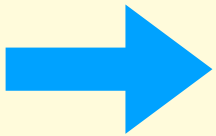
CS-E4840
Information Visualization
Lecture 9: Dimensionality reduction

Tassu Takala <tapio.takala@aalto.fi>
29 March 2021

RECAP FROM LAST LECTURE

Big data: too much for one view?



- Dynamic visualization
  - interactive navigation in information space
  - show only a selection of data at a time

- Algorithmic data mining
  - clustering and aggregation
  - dimensionality reduction

Landmarks (focus) and overview map (context)

https://de.maps-london.com

Exploring information space: navigation + focus&context

- **Focus+context problem**: how to find details from a larger context in information space. Or, how to *navigate efficiently* in abstract spaces.

- There are several visual techniques to help this (providing user overview, position and landmarks):

  - **Elision techniques.** Part of the structure are hidden until they are needed.
  - **Distortion techniques.** Magnify regions of interest, decrease space of irrelevant regions.
  - **Rapid zooming techniques.** User zooms in and out of regions of interest.
  - **Multiple windows.** Some windows show overview and others content.
  - **Micro-macro readings.** A high-resolution static visualisation supports focus+context.

  - Often used in combinations

Effective View Navigation in abstract information space

- **Theoretical view** by Furnas (1997) https://doi.org/10.1145/258549.258800
- The information landscape can be thought as a tree or network G
- Effective View Navigation in G, EVN(G): how to organise information with links so that we have

EVT efficient traversal

- small views: number of outgoing links from a view (maximal *out-degree*, MOD) is small;
- short paths: the expected cost of traversal (number of steps, defined by *network diameter*, DIA) is minimised;

and

VN view navigable

- all targets have a good *residue* ('scent' of target) in each node, and *outlink-info* is small
  - requires good semantic classification of nodes

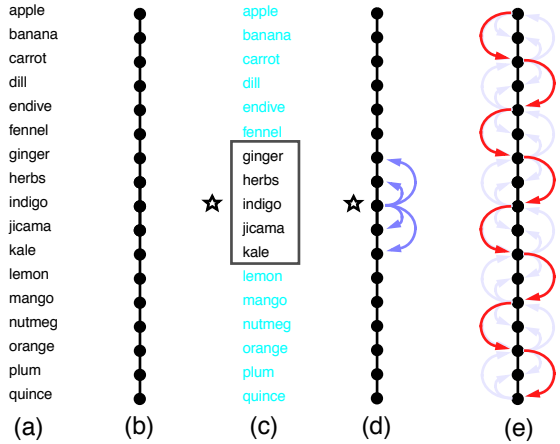


Figure 1. *(a) Schematic of an ordered list, (b) logical graph of the list, (c) local window view of the list, (d) associated part of viewing graph, showing that out degree is constant, (e) sequence of traversal steps showing the diameter of viewing graph is O(n).*

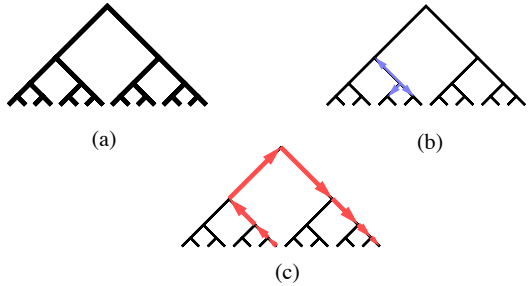


Figure 2. *An example of an Efficiently View Traversable Structure (a) logical graph of a balanced tree, (b) in gray, part of the viewing graph for giving local views of the tree showing the outdegree is constant, (c) a path showing the diameter to be O(log(n)).*

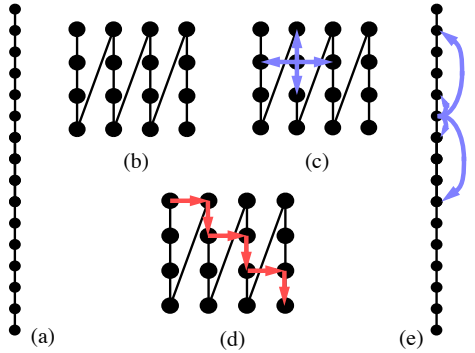
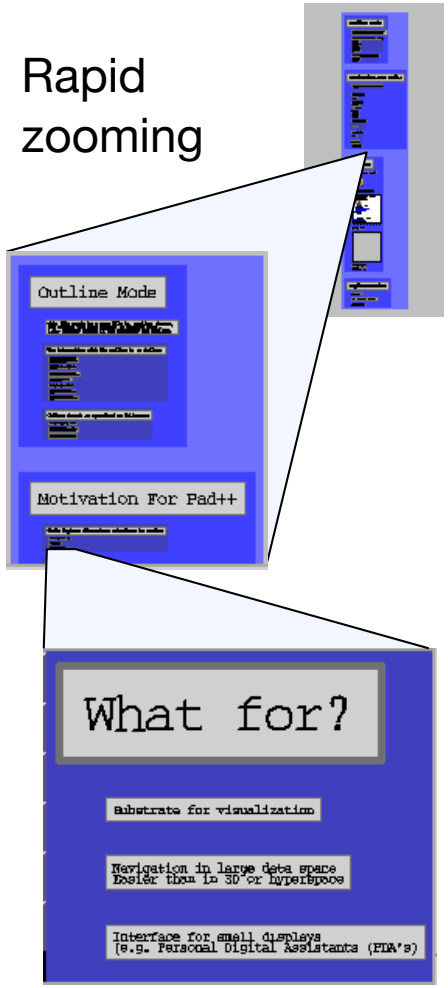


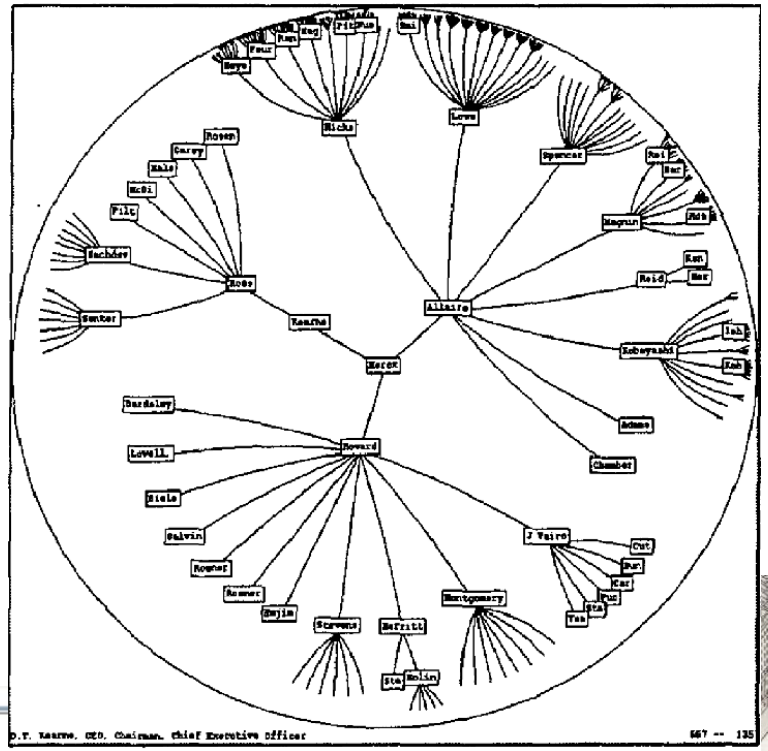
Figure 3. *Fixing the list viewer. (a) logical graph of the ordered list again, (b) the list is folded up in 2-D (c) part of the viewing graph showing the 2-D view-neighbors of Node6 in the list: out degree is O(1), (d) diameter of viewing graph is now reduced to O(sqrt(n)). (e) Unfolding the list, some view-neighbors of Node6 are far away, causing a decrease in diameter.*

Techniques

Rapid zooming

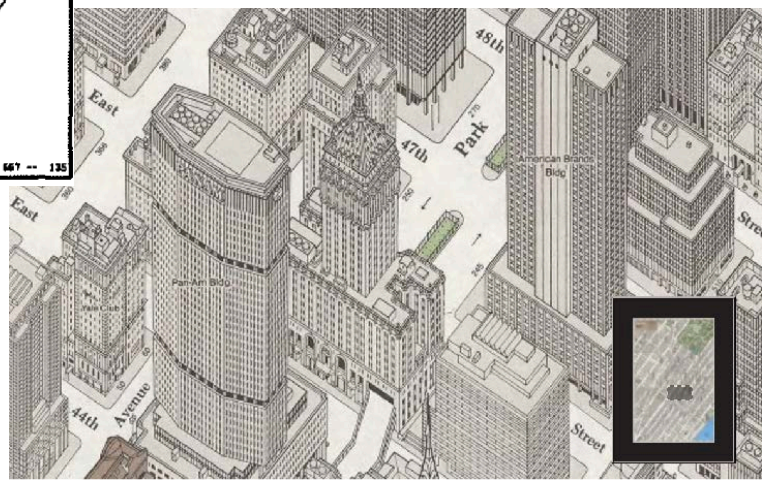


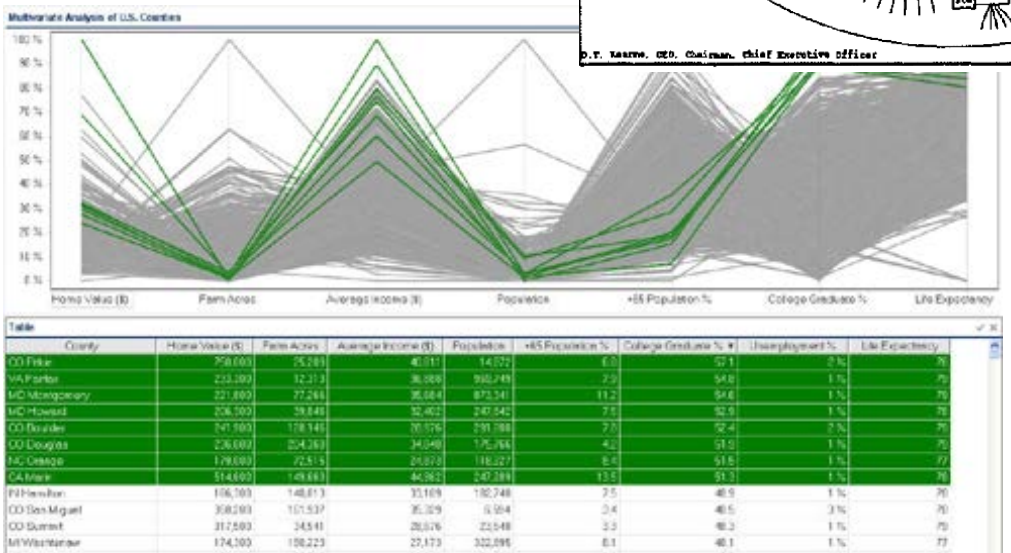
Distortion



Elision

Micro-macro readings





Multiple windows

Summary

- **Focus+context problem**: how to find details from a larger context in information space. Or, how to *navigate efficiently* in abstract spaces.

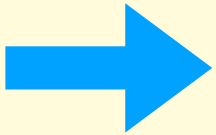
- Several techniques, often in combination:
  - **Elision techniques**
  - **Distortion techniques**
  - **Rapid zooming techniques**
  - **Multiple windows**
  - **Micro-macro readings**

- Furnas' theory of effective view navigation

NEW IN THIS LECTURE

Big data: too much for one view?

- Dynamic visualization
  - interactive navigation in information space
  - show only a selection of data at a time



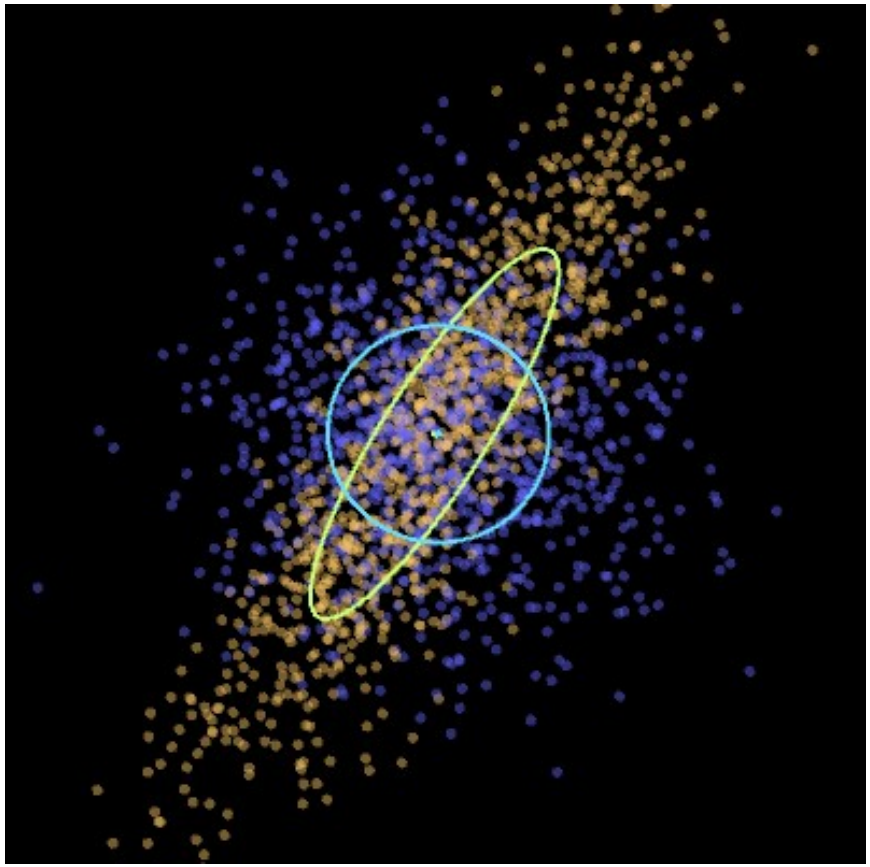
- Algorithmic data mining
  - clustering and aggregation
  - dimensionality reduction

Clustering and aggregation

- General idea: represent a subset of data by a simpler shape (a glyph)
  - elision technique

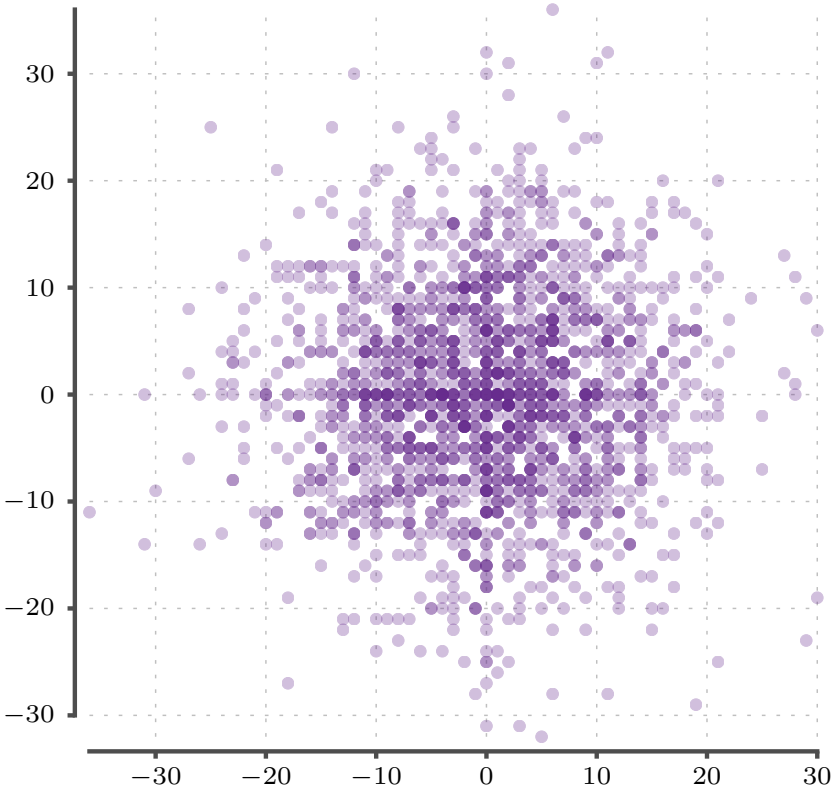
- Example: a normal (= Gaussian) distribution shown by its mean and STD ellipse (analogously with a boxplot)

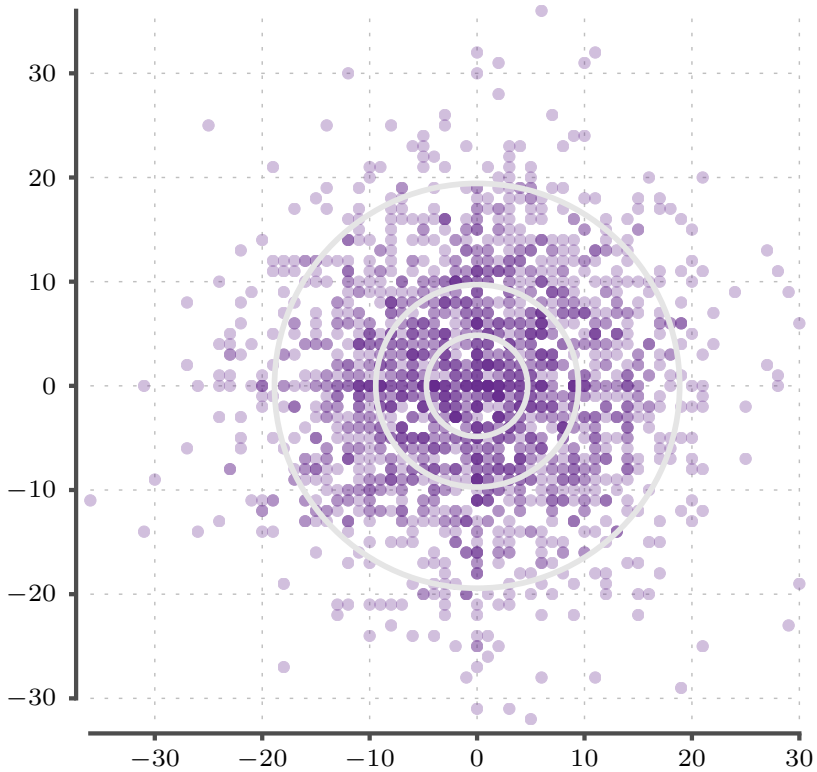
- Level-Of-Detail (LOD): show a shape in simplified form, depending on the scale of presention
  - e.g. cities on a map



Reminder (lecture 4): Dealing with overplotting

- Tricks to reduce clutter:
  - make your markers **transparent**
  - consider **contour lines** or **heat map**
  - **downsample** data (it probably makes no sense to show million points anyway)





Dimensionality reduction

- Assume your **data points** are m-dimensional, i.e. each item is defined by *m* measurable properties
- Assume dimensionality *m* is so large that a data point cannot be visualised by "traditional" methods

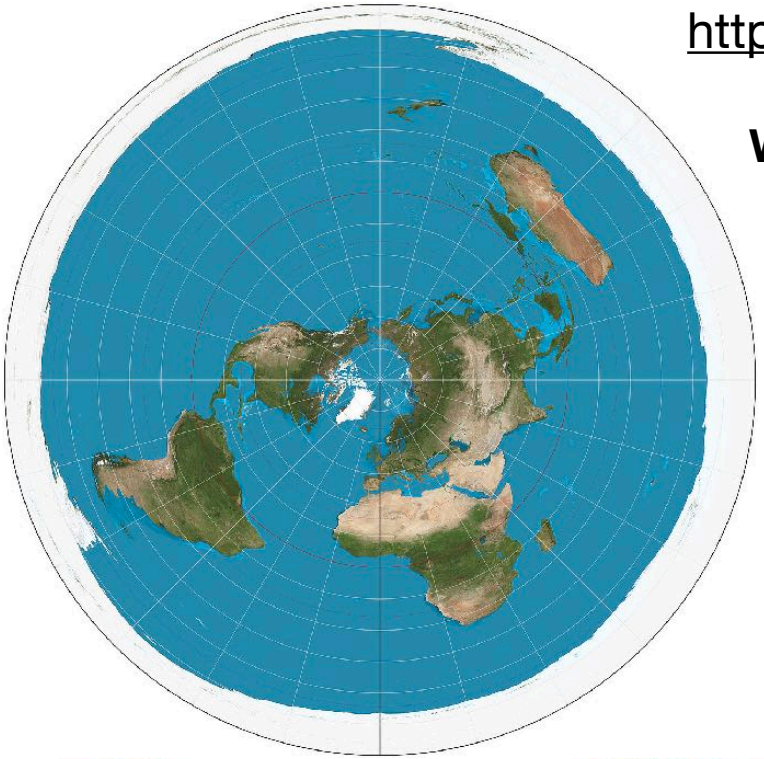
- *Problem statement:* Given a dimensionality k (typically k=2 or k=3), find an embedding X of data points into k-D space (=locations of data points) such that some properties in the embedding match the original as well as possible.

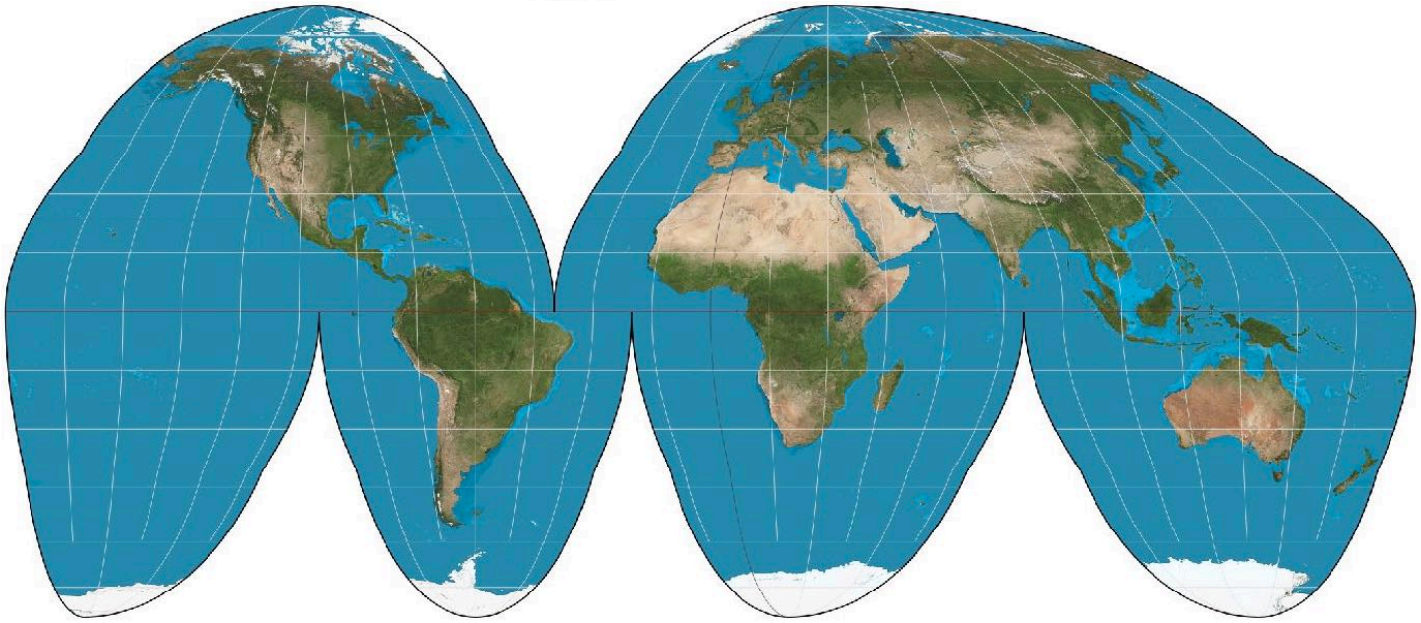
- The property to retain can be e.g. distances $d_{ij}(X)$ between corresponding points, directional angles, local shapes, etc.

