UCL Université catholique de Louvain

miro molecular imaging, radiotherapy & oncology

Unsupervised Dimensionality Reduction:

State of the Art, Quality Assessment, and Scalability

John A. Lee

L'Institut Agro Dijon, février 2026

How can we detect structure in data?

- Two main solutions
  - Visualize data directly
    (the user's eyes play a central part)
    - Data are left unchanged
    - Many views are proposed
    - Interactivity is inherent

    Examples:
    - Scatter plots
    - Projection pursuit
    - …

  - Derive new data prior to visualization
    (the software does a data processing job)
    - Data are appropriately modified
    - A single interesting representation is to be found

    → **(nonlinear) dimensionality reduction**

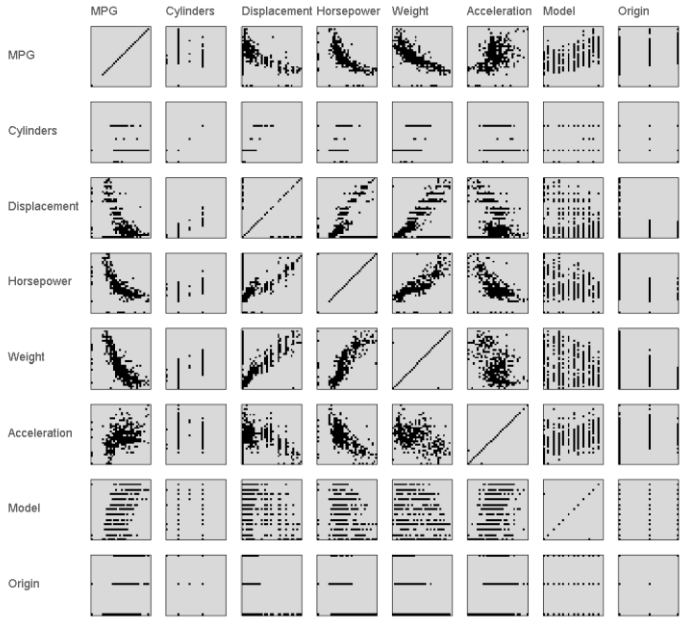
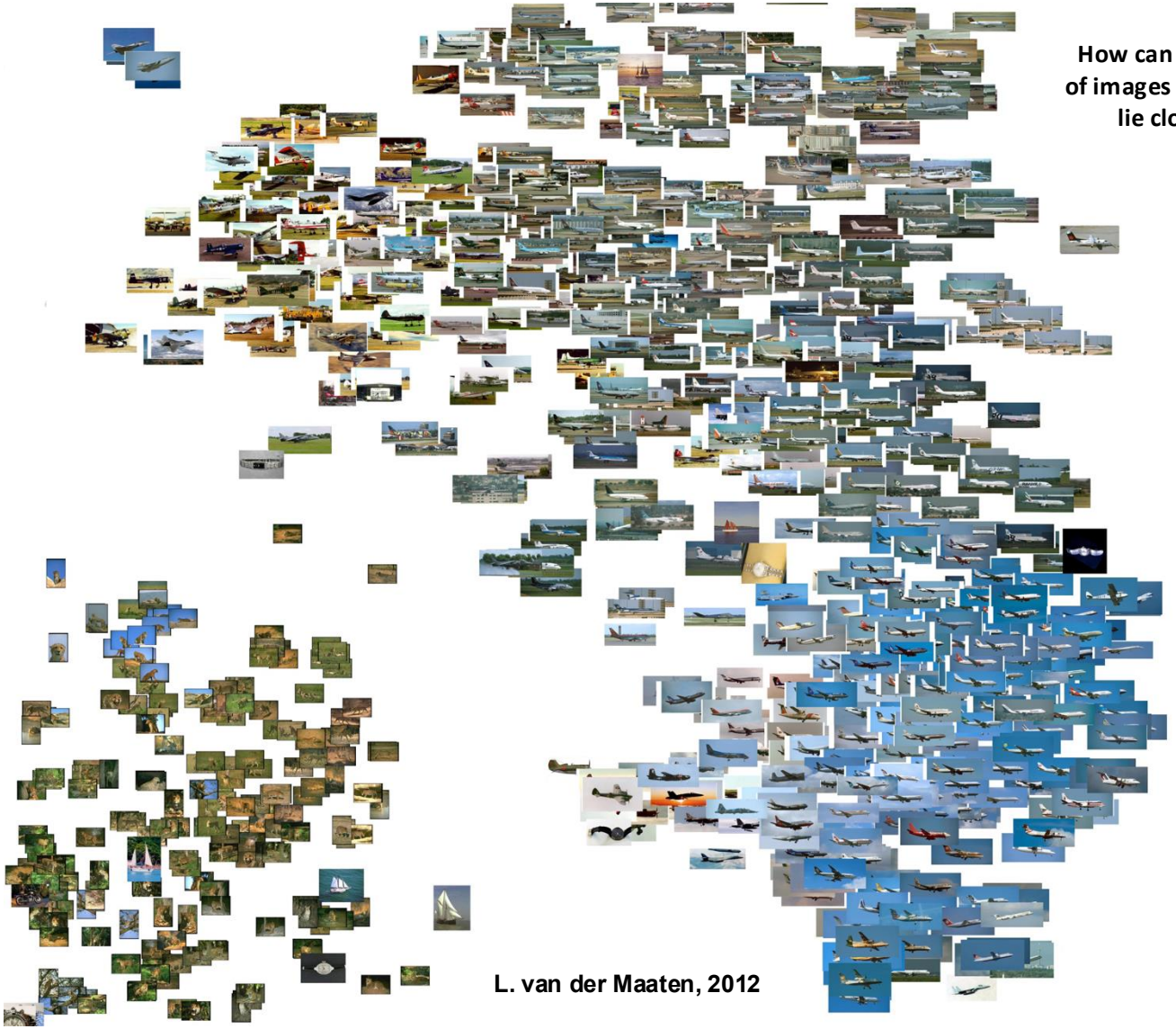


Image banks



How can we display big banks of images so that similar images lie close to each other?

L. van der Maaten, 2012

Image banks (zoom)

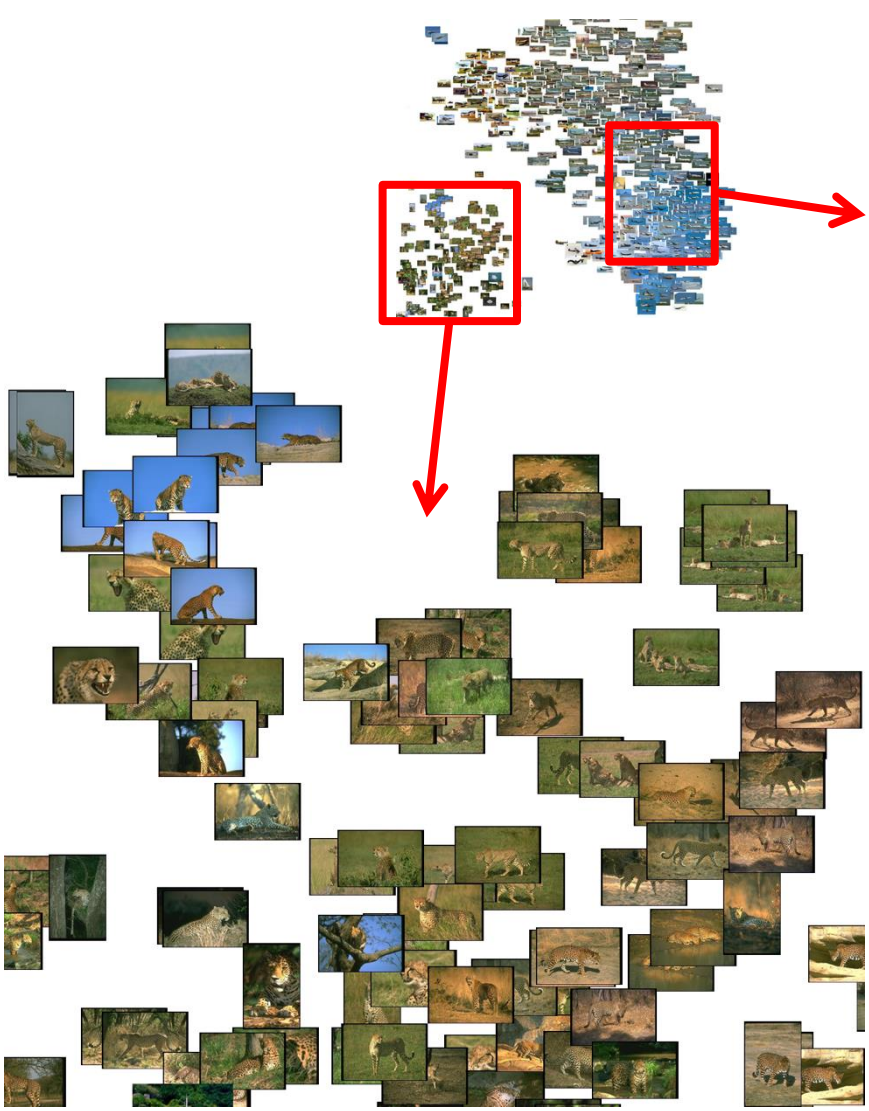
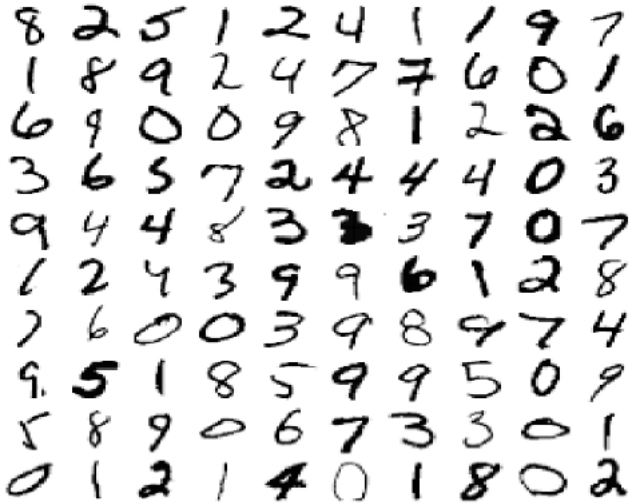


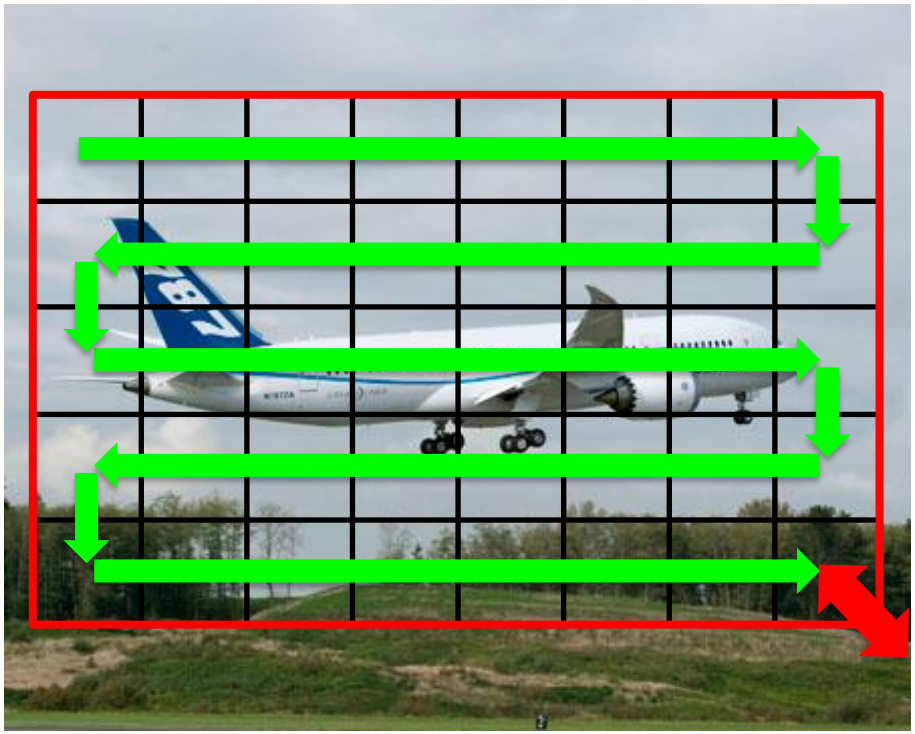
Image banks

COIL data set
(pictures of rotated objects)



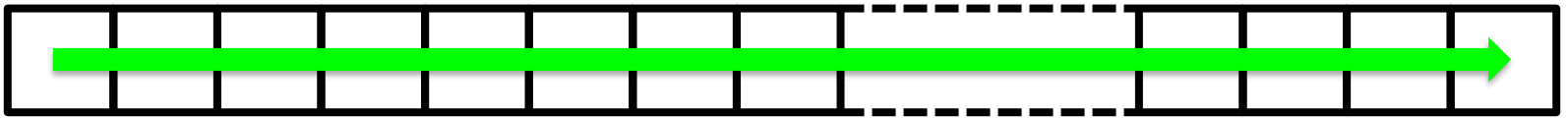
MNIST data set
(scanned handwritten digits)

How to encode images?



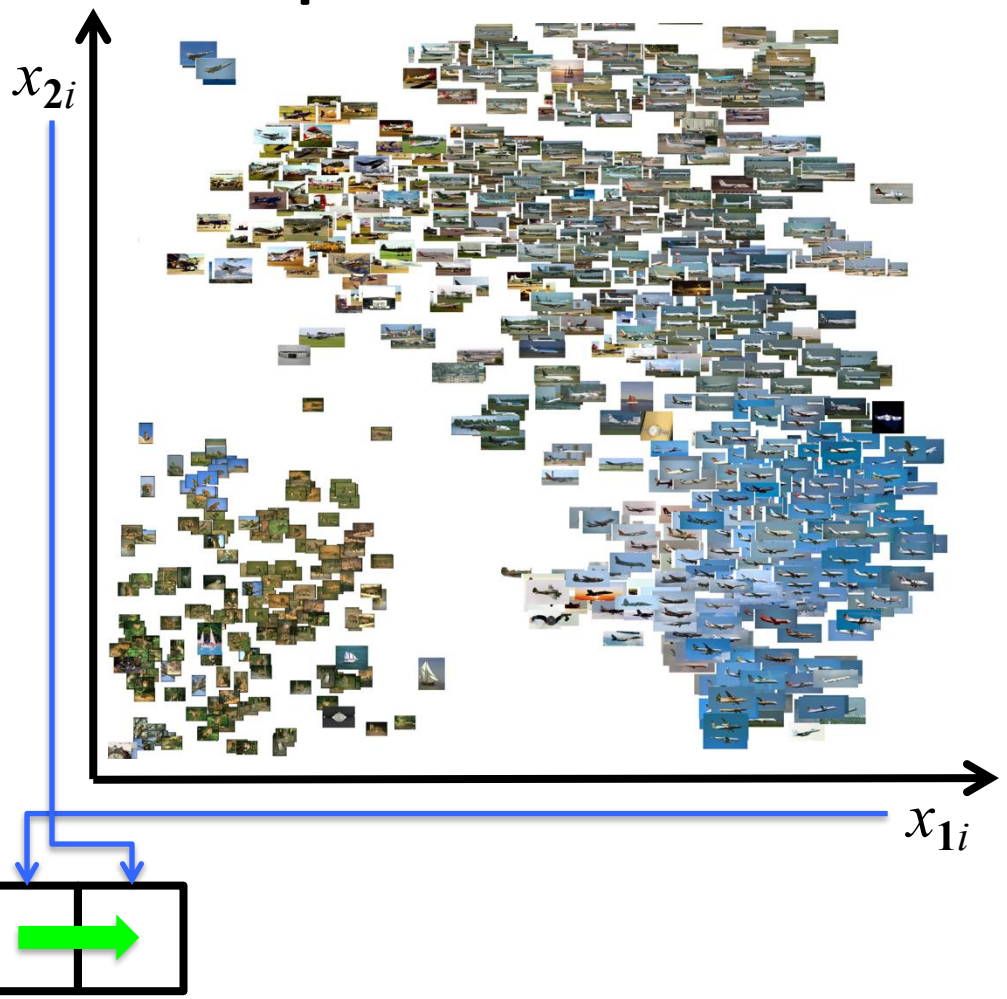
Features:

$\xi_i' = f(\xi_i)$



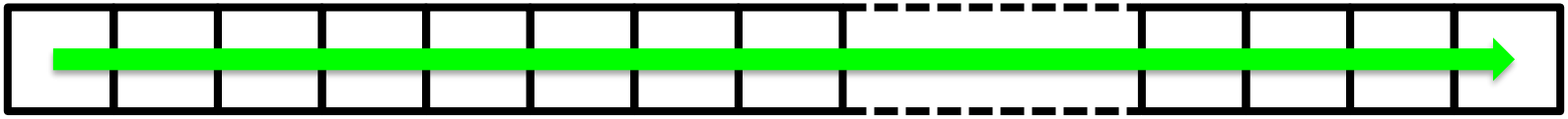
M-dimensional vectors: $\Xi = [\xi_i]_{1 \le i \le N}$

How to encode the representation?

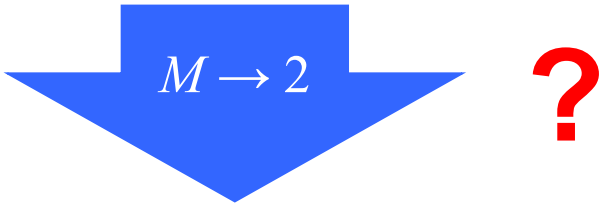


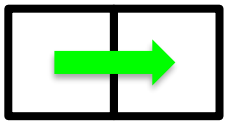
2-dimensional vectors: $\mathbf{X} = [\mathbf{x}_i]_{1 \leq i \leq N}$

From the image to the representation…



M-dimensional vectors: $\Xi = [\boldsymbol{\xi}_i]_{1 \leq i \leq N}$





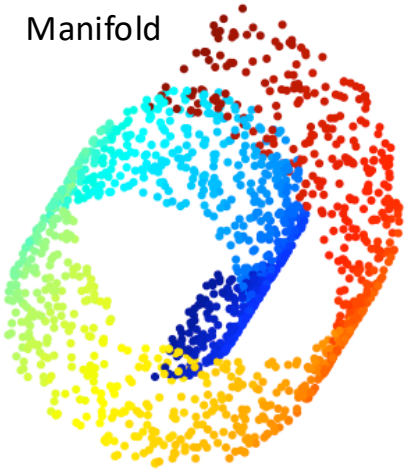
2-dimensional vectors: $\mathbf{X} = [\mathbf{x}_i]_{1 \leq i \leq N}$

Dimensionality reduction

(a.k.a. (NL)DR, manifold learning, embedding, projection, …)

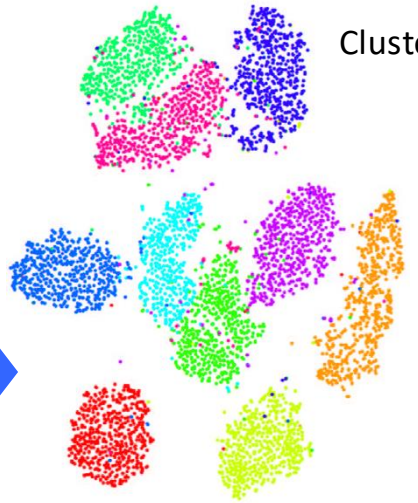
- The problem:
  - How can we detect structure in high-dimensional data, difficult to represent for human vision?
- The solutions:
  - Drop useless dimensions → Feature selection (not today's topic)
  - Find a simpler representation of data, with fewer dimensions → DR
- In practice:
  - DR can combine variables in a linear or nonlinear way (LDR/NLDR)
  - Meaningful data representation
    - Dissimilar items are represented far from each other
    - [Similar data items are represented close to each other]
  - Several possible hypotheses about the data distribution:

HD / LD	Near	Far
Near	☺	☹
Far	😐	☺



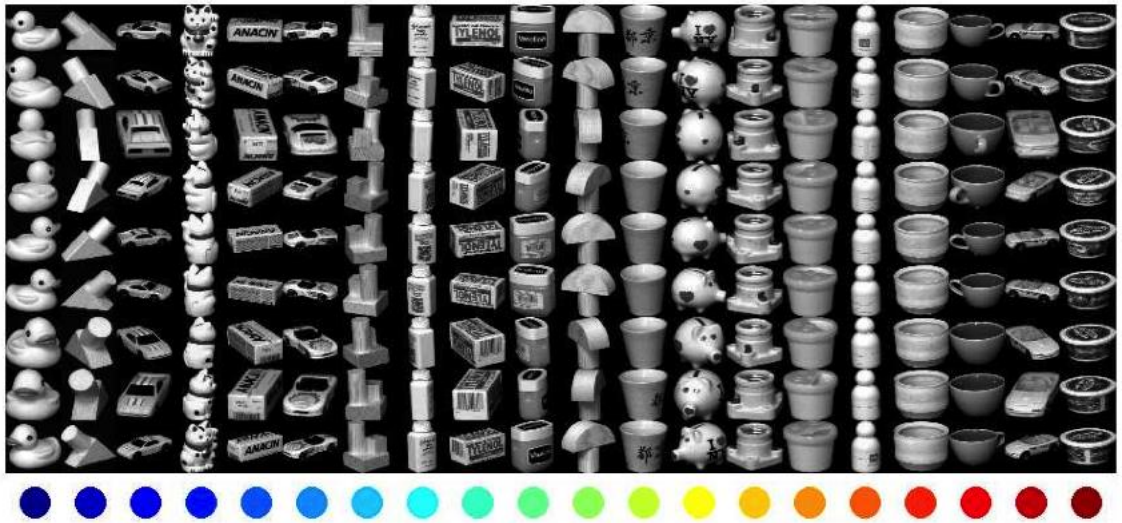
$$\Xi = [\xi_i]_{1 \leq i \leq N}$$

Probability distribution



High-dimensional data

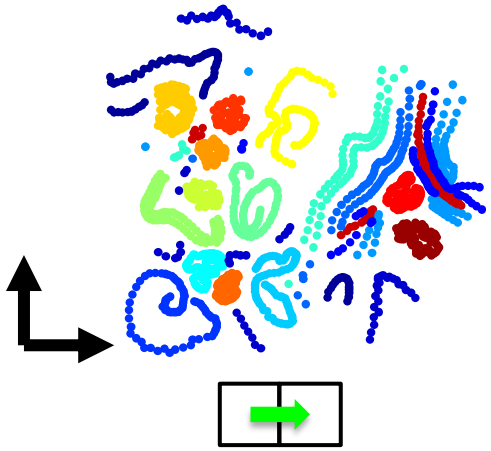
COIL-20 data set (1440 pictures of 20 rotated objects, 72 poses, every 5°)



Vectorised
128-by-128 images

$$\Xi = [\boldsymbol{\xi}_i]_{1 \le i \le N}$$

MD → 2D

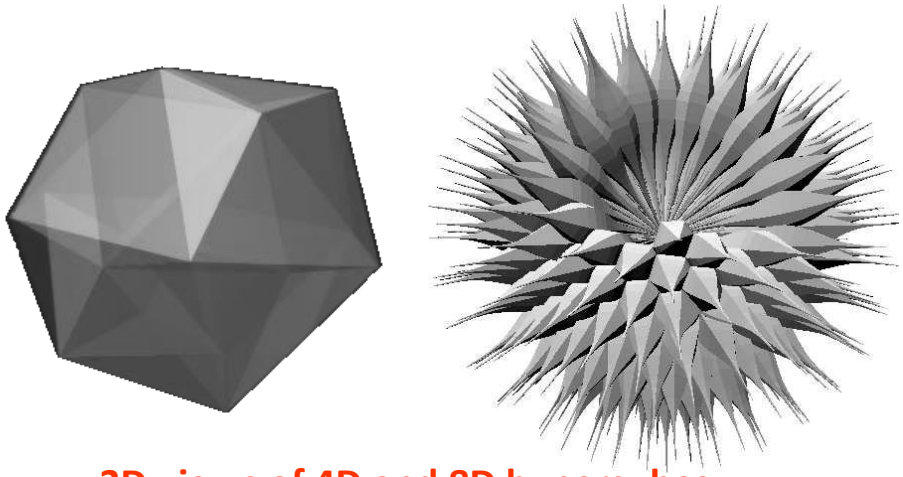


$$\mathbf{X} = [\mathbf{x}_i]_{1 \le i \le N}$$

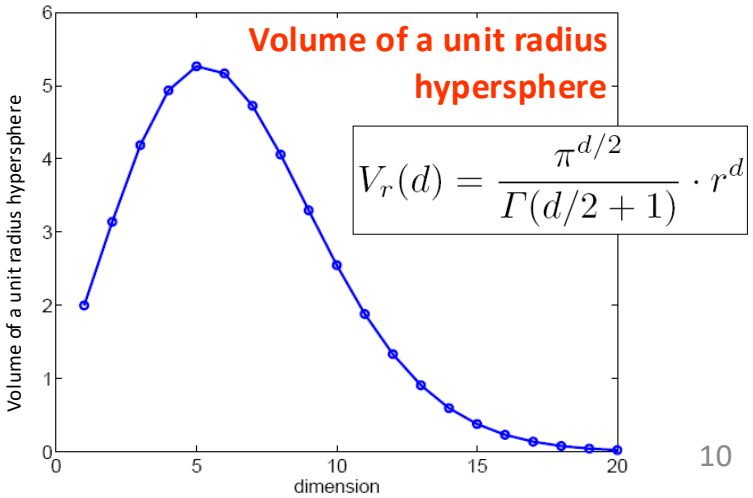
Dimensionality reduction

Why?

- The curse of dimensionality…
  - Empty space phenomenon
    (function approximation requires an exponential number of points w.r.t. *M*)
  - Norm concentration phenomenon
    (distances in a normal distribution have a chi distribution with *M* degrees of freedom)
- … and its unexpected consequences
  - A hypercube looks like a sea urchin (many spiky corners!)
  - Hypercube corners collapse towards the center in any projection
  - The volume of a unit hypersphere tends to zero
  - The sphere volume concentrates in a thin shell
  - Tails of a Gaussian get heavier than the central bell
  - Some points get very popular neighbors ('hubs')

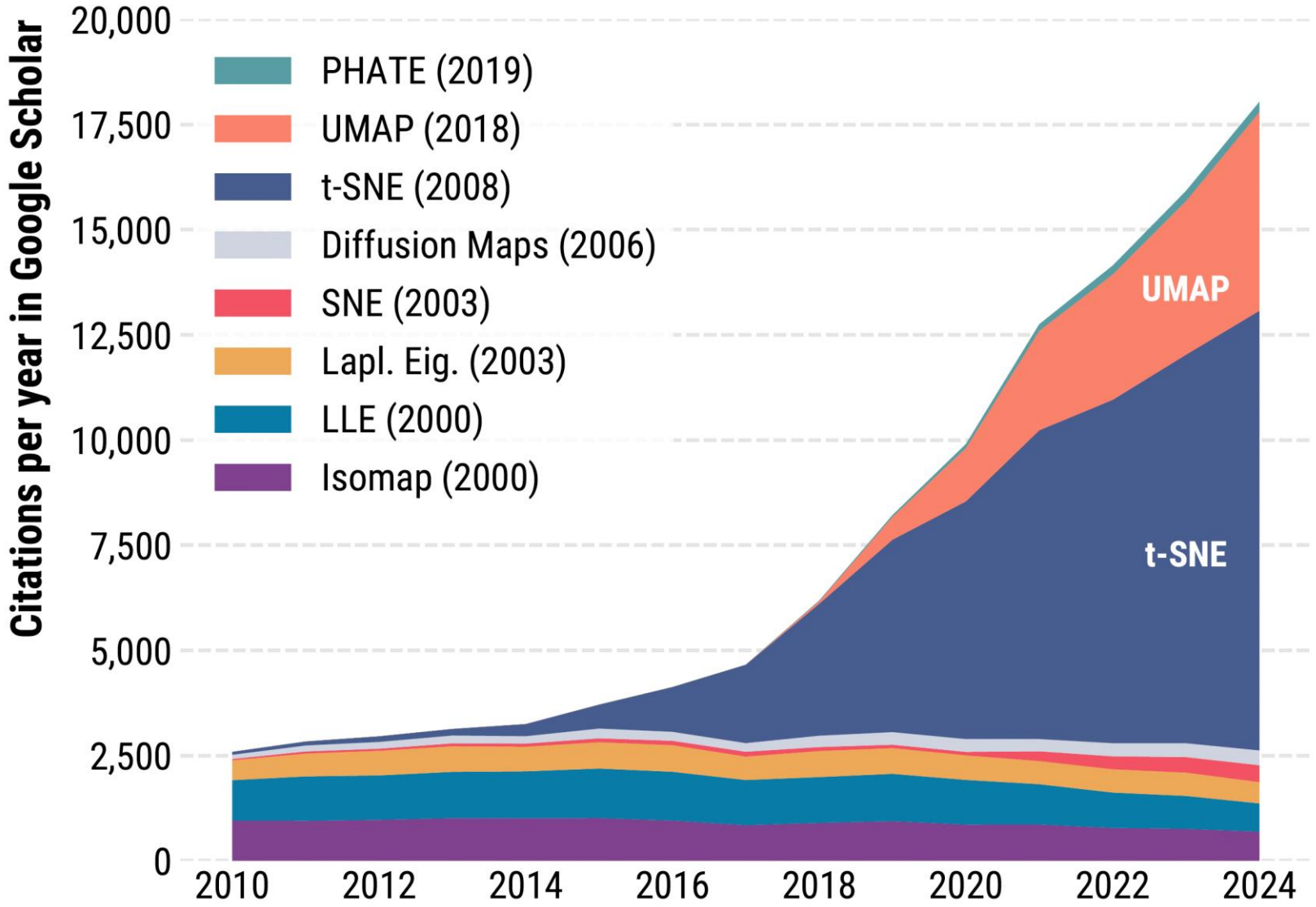


3D views of 4D and 8D hypercubes



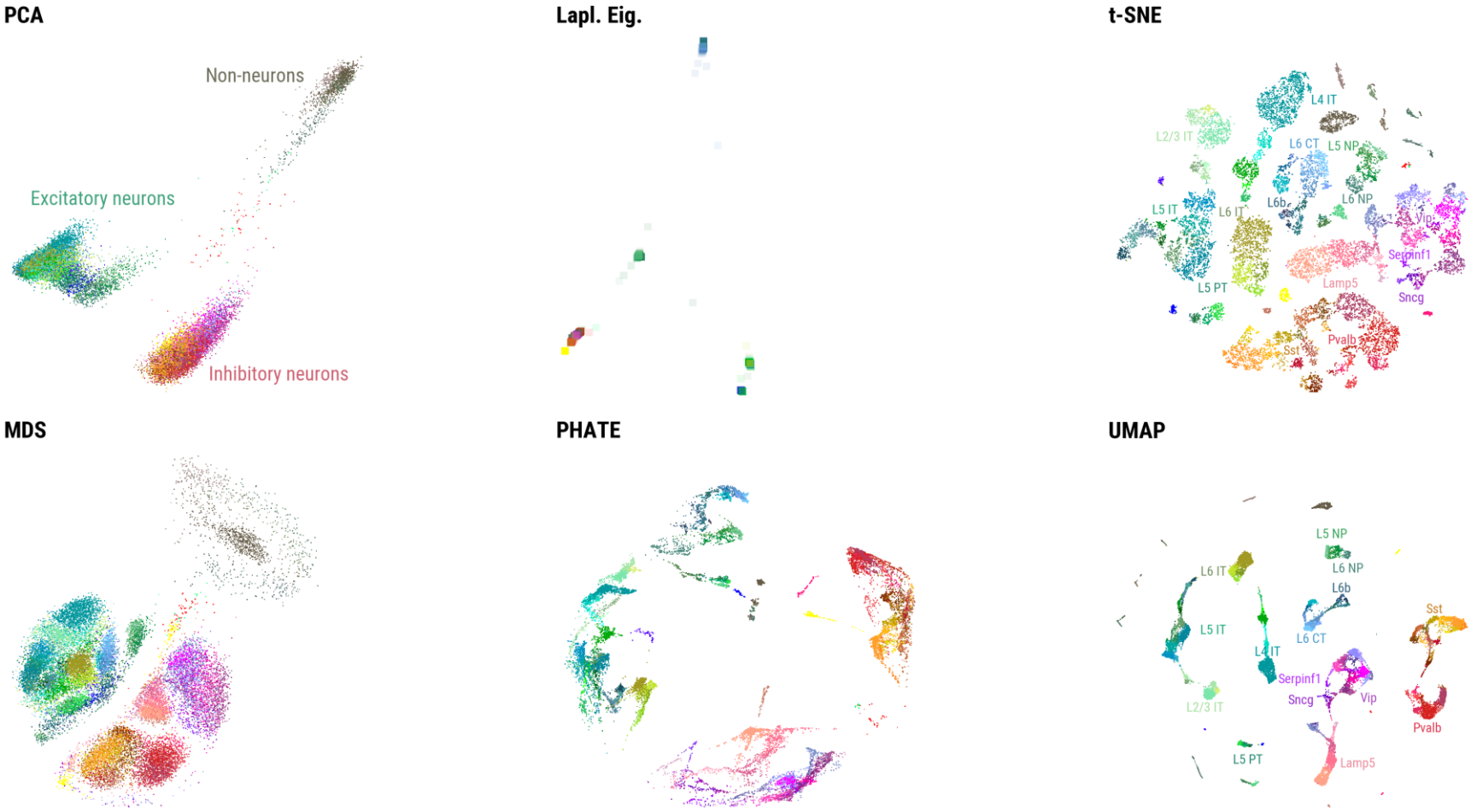
Dimensionality reduction

Why ***now***?



