to be used. This must be one of
"euclidean","maximum","manhattan","canberra","binary"or"
Any unambiguous substring can be given.

dist returns an object of class "dist".
> disto=dist(olympic$tab)
> str(disto)
Class 'dist'  atomic [1:528] 6.59 8.67 17.02 15.53 12.39 ..
  ..- attr(*, "Size")= int 33
  ..- attr(*, "Labels")= chr [1:33] "1" "2" "3" "4" ...
  ..- attr(*, "Diag")= logi FALSE
  ..- attr(*, "Upper")= logi FALSE
  ..- attr(*, "method")= chr "euclidean"
  ..- attr(*, "call")= language dist(x = olympic$tab)
```

Example of Distances: functions in R

```
> as.matrix(disto)[1:5,1:5]
          1        2         3        4         5
1  0.000000  6.59430  8.670087 17.01519 15.534899
2  6.594301  0.00000 10.190211 12.39208 17.352000
3  8.670087 10.19021  0.000000 21.97831  9.824963
4 17.015190 12.39208 21.978314  0.00000 29.566708
5 15.534899 17.35200  9.824963 29.56671  0.000000
```

Chisquare: Aims and Relevant Data

What is a contingency table?
An example (thanks to the xkcd blag):

	black	blue	green	grey	orange	purple	white
quiet	27700	21500	21400	8750	12200	8210	25100
angry	29700	15300	17400	7520	10400	7100	17300
clever	16500	12700	13200	4950	6930	4160	14200
depressed	14800	9570	9830	1470	3300	1020	12700
happy	193000	83100	87300	19200	42200	26100	91500
lively	18400	12500	13500	6590	6210	4880	14800
perplexed	1100	713	801	189	233	152	1090
virtuous	1790	802	1020	200	247	173	1650

Correspondence analysis (CA, also called homogeneity analysis and reciprocal averaging), can be used to analyse several types of multivariate data. All involve some categorical variables. Here are some examples of the type of data that can be decomposed using this method:

- Contingency Tables (cross between two categorical variables)
- Multiple Contingency Tables (cross between several categorical variables).
- Binary tables obtained by cutting continuous variables into classes and then recoding both these variables and any extra categorical variables into 0/1 tables, 1 indicating presence in that class. So for instance a continuous variable cut into three classes will provide three new binary variables of which only one can take the value 1 for any given observation.

To first approximation, correspondence analysis can be understood as an extension of principal components analysis (PCA) where the variance in PCA is replaced by an inertia proportional to the χ^2 distance of the table from independence. CA decomposes this measure of departure from independence along axes that are orthogonal according to the χ^2 inner product.

Plato sentences

	Rep	Laws	Crit	Phil	Pol	Soph	Tim
1	42	91	5	24	13	26	18
2	60	144	3	27	19	33	30
3	64	72	3	20	24	31	46
4	72	98	2	25	20	24	14
5	79	113	10	38	25	22	26
6	76	144	6	46	22	23	27
7	79	102	5	41	25	30	26
8	83	68	3	14	18	37	26
9	106	23	2	7	3	19	13
10	174	333	9	62	31	21	25
11	125	129	4	64	41	30	26
12	98	38	4	6	7	15	17
13	174	42	3	7	8	28	21
14	98	57	4	30	24	28	23
15	166	113	5	18	23	28	17
16	94	216	10	52	34	47	30
17	110	159	4	53	53	48	23
18	113	53	3	7	21	24	25
19	128	38	1	4	5	21	25

We will also consider `multiple contingency tables' where more than two categorical variables are compared.

```
, , Sex = Male
       Eye
Hair    Brown Blue Hazel Green
  Black    32   11    10     3
  Brown    53   50    25    15
  Red      10   10     7     7
  Blond     3   30     5     8
, , Sex = Female
       Eye
Hair    Brown Blue Hazel Green
  Black    36    9     5     2
  Brown    66   34    29    14
  Red      16    7     7     7
  Blond     4   64     5     8
```

```
> HairColor=HairEyeColor[,,2]
> chisq.test(HairColor)