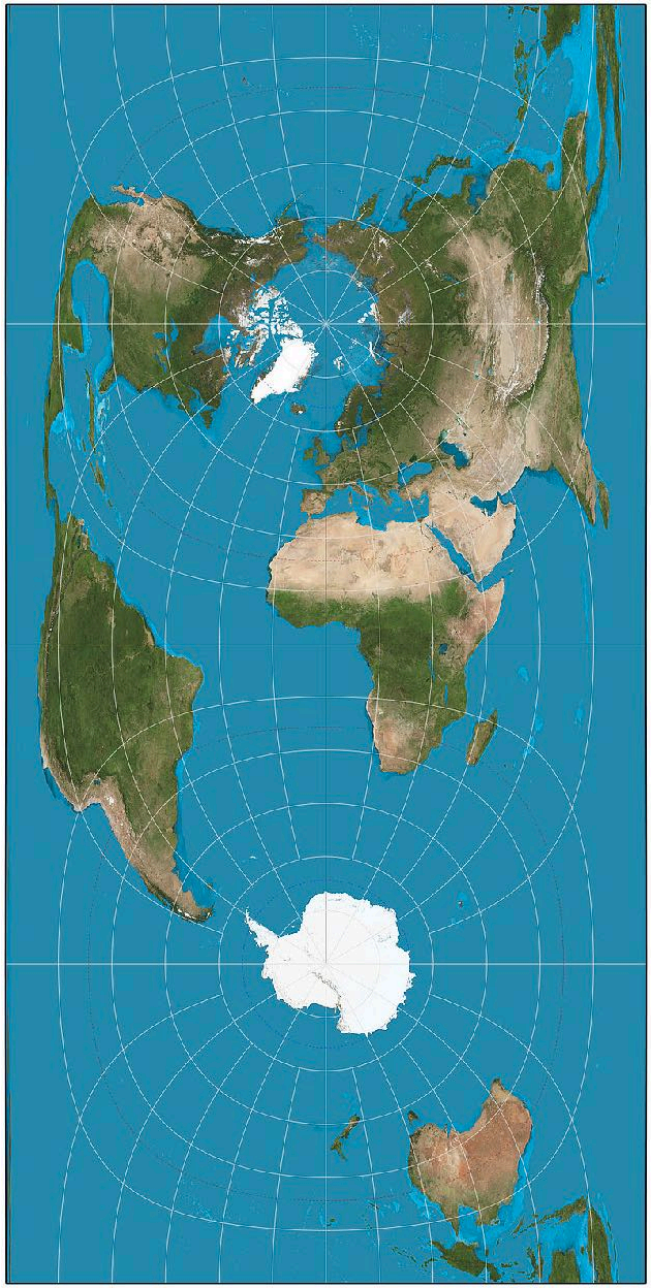
World map: what to optimize in 3D→2D embedding?

https://en.wikipedia.org/wiki/List_of_map_projections

Which properties are important to retain in projection?

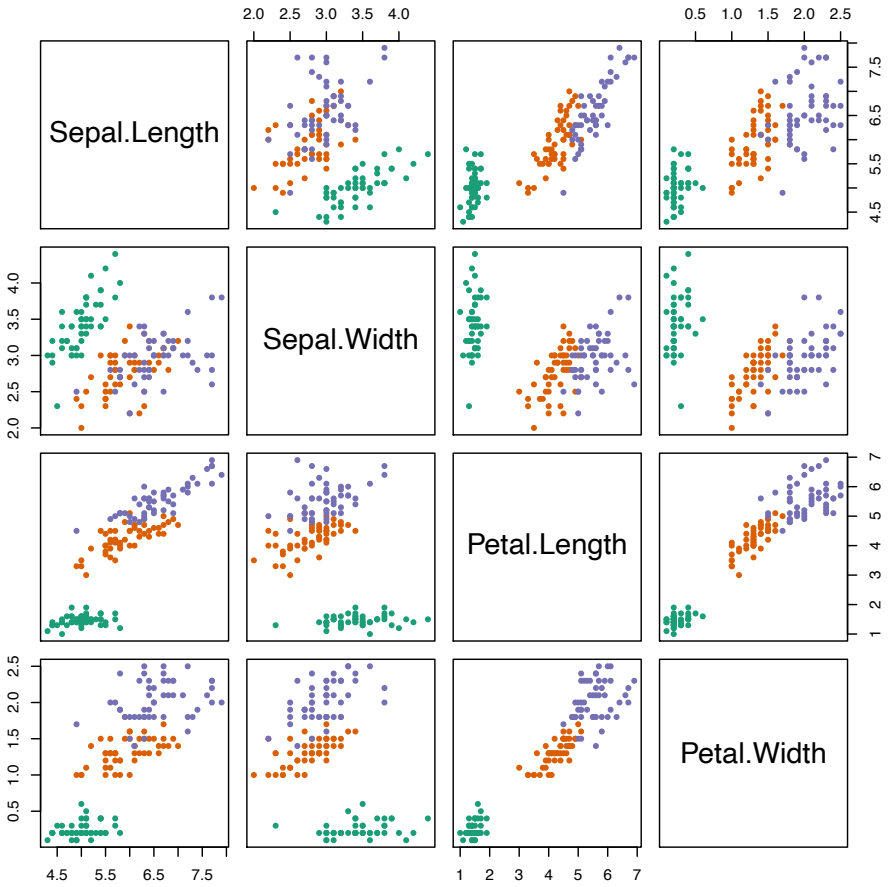






Simple projections

- Assume the data is composed of **vectors** in m-D space.
- Simple **orthogonal** projection takes 2 (or 3) of the original dimensions and neglects others.
- Example:

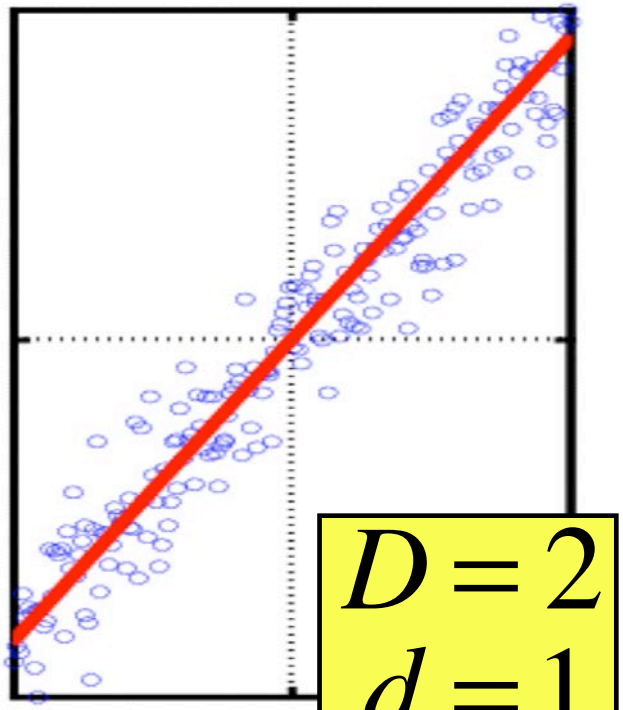


What if none of these reveals the true shape of the data set?

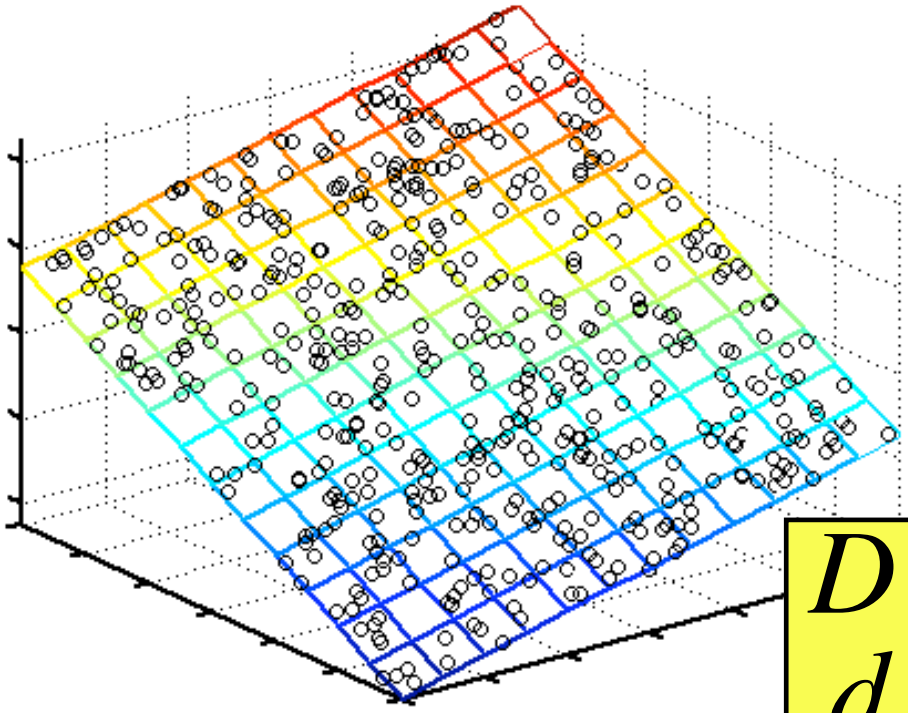
Projection pursuit methods

- Try to **find a linear subspace** u that maximises some quantity f(X) of the data X
  - f is variance: principal component analysis (PCA)
  - f non-gaussianity: independent component analysis (ICA)
- Typically, we can find several directions (base vectors) of u , possibly with orthogonality conditions.

Principal component analysis (PCA)



$D = 2$
$d = 1$



$D = 3$
$d = 2$

- Basic idea: **rotate** the space such that the data becomes maximally aligned with the coordinate axes.
  Then take an orthogonal projection.

Principal component analysis (PCA)

- The **principal component analysis** (PCA) finds the eigenvalues and -vectors of a matrix
- PCA is an example of the projection pursuit methods. It tries to find a linear subspace that has **maximal variance**.
- Thus, the interesting quality in PCA is variance (distance).
  - you could think PCA as a linearised version of MDS (actually PCA is equivalent to one modification of MDS).
- PCA (unlike MDS) assumes that the data points are vectors in a high-dimensional Euclidean space,
- The data points are projected to d-dimensional Euclidean subspace (d≪D) of the original space.
- The projection to d-dimensional subspace is linear, $A = \sum_{\alpha=1}^{d} e_\alpha e_\alpha^T$

  $y_i = Ax_i$, where e_α are orthogonal unit vectors.
- Goal: nearby points remain nearby, distant points remain distant.

Principal component analysis (PCA)

- Goal, more formally: *find such a projection (matrix A) to d-dimensional subspace that the average error in the squared Euclidean distances between data points is minimised.*

$$\sum_{i,j=1}^{N} \left| \| x_i - x_j \|^2 - \| y_i - y_j \|^2 \right|$$

  where || || is the Euclidean distance and $y_i = A\, x_i$.

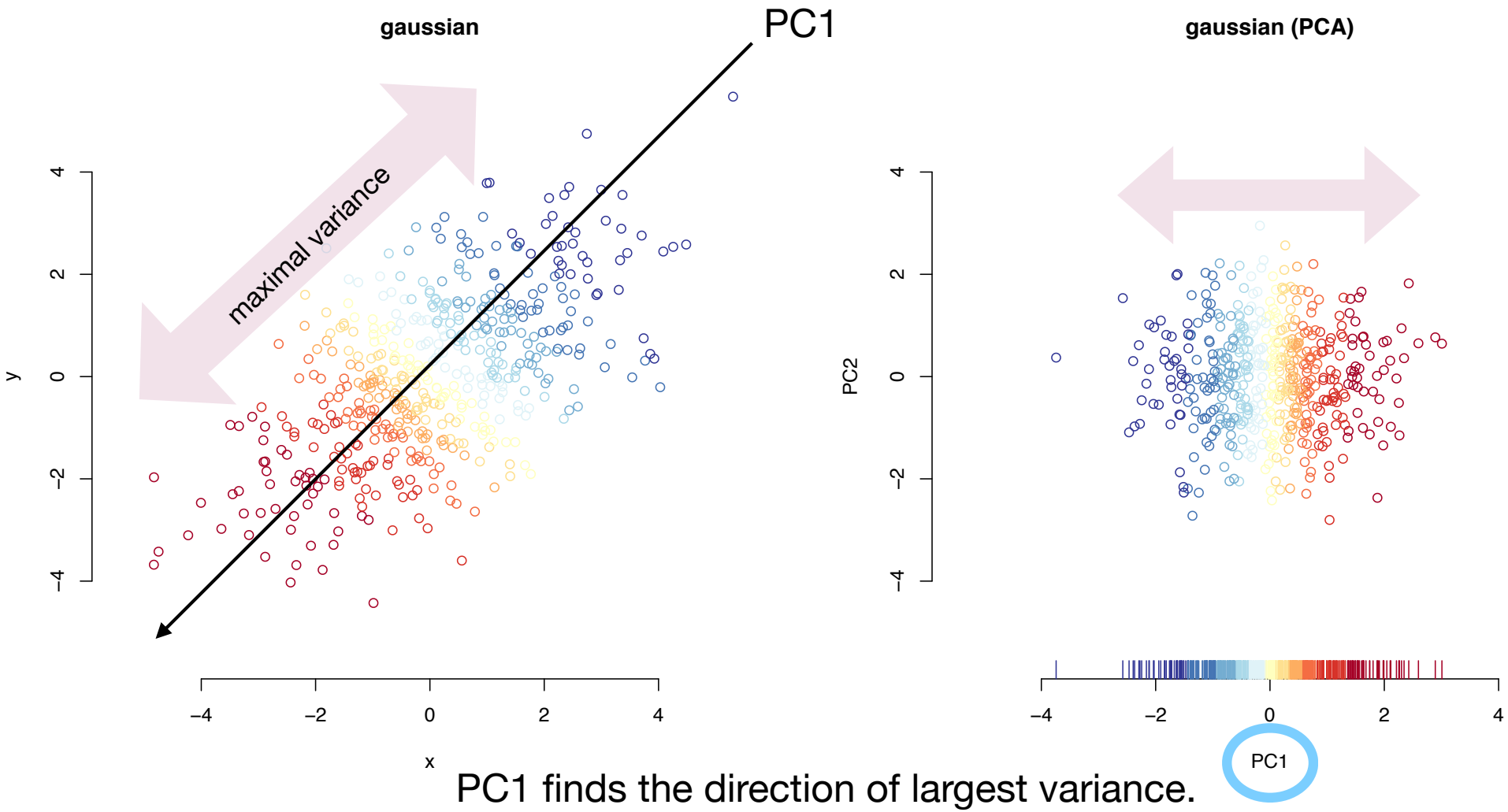
- Denote the mean vector by, $\overline{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$
- The *covariance matrix* reads then, $C = \frac{1}{N} \sum_{i=1}^{N} (x_i - \overline{x})(x_i - \overline{x})^T.$
- The covariance matrix can be decomposed (*spectral decomposition*) as

$$C = \sum_{\alpha=1}^{D} \lambda_a e_\alpha e_\alpha^T$$

  where λ_α are the *eigenvalues* ($\lambda_1 \geq \lambda_2 \geq ... \geq 0$) and e_α are the corresponding orthogonal unit *eigenvectors*.

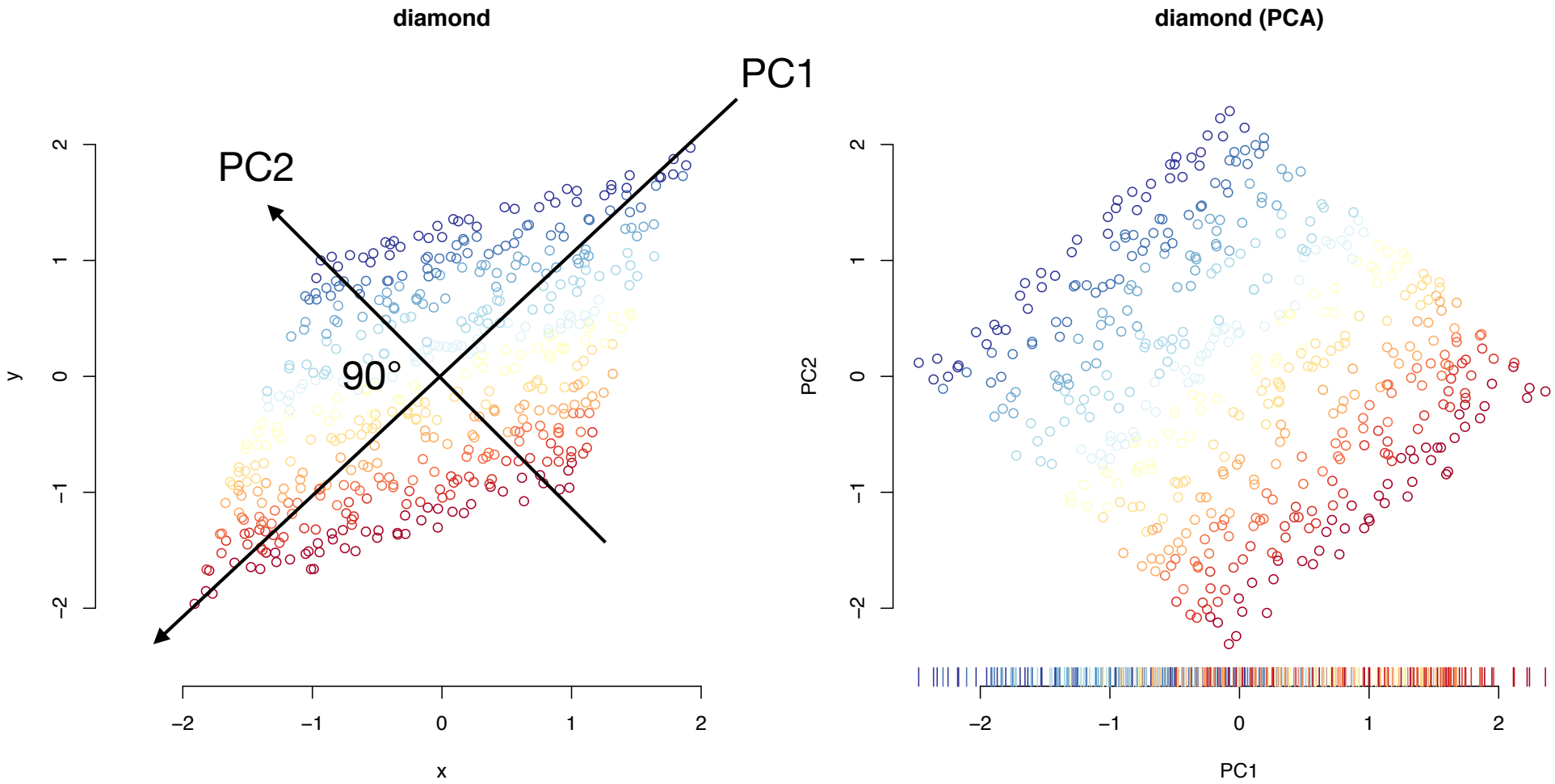
- The maximum variance projection is then given by $A = \sum_{\alpha=1}^{d} e_\alpha e_\alpha^T$

Gaussian data



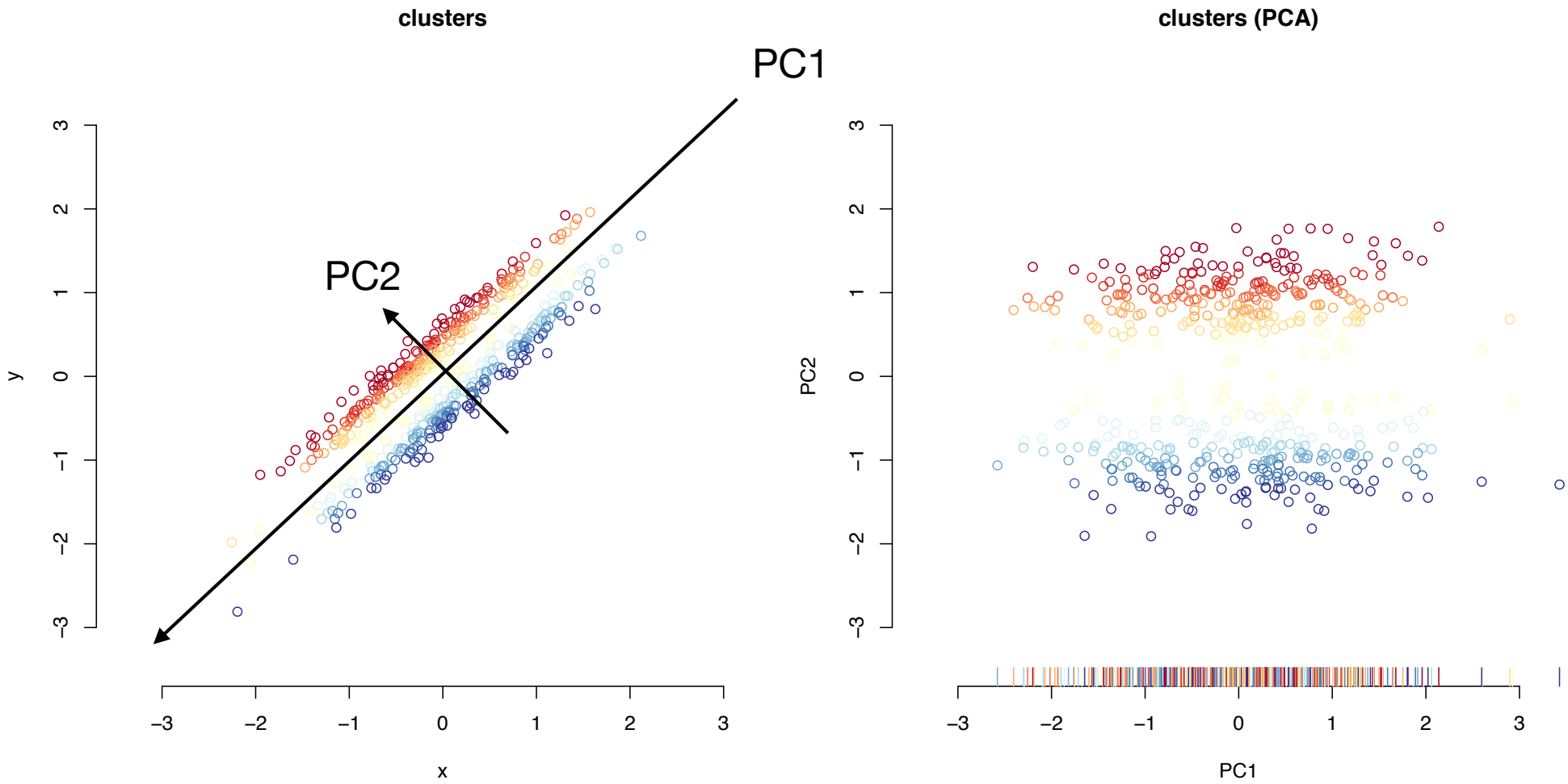
PC1 finds the direction of largest variance.
(PC2 the next largest, etc.)

Diamond shaped data



PC1 misses the square structure.

Two clusters



PC1 misses the cluster structure.