Dimensionality reduction

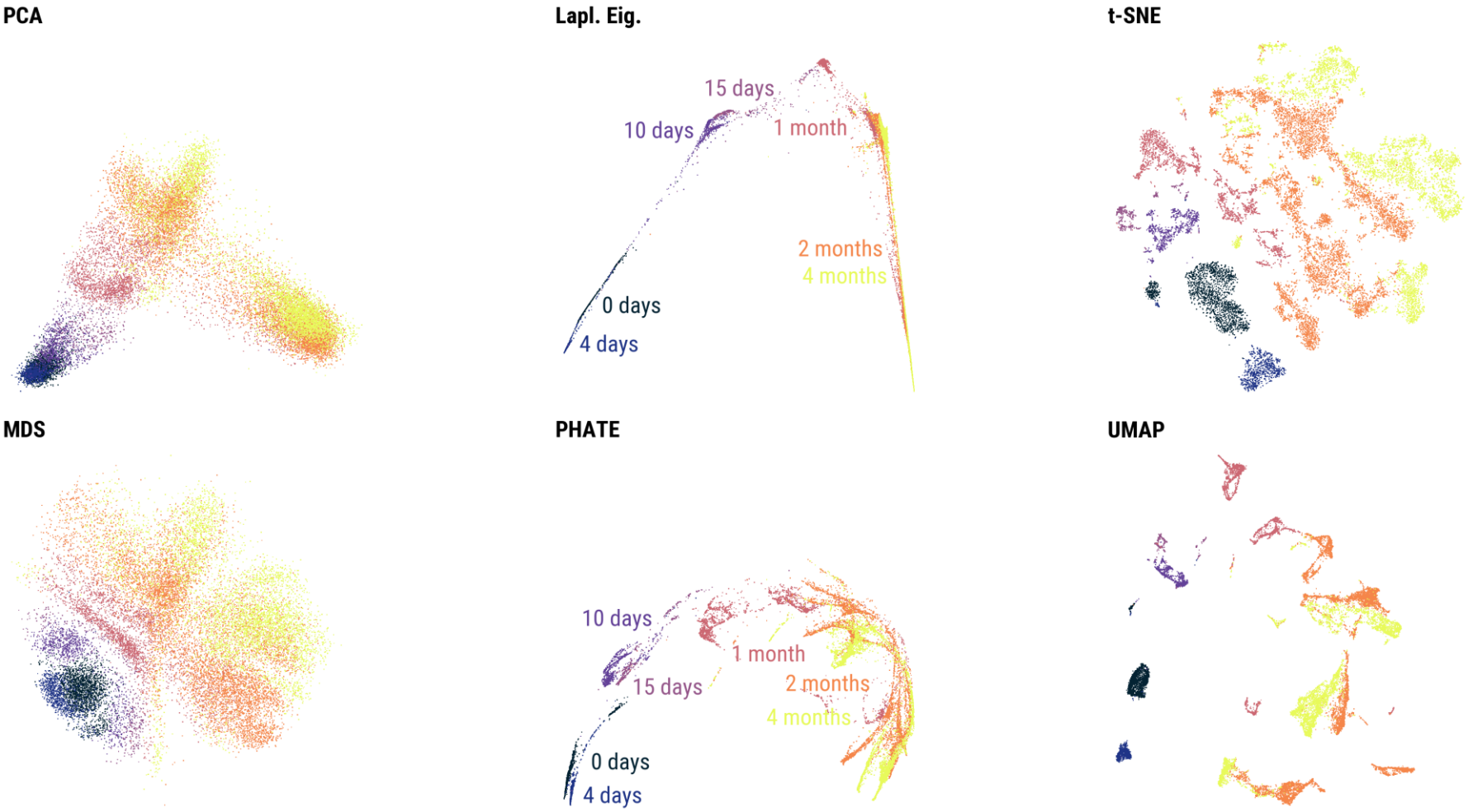
Why ***now***? Successful in applications…



2D embeddings of 23 800 cells from the mouse cortex (Tasic et al., 2018). Colors correspond to transcriptomic cell types, taken from the original publication. The first two principal components explained 49.1% of the variance of the preprocessed data. As Laplacian eigenmaps had many almost-overlapping points, they are shown with larger semi-transparent markers.

Dimensionality reduction

Why ***now***? Successful in applications…

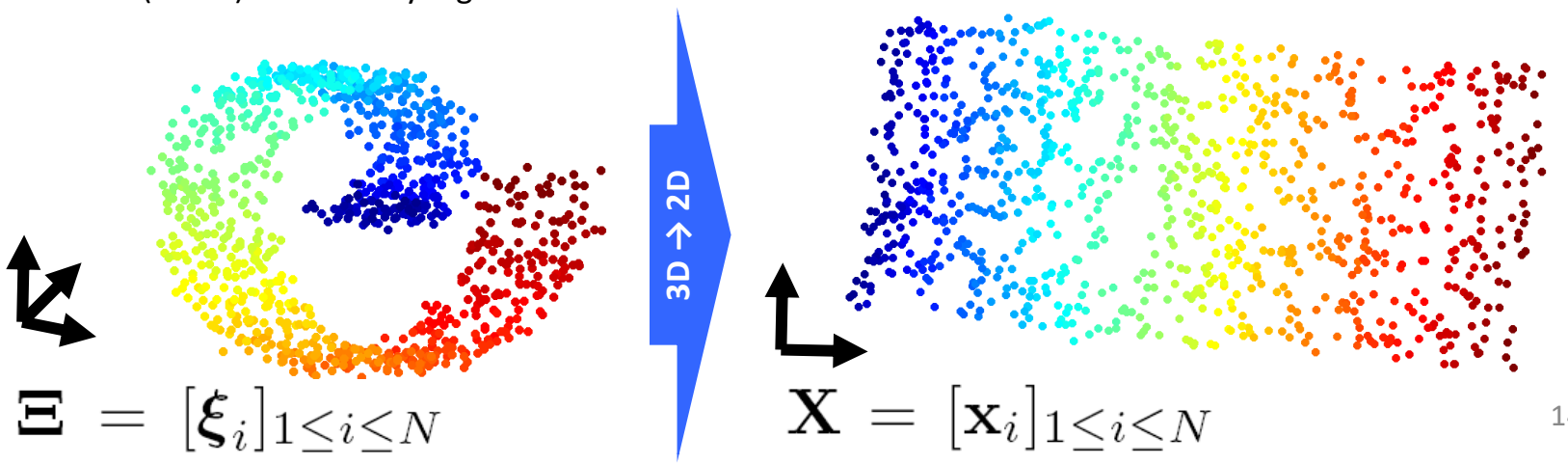


2D embeddings of 20 300 cells from primate brain organoids (Kanton et al., 2019). Colors correspond to sample age (from 0 to 120 days). The first two principal components explained 48.9% of the variance of the preprocessed data.

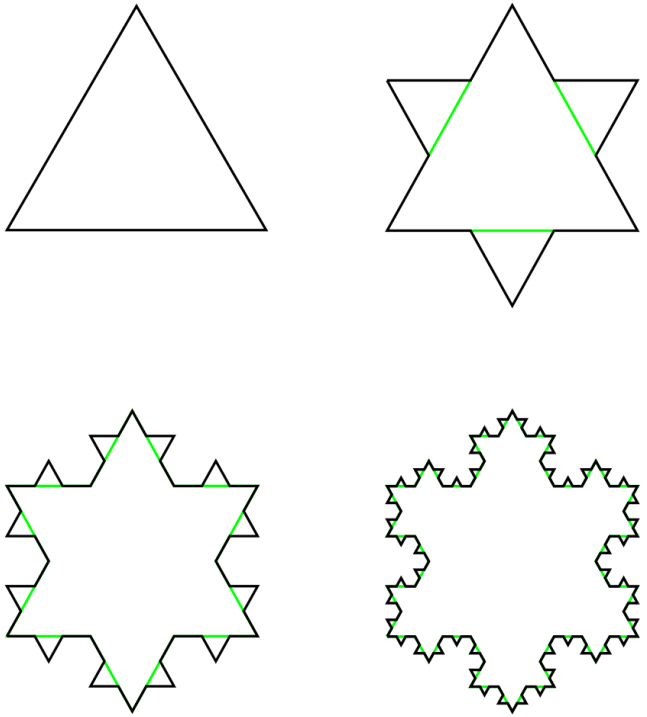
Dimensionality reduction

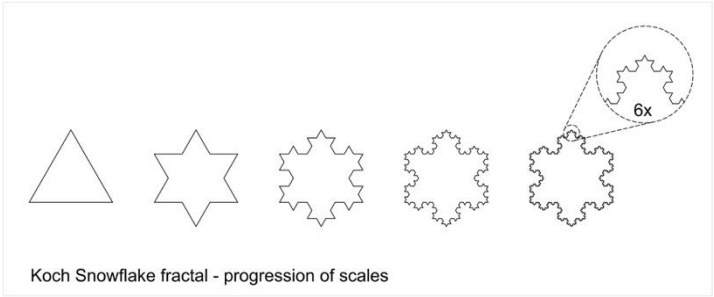
The manifold hypothesis

- The key idea:
  - Data live in a *M*-dimensional space
  - Data lie on some *P*-dimensional subspace → ***a smooth manifold***
- The manifold can be
  - A linear subspace
  - Any other function of some latent variables
- Dimensionality reduction aims at
  - Inverting the latent variable mapping → ***visualisation in the latent space***
  - Unfolding the manifold (topology allows us to 'deform' it)
- A noise model can come on top of this to give a probabilistic flavour
- Two problems show up:
  - How can we estimate *P*? → ***intrinsic/fractal dimension***
  - *M* (and *P*) can be very high…



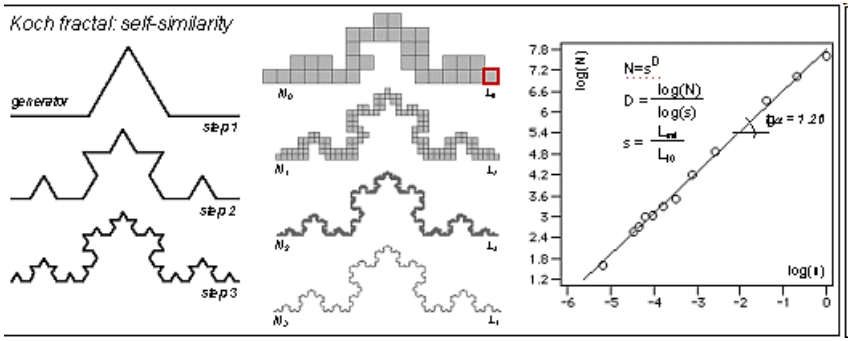
Estimator of the intrinsic dimensionality

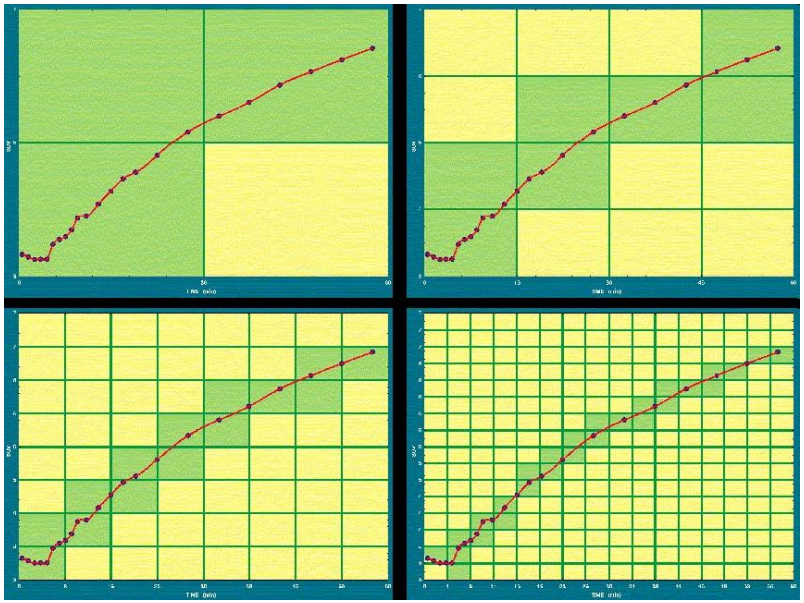




Estimator of the intrinsic dimensionality

- General idea: estimate the fractal dimension
- Box counting (or capacity dimension)
  - Create bins of width ε along each dimension (e.g. $\varepsilon = 1/2^i$)
  - Data sampled on a *P*-dimensional manifold occupy $N(\varepsilon) \approx \alpha\ \varepsilon^{-P}$ boxes
  - Compute the slope in a log-log diagram of $N(\varepsilon)$ w.r.t. ε
  - Simple but
    - Subjective method (slope estimation at some scale)
    - Not robust againt noise
    - Computationally expensive

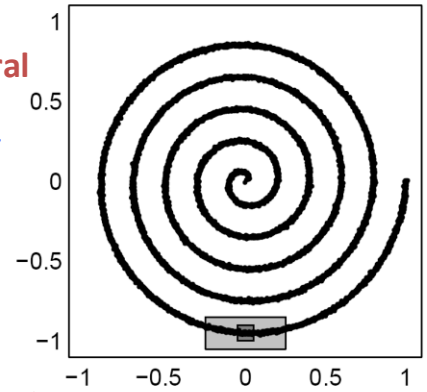




Estimator of the intrinsic dimensionality

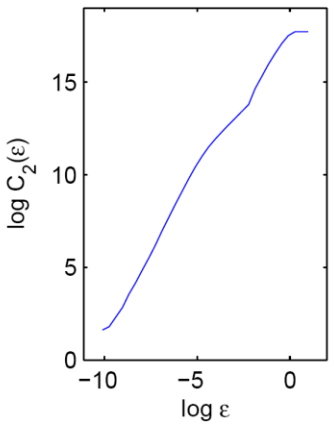
- Correlation dimension
  - Any datum of a *P*-dimensional manifold is surrounded by $C_2(\varepsilon) \approx \alpha\, \varepsilon^P$ neighbours, where ε is a small neighborhood radius
  - Compute the slope of the correlation sum in a log-log diagram

Noisy spiral

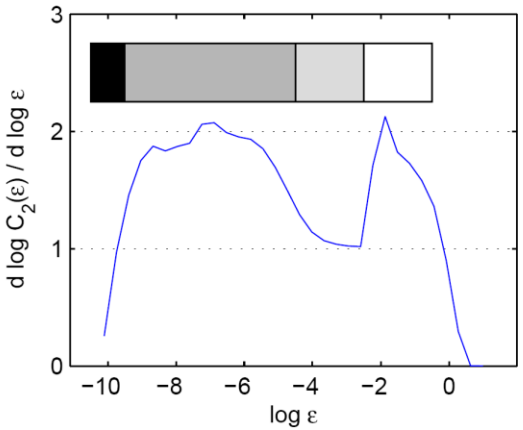


$$C_2(\varepsilon) \approx \Sigma_i \, |\{ j : ||\xi_i - \xi_j||_2 \leq \varepsilon \}| \, / \, N$$

Log-log plot of correlation sum



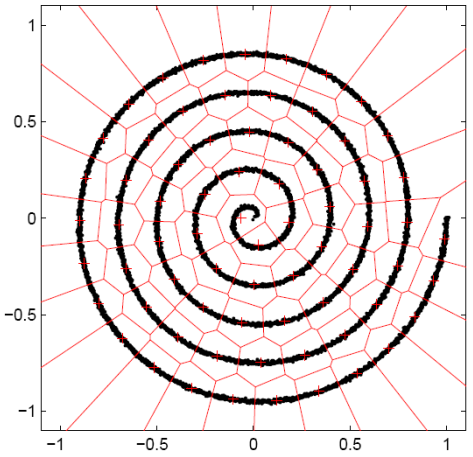
Slope ≈ int.dim.



.

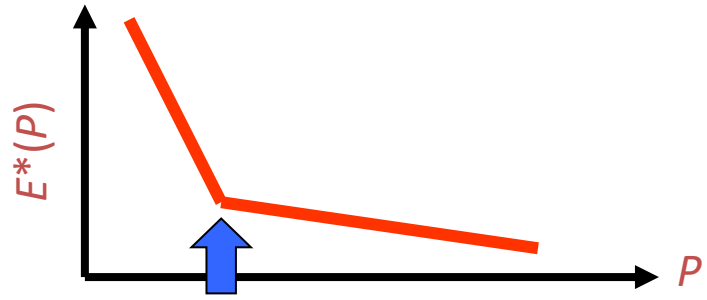
Estimator of the intrinsic dimensionality

- Other techniques
  - Local PCAs
    - Split manifold into small patch (on a well chosen scale…)
    - Manifold is locally linear
      ⟶ Apply PCA on each patch



  - Trial-and-error:
    - Pick an appropriate DR method
    - Run it for $P = 1, \ldots, M$ and record the value $E^*(P)$ of the cost function after optimisation
    - Draw the curve $E^*(P)$ w.r.t. P and detect its elbow

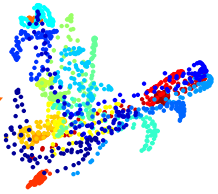
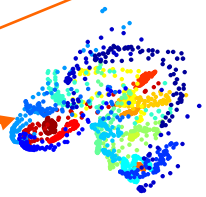
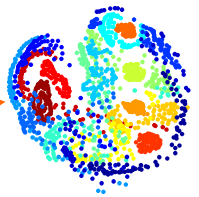
.

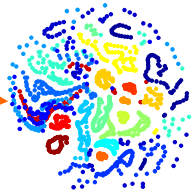
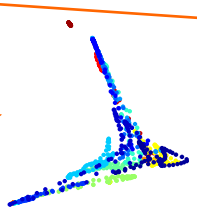


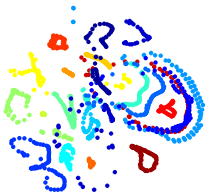
Computationally demanding!

(NL)DR through time…

1901 Principal component analysis (PCA)
1938 Classical multidimensional scaling (CMDS)
1962 Nonmetric MDS (NMDS)
1969 Sammon's nonlinear mapping (NLM)
1982 Self-organising maps (SOMs)
1991 Auto-encoder (back prop.)
1993 Curvilinear component analysis (CCA)
1996 Kernel PCA
1998 Isomap
2000 Locally linear embedding (LLE)
2002 Laplacian eigenmaps (LE)
2002 Stochastic neighbour embedding (SNE)
2006 Auto-encoder (deep learning)
2008 Student-distributed SNE (*t*-SNE)
2010 Neighbour retrieval & vis. (NeRV)
2012 Jensen-Shannon Embedding (JSE)
2014 Multiscale JSE (Ms JSE)
2018 UMAP, *tt*-SNE, Ms *t*-SNE
2019 FI*t*-SNE, NE with missing data
2022 Fast Multiscale NE

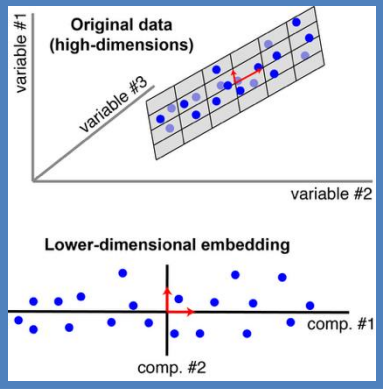
Spectral
Neural network
Reconstruction error
Distance preservation
Similarity preservation

DR as a ‘four quarter’ cake

An early cheat sheet as a quick summary…

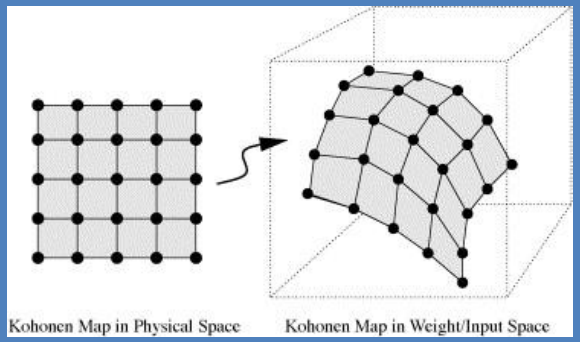
PCA

Rigid plane fitting
Preservation of variance



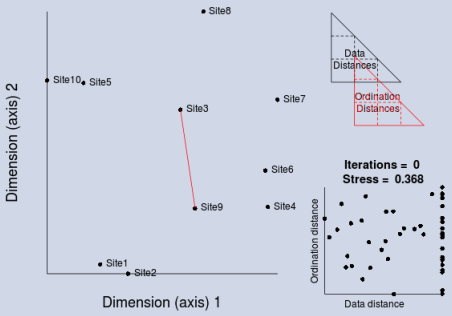
SOM

Articulated grid fitting



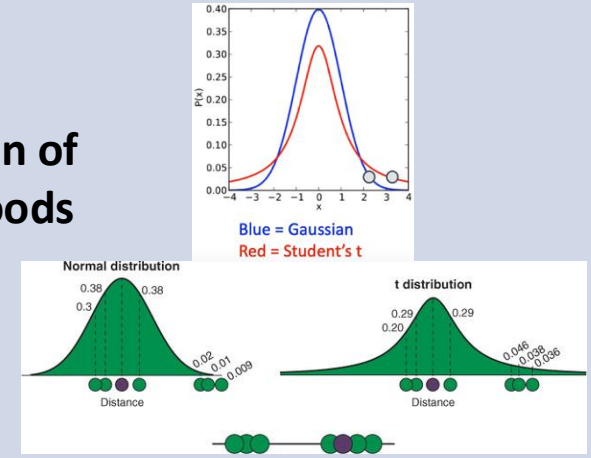
MDS

Preservation of scalar products or distances



(*t*-)SNE

Preservation of neighborhoods



A technical slide… (some reminders)

Norm and positive semidefinitedness

- The norm of matrix $\mathbf{A} = [a_{ij}]_{ij}$ is defined as $\|\mathbf{A}\|_2^2 = \sum_{i,j} a_{ij}^2 = \mathrm{Tr}(\mathbf{A}^T\mathbf{A}) = \mathrm{Tr}(\mathbf{A}\mathbf{A}^T)$
- Square matrix $\mathbf{B}$ is positive semidefinite iff $\mathbf{x}^T\mathbf{B}\mathbf{x} \geq 0$ for all $\mathbf{x}$
- Matrix $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}\mathbf{A}^T$ are positive semidefinite: $\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} = \|\mathbf{A}\mathbf{x}\|_2^2 \geq 0$ and $\mathbf{y}^T\mathbf{A}\mathbf{A}^T\mathbf{y} = \|\mathbf{A}^T\mathbf{y}\|_2^2 \geq 0$

Covariance and Gram matrices

- Dataset $\mathbf{\Xi} = [\boldsymbol{\xi}_i]_{1 \leq i \leq N}$ gets centered in $\mathbf{\Xi} - \frac{1}{N}\mathbf{\Xi}\mathbf{1}\mathbf{1}^T = \mathbf{\Xi}\mathbf{C}$, where $\mathbf{C} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T = \mathbf{C}^T$ is the $N \times N$ centering matrix
- The covariance matrix of $\mathbf{\Xi}$ is defined as $\mathbf{\Sigma}_{\mathbf{\Xi}\mathbf{\Xi}} = \frac{1}{N}\mathbf{\Xi}\mathbf{C}\mathbf{C}^T\mathbf{\Xi}^T$; it reduces to $\frac{1}{N}\mathbf{\Xi}\mathbf{\Xi}^T$ if data need not be centered (i.e. $\mathbf{\Xi}\mathbf{1} = \mathbf{0}$)
- The Gram matrix is defined as $\mathbf{G} = \mathbf{\Xi}^T\mathbf{\Xi}$; for centered data, it is given by $\mathbf{C}\mathbf{\Xi}^T\mathbf{\Xi}\mathbf{C}$
- The covariance and the Gram matrices are symmetric and positive semidefinite

Yet another bad guy…

Spectral decompositions

- Square real matrix $\mathbf{U}$ is orthogonal iff $\mathbf{U}^T = \mathbf{U}^{-1}$
- The singular value decomposition of real matrix $\mathbf{A}$ is written as $\mathbf{A} = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T$, where both $\mathbf{V}$ and $\mathbf{U}$ are real and orthogonal, and $\mathbf{\Sigma}$ is real with $\sigma_{ij} = 0$ for $i \neq j$ and $\sigma_{11} \geq \sigma_{22} \geq \ldots \geq 0$
- The eigenvalue decomposition of symmetric matrix $\mathbf{B}$ can be written as $\mathbf{B} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$, where $\mathbf{V}$ is real and orthogonal, and $\mathbf{\Lambda}$ is real and diagonal, with $\lambda_{11} \geq \lambda_{22} \geq \ldots$ and $\mathrm{Tr}(\mathbf{B}) = \mathrm{Tr}(\mathbf{\Lambda})$
- If $\mathbf{B}$ is symmetric and positive semidefinite, then $\lambda_{ii} \geq 0$
- The SVD of $\mathbf{A}$ allows us to relate the EVDs of $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$; we have $\mathbf{A}\mathbf{A}^T = \mathbf{V}\mathbf{\Sigma}\mathbf{\Sigma}^T\mathbf{V}^T$ and $\mathbf{A}^T\mathbf{A} = \mathbf{U}\mathbf{\Sigma}^T\mathbf{\Sigma}\mathbf{U}^T$, showing that $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$ have the same eigenvalues

Jamais deux sans trois (never 2 w/o 3)

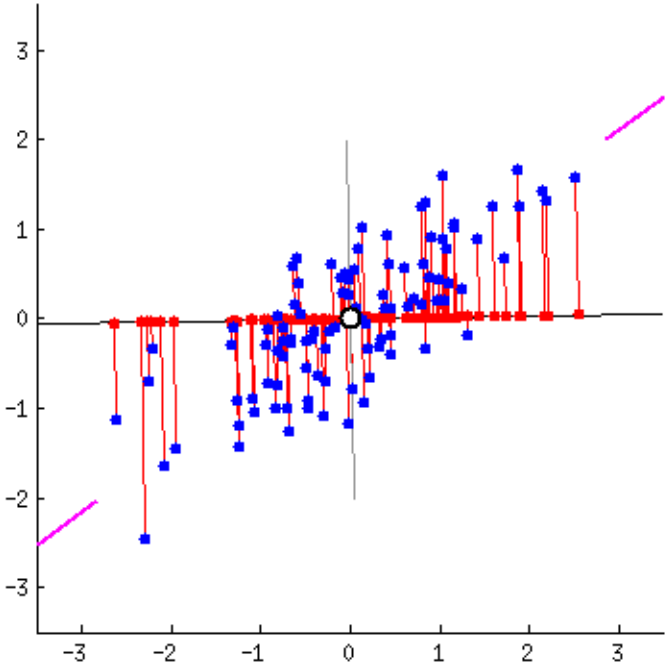
Spectral decompositions and optimization

- Singular value decomposition $\mathbf{A} = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T$ solves
  $\arg\min_X \|\mathbf{A} - \mathbf{X}\|_2^2$ s.t. $\mathrm{rank}(\mathbf{X}) = P \leq \mathrm{rank}(\mathbf{A})$;
  the solution is $\mathbf{X} = \sum_{i=1}^{P} \mathbf{v}_i \sigma_{ii} \mathbf{u}_i^T$
  (the smallest singular values are discarded)
- Eigenvalue decomposition $\mathbf{B} = \mathbf{B}^T = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ solves
  $\arg\min_X \|\mathbf{B} - \mathbf{X}^T\mathbf{X}\|_2^2$ s.t. $\mathrm{rank}(\mathbf{X}) = P \leq \mathrm{rank}(\mathbf{B})$;
  the solution is $\mathbf{X} = \left[\sqrt{\lambda_{ii}}\mathbf{v}_i\right]_{1 \leq i \leq P}^T$
  (the eigenvalues with the smallest magnitude are discarded)

Principal component analysis

- Pearson, 1901; Hotelling, 1933; Karhunen, 1946; Loève, 1948.
- Idea
  - Decorrelate zero-mean data by placing a rotated coordinate system
  - Keep large variance axes

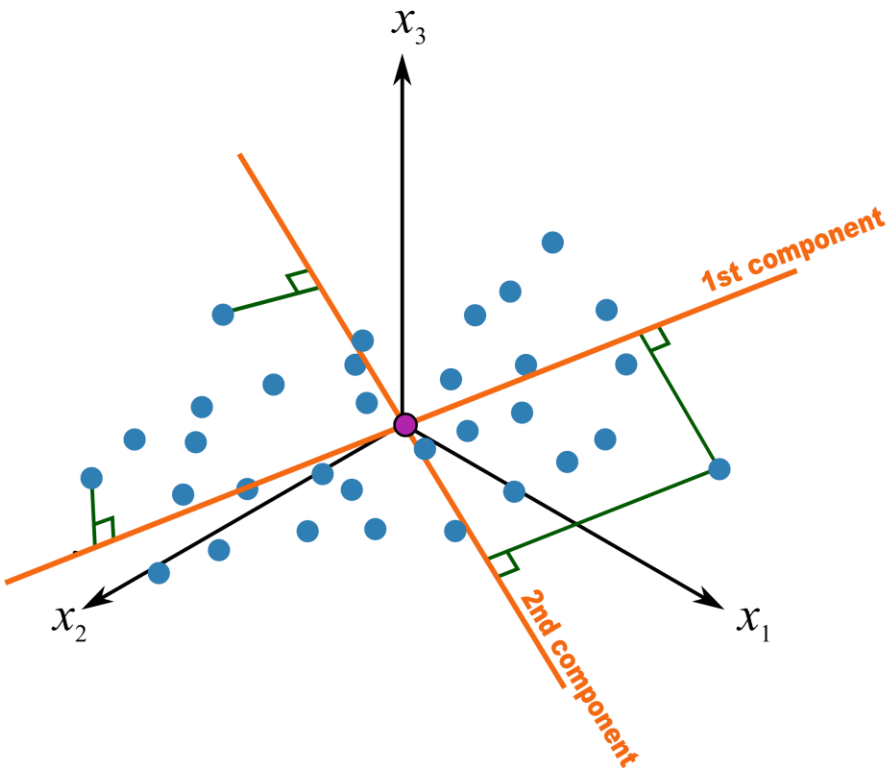
  → Fit a line/(hyper)plane (the coordinate system) through the data cloud and project on it

