Distances

PH143 – Spring 2004
Lectures 25 – 26

Sandrine Dudoit

©Copyright 2004, all rights reserved

Outline

- Role of distances.
- Distances.
- Distances between clusters.
- Standardization.
- Experiment-specific distances between genes.
- Principal component analysis (PCA).
- Multidimensional scaling (MDS).

Role of distances

Task. Assign observational units to classes on the basis of variables describing/characterizing these observations.

Clustering. The classes are unknown a priori and need to be "discovered" from the data.
A.k.a. class discovery; unsupervised learning; unsupervised pattern recognition.

Class prediction. The classes are predefined and the task is to understand the basis for the classification from a set of labeled observations (learning set). This information is then used to predict the class of future observations.
A.k.a. classification; discriminant analysis; supervised learning; supervised pattern recognition.

Role of distances

Inherent in clustering and class prediction methods is a notion of distance or similarity between the observations to be clustered or classified.

- Many clustering procedures operate directly on a matrix of pairwise distances between the observations to be clustered, e.g., partitioning around medoid (PAM) and hierarchical clustering methods.
- In class prediction, new observations are typically assigned to classes on the basis of their distances from observations with known class labels (learning set).
  - k-nearest neighbor classifiers: based on an explicit choice of distance function.
  - Linear discriminant analysis: based on the Mahalanobis distance of observations from class means.

Role of distances

The choice of distance is important and can have a large impact on the results of supervised and unsupervised learning analyses.

In some cases, the Euclidean metric will be sensible, while in others, a distance based on correlations will be a better choice.

Subject matter knowledge is very helpful in selecting an appropriate distance for a given project.

Distances

Let $x = (x(j) : j = 1, \ldots, J)$ denote a J-dimensional vector of variables describing a particular observational unit in the population or sample of interest.

Depending on the context, the entries $x(j)$ can be referred to as covariates, explanatory variables, features, measurements.

The variables $x(j)$ can be

- quantitative (= numerical): continuous or discrete;
- qualitative (= categorical): ordinal or nominal.

E.g. x could refer to a vector of microarray expression measures and clinical covariates (e.g., age, sex, blood pressure, censored survival time).

Distances

A distance function, or in short a distance, is a function d that satisfies the following three properties.

1. Non-negativity: $d(x, y) \geq 0$.
2. Symmetry: $d(x, y) = d(y, x)$.
3. Identification mark: $d(x, x) = 0$.

A metric is a distance function that satisfies also the following two properties.

4. Definiteness: $d(x, y) = 0$ if and only if $x = y$.
5. Triangle inequality: $d(x, z) \leq d(x, y) + d(y, z)$.

In general, we will use the term *distance*, which includes *metrics*, and only mention metrics when the behavior of interest is specific to them.

Distances

A similarity function, s, is more loosely defined and satisfies the following three properties.

1. Non-negativity: $s(x, y) \geq 0$.
2. Symmetry: $s(x, y) = s(y, x)$.
3. The more *similar* the objects x and y, the greater $s(x, y)$.

A dissimilarity function, d, satisfies 1. and 2., above, and

3. The more *dissimilar* the objects x and y, the greater $d(x, y)$.

Distances

There is a great deal of choice (and hence literature) on selecting a distance function.

Some references that pay particular attention to distances in the context of classification and clustering include

- Section 4.7 of Duda, Hart, & Stork (2000);
- Chapter 2 of Gordon (1999);
- Chapter 1 of Kaufman & Rousseeuw (1990);
- Chapter 13 of Mardia, Kent, & Bibby (1979).

Distances

The following are common distance functions for covariate vectors belonging to the J-dimensional Euclidean set $\mathbb{R}^J$.

- Euclidean metric (possibly standardized);
- Mahalanobis metric;
- Manhattan metric;
- Minkowski metric (special cases are the Euclidean and Manhattan metrics);
- Canberra metric;
- One minus correlation/absolute correlation distance.

Distances

Denote the average and variance of the entries of $x \in \mathbb{R}^J$ by

$$\bar{x} \equiv \frac{1}{J}\sum_{j=1}^{J} x(j) \qquad \text{and} \qquad s_x^2 \equiv \frac{1}{J}\sum_{j=1}^{J}(x(j)-\bar{x})^2.$$

Given n covariate vectors, $x_1, \ldots, x_n \in \mathbb{R}^J$, let μ_n denote the J-dimensional sample mean vector, with jth entry

$$\mu_n(j) \equiv \frac{1}{n}\sum_{i=1}^{n} x_i(j).$$

Let Σ_n denote the $J \times J$ sample covariance matrix, with entries

$$\Sigma_n(j,j') \equiv \frac{1}{n}\sum_{i=1}^{n}(x_i(j)-\mu_n(j))(x_i(j')-\mu_n(j')).$$

In particular, let $\sigma_n^2(j) \equiv \Sigma_n(j,j)$ denote the sample variance of the jth variable and let $R_n(j) \equiv \max_{i,i'} |x_i(j) - x_{i'}(j)|$ denote its range.