Principal component analysis (PCA)

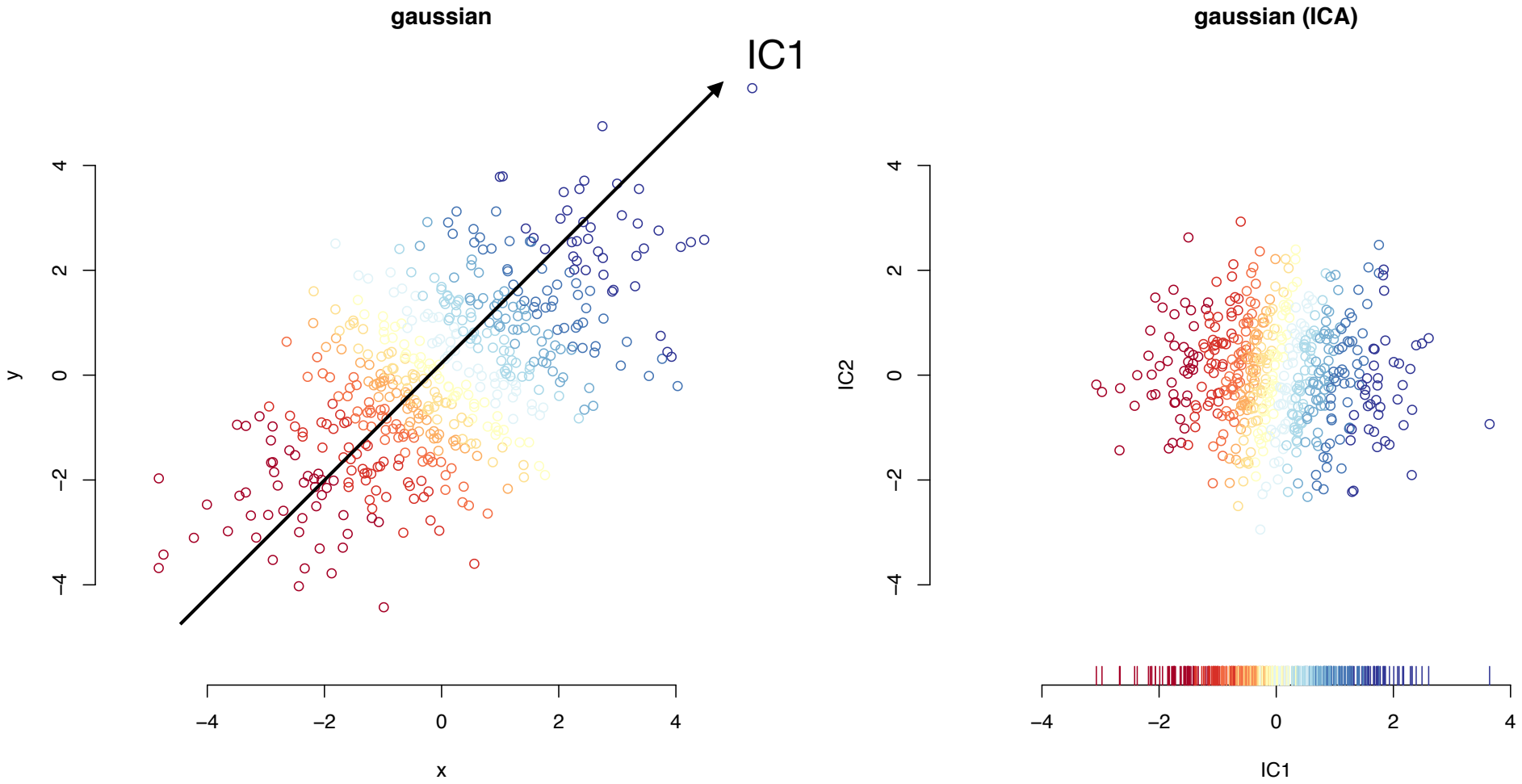
- PCA can be computed easily with (almost) any software that is capable of doing linear algebra.
- PCA is stable, there are no additional parameters, and it is guaranteed always to converge to the same optima.
- Hence, PCA is usually the first dimension reduction method to try (if it doesn't work, then try something more fancy)

- If you find PCA difficult, this may be helpful 😀 https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

Independent component analysis (ICA)

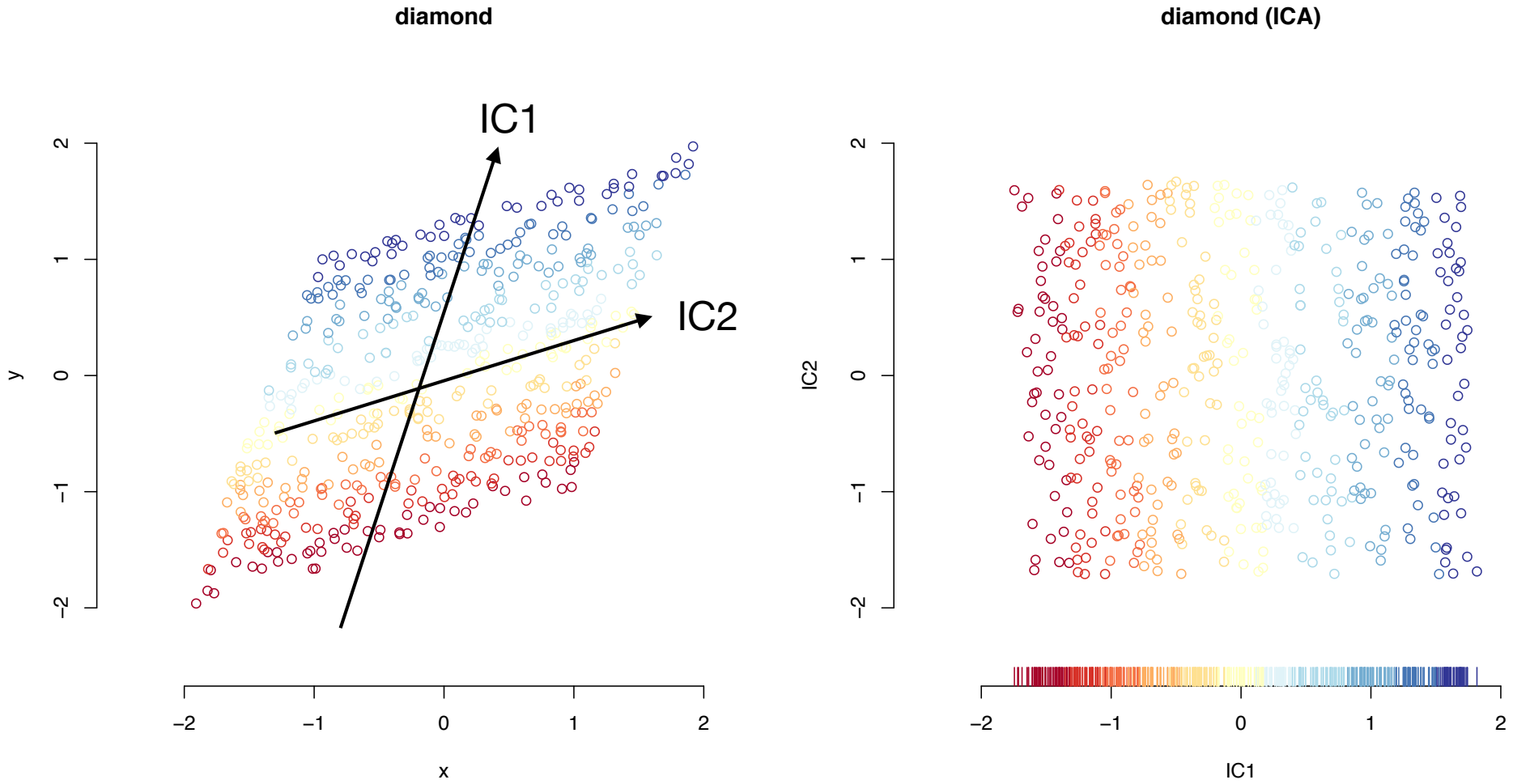
- Goal: function *f* is a measure of non-Gaussianity. Non-Gaussian directions are usually most independent.
- Hence, ICA finds separate processes. Directions are not necessarily orthogonal.
- ICA is unstable and may end up to a local minimum.
- There are robust libraries to compute ICA: use the libraries!

Gaussian data



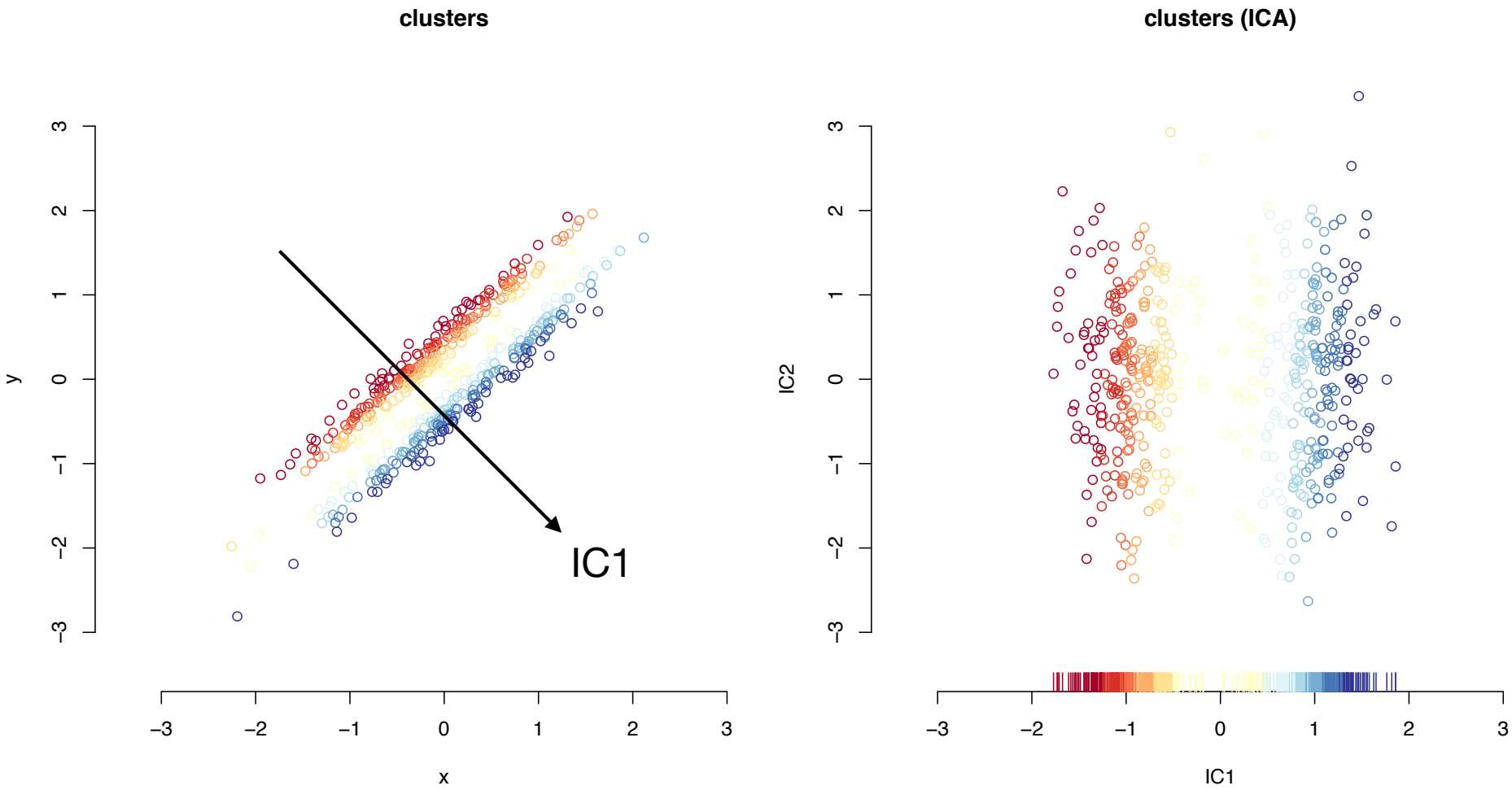
ICA ignores total variance (but finds the maximal variance direction here by co-incidence).

Diamond shaped data



IC1 finds the box in the diamond.

Two clusters



IC1 finds the two clusters.

Non-linear methods

- *Problem statement:* Given a dimensionality k (typically k=2 or k=3), find an embedding X of data points into k-dimensional space (=locations of data points) such that some properties in the embedding match the original as well as possible.

- Assume that we can define *proximity* p_{ij} (= *meaningful* **distance**) between data points i and j
  - note: for N points there are N(N-1)/2 pairwise distances

- Instead of linear projection, try to find a mapping, such that distances $d_{ij}(X)$ between corresponding points match.

  - only mutual distances (or similarities) matter, the original points may or may not be located in any dimensional vector space

Non-linear methods

- *Problem statement:* Find an embedding such that distances in the embedding match those between corresponding original points as well as possible.

- What does "as well as possible" mean?
  - Long distances?
  - Short distances?
  - Neighbourhood relations?

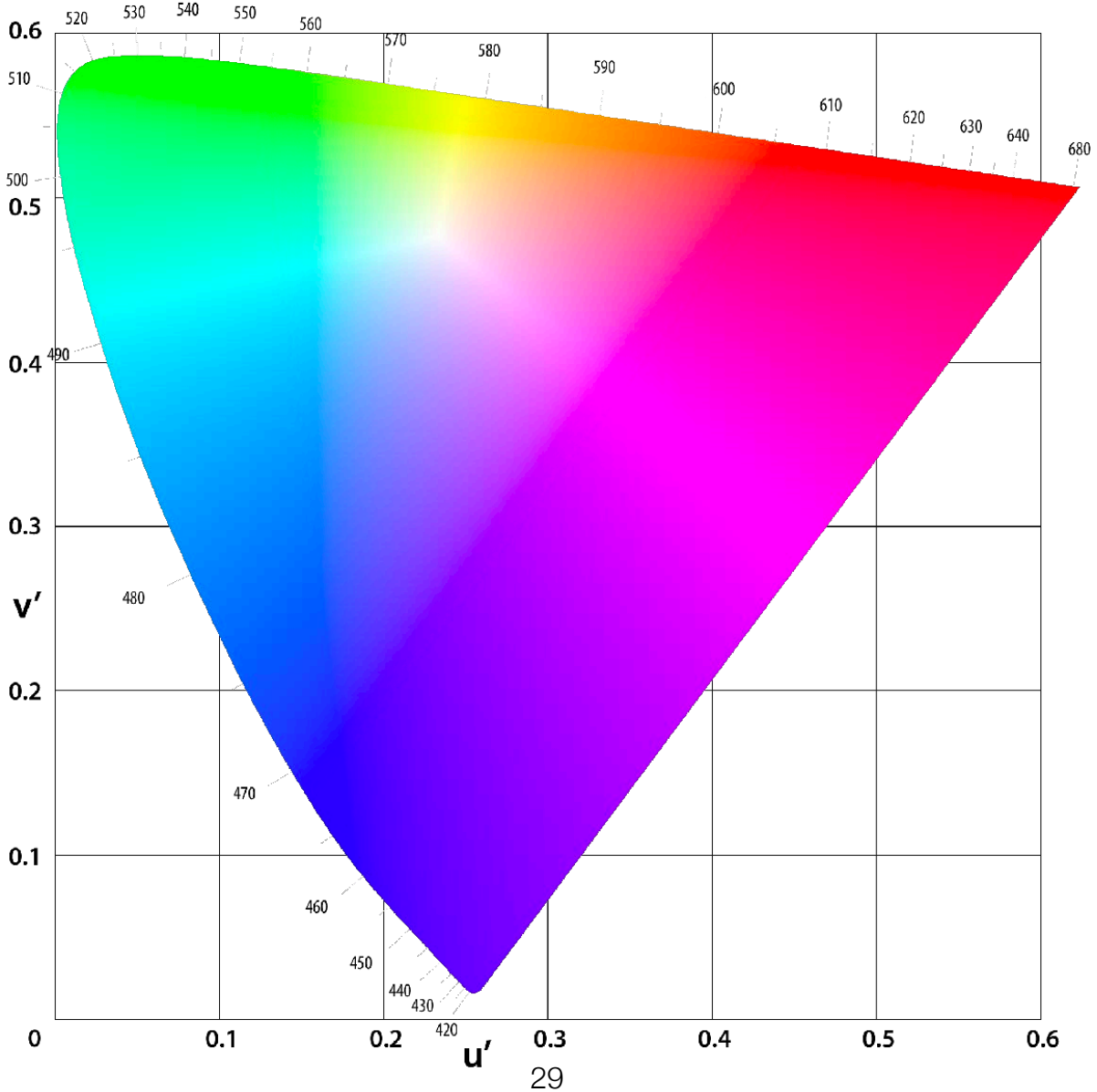
  - maintain correct (visual perception of) relations between nodes

- All embeddings have to make compromises. We will now study embeddings that **preserve long distances.**

Non-linear methods

- Methods that try to preserve **long distances** as well as possible:
  - Metric multidimensional scaling (MDS)
  - Nonmetric MDS
  - Sammon mapping
    - (also PCA and ICA have this property)
  - The alternative is to preserve short distances (=neighborhoods), leading to manifold embeddings (later)
- See examples on these methods analyzed with R: http://www.iki.fi/kaip/p/dimensionality_reduction_1.nb.html

Example: colours



CIELUV

Example: colours

- How is the similarity of colors perceived?
- Pairs of 14 colors were rated by 31 people. Ratings were averaged (Ekman 1954, https://doi.org/10.1080/00223980.1954.9712953).

nm	434	445	465	472	490	504	537	555	584	600	610	628	651	674
434	–	.14	.17	.38	.22	-.73	-1.07	-1.21	-.62	-.06	.42	.38	.28	.26
445	.86	–	.25	.11	-.05	-.75	-1.09	-.68	-.35	-.04	.44	.65	.55	.53
465	.42	.50	–	.08	-.32	-.57	-.47	-.06	.00	-.32	.17	.12	.91	.82
472	.42	.44	.81	–	.12	-.36	-.26	.15	.00	-.11	.00	.33	.23	1.03
490	.18	.22	.47	.54	–	-.07	.08	.48	.40	.00	.22	.17	.07	.00
504	.06	.09	.17	.25	.61	–	.31	.28	.45	.68	.01	.00	.00	-.15
537	.07	.07	.10	.10	.31	.62	–	.13	.35	.09	.31	.00	.00	-.75
555	.04	.07	.08	.09	.26	.45	.73	–	-.05	.17	-.09	-.22	-.32	-.34
584	.02	.02	.02	.02	.07	.14	.22	.33	–	-.05	-.01	-.06	-.16	-.18
600	.07	.04	.01	.01	.02	.08	.14	.19	.58	–	.21	.07	-.39	-.40
610	.09	.07	.02	.00	.02	.02	.05	.04	.37	.74	–	-.08	-.13	-.11
628	.12	.11	.01	.01	.01	.02	.02	.03	.27	.50	.76	–	-.03	-.16
651	.13	.13	.05	.02	.02	.02	.02	.02	.20	.41	.62	.85	–	-.11
674	.16	.14	.03	.04	.00	.01	.00	.02	.23	.28	.55	.68	.76	–

Similarities of colors with different wavelengths (lower half, Ekman 1954) and residuals of 1D MDS representation (upper half) [B 4.1].

Multidimensional scaling (MDS)

- Formally, an MDS algorithm is given as input *) the original distances p_{ij} (called ***proximities***) between data points i and j

- MDS algorithm then tries to find a k-dimensional (usually k=2 or k=3) representation X for the points that minimises the error function (called ***stress***, by convention)

$$\sigma_r = \sum_{i<j} \left(f(p_{ij}) - d_{ij}(X)\right)^2$$

- …where $d_{ij}(X)$ is the Euclidean distance between the data points *i* and *j* in representation X and *f* is a function that defines the MDS model (next slide).

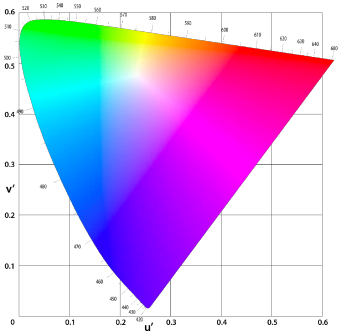
*) NOTE: only pairwise distances given as input, no coordinate system (the spectral wavelength is not utilized)

Multidimensional scaling (MDS)

$$\sigma_r = \sum_{i<j} \left(f(p_{ij}) - d_{ij}(X) \right)^2$$

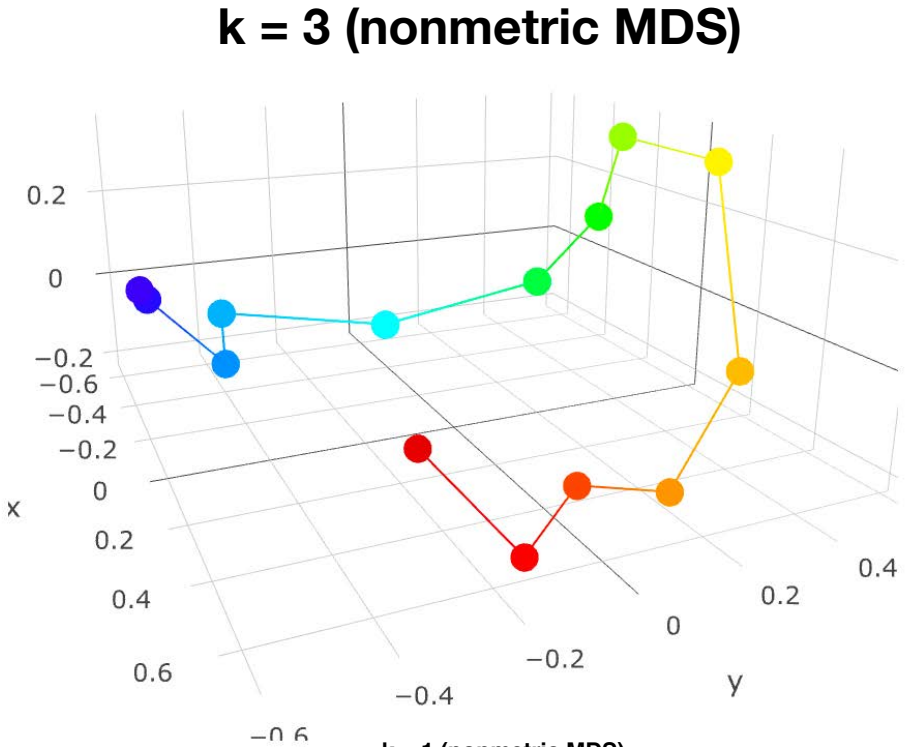
- The choice of *f* defines the MDS model. For example:
  - $f(p_{ij})=p_{ij}$ - absolute MDS (linear model)
  - $f(p_{ij})=b\ p_{ij}$ - ratio MDS (linear model)
  - $f(p_{ij})=a+b\ p_{ij}$ - interval MDS (linear model)
  - $f(p_{ij})=a+b \log p_{ij}$ - useful in psychology (logarithmic)
  - $f(p_{ij})$ can be any monotonically increasing function (ordinal or **nonmetric MDS**)
    - this would be the most important special case of MDS

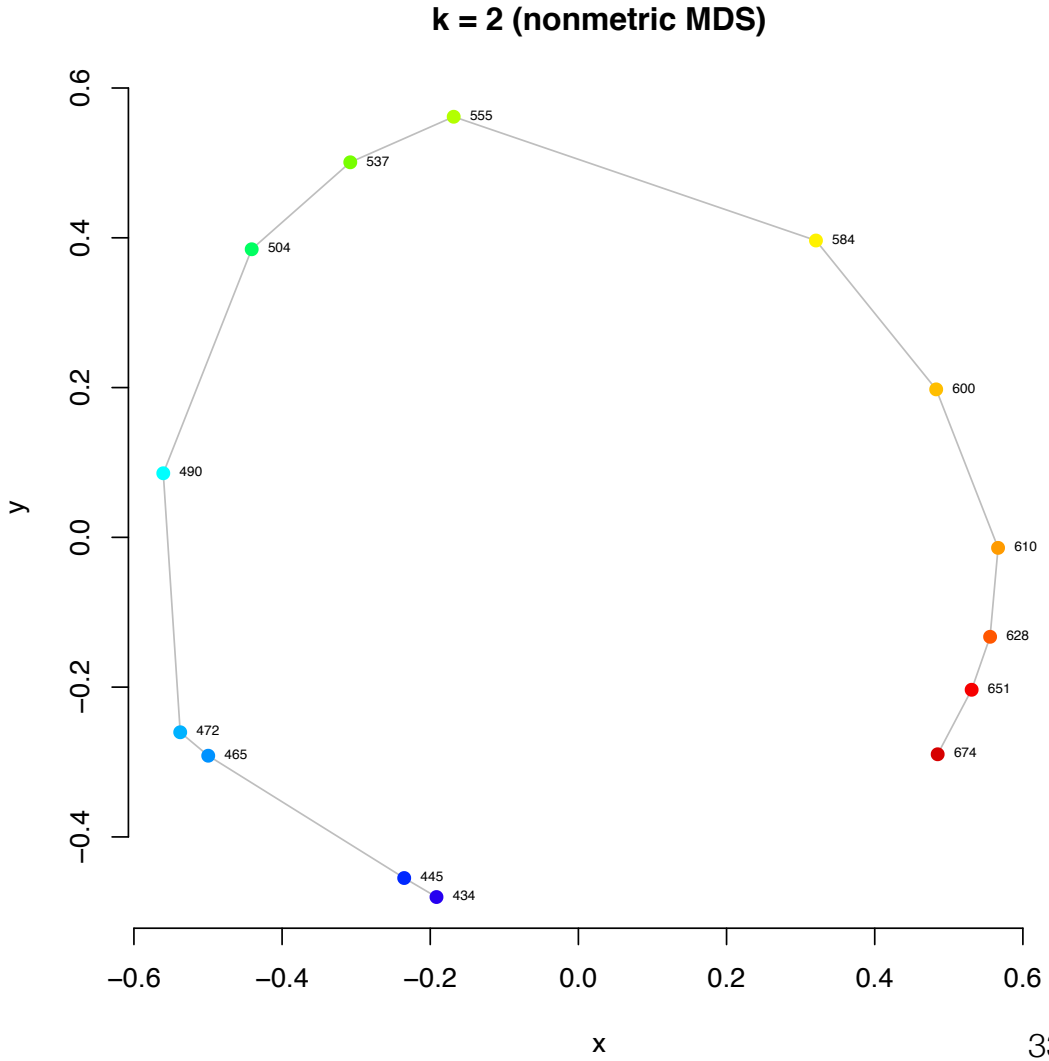
- The parameters of f (such as a and b above) are optimised at the same time as the representation X (i.e., the locations of the projected points)
  - details of the optimisation algorithms is outside the scope of this course