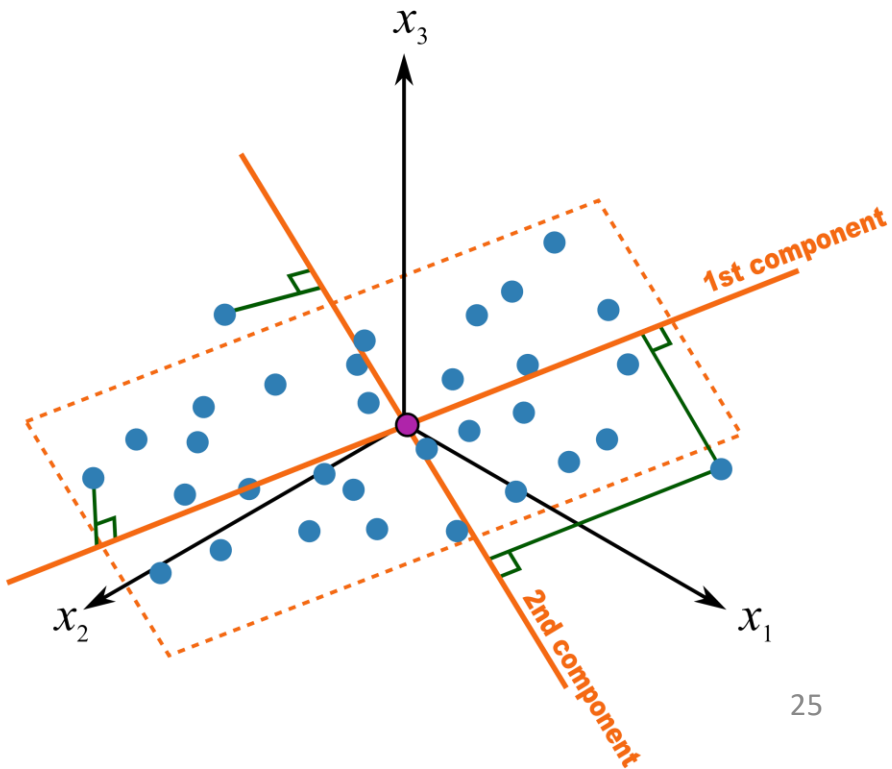


Principal component analysis

- Pearson, 1901; Hotelling, 1933; Karhunen, 1946; Loève, 1948.
- Idea
  - Decorrelate zero-mean data by placing a rotated coordinate system
  - Keep large variance axes

  → Fit a line/(hyper)plane (the coordinate system) through the data cloud and project on it
- Details (maximise projected variance)





Principal component analysis

- Pearson, 1901; Hotelling, 1933; Karhunen, 1946; Loève, 1948.
- Idea
  - Decorrelate zero-mean data by placing a rotated coordinate system
  - Keep large variance axes

  → Fit a line/(hyper)plane (the coordinate system) through the data cloud and project on it
- Details (maximise projected variance)

Model: $\mathbf{x} = \mathbf{W}^T \boldsymbol{\xi}$ with

- $\mathrm{E}\{\boldsymbol{\xi}\} \approx \frac{1}{N}\boldsymbol{\Xi}\mathbf{1} = \mathbf{0}$ and $\mathrm{E}\{\mathbf{x}\} \approx \frac{1}{N}\mathbf{X}\mathbf{1} = \mathbf{0}$
- $\mathbf{W}$ has P orthonormal columns

The covariance of $\mathbf{x}$ is $\mathrm{E}\{\mathbf{x}\mathbf{x}^T\} \approx \mathbf{C}_{\mathbf{X}\mathbf{X}} = \frac{1}{N}\mathbf{X}\mathbf{X}^T = \mathbf{W}^T\mathbf{C}_{\boldsymbol{\Xi}\boldsymbol{\Xi}}\mathbf{W}$
where $\mathbf{C}_{\boldsymbol{\Xi}\boldsymbol{\Xi}} = \frac{1}{N}\boldsymbol{\Xi}\boldsymbol{\Xi}^T$ is positive semidefinite
In order to maximise $E(\mathbf{W};\boldsymbol{\Xi}) = \mathrm{Tr}\,(\mathbf{C}_{\mathbf{X}\mathbf{X}})$, we use the eigenvalue decomposition $\mathbf{C}_{\boldsymbol{\Xi}\boldsymbol{\Xi}} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$ with the following properties:

- $\mathbf{V}^T\mathbf{V} = \mathbf{I}$
- $\boldsymbol{\Lambda} = \mathrm{diag}(\boldsymbol{\lambda})$, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_D \geq 0$
- $\mathbf{C}_{\mathbf{X}\mathbf{X}}$ is diagonal

and $E(\mathbf{W};\boldsymbol{\Xi}) = \frac{1}{N}\,\mathrm{Tr}\left(\mathbf{W}^T\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T\mathbf{W}\right)$ is maximal iff $\mathbf{W} = \mathbf{V}\mathbf{I}_{D\times P}$

Principal component analysis

- Details (minimise the reconstruction error)

Model: $\mathbf{X} = \mathbf{W}^T(\mathbf{\Xi} - \boldsymbol{\tau}\mathbf{1}^T)$ s.t. $\mathbf{W}$ has P columns $(\mathbf{W}^T\mathbf{W} = \mathbf{I})$
The reconstruction error is

$$\begin{aligned} E(\mathbf{W}, \boldsymbol{\tau}; \mathbf{\Xi}) &= \|\mathbf{\Xi} - (\mathbf{W}\mathbf{X} + \boldsymbol{\tau}\mathbf{1}^T)\|_2^2 \\ &= \mathrm{Tr}\left((\mathbf{\Xi} - \boldsymbol{\tau}\mathbf{1}^T)^T(\mathbf{I} - \mathbf{W}\mathbf{W}^T)^T(\mathbf{I} - \mathbf{W}\mathbf{W}^T)(\mathbf{\Xi} - \boldsymbol{\tau}\mathbf{1}^T)\right) \\ &= \mathrm{Tr}\left((\mathbf{\Xi} - \boldsymbol{\tau}\mathbf{1}^T)^T(\mathbf{I} - \mathbf{W}\mathbf{W}^T)(\mathbf{\Xi} - \boldsymbol{\tau}\mathbf{1}^T)\right) \end{aligned}$$

Step 1: determine $\boldsymbol{\tau}$

$$\frac{\partial E(\mathbf{W}, \boldsymbol{\tau}; \mathbf{\Xi})}{\partial \boldsymbol{\tau}} = 2(\mathbf{I} - \mathbf{W}\mathbf{W}^T)(\mathbf{\Xi} - \boldsymbol{\tau}\mathbf{1}^T)\mathbf{1} = \mathbf{0} \quad \Rightarrow \quad \boldsymbol{\tau} = \frac{1}{N}\mathbf{\Xi}\mathbf{1}$$

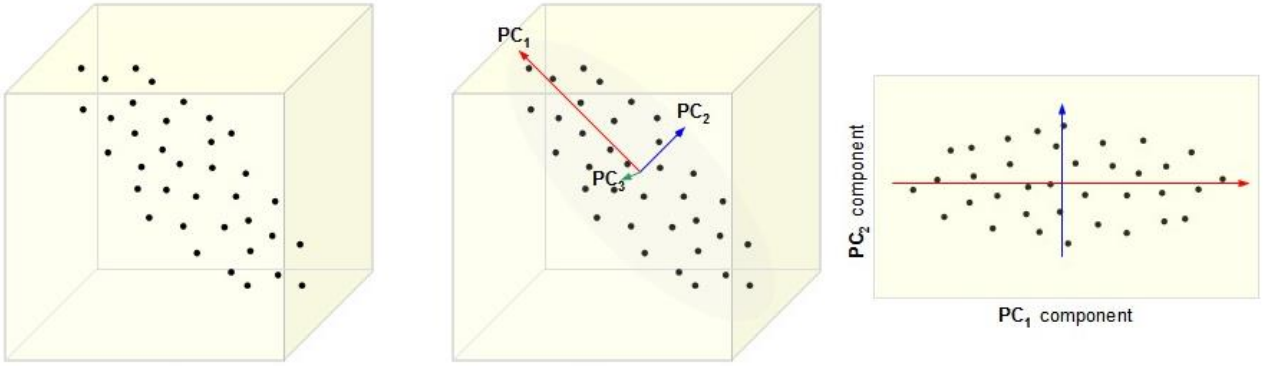
Step 2: determine $\mathbf{W}$ with simplified model $\mathbf{X} = \mathbf{W}^T\mathbf{\Xi}$ s.t. $\mathbf{\Xi}\mathbf{1} = \mathbf{0}$

$$\begin{aligned} E(\mathbf{W}; \mathbf{\Xi}) &= \mathrm{Tr}(\mathbf{\Xi}^T(\mathbf{I} - \mathbf{W}\mathbf{W}^T)\mathbf{\Xi}) = \mathrm{Tr}(\mathbf{\Xi}^T\mathbf{\Xi}) - \mathrm{Tr}(\mathbf{\Xi}^T\mathbf{W}\mathbf{W}^T\mathbf{\Xi}) \\ \arg\min_{\mathbf{W}} E(\mathbf{W}; \mathbf{\Xi}) &= \arg\max_{\mathbf{W}} \mathrm{Tr}(\mathbf{\Xi}^T\mathbf{W}\mathbf{W}^T\mathbf{\Xi}) \\ &= \arg\max_{\mathbf{W}} \mathrm{Tr}(\mathbf{W}^T\mathbf{\Xi}\mathbf{\Xi}^T\mathbf{W}) \\ &= \arg\max_{\mathbf{W}} \mathrm{Tr}(\mathbf{W}^T\mathbf{C}_{\mathbf{\Xi}\mathbf{\Xi}}\mathbf{W}) \end{aligned}$$

(We come back to the same problem)

Principal component analysis

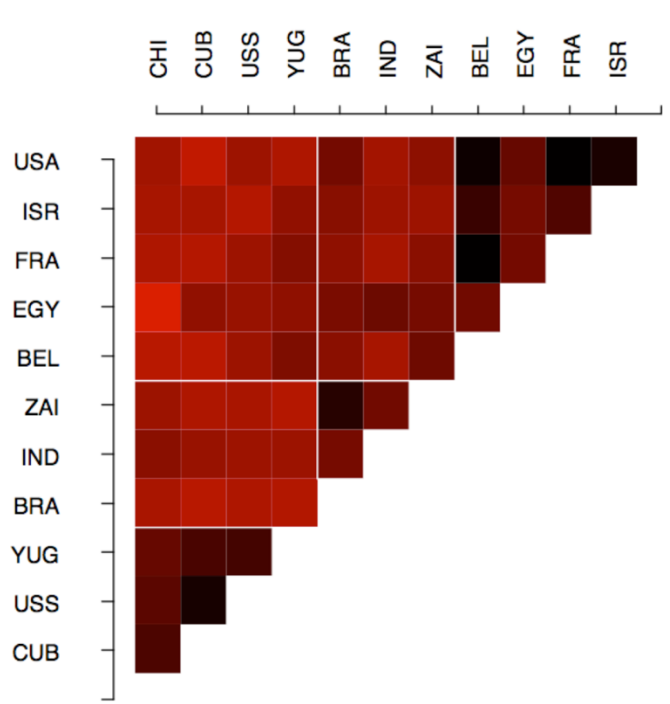
- Implementation
  - Center data by removing the sample mean
  - Multiply data set with top eigenvectors of the sample covariance matrix
- Illustration



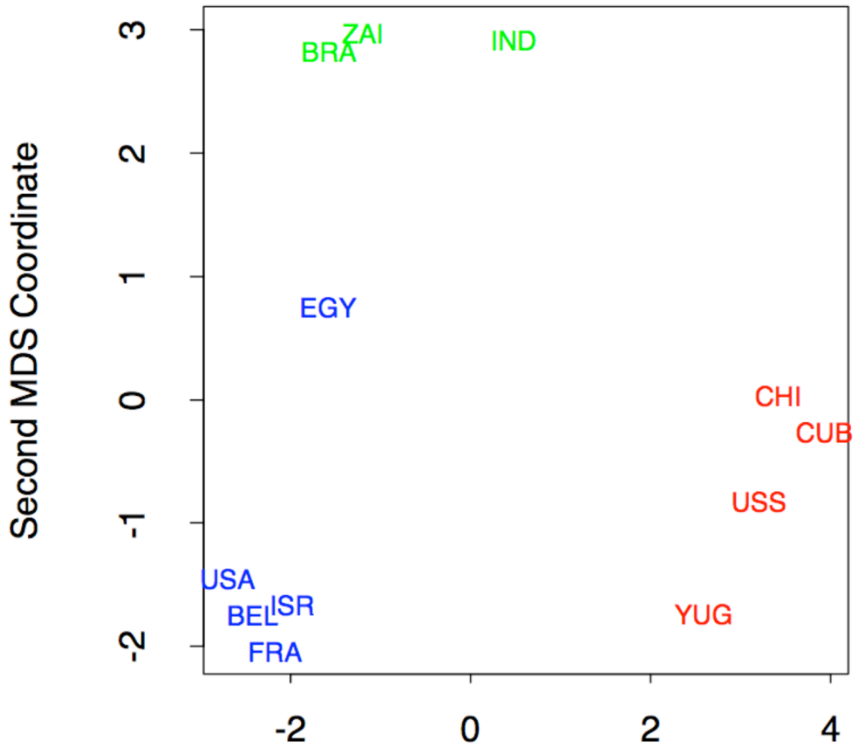
- Salient features
  - Spectral method
    - Incremental embeddings
    - Estimator of the intrinsic dimensionality
    - (covariance eigenvalues = variance along the projection axes)
  - Parametric mapping model

Classical metric multidimensional scaling

- Young & Householder, 1938; Torgerson, 1952.
- Idea
  - Fit a line/(hyper)plane through the data cloud and project on it
  - Inner product preservation (≈ distance preservation)



Reordered Dissimilarity Matrix



Classical metric multidimensional scaling

- Young & Householder, 1938; Torgerson, 1952.
- Idea
  - Fit a line/(hyper)plane through the data cloud and project on it
  - Inner product preservation (≈ distance preservation)
- Details
  - Same problem, same model, and same solution as PCA, except that $\boldsymbol{\Xi}$ is unknown; only the Gram matrix $\mathbf{G}$ is available
  - The SVD $\boldsymbol{\Xi} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T$ allows us to write $\mathbf{C}_{\boldsymbol{\Xi}\boldsymbol{\Xi}} = \frac{1}{N}\boldsymbol{\Xi}\boldsymbol{\Xi}^T = \frac{1}{N}\mathbf{V}(\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T)\mathbf{V}^T$ and $\mathbf{X} = \mathbf{I}_{P\times D}\mathbf{V}^T\boldsymbol{\Xi} = \mathbf{I}_{P\times D}\boldsymbol{\Sigma}\mathbf{U}^T$
  - Similarly, plugging the SVD in $\mathbf{G} = \boldsymbol{\Xi}^T\boldsymbol{\Xi} = \mathbf{U}(\boldsymbol{\Sigma}^T\boldsymbol{\Sigma})\mathbf{U}^T$ leads to its EVD (with $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^T\boldsymbol{\Sigma}$)
  - Hence, the EVD of $\mathbf{G}$ provides us with $\mathbf{U}$ and the PCA solution is equivalently given by $\mathbf{X} = \mathbf{I}_{P\times N}\boldsymbol{\Lambda}^{1/2}\mathbf{U}^T = [\sqrt{\lambda_{ii}}\mathbf{u}_i]_{1\leq i\leq P}^T$ This solution minimizes $\|\mathbf{G} - \mathbf{X}^T\mathbf{X}\|_2^2$.

Classical metric multidimensional scaling

- Details (cont'd)

If $\boldsymbol{\Xi}\mathbf{1} \neq \mathbf{0}$, then it cannot be directly centered...

But double centering can be applied to $\mathbf{G}$, thanks to centering matrix $\mathbf{C} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$: $\mathbf{CGC} = (\boldsymbol{\Xi} - \frac{1}{N}\boldsymbol{\Xi}\mathbf{1}\mathbf{1}^T)^T(\boldsymbol{\Xi} - \frac{1}{N}\boldsymbol{\Xi}\mathbf{1}\mathbf{1}^T)$ corresponds to the Gram matrix of the centered data set

If pairwise Euclidean distances are available instead of inner products, double centering works as well!

Double centering for pairwise Euclidean distances

Squared Euclidean pairwise distances are given by
$\boldsymbol{\Delta} = [\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|_2^2]_{1\leq i,j\leq N} = \operatorname{diag}(\mathbf{G})\mathbf{1}^T - 2\mathbf{G} + \mathbf{1}\operatorname{diag}(\mathbf{G})^T,$
where $\mathbf{G} = \boldsymbol{\Xi}^T\boldsymbol{\Xi}$
As distances are translation invariant, we freely assume that $\boldsymbol{\Xi}\mathbf{1} = \mathbf{0}$
The formula of double centering for distances is
$\mathbf{G} = -\frac{1}{2}\mathbf{C}\boldsymbol{\Delta}\mathbf{C} = -\frac{1}{2}\left(\boldsymbol{\Delta} - \frac{1}{N}\boldsymbol{\Delta}\mathbf{1}\mathbf{1}^T - \frac{1}{N}\mathbf{1}\mathbf{1}^T\boldsymbol{\Delta} + \frac{1}{N^2}\mathbf{1}\mathbf{1}^T\boldsymbol{\Delta}\mathbf{1}\mathbf{1}^T\right)$
where $\mathbf{C} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$ is the centering matrix
Expanding $\boldsymbol{\Delta}$ proves the equality

A.k.a. PCoA
Principal Coordinate Analysis
JC Gower, Biometrika 1966

Classical metric multidimensional scaling

- Implementation
  - 'Double centering':
    - It converts distances into inner products
    - It indirectly cancels the sample mean in the Gram matrix
  - Eigenvalue decomposition of the centered Gram matrix
  - Scaled top eigenvectors provide projected coordinates

- Salient features
  - Provides same solution as PCA iff dissimilarity = Eucl. distance
  - Nonparametric model
    (Out-of-sample extension is possible with Nyström formula)

Stress-based MDS & Sammon mapping

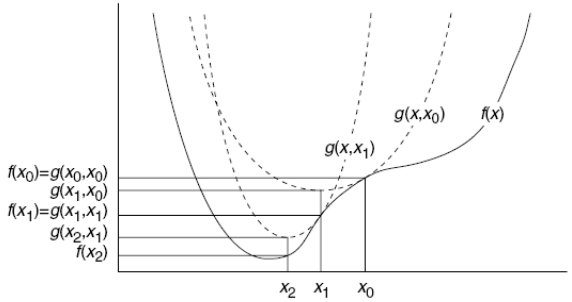
- Kruskal, 1964; Sammon, 1969; de Leeuw, 1977.
- Idea
  - True distance preservation, quantified by a cost function
  - Particular case of stress-based MDS
- Details
  - Distances: $\delta_{ij} = \|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|_2$
    $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$
  - Objective functions:
    - 'Strain' $E(\mathbf{X}; \boldsymbol{\Delta}, \mathbf{W}) = \frac{1}{C} \sum_{i,j=1}^{N} w_{ij} \left(\delta_{ij}^2 - d_{ij}^2\right)^2$
    - 'Stress' $E(\mathbf{X}; \boldsymbol{\Delta}, \mathbf{W}) = \frac{1}{C} \sum_{i,j=1}^{N} w_{ij} \left(\delta_{ij} - d_{ij}\right)^2$
    - Sammon's stress $E(\mathbf{X}; \boldsymbol{\Delta}) = \frac{1}{\sum_{i,j=1}^{N} \delta_{ij}} \sum_{i,j=1}^{N} \frac{(\delta_{ij} - d_{ij})^2}{\delta_{ij}}$

.

		1 BOST	2 NY	3 DC	4 MIAM	5 CHIC	6 SEAT	7 SF	8 LA	9 DENV
1	BOSTON	0	206	429	1504	963	2976	3095	2979	1949
2	NY	206	0	233	1308	802	2815	2934	2786	1771
3	DC	429	233	0	1075	671	2684	2799	2631	1616
4	MIAMI	1504	1308	1075	0	1329	3273	3053	2687	2037
5	CHICAGO	963	802	671	1329	0	2013	2142	2054	996
6	SEATTLE	2976	2815	2684	3273	2013	0	808	1131	1307
7	SF	3095	2934	2799	3053	2142	808	0	379	1235
8	LA	2979	2786	2631	2687	2054	1131	379	0	1059
9	DENVER	1949	1771	1616	2037	996	1307	1235	1059	0

Stress-based MDS & Sammon mapping

- Implementation
  - Steepest descent of the stress function (Kruskal, 1964)
  - Pseudo-Newton minimization of the stress function (diagonal approximation of the Hessian; used in Sammon, 1969)
  - SMaCoF for weighted stress (scaling by majorizing a complex function; de Leeuw, 1977)



- Salient features
  - Nonparametric mapping
  - Main metaparameter: distance weights w_{ij}
  - How can we choose them?
    → Give more importance to small distances
    → Pick a decreasing function function of distance δ_{ij} such as in Sammon mapping
  - Sammon mapping has almost no metaparameters
  - Any distance in the high-dim space can be used (e.g. geodesic distances; see Isomap)
  - Optimization procedure can get stuck in local minima

Nonmetric multidimensional scaling

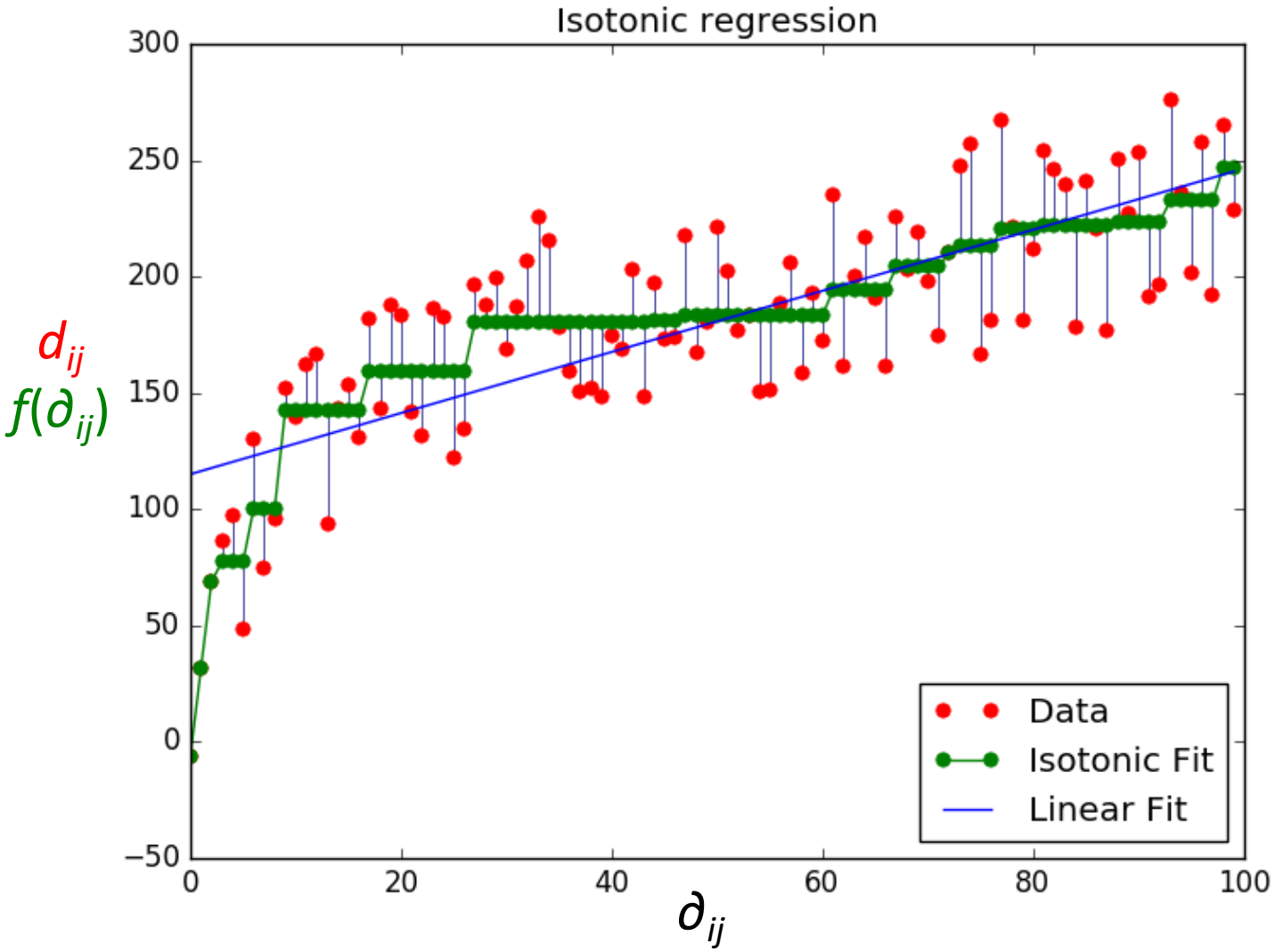
- Shepard, 1962; Kruskal, 1964.
- Idea
  - Stress-based MDS for ordinal (nonmetric) data
  - Try to preserve monotically transformed distances (and optimise the transformation)
- Details
  - Cost function

$$E(\mathbf{X}, f; \mathbf{\Delta}, \mathbf{W}) = \frac{\sum_{i,j=1}^{N} w_{ij}(f(\delta_{ij}) - d_{ij})^2}{\sum_{i,j=1}^{N} d_{ij}^2}$$

$$\arg\min_{\mathbf{X},f} E(\mathbf{X}, f; \mathbf{\Delta}) \text{ such that } 0 \leq u \leq v \Rightarrow 0 \leq f(u) \leq f(v)$$

- Implementation
  - Monotone (a.k.a. isotonic) regression
- Salient features
  - Ad hoc optimization (denominator to avoid trivial collapsing in solution)
  - Nonparametric model

Nonmetric multidimensional scaling



Curvilinear component analysis

- Demartines & Hérault, 1995.
- Idea
  - Distance preservation
  - Change Sammon weighting scheme
    (use decreasing function of the low-dim distance
    instead of decreasing function of the high-dim distance)
- Cost function

$$E(\mathbf{X};\boldsymbol{\Delta},\lambda)=\sum_{i,j=1}^{N}\left(\delta_{ij}-d_{ij}\right)^{2}H(\lambda-d_{ij})$$

- Implementation
  - Stochastic gradient descent (or 'pin-point' radial update)

$$\mathbf{x}_j \leftarrow \mathbf{x}_j-\alpha\frac{\delta_{ij}-d_{ij}}{d_{ij}}H(\lambda-d_{ij})\left(\mathbf{x}_i-\mathbf{x}_j\right)$$

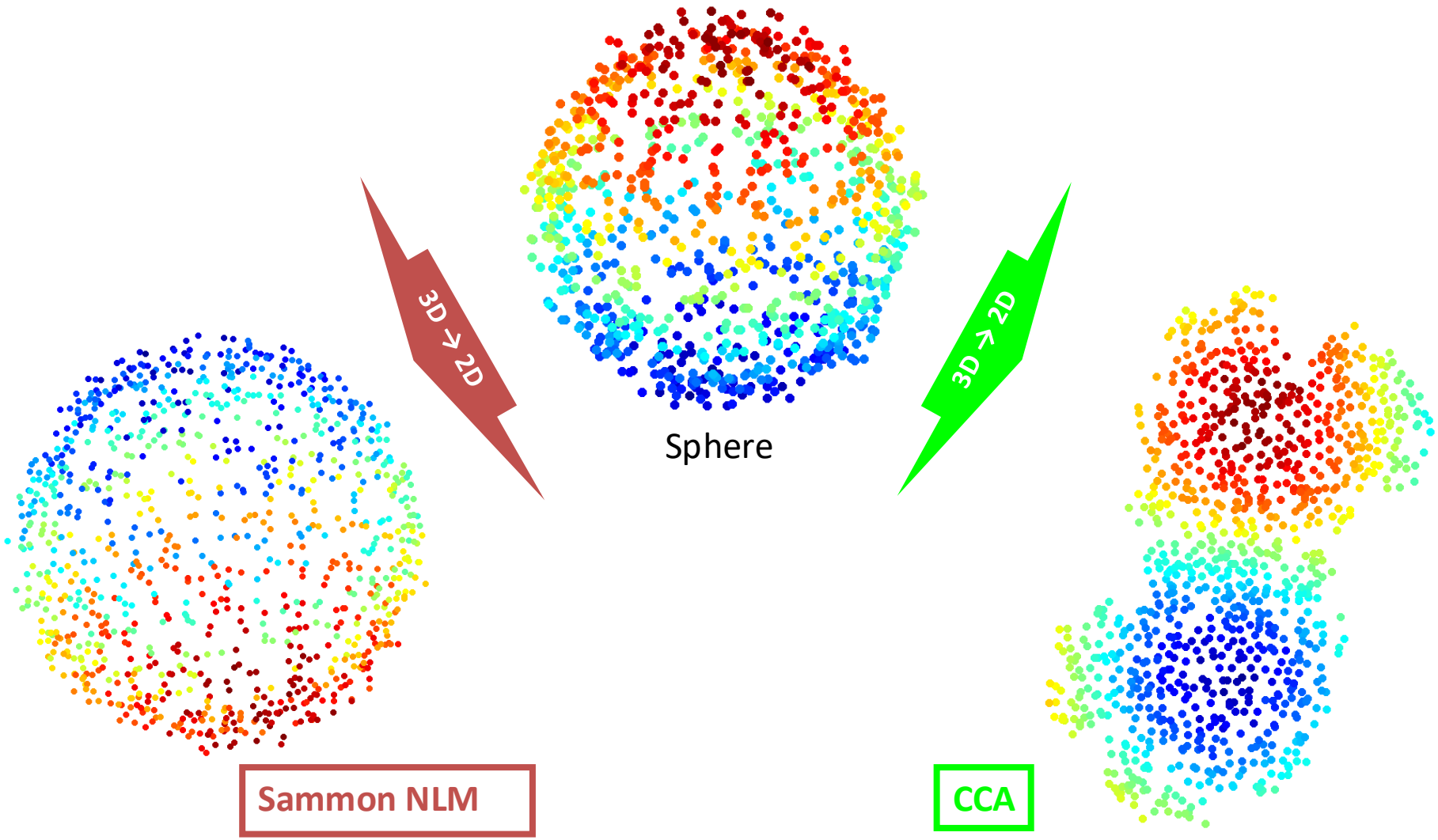
- Salient features
  - Nonparametric mapping
  - Metaparameters are the decay laws of α and λ
  - Can be used with geodesic distances (Lee & Verleysen, 2000)
  - Able to 'tear' manifolds

Distance preservation

- Idea
  - Near, far ⟶ Distances
  - True distance preservation quantified by a cost function
- Details
  - Distances: $\delta_{ij} = \|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|_2$ Not necessarily Euclidean in HD
    $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$ Euclidean in LD (comp. easier)
  - Objective functions:
    - 'Stress': $E(\mathbf{X}; \boldsymbol{\Delta}, \mathbf{W}) = \frac{1}{C} \sum_{i,j=1}^{N} w_{ij} \left(\delta_{ij} - d_{ij}\right)^2$
    - 'SStress': $E(\mathbf{X}; \boldsymbol{\Delta}, \mathbf{W}) = \frac{1}{C} \sum_{i,j=1}^{N} w_{ij} \left(\delta_{ij}^2 - d_{ij}^2\right)^2$
    - Sammon's stress: $E(\mathbf{X}; \boldsymbol{\Delta}) = \frac{1}{\sum_{i,j=1}^{N} \delta_{ij}} \sum_{i,j=1}^{N} \frac{(\delta_{ij} - d_{ij})^2}{\delta_{ij}}$
    - CCA: $E(\mathbf{X}; \boldsymbol{\Delta}, \lambda) = \sum_{i,j=1}^{N} \left(\delta_{ij} - d_{ij}\right)^2 H(\lambda - d_{ij})$