Pearson's Chi-squared test

data:  HairColor
X-squared = 106.6637, df = 9, p-value < 2.2e-16

Warning message:
In chisq.test(HairColor) : Chi-squared approximation
```

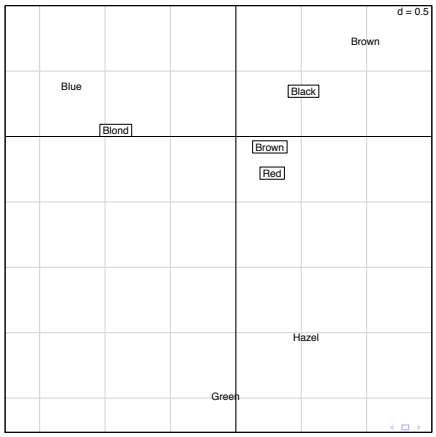
The data do not actually come in the form of a table usually

```
> HairColor
       Eye
Hair    Brown Blue Hazel Green
  Black    36    9     5     2
  Brown    66   34    29    14
  Red      16    7     7     7
  Blond     4   64     5     8
> sum(HairColor)
[1] 313
```

but as categorical variables, for instance:

```
  Id Hair_color Eye_color Sex
  1    Brown      Brown      F
  2    Blonde     Blue       M
  3    Black      Hazel      F
  ..........  ........     ..
  313 Red        Brown      M
```

```
> res.coa=dudi.coa(HairColor)
> s.label(res.coa$c1,boxes=F)
> s.label(res.coa$li,add.plot=TRUE)
```



Independence

If we are comparing two categorical variables, (hair color, eye color), (color, emotion), the simplest possible model is that of independence in which case the counts in the table would obey approximately the margin products identity for a $I \times J$ contingency table with a total sample size of $n = \sum_{i=1}^{I} \sum_{j=1}^{J} n_{ij} = n_{..}$.

$$n_{ij} \doteq \frac{n_{i.}}{n} \frac{n_{.j}}{n} n$$

can also be written: $\mathbf{N} \doteq \mathbf{c r}' \mathbf{n}$, where

$$c = \frac{1}{n} \mathbf{N} \mathbf{1}_m \quad \text{and} \quad \mathbf{r}' = \frac{\mathbf{1}}{\mathbf{n}} \mathbf{N}' \mathbf{1}_p$$

The departure from independence is measured by the χ^2 statistic

$$\mathcal{X}^2 = \sum_{i,j} [\frac{(n_{ij} - \frac{n_{i.}}{n} \frac{n_{.j}}{n} n)^2}{\frac{n_{i.} n_{.j}}{n^2} n}]$$

In the case of independence $\mathbf{F} \doteq \mathbf{r c}'$
All the rows would be multiples of each other or as this is sometimes called, homogeneous.
So, if all the rows were divided by the weight of that row, these so-called row profiles $\mathbf{F D_r}^{-1}$ would be equal ($\mathbf{F D_r}^{-1} = \mathbf{1_m c}$), where $\mathbf{D_r}^{-1}$ denotes the diagonal matrix with the vector $\mathbf{r}^{-1}$ on its diagonal.

Verification in **R**:

```
> F=HairColor/sum(HairColor)
> r=apply(F,1,sum)
> c=apply(F,2,sum)
> E=outer(r,c)
> E
           Brown       Blue      Hazel      Green
Black 0.06475518 0.06050894 0.02441589 0.01645418
Brown 0.17807674 0.16639958 0.06714369 0.04524901
Red   0.04607580 0.04305444 0.01737284 0.01170779
Blond 0.10086864 0.09425430 0.03803244 0.02563056
> sum((F-E)^2/E)
[1] 0.3407787
> sum((F-E)^2/E)*313
[1] 106.6637
> sum(E)
[1] 1
```

Formulation as a generalized singular value decomposition

Given an $m \times p$ contingency table of counts $\mathbf{N}$ of m levels for a row variable and p levels for a column variable. (This is equivalent to a binary matrix X with $n = \sum_{ij} n_{ij} = n_{..}$ observations on $m + p$ columns, a notion that is useful of the generalisation later.)
The first transformation makes the contingency matrix $\mathbf{N}$ into a frequency matrix $\mathbf{F} = \frac{1}{n}\mathbf{N}$. We will denote the row sums by $\mathbf{r} = \mathbf{F 1_p}$ and the column sums by the vector $\mathbf{c} = \mathbf{F}' \mathbf{1_m}$. These both sum to one

$$\mathbf{r}' \mathbf{1_m} = \mathbf{c}' \mathbf{1_p} = \mathbf{1}$$

In the case of independence

$$\mathbf{F} \doteq \mathbf{r c}'$$

The average row in the case of homogeneity and independence is obtained by averaging the rows with the relevant weights for each column. The average of the row-profiles is $\mathbf{c}$. The departure from independence and homogeneity is measured by some norm of $\mathbf{F D_r}^{-1} - \mathbf{1_m c}$ (or at the term by term level $\frac{f_{ij}}{f_{i.}} - f_{.j}$). With this notation we remark that

$$\begin{aligned} \chi^2 &= n \sum_{i,j} \frac{(f_{ij} - r_i c_j)^2}{r_i c_j} \\ &= n \sum_{i,j} r_i c_j (\frac{f_{ij}}{r_i c_j} - 1)^2 \end{aligned}$$

Matrix decomposition and χ^2 distances

To compute the distance between profiles, each column is reweighted by the inverse of its sum, this gives the χ^2 distance between row profiles.

$$\begin{aligned} \mathcal{X}^2 &= n \text{ trace } ((\mathbf{F} - \mathbf{r c}')' \mathbf{D_r}^{-1} (\mathbf{F} - \mathbf{r c}') \mathbf{D_c}^{-1}) \\ &= \text{ trace } (\mathbf{A}'\mathbf{A}) \text{ where } \mathbf{A} = \mathbf{D}_{\sqrt{r}}^{-1} (\mathbf{F} - \mathbf{r c}') \mathbf{D}_{\sqrt{c}}^{-1} \end{aligned}$$

The latter decomposition shows a justification for choosing the matrix $\mathbf{A}$ as a natural square root. $\mathbf{W} = \mathbf{A}'\mathbf{A}$ is in a sense the characteristic matrix-operator of the analysis, in the same way the covariance or correlation matrices are those of principal components analysis.

Correspondence analysis decomposes the matrix $\mathbf{W}$: its eigenvectors give the axes that account for the largest part of the departure from independence, just as principal components provides the axes accounting for the largest variability. Computationally this is achieved by a generalized singular value decomposition

$$\begin{aligned} \mathbf{D_r^{-1}FD_c^{-1}} - \mathbf{1_m'1_p} &= \mathbf{USV'}, \\ \text{with } \mathbf{V'D_cV} = \mathbf{I_p}, \mathbf{U'D_rU} &= \mathbf{I_m} \end{aligned}$$

equivalent to the eigendecompostion $\mathbf{W} = \mathbf{A'A} = \mathbf{V'S^2V}$ or the singular value decomposition

$$\mathbf{D_r}^{-\frac{1}{2}}\mathbf{FD_c}^{-\frac{1}{2}} - \sqrt{\mathbf{r}}\sqrt{\mathbf{c}}' = (\mathbf{D_r}^{\frac{1}{2}}\mathbf{U})\mathbf{S}(\mathbf{D_c}^{\frac{1}{2}}\mathbf{V})',$$

where $(\mathbf{D_c}^{\frac{1}{2}}\mathbf{V})'(\mathbf{D_c}^{\frac{1}{2}}\mathbf{V}) = \mathbf{I_p}$, and $(\mathbf{D_r}^{\frac{1}{2}}\mathbf{U})'(\mathbf{D_c}^{\frac{1}{2}}\mathbf{U}) = \mathbf{I_p}$.

Matrix Diagonalised and Diagram

```
> res.eigen=eigen(t(X)%*%diag(r)%*%X%*%diag(c))
> res.eigen
$values
[1] 1.000000000 0.302459246 0.032631660 0.005687796

$vectors
     [,1]        [,2]       [,3]        [,4]
[1,]  0.5  0.58634594 -0.2752916  0.08027417
[2,]  0.5 -0.74350003 -0.1444609 -0.05137694
[3,]  0.5  0.31830243  0.5814691 -0.61374293
[4,]  0.5 -0.04571334  0.7518240  0.78373215

> res.coa$eig
[1] 0.302459246 0.032631660 0.005687796

> t(res.coa$co[,1])%*%diag(c)%*%res.coa$co[,1]
          [,1]
[1,] 0.3024592
```

Matrix Diagonalized and Diagram

```
> res.eigen=eigen(t(X)%*%diag(r)%*%X%*%diag(c))
> res.eigen
$values
[1] 1.000000000 0.302459246 0.032631660 0.005687796

