Lecture 14: Dimension Reduction beyond PCA: NMF, ICA and MDS

Statistical Learning and Data Mining

Xingye Qiao

Department of Mathematical Sciences
Binghamton University

E-mail: qiao@math.binghamton.edu

Read about
NMF: ELSII Ch. 14.6, and SLS 7.3
ICA: ELSII Ch. 14.7
MDS: ELSII Ch. 14.8

Outline

1. Non-Negative Matrix Factorization (NMF)

2. Independent Components Analysis (ICA)

3. Multidimensional Scaling (MDS)

The next section would be . . . . . .

1 Non-Negative Matrix Factorization (NMF)

2 Independent Components Analysis (ICA)

3 Multidimensional Scaling (MDS)

Non-Negative Matrix Factorization

- Idea: $\mathbf{X}_{n\times p} \approx \mathbf{W}_{n\times q}\mathbf{H}_{q\times p} = \sum_{k=1}^{q} \mathbf{W}_{:,k}\mathbf{H}_{k,:}$ with $q \ll p$
- $\mathbf{X} \geq 0$ is a non-negative matrix.
- $\mathbf{W} \geq 0$ is a non-negative matrix for observation scores
- $\mathbf{H} \geq 0$ is a non-negative matrix for factors
- $\mathbf{W}$ and $\mathbf{H}$ are often sparse.

Like PCA except finds patterns with same direction of correlation.

NMF Interpretation

Topic Modeling:

- $\mathbf{X}$ a matrix of news articles (rows) by words (columns) whose entries are word counts.
  - $\mathbf{X}_{n \times p} \approx \sum_{k=1}^{q} \mathbf{W}_{:,k} \mathbf{H}_{k,:}$ is sum of (unknown) topics (e.g. sports, politics, equality, etc.)
  - $\mathbf{X}_{ij} = \sum_{k=1}^{q} W_{ik} H_{kj}$
  - W_{ik}: kth topic in the ith article.
  - H_{kj}: jth word in the kth topic.
- News articles involving topic $k \leftrightarrow$ non-zeros in $\mathbf{W}_{:,k}$
  - E.g. "North Carolina Allows Officials to Refuse to Perform Gay Marriages" (New York Times)
- Words commonly associated with topic $k \leftrightarrow$ non-zeros in $\mathbf{H}_{k,:}$
  - E.g. marriage, gay, Supreme, Court, district, equal, etc.

NMF Criterion - Continuous Data

$$\min_{\substack{\mathbf{W}\in\mathbb{R}^{n\times q},\\ \mathbf{H}\in\mathbb{R}^{q\times p}}} \|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2$$
$$\text{subject to } W_{ik} \geq 0,\; H_{kj} \geq 0$$

(PCA criterion except with non-negativity constraints.)
Algorithm Updates: (Alternating Non-negative Least Squares)

$$\widehat{\mathbf{W}} = \left(\mathbf{X}\mathbf{H}^T(\mathbf{H}^T\mathbf{H})^{-1}\right)_+$$
$$\widehat{\mathbf{H}} = \left((\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{X}\right)_+$$

Local solution.

NMF Criterion - Count Data

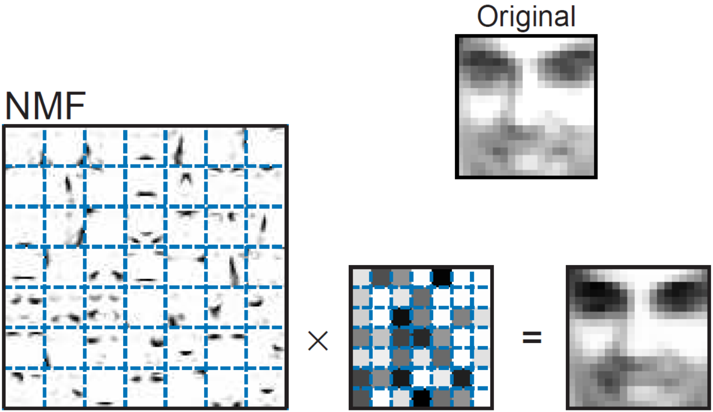
Consider a model $X_{ij} \sim \mathrm{Poisson}((\mathbf{WH})_{ij})$

$$\min_{\substack{\mathbf{W}\in\mathbb{R}^{n\times q},\\ \mathbf{H}\in\mathbb{R}^{q\times p}}} \sum_{i=1}^{n}\sum_{j=1}^{p}(X_{ij}\log((\mathbf{WH})_{ij}) - (\mathbf{WH})_{ij})$$
$$\text{subject to } W_{ik} \geq 0,\ H_{kj} \geq 0$$

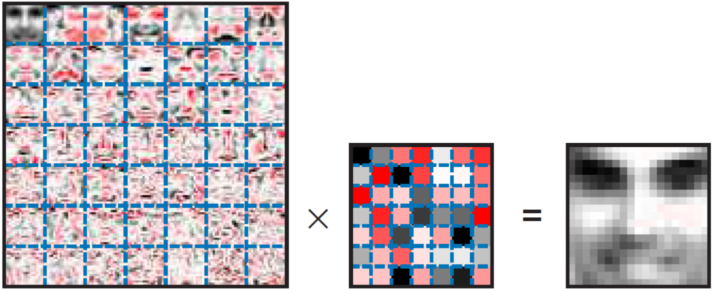
Iterative updates (formula omitted).
Local solution.

Uses

- Dimension Reduction / Pattern Recognition.
    - Similar to PCA (e.g. component scatterplots) except that patterns of correlation found in the same (positive) direction.
- Archetypal Analysis (vs. typical observations i.e. cluster centers)
    - Caricatures (segments; contrastive categorization) vs. Prototypes (averages).
- Soft-clustering.
    - Discussed Next Chapter!


Original
NMF
×
=

PCA



PCA vs. NMF

Similarities:

- Linear Dimension Reduction.
- Interpretation.

Differences:

- Factors are unordered.
- Factors NOT orthogonal.
- Changing q can fundamentally change factors.
- Non-unique, non-global solution.
- Depends on initialization. (Run several times and take the best).

Choosing q

Choice depends on goal:

- Dimension Reduction:
  - Residual sums of squares (or dispersion) - Screeplot.
- Clustering:
  - Consensus, silhouette, etc. (Discussed next lecture!).
- Archetypal Analysis:
  - Sparsity, factor purity, etc.

NMF - Summary

Strengths:

- Interpretation (often more appealing than PCA!).
- Applications - Clustering & Archetypal Analysis.
- Pattern Recognition.
- Others?

Weaknesses:

- Local solutions that depend strongly on q.
- Others?

In `R`: `NMF` package.

The next section would be . . . . . .

1 Non-Negative Matrix Factorization (NMF)

2 Independent Components Analysis (ICA)

3 Multidimensional Scaling (MDS)

ICA

Pre-processing Step: Reduce $\mathbf{X}_{n\times p}$ to $\tilde{\mathbf{X}}_{q\times p}$ with $q < n$ ($q = \#$ of independent sources). (Typically done by PCA!)

Idea: $\tilde{\mathbf{X}}_{q\times p} = \mathbf{A}_{q\times q}\mathbf{S}_{q\times p}$

- Assumption: $\mathbf{X}$ a matrix of q scrambled independent signals.
- $\mathbf{A}_{q\times q}$ Mixing Matrix - denotes how signals are scrambled to form sources in data.
- $\mathbf{S}_{q\times p}$ Signal Matrix - each row of $\mathbf{S}$ is an independent signal.