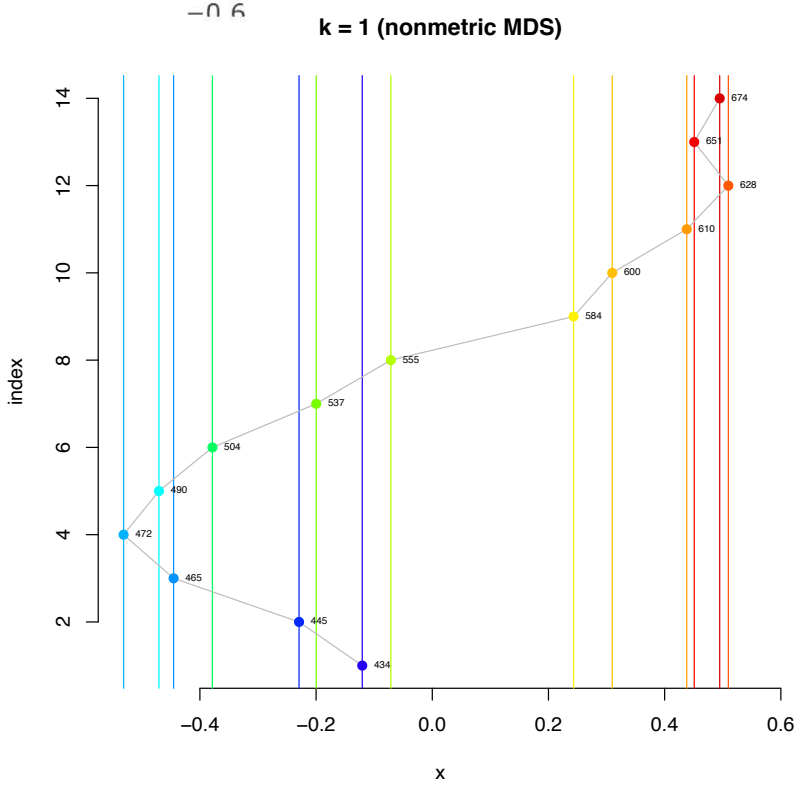


CIELUV

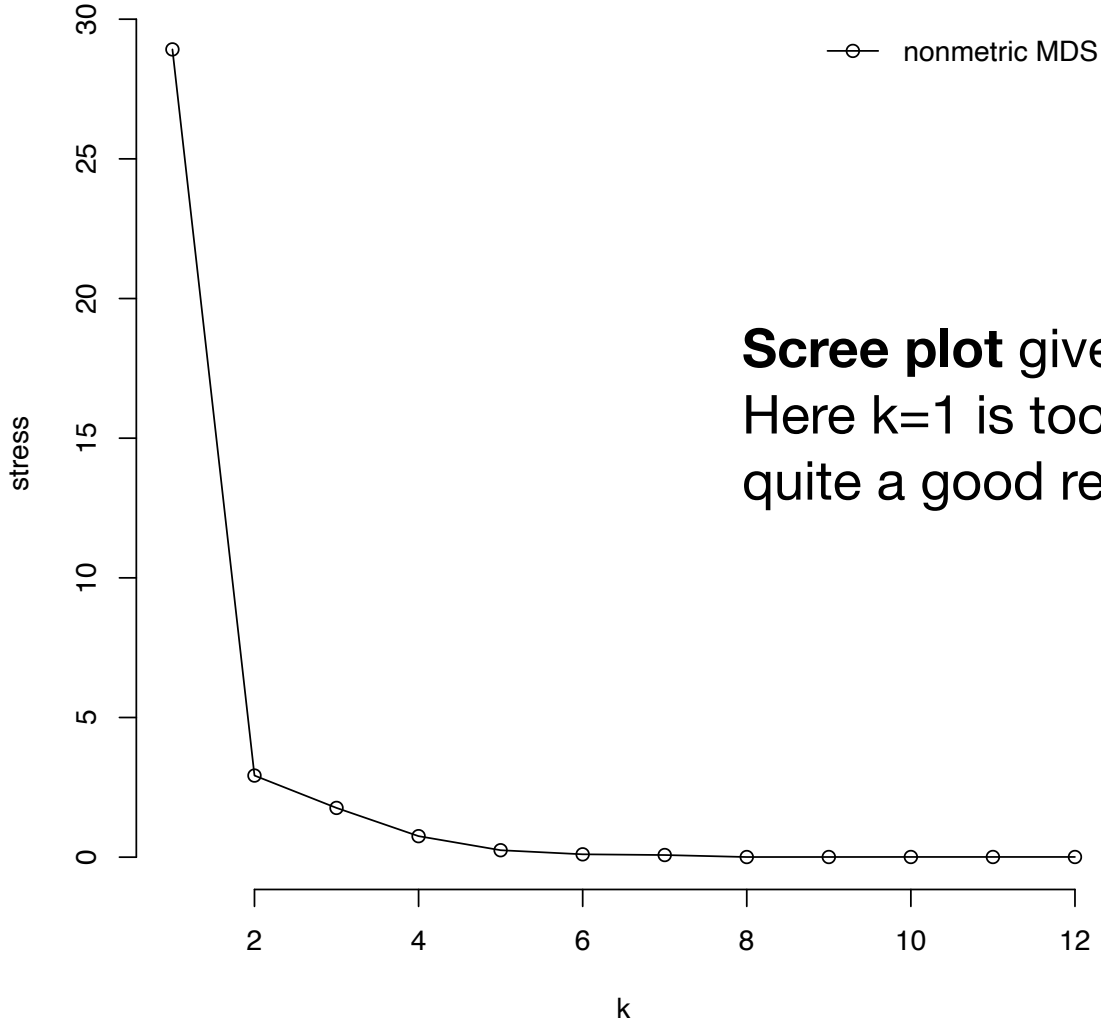
Example: colour







Evaluating the mapping
example: colour



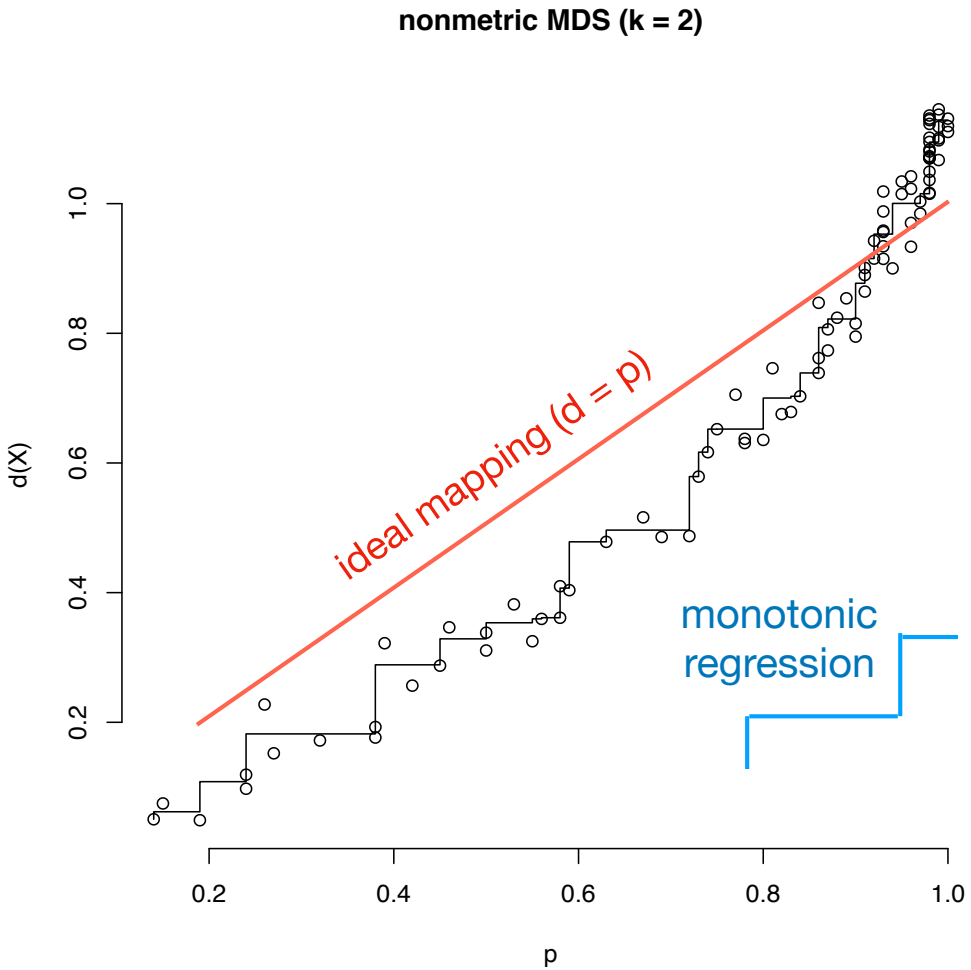
Scree plot gives the stress as a function of k. Here k=1 is too small but k=2 already gives quite a good result.

Evaluating the mapping
example: colour

Shepard plot gives the distances in the embedding as a function of the proximities in the original space.

Shepard diagram: "A plot of two measurements of the distances between objects. One measurement is the true distance, and the other measurement is the apparent distance in some representation of the objects. For example, the apparent distance between objects in a photograph (two dimensions) and the real three-dimensional distance. The diagram is used in multidimensional scaling to assess the extent of any distortion. Zero distortion would correspond to a set of collinear points."

oxfordreference.com



Classical MDS and Sammon mapping

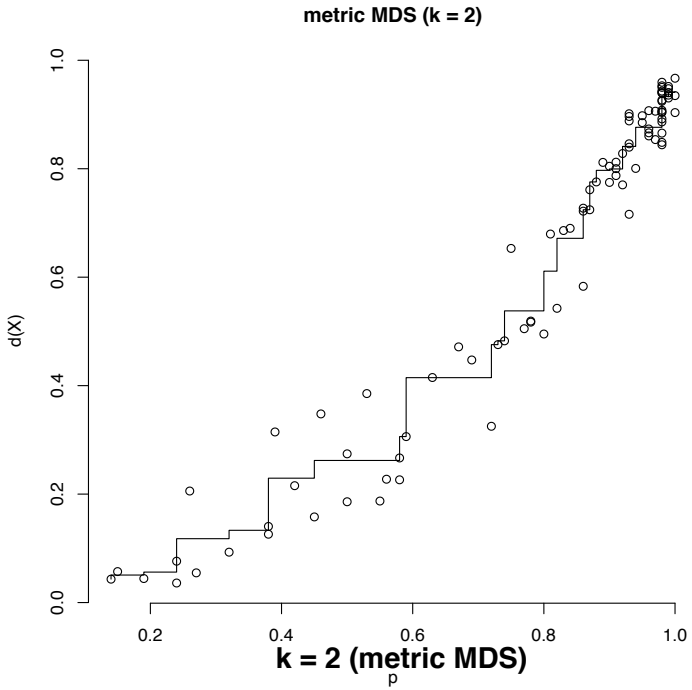
- **Sammon mapping**: given a distance p_{ij} find a representation X that minimises

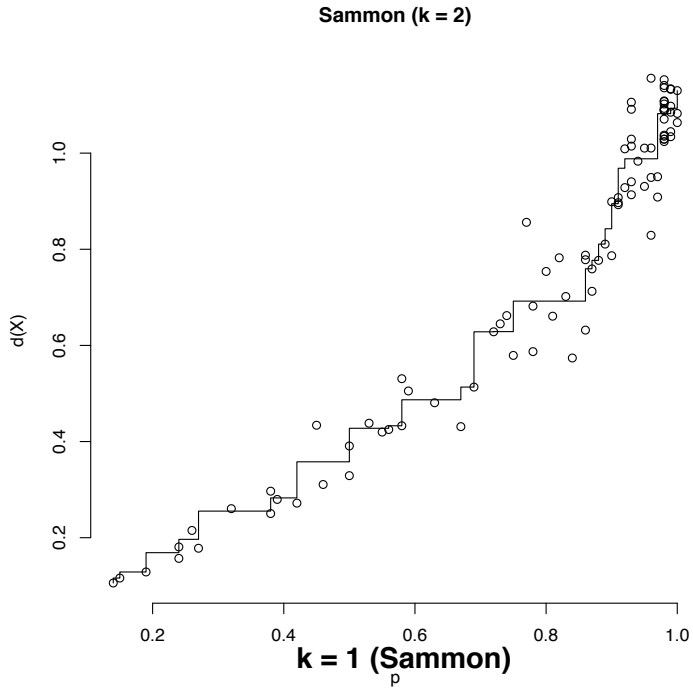
$$E = \sum_{i<j} \frac{\left(d_{ij}(X) - p_{ij}\right)^2}{p_{ij}}$$

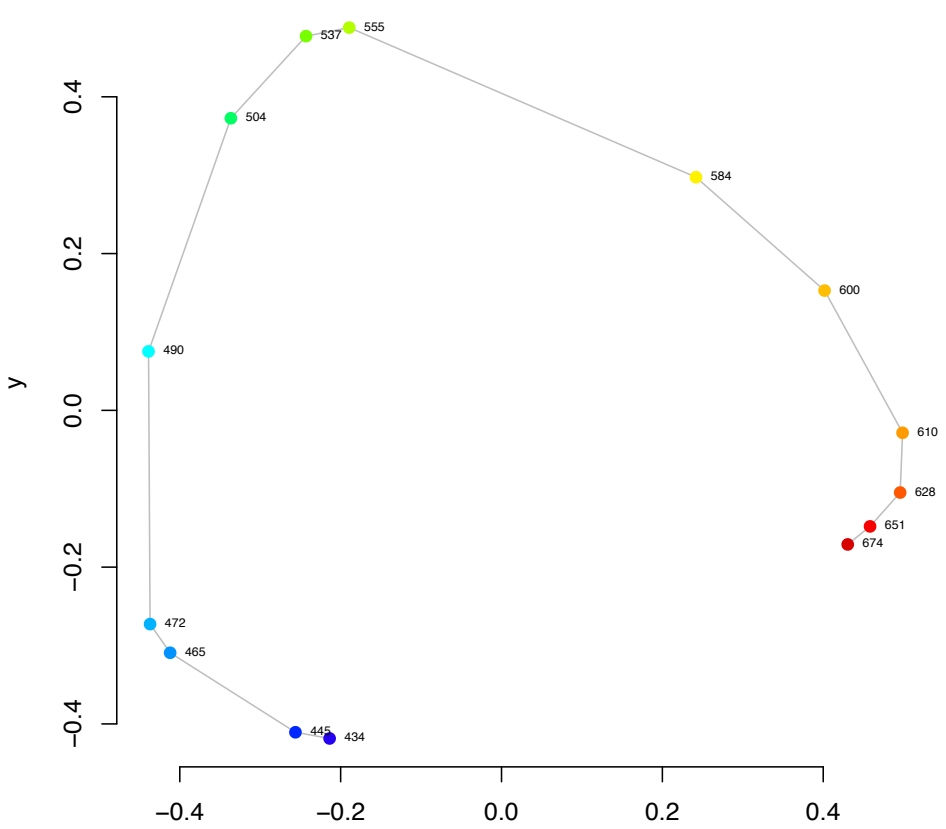
classical MDS: the same without this

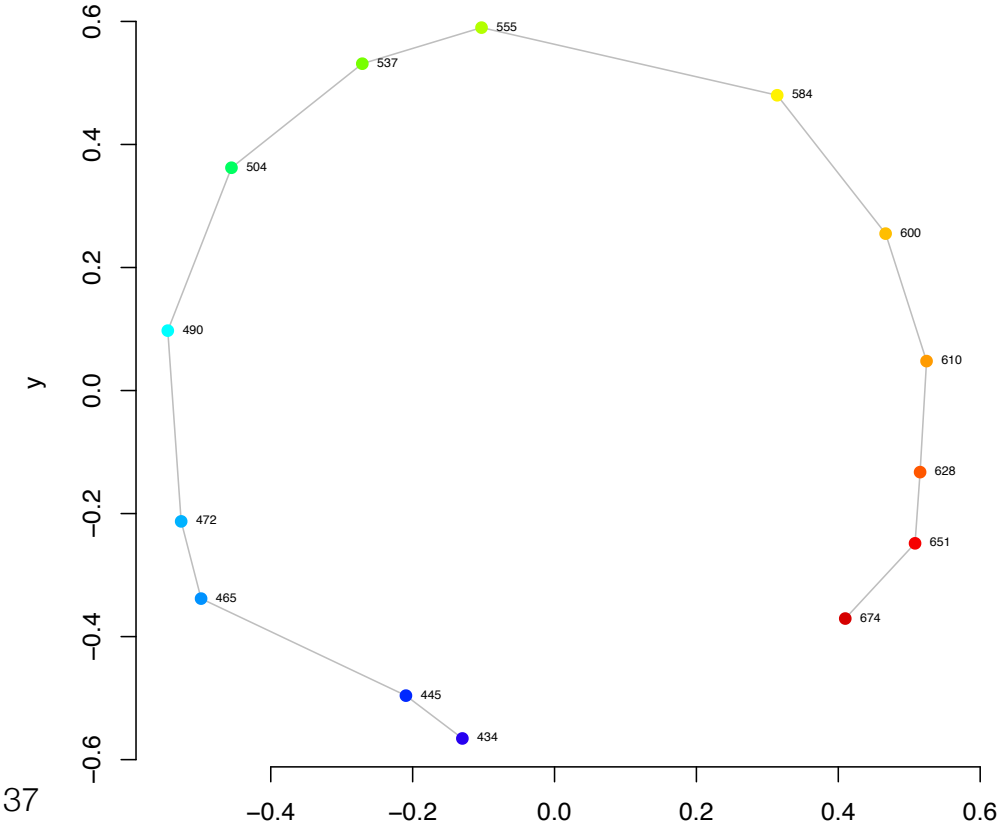
  - As compared to MDS Sammon mapping should be more accurate for shorter distances but less accurate for longer (why?)
  - Like in nonmetric MDS, solution is found by gradient descent, which may end up in a local minimum

- **Classical MDS** is an instance of metric MDS
  - a.k.a. Principal Coordinates Analysis (PCoA), Torgerson Scaling, or Torgerson–Gower scaling.


metric MDS (k = 2)
d(X)
p
0.0
0.2
0.4
0.6
0.8
1.0


Sammon (k = 2)
d(X)
p
0.2
0.4
0.6
0.8
1.0


k = 2 (metric MDS)
y
−0.4
−0.2
0.0
0.2
0.4
555
537
504
584
600
610
628
651
674
490
472
465
445
434

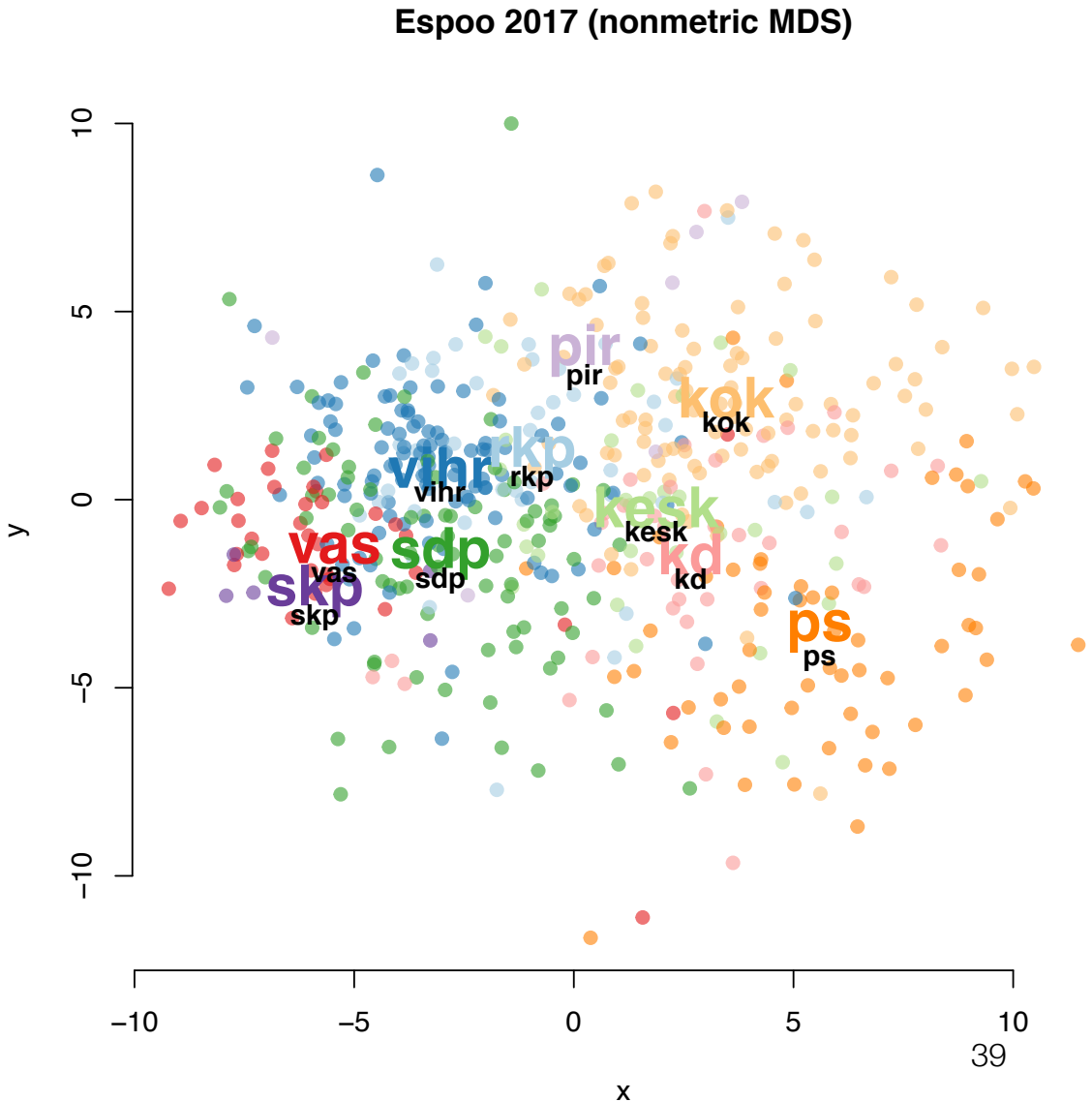

k = 1 (Sammon)
y
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
555
537
584
504
600
610
490
628
472
651
465
674
445
434

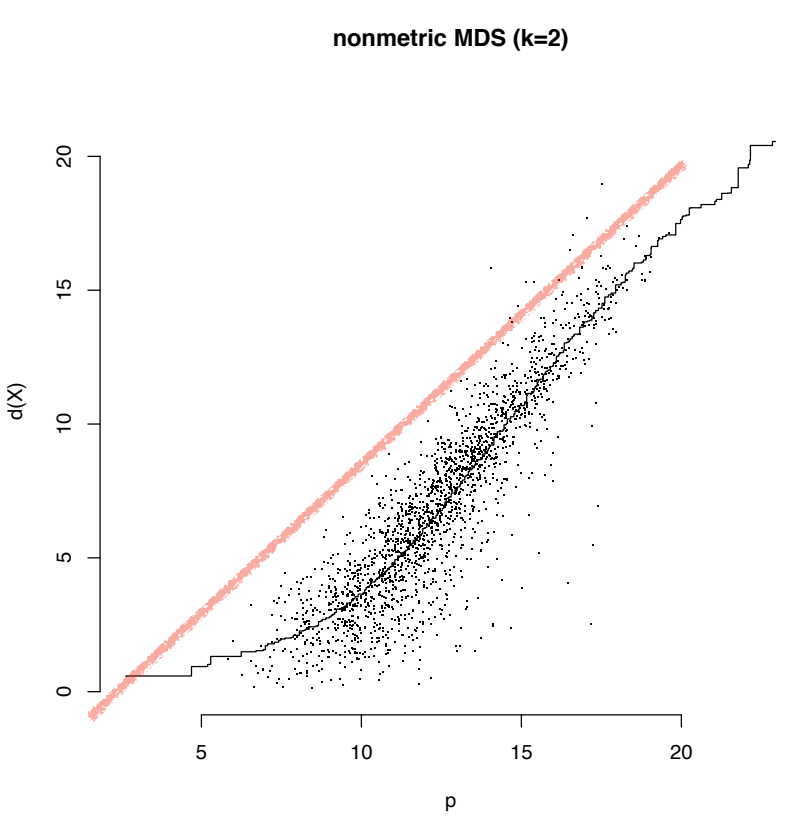
Municipal elections in Espoo in 2017

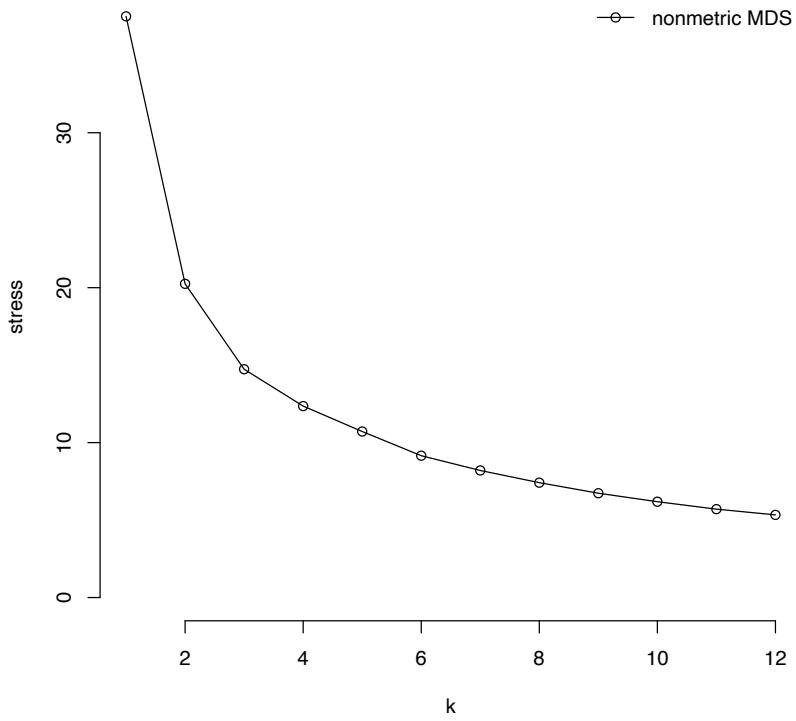
- Survey of candidates done by Helsingin Sanomat
- Here included only 10 parties with largest number of candidates nationally.
- Each candidate rated each of the **m=49** statements on a scale from 1 to 5, where 1=disagree and 5=agree:
    1. *Euthanasia should be allowed.*
    2. *I prefer public instead of the private sector to produce my local health services.*
    3. *Same gender couples should have the same marital and adoptation rights than the different genre couples.*
    4. *Good brother networks influence municipal decision-making.*
    5. *...*
- **n=515** candidates in total, i.e., we have a data 515x49 matrix.