Distances

Table 1: *Metrics and distances.* x and $y \in \mathbb{R}^J$.

Name	Formula
Euclidean metric	$d_E(x,y) = \{\sum_j w(j)(x(j)-y(j))^2\}^{1/2}$.
Unstandardized	$w(j) = 1$;
Standardized by variance (Karl Pearson distance)	$w(j) = 1/\sigma_n^2(j)$;
Standardized by range	$w(j) = 1/R_n^2(j)$.
Mahalanobis metric	$d_{Ml}(x,y) = \{(x-y)^\top S^{-1}(x-y)\}^{1/2}$ $= \{\sum_j \sum_{j'} S^{-1}(j,j')(x(j)-y(j))(x(j')-y(j'))\}^{1/2}$, where $S = (S(j,j'))$ is any $J \times J$ positive definite matrix, usually the sample covariance matrix Σ_n of the variables. When the matrix S is the identity, I_J, d_{Ml} reduces to the unstandardized Euclidean distance.
Manhattan metric	$d_{Mn}(x,y) = \sum_j w(j)\lvert x(j)-y(j)\rvert$.
Minkowski metric	$d_{Mk}(x,y) = \{\sum_j w(j)\lvert x(j)-y(j)\rvert^\lambda\}^{1/\lambda}$, $\lambda \geq 1$. $\lambda = 1$: Manhattan metric; $\lambda = 2$: Euclidean metric.
Canberra metric	$d_C(x,y) = \sum_j \frac{\lvert x(j)-y(j)\rvert}{\lvert x(j)+y(j)\rvert}$.
One minus Pearson correlation	$d_{cor}(x,y) = 1 - Cor(x,y) = 1 - \frac{\sum_j (x(j)-\bar{x})(y(j)-\bar{y})}{\sqrt{\sum_j (x(j)-\bar{x})^2}\sqrt{\sum_j (y(j)-\bar{y})^2}}$.

Distances

Suppose x and y are two J-dimensional feature vectors belonging to the Euclidean set $\mathbb{R}^J$ and such that $y(j) = ax(j) + b$ for all j, where $a, b \in \mathbb{R}$ and $a \neq 0$. Then, the Euclidean distance between x and y is given by

$$\begin{aligned} d_E(x,y) &= \left\{ \sum_{j=1}^{J} (x(j) - y(j))^2 \right\}^{1/2} \\ &= \left\{ \sum_{j=1}^{J} ((1-a)x(j) - b)^2 \right\}^{1/2} \\ &= \left\{ (1-a)^2 \sum_{j=1}^{J} x^2(j) - 2b(1-a)J\bar{x} + Jb^2 \right\}^{1/2}, \end{aligned}$$

and is a function of a and b, and the mean $\bar{x}$ and variance s_x^2 of x. When $a = 1$, $d_E(x,y) = \sqrt{J}|b|$.

Distances

Consider next the following two correlation-based distances

$$d_{cor}(x,y) = 1 - Cor(x,y) \in [0,2],$$

and

$$d_{|cor|}(x,y) = 1 - |Cor(x,y)| \in [0,1].$$

Note that $d_{cor}(x,y)$ achieves its minimum value when $Cor(x,y) = 1$ and maximum value when $Cor(x,y) = -1$. In contrast, $d_{|cor|}(x,y)$ achieves its minimum when $Cor(x,y) = \pm 1$ and its maximum when $Cor(x,y) = 0$.

Distances

When $y(j) = ax(j) + b$ for all j, the Pearson correlation between x and y is

$$Cor(x,y) = \text{sign}(a) = \begin{cases} 1, & \text{if } a > 0, \\ -1, & \text{if } a < 0. \end{cases}$$

The correlation-based distances are

$$d_{cor}(x,y) = 1 - Cor(x,y) = \begin{cases} 0, & \text{if } a > 0, \\ 2, & \text{if } a < 0, \end{cases}$$

and

$$d_{|cor|}(x,y) = 1 - |Cor(x,y)| = 0.$$

The appropriate distance depends on the subject matter and question of interest.