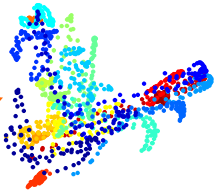
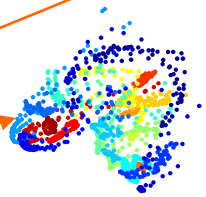
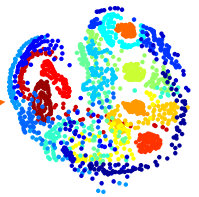
Monotically decreasing function
(often a step function)

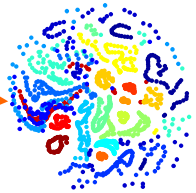
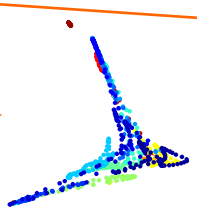
Distance preservation

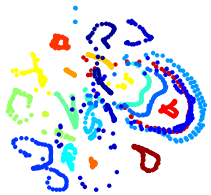


(NL)DR through time…

1901 Principal component analysis (PCA)
1938 Classical multidimensional scaling (CMDS)
1962 Nonmetric MDS (NMDS)
1969 Sammon's nonlinear mapping (NLM)
1982 Self-organising maps (SOMs)
1991 Auto-encoder (back prop.)
1993 Curvilinear component analysis (CCA)
1996 Kernel PCA
1998 Isomap
2000 Locally linear embedding (LLE)
2002 Laplacian eigenmaps (LE)
2002 Stochastic neighbour embedding (SNE)
2006 Auto-encoder (deep learning)
2008 Student-distributed SNE (*t*-SNE)
2010 Neighbour retrieval & vis. (NeRV)
2012 Jensen-Shannon Embedding (JSE)
2014 Multiscale JSE (Ms JSE)
2018 UMAP, *tt*-SNE, Ms *t*-SNE
2019 FI*t*-SNE, NE with missing data
2022 Fast Multiscale NE

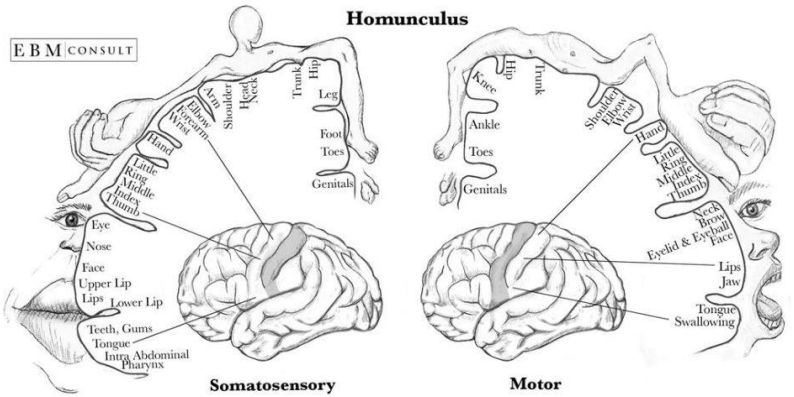



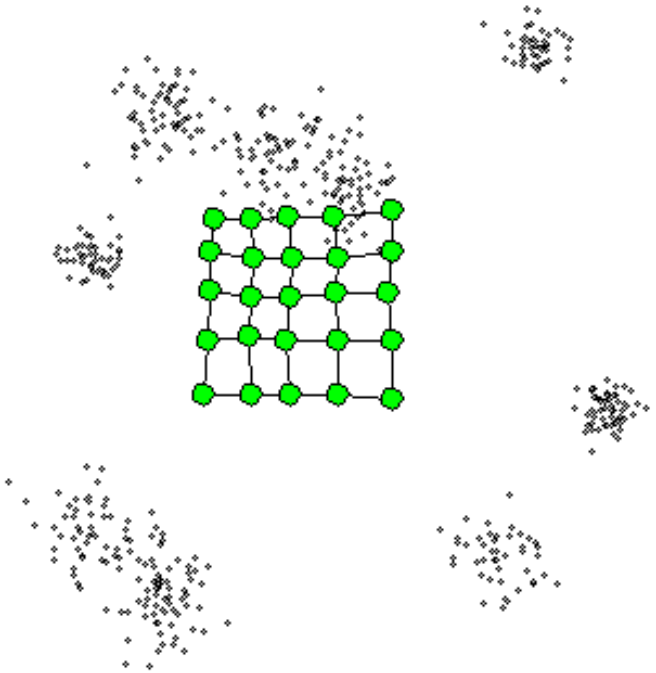
Spectral
Neural network
Reconstruction error
Distance preservation
Similarity preservation

Self-organizing map

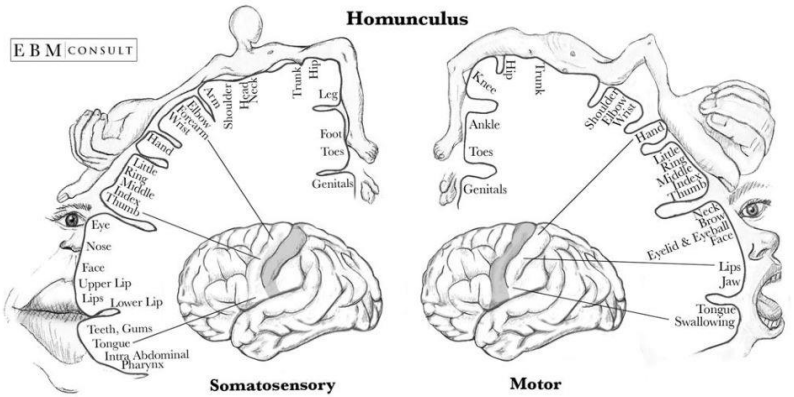
- von der Malsburg, 1973; Kohonen, 1982.
- Idea
  - Biological inspiration (brain cortex)
  - Nonlinear version of PCA
    - Replace PCA plane with an articulated grid
    - Fit the grid through the data cloud

(≈ K-means with a priori topology and 'winner takes most' rule)





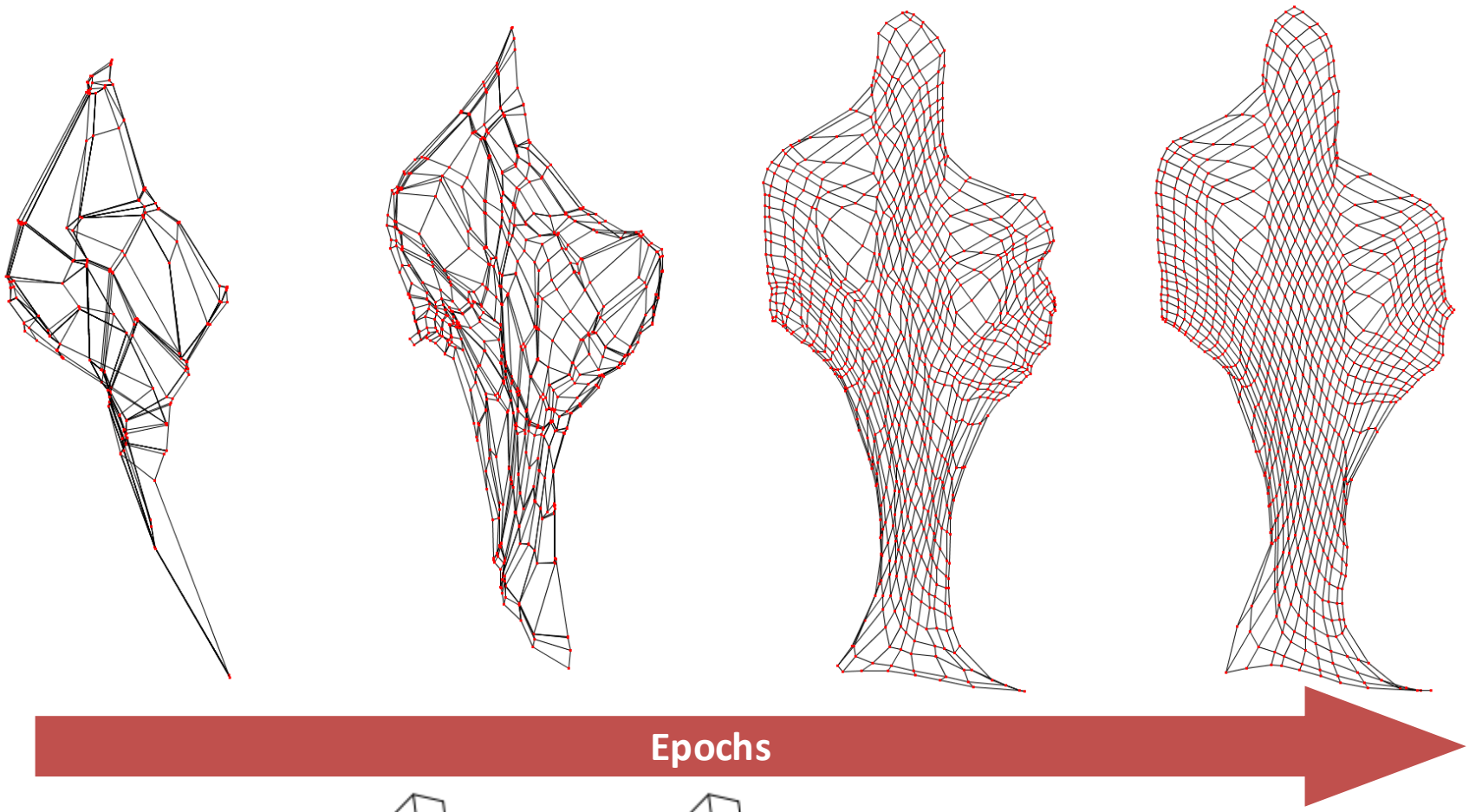
Self-organizing map



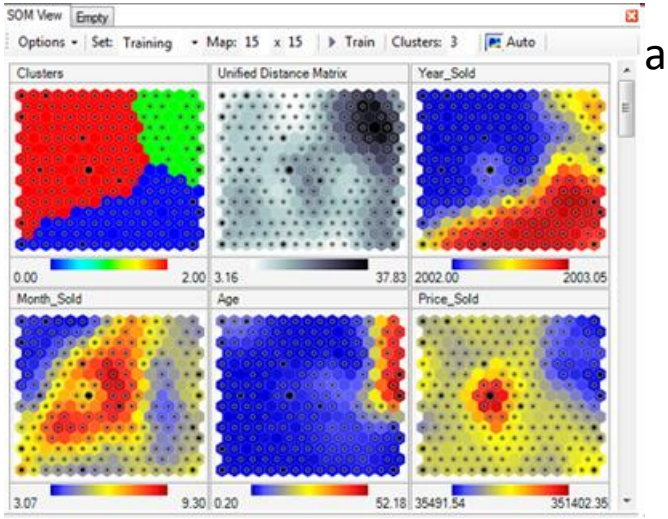
- von der Malsburg, 1973; Kohonen, 1982.
- Idea
  - Biological inspiration (brain cortex)
  - Nonlinear version of PCA
    - Replace PCA plane with an articulated grid
    - Fit the grid through the data cloud

  (≈ K-means with a priori topology and 'winner takes most' rule)
- Details
  - A grid is defined in the low-dim space: $\mathbf{G} = [\mathbf{g}_i]_{1 \le i \le N}$ and $d(\mathbf{g}_i, \mathbf{g}_j)$
  - Grid nodes have high-dim coordinates as well: $\mathbf{\Gamma} = [\boldsymbol{\gamma}_i]_{1 \le i \le N}$
  - The high-dim coordinates are updated in an adaptive procedure (at each epoch, all data vectors are presented 1 by 1 in random order):
    - Best matching node: $j = \arg\min_i \|\boldsymbol{\xi}_k - \boldsymbol{\gamma}_i\|_2$
    - Coordinate update: $\boldsymbol{\gamma}_i \leftarrow \boldsymbol{\gamma}_i + \alpha K\left(\frac{d(\mathbf{g}_i, \mathbf{g}_j)}{\lambda}\right)(\boldsymbol{\xi}_k - \boldsymbol{\gamma}_i),$
      where K is a decreasing function from $\mathbb{R}^+$ to $\mathbb{R}^+$

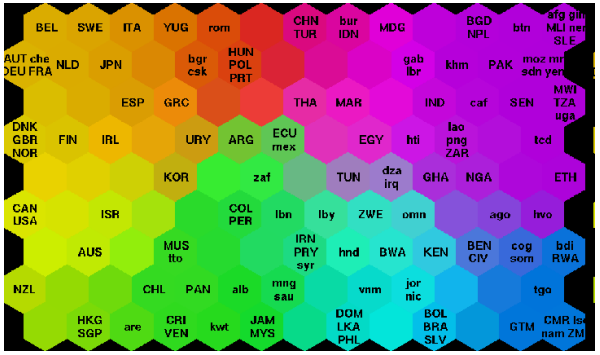
Self-organizing maps

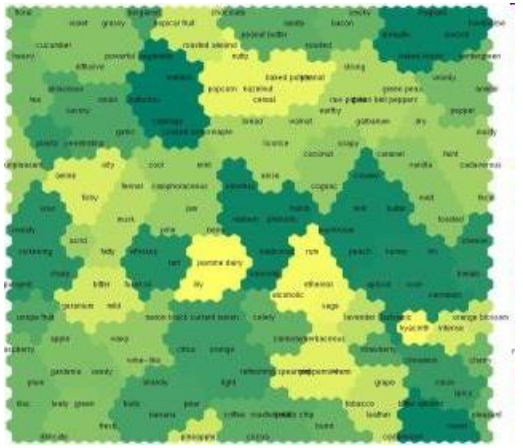


Self-organizing map

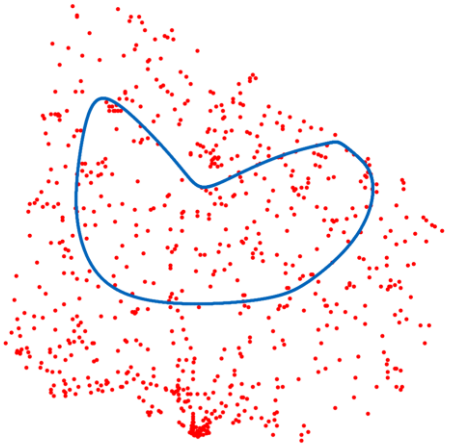


ace



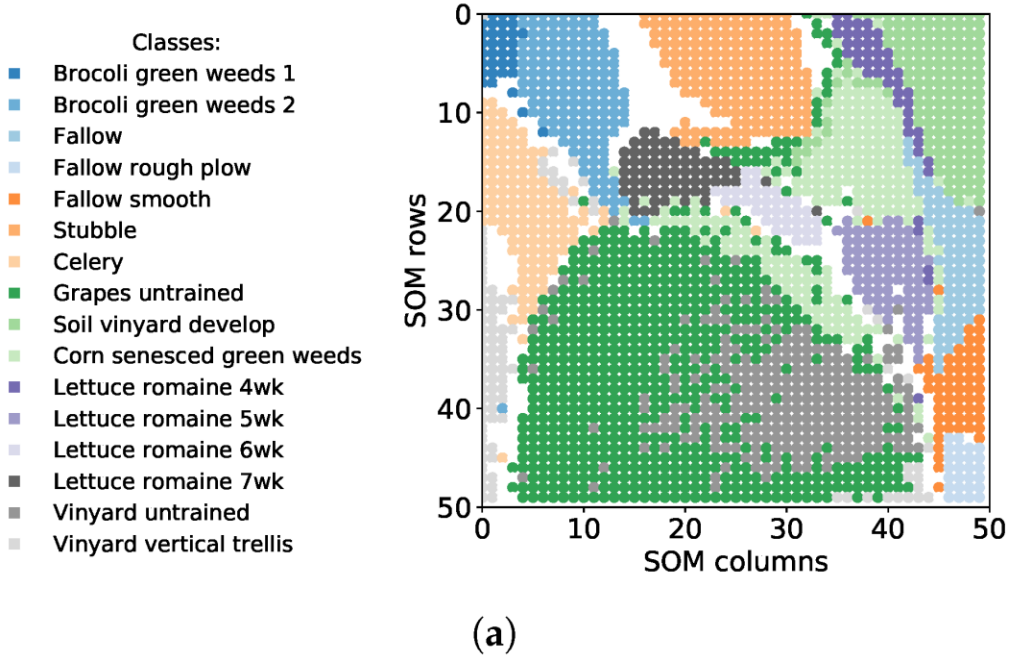


- Salient features
  - Nonparametric model
  - Many metaparameters: grid topology and decay laws for α and λ
  - Performs a vector quantization
  - Batch (non-adaptive) versions exist
  - Popular in visualization and exploratory data analysis
  - Low-dim coordinates are fixed…
  - … but principle can be ‘reversed’ → Isotop, XOM

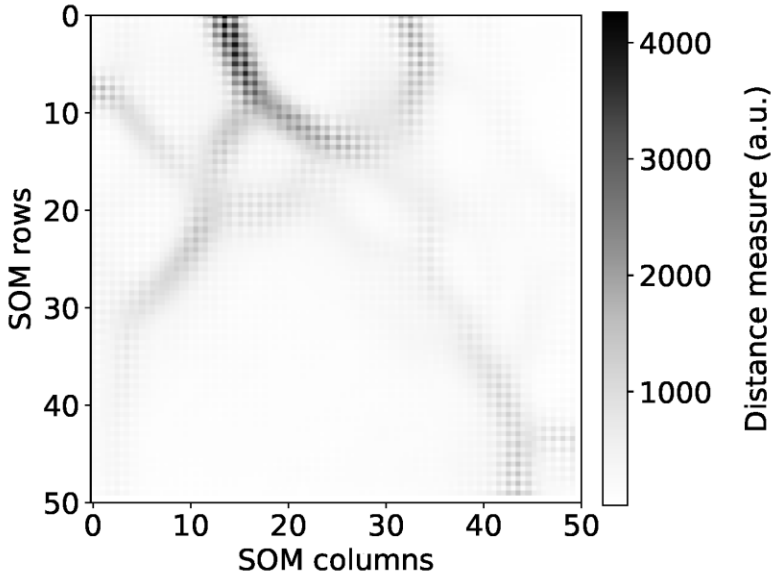


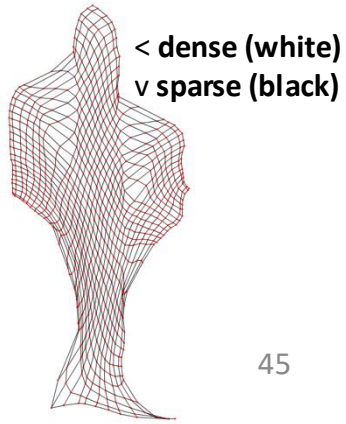
Clustering with SOMs…

SOMs differ from other methods of prototype-based vector quantization (K-means, competitive learning, neural gas, etc.) due to the (regular) grid



(a)





Riese, Keller & Hinz, MDPI Remote Sensing 2019

Auto-encoder

- Kramer, 1991; DeMers & Cottrell, 1993;
  Hinton & Salakhutdinov, 2006.
- Idea
  - Based on the TLS reconstruction error like PCA
  - Cascaded codec with a 'bottleneck' (as in an hourglass)
  - Replace PCA linear mapping with a nonlinear one
- Details
  - Depends on chosen function approximator
    (often a feed-forward ANN such as a multilayer perceptron)
- Implementation
  - Apply the learning procedure to the cascaded networks
  - Catch output value of the bottleneck layer
- Salient features
  - Parametric model (out-of-sample extension is straightforward)
  - Provides both backward and forward mapping
  - The cascaded networks have a 'deep architecture'
    → learning can be inefficient
    Solution: initialize backpropagation with restricted Boltzmann machines

Auto-encoder

PCA = minimal reconstruction error in HD after forward/backward linear transformation (HD-LD-HD)
AE = the same with *nonlinear* transformation (e.g. feed forward neural network)

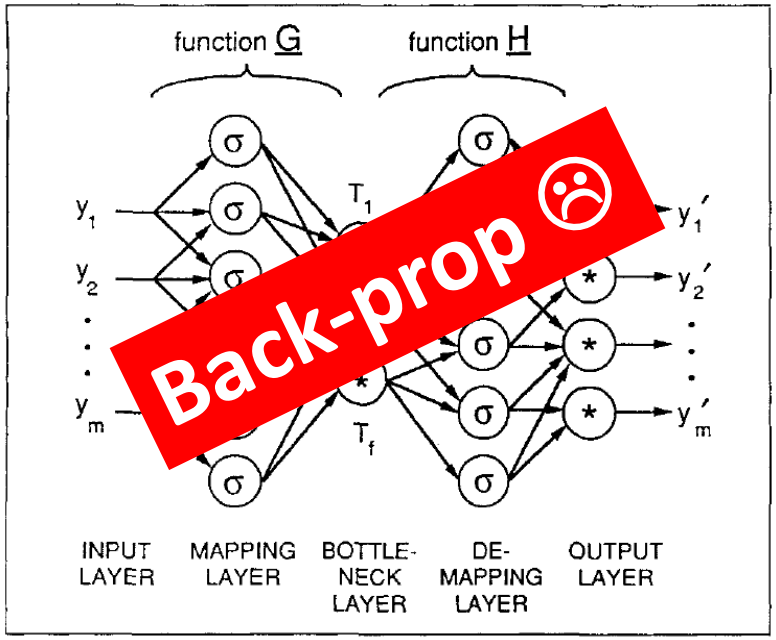


Figure 2. Network architecture for simultaneous determination of *f* nonlinear factors using an autoassociative network.

σ indicates sigmoidal nodes, * indicates sigmoidal or linear nodes.

Original figure from Kramer, 1991.

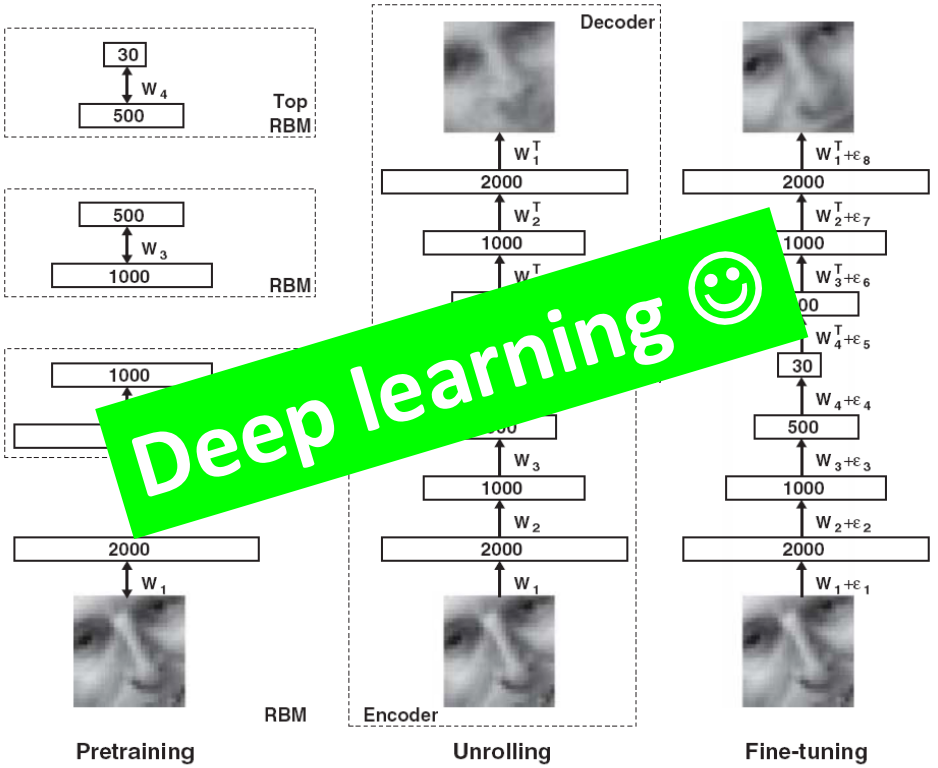
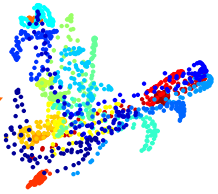
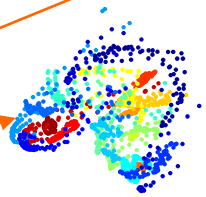
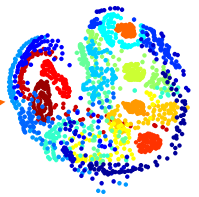


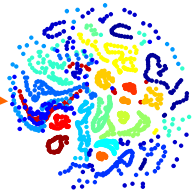
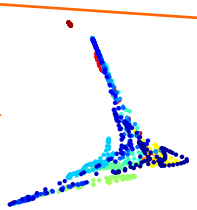
Fig. 1. Pretraining consists of learning a stack of restricted Boltzmann machines (RBMs), each having only one layer of feature detectors. The learned feature activations of one RBM are used as the "data" for training the next RBM in the stack. After the pretraining, the RBMs are "unrolled" to create a deep autoencoder, which is then fine-tuned using backpropagation of error derivatives.

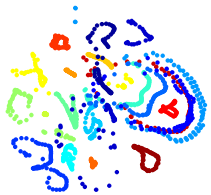
Original figure from Salakhutdinov, 2006.

(NL)DR through time…

1901 Principal component analysis (PCA)
1938 Classical multidimensional scaling (CMDS)
1962 Nonmetric MDS (NMDS)
1969 Sammon's nonlinear mapping (NLM)
1982 Self-organising maps (SOMs)
1991 Auto-encoder (back prop.)
1993 Curvilinear component analysis (CCA)
1996 Kernel PCA
1998 Isomap
2000 Locally linear embedding (LLE)
2002 Laplacian eigenmaps (LE)
2002 Stochastic neighbour embedding (SNE)
2006 Auto-encoder (deep learning)
2008 Student-distributed SNE (*t*-SNE)
2010 Neighbour retrieval & vis. (NeRV)
2012 Jensen-Shannon Embedding (JSE)
2014 Multiscale JSE (Ms JSE)
2018 UMAP, *tt*-SNE, Ms *t*-SNE
2019 FI*t*-SNE, NE with missing data
2022 Fast Multiscale NE

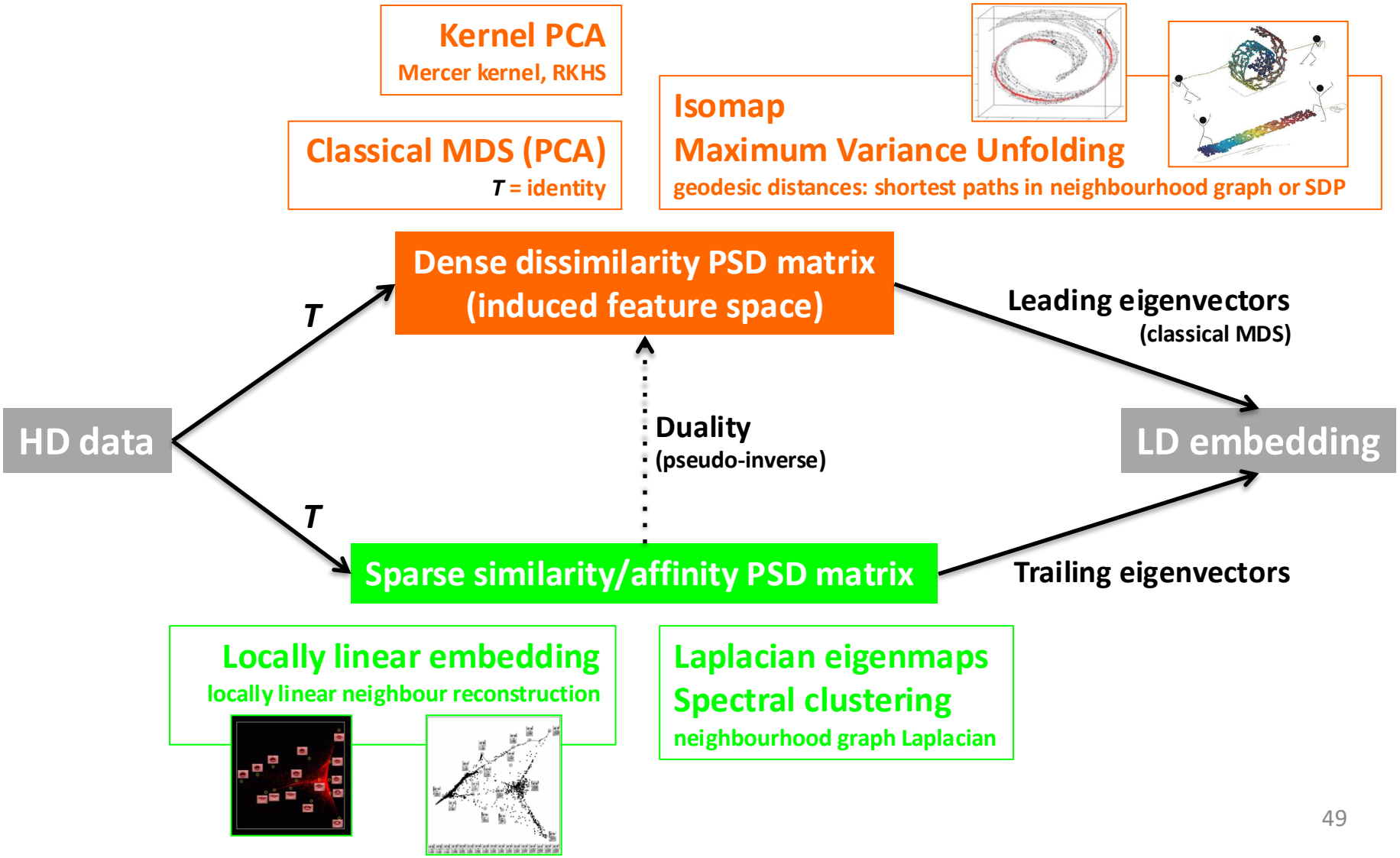



Spectral
Neural network
Reconstruction error
Distance preservation
Similarity preservation

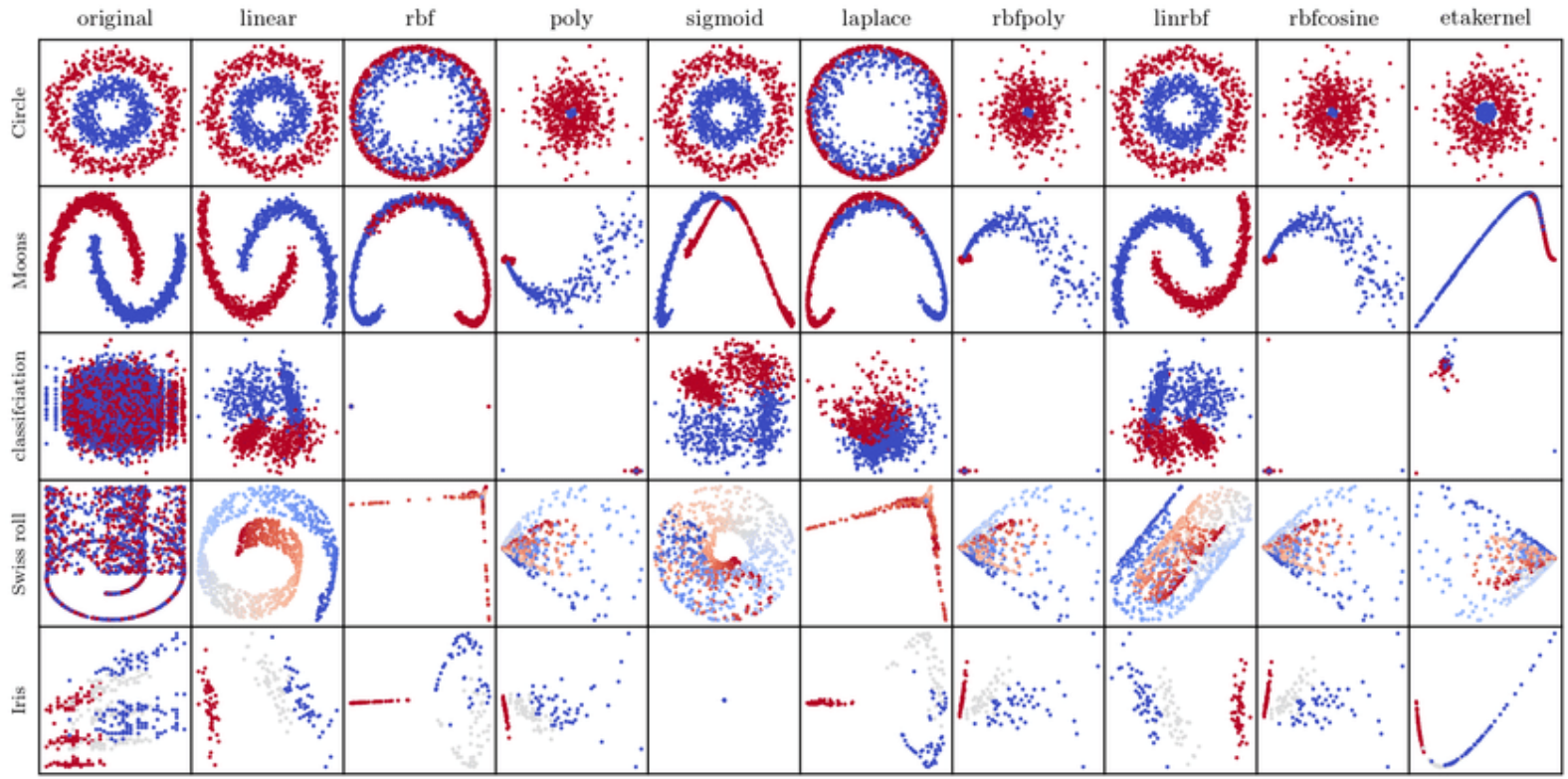
Spectral embedding

Nonlinear data mapping + classical MDS in induced feature space



Kernel PCA

- Schölkopf, Smola & Müller, 1996.
- Idea
  - Apply ‘kernel trick’ to classical metric MDS (and not to PCA!)
  - Apply MDS in an (unknown) ‘feature space’ $\mathcal{F}$