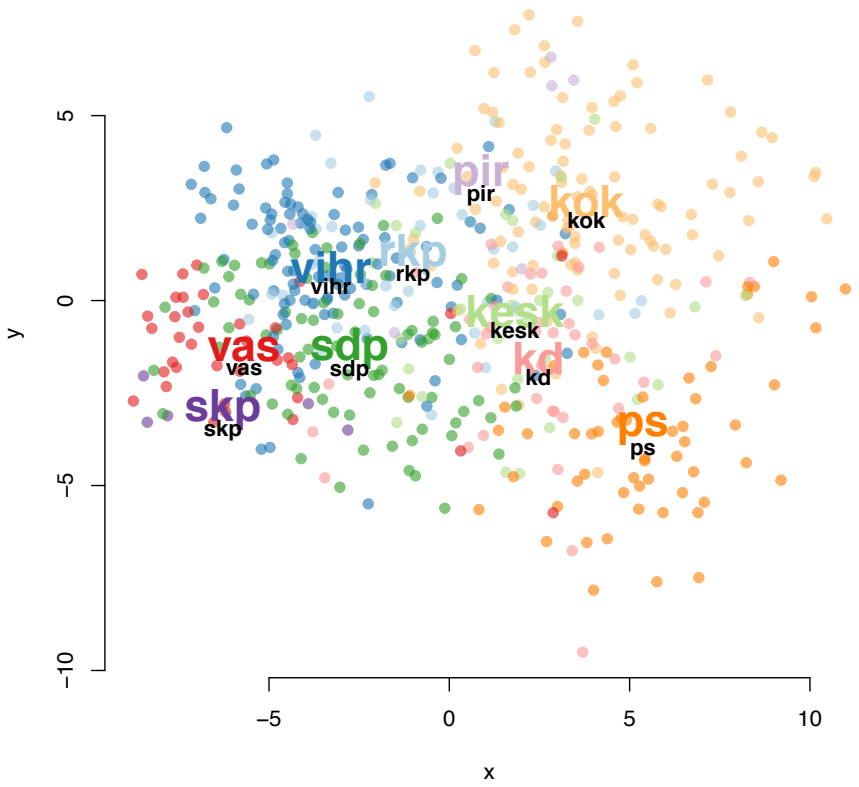
- Distance p_{ij} between candidates i and j is the Euclidean distance of their respective 49-dimensional rating vectors. What is a 2-dimensional representation that preserves these distances faithfully?

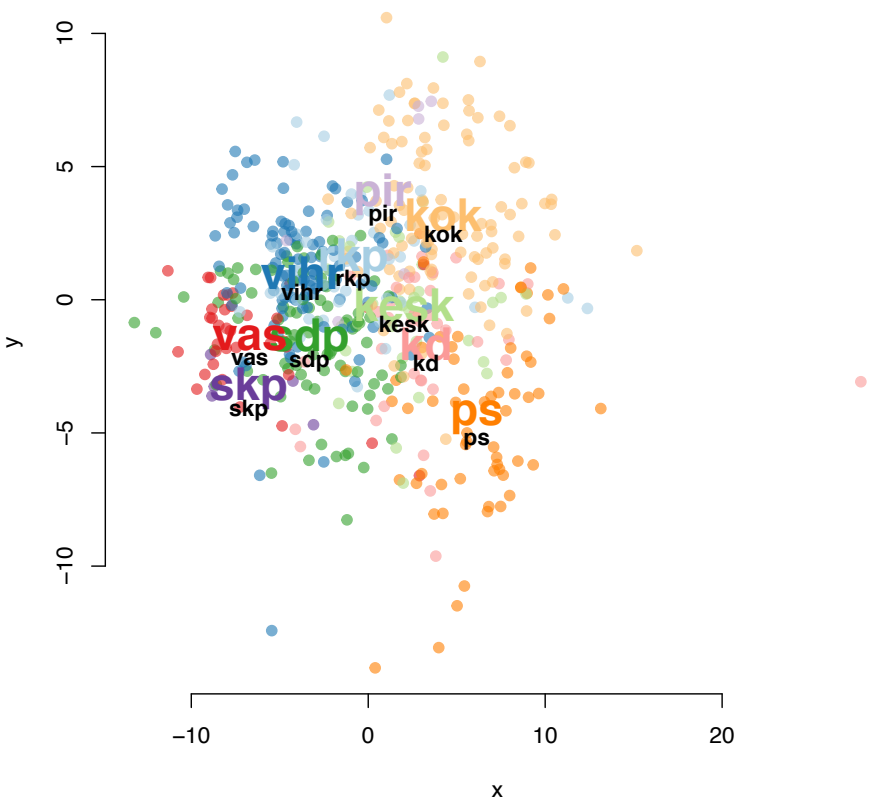
Municipal elections in Espoo in 2017

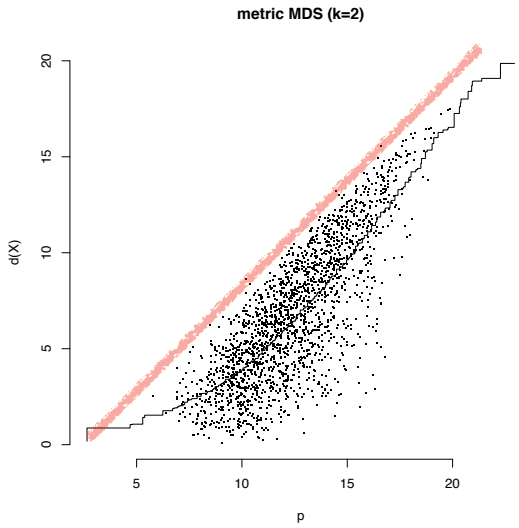


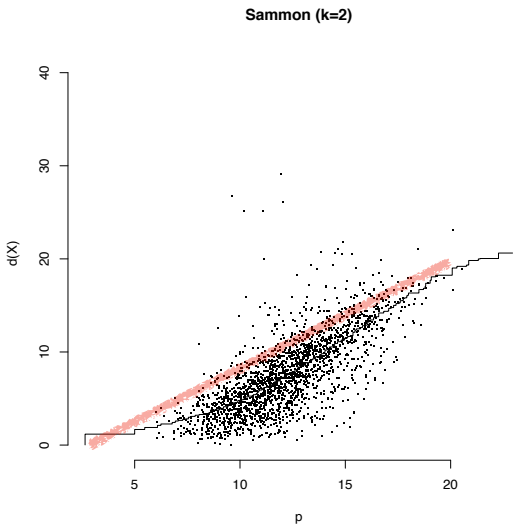






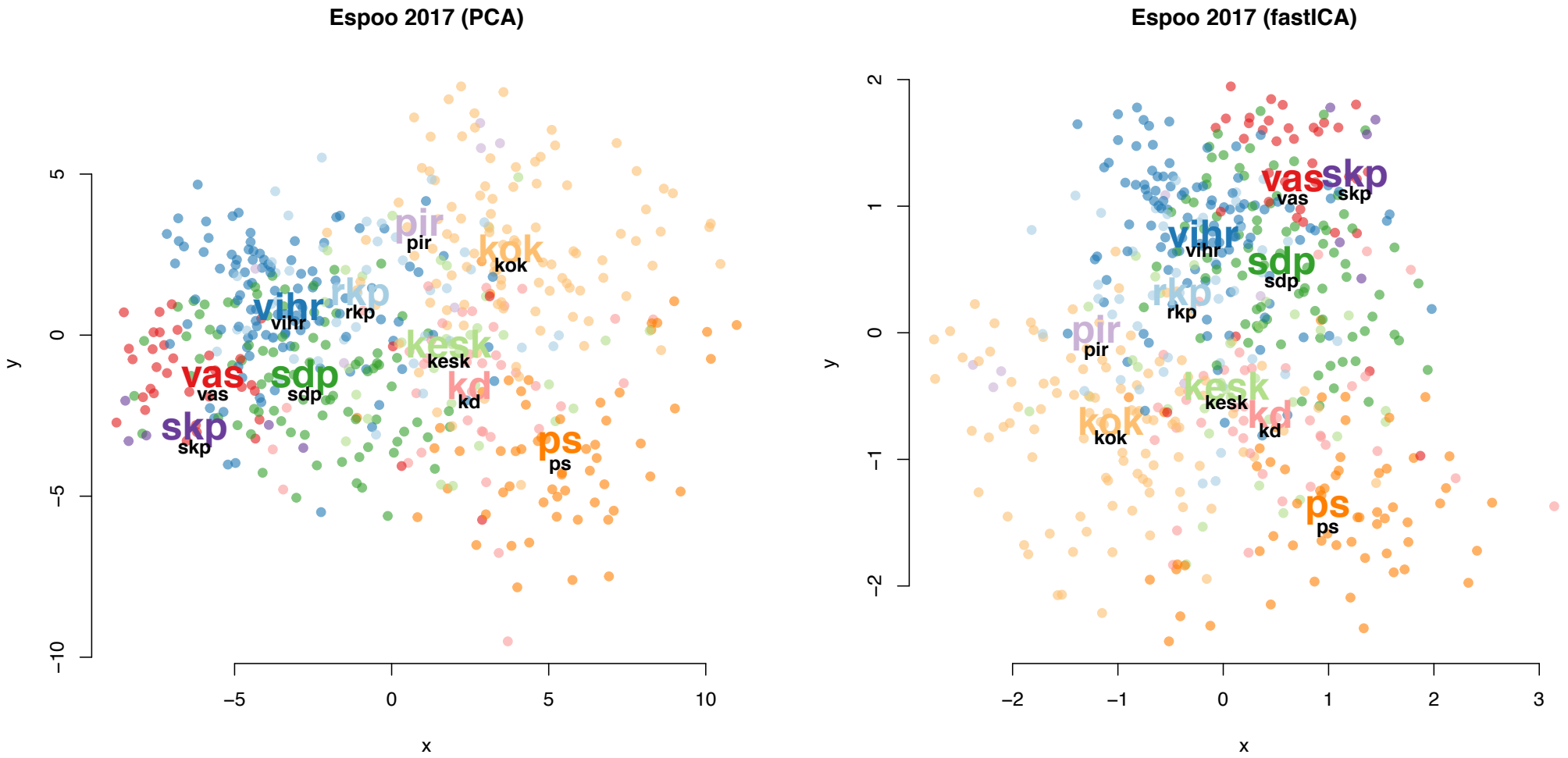






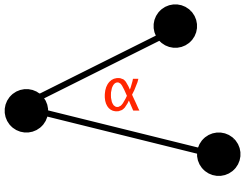
See http://www.iki.fi/kaip/p/dimensionality_reduction_1.nb.html

Municipal elections in Espoo in 2017

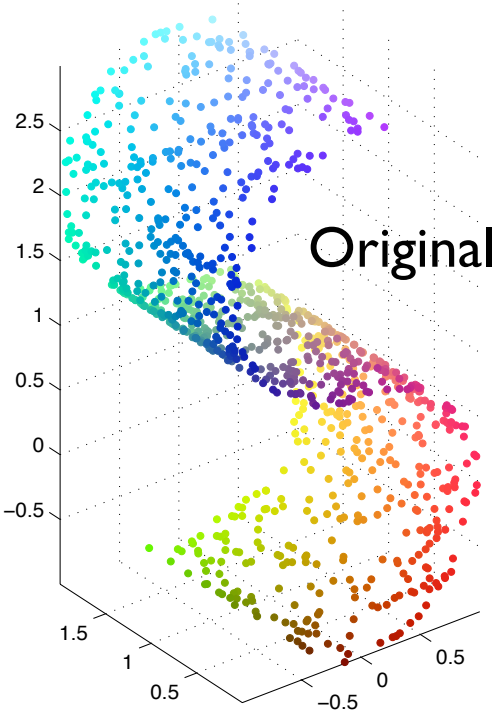


What to optimize

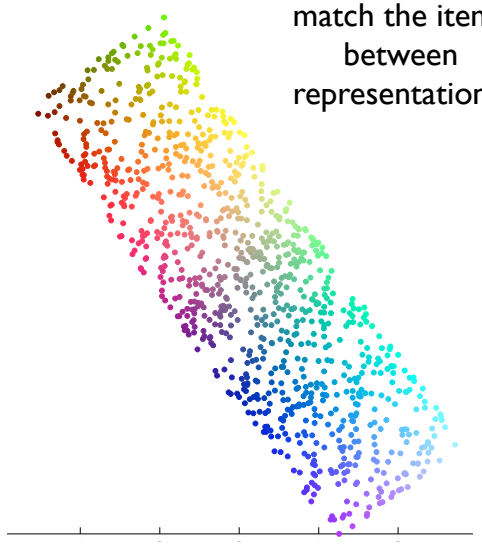
- Varying optimisation goals and complexities:
  - The (global) distances between nodes should be preserved as well as possible.
  - Focus on (local) neighborhoods of points
- Criteria for neighborhood similarity:
  - ***recall*** *(*sometimes called continuity, or *preservation of the original neighborhoods*): If the nodes are nearby in the original representation, they should also be nearby in the projection.
  - ***precision*** (sometimes called *trustworthiness*): If the nodes are nearby in the projection, they should also be nearby in the original representation.
  - **angles** between nearby nodes should be preserved as well as possible (*conformality*).



- Obviously, in a general case, some information will be inevitably lost in the projection (e.g., there is a trade-off between precision and recall)



RGB color coding is used to help to match the items between representations.



Embedding in 2D plane

Precision and recall

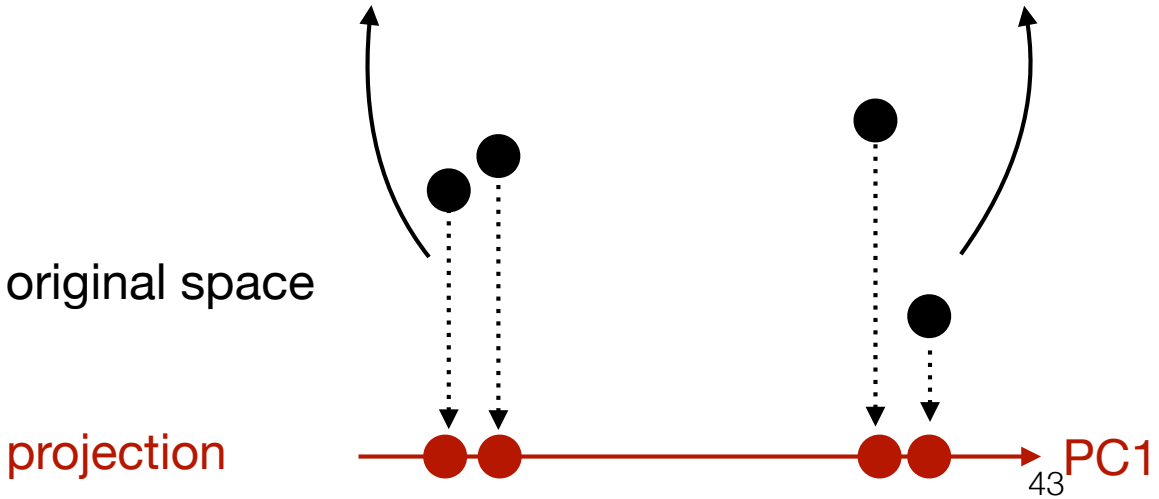
- **Precision:** if the points are nearby in embedding they are nearby in the original space

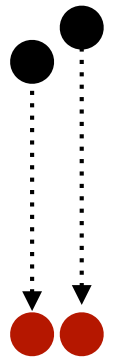
  proximity in the visualization is truthful

- **Recall:** if the points are nearby in the original space they are nearby in the embedding

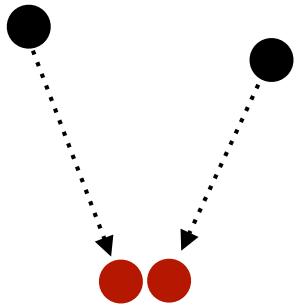
  proximities of the original are preserved

- Projection pursuit methods such as **PCA**:
  - the distance between the points in projection is at most the distance in the original space
  - always **good recall,** but possibly **bad precision**

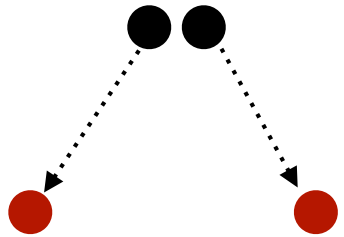




good precision and recall



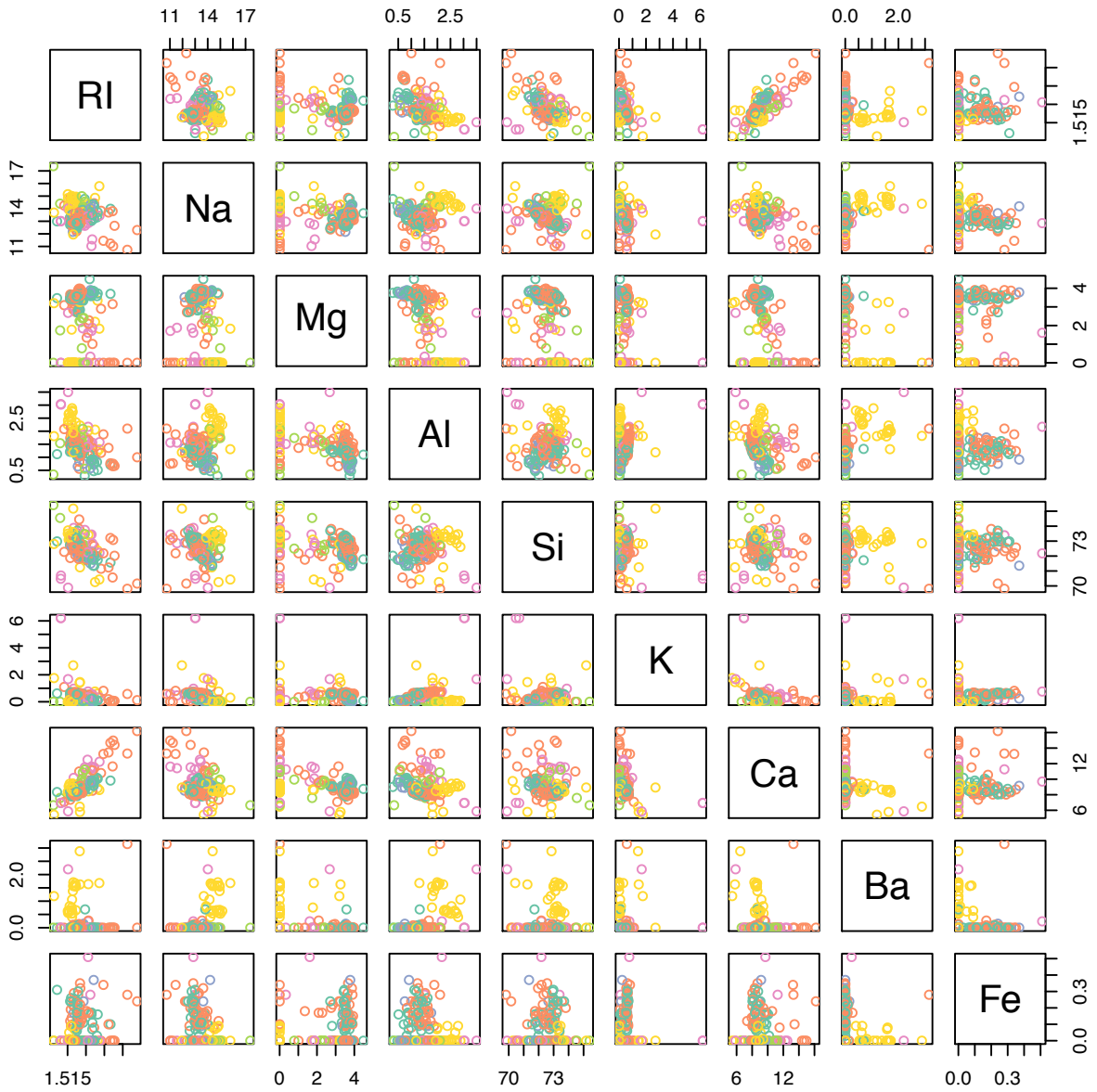
bad precision



bad recall

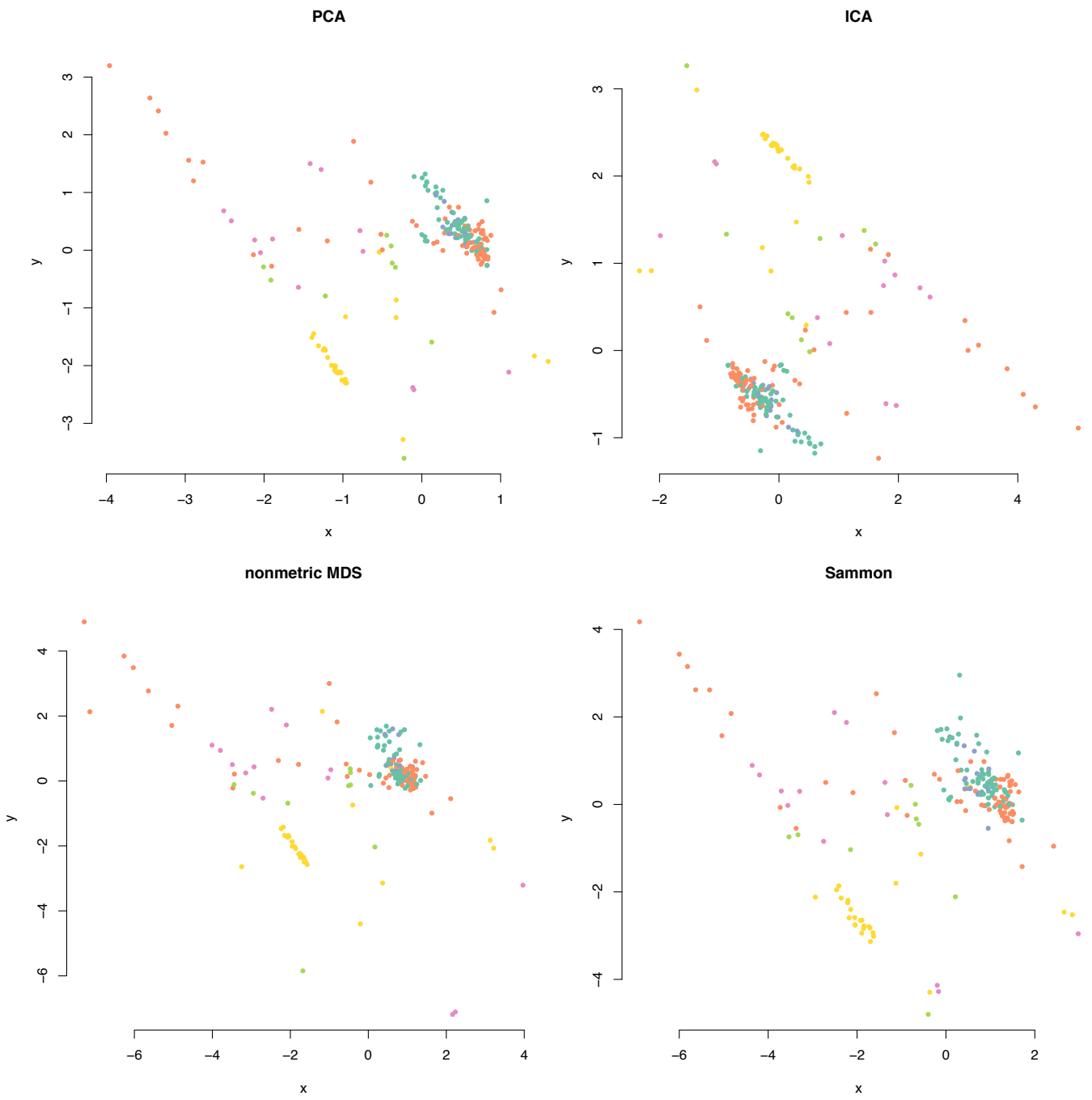
Glass data

- 9D glass identification database
- the amount of various metal elements in theglass