Distances

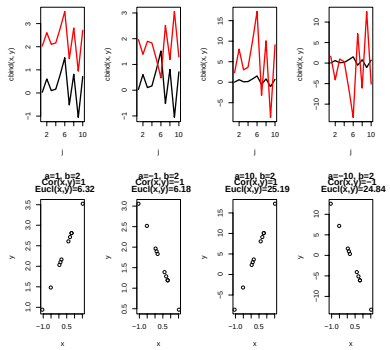


Figure 1: *Euclidean distance d_E vs. correlation-based distance d_{cor}.* Covariate vectors x and $y \in \mathbb{R}^{10}$, with $y(j) = ax(j) + b$ for all j. Top: Plots of $x(j)$ and $y(j)$ vs. j. Bottom: Plots of y vs. x.

Distances

```
pdf("euclVsCor.pdf")
a<-c(1,-1,10,-10)
b<-c(2,2,2,2)
x<-rnorm(10)
par(mfcol=c(2,4))
for(j in 1:length(a))
{
  y<-a[j]*x+b[j]
  matplot(cbind(x,y),type="l",lwd=2,lty=1,xlab="j")
  plot(x,y,main=paste("a=", a[j],", b=", b[j], "\n Cor(x,y)=", cor(x,y),"\n  Euc
}
dev.off()
```


Distances

For binary vectors x and y, i.e., $x, y \in \{0,1\}^J$, a possible measure of distance is the number of entries that are one (i.e., *on bits*) in only one of the vectors over the number of entries that are one in at least one vector

$$\begin{aligned} d_{bin}(x,y) &= \frac{\text{Number of discrepant bits}}{\text{Number of bits that are } on \text{ in at least one vector}} \\ &= \frac{\sum_{j=1}^{J} \mathrm{I}(x(j) != y(j))}{J - \sum_{j=1}^{J} \mathrm{I}(x(j) = y(j) = 0)} \\ &= \frac{\sum_{j=1}^{J} x(j)(1-y(j)) + (1-x(j))y(j)}{J - \sum_{j=1}^{J}(1-x(j))(1-y(j))}. \end{aligned}$$

E.g. Distances between SNP haplotypes. Could use weights that reflect major/minor allele frequencies.

Distances

Distances may need to be extended in various ways to deal with different types of problems. When some variables are numerical and others categorical, there are more choices and the implications of these different choices should be examined carefully.

E.g. The feature vectors consist of (categorical) patient level covariates and (numerical) gene expression measures. The Euclidean distance might be appropriate for the gene expression measures, but not for the patient covariates.

Weights may be incorporated in any of the distances above to deal with different types of variables.

Distances

One might consider hybrid versions of the distances, such as convex combinations of two or more distance functions

$$d(x,y) = \sum_{k=1}^{K} \delta_k d_k(x,y),$$

where $\delta_k \geq 0$, $\sum_k \delta_k = 1$, and each distance function $d_k(x,y)$ may only operate on certain entries of the vectors x and y.

For two distance functions one has

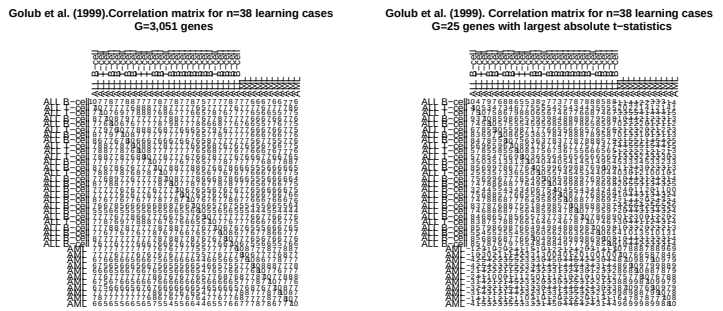
$$d(x,y) = \delta d_1(x,y) + (1-\delta) d_2(x,y), \qquad \delta \in [0,1].$$

Distances: R functions

R has a number of functions for computing and displaying distance and similarity matrices.

- Distance functions
  - `dist` (`mva`): Euclidean, Manhattan, Canberra, binary;
  - `daisy` (`cluster`): Euclidean, Manhattan.
- Correlation functions
  - `cor`, `cor.wt` (`base`).
- Plotting functions
  - `image` (`base`);
  - `plotcorr` (`ellipse`);
  - `levelplot` (`lattice`);
  - `plot.cor`, `plot.mat` (`sma`).