$vectors
     [,1]        [,2]       [,3]        [,4]
[1,]  0.5  0.58634594 -0.2752916  0.08027417
[2,]  0.5 -0.74350003 -0.1444609 -0.05137694
[3,]  0.5  0.31830243  0.5814691 -0.61374293
[4,]  0.5 -0.04571334  0.7518240  0.78373215

> res.coa$co
            Comp1       Comp2
Brown  0.54472970  0.13159215
Blue  -0.69072969  0.06905379
Hazel  0.29571073 -0.27794807
Green -0.04246881 -0.35937943
```

Decomposition of the difference from independence

A cloud, or scatter of weighted points

:
These are points defined in an euclidean space, say $\mathbb{R}^p$ for instance, so that distances between them are easy to compute. However we associate to each multidimensional point a weight that changes the inertia of the scatterpoints. For instance if we have two points the one with a higher weight will `pull ' the centre of gravity towards it. The same will happen for the `minimum inertia ' line, it will be pulled towards highly weighted points.

Distributional Equivalence :
If we add two rows that have the same profiles, this will not change the axes chosen to represent the data, (the column profiles' geometry remains unchanged). Thinking of the points as weighted points in a cloud, two points that would be at the same spot can be merged because we can add their weights.

Barycentric Representation

Take the simplest case: row profiles of a 3-column contingency table. The profiles sum to one so are all representable in a triangle (called the 3-dimensional simplex). The vertices are the extreme profiles, say (1,0,0), (0,1,0) and (0,0,1). Although the row profiles are in a three dimensional space as they belong to this triangle they can be taken out and just looked at in these coordinates, called the barycentric co-ordinate system. Now an extra scale change will bring this representation to the correpondence analysis one : the dimensions will be weighted inversely by the relative weights of the columns , called column mean profiles, (which also add to 1).

The distances we want to represent between points are to be the χ^2 distances relevant here, so the sides of the equilateral triangle are stretched to have sides inversely proportional to the square roots of their mean values.
The side of the triangle the most stretched corresponds to the least frequent column.
This representation is the one chosen by default by the function `scatter`, there is a delicate issue of choosing the scales in the two dimensions so that simultaneous representations of rows and columns are valid .
In the relevant choice of scaling, proximities between row and column points are hard to interpret, however it is easier to interpret the directions of the different rows and columns.

Reading the Output

Although the maps provided by doing both correspondence and principal components analysis look quite simple there are traps that lead to misinterpretations that must be avoided. Associated to the co-ordinates in the new spaces are what we call loadings or contributions and which are indicators of how true the proximities in the image space are. To this end the object that the function `dudi.coa` produce a listed output containing the eigenvalues, coordinates for the rows, for the columns, that are used for building the graphical representations and weights, also we can use the function `contrib` that are important diagnostic tools.

```
> names(res.coa)
 [1] "tab"  "cw"   "lw"   "eig"  "rank" "nf"   "c1"
[8]  "li"   "co"   "l1"   "call" "N"
```

Contributions

- When trying to understand the most meaningful rows or columns for a given axis we look at the absolute contributions of rows or columns to given axis, this gives the amount of an axis's inertia explained by a single row or column.
- The relative contribution of an axis/ of two axes to the inertia of a row. This is the same as the cosine of the point with the axis that says how well a point is being projected onto the axis.

Contribution[1] to the inertia from row i:
Distance from the ith row to the center of the row-points:

$$d^2_{\chi^2}(\text{profile}_i, \text{center}) = \sum_j \frac{1}{f_{\cdot j}}\left(\frac{f_{ij}}{f_{i\cdot}} - f_{\cdot j}\right)^2$$

$$= \sum f_{\cdot j}\left(\frac{f_{ij}}{f_{i\cdot} f_{\cdot j}} - 1\right)^2 = \sum_k (\sqrt{\lambda_k} u_i^k)^2$$

This row will thus participate to the inertia by this amount weighted by the row's mass r_i. This can be decomposed into each of the axis separately thus giving an idea of the contribution of each row to the inertia of each axis, this is called the absolute contribution of row i to axis k, $r_i \sum_k (\sqrt{\lambda_k} u_i^k)^2$. The sum of all row's contributions to a given axis add to one. This translates the fact that $U'D_rU = I_m$, here are the absolute contributions for the `eyes` data:

```
> inertia.dudi(res.coa,row.inertia=TRUE,col.inertia=
$TOT
      inertia       cum     ratio
1 0.302459246 0.3024592 0.8875533
2 0.032631660 0.3350909 0.9833094
3 0.005687796 0.3407787 1.0000000

$row.abs
      Axis1 Axis2
Black  1469  6096
Brown  1021   889
Red     302  2843
Blond  7209   172

$row.rel
      Axis1 Axis2 con.tra
Black  6860  3072    1900
Brown  8630  -811    1050
Red    4219 -4290     635
Blond -9974    26    6415

$row.cum
      Axis1 Axis2 remain
Black  6860  9932     68
Brown  8630  9441    559
Red    4219  8508   1492
Blond  9974  9999      1

$col.abs
      Comp1 Comp2
Brown  3824  2068
Blue   5745   532
Hazel   425  3479
Green     6  3920

$col.rel
      Comp1 Comp2 con.tra
Brown  9439   551    3596
```

```
Blue   -9898    99    5152
Hazel   4789 -4231     788
Green   -113 -8064     465

$col.cum
      Comp1 Comp2 remain
Brown  9439  9990     10
Blue   9898  9997      3
Hazel  4789  9020    980
Green   113  8177   1823
```

Thus we can see that the most important row category for explaining the first axis is Blonde hair.

Distances and Inertia

Variance is at the heart of ANOVA: decomposition of variability into parts that can be explained by a factor and a residual part.

$$\text{Test statistic: } \frac{\text{average ss explained by factor}}{\text{average of residual ss}}$$

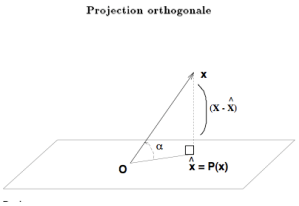
Features

1. Inertia : $\text{Trace}(VQ) = \text{Trace}(WD)$
(inertia in the sense of Huyghens inertia formula for instance). Huygens, C. (1657),

$$\sum_{i=1}^{n} p_i d^2(x_i, a)$$

Inertia with regards to a point a of a cloud of p_i-weighted points.
PCA with $Q = \mathcal{I}_p$, $D = \frac{1}{n}\mathcal{I}_n$, and the variables are centered, the inertia is the sum of the variances of all the variables.
If the variables are standardized (Q is the diagonal matrix of inverse variances), then the inertia is the number of variables p.
For correspondence analysis the inertia is the Chi-squared statistic.

Quality of Representations



- The cosine again, $\cos(x, y) = \frac{<x,y>}{||x||||y||}$
- $$\cos^2\alpha = \frac{||\hat{x}||^2}{||x||^2}$$
  tells us how well x is represented by its projection.

Inertia and Contributions

- $$In(X) = ||X||^2 = \sum_i p_i ||x_{i\cdot}||^2 = \sum_j q_j ||x^{\cdot j}||^2 = \sum_{\ell=1}^{p} \lambda_\ell$$
- Contribution of an observation to the total inertia: $\frac{p_i ||x_{i\cdot}||^2}{||X||^2}$
- Contribution of a variable to the total inertia: $\frac{q_j ||x^{\cdot j}||^2}{||X||^2}$

Inertia and Contributions

- Contribution of the kth axis to variable j: $\frac{\lambda_k v_{kj}^2}{||x^j||_D^2}$
- Contribution of variable j to the kth axis $q_j v_{kj}^2$.
- Contribution of the kth axis to observation i: $\frac{\lambda_k u_{ik}^2}{||x_i||_Q^2}$
- Contribution of observation i to the kth axis $p_i u_{ik}^2$.

Quality of approximations

$$\frac{||X^{[k]}||^2}{||X||^2} = \frac{\sum_{\ell=1}^{k} \lambda_\ell}{\sum_{\ell=1}^{p} \lambda_\ell}$$

This is like a cosine, we can compare two operators with this.

Data / Rows/ Columns/ Dimensions

There is an important symmetry between the rows and columns of X in the diagram, and one can imagine situations where the role of observation or variable is not uniquely defined. For instance in microarray studies the genes can be considered either as variables or observations. This makes sense in many contemporary situations which evade the more classical notion of n observations seen as a random sample of a population. It is certainly not the case that the 9,000 species are a random sample of bacteria since these probes try to be an exhaustive set.

Two Dual Geometries