Kernel PCA

- Schölkopf, Smola & Müller, 1996.
- Idea
  - Apply 'kernel trick' to classical metric MDS (and not to PCA!)
  - Apply MDS in an (unknown) 'feature space' $\mathcal{F}$
- Details
  - Property of any Mercer kernel k: $k(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j) = \boldsymbol{\phi}(\boldsymbol{\xi}_i)^T \boldsymbol{\phi}(\boldsymbol{\xi}_j)$
  - $\boldsymbol{\phi} : \mathbb{R}^D \to \mathcal{F},\ \boldsymbol{\xi} \mapsto \boldsymbol{\phi}(\boldsymbol{\xi})$ is an unknown mapping
  - The unknown mapped coordinates in $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\boldsymbol{\xi}_i)]_{1 \le i \le N}$ are involved in $\mathbf{K} = [k(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j)]_{1 \le i,j \le N} = \boldsymbol{\Phi}^T \boldsymbol{\Phi}$, which is a Gram matrix
  - Computing $\mathbf{C}^T \mathbf{K} \mathbf{C}$ (where $\mathbf{C} = \mathbf{I} - \frac{1}{N} \mathbf{1}\mathbf{1}^T$) allows us to center $\boldsymbol{\Phi}$ without knowing it
  - Eigenvalue decomposition $\mathbf{C}^T \mathbf{K} \mathbf{C} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^T$ yields $\boldsymbol{\Phi} = \boldsymbol{\Lambda}^{1/2} \mathbf{U}^T$

Kernel PCA

- Implementation
  - Compute the kernel matrix **K** (starting from pairwise distances or inner products)
  - Perform 'double centering' of **K**
  - Run classical metric MDS on centered **K**
- Kernels from kernel…

1. $K(x,y) = K_1(x,y) + K_2(x,y)$
2. $K(x,y) = c\,K_1(x,y) + K_2(x,y)$ for $c \in \mathbb{R}_+$
3. $K(x,y) = K_1(x,y) + c$ for $c \in \mathbb{R}_+$
4. $K(x,y) = K_1(x,y)\,K_2(x,y)$
5. $K(x,y) = f(x)\,f(y)$ for $f : \mathcal{X} \longrightarrow \mathbb{R}$
6. $K(x,y) = (K_1(x,y) + c)^d$ for $\theta_1 \in \mathbb{R}_+$ and $d \in \mathbb{N}$
7. $K(x,y) = \exp\left(K_1(x,y)/\sigma^2\right)$ for $\sigma \in \mathbb{R}$
8. $K(x,y) = \exp\left(-(K_1(x,x) - 2K_1(x,y) + K_1(y,y))/2\sigma^2\right)$
9. $K(x,y) = K_1(x,y)/\sqrt{K_1(x,x)\,K_1(y,y)}$

- Salient features
  - Nonparametric mapping (Nyström formula can be used)
  - Choice of the kernel? How to adjust its parameter(s)?
  - Important milestone in the history of spectral embedding, but not very effective in actual DR problems

Isomap

- Tenenbaum, 1998, 2000.
- Idea
  - Apply classical metric MDS with a 'smart metric'
  - Replace Euclidean distance with geodesic distance
  - Data-driven approximation of the geodesic distances with shortest paths in a graph of *K*-ary neighbourhoods

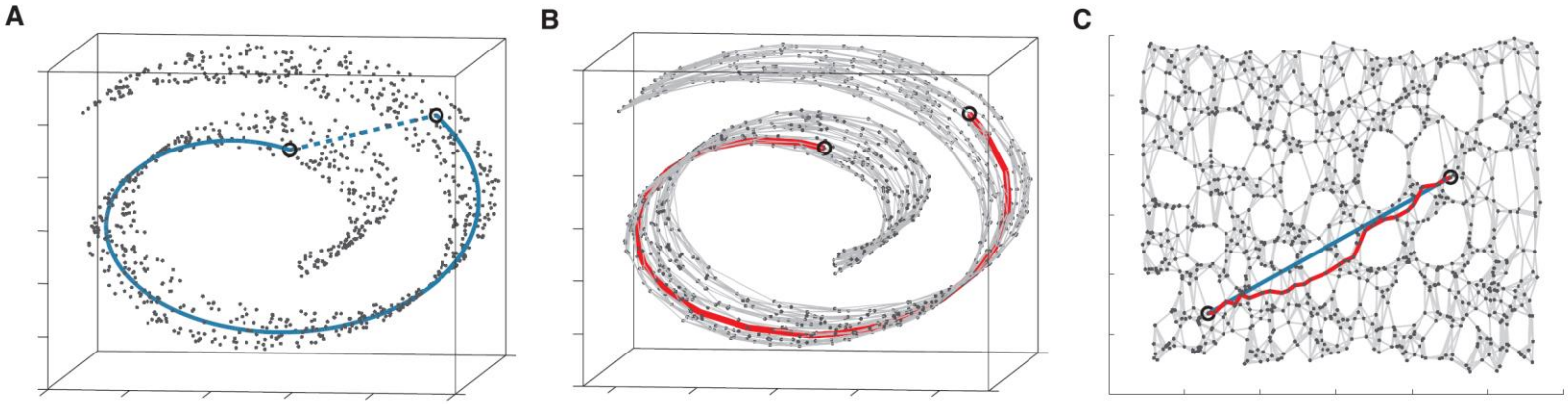


Fig. 3. The "Swiss roll" data set, illustrating how Isomap exploits geodesic paths for nonlinear dimensionality reduction. (**A**) For two arbitrary points (circled) on a nonlinear manifold, their Euclidean distance in the high-dimensional input space (length of dashed line) may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low-dimensional manifold (length of solid curve). (**B**) The neighborhood graph *G* constructed in step one of Isomap (with $K = 7$ and $N =$ 1000 data points) allows an approximation (red segments) to the true geodesic path to be computed efficiently in step two, as the shortest path in *G*. (**C**) The two-dimensional embedding recovered by Isomap in step three, which best preserves the shortest path distances in the neighborhood graph (overlaid). Straight lines in the embedding (blue) now represent simpler and cleaner approximations to the true geodesic paths than do the corresponding graph paths (red).

Original figure in Tenenbaum, 2000.

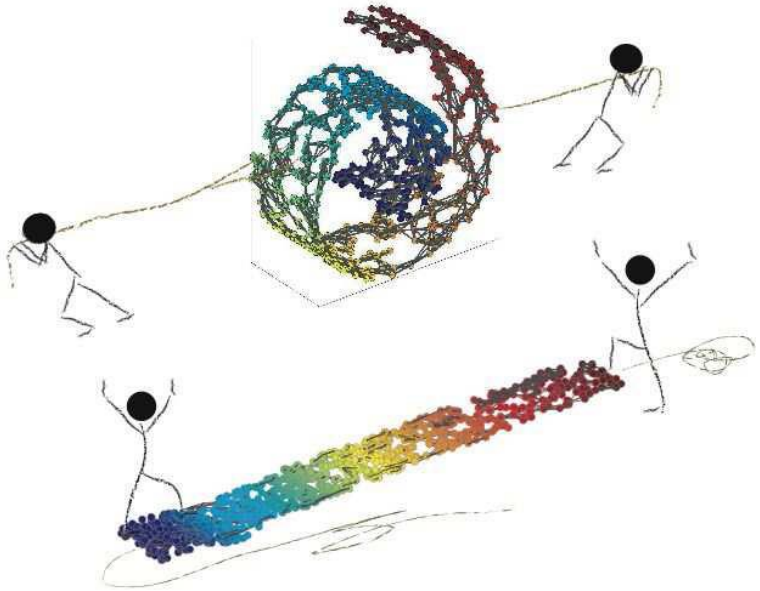
Isomap

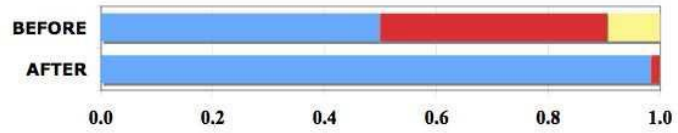
- Model
  - Classical metric MDS is optimal for linear manifold
    → Isomap is optimal for Euclidean manifolds
    (a *P*-dimensional manifold is Euclidean iff
    it is isometric to a *P*-dimensional Euclidean space)
- Implementation
  - Compute/collect pairwise distances
  - Compute graph of *K*-ary neighbourhoods
  - Compute the weighted shortest paths in the graph
  - Apply classical metric MDS on the pairwise geodesic distances
- Salient features
  - Nonparametric mapping (Nyström not applicable because…)
  - Double-centred Gram matrix is not positive semidefinite
    (but fortunately not far from being so)
  - Manifold must be convex
  - Parameter *K* is critical with noisy data (‘short circuits’)

Maximum variance unfolding

- Weinberger & Saul, 2004. (a.k.a. 'semidefinite embedding')
- Idea
  - Do the opposite of Laplacian eigenmaps and try to unfold data
    → ***Stretch*** distance of ***non-neighbouring*** points
  - Classical metric MDS with missing pairwise distances
    → Use semi definite programming (SDP) to maintain the properties of the Gram matrix

.





Original figure in Weinberger, 2004.

Maximum variance unfolding

- Details

$E(\mathbf{X}) = \sum_{i,j=1}^{N} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$

$\arg\max_{\mathbf{X}} E(\mathbf{X})$ such that

- $\sum_{i=1}^{N} \mathbf{x}_i = \mathbf{0}$
- $\|\mathbf{x}_i - \mathbf{x}_j\|_2 = \|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|_2$ if $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_i$ are neighbors

Convert distances into inner products:
$\mathbf{K} = [k_{ij}]_{1 \le i,j \le N} = \left[\mathbf{x}_i^T \mathbf{x}_j\right]_{1 \le i,j \le N}$ and $\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 = k_{ii} - 2k_{ij} + k_{jj}$

$\arg\max_{\mathbf{X}} \mathrm{Tr}(\mathbf{K})$ such that

- $\mathbf{K}$ is positive semidefinite
- $\sum_{i,j=1}^{N} k_{ij} = 0$
- $k_{ii} - 2k_{ij} + k_{jj} = \|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|_2^2$ if $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_i$ are neighbors

.

Maximum variance unfolding

- Implementation
  - Collect pairwise distances and compute *K*-ary neighbourhoods
  - Deduce the corresponding constraints on pairwise distances
  - Formulate everything with inner products and run SDP engine
  - Apply classical metric MDS
- Variants
  - Distances between neighbours can shrink
    (and distances between non-neighbours can only grow as usual)
  - Introduction of slack variables to soften the constraints
- Salient features
  - MVU ≈ KPCA with data-driven local kernels
  - MVU ≈ smart Isomap
  - Semidefinite programming is computationally demanding
  - Metaparameters are *K* and all flags of SDP engine

.

Locally linear embedding

- Roweis & Saul, 2000.
- Idea
  - Each datum can be approximated by a (regularised) linear combination of its *K* nearest neighbors
  - LLE tries to reproduce similar linear combinations in a lower-dimensional space

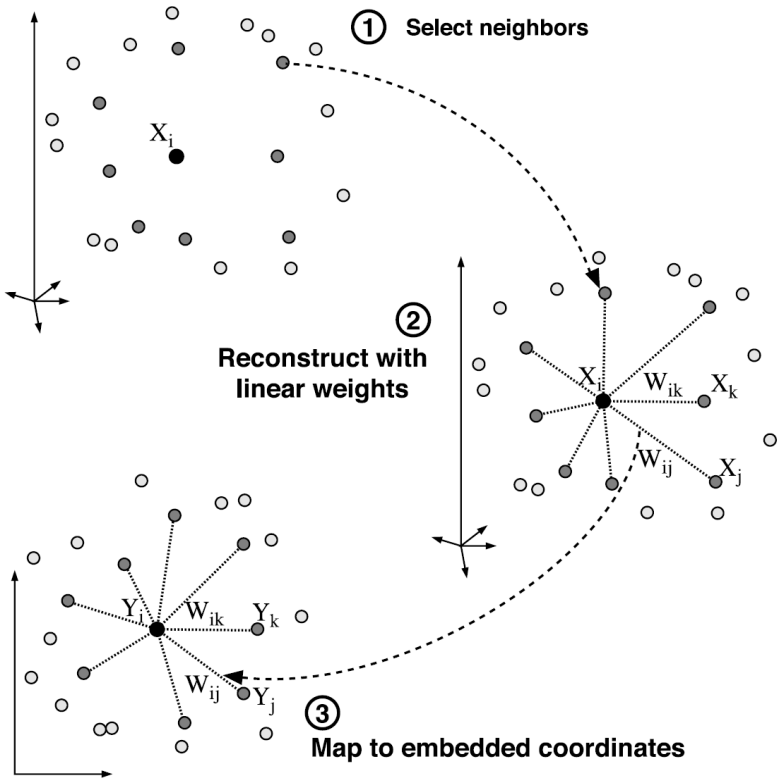


Fig. 2. Steps of locally linear embedding: (1) Assign neighbors to each data point $\vec{X}_i$ (for example by using the *K* nearest neighbors). (2) Compute the weights W_{ij} that best linearly reconstruct $\vec{X}_i$ from its neighbors, solving the constrained least-squares problem in Eq. 1. (3) Compute the low-dimensional embedding vectors $\vec{Y}_i$ best reconstructed by W_{ij}, minimizing Eq. 2 by finding the smallest eigenmodes of the sparse symmetric matrix in Eq. 3. Although the weights W_{ij} and vectors Y_i are computed by methods in linear algebra, the constraint that points are only reconstructed from neighbors can result in highly nonlinear embeddings.

Original figure in Roweis, 2000.

Locally linear embedding

- Details
  - Step 1: $E_1(\mathbf{W};\boldsymbol{\Xi}) = \sum_{i=1}^{N} \left\| \boldsymbol{\xi}_i - \sum_{j\in\mathcal{N}(\boldsymbol{\xi}_i)} w_{ij}\boldsymbol{\xi}_j \right\|^2$
    $\arg\min_{\mathbf{W}} E_1(\mathbf{W};\boldsymbol{\Xi})$ such that $w_{ij} = 0$ if $j \notin \mathcal{N}(\boldsymbol{\xi}_i)$ and $|\mathcal{N}(\boldsymbol{\xi}_i)| > P$
  - Step 2: $E_2(\mathbf{X};\mathbf{W}) = \sum_{i=1}^{N} \left\| \mathbf{x}_i - \sum_{j\in\mathcal{N}(\boldsymbol{\xi}_i)} w_{ij}\mathbf{x}_j \right\|_2^2$
    $= \sum_{i,j=1}^{N} m_{ij}\mathbf{x}_i^T\mathbf{x}_j = \mathrm{Tr}(\mathbf{X}\mathbf{M}\mathbf{X}^T)$
    where $\mathbf{M} = [m_{ij}]_{1\le i,j\le N} = (\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})$
    $\arg\min_{\mathbf{X}} E_2(\mathbf{X};\mathbf{W})$ such that $\mathbf{X}\mathbf{1} = \mathbf{0}$ and $\mathbf{X}\mathbf{X}^T = N\mathbf{I}$
- Implementation
  - Approximate each datum with a regularised linear combination of its *K* nearest neighbours
  - Build the sparse matrix of neighbor weights ($\mathbf{W}$)
  - Compute the eigenvalue decomposition of $\mathbf{M}$
  - Bottom eigenvectors provides embedding coordinates
- Salient features
  - Metaparameters are *K* and the regularization coefficient
  - EVD such as in MDS, but bottom eigenvectors are used
  - Nonparametric mapping (Nyström formula can be used)

Laplacian eigenmaps

- Belkin & Niyogi, 2002.
- Idea
  - Embed neighbouring points close to each other
    → shrink distances between neighbors in the embedding
  - Avoid trivial solutions and undeterminacies
    by constraining the covariance matrix of the embedding
- Details
  - Symmetric affinity matrix:

$$\mathbf{W} = [w_{ij}]_{1 \leq i,j \leq N} \text{ where } w_{ij} = \begin{cases} 1 & \text{if } j \in \mathcal{N}(\boldsymbol{\xi}_i) \\ 0 & \text{otherwise} \end{cases}$$

  - Cost function:

$$E(\mathbf{X}; \mathbf{W}) = \sum_{i,j=1}^{N} w_{ij} \, \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 = \mathrm{Tr}(\mathbf{X}^T \mathbf{L} \mathbf{X})$$
$$\text{where } \mathbf{L} = \mathbf{D} - \mathbf{W} \text{ with } \mathbf{D} = \mathrm{diag}(\mathbf{W1}) \text{ is the Laplacian matrix}$$

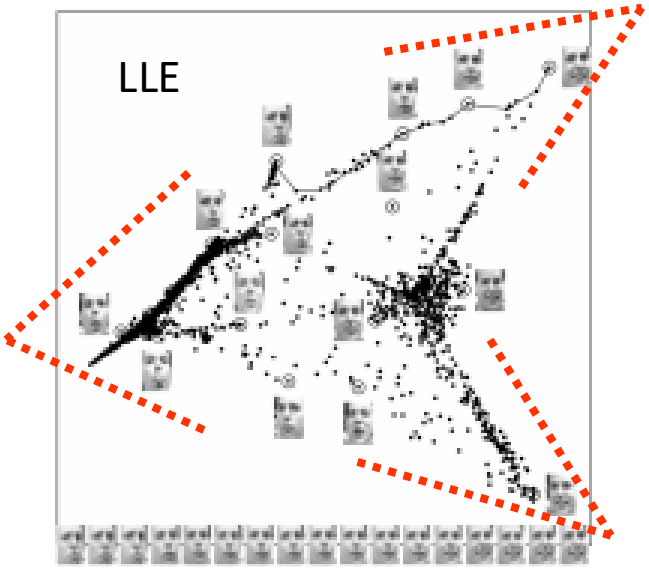
  - Constrained optimization:

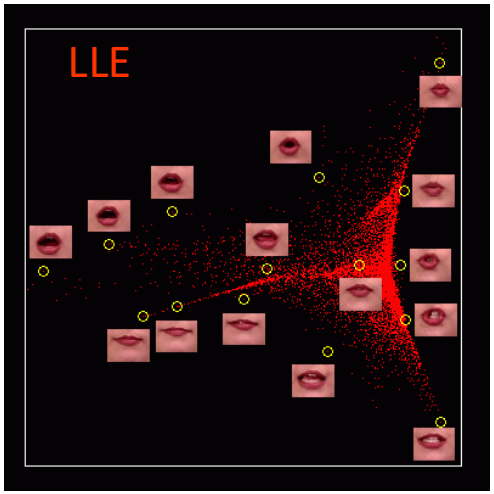
$$\arg\min_{\mathbf{X}} E(\mathbf{X}; \mathbf{W}) \text{ such that } \mathbf{X1} = \mathbf{0} \text{ and } \mathbf{X}^T \mathbf{D} \mathbf{X} = \mathbf{I}$$

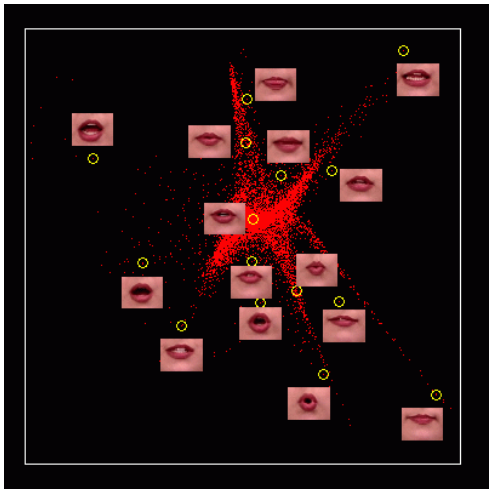
Laplacian eigenmaps

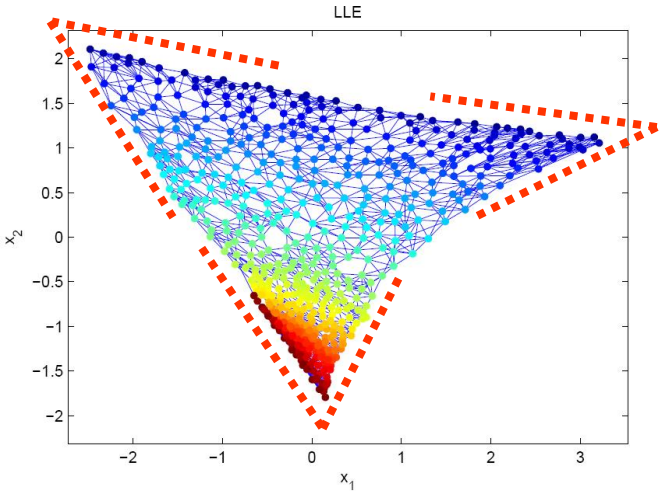
- Implementation
  - Collect distances and compute *K*-ary neighbourhoods
  - Compute the adjacency matrix and the corresponding weight matrix
  - Compute the eigenvalue decomposition of the Laplacian matrix
  - The bottom eigenvectors provide the embedding coordinates
- Salient features
  - Nonparametric mapping (Nyström formula can be used)
  - Connection with
    - LLE (Laplacian operator applied twice)
    - Spectral clustering and graph min-cut (Laplacian matrix normalization is different)
    - Diffusion maps
    - Classical metric MDS with commute time distance
  - Metaparameters are *K* and/or soft neighbourhood kernel parameters

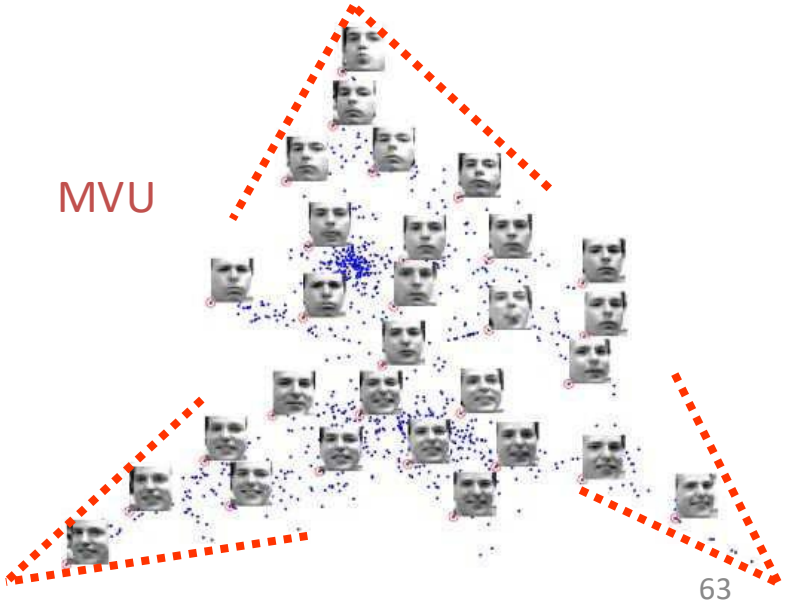
Sparse spectral methods: spiky embeddings











Spectral methods: duality

- Two types of spectral NLDR methods
  - 'Dense' matrix of '***dis***similarities' (e.g. distances)
    → Top eigenvectors
    (CM MDS, Isomap, MVU)
  - 'Sparse' matrix of 'similarities' (or 'affinities')
    → Bottom eigenvectors (except last one)
    (LLE, LE, diff.maps, spectral clustering)
- Duality
  - Pseudo-inverse of sparse matrix
    - Yields a dense matrix
    - Inverts and therefore flips the eigenvalue spectrum
      (bottom eigenvectors become leading ones and vice versa)
- Corollary
  - All spectral methods (both sparse and dense) can be reformulated as applying CM MDS on a dense matrix
  - Example: Laplacian eigenmaps = CM MDS with commute time distances
    (CTDs are related to the pseudo-inverse of the Laplacian matrix)

Spectral versus non-spectral NLDR

Spectral NLDR

☺ Cost function is convex
- Convex optimization (spectral decomposition)
- Global optimum
- Incremental embeddings
- Intrinsic dimensionality can be estimated

☹ Cost function must fit within the spectral framework

☹ It often amounts to applying
1. An a priori nonlinear distance tranformation
2. Classical metric MDS

(Dense/sparse duality!)

☹ Eigenspectrum tail of sparse methods tend to be flat
→ 'Spiky' embeddings

Non-spectral NLDR

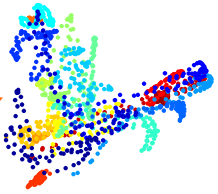
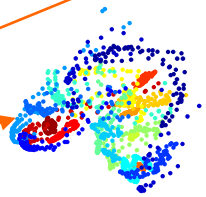
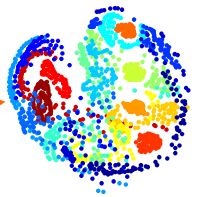
☹ Cost function is not convex
- Ad hoc optimization (e.g. gradient descent)
- Local optima
- Independent embeddings
- No simple way to estimate intrinsic dimensionality

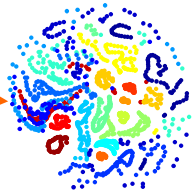
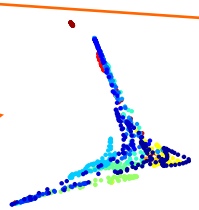
☺ More freedom is granted in the choice of the cost fun.

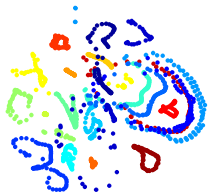
☺ It is often fully data-driven

(NL)DR through time…

1901 Principal component analysis (PCA)
1938 Classical multidimensional scaling (CMDS)
1962 Nonmetric MDS (NMDS)
1969 Sammon's nonlinear mapping (NLM)
1982 Self-organising maps (SOMs)
1991 Auto-encoder (back prop.)
1993 Curvilinear component analysis (CCA)
1996 Kernel PCA
1998 Isomap
2000 Locally linear embedding (LLE)
2002 Laplacian eigenmaps (LE)
2002 Stochastic neighbour embedding (SNE)
2006 Auto-encoder (deep learning)
2008 Student-distributed SNE (*t*-SNE)
2010 Neighbour retrieval & vis. (NeRV)
2012 Jensen-Shannon Embedding (JSE)
2014 Multiscale JSE (Ms JSE)
2018 UMAP, *tt*-SNE, Ms *t*-SNE
2019 FI*t*-SNE, NE with missing data
2022 Fast Multiscale NE

Spectral
Neural network
Reconstruction error
Distance preservation
Similarity preservation

Similarity-based embedding

- Examples
  - Stochastic neighbor embedding (SNE, 2002)
  - *t*-distributed SNE (2008)
  - Neighbour retrieval and visualisation (NeRV, 2010)
- Ingredients
  - Softmax similarities a.k.a. neighbourhood probabilities:

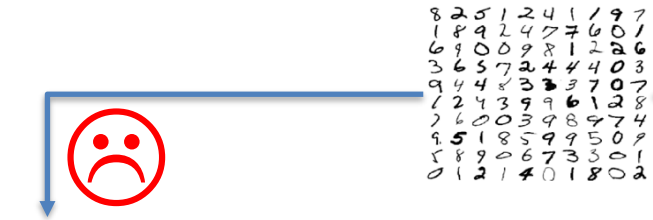
$$\sigma_{ij} = \frac{\exp(-\delta_{ij}^2/(2\lambda_i^2))}{\sum_{k,k\neq i}\exp(-\delta_{ik}^2/(2\lambda_i^2))} \quad \text{and} \quad s_{ij} = \frac{\exp(-d_{ij}^2/2)}{\sum_{k,k\neq i}\exp(-d_{ik}^2/2)}$$

$$t\text{-SNE (heavy-tailed)} \rightarrow \quad s_{ij} = \frac{(1+d_{ij}^2)^{-1}}{\sum_{k,l,k\neq l}(1+d_{kl}^2)^{-1}}$$

  - Similarity preservation (sum of KL divergences):

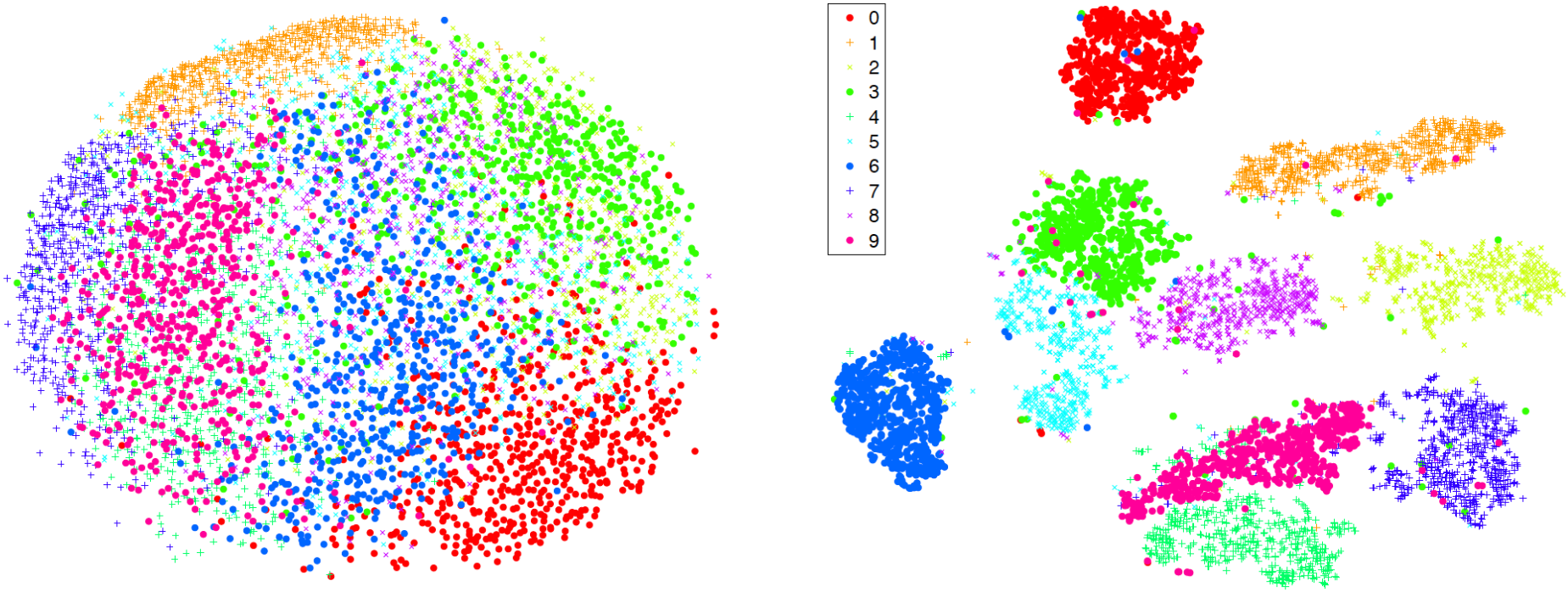
$$E(\mathbf{X};\boldsymbol{\Xi},\boldsymbol{\Lambda}) = \sum_i E_i(\mathbf{X};\boldsymbol{\Xi},\lambda_i) = \sum_{i,j}\sigma_{ij}\log(\sigma_{ij}/s_{ij})$$