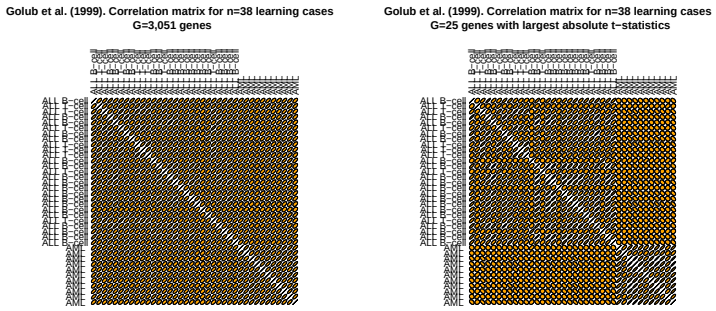
Distances: R functions, `plotcorr`



(a) (b)

Figure 2: *Golub et al. (1999) leukemia dataset.* Correlation matrix of expression measures for 38 learning set cases: (a) $G = 3,051$ genes and (b) $G = 25$ genes with largest absolute t-statistics (ALL vs. AML).

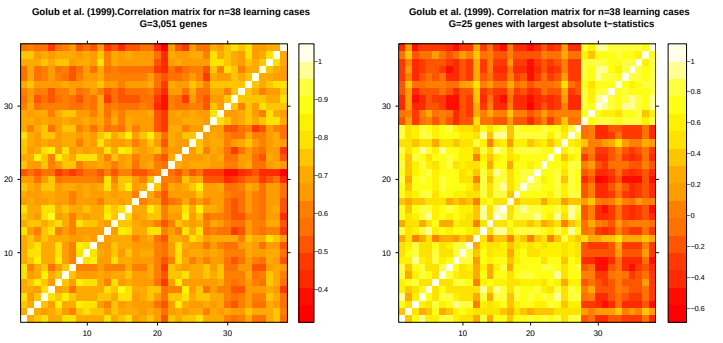
Distances: R functions, `plotcorr`



(a) (b)

Figure 3: *Golub et al. (1999) leukemia dataset.* Correlation matrix of expression measures for 38 learning set cases: (a) $G = 3,051$ genes and (b) $G = 25$ genes with largest absolute t-statistics (ALL vs. AML).

Distances: R functions, `levelplot`



(a) (b)

Figure 4: *Golub et al. (1999) leukemia dataset.* Correlation matrix of expression measures for 38 learning set cases: (a) $G = 3,051$ genes and (b) $G = 25$ genes with largest absolute t-statistics (ALL vs. AML).

Distances between clusters

Many clustering algorithms require a measure of distance between clusters. There are a number of different ways of defining a distance between groups, or between one observation and a group of observations.

Single linkage. The distance between two clusters is the *minimum* distance between two observations, one from each cluster.

Average linkage. The distance between two clusters is the *average* of all pairwise distances between the members of both clusters.

Complete linkage. The distance between two clusters is the *maximum* distance between two observations, one from each cluster.

Centroid distance. The distance between two clusters is the distance between their *centroids*. The definition of centroid may depend on the clustering algorithm being used, e.g., medoid, average, median.

Distances between clusters

The choice of distance measure between clusters can have a large effect on the shape of the resulting clusters.

For instance, single linkage tends to lead to long, thin clusters, while average linkage leads to round clusters.

Standardization

- Standardization is an important issue when considering distances between observations and/or variables.
- The distance function and its behavior are intimately linked to the location and scale of the measured variables.
- There are no objective methods for dealing with this problem. The solution is generally problem specific.

Standardization

For microarray data both genes and/or arrays can be standardized. Which of the two types of standardization should be carried out is dependent, among other considerations, upon whether samples or genes are being clustered or classified. Gene expression data on G genes (features) for n mRNA samples (observations)

$$X_{G\times n} = \begin{array}{c} \text{mRNA samples} \\ \left[\begin{array}{cccc} X(1,1) & X(1,2) & \dots & X(1,n) \\ X(2,1) & X(2,2) & \dots & X(2,n) \\ \vdots & \vdots & \ddots & \vdots \\ X(G,1) & X(G,2) & \dots & X(G,n) \end{array}\right] \end{array} \text{Genes}$$

$X(g,i)$ = expression measure for gene g in mRNA sample i.

Standardization

Standardizing genes

$$X(g,i) \leftarrow \frac{X(g,i) - loc(g,\cdot)}{scale(g,\cdot)},$$

where $loc(g,\cdot)$ is a measure of the location (i.e., center) of the distribution of the set of values $\{X(g,i) : i = 1,\ldots,n\}$, such as the mean or median, and $scale(g,\cdot)$ is a measure of scale, such as the standard deviation, IQR, or MAD.