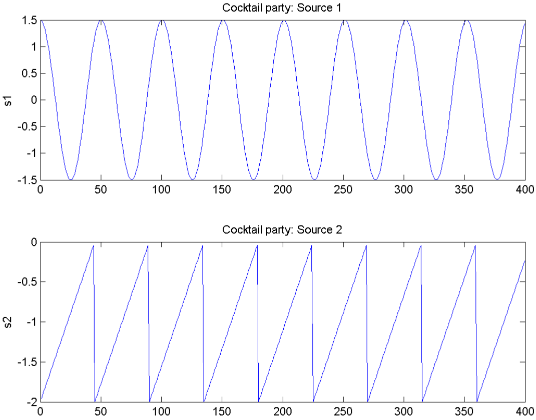
PCA finds uncorrelated, but not independent signals (independence $\neq$ no correlation)

Independent Component Analysis (ICA)

- ICA is a multivariate statistical technique that seeks to uncover hidden variables.
- Basic form is a linear dimension reduction: find *interesting* directions $\boldsymbol{u}_i$, such that the projected scores $(S_i = \boldsymbol{u}_i'\boldsymbol{X},\ i = 1, \ldots, k)$ are *independent* of each other.
- Fairly new technique: first appeared in Cardoso 1993.
- Good source of information can be found at `http://research.ics.aalto.fi/ica/`
- Motivating example: Cocktail party problem. Fun examples at `http://research.ics.aalto.fi/ica/cocktail/cocktail_en.cgi`
- Demo that seem to be working: `http://mcdermottlab.mit.edu/cocktail_examples/index.html` `http://cnl.salk.edu/~tewon/Blind/blind_audio.html`

Cocktail party problem

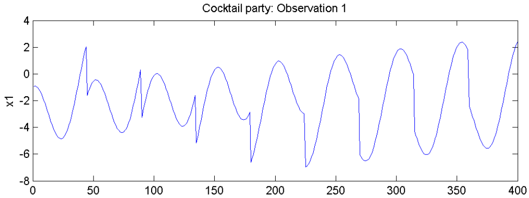
- Hear several simultaneous conversations;
- Wish to separate them.
- Conversations are modeled as time series: $s_1(t)$ and $s_2(t)$.

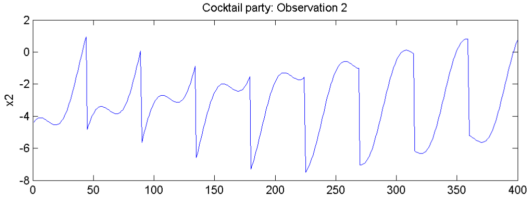


Cocktail party problem

- What we actually hear is a mixture of both conversations

$$
\begin{aligned}
x_1(t) &= a_{11}s_1(t) + a_{12}s_2(t), \\
x_2(t) &= a_{21}s_1(t) + a_{22}s_2(t).
\end{aligned}
$$





Cocktail party problem

- Without knowing the actual source and the mixing matrix $\mathbf{A} = (a_{ij})$, ICA tries to recover the source $\boldsymbol{S} = (S_1, S_2)'$ from data $\boldsymbol{X} = (X_1, X_2)'$;

$$\boldsymbol{X} = (X_1, X_2)' = \mathbf{A}\boldsymbol{S}.$$

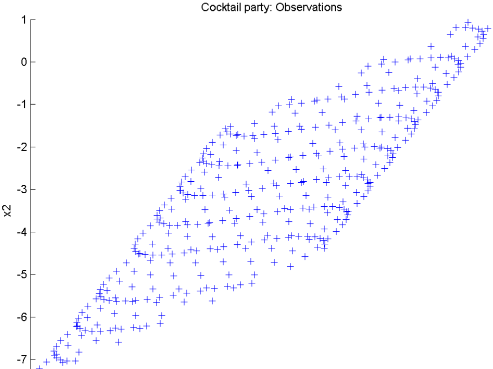
- Since $\boldsymbol{S} = \mathbf{W}\boldsymbol{X}$ for $\mathbf{W} = \mathbf{A}^{-1} = \begin{pmatrix} \boldsymbol{w}_1' \\ \boldsymbol{w}_2' \end{pmatrix}$, the method suggests a linear dimension reduction, with projection vectors $\boldsymbol{w}_i$ and scores S_i:

$$S_i = \boldsymbol{w}_i'\boldsymbol{X}, \ \ i = 1, 2.$$

- In general, the sources are non-Gaussian (with at most one exception). **Otherwise, little hope to separate them.**

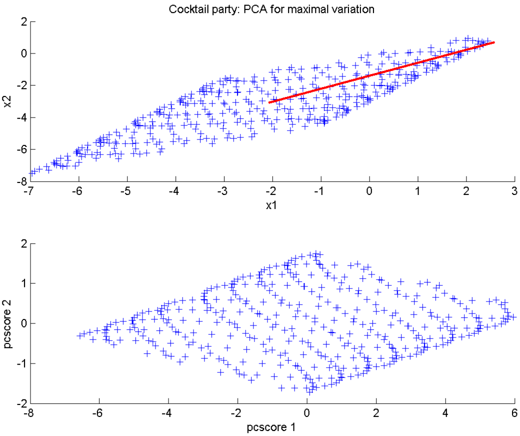
Cocktail party problem

- Ignoring the order between time stamps, the data is bivariate (two dimensions).
- View the sampling size, i.e. the number of time-sampling points, as the sample size (each sampling point is one observation).
- Can PCA help?



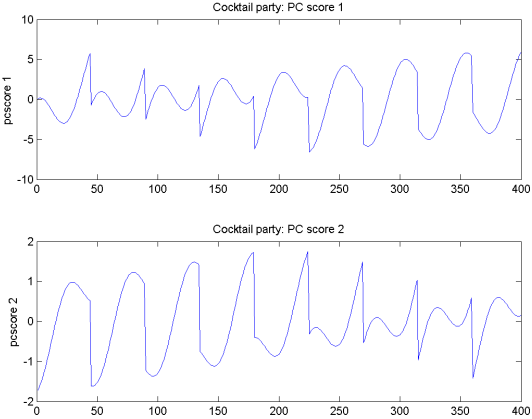
Cocktail party problem–PCA?

- $\boldsymbol{w}_i$ as PC directions is wrong for source separation.
- Since PCA finds the direction of greatest variation.



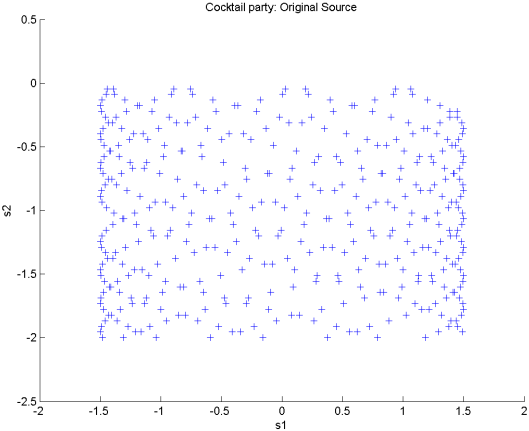
Cocktail party problem–PCA?

- PCA scores are just another mixtures of signals-no good.