Similarity-based embedding



Sammon's nonlinear mapping

t-SNE

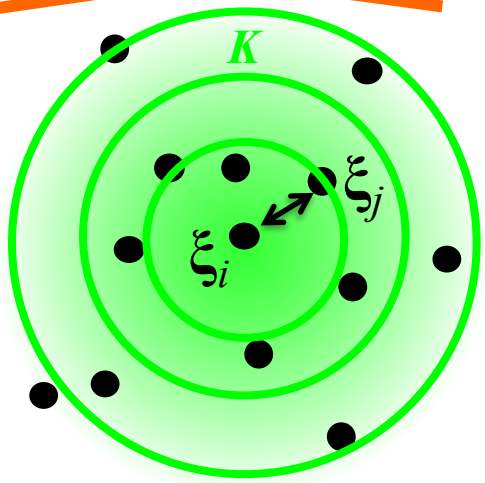


MNIST database of handwritten digits, pictures from L. Vander Maaten & G. Hinton 2008

Student *t*-distributed

Stochastic neighbour embedding

1. Choose size ~~of~~ neighbourhoods in HD space



$$\delta_{ij} = \|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|_2$$

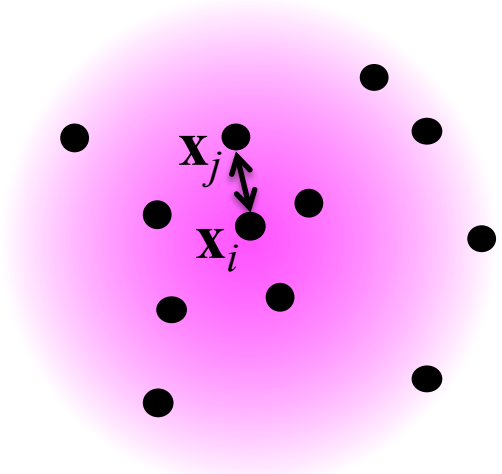
NeRV

Ms. JSE

mixtures of

5. Minimise KL divergences (for all *i*)

$$D_{\mathrm{KL}}(\boldsymbol{\sigma}_i \| \mathbf{s}_i) = \sum_{j=1}^{N} \sigma_{ij} \log(\sigma_{ij}/s_{ij})$$



$$d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$$

2. Convert hard neighbourhoods into soft ones

$$\sigma_{ij} = \frac{\exp(-\delta_{ij}^2/(2\lambda_i^2))}{\sum_{k,k\neq i}\exp(-\delta_{ik}^2/(2\lambda_i^2))}$$

3. Adjust all bandwidths (same entropies for all *i*)

$$\log(K) = -\sum_{j=1}^{N} \sigma_{ij} \log \sigma_{ij}$$

4. Define soft neighbourhoods in LD space

$$s_{ij} = \frac{\exp(-d_{ij}^2/2)}{\sum_{k,k\neq i}\exp(-d_{ik}^2/2)}$$

(with unit bandwidths)

$$s_{ij} = \frac{(1+d_{ij}^2)^{-1}}{\sum_{k,l,k\neq l}(1+d_{kl}^2)^{-1}}$$

Beyond Kullback-Leibler…

Neighbourhood retrieval and visualisation (NeRV)

Type 1 mixture of KL divergences
Venna et al., JMLR 2010

$$D^{\beta}_{\mathrm{KLs1}}(\boldsymbol{\sigma}_i \| \mathbf{s}_i) = (1-\beta) D_{\mathrm{KL}}(\boldsymbol{\sigma}_i \| \mathbf{s}_i) + \beta D_{\mathrm{KL}}(\mathbf{s}_i \| \boldsymbol{\sigma}_i)$$

Jensen-Shannon embedding (JSE, 'Jessie')

Type 2 mixture of KL divergences
Lee et al., ESANN 2012, Neurocomputing 2013

$$D^{\beta}_{\mathrm{KLs2}}(\boldsymbol{\sigma}_i \| \mathbf{s}_i) = (1-\beta) D_{\mathrm{KL}}(\boldsymbol{\sigma}_i \| \mathbf{z}_i) + \beta D_{\mathrm{KL}}(\mathbf{s}_i \| \mathbf{z}_i)$$

$$\text{where } \mathbf{z}_i = (1-\beta)\boldsymbol{\sigma}_i + \beta \mathbf{s}_i$$

Multi-scale JSE

Lee et al., ESANN 2014, Neurocomputing 2015

- $K_l = 2, 4, \ldots, 2^{L_{\max}-l+1}$ with $1 \leq l \leq L \leq L_{\max} \leq \log(N/2)$
- Multi-scale similarities

  Non-weighted average of single-scale similarities

$$\sigma_{ij} = \frac{1}{L}\sum_{l=1}^{L}\sigma_{ijl} \qquad \sigma_{ijl} = \frac{\exp(-\pi_{il}\delta_{ij}^2/2)}{\sum_{k,k\neq i}\exp(-\pi_{il}\delta_{ik}^2/2)}$$

$$s_{ij} = \frac{1}{L}\sum_{l=1}^{L}s_{ijl} \qquad s_{ijl} = \frac{\exp(-p_{il}d_{ij}^2/2)}{\sum_{k,k\neq i}\exp(-p_{il}d_{ik}^2/2)}$$

- Sequential identification of the precisions π_{il} in HD space

  (Usual entropy equalisation, backward from $2^{L_{\max}}$ to 2)
- A priori precisions in LD space: $p_{il} = K_l^{-2/P}$

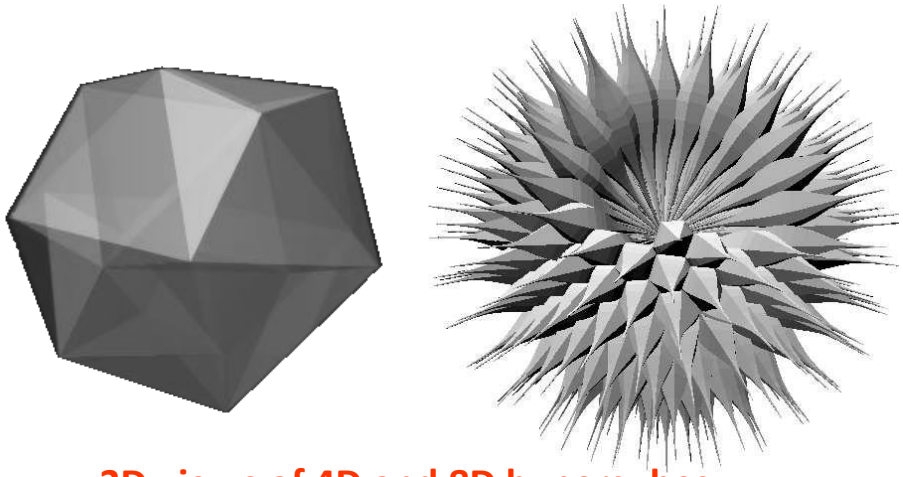
  (Exponential relationship between the size and radius of a K-ary neighborhood in a uniform P-dimensional distribution)
- Multi-scale minimisation of JS divergences

  (From $L = 1$ to $L = L_{\max}$, limited memory BFGS)

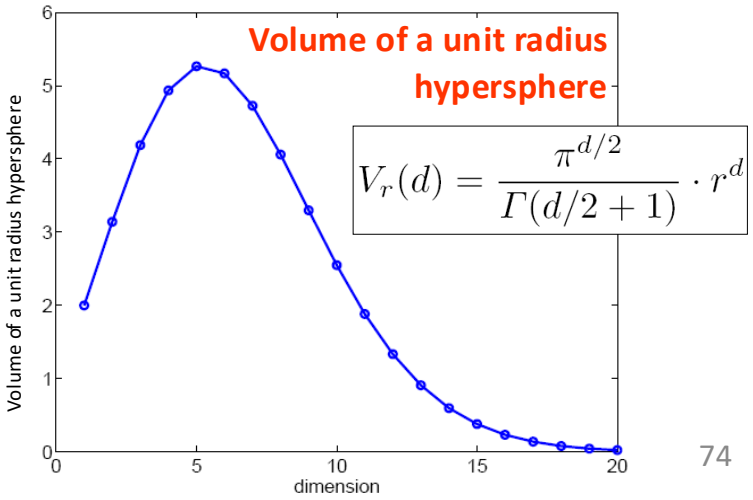
Dimensionality reduction

Why?

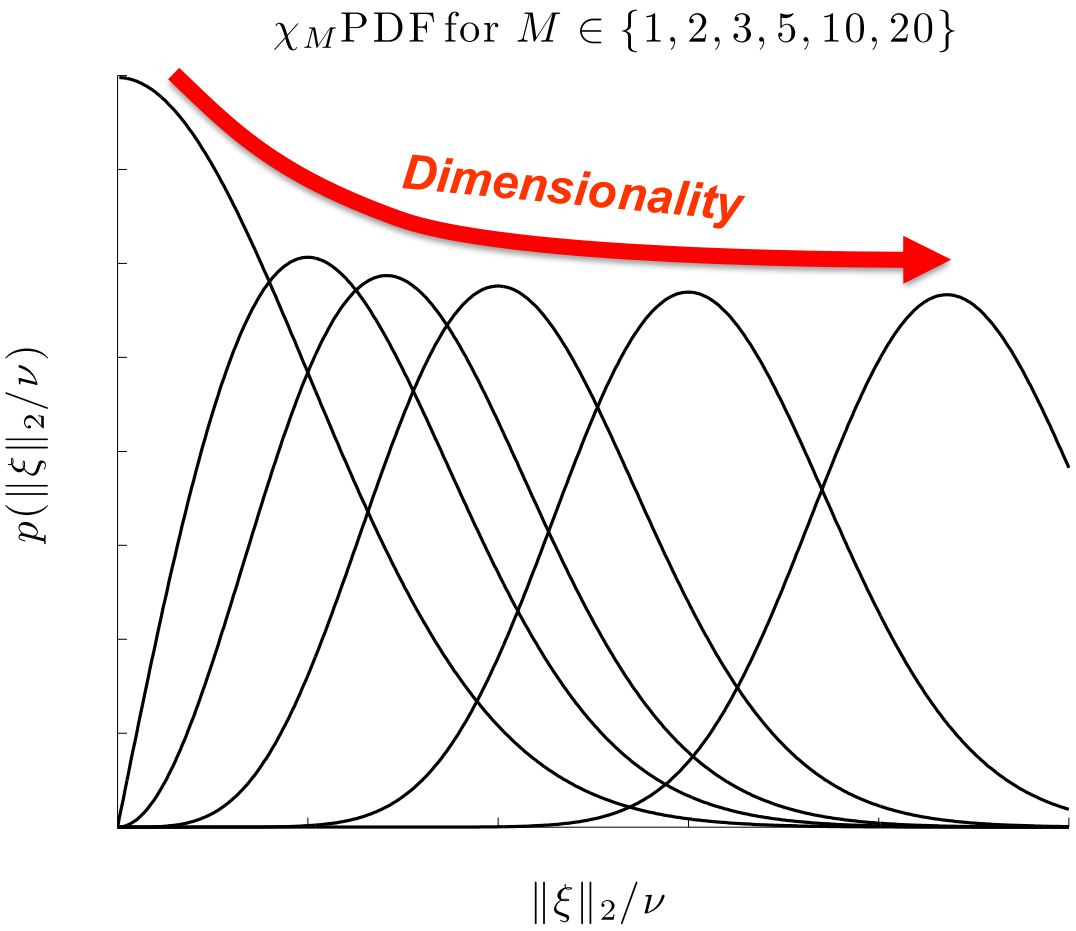
- The curse of dimensionality…
  - Empty space phenomenon
    (function approximation requires an exponential number of points w.r.t. *M*)
  - Norm concentration phenomenon
    (distances in a normal distribution have a chi distribution with *M* degrees of freedom)
- … and its unexpected consequences
  - A hypercube looks like a sea urchin (many spiky corners!)
  - Hypercube corners collapse towards the center in any projection
  - The volume of a unit hypersphere tends to zero
  - The sphere volume concentrates in a thin shell
  - Tails of a Gaussian get heavier than the central bell
  - Some points get very popular neighbors ('hubs')



3D views of 4D and 8D hypercubes



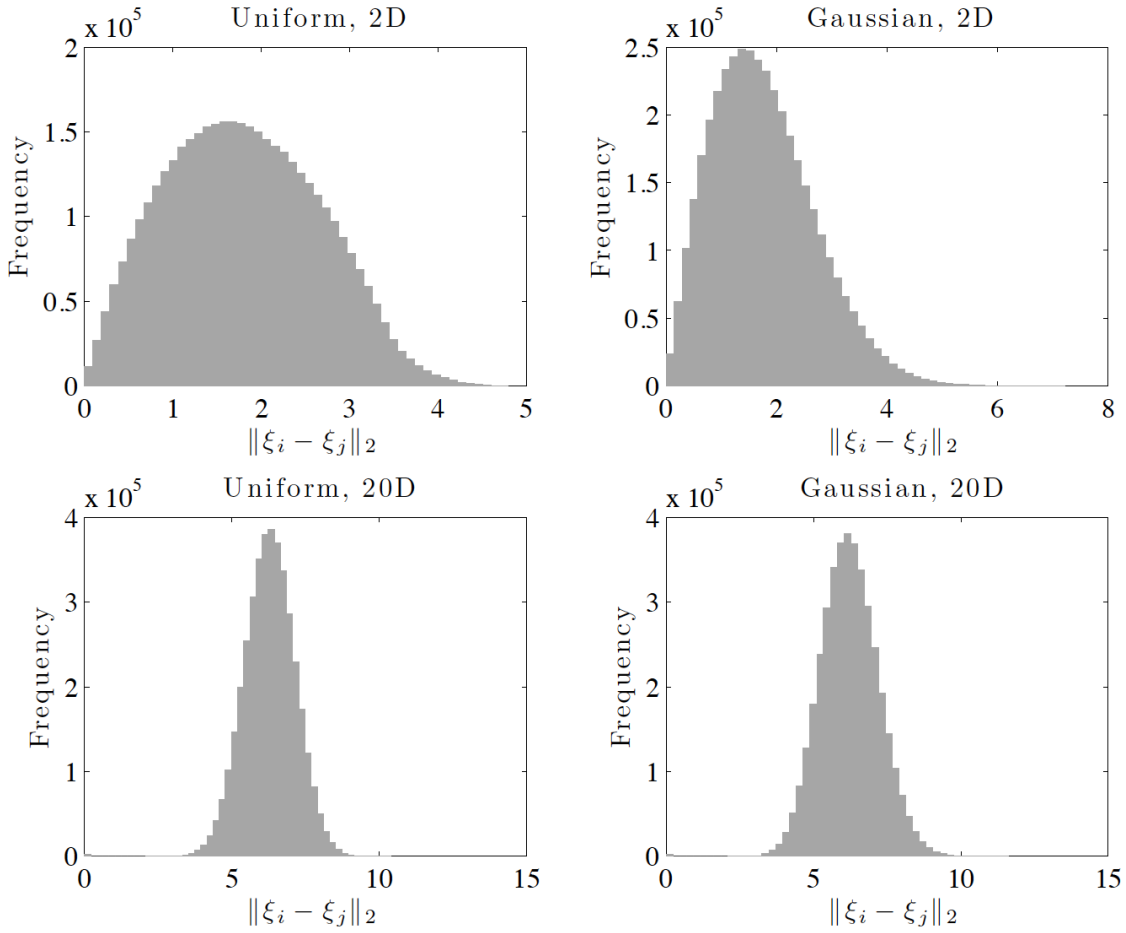
Distributions of Euclidean norms & distances



Euclidean norms of vectors with zero-mean unit variance Gaussian coordinates have a chi distribution with M DOFs ➔ the norms ***concentrate***

Distributions of Euclidean norms & distances

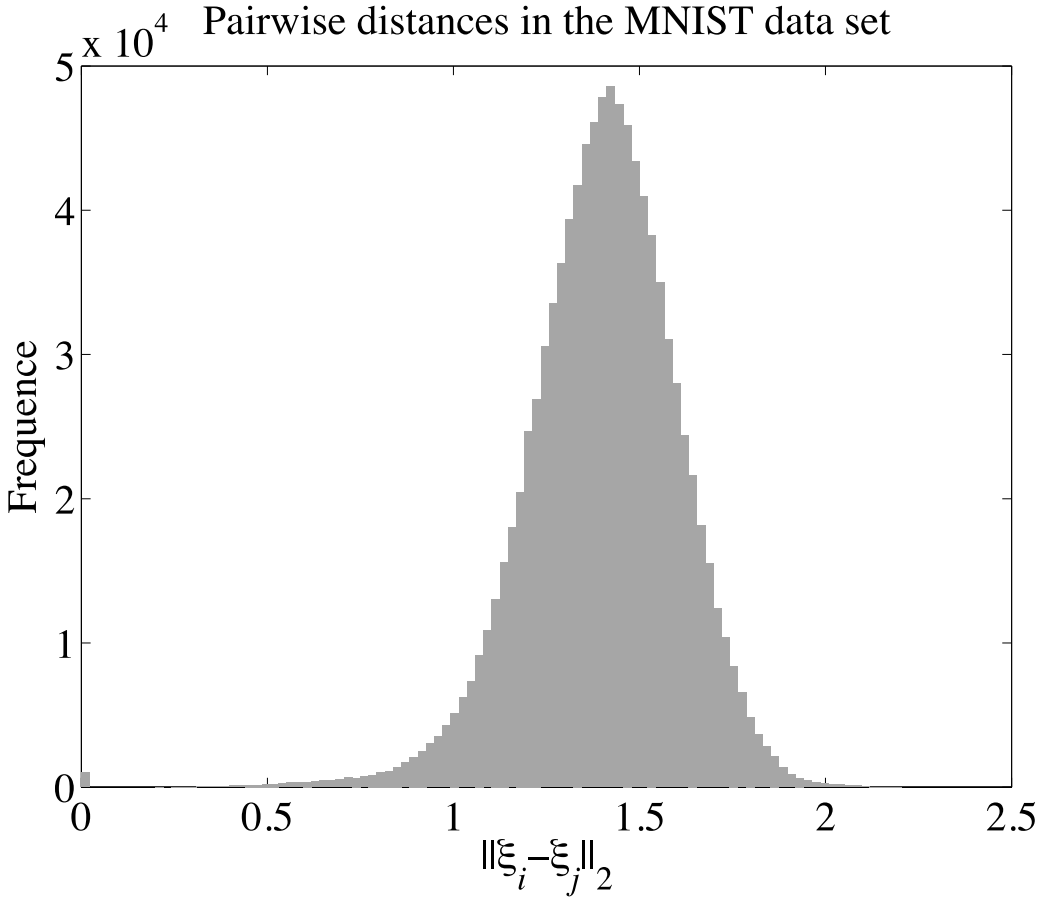
The data distribution barely affects the norm distribution



Distributions of Euclidean norms & distances

The data distribution barely affects the norm distribution

28-by-28 images
=> 784 dimensions



Distance preservation is hopeless…

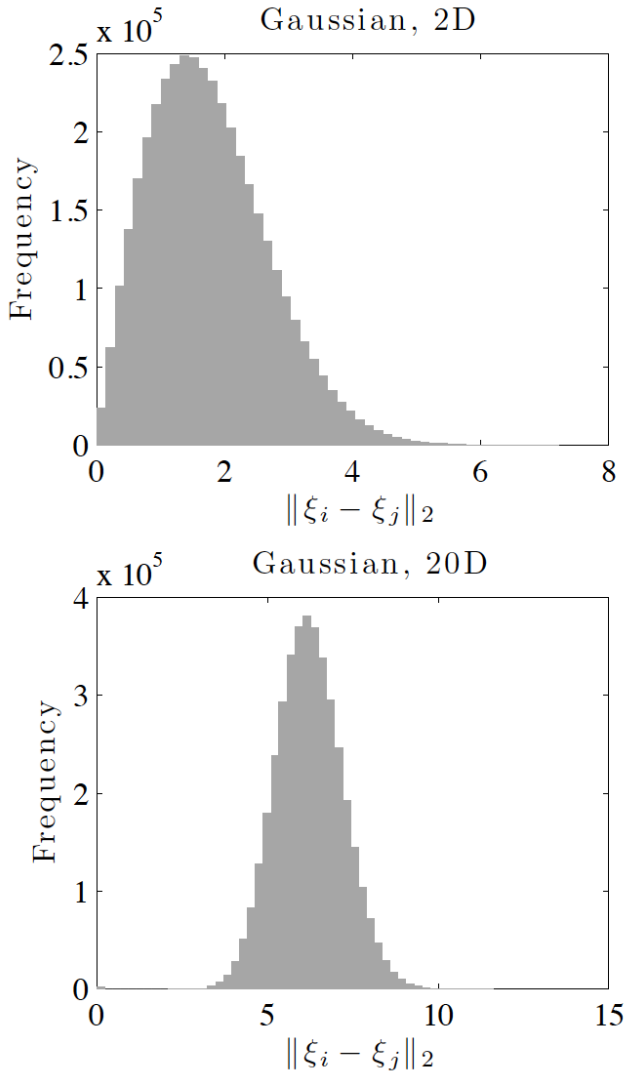
Do not compare apples and oranges!

$$\frac{1}{C}\sum_{i,j=1}^{N} w_{ij}\,(\delta_{ij} - d_{ij})^2$$

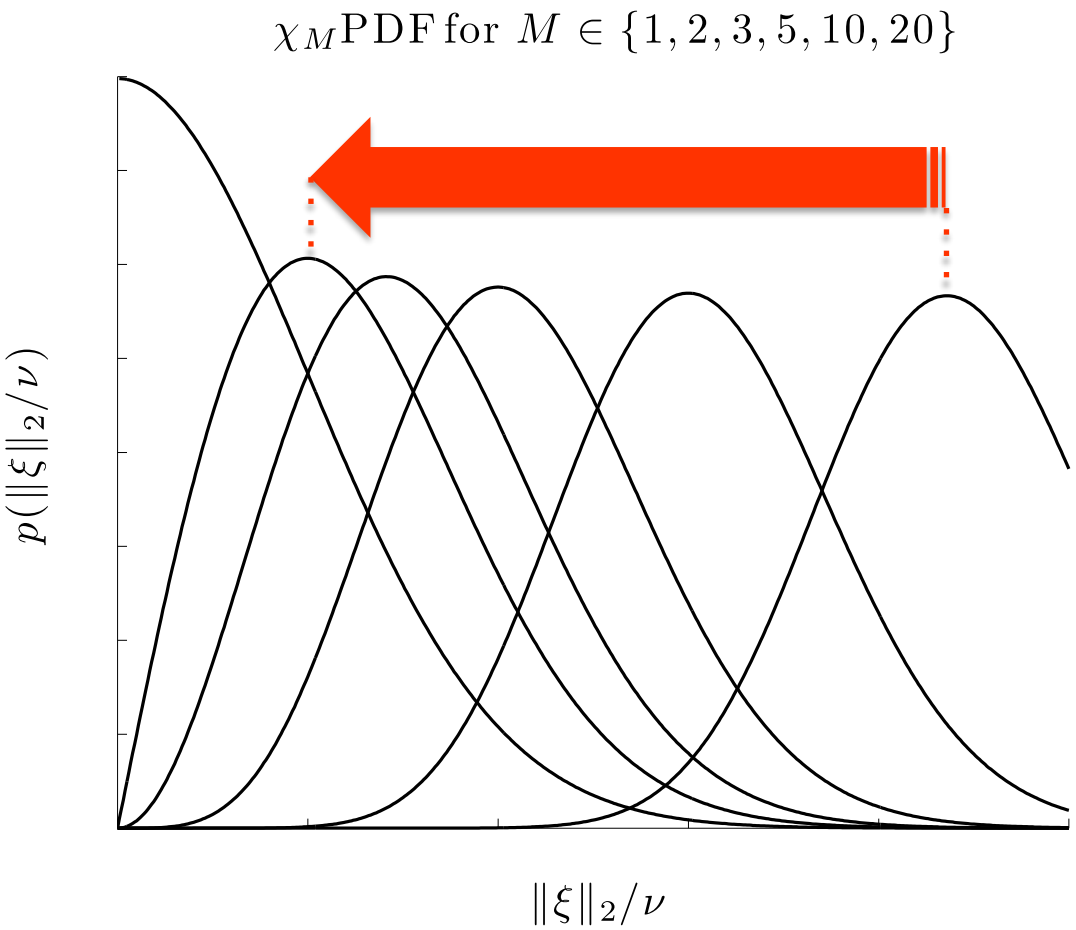
Low-dimensional

High-dimensional

?

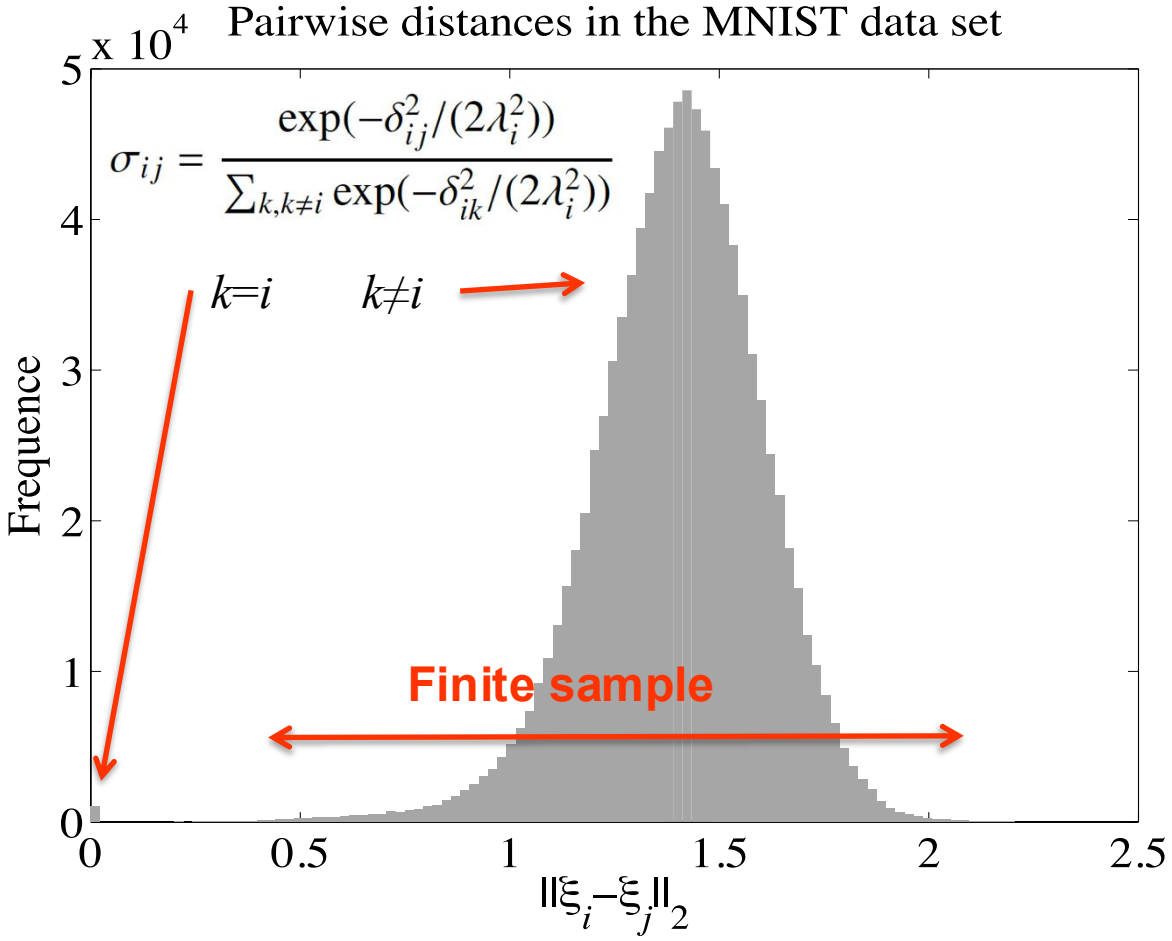


Curse of dimensionality: norm concentration



NLDR from HD to 2D requires a shift!

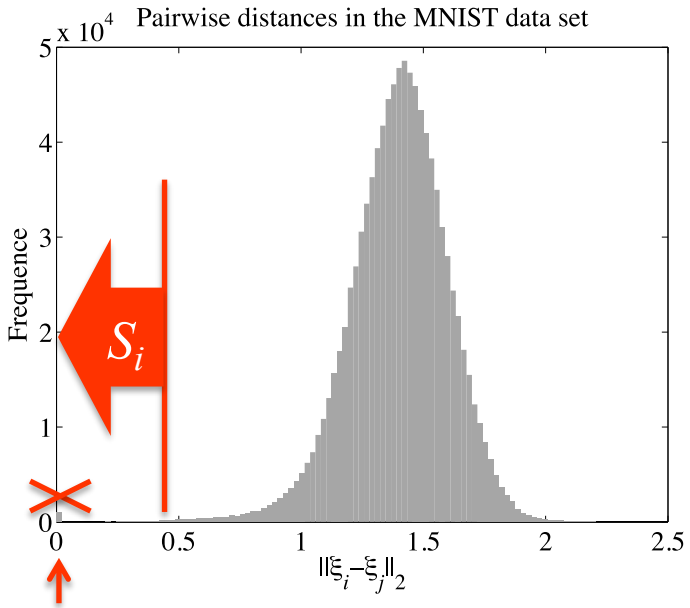
Curse of dimensionality: norm concentration



Shift-invariant similarities

$$\sigma_{ij} = \frac{\exp(-\delta_{ij}^2/(2\lambda_i^2))}{\sum_{k,k\neq i}\exp(-\delta_{ik}^2/(2\lambda_i^2))} = \sigma_{ij}\frac{\exp(S_i^2)}{\exp(S_i^2)} = \frac{\exp(S_i^2 - \delta_{ij}^2/(2\lambda_i^2))}{\sum_{k,k\neq i}\exp(S_i^2 - \delta_{ik}^2/(2\lambda_i^2))}$$

$$S_i \leq \min_{k,k\neq i} 2^{-1/2}\delta_{ik}/\lambda_i$$

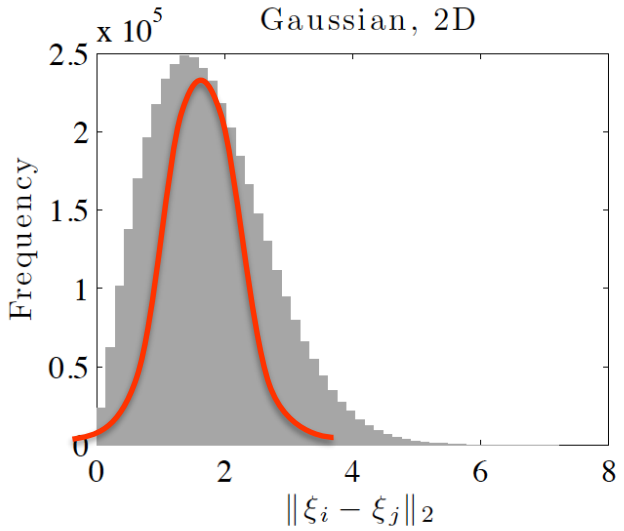


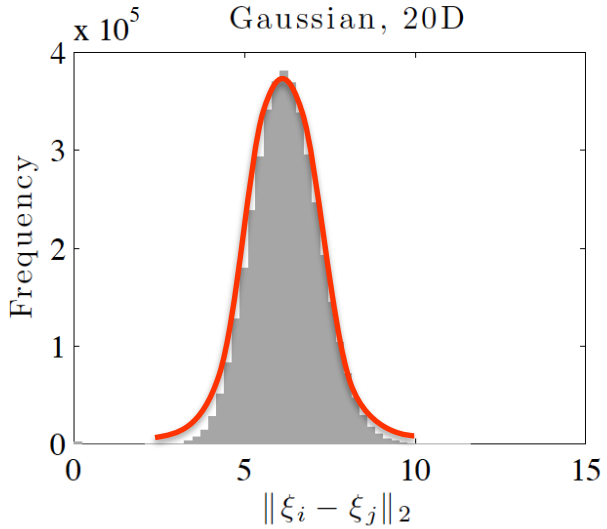
Similarity preservation is hopeful…

Appropriate fruit invariance makes apples and oranges comparable!