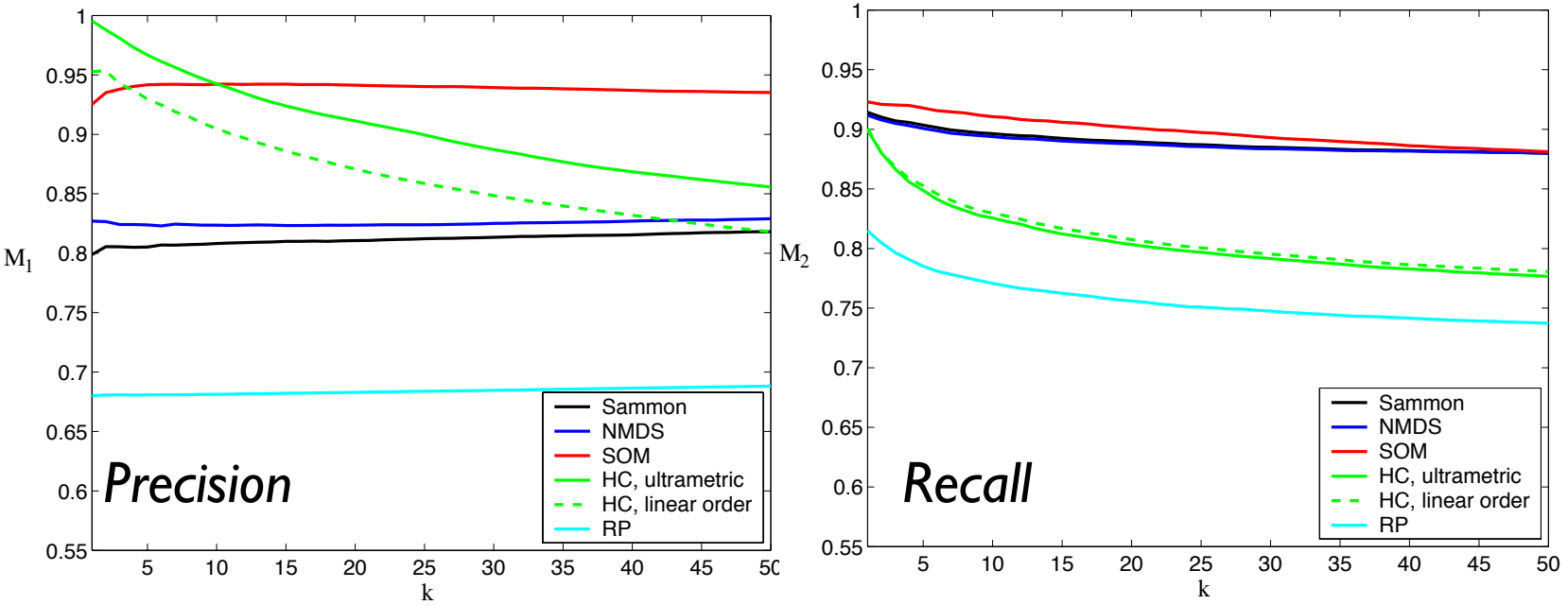
Glass data

- 9D glass identification database
- PCA always projects close-by points close to each other, resulting to reasonable recall
- However, PCA (and MDS) may also “collapse” far away data points into the same location (unless the data lies within low-dimensional linear subspace of the original space), this may lead to not so good precision



Performance of MDS

- MDS is tries to preserve the large distances at the expense of small ones, hence, it can "collapse" some small distances on the expense of preserving large distances
- *Precision* and *recall* defined for neighbourhoods of data points:
  - *trustworthy (precision)*: *k* closest neighbours of a point in projection are also close in the original
  - *preserves the original neighbourhoods* (*recall*): all *k* closest neighbours of a point in original space are also close in the projection.



Figures from Kaski, et al. 2003, https://doi.org/10.1186/1471-2105-4-48

Precision and recall as a function of the neighbourhood size *k* for a yeast data set. Non-metric (ordinal) MDS (NMDS) is shown in blue. Larger precision and recall is better.

Performance of MDS

- Relatively better recall, worse precision

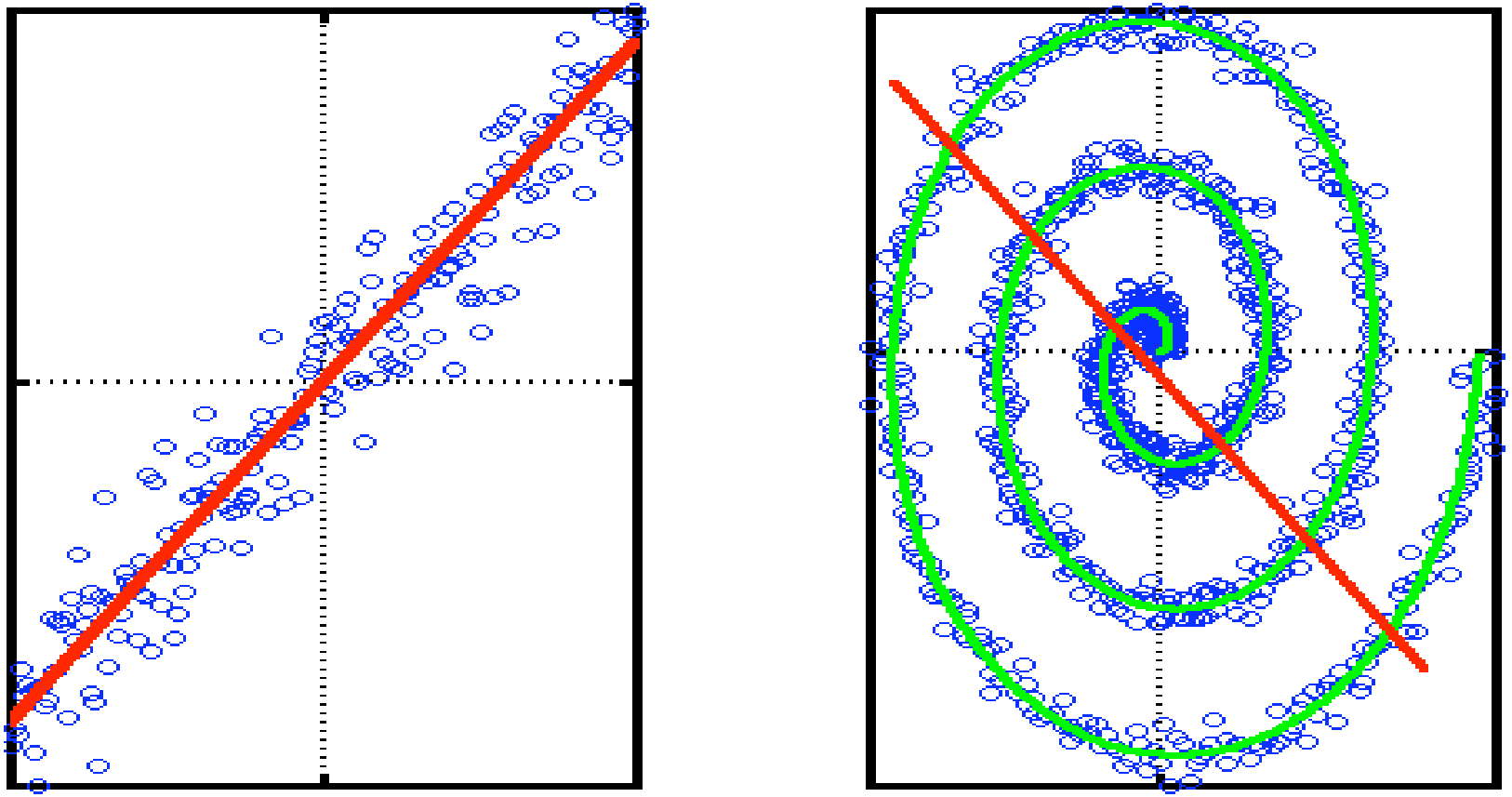
- MDS algorithms typically have running times of the order $O(N^2)$, where N is the number of data items.
- This is not very good: N=1,000 data items are ok, but N=1,000,000 is getting slow.

- Some solutions: use landmark points (i.e., use MDS only on a subset of data points and place the remaining points according to those, use MDS on cluster centroids etc.), use some other algorithm or modification of MDS.

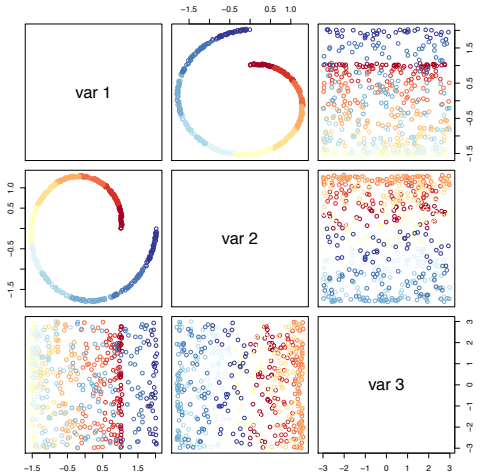
- MDS is not guaranteed to find the global optimum of the stress (cost) function, nor it is guaranteed to converge to the same solution at each run (many of the MDS algorithms are quite good and reliable, though)

Visualising manifolds

manifold = low-dimensional set (curve, surface, …) in higher-dimensional space



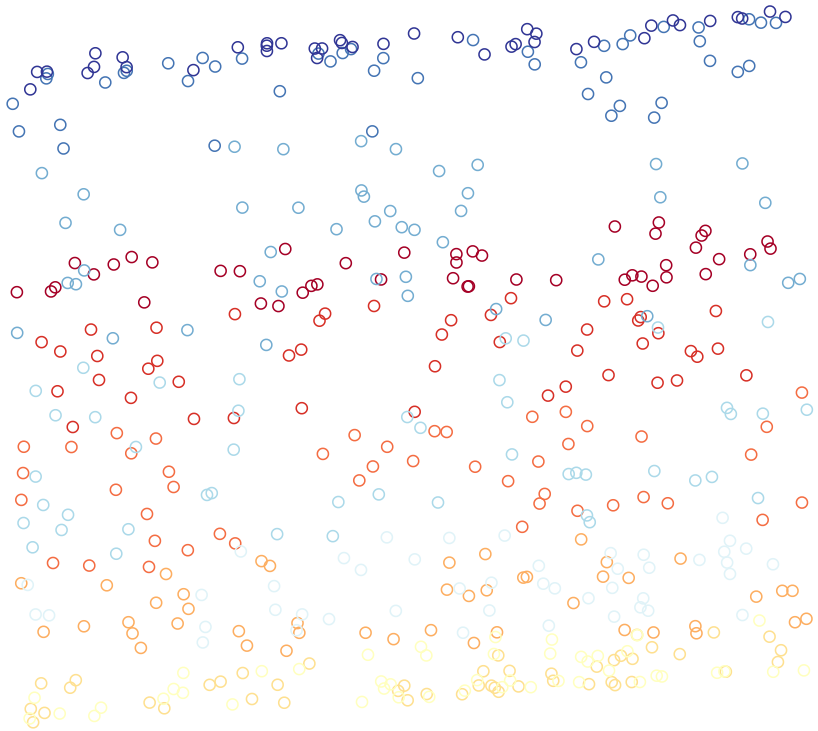
- The first principal component is given by the red line. The green line on the right gives the “correct” non-linear dimension (which PCA is of course unable to find).



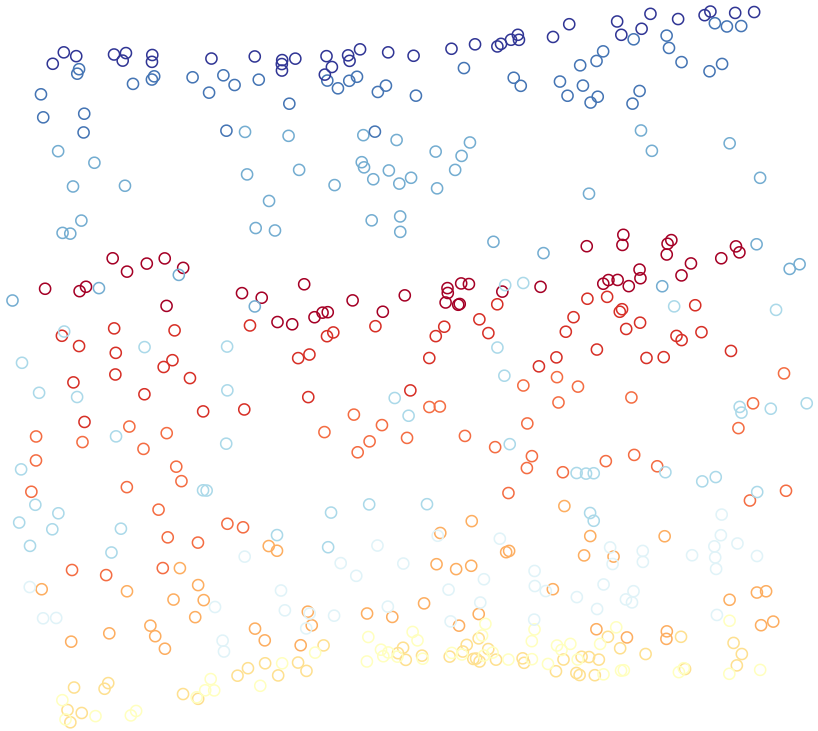
Swiss roll

(a curved 2D-manifold in 3D space)

PCA



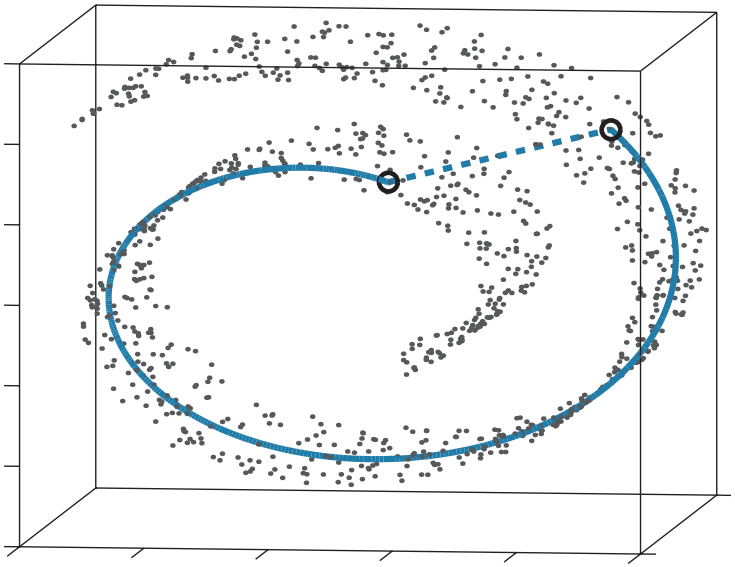
nonmetric MDS



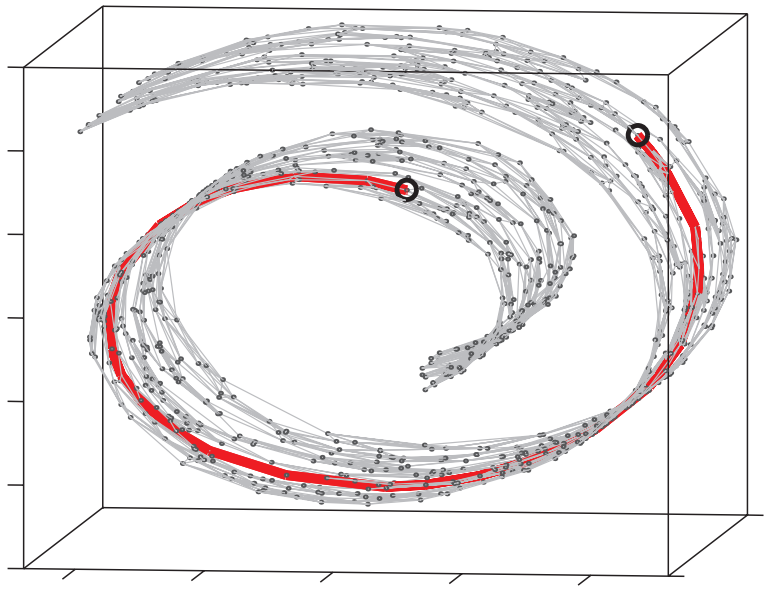
Isometric mapping of data manifolds (ISOMAP)

- Tenenbaum et al. 2000, https://doi.org/10.1126/science.290.5500.2319 See http://web.mit.edu/cocosci/isomap/datasets.html (fig)
- ISOMAP is an example of graph-based methods.
- ISOMAP is a variant of MDS. The difference to MDS is in how the distances (or proximities) are defined.
- ISOMAP first finds **k nearest neighbours** for each data point and constructs a k-nearest-neighbours graph. The distance between two data points (that are not nearest neighbours) is defined as the **topological** a.k.a. **graph-theoretical distance** (shortest path, i.e. minimum number of links) between the points.
- The resulting distances are fed to the standard linear (metric, because triangle inequality is satisfied) MDS, which finds the actual embedding.

Original data. The graph-distance between two items is shown by solid line, a shortcut is shown by the dotted line.



k-nearest neighbours graph used to find the graph-theoretical distances.

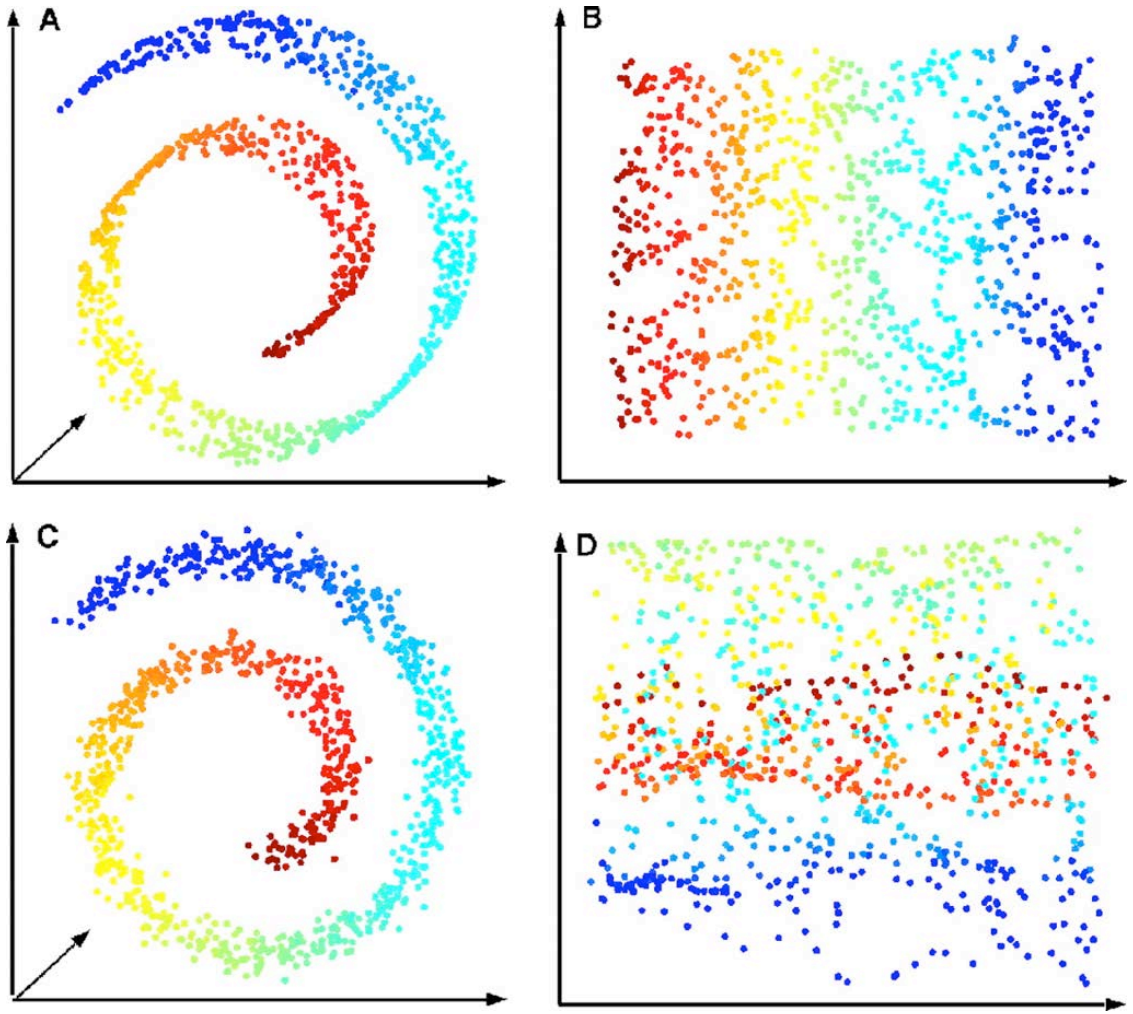


Isometric mapping of data manifolds (ISOMAP)

- Assumptions:
  - graph is connected
  - neighbourhood on graph reflects neighbourhoods on manifolds (no “shortcuts”)

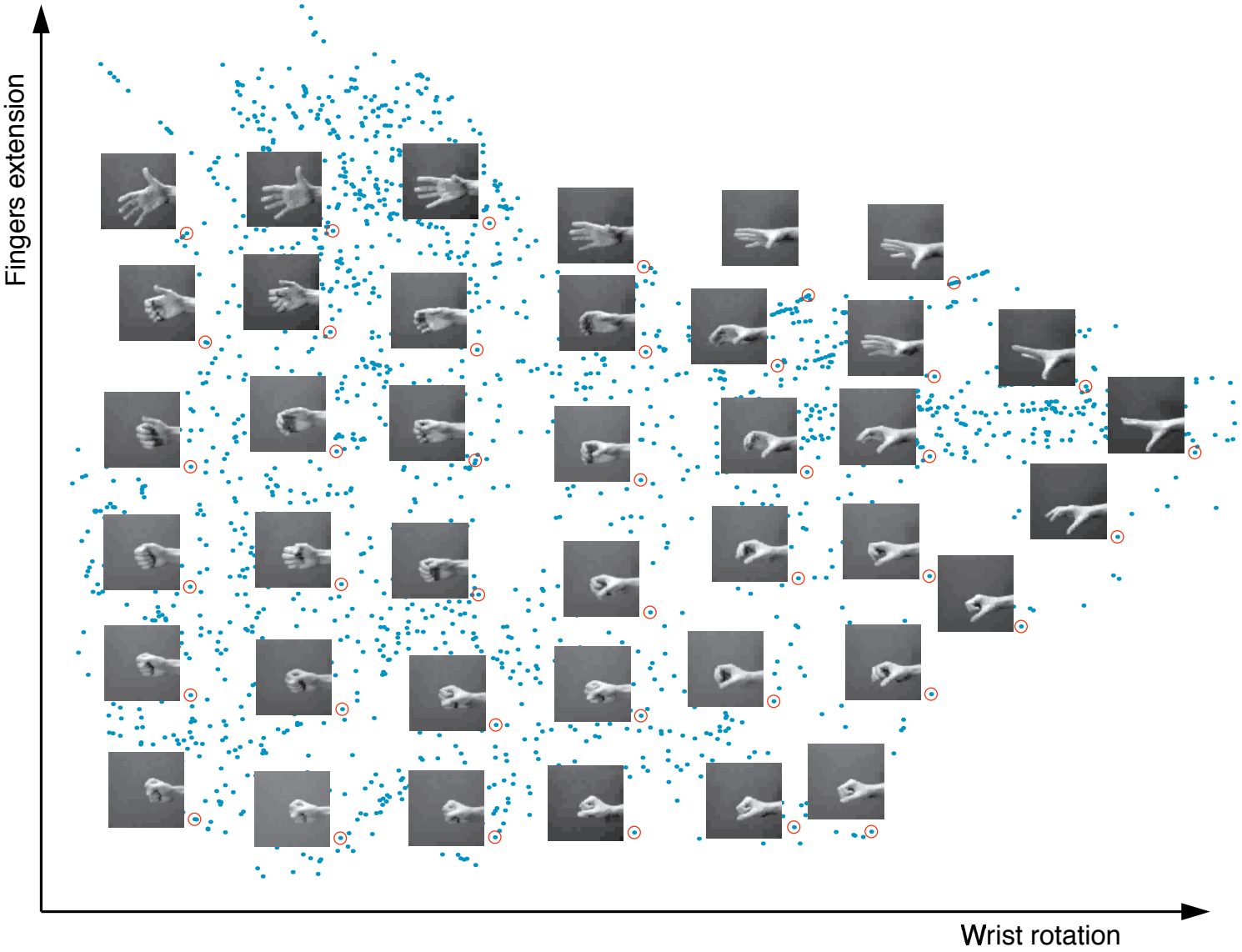
- Weakness (Balasubramian et al.2002, https://doi.org/10.1126/science.295.5552.7a, fig):
  - sensitive to shortcuts (making the algorithm topologically unstable, see the figure right)
- Time complexity ~$O(N^2)$

- Extension: landmark ISOMAP
  - identify subsets of inputs as landmarks, makes the algorithm faster



(A) The “Swiss roll” data used by Tenenbaum et al. (1) to illustrate their algorithm (n = 1000). (B) The two-dimensional (2D) representation computed by the ε−Isomap variant of the Isomap algorithm, with ε = 5. Nearby points in the 2D embedding are also nearby points in the 3D manifold, as desired. (C) Data shown in A, with zero-mean normally distributed noise added to the coordinates of each point, where the standard deviation of the noise was chosen to be 2% of smallest dimension of the bounding box enclosing the data. (D) The Isomap (ε = 5) solution for the noisy data. There are gross “folds” in the embedding, and neither the metric nor the topological structure of the solution in (B) is preserved.