Cocktail party problem

- Scatters of original source (which is unknown in practice)
- Understood as uniform distribution on a rectangle
- Both marginal distributions are uniform and they are independent of each other

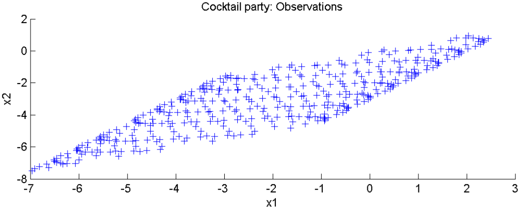


Cocktail party problem–Trial & Error

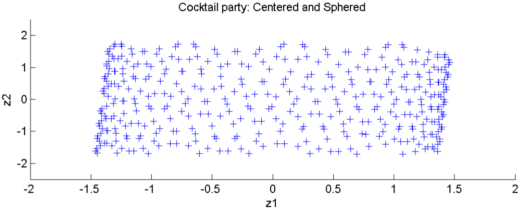
- Centering and sphering (a.k.a. whitening) the observations

$$\boldsymbol{z} = \widehat{\Sigma}^{-\frac{1}{2}}(\boldsymbol{x} - \hat{\mu}).$$

- Find directions in the transformed ($\boldsymbol{z}$-) space



Cocktail party: Observations

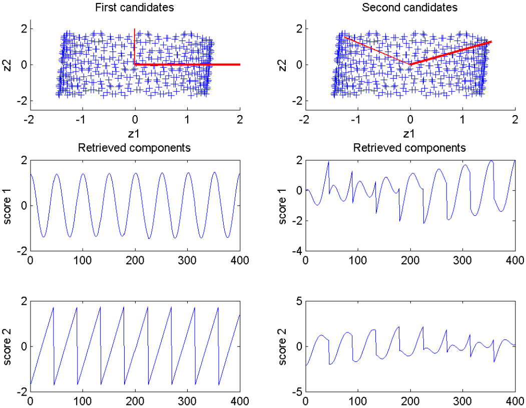


Cocktail party: Centered and Sphered

- First candidate $\boldsymbol{U} = [\boldsymbol{u}_1 \ \boldsymbol{u}_2] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ surprisingly well-separates two signals.
- Back to original $\boldsymbol{x}$-space,

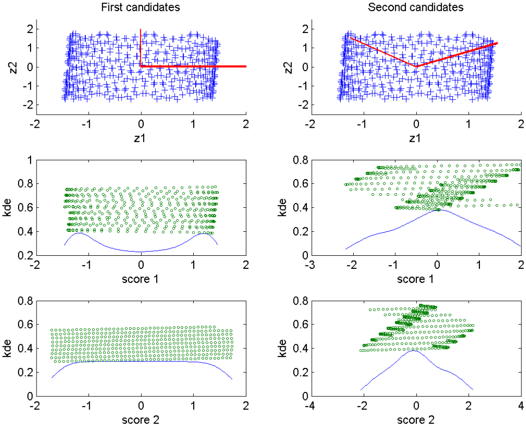
$$s_i = \boldsymbol{u}_i'\boldsymbol{z} = \boldsymbol{u}_i'\widehat{\Sigma}^{-\frac{1}{2}}(\boldsymbol{x} - \hat{\mu}) = \widehat{\boldsymbol{w}}_i'\boldsymbol{x} + \boldsymbol{c},$$

where $\widehat{\boldsymbol{w}}_i = \widehat{\Sigma}^{-\frac{1}{2}}\boldsymbol{u}_i = \{i\text{th column of } \widehat{\Sigma}^{-\frac{1}{2}}\}$.



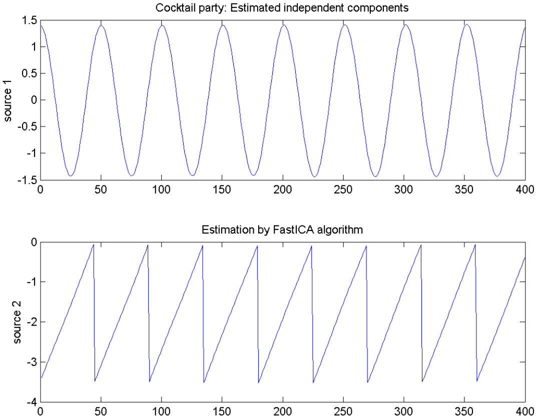
Cocktail party problem–Trial and Error

- Non-Gaussianity is the key
  1st cand. (left) less Gaussian than 2nd (right)
- Systematic way of separating sources?
  Ans: Measuring independence through non-Gaussianity



Cocktail party problem–ICA

- Solution by `FastICA` algorithm (to be discussed)



Independent Component Analysis

- The cocktail party is an example of ICA models.
- General ICA model– Observation is a mixed source plus error:

  $$\boldsymbol{X} = f(\boldsymbol{S}) + \boldsymbol{e},$$

  where the m sources $S_1, \ldots, S_m$ are standardized and independent.
- Special case: Noiseless linear mixing ICA model

  $$f(\boldsymbol{S}) = \mathbf{A}\boldsymbol{S}, \; \mathrm{Var}(\boldsymbol{e}) = 0.$$

- If the "number of observations" (dimension of $\boldsymbol{X}$) equals the number of sources (dimension of $\boldsymbol{S}$, m above), then there exists an unmixing matrix $\mathbf{W} = \mathbf{A}^{-1}$ such that

  $$\boldsymbol{X} = \mathbf{A}\boldsymbol{S} \Leftrightarrow \boldsymbol{S} = \mathbf{W}\boldsymbol{X},$$

  so that the sources are exactly recovered from $\boldsymbol{X}$.

- ICA finds an estimate $\widehat{\mathbf{W}}$ of $\mathbf{W}$ so that the components of $\boldsymbol{Y} = \widehat{\mathbf{W}}\boldsymbol{X}$ are as **independent** (and as non-Gaussian) as possible.

ICA vs PCA

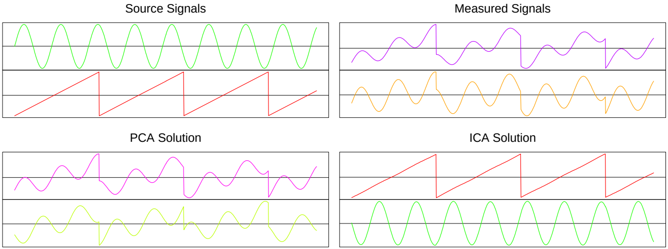


FIGURE 14.37. *Illustration of ICA vs. PCA on artificial time-series data. The upper left panel shows the two source signals, measured at* 1000 *uniformly spaced time points. The upper right panel shows the observed mixed signals. The lower two panels show the principal components and independent component solutions.*

ICA Algorithms

Fast ICA:

- Finds rotations of $\mathbf{X}$ that are "non-Gaussian".
- Uses non-Gaussian contrast functions:
  - $g(x) = x^4$.
  - $g(x) = tanh(x)$.
- Generalization of projection pursuit.

Others:

- Infomax (entropy).

Goal: Find components that are statistically independent (beyond the sense of "0-correlation"), or as independent as possible.

PCA vs. ICA

Similarities:

- Linear Dimension Reduction.
- Interpretation.

Differences: For ICA,

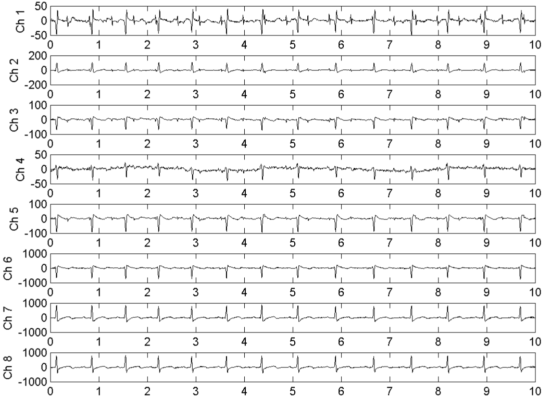
- Factors are unordered.
- Not identifiable with respect to scaling and rotation, hence Non-unique.
- Factors NOT orthogonal.
- Changing q can fundamentally change factors.
- No optimization criterion to evaluate solution.

Cutaneous potential recordings of a pregnant woman

- Section 15.3.2 Izenman
- L. De Lathauwer, B. De Moor, J. Vandewalle, “Fetal Electrocardiogram Extraction by Blind Source Subspace Separation”, IEEE Trans. Biomedical Engineering, Vol. 47, No. 5, May 2000. search for the title at `http://homes.esat.kuleuven.be/~smc/daisy/daisydata.html`
- Monitoring fetal heart activity of a pregnant woman to assess health of the fetus.
- Multichannel electro’cardio’gram (ECG) is used to maternal and fetal electrical activity.
- Challenge: Maternal ECG signal is stronger than fetal, contaminated by respiration.
- Goal: Separate fetal heart activity from mixed signal.