Low-dimensional

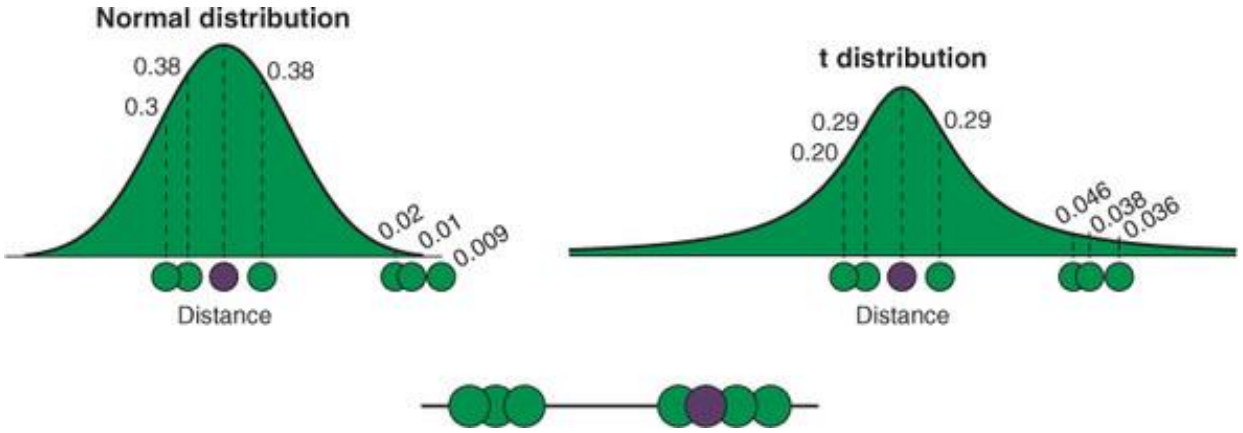
High-dimensional

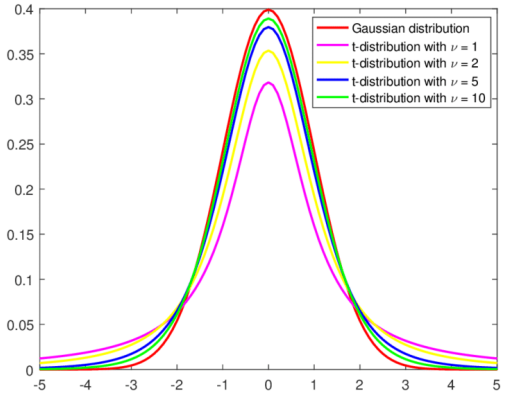




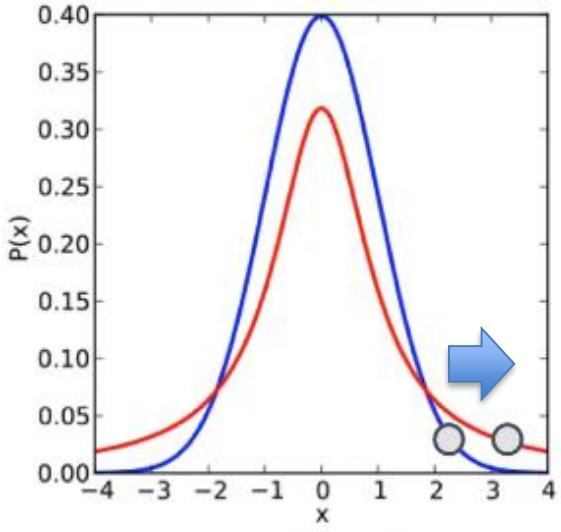
Exponential distance stretch

Making room for … clusters!





(Symmetric) SNE is
a limit case of *t*-SNE
for ∞ many DOFs $\nu = \infty$

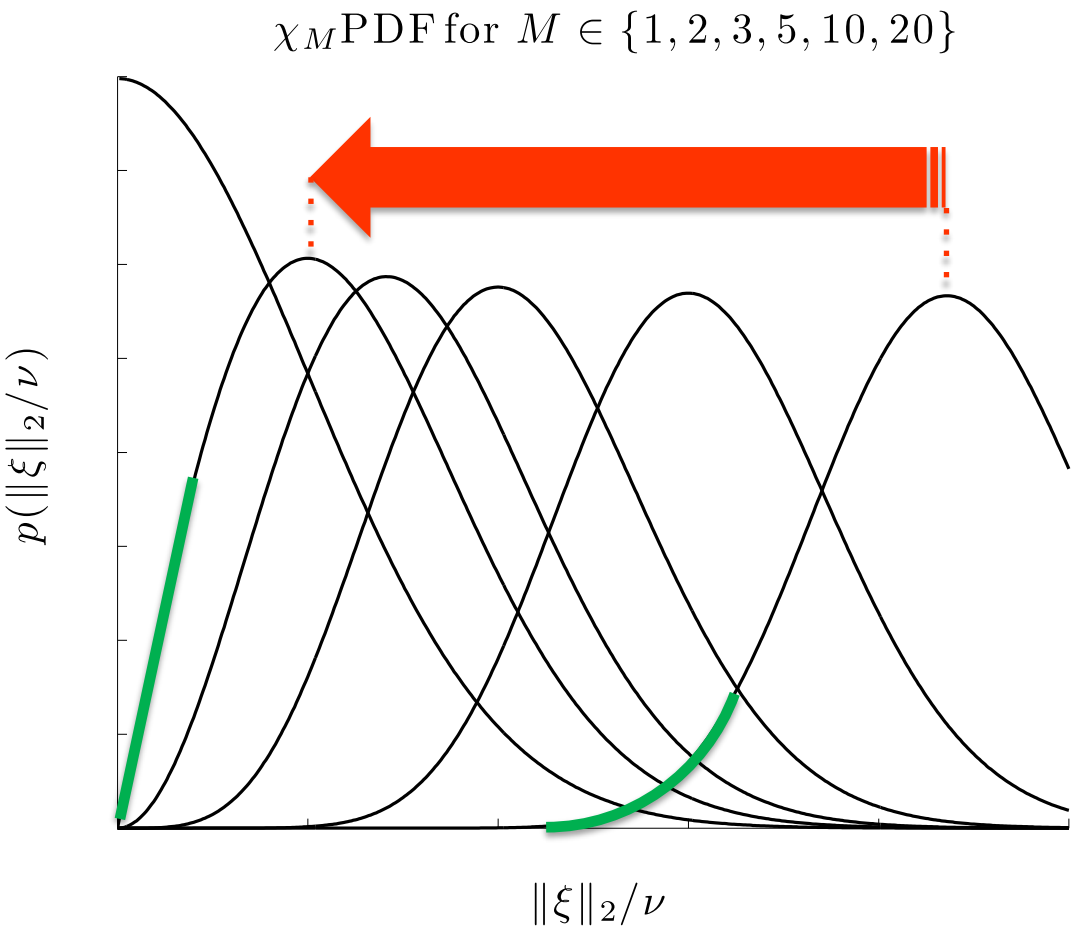


KL divergence between Gauss and Student
induces an exponential distance stretch from HD to LD

Inductive bias:
Empty space phenomenon in HD *vs* 'crowding problem' in LD

Shift invariance cared for empty left tail of distance distribution;
Gauss/Student discrepancy cares for its left ascent

Curse of dimensionality: norm concentration



NLDR from HD to 2D requires a shift!

… and some further transformation!

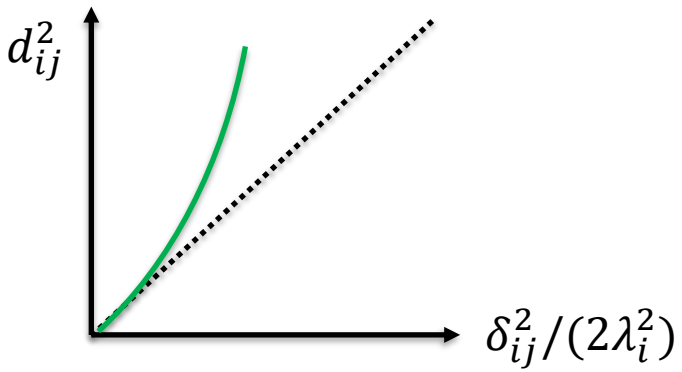
Exponential distance stretch

Making room for … clusters!

$$\sigma_{ij} = \frac{\exp(-\delta_{ij}^2/(2\lambda_i^2))}{\sum_{k,k\neq i}\exp(-\delta_{ik}^2/(2\lambda_i^2))} \qquad s_{ij} = \frac{\textbf{exp log}(1+d_{ij}^2)^{-1}}{\sum_{k,l,k\neq l}\textbf{exp log}(1+d_{kl}^2)^{-1}}$$

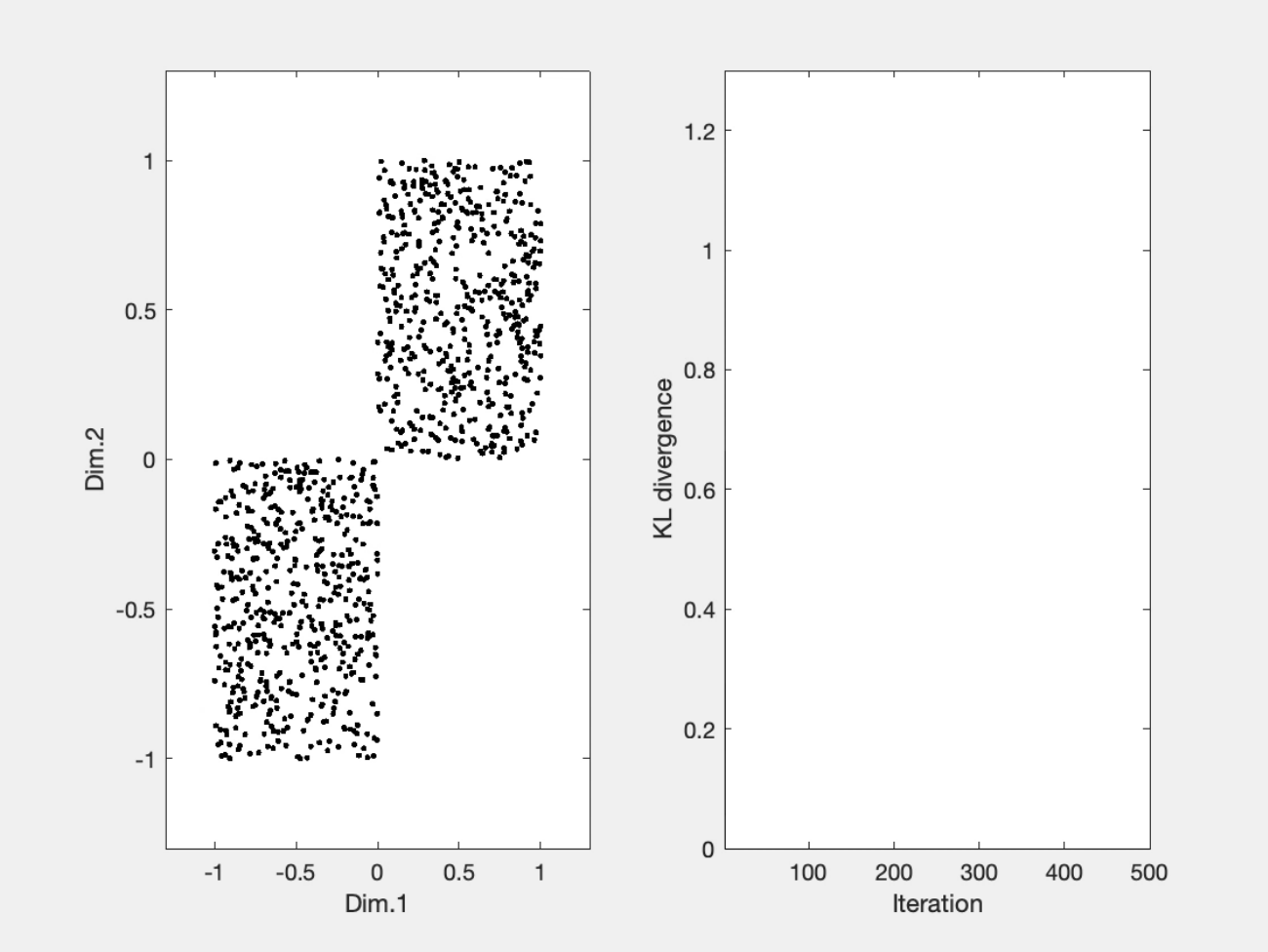
$$\delta_{ij}^2/(2\lambda_i^2) - S_i^2 \cong \log(1+d_{ij}^2)$$

$$d_{ij}^2 \cong \exp\left(\delta_{ij}^2/(2\lambda_i^2) - S_i^2\right) - 1$$



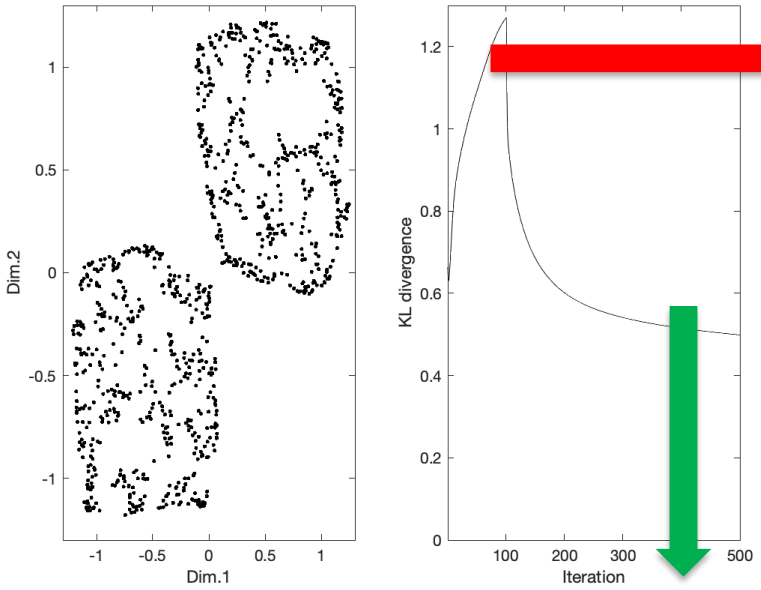
t-SNE performs DR…

Plus clustering with a strong inductive bias!



Rule out DR by running *t*-SNE from 2D to 2D (two uniform ‘clusters’)

Early exaggeration



EE: 'early exaggeration'
Preliminary optimization phase where s_{ij} tries to match $EE * \sigma_{ij}$ with $EE > 1$

Amplifies short-range attractive forces and long-range repulsive forces

Regular optimization phase where s_{ij} tries to match σ_{ij}

SIAM J. MATH. DATA SCI.
Vol. 1, No. 2, pp. 313–332

Clustering with t-SNE, Provably*

George C. Linderman† and Stefan Steinerberger‡

Abstract. t-distributed stochastic neighborhood embedding (t-SNE), a clustering and visualization method proposed by van der Maaten and Hinton in 2008, has rapidly become a standard tool in a number of natural sciences. Despite its overwhelming success, there is a distinct lack of mathematical foundations, and the inner workings of the algorithm are not well understood. The purpose of this paper is to prove that t-SNE is able to recover well-separated clusters; more precisely, we prove that t-SNE in the "early exaggeration" phase, an optimization technique proposed by van der Maaten and Hinton [*J. Mach. Learn. Res.*, 9 (2008), pp. 2579–2605] and van der Maaten [*J. Mach. Learn. Res.*, 15 (2014), pp. 3221–3245], can be rigorously analyzed. As a byproduct, the proof suggests novel ways for setting the exaggeration parameter α and step size h. Numerical examples illustrate the effectiveness of these rules: in particular, the quality of embedding of topological structures (e.g., the swiss roll) improves. We also discuss a connection to spectral clustering methods.

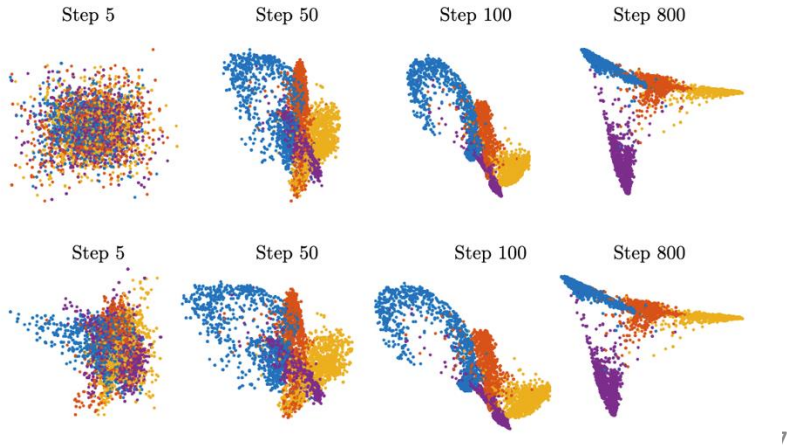


Figure 4.2. *Early exaggeration via t-SNE with $\alpha \sim n/3, h = 1$ (top, parameter selection via guideline) and iterations of the spectral method (bottom).*

The dense jungle of meta-parameters

Belkina et al. Nat.comm. 2019

Typical *t*-SNE function call:

```
Y = tsne(X,dim=2,pxt=30,dof=1,itr=1000,EE=12,init='PCA',…)
```

Dimension (target dimensionality)

Perpexity (neighborhood size)

Degrees of freedom (*t* distribution)

iterations (gradient descent)

Early exaggeration factor

PCA initialization (30 components)

… and many others!

(EE plateau length, learning rate,

Barnes-Hut angle, …)

ARTICLE

https://doi.org/10.1038/s41467-019-13055-y OPEN

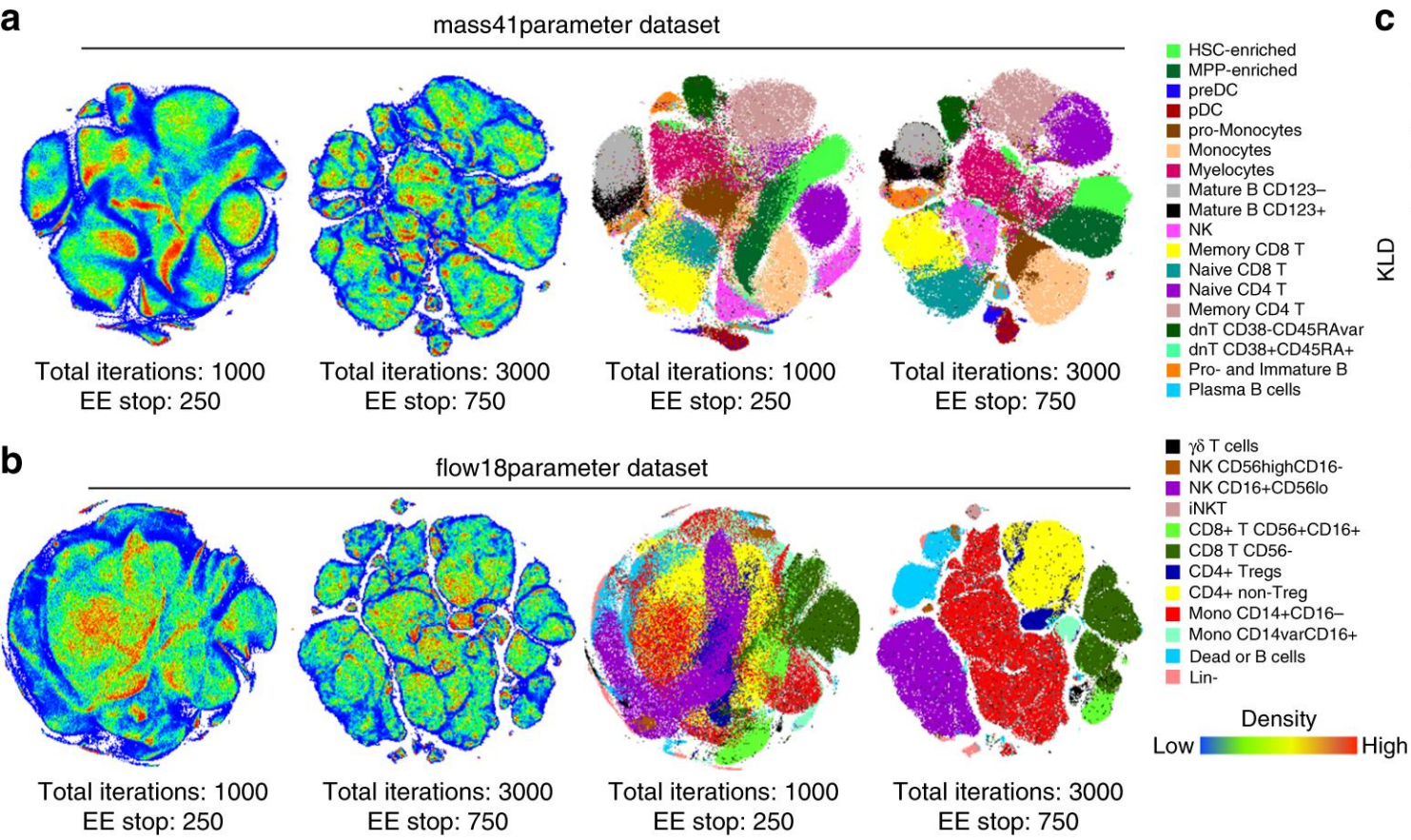
Automated optimized parameters for T-distributed stochastic neighbor embedding improve visualization and analysis of large datasets

Anna C. Belkina[1,2*], Christopher O. Ciccolella[3], Rina Anno[4], Richard Halpert[5], Josef Spidlen[5] & Jennifer E. Snyder-Cappione[2,6]

Accurate and comprehensive extraction of information from high-dimensional single cell datasets necessitates faithful visualizations to assess biological populations. A state-of-the-art algorithm for non-linear dimension reduction, t-SNE, requires multiple heuristics and fails to produce clear representations of datasets when millions of cells are projected. We develop opt-SNE, an automated toolkit for t-SNE parameter selection that utilizes Kullback-Leibler divergence evaluation in real time to tailor the early exaggeration and overall number of gradient descent iterations in a dataset-specific manner. The precise calibration of early exaggeration together with opt-SNE adjustment of gradient descent learning rate dramatically improves computation time and enables high-quality visualization of large cytometry and transcriptomics datasets, overcoming limitations of analysis tools with hard-coded parameters that often produce poorly resolved or misleading maps of fluorescent and mass cytometry data. In summary, opt-SNE enables superior data resolution in t-SNE space and thereby more accurate data interpretation.

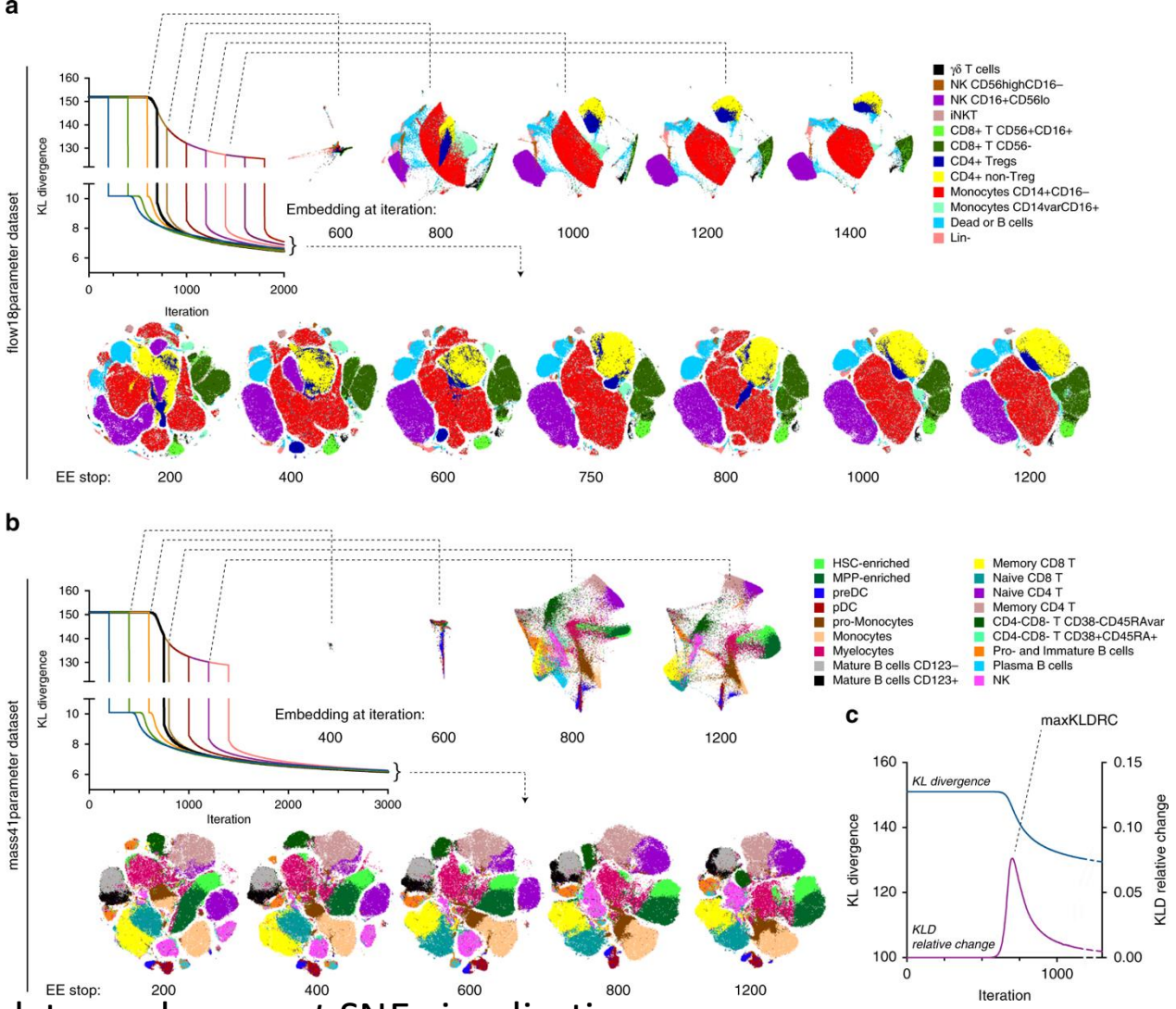
The dense jungle of meta-parameters

Belkina et al. Nat.comm. 2019



Performance of Barnes-Hut t-SNE implementation for cytometry data visualization.

Standard (1000 iterations) and extended (3000 iterations) embeddings of mass cytometry (**a**) or flow cytometry (**b**) data are presented as heatmap density plots (left) or color-coded population overlays based on ground-truth classification of single cell in the datasets (right). **c** KLD change over iteration time of gradient descent for standard 1000 iterations (red line) or extended 3000 iterations (black line) embeddings of mass41parmeter dataset. Representative examples of multiple runs with varying seed values are shown.



Effect of EE plateau phase on *t*-SNE visualization.

EE was stopped after varying number of iterations and embedding visualization was examined at several intermediate timepoints and in the end of embedding for flow cytometry (total of 2000 iterations), (**a)** and mass cytometry (total of 3000 iterations).

b Graphs showing KLD change over iteration time are color-labeled to distinguish curves corresponding to experiment perturbations, with black line indicating the run with the shortest EE but uninterrupted plateau.

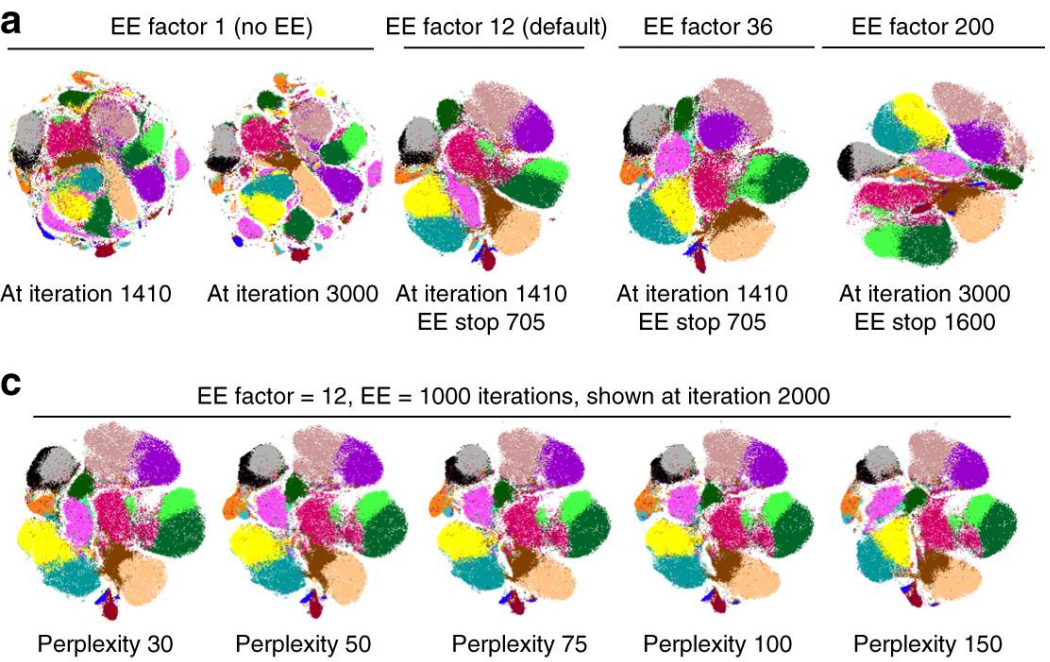
t-SNE maps are annotated with color-coded population overlays based on ground-truth classification of single cell in the datasets.

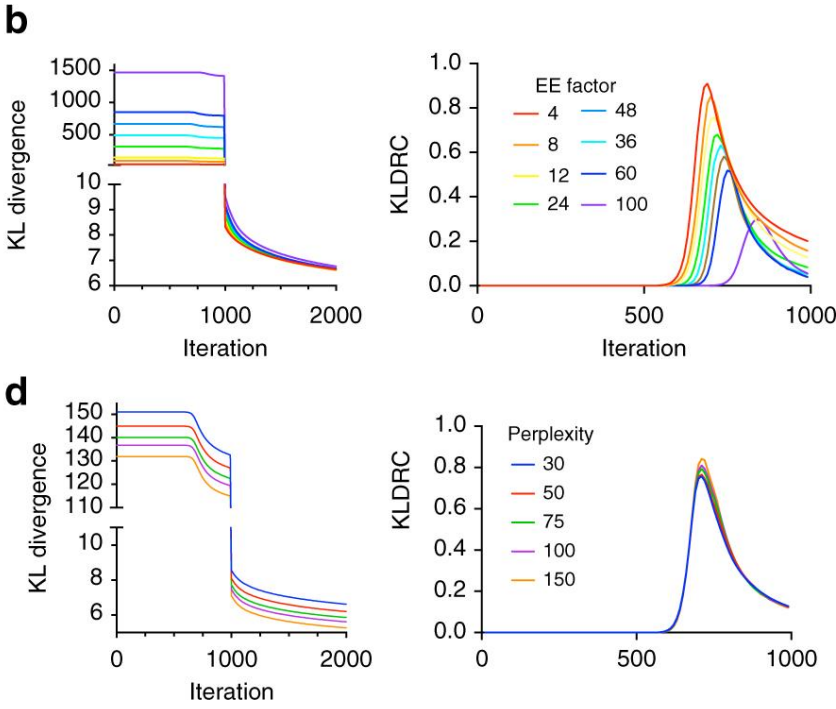
c KLD and KLD relative change plotted against iteration time for the mass41parameter embedding.