Application case of ISOMAP



ISOMAP (k=6) applied to 2,000 images of a hand in different configurations. The images were generated by making a series of opening and closing movements of the hand at different wrist orientations, designed to give rise to a two-dimensional manifold. The images were treated as 4,096-dimensional (= 64x64 pixels) vectors, with input-space distances defined in the Euclidean metric.

Locally linear embedding (LLE)

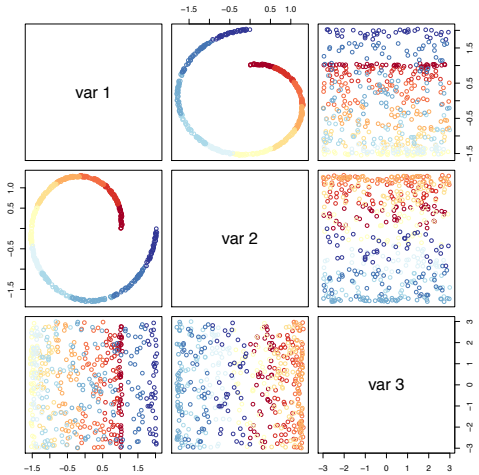
- LLE tries to maintain the relationships of nearby points
- Roweis et al. 2000, https://doi.org/10.1126/science.290.5500.2323

- Recipe:
  1. find the set *N(i)*, *k* closest data points to *i*th data point x_i
  2. try to express x_i as a linear combination of its neighbours: find weights minimising

in original space $$\sum_i \left(x_i - \sum_{j \in N(i)} w_{ij} x_j \right)^2 \quad \text{s.t.} \quad \sum_{j \in N(i)} w_{ij} = 1$$

  3. fix the weights, and find points in plane (y_i are the coordinates in embedding) minimising

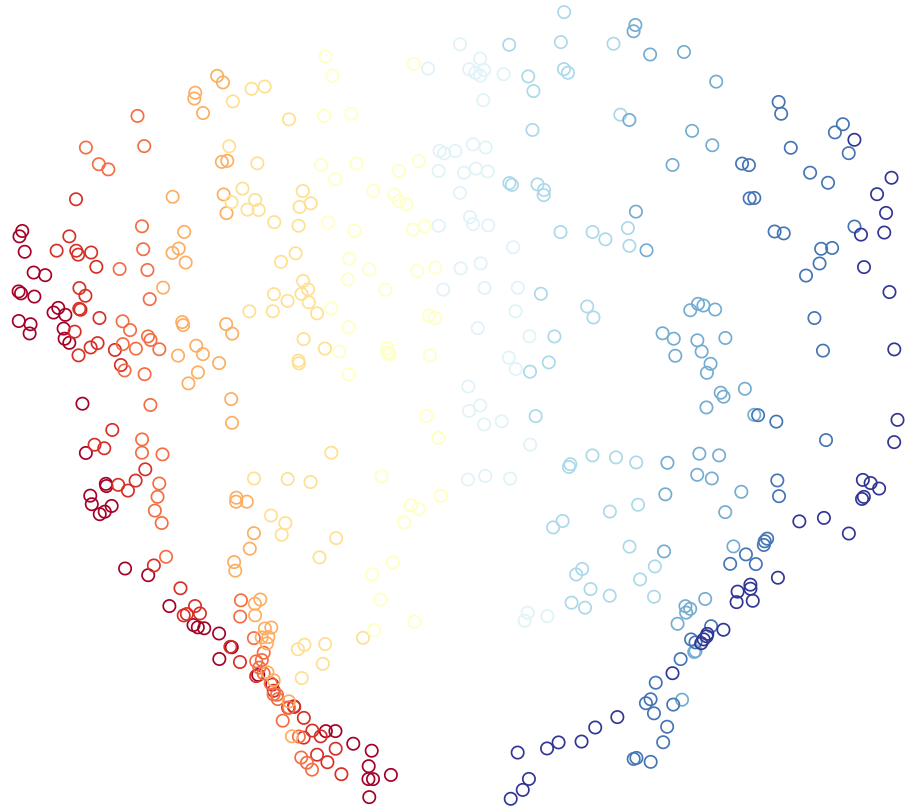
in target space $$\sum_i \left(y_i - \sum_{j \in N(i)} w_{ij} y_j \right)^2$$



Swiss roll

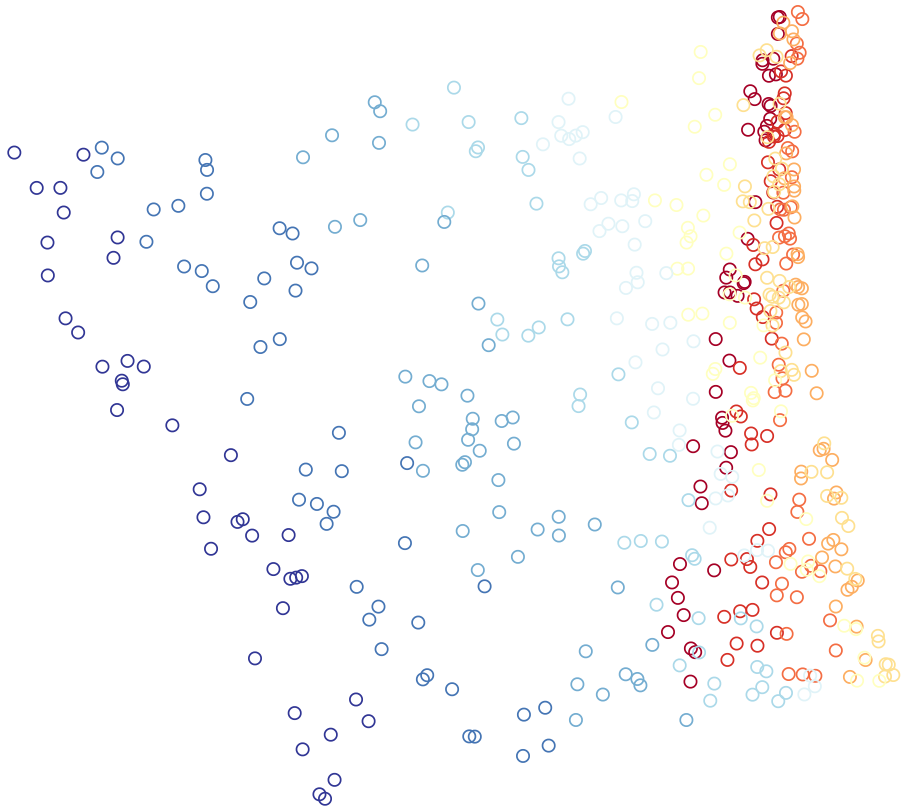
ISOMAP

topological distances

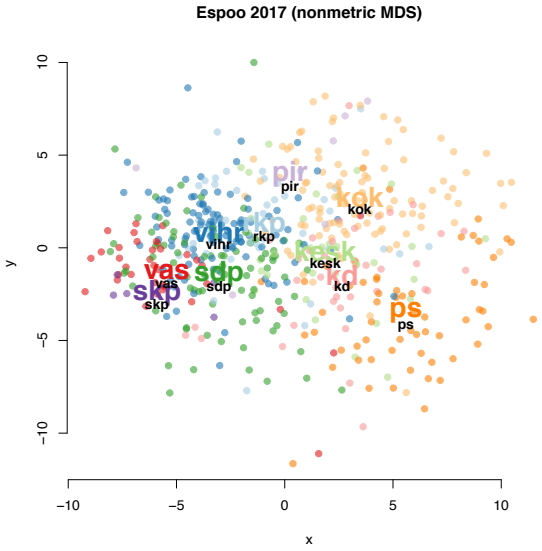


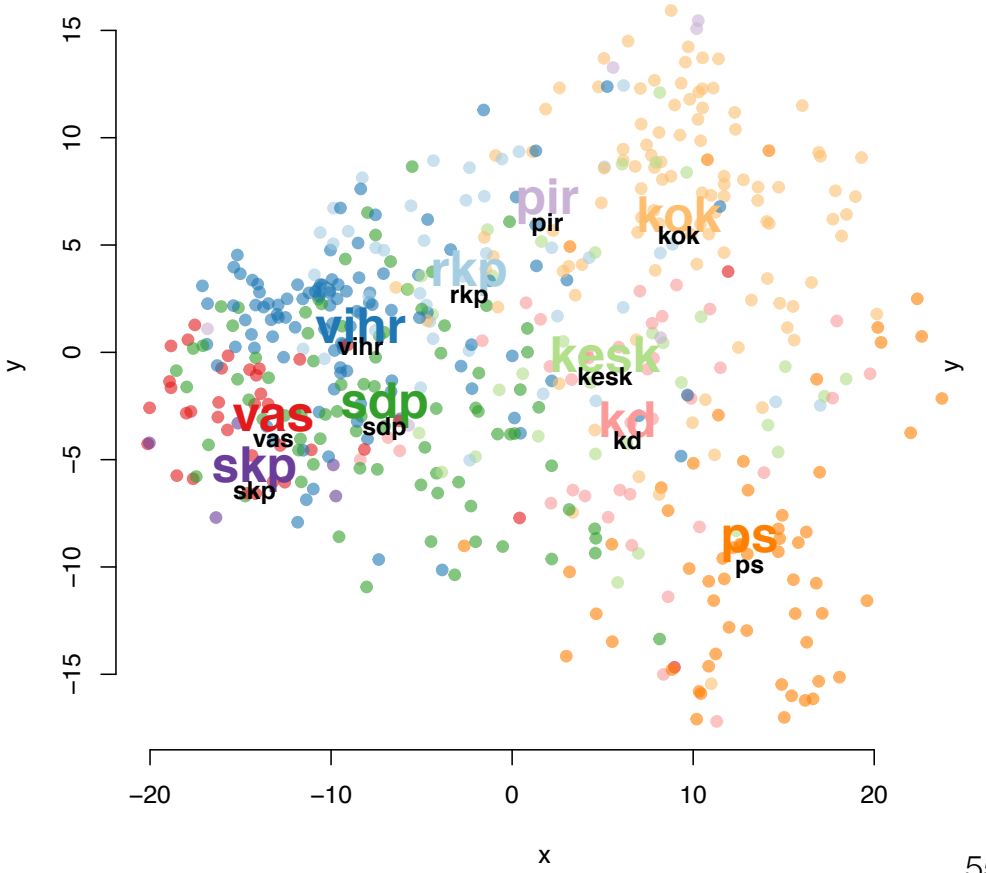
LLE

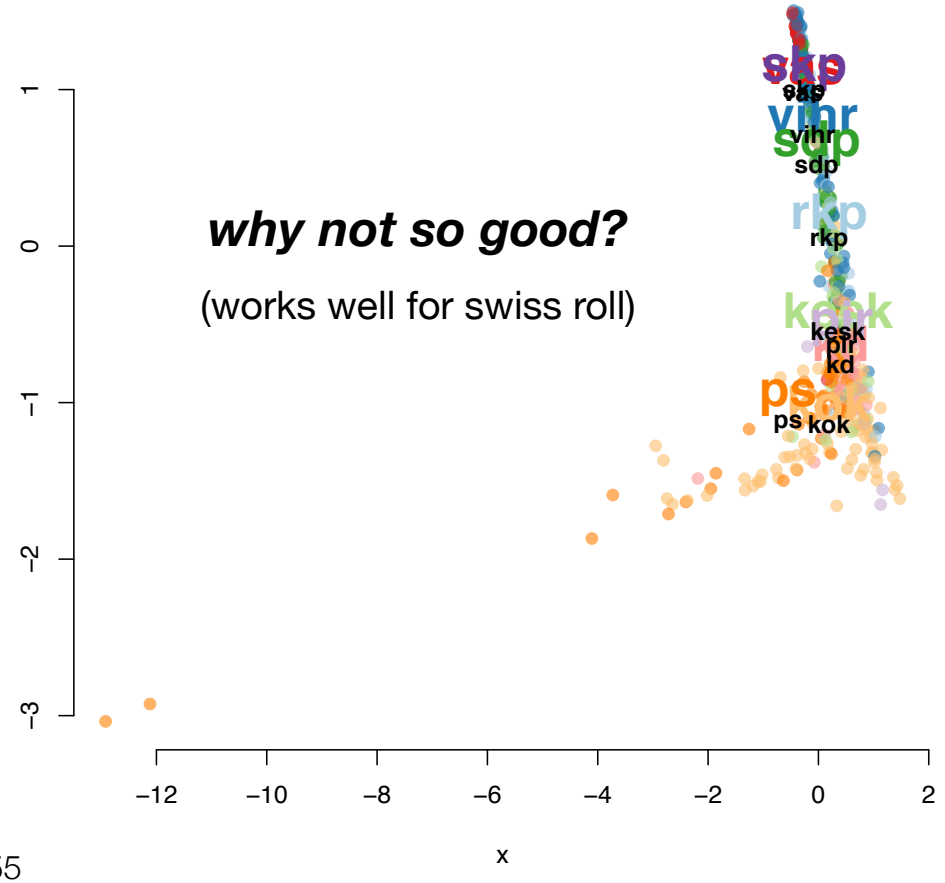
local metric distances



Municipal elections in Espoo in 2017







Laplacian eigenmap

- Eigenmap is a spectral method, like PCA.

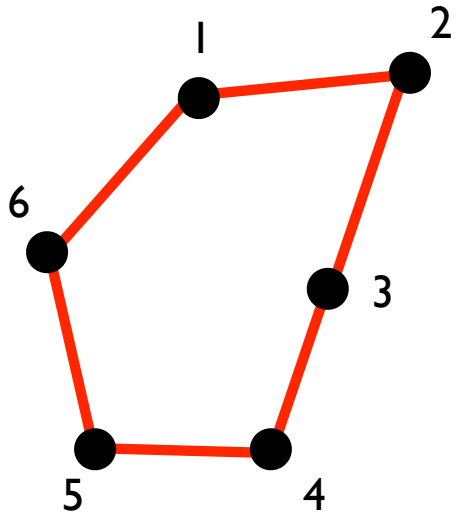
- Recipe:
  1. As in ISOMAP, construct *k*-nearest neighbors graph.
  2. Assign $W_{ij}=1$, if *i* and *j* are neighbours, otherwise assign $W_{ij}=0$.
  3. Define diagonal matrix *D*, $D_{ii}=\sum_j W_{ij}$, and graph Laplacian, $L=D-W$.
  4. The embedding of data points is given by the eigenvectors of *L*, corresponding to the *d* smallest non-zero eigenvalues.

- Physical intuition: find lowest frequency vibrational modes of a mass-spring system (mass=nodes, springs=links of the graph).
- Very straightforward to implement, e.g., with R

Laplacian eigenmap

N=6

k=2




```
> L
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]   -2    1    0    0    0    1
[2,]    1   -2    1    0    0    0
[3,]    0    1   -2    1    0    0
[4,]    0    0    1   -2    1    0
[5,]    0    0    0    1   -2    1
[6,]    1    0    0    0    1   -2
> s <- svd(L)
> s$u
           [,1]          [,2]       [,3]       [,4]          [,5]       [,6]
[1,] -0.4082483 -1.934666e-16 -0.5773503 -0.5773503 -3.951502e-16  0.4082483
[2,]  0.4082483  5.000000e-01  0.2886751 -0.2886751 -5.000000e-01  0.4082483
[3,] -0.4082483 -5.000000e-01  0.2886751  0.2886751 -5.000000e-01  0.4082483
[4,]  0.4082483  4.163336e-16 -0.5773503  0.5773503  2.081668e-16  0.4082483
[5,] -0.4082483  5.000000e-01  0.2886751  0.2886751  5.000000e-01  0.4082483
[6,]  0.4082483 -5.000000e-01  0.2886751 -0.2886751  5.000000e-01  0.4082483
> s$d
[1] 4.000000e+00 3.000000e+00 3.000000e+00 1.000000e+00 1.000000e+00 1.155603e-16
```

eigenvectors

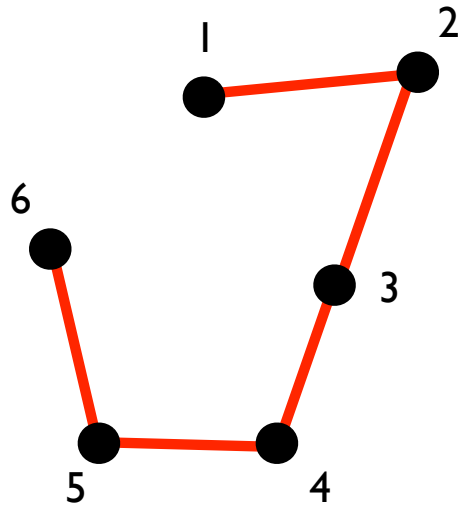
circular shape

zero eigenvalue

Laplacian eigenmap

N=6

k=1

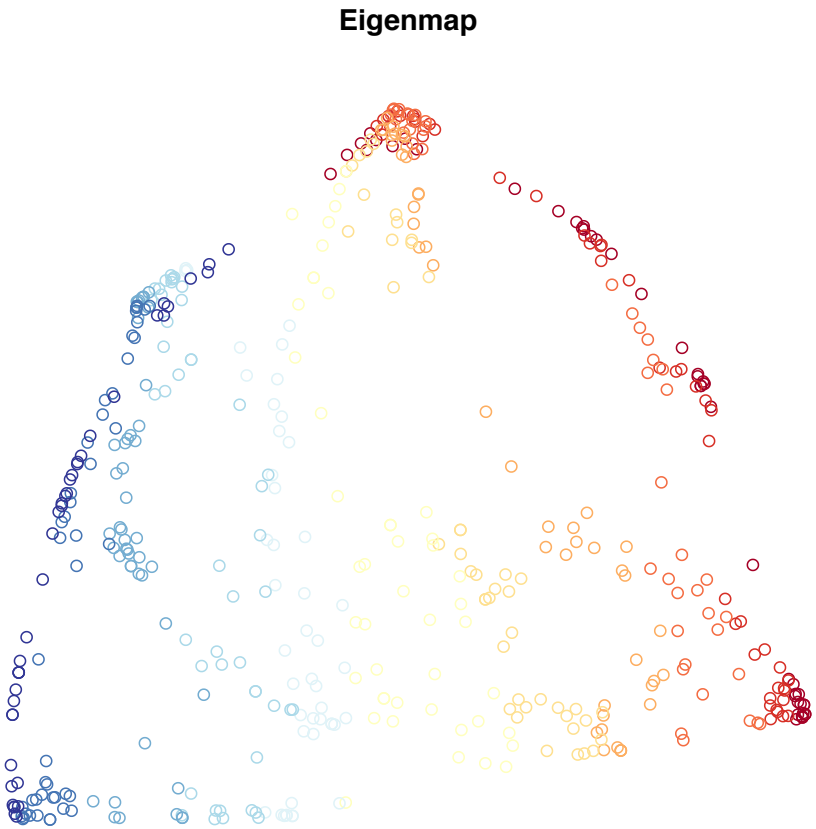


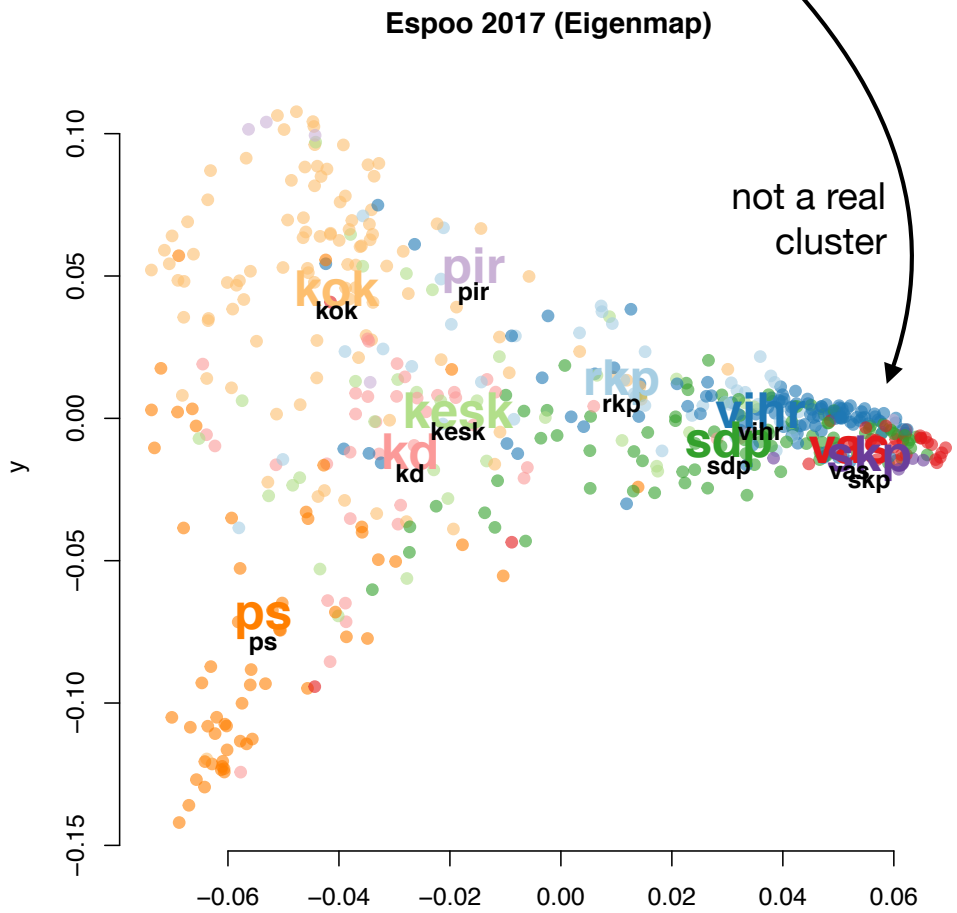

```
> L
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]   -1    1    0    0    0    0
[2,]    1   -2    1    0    0    0
[3,]    0    1   -2    1    0    0
[4,]    0    0    1   -2    1    0
[5,]    0    0    0    1   -2    1
[6,]    0    0    0    0    1   -1
> s <- svd(L)
> s$u
           [,1]       [,2]       [,3]          [,4]       [,5]      [,6]
[1,] -0.1494292 -0.2886751 -0.4082483  5.000000e-01  0.5576775 0.4082483
[2,]  0.4082483  0.5773503  0.4082483  2.775558e-16  0.4082483 0.4082483
[3,] -0.5576775 -0.2886751  0.4082483 -5.000000e-01  0.1494292 0.4082483
[4,]  0.5576775 -0.2886751 -0.4082483 -5.000000e-01 -0.1494292 0.4082483
[5,] -0.4082483  0.5773503 -0.4082483  8.326673e-17 -0.4082483 0.4082483
[6,]  0.1494292 -0.2886751  0.4082483  5.000000e-01 -0.5576775 0.4082483
> s$d
[1] 3.732051e+00 3.000000e+00 2.000000e+00 1.000000e+00 2.679492e-01 7.510881e-17
```