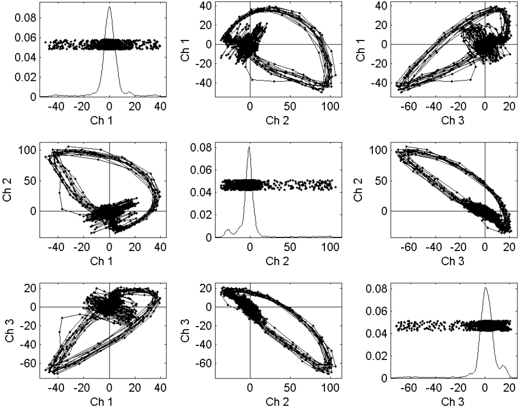
Cutaneous recordings of pregnant woman

- 8-channel recordings of ECG over time (10 secs)
- First five are measured near fetus
- Last three are on the mother's chest



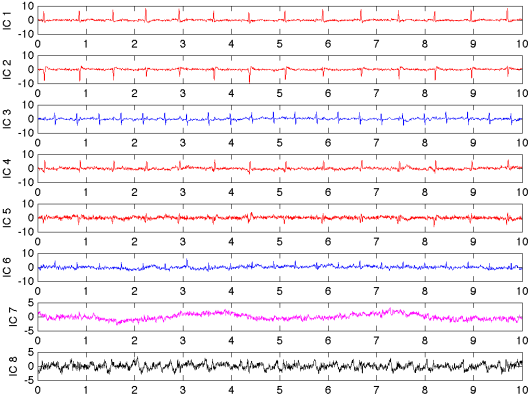
Cutaneous recordings of pregnant woman

- Marginal and joint distributions of the input $\boldsymbol{X}$ are severely non-Gaussian



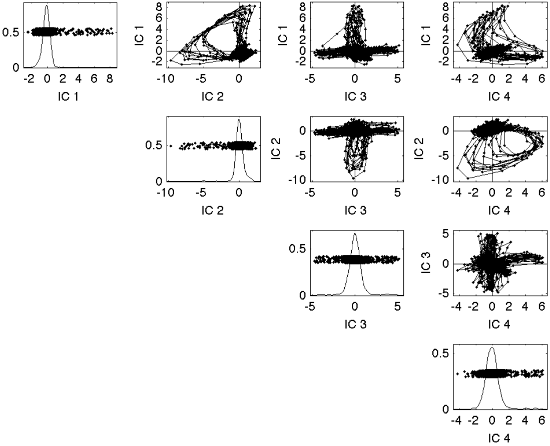
Cutaneous recordings of pregnant woman

- Result from a `"FastICA"` algorithm
- cardiac rhythms of the mother, cardiac rhythms of the fetus,
- respiration component, sensor noise



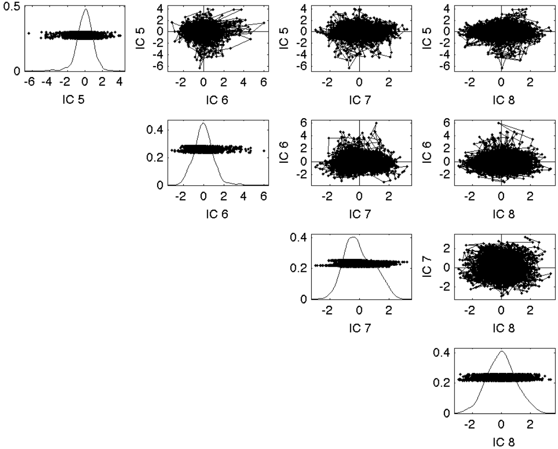
Potential problems

- The order of ICs given by non-Gaussianity
- Are the components really independent?



Potential problems

- The order of ICs given by non-Gaussianity
- Are the components really independent? *as independent as possible*



fMRI application

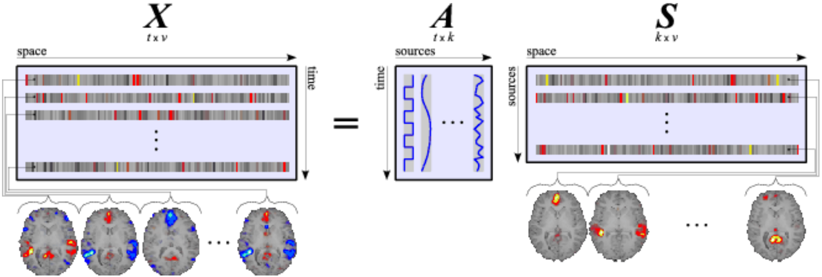


Image from http://users.ics.aalto.fi/whyj/publications/thesis/thesis_node8.html

The next section would be . . . . . .

1 Non-Negative Matrix Factorization (NMF)

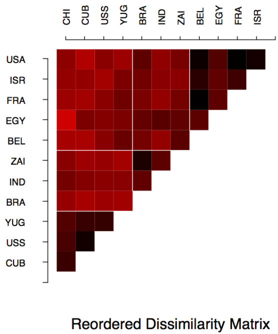
2 Independent Components Analysis (ICA)

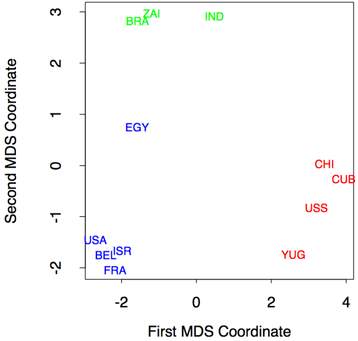
3 Multidimensional Scaling (MDS)

Multidimensional scaling

Goal of Multidimensional scaling (MDS): Given pairwise dissimilarities, reconstruct a map that preserves distances.

- From any dissimilarity (no need to be a metric)
- Reconstructed map has coordinates $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip})$ and the natural distance ($\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$)





Multidimensional scaling

- MDS is a family of different algorithms, each designed to arrive at some optimal low-dimensional configuration ($p = 2$ or 3)
- MDS methods include
  1. Classical MDS
  2. Metric MDS
  3. Non-metric MDS

Perception of Color in human vision

- To study the perception of color in human vision (Ekman, 1954, Izenman 13.2.1)
- 14 colors differ only in their hue (i.e., wavelengths from 434 μm to 674 μm)
- 31 people rate for each of $\binom{14}{2}$ pairs of colors on a five-point scale from 0 (no similarity at all) to 4 (identical).
- Average of 31 ratings for each pair (representing similarity) is then scaled (by $1/4$) and subtracted from 1 to represent dissimilarities

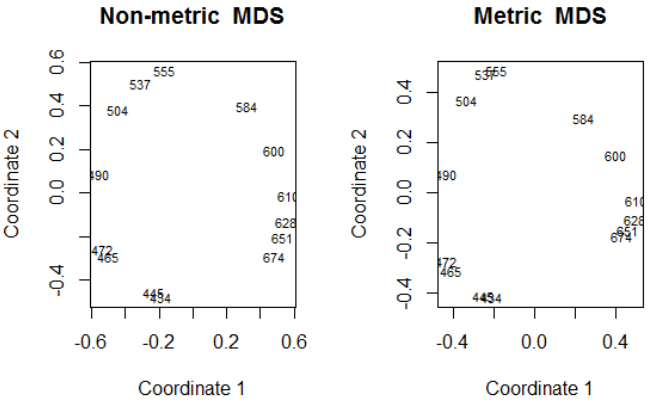
Perception of Color in human vision

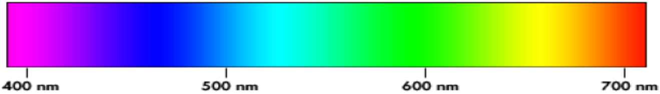
The resulting 14 × 14 dissimilarity matrix is symmetric, and contains zeros in the diagonal. MDS seeks a 2D configuration to represent these colors.