Standardizing arrays

$$X(g,i) \leftarrow \frac{X(g,i) - loc(\cdot,i)}{scale(\cdot,i)}.$$

Standardization

- Gene standardization in some sense puts all genes on an equal footing and weighs them equally in the classification or clustering. Common standardization procedures are
- $X(g,i) \leftarrow \frac{X(g,i)-\mu_n(g,\cdot)}{\sigma_n(g,\cdot)}$,
  where $\mu_n(g,\cdot)$ and $\sigma_n(g,\cdot)$ denote, respectively, the average and standard deviation of gene g's expression measures across the n arrays.
- $X(g,i) \leftarrow \frac{X(g,i)-Med_n(g,\cdot)}{MAD_n(g,\cdot)}$,
  where $Med_n(g,\cdot)$ and $MAD_n(g,\cdot)$ denote, respectively, the median and median absolute deviation (MAD) of gene g's expression measures across the n arrays. These are robust estimates of location and scale.
- $X(g,i) \leftarrow \frac{X(g,i)-X(g,O_n(1))}{X(g,O_n(n))-X(g,O_n(1))} = \frac{X(g,i)-X(g,O_n(1))}{R_n(g,\cdot)}$,
  where $X(g,O_n(i))$ denote the ordered expression measures for gene g, $X(g,O_n(1)) \leq X(g,O_n(2)) \leq \ldots \leq X(g,O_n(n))$, and $R_n(g,\cdot)$ is the range.

Standardization

Standardization of arrays can be viewed as part of the normalization step.

It is consistent with the common practice of using the correlation between the gene expression profiles of two mRNA samples to measure their similarity.

In practice, we recommend more general adaptive and robust normalization methods, which correct for intensity, spatial, and other types of bias using robust local regression (see lecture notes on pre-processing).

Standardization

Table 2: *Impact of standardization of observations and variables on the distance function between observations.*

Distance between observations	Standardize variables (genes)	Standardize observations (arrays)
Euclidean, $w(j) = 1$	Changed	Changed
Euclidean, $w(j) = 1/\sigma_n^2(j)$	Unchanged	Changed
Mahalanobis	Changed, unless S diagonal	Changed
One minus Pearson correlation	Changed	Unchanged

Standardization

Note the relationship between the Euclidean distance $d_E(\cdot, \cdot)$, for standardized vectors x and y, and the correlation-based distance $d_{cor}(x, y) = 1 - Cor(x, y)$, defined as one minus the Pearson correlation $Cor(x, y)$,

$$d_E(x, y) = \sqrt{2Jd_{cor}(x, y)}.$$

[*By standardized vector, we mean:* $x(j) \leftarrow (x(j) - \bar{x})/s_x$.]

Experiment-specific distances between genes

The gene distance functions considered thus far do not take into account the structure or design of the microarray experiment, i.e., they treat the columns of the genes-by-arrays matrix of expression measures interchangeably.

However, microarray experiments can be highly-structured, e.g., as in timecourse and multifactorial experiments.

Below are general approaches to supervise the distances so that they reflect the design of the experiment under consideration.

Experiment-specific distances between genes

One can exploit covariate information (e.g., treatment, cell type, dose, time) to derive suitable transformations of the genes-by-arrays data matrix, e.g., using linear modeling.
genes-by-arrays matrix $\Longrightarrow$ genes-by-estimated-effects matrix.

Instead of computing distances directly on the genes-by-arrays data matrix, distances can be computed on the new genes-by-estimated-effects matrix. Genes can then be clustered based on these new distances.

Experiment-specific distances between genes

In timecourse experiments, it makes sense to consider distances that are not time exchangeable and use the time index in an essential way.

For a large enough number of timepoints, one may

- penalize for non-smoothness as in Sobolev metrics;
- use one of the standard wavelet decompositions to transform expression profiles into potentially interpretable quantities corresponding to local frequency components.

Distances can then be computed for the new gene expression profiles and genes clustered based on these distances.

Experiment-specific distances between genes

The transformed gene expression profiles can be matched to a library of reference profiles of interest for the particular experiment.

For instance, in factorial experiments across time, interesting reference profiles for main effects and interactions might include: cyclical, early, or late effects, or the effects over time for a known gene.

One may also compare genes based on biological metadata, e.g., co-citation in PubMed abstracts.

Some linear algebra

Consider an $M \times M$ matrix A, with entries $A(m, m')$, $m, m' = 1, \ldots, M$.

Definition. The trace of A is the sum of the diagonal elements, $trA \equiv \sum_{m=1}^{M} A(m, m)$.

Definition. The matrix A is said to be orthogonal if $AA^\top = I_M$, where I_M is the $M \times M$ identity matrix.

Definition. The determinant of A is defined as

$$|A| \equiv \sum_{\tau} (-1)^{|\tau|} A(1, \tau(1)) \ldots A(M, \tau(M)),$$

where the summation is taken over all permutations τ of the integers $\{1, 2, \ldots, M\}$ and $|\tau|$ equals either $+1$ or -1 depending on whether τ can be written as the product of an even or odd number of transpositions. For $M = 2$, $|A| = A(1,1)A(2,2) - A(1,2)A(2,1)$.