linear ordering

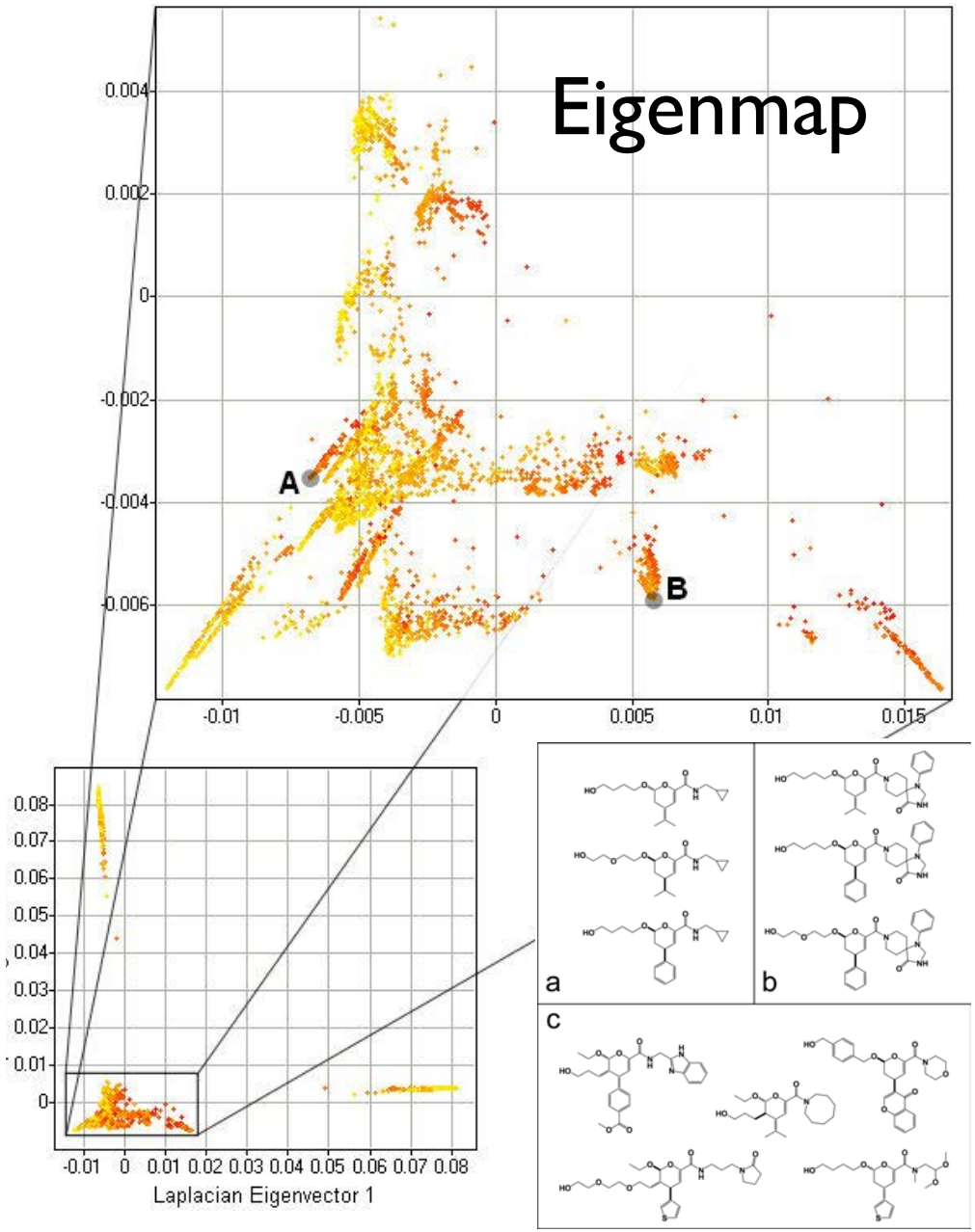
Laplacian eigenmap

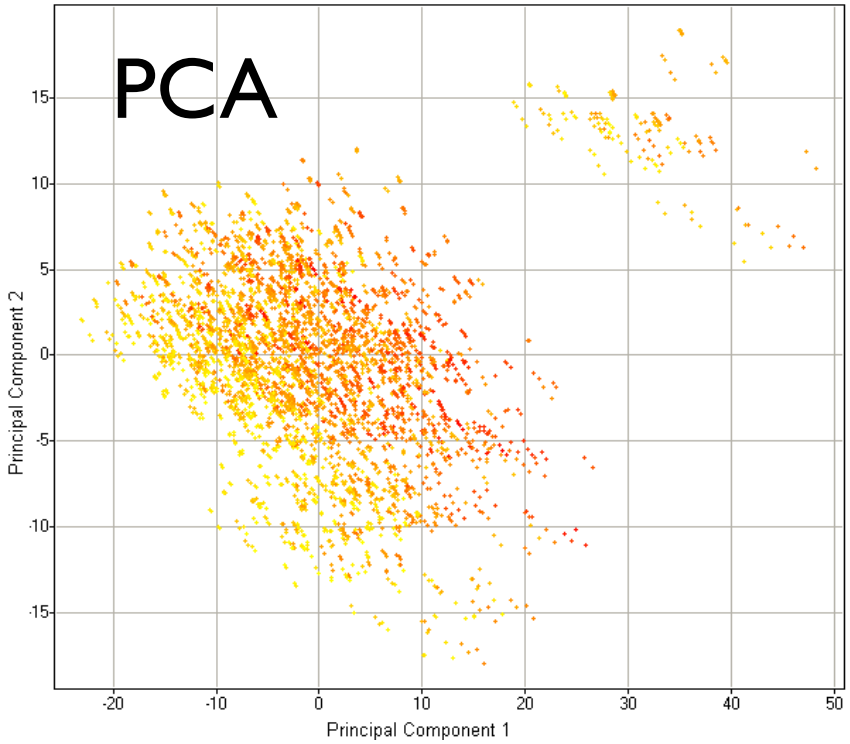
- Eigenmap can be viewed as trying to preserve the expected time a random walk on the neighbourhood graph takes to travel from one point to the other and back. This leads to tendency to magnify some distances (and shrink others), leading to relatively bad precision.





Laplacian eigenmap





Eigenmap (unlike PCA) shows clusters of similar chemical compounds (A&B). The input data is a network of small molecules encoded as molecular descriptors and connected by similarity.

Forman et al. 2005, https://doi.org/10.1186/1471-2105-6-260

Curvilinear component analysis (CCA)

- Demartines et al. 1997, https://doi.org/10.1109/72.554199
- *Curvilinear component analysis* (CCA) is like (absolute) MDS, except that only short distances are taken into account.
- More formally, the cost function reads

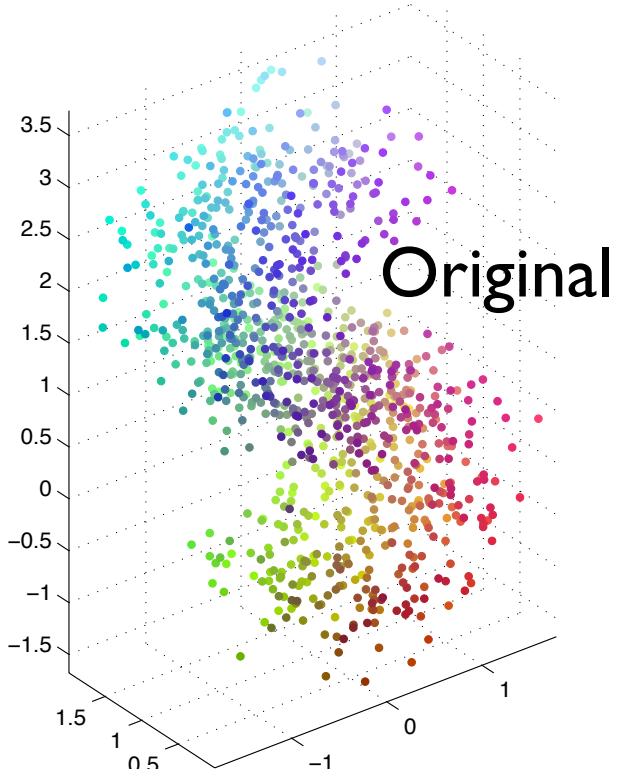
$$\sigma_r = \sum_{i<j} \left(d(x_i, x_j) - d(y_i, y_j)\right)^2 F(d(y_i, y_j), \lambda_y)$$

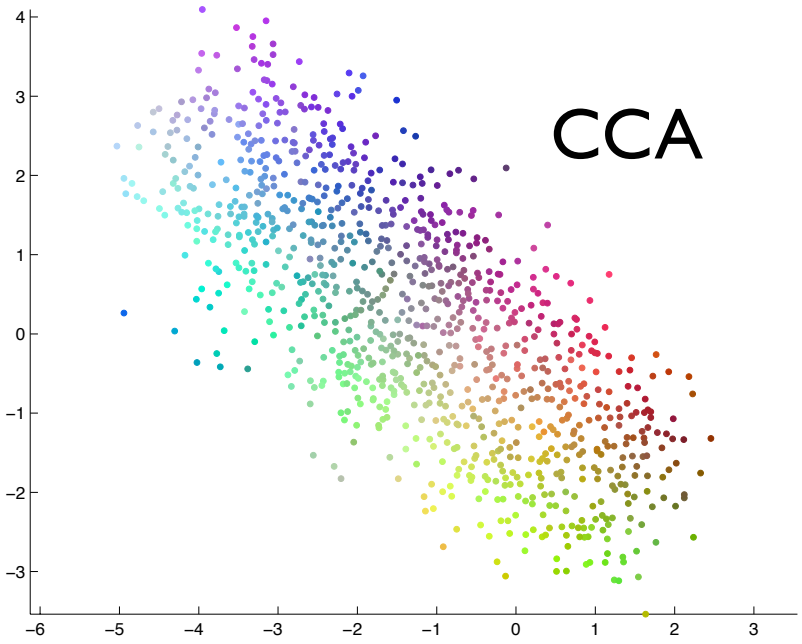
where $F(d,\lambda_y)$ equals unity, if $d<\lambda_y$, and zero otherwise; and d denotes the Euclidean distance of points in the original space (x) and in the projection (y), respectively.

[$F(d,\lambda_y)$, could be any monotonically decreasing function in d.]

Curvilinear component analysis (CCA)

- CCA performs generally well in terms of precision; it appears to be quite robust.
- Notice outliers at right: they are result of small neighbourhood.





Figures by Jarkko Venna

Local multidimensional scaling

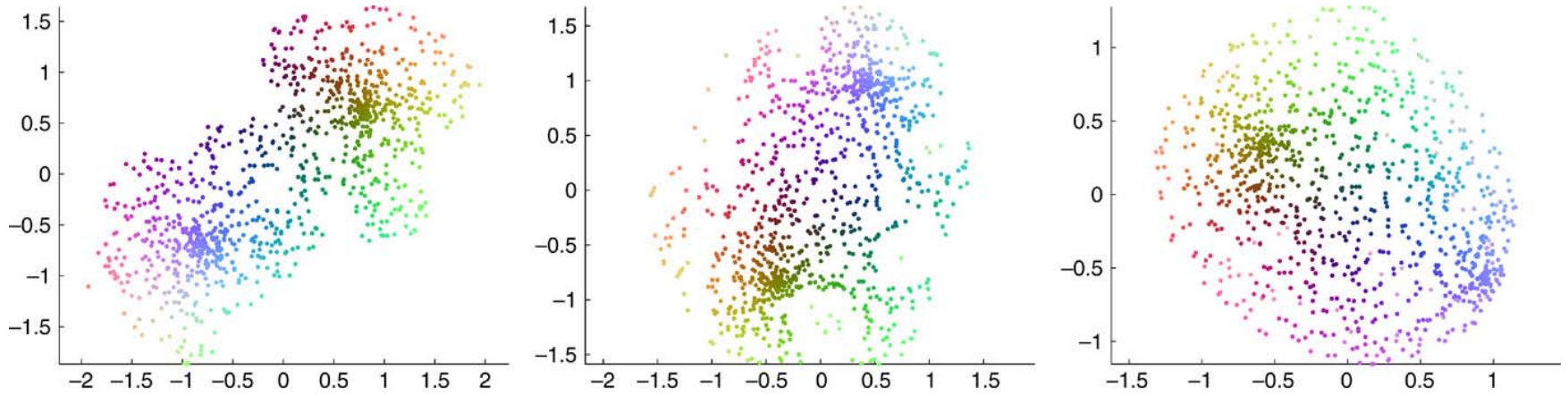


Fig. 6. Three projections of a three-dimensional spherical cell with local MDS. On the left, trustworthiness of the projection is maximized by selecting $\lambda = 0$. In the middle and right, discontinuity of the projection is penalized as well, by setting $\lambda = 0.1$ and $\lambda = 0.9$, respectively. *)

- Extension of curvilinear component analysis (CCA obtained when λ=0)
- Parameter λ controls the tradeoff between precision and recall
- Venna et al. 2006, https://doi.org/10.1016/j.neunet.2006.05.014

$$
\begin{aligned}
E = & \frac{1}{2}\sum_i \sum_{j\neq i} [(1-\lambda)(d(\mathbf{x}_i,\mathbf{x}_j) \\
& - d(\mathbf{y}_i,\mathbf{y}_j))^2 F(d(\mathbf{y}_i,\mathbf{y}_j),\sigma_i) \\
& + \lambda(d(\mathbf{x}_i,\mathbf{x}_j) - d(\mathbf{y}_i,\mathbf{y}_j))^2 F(d(\mathbf{x}_i,\mathbf{x}_j),\sigma_i)] \\
= & \frac{1}{2}\sum_i \sum_{j\neq i} (d(\mathbf{x}_i,\mathbf{x}_j) - d(\mathbf{y}_i,\mathbf{y}_j))^2 \\
& \times [(1-\lambda)F(d(\mathbf{y}_i,\mathbf{y}_j),\sigma_i) + \lambda F(d(\mathbf{x}_i,\mathbf{x}_j),\sigma_i)].
\end{aligned}
$$

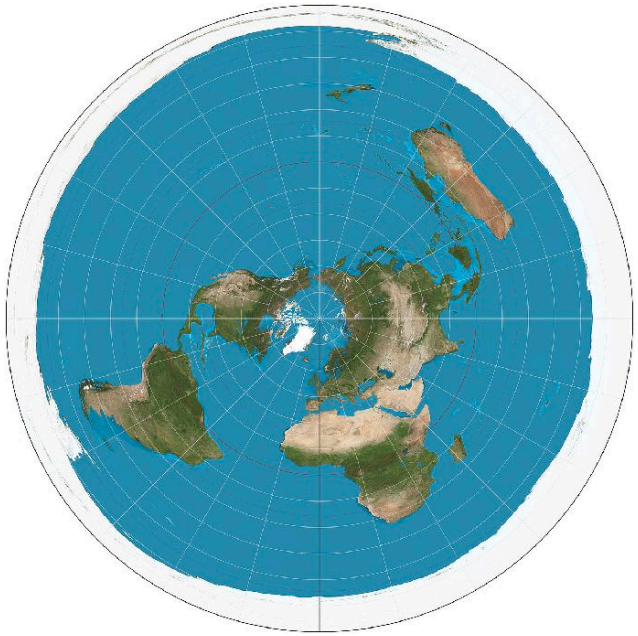
*) "trustworthiness" = precision, "continuity" = recall

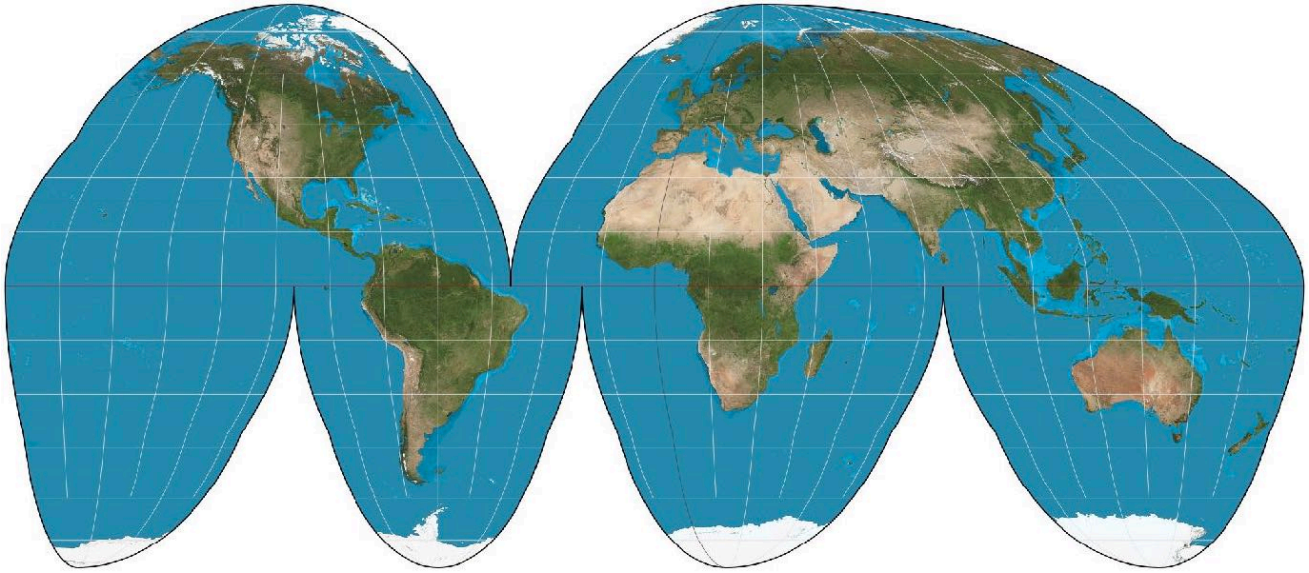
Which properties are important to retain?

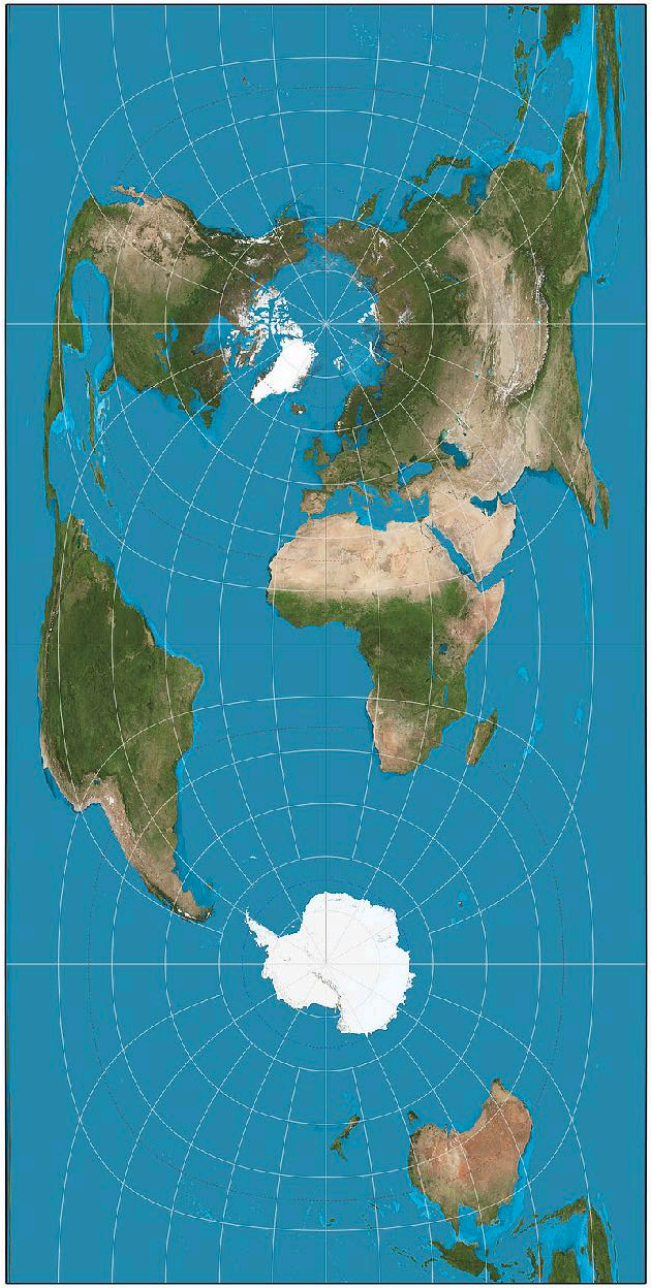
https://en.wikipedia.org/wiki/List_of_map_projections

Select embedding based on:

- long/short distances
- directions
- local shape
- connectivity (topology)
- other…

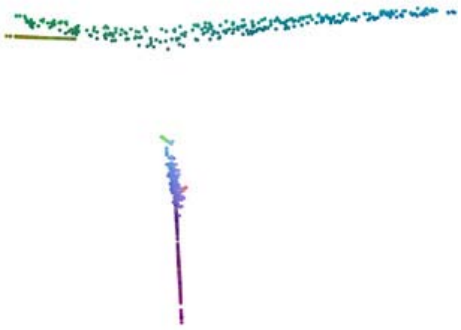






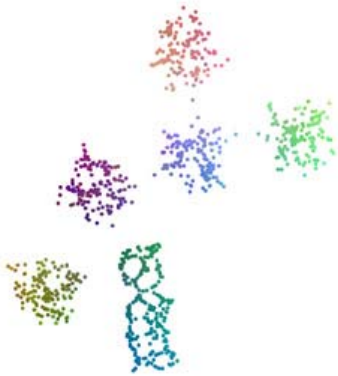
PCA

Isomap



LLE

SNE

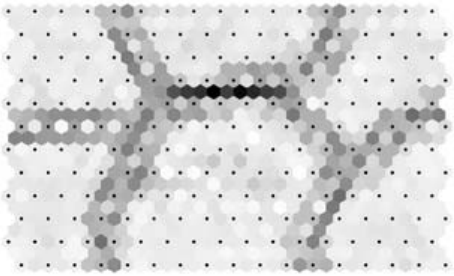


SNEG

Eigenmap

CCA

CDA



SOM

Venna et al. 2006, https://doi.org/10.1016/j.neunet.2006.05.014

Recap

- PCA and MDS variants will struggle with non-linear manifolds
- PCA/Torgerson scaling is a linear projection
- large distances dominate the cost function in MDS methods

- techniques specifically designed to flatten manifolds
  - ISOMAP
  - LLE
  - Laplacian eigenmap
  - local multidimensional scaling
  - many more exist...
- either redefine the distance or look only at the vicinity of individual points
- practical issues: distortions, may be computationally expensive

Literature on dimensionality reduction for visualisation

- MDS: Borg, Kroenen, Modern multidimensional scaling: theory and applications. Springer, 1997.
- PCA: any book on matrix algebra.
- Jarkko Venna 2007, Academic Dissertation, http://lib.tkk.fi/Diss/2007/isbn9789512287529/
- Lee & Verleysen, 2007. Nonlinear dimensionality reduction. Springer.
- For a more recent review see Verleysen & Lee, 2013 https://doi.org/10.1007/978-3-642-42054-2_77

- Also see the references in the slides! Notice that most doi.org links can be accessed from within Aalto network (but usually not from home).

Next lecture

- **Thu 1 April lecture is canceled**
- next (and last) lecture is on Thu 8 April

- Topics of the last lecture:

  - Student presentations (assignments 1 & 2)
  - Visualization of networks
  - Avoiding bullshit (i.e. showing the right data)