	434	445	465	472	490	504	537	555	584	600	610	628	651
445	0.14												
465	0.58	0.50											
472	0.58	0.56	0.19										
490	0.82	0.78	0.53	0.46									
504	0.94	0.91	0.83	0.75	0.39								
537	0.93	0.93	0.90	0.90	0.69	0.38							
555	0.96	0.93	0.92	0.91	0.74	0.55	0.27						
584	0.98	0.98	0.98	0.98	0.93	0.86	0.78	0.67					
600	0.93	0.96	0.99	0.99	0.98	0.92	0.86	0.81	0.42				
610	0.91	0.93	0.98	1.00	0.98	0.98	0.95	0.96	0.63	0.26			
628	0.88	0.89	0.99	0.99	0.99	0.98	0.98	0.97	0.73	0.50	0.24		
651	0.87	0.87	0.95	0.98	0.98	0.98	0.98	0.98	0.80	0.59	0.38	0.15	
674	0.84	0.86	0.97	0.96	1.00	0.99	1.00	0.98	0.77	0.72	0.45	0.32	0.24

Perception of Color in human vision

MDS reproduces the well-known two-dimensional *color circle*.





Distance, dissimilarity and similarity

Distance, dissimilarity and similarity (or proximity) are defined for any pair of objects in any space. In mathematics, a distance function (that gives a distance between two objects) is also called *metric*, satisfying

1. $d(x, y) \geq 0$,
2. $d(x, y) = 0$ if and only if $x = y$,
3. $d(x, y) = d(y, x)$,
4. $d(x, z) \leq d(x, y) + d(y, z)$.

Given a set of dissimilarities, one can ask whether these values are distances and, moreover, whether they can even be interpreted as Euclidean distances

Euclidean and non-Euclidean distance

Given a dissimilarity (distance) matrix $D = (d_{ij})$, MDS seeks to find $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^p$ (called a configuration) so that

$$d_{ij} \approx \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \text{ as close as possible.}$$

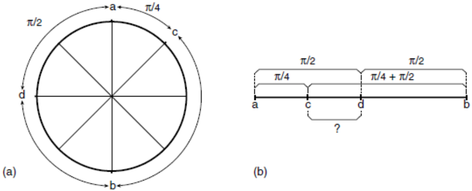
Oftentimes, for some large p, there always exists a configuration $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ with exact/perfect distance match $d_{ij} \equiv \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$. In such a case the distance d involved is called a Euclidean distance. There are, however, cases where the dissimilarity is distance, but there exists no configuration in any p with perfect match

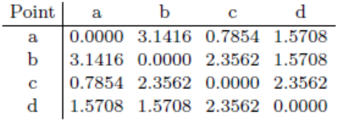
$$d_{ij} \neq \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2, \text{ for some } i, j.$$

Such a distance is called non-Euclidean distance.

non-Euclidean distance

- Radian distance function on a circle is a metric.
- Cannot be embedded in $\mathbb{R}^1$ (in other words, cannot find $x_1, \ldots, x_4 \in \mathbb{R}$ to match the distance)
  (Not for any $\mathbb{R}^p$, not shown here)





Point	a	b	c	d
a	0.0000	3.1416	0.7854	1.5708
b	3.1416	0.0000	2.3562	1.5708
c	0.7854	2.3562	0.0000	2.3562
d	1.5708	1.5708	2.3562	0.0000

- Nevertheless, MDS seeks to find an optimal configuration $\boldsymbol{x}_i$ that gives $d_{ij} \approx \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$ as close as possible.

classical Multidimensional Scaling (cMDS)–theory

Also known as Principal Coordinates Analysis (PCoA). Suppose for now we have Euclidean distance matrix $D = (d_{ij})$.

The objective of classical Multidimensional Scaling (cMDS) is to find $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ so that $\|\boldsymbol{x}_i - \boldsymbol{x}_j\| = d_{ij}$. Such a solution is not unique, because if $\mathbf{X}$ is the solution, then $\boldsymbol{x}_i^* := \boldsymbol{x}_i + c$, $c \in \mathbb{R}^q$ also satisfies $\left\|\boldsymbol{x}_i^* - \boldsymbol{x}_j^*\right\| = \|(\boldsymbol{x}_i + c) - (\boldsymbol{x}_j + c)\| = \|\boldsymbol{x}_i - \boldsymbol{x}_j\| = d_{ij}$. Any location c can be used, but the assumption of centered configuration, i.e.,

$$\sum_{i=1}^{n} \boldsymbol{x}_i = \mathbf{0} \tag{1}$$

serves well for the purpose of dimension reduction.

In short, the cMDS finds the centered configuration $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^q$ for some $q \leq n-1$ so that their pairwise distances are the same as those corresponding distances in D.

We may find the $n \times n$ Gram matrix $\mathbf{B} = \mathbf{X}'\mathbf{X}$, rather than $\mathbf{X}$. The Gram matrix is the matrix of inner products. Denote the *ij*th element of $\mathbf{B}$ as b_{ij}. We have

$$d_{ij}^2 = b_{ii} + b_{jj} - 2b_{ij}, \tag{2}$$

from the fact $d_{ij}^2 = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = \boldsymbol{x}_i'\boldsymbol{x}_i + \boldsymbol{x}_j'\boldsymbol{x}_j - 2\boldsymbol{x}_i'\boldsymbol{x}_j$. Remember, we seek to solve b_{ij}'s from d_{ij}'s (see the next few slides.)

- The centering constraint (1) leads to

  $$\sum_{i=1}^{n} b_{ij} = \sum_{i=1}^{n} \boldsymbol{x}_i' \boldsymbol{x}_j = 0,$$

  for $j = 1, \ldots, n$. Hence, the sum of each row or column of $\mathbf{B}$ is 0.
- With a notation $T = \mathrm{trace}(\mathbf{B}) = \sum_{i=1}^{n} b_{ii}$, we have

  $$\sum_{i=1}^{n} d_{ij}^2 = T + nb_{jj}, \ \sum_{j=1}^{n} d_{ij}^2 = T + nb_{ii}, \ \sum_{j=1}^{n} \sum_{i=1}^{n} d_{ij}^2 = 2nT. \quad (3)$$

Combining (2) and (3), the solution is unique:

$$b_{ij} = -1/2(d_{ij}^2 - d_{\cdot j}^2 - d_{i\cdot}^2 + d_{\cdot\cdot}^2),$$

where $d_{\cdot j}^2$ is the average of $\left\{d_{ij}^2, i = 1, \ldots, n\right\}$ for each j, $d_{i\cdot}^2$ is the average of $\left\{d_{ij}^2, j = 1, \ldots, n\right\}$ for each i, and $d_{\cdot\cdot}^2$ is the average of $\left\{d_{ij}^2, i, j = 1, \ldots, n\right\}$, or equivalently

$$\mathbf{B} = -1/2\mathbf{C}\mathbf{D}_2\mathbf{C}',$$

where $\mathbf{D}_2 = \{d_{ij}^2\}$ and $\mathbf{C}$ is the centering matrix.
A solution $\mathbf{X}$ is then given by the eigen-decomposition of $\mathbf{B}(:= \mathbf{X}'\mathbf{X})$. That is, for $\mathbf{B} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}'$,

$$\mathbf{X} = \boldsymbol{\Lambda}^{1/2}\mathbf{V}'. \tag{4}$$

- Each row of $\mathbf{X}$ is along the same direction as a row of $\mathbf{V}^T$. Hence $\mathbf{X}\mathbf{X}'$ ($d \times d$) is diagonal.
- Consider PCA based on $\{\boldsymbol{x}_i, i = 1, \ldots, n\}$ (centered) through singular-value-decomposition. We have $\mathbf{X} = \mathbf{U}\Theta\mathbf{V}'$, and the PC scores are $\mathbf{Z} = \mathbf{U}'\mathbf{X} = \Theta\mathbf{V}'$.
  - It would turn out that $\mathbf{U} = \mathbb{I}_q$ and $\Theta = \Lambda^{1/2}$
- The first coordinate of $\mathbf{X}$ has the largest variation (recall the interpretation of $\mathbf{X}$ using PCA scores above)
- If we wish to reduce the dimension to $p \leq q$, then the first p rows of $\mathbf{X}$, $\mathbf{X}_{(p)}$, best preserves the distances d_{ij} among all other linear dimension reduction of $\mathbf{X}$.