Some linear algebra

Definition. The eigenvalues of the square matrix A are the roots of $|A - \lambda I_M|$, an Mth order polynomial in λ.

These roots, $\lambda(m)$, $m = 1, \ldots, M$, can possibly be complex.

Definition. The (right) eigenvectors of the square matrix A are the non-zero M-vectors satisfying

$$A\gamma = \lambda(m)\gamma, \qquad m = 1, \ldots, M.$$

R function: `eigen`.

Some linear algebra

Results.

- $|A| = \prod_{m=1}^{M} \lambda(m)$.
- $trA = \sum_{m=1}^{M} A(m, m) = \sum_{m=1}^{M} \lambda(m)$.
- The eigenvalues of a symmetric matrix are real.
- The eigenvalues of a non-singular matrix are strictly positive.

Some linear algebra

Spectral Decomposition Theorem. Any $M \times M$ symmetric matrix A, with entries $A(m, m')$, m, $m' = 1, \ldots, M$, can be written as

$$A = \Gamma \Lambda \Gamma^\top = \sum_{m=1}^{M} \Lambda(m) \Gamma(\cdot, m) \Gamma^\top(\cdot, m),$$

where Λ is a diagonal matrix of eigenvalues of A and Γ is an $M \times M$ orthogonal matrix whose columns $\Gamma(\cdot, m)$ are the standardized eigenvectors of A.

R function: `svd`.

See Appendix A, Mardia, Kent, & Bibby (1979), for a summary of linear algebra results.

Principal component analysis

Consider a J-dimensional covariate vector
$X = (X(j) : j = 1, \ldots, J) \in I\!R^J$.

The main objective of principal component analysis (PCA) is to find linear combinations (i.e., weighted averages)
$a^\top X = \sum_j a(j) X(j)$ of the variables $X(j)$ with maximal variance.

That is, PCA seeks linear combinations $a^\top X$ which summarize the data, loosing as little information as possible.
→ Dimensionality reduction.
→ Parsimonious summarization of data.

Principal component analysis

E.g. Given scores on five exams, find the linear combination of the five scores that is "best at separating out" the students.

E.g. Given microarray expression measures on 100 genes, find two independent linear combinations of these expression measures that provide the "best separation" of the mRNA samples.

E.g. Defining an index for cost of living based on a collection of variables.

Principal component analysis

Definition. Consider a J-dimensional covariate vector
$X = (X(j) : j = 1, \ldots, J) \in I\!R^J$, from a data generating distribution P, with mean vector $E_P[X] = \mu$ and covariance matrix $Cov_P[X] = \Sigma$. The principal component transformation is

$$\widetilde{X} \equiv \Gamma^\top (X - \mu),$$

where Γ is a $J \times J$ orthogonal matrix and $\Gamma^\top \Sigma \Gamma = \Lambda$ is diagonal, with entries $\Lambda(1) \geq \Lambda(2) \geq \ldots \geq \Lambda(J) \geq 0$.
The jth principal component of X is

$$\widetilde{X}(j) \equiv \Gamma^\top (\cdot, j)(X - \mu),$$

where $\Gamma(\cdot, j)$ is the jth column of Γ, $j = 1, \ldots, J$.

Principal component analysis

Theorem. Consider a J-dimensional covariate vector $X = (X(j) : j = 1, \ldots, J) \in I\!R^J$, from a data generating distribution P, with mean vector $E_P[X] = \mu$ and covariance matrix $Cov_P[X] = \Sigma$. Then, the J-vector $\widetilde{X} = (\widetilde{X}(j) : j = 1, \ldots, J)$ of principal components satisfies the following.

1. $E_P[\widetilde{X}] = 0$.
2. $Cov_P[\widetilde{X}] = \Lambda$, i.e.,
   $Var_P[\widetilde{X}(j)] = \Lambda(j)$ and
   $Cov_P[\widetilde{X}(j), \widetilde{X}(j')] = 0$, for $j \neq j'$.
3. $Var_P[\widetilde{X}(1)] \geq Var_P[\widetilde{X}(2)] \geq \ldots \geq Var_P[\widetilde{X}(J)]$.
4. $\sum_{j=1}^{J} Var_P[\widetilde{X}(j)] = tr\Sigma$.
5. $\prod_{j=1}^{J} Var_P[\widetilde{X}(j)] = |\Sigma|$.

Principal component analysis

Let $X_1, \ldots, X_n$ be a random sample from the data generating distribution P. The sample analogues of the population principal components are obtained by replacing the data generating distribution P by the empirical distribution P_n.