$$\mathbf{X}_{(p)} = \Lambda_p^{1/2}\mathbf{V}_p',$$

where Λ_p is the first $p \times p$ submatrix of Λ, $\mathbf{V}_p$ is the first p columns of $\mathbf{V}$.

To see that the first p coordinates of $\boldsymbol{x}_i$ indeed best preserve the distance, note that the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j \in \mathbb{R}^q$ is

$$d_{ij}^2 = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = \left\|\boldsymbol{x}_i^{(1-p)} - \boldsymbol{x}_j^{(1-p)}\right\|^2 + \left\|\boldsymbol{x}_i^{(*)} - \boldsymbol{x}_j^{(*)}\right\|^2$$

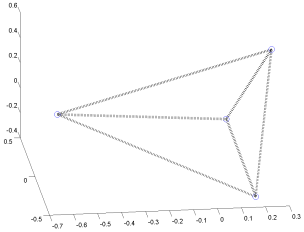
where $\boldsymbol{x}_i^{(1-p)}$ is the subvector of $\boldsymbol{x}_i$ which we keep and $\boldsymbol{x}_i^{(*)}$ is the part we throw away. It is easy to see that since the variation of $\boldsymbol{x}_i^{(*)}$ is small, the value of $\left\|\boldsymbol{x}_i^{(*)} - \boldsymbol{x}_j^{(*)}\right\|^2$ is small too (on average).

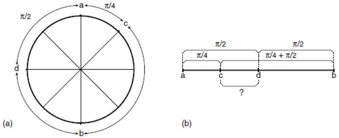
cMDS remarks

- cMDS gives configurations $\mathbf{X}_{(p)}$ in $\mathbb{R}^p$ for any dimension $1 \leq p \leq q$.
- Configuration is centered.
- Coordinates are given by the principal scores, ordered from largest-to-smallest variation.
- Dimension reduction from $X = X_{(q)}$ to $X_{(p)}$ $(p < q)$ is same as PCA (cutting some PC scores out).
- Leads to exact solution if the dissimilarity is based on Euclidean distances
- *Can also be used for non-Euclidean distances, in fact, for any dissimilarities*.

cMDS examples

- Consider two working examples:
  1. with Euclidean geometry (tetrahedron – unit edge length)
  2. with circular geometry





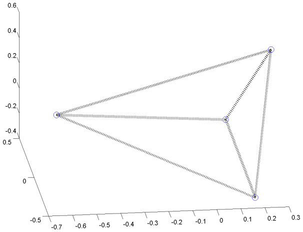
- And the airline distances example (Izenman 13.1.1)

cMDS examples: tetrahedron

Pairwise distance matrix for tetrahedron (with distance 1)

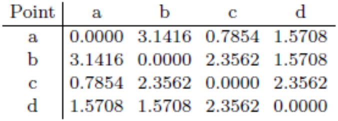
$$D = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix},$$

leading to the gram matrix $B_{(4\times4)}$ with eigenvalues $(.5, .5, .5, 0)$. Using dimension $p = 3$, we have perfectly retrieved the tetrahedron.



cMDS examples: circular distances

Pairwise distance matrix



Point	a	b	c	d
a	0.0000	3.1416	0.7854	1.5708
b	3.1416	0.0000	2.3562	1.5708
c	0.7854	2.3562	0.0000	2.3562
d	1.5708	1.5708	2.3562	0.0000

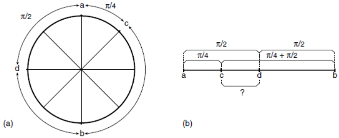
leading to the gram matrix $B_{(4\times 4)}$ with eigenvalues

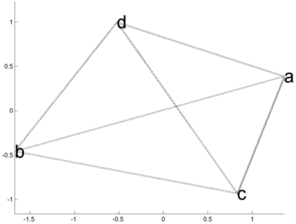
$$\mathrm{diag}(\Lambda) = (5.6117, -1.2039, -0.0000, 2.2234)$$

In retrieving the coordinate matrix X, we cannot take a squareroot of Λ since it gives complex numbers.
Remedy: *Keep only positive eigenvalues and corresponding coordinates*. In this case, take coordinates 1 and 4. This is the price we pay to represent non-Euclidean geometry by Euclidean geometry.

cMDS examples: circular distances

Using dimension $p = 2$ (cannot use $p > 2$), configuration $\mathbf{X}_{(2)}$ is





Compare the original distance matrix $\mathbf{D}$ and approximated distance matrix $\hat{\mathbf{D}} = \|x_i - x_j\|_2$:

$$\begin{pmatrix} 0 & 3.1416 & 0.7854 & 1.5708 \\ 3.1416 & 0 & 2.3562 & 1.5708 \\ 0.7854 & 2.3562 & 0 & 2.3562 \\ 1.5708 & 1.5708 & 2.3562 & 0 \end{pmatrix}, \quad \hat{\mathbf{D}} = \begin{pmatrix} 0 & 3.1489 & 1.4218 & 1.9784 \\ 3.1489 & 0 & 2.5482 & 1.8557 \\ 1.4218 & 2.5482 & 0 & 2.3563 \\ 1.9784 & 1.8557 & 2.3563 & 0 \end{pmatrix}$$

cMDS examples: Airline distances

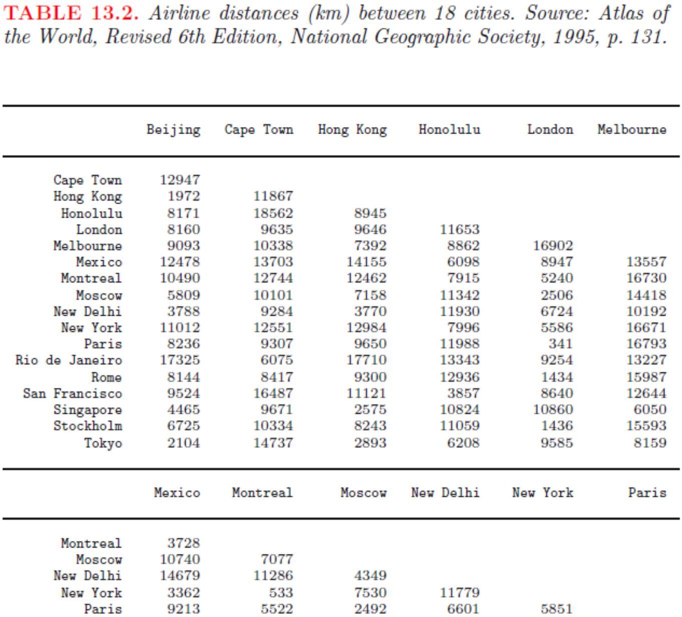


TABLE 13.2. *Airline distances (km) between 18 cities. Source: Atlas of the World, Revised 6th Edition, National Geographic Society, 1995, p. 131.*

	Beijing	Cape Town	Hong Kong	Honolulu	London	Melbourne
Cape Town	12947					
Hong Kong	1972	11867				
Honolulu	8171	18562	8945			
London	8160	9635	9646	11653		
Melbourne	9093	10338	7392	8862	16902	
Mexico	12478	13703	14155	6098	8947	13557
Montreal	10490	12744	12462	7915	5240	16730
Moscow	5809	10101	7158	11342	2506	14418
New Delhi	3788	9284	3770	11930	6724	10192
New York	11012	12551	12984	7996	5586	16671
Paris	8236	9307	9650	11988	341	16793
Rio de Janeiro	17325	6075	17710	13343	9254	13227
Rome	8144	8417	9300	12936	1434	15987
San Francisco	9524	16487	11121	3857	8640	12644
Singapore	4465	9671	2575	10824	10860	6050
Stockholm	6725	10334	8243	11059	1436	15593
Tokyo	2104	14737	2893	6208	9585	8159

	Mexico	Montreal	Moscow	New Delhi	New York	Paris
Montreal	3728					
Moscow	10740	7077				
New Delhi	14679	11286	4349			
New York	3362	533	7530	11779		
Paris	9213	5522	2492	6601	5851	

cMDS examples: Airline distances

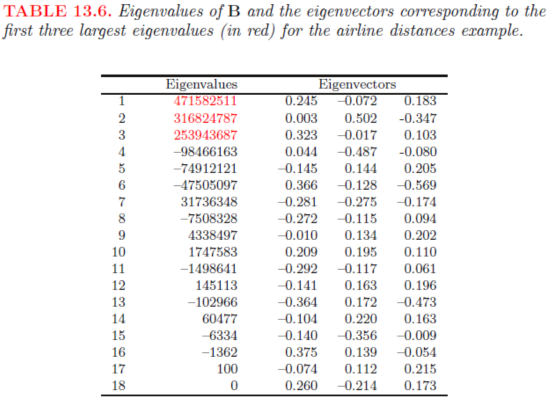


TABLE 13.6. *Eigenvalues of* B *and the eigenvectors corresponding to the first three largest eigenvalues (in red) for the airline distances example.*

	Eigenvalues	Eigenvectors		
1	471582511	0.245	–0.072	0.183
2	316824787	0.003	0.502	-0.347
3	253943687	0.323	–0.017	0.103
4	–98466163	0.044	–0.487	-0.080
5	–74912121	–0.145	0.144	0.205
6	–47505097	0.366	–0.128	–0.569
7	31736348	–0.281	–0.275	–0.174
8	–7508328	–0.272	–0.115	0.094
9	4338497	–0.010	0.134	0.202
10	1747583	0.209	0.195	0.110
11	–1498641	–0.292	–0.117	0.061
12	145113	–0.141	0.163	0.196
13	–102966	–0.364	0.172	–0.473
14	60477	–0.104	0.220	0.163
15	–6334	–0.140	–0.356	–0.009
16	–1362	0.375	0.139	–0.054
17	100	–0.074	0.112	0.215
18	0	0.260	–0.214	0.173

- Airline distance is non-Euclidean
- Take the first 3 largest eigenvalues (inspection of scree plot)

cMDS examples: Airline distances

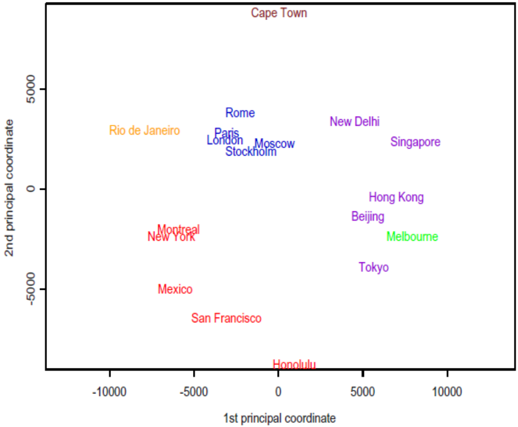


FIGURE 13.1. *Two-dimensional map of 18 world cities using the classical scaling algorithm on airline distances between those cities. The colors reflect the different continents: Asia (purple), North America (red), South America (orange), Europe (blue), Africa (brown), and Australasia (green).*

cMDS examples: Airline distances

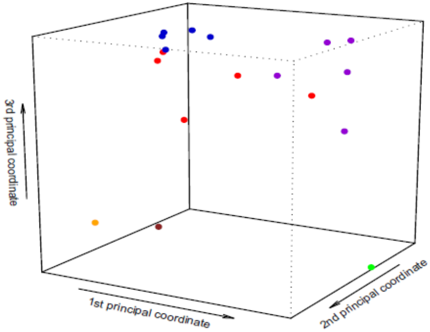


FIGURE 13.2. *Three-dimensional map of 18 world cities using the classical scaling algorithm on airline distances between those cities. The colors reflect the different continents: Asia (purple), North America (red), South America (yellow), Europe (blue), Africa (brown), and Australasia (green).*

MDS - Stress Functions

- Input: $\mathbf{D}_{n \times n}$: d_{ij} is the dissimilarity between objects i and j.
- Output: $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n \in \mathbb{R}^n$ that preserve the distances (dissimilarity)

Stress Functions: Let $\hat{d}_{ij} = \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2$.

- Least squares or Kruskal-Shephard Scaling:

$$\sum_{i<j} (\hat{d}_{ij} - d_{ij})^2$$

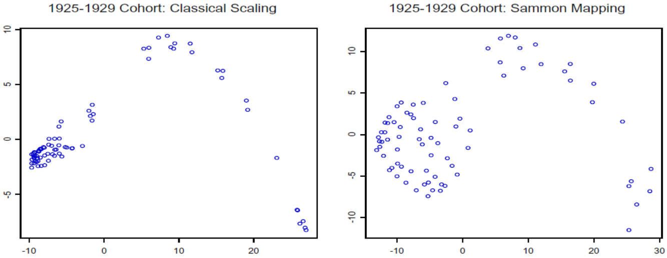
- Sammon mapping: preserve smaller pairwise distances

$$\sum_{i<j} \frac{(\hat{d}_{ij} - d_{ij})^2}{d_{ij}}$$

- Shepard-Kruskal **nonmetric** scaling [$\theta(\cdot)$ is a monotone increasing function which preserve the order!]

$$\sum_{i<j} (\theta(\hat{d}_{ij}) - d_{ij})^2.$$

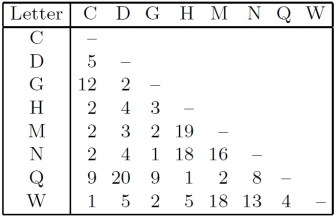
cMDS vs. Sammon Mapping



- Izenman Figure 13.9 (lower panel)
- Results of cMDS and Sammon mapping for $p = 2$: Sammon mapping better preserves inter-distances for smaller dissimilarities, while proportionally squeezes the inter-distances for larger dissimilarities.
- There is NO ground truth here.

Non-metric MDS Example: Letter recognition

Wolford and Hollingsworth (1974) were interested in the confusions made when a person attempts to identify letters of the alphabet viewed for some milliseconds only. A confusion matrix was constructed that shows the frequency with which each stimulus letter was mistakenly called something else. A section of this matrix is shown in the table below.