Let μ_n and Σ_n denote the sample mean vector and covariance matrix, respectively, and let Γ_n denote the $J \times J$ orthogonal matrix with $\Gamma_n^\top \Sigma_n \Gamma_n = \Lambda_n$ diagonal. The *j*th sample principal components are

$$\widetilde{X}_i(j) \equiv \Gamma_n^\top(\cdot, j)(\widetilde{X}_i - \mu_n), \qquad i = 1, \ldots, n.$$

The covariance matrix of the $\widetilde{X}_i$ is $\widetilde{\Sigma}_n = \Lambda_n$.

Finite sample and asymptotic distributional properties of the sample principal components for Gaussian data generating distributions have been well-studied (Mardia, Kent, & Bibby, 1979).

Principal component analysis

The ratio

$$\frac{\Lambda(1) + \ldots + \Lambda(k)}{\Lambda(1) + \ldots + \Lambda(J)}$$

represents the "proportion of total variation explained by the first k principal components".

The reduction in dimension afforded by PCA can be used for visualization purposes. If the first two (or some small number of) principal components explain "most" of the variance, then a scatterplot (scatterplot matrix) of the data on these two (or some small number of) transformed variables can give some information on the distribution of the data, such as, the existence of clusters or one-dimensional structure (e.g., seriation).

Principal component analysis

PCA can be used for variable selection purposes, as follows.

Starting with the smallest eigenvalue, discard the variable with the largest absolute coefficient (i.e., loading) in the corresponding eigenvector.

Move to the next smallest eigenvalue and repeat the process until the desired number of variables have been discarded.

Principal component analysis: R functions

The R function `princomp`, from the `mva` package, performs PCA on a given numeric data matrix or data frame and returns the results as an object of class `princomp`.

Methods for operating on `princomp` objects include: `summary`, `screeplot`, `biplot`.

```
data(iris)
pairs(iris[,-5], pch=c("S","C","V")[as.numeric(iris[,5])], col=as.numeri:(iris[,
X<-as.matrix(iris[,-5])
row.names(X)<-c("S","C","V")[as.numeric(iris[,5])]
iris.pca<-princomp(X)
screeplot(iris.pca)
biplot(iris.pca)
```


Principal component analysis: R functions

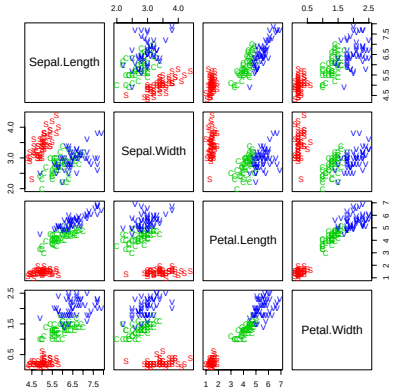


Figure 5: *iris dataset.* `Sepal.Length`, `Sepal.Width`, `Petal.Length`, and `Petal.Width` for 150 flowers.

Principal component analysis: R functions

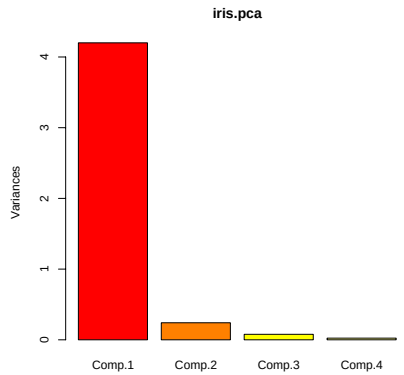


Figure 6: *iris dataset.* Screeplot of the variances (eigenvalues) for each principal component.

Principal component analysis: R functions

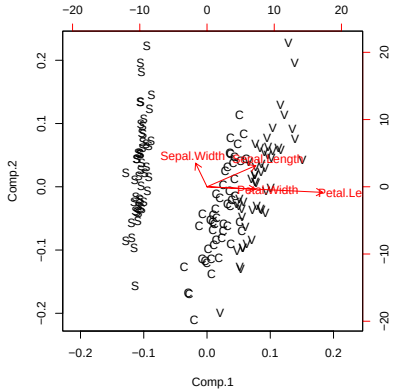


Figure 7: *iris dataset.* Biplot.

Principal component analysis: R functions

```
> iris.pca
Call:
princomp(x = X)
 Standard deviations:
   Comp.1    Comp.2    Comp.3    Comp.4
2.0494032 0.4909714 0.2787259 0.1538707
 4  variables and  150 observations.

> summary(iris.pca)
Importance of components:
                          Comp.1     Comp.2     Comp.3      Comp.4
Standard deviation     2.0494032 0.49097143 0.27872586 0.153870700
Proportion of Variance 0.9246187 0.05306648 0.01710261 0.005212184
Cumulative Proportion  0.9246187 0.97768521 0.99478782 1.000000000

> names(summary(iris.pca))
[1] "sdev"           "loadings"       "center"         "scale"
[5] "n.obs"          "scores"         "call"           "cutoff"
[9] "print.loadings"
```


Multidimensional scaling

Given any $n \times n$ distance matrix D, multidimensional scaling (MDS) is concerned with identifying n points in Euclidean space with a *similar* distance structure D'.

The purpose is to provide a low dimensional representation of the distances, which conveys information on the relationships among the n observations, such as the existence of clusters or one-dimensional structure (e.g., seriation).

Multidimensional scaling

There are different approaches for reducing dimensionality, depending on how one defines similarity between the old and new distance matrices, i.e., depending on the objective or stress function S one seeks to minimize.

- Least-squares scaling: $S(D, D') \equiv (\sum_{i,j}(D(i,j) - D'(i,j))^2)^{1/2}$.
- Sammon mapping: $S(D, D') \equiv \sum_{i,j}(D(i,j) - D'(i,j))^2 / D(i,j)$. Places more emphasis on smaller distances.
- Shepard-Kruskal non-metric scaling: based on ranks, i.e., the order of the distances is more important than their actual values.

Multidimensional scaling

When the distance matrix D is the Euclidean distance matrix between the rows of an $n \times J$ matrix X, there is a duality between principal component analysis (PCA) and MDS.

The k-dimensional classical solution to the MDS problem is given by the centered scores of the n observations on the first k principal components.

The classical solution of MDS in k-dimensional space minimizes the sum of squared differences between the entries of the new and old distance matrices, i.e., is optimal for least-squares scaling.

Multidimensional scaling

As with PCA, the quality of the representation depends on the magnitude of the first k eigenvalues.

The data analyst should choose a value for k that is small enough for ease representation but also corresponds to a substantial "proportion of the distance matrix explained".

Multidimensional scaling

N.B. The results of MDS and PCA reflect not only the choice of a distance (standardization) function, but also the features selected.

If features were selected to separate the data into two groups (e.g., on the basis of two-sample t-statistics), it should come as no surprise that an MDS or PCA plot shows two groups!

Multidimensional scaling: R functions

- `cmdscale`: Classical solution to MDS, in package `mva`.
- `sammon`: Sammon mapping, in package `MASS`.
- `isoMDS`: Kruskal's non-metric MDS, in package `MASS`.

Multidimensional scaling: R functions

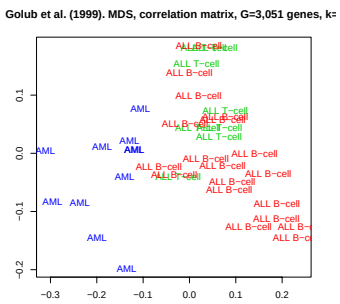


Figure 8: *Golub et al. (1999) leukemia dataset.* Classical MDS, correlation-based distance matrix, $G = 3,051$ genes, $k = 2$, $\frac{|\lambda(1)|+|\lambda(2)|}{\sum_l |\lambda(l)|} = 0.43$.

Multidimensional scaling: R functions

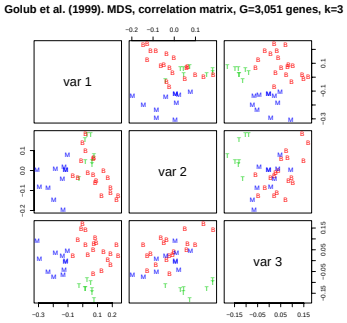


Figure 9: *Golub et al. (1999) leukemia dataset.* Classical MDS, correlation-based distance matrix, $G = 3,051$ genes, $k = 3$, $\frac{|\lambda(1)|+|\lambda(2)|+|\lambda(3)|}{\sum_l |\lambda(l)|} = 0.55$.

Multidimensional scaling: R functions

```
Y<-paste(golub$ALL.AML,golub$T.B.cell)
Y<-sub("NA","",Y)
corr<-cor(exprs(golub))
d<-1-corr
mds<- cmdscale(d, k=2, eig=TRUE)
plot(mds$points, type="n", xlab="", ylab="", main="Golub et al. (1999). MDS, cor
text(mds$points[,1],mds$points[,2],Y, col=rank(unique(Y))[factor(Y)]+1)

> names(mds)
[1] "points" "eig"    "x"      "ac"     "GOF"
> mds$GOF
[1] 0.4299958 0.4407582
```