Letter	C	D	G	H	M	N	Q	W
C	–							
D	5	–						
G	12	2	–					
H	2	4	3	–				
M	2	3	2	19	–			
N	2	4	1	18	16	–		
Q	9	20	9	1	2	8	–	
W	1	5	2	5	18	13	4	–

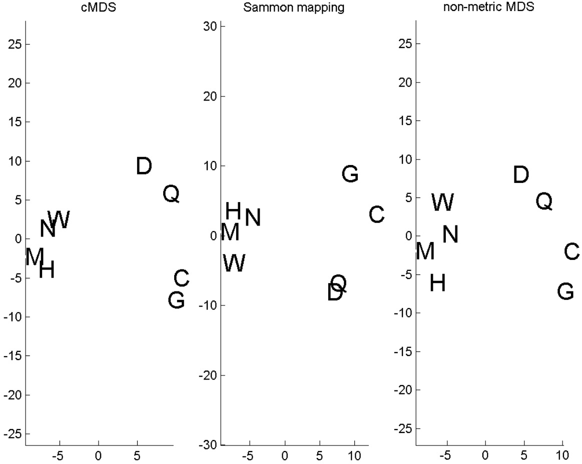
Is this a dissimilarity matrix?

Example: Letter recognition

- How to deduce dissimilarities from a similarity matrix?
  From similarities δ_{ij}, choose a maximum similarity $c \geq \max \delta_{ij}$, so that $d_{ij} = \begin{cases} c - \delta_{ij}, & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases}$. $d_{ij} \uparrow$, i and j are less similar.
- Which method is more appropriate?
  Because we have deduced dissimilarities from similarities, the absolute dissimilarities d_{ij} depend on the value of personally chosen c. This is the case where the non-metric MDS makes most sense.
  However, we will also see that metric scalings (cMDS and Sammon mapping) does the job as well.
- How many dimension?
  By inspection of eigenvalues from the cMDS solution.

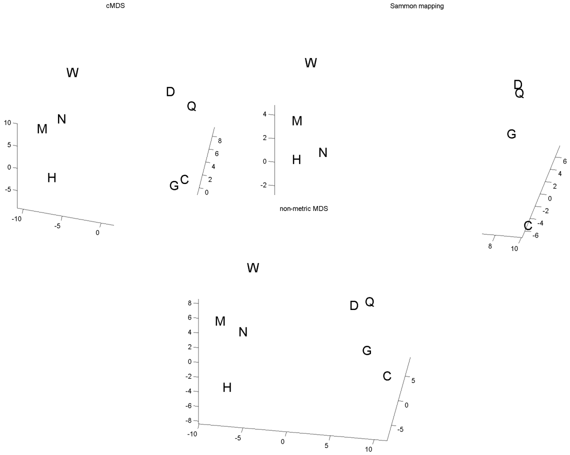
Letter recognition

- First choose $c = 21 (= \max \delta_{ij} + 1)$.
- Compare MDS with $p = 2$, from cMDS, Sammon mapping, and non-metric scaling (stress1):



Letter recognition:

- First choose $c = 21 = \max \delta_{ij} + 1$.
- Compare MDS with $p = 3$, from cMDS, Sammon mapping, and non-metric scaling (stress1):



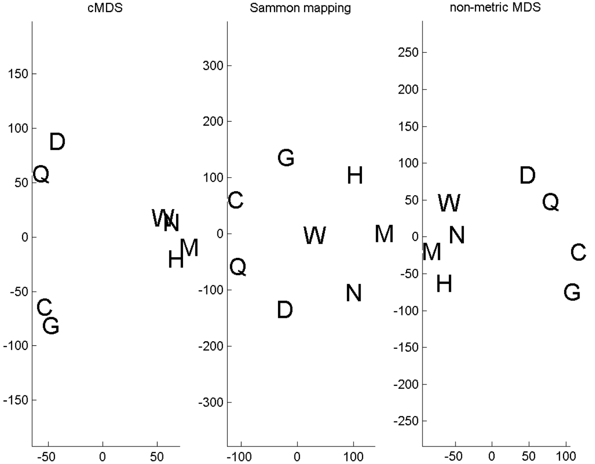
Letter recognition:

- Do you see any clusters?
- With $c = 21 = \max \delta_{ij} + 1$, the eigenvalues of the Gram-matrix $\mathbf{B}$ in the calculation of cMDS are:

  ```
  508.5707
  236.0530
  124.8229
  56.0627
  39.7347
  -0.0000
  -35.5449
  -97.1992
  ```
- The choice of $p = 2$ or $p = 3$ seems reasonable.

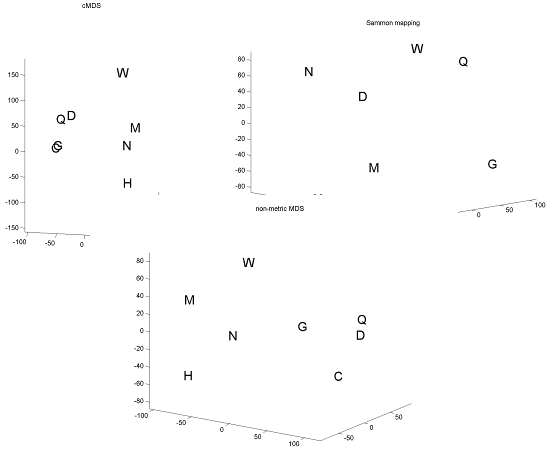
Letter recognition

- Second choice of $c = 210 = \max \delta_{ij} + 190$.
- Compare MDS with $p = 2$, from cMDS, Sammon mapping, and non-metric scaling (stress1):



Letter recognition:

- Second choice of $c = 210 = \max \delta_{ij} + 190$.
- Compare MDS with $p = 3$, from cMDS, Sammon mapping, and non-metric scaling (stress1):



Letter recognition:

- With $c = 210$, the eigenvalues of the Gram-matrix $\mathbf{B}$ in the calculation of cMDS are:

  ```
  1.0e+04 *

  2.7210
  2.2978
  2.1084
  1.9623
  1.9133
  1.7696
  1.6842
  0.0000
  ```

- May need more than $p > 3$ dimensions.

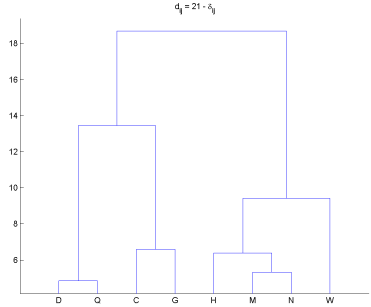
Letter recognition: Summary

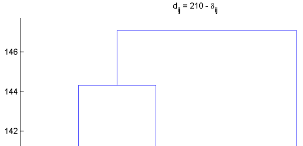
- The structure of the data appropriate for non-metric MDS.
- Kruskal's non-metric scaling:
  1. Appropriate for non-metric dissimilarities (goal is to preserve order)
  2. Optimization: susceptible to local minima (leading to different configurations);
  3. Time-consuming
- cMDS fast, overall good.
- Sammon mapping fails when $c = 210$.

Letter recognition: Summary

- Clusters (C, G), (D, Q), (H, M, N, W) are confirmed by a cluster analysis for either choice of c.

Use agglomerative hierarchical clustering with average linkage:





MDS in R

```
library(MASS)

# compute dissimilarity matrix from a dataset
d <- dist(swiss)
# d is (n x n-1) lower triangle matrix

cmdscale(d, k =2) # classical MDS
sammon(d,k=1) # Sammon Mapping
isoMDS(d,k=2) # Kruskal’s Non-metric MDS
```

MDS Properties

- Data not needed - only dissimilarities.
- Algorithm - gradient descent.
- Choosing q:
  - Scree plot (like PCA).
  - Shepard Diagram - plot proximities against distances in Z.
- Interpreting MDS maps:
  - Axes and orientation arbitrary.
  - Can be rotated.
  - Only relative locations important.
  - Typically looks for objects close in the MDS map.

MDS vs. PCA

- Similarities:
    - Dimension reduction for visualization.
- Differences: MDS is
    - Non-linear
    - Local solution & arbitrary map.
    - Non-unique & local solution.

Dimension Reduction Wrap-Up

Techniques Covered:

- PCA.
- NMF.
- ICA.
- MDS.

Relative strengths and weakness?