All embeddings were generated with standard BH-tSNE implementation and representative examples of multiple runs with varying seed values are shown.

The dense jungle of meta-parameters

Belkina et al. Nat.comm. 2019





Effects of perplexity and EE factor adjustments on *t*-SNE visualization of cytometry data.

a, **b** KLD, KLDRC, and t-SNE biaxial plots generated with varying EE factor values.

c, **d** KLD, KLDRC, and t-SNE biaxial plots generated with varying perplexity.

Graphs showing KLD and KLDRC change over iteration time are color-labeled to distinguish curves corresponding to experiment perturbations. Color overlays on t-SNE plots correspond to cell type classes labeled as in Figs. 1, 2.

Representative examples of multiple runs with varying seed values are shown.

Conclusions about *t*-SNE

Ongoing investigation (2008-2024 so far…)

Claimed (advertised statistics)

- Dimensionality reduction (according to the authors)
- Stochastic neighborhoods in KL divergences:
  - ‘probability to be a neighbor’, ‘entropic affinities’,
  - Gauss in HD *vs* Student in LD, to cope with a ‘crowding problem’

Observed (hidden statistics)

- Clustering (according to the users)
- Distance transformations to cope with:
  - local density variations
  - HD/LD-discrepant distance concentration

Conclusions about *t*-SNE

Ongoing investigation (2008-2024 so far…)

- *t*-SNE is a hybrid method:
  dimensionality reduction with inductive bias: emphasize/over-express clusters
  (magnify existing cluster gaps… and show spurious clusters?)

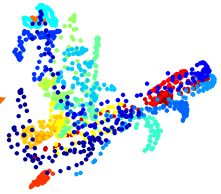
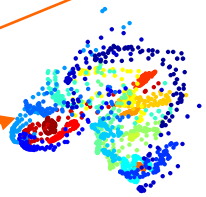
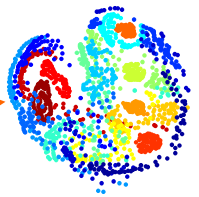
- Caveats/drawbacks:
  - Non-convex optimization with non-deterministic initialization,
    stochasticity in the results, ever-expanding embedding
  - Non-parametric method
  - Inter-dependent metaparameters:
    perplexity, Student *t* degrees of freedom, early exaggeration, …

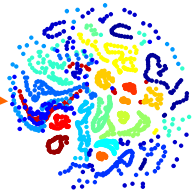
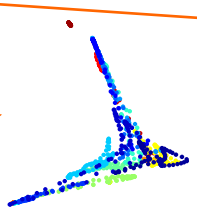
- Alternatives?
  Deep learning of auto-encoders?
  They are parametric, invertible, scalable
  but still not outperforming t-SNE in 2D visualization

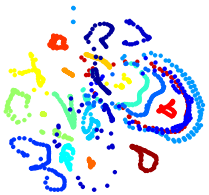
- Current trend: accelerated (approximate) neighbor embedding for big data

(NL)DR through time…

1901 Principal component analysis (PCA)
1938 Classical multidimensional scaling (CMDS)
1962 Nonmetric MDS (NMDS)
1969 Sammon's nonlinear mapping (NLM)
1982 Self-organising maps (SOMs)
1991 Auto-encoder (back prop.)
1993 Curvilinear component analysis (CCA)
1996 Kernel PCA
1998 Isomap
2000 Locally linear embedding (LLE)
2002 Laplacian eigenmaps (LE)
2002 Stochastic neighbour embedding (SNE)
2006 Auto-encoder (deep learning)
2008 Student-distributed SNE (*t*-SNE)
2010 Neighbour retrieval & vis. (NeRV)
2012 Jensen-Shannon Embedding (JSE)
2014 Multiscale JSE (Ms JSE)
2018 UMAP, *tt*-SNE, Ms *t*-SNE
2019 FI*t*-SNE, NE with missing data
2022 Fast Multiscale NE

Spectral
Neural network
Reconstruction error
Distance preservation
Similarity preservation

Global & Local DR

Distinguishing HD & LD spaces…

HD

LD

Global

Local

Global

Local

Spectral (dense):
PCA/CMDS, nMDS, Isomap

Spectral (sparse):
KPCA, LE, LLE

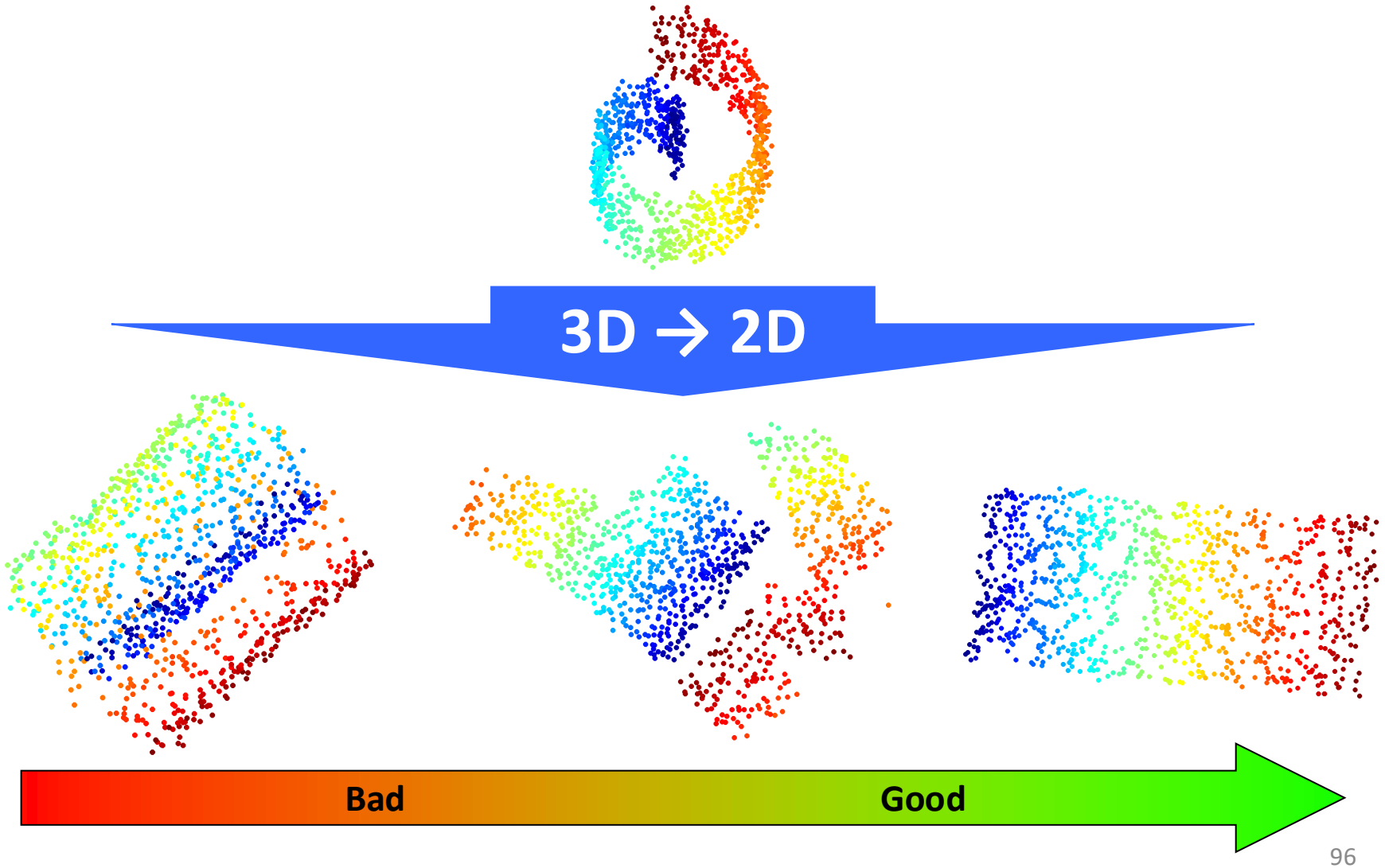
Stress-based MDS:
Sammon mapping

Neighbor embbedding:
SNE

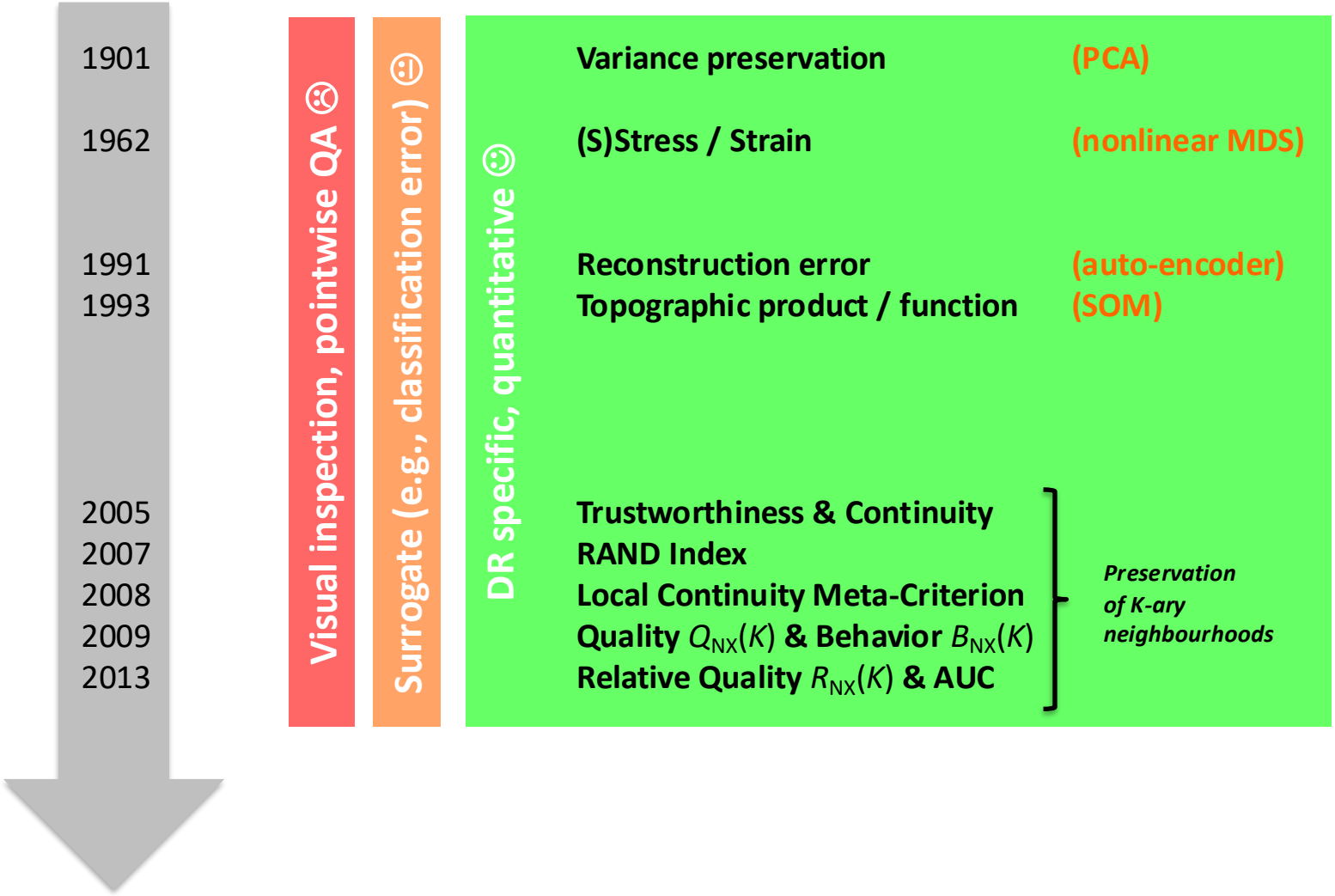
SOM
CCA

Neighbor embedding (NE):
(*t*-)SNE, UMAP, etc.

Quality Assessment: Intuition



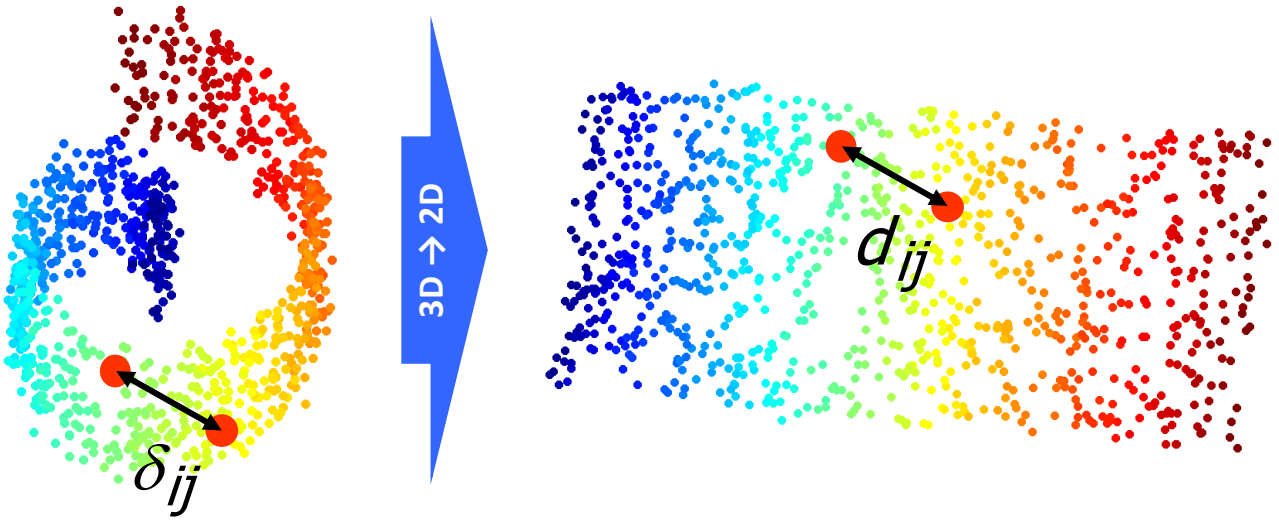
DR QA through time…



Distances, Ranks, and Neighbourhoods

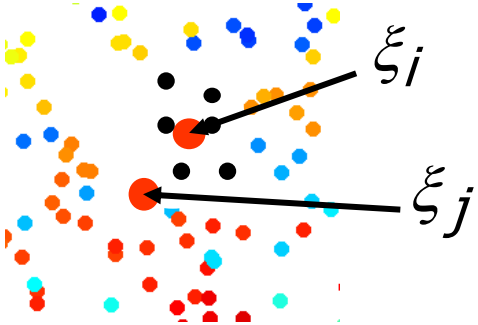
- Distances: δ_{ij} denotes the distance from $\boldsymbol{\xi}_i$ to $\boldsymbol{\xi}_j$
  d_{ij} is the distance from $\mathbf{x}_i$ to $\mathbf{x}_j$

$$\boldsymbol{\Xi} = [\boldsymbol{\xi}_i]_{1 \leq i \leq N} \qquad \mathbf{X} = [\mathbf{x}_i]_{1 \leq i \leq N}$$

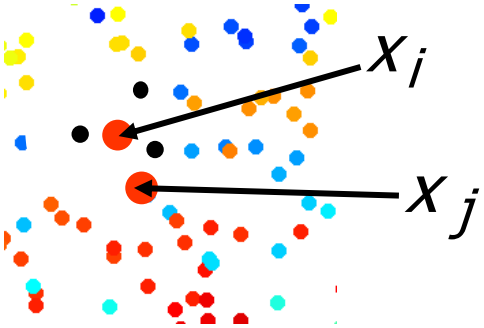


Distances, Ranks, and Neighbourhoods

- Ranks: $\rho_{ij} = |\{k : \delta_{ik} < \delta_{ij} \text{ or } (\delta_{ik} = \delta_{ij} \text{ and } k < j)\}|$
  $r_{ij} = |\{k : d_{ik} < d_{ij} \text{ or } (d_{ik} = d_{ij} \text{ and } k < j)\}|$



$\rho_{ij} = 6$



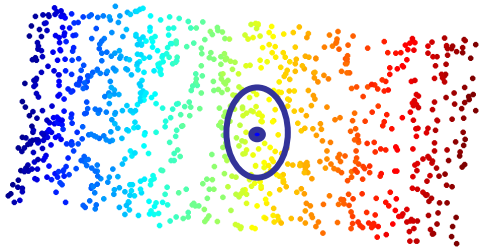
$r_{ij} = 4$

- Neighborhoods: sets of indexes of black points (up to neighbor *K*)

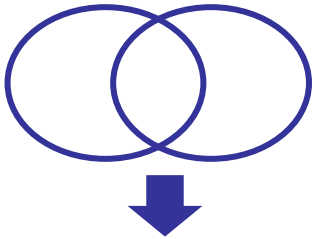
$$\nu_i^K = \{j : 1 \le \rho_{ij} \le K\}$$
$$n_i^K = \{j : 1 \le r_{ij} \le K\}$$

Multi-scale quality assessment

K-ary neighbourhood in LD

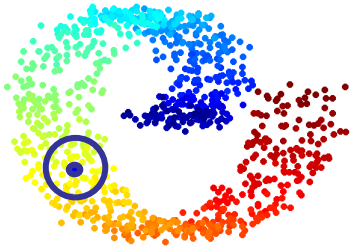


n_i^K



ν_i^K

K-ary neighbourhood in HD



$$Q_{\mathrm{NX}}(K) = \sum_{i=1}^{N} \frac{|\nu_i^K \cap n_i^K|}{KN}$$

Average agreement of the K-ary neighbourhoods

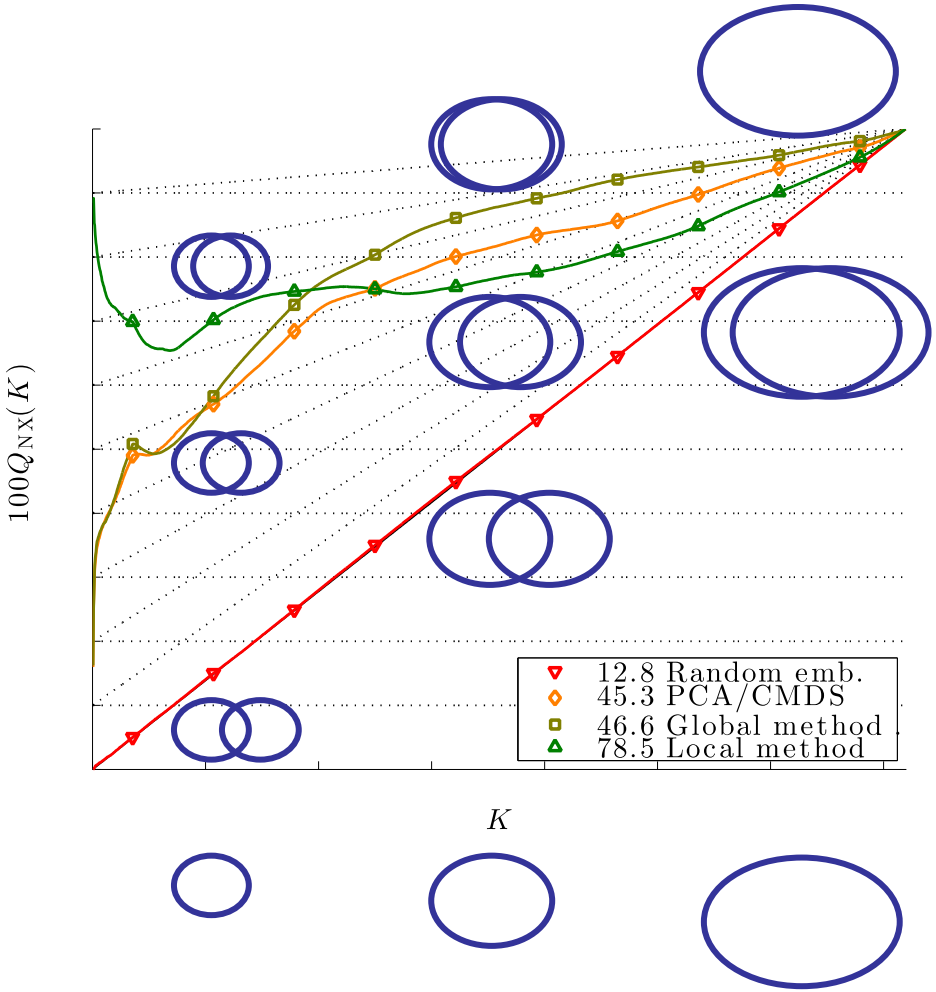
Due to DR, a point can:

- keep faithful neighbors,
- loose missing neighbors,
- gain spurious neighbors

LD \ HD	Near	Far
Near	☺	☹
Far	😐	☺

Multi-scale quality assessment

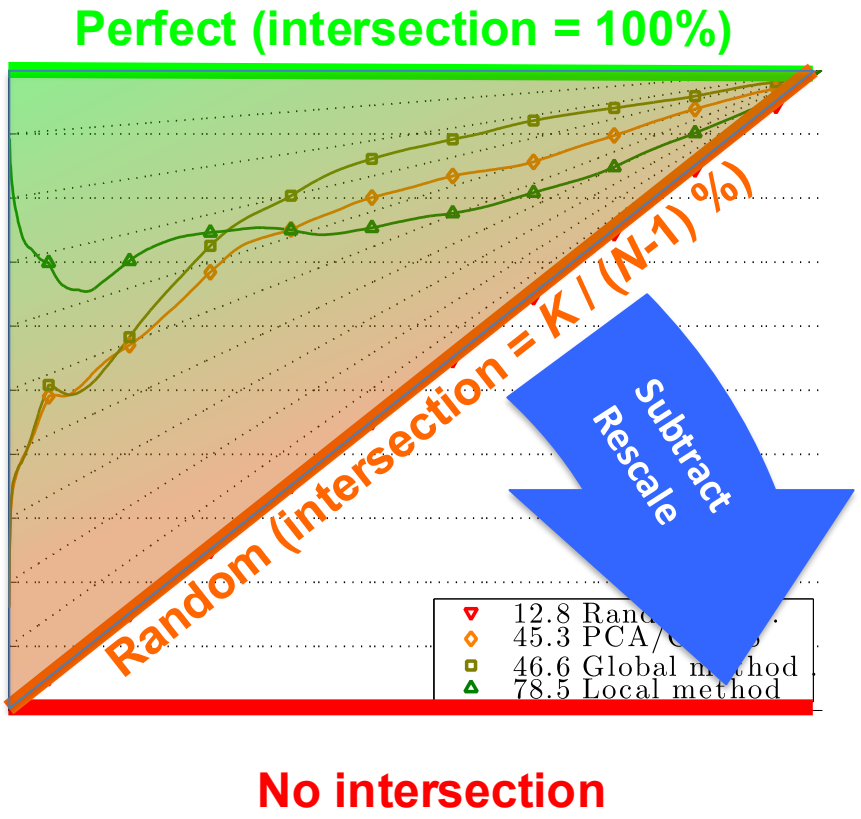
$$Q_{\mathrm{NX}}(K) = \sum_{i=1}^{N} \frac{|\nu_i^K \cap n_i^K|}{KN}$$



Multi-scale quality assessment

$$Q_{\mathrm{NX}}(K) = \sum_{i=1}^{N} \frac{|\nu_i^K \cap n_i^K|}{KN}$$

$$R_{\mathrm{NX}}(K) = \frac{(N-1)Q_{\mathrm{NX}}(K) - K}{N-1-K}$$

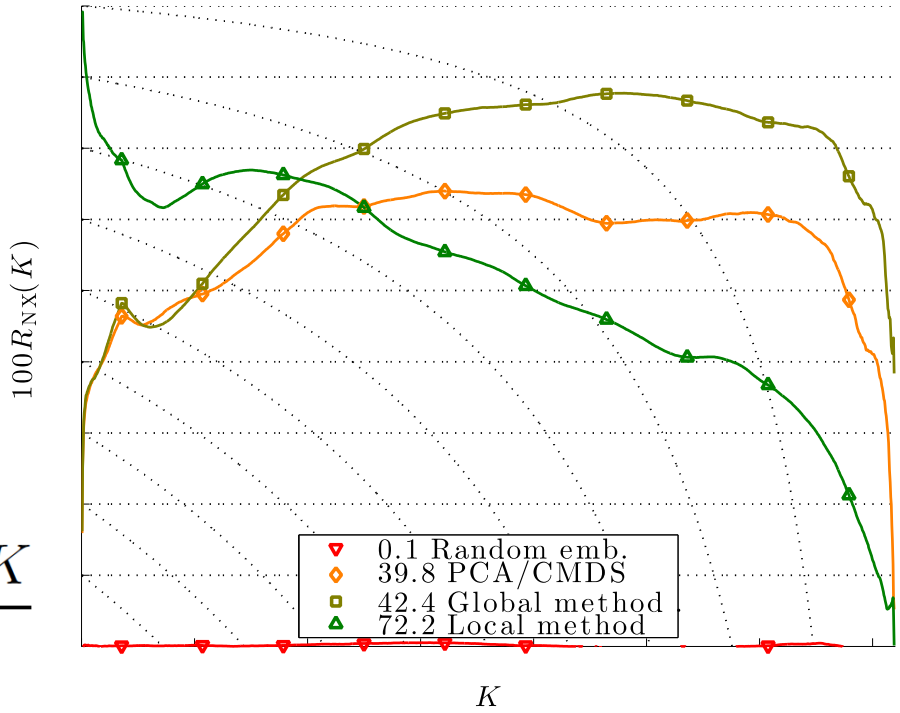


Relative quality between a perfect embedding and a random one

Multi-scale quality assessment

$$Q_{\mathrm{NX}}(K) = \sum_{i=1}^{N} \frac{|\nu_i^K \cap n_i^K|}{KN}$$

$$R_{\mathrm{NX}}(K) = \frac{(N-1)Q_{\mathrm{NX}}(K) - K}{N-1-K}$$

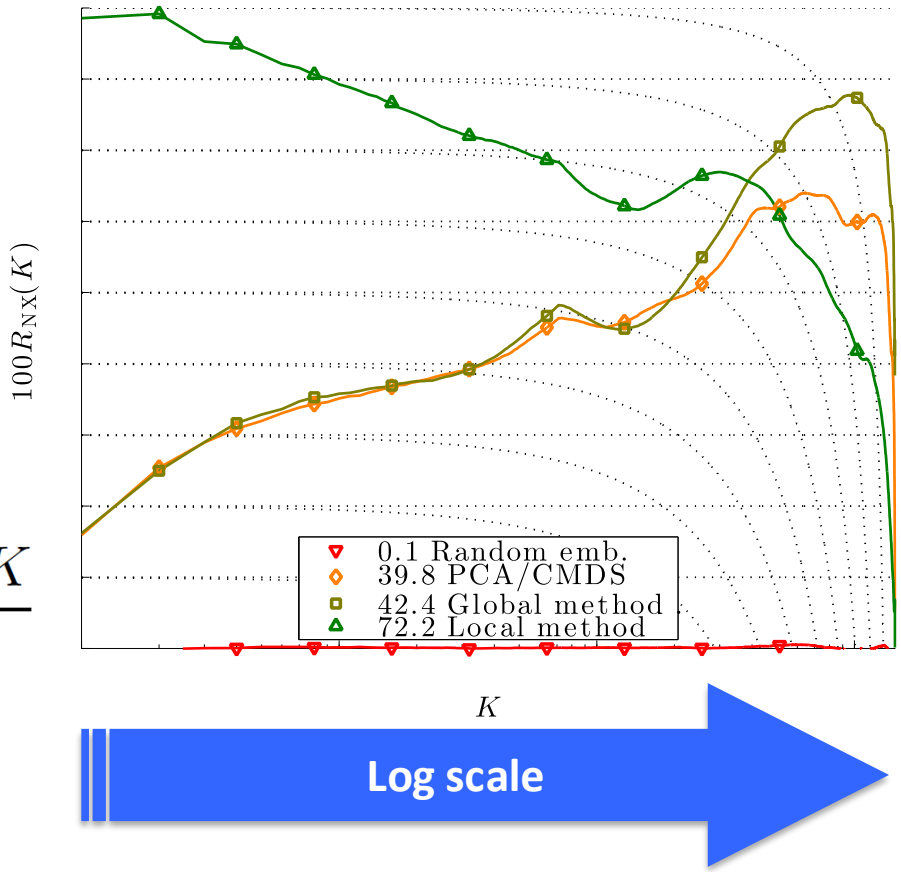


Relative quality between a perfect embedding and a random one

Multi-scale quality assessment

$$Q_{\mathrm{NX}}(K) = \sum_{i=1}^{N} \frac{|\nu_i^K \cap n_i^K|}{KN}$$

$$R_{\mathrm{NX}}(K) = \frac{(N-1)Q_{\mathrm{NX}}(K) - K}{N-1-K}$$

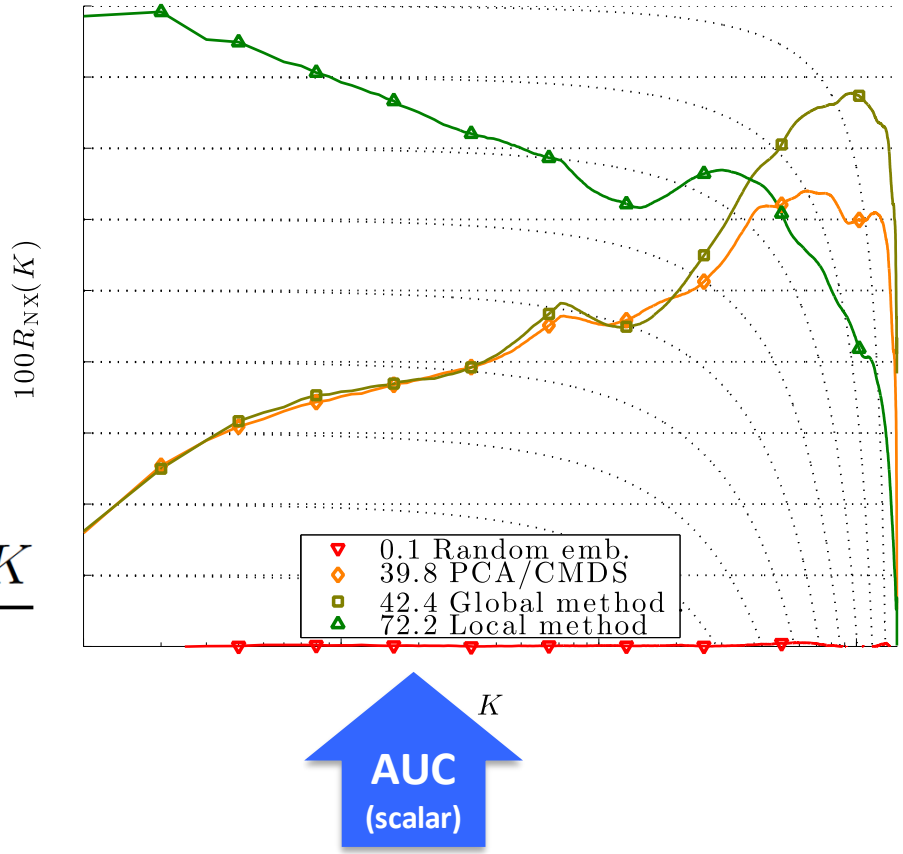


Exponential relationship between the size K and radius r of a K-ary neighborhood in a uniform P-dimensional distribution: K proportional to r^P

Multi-scale quality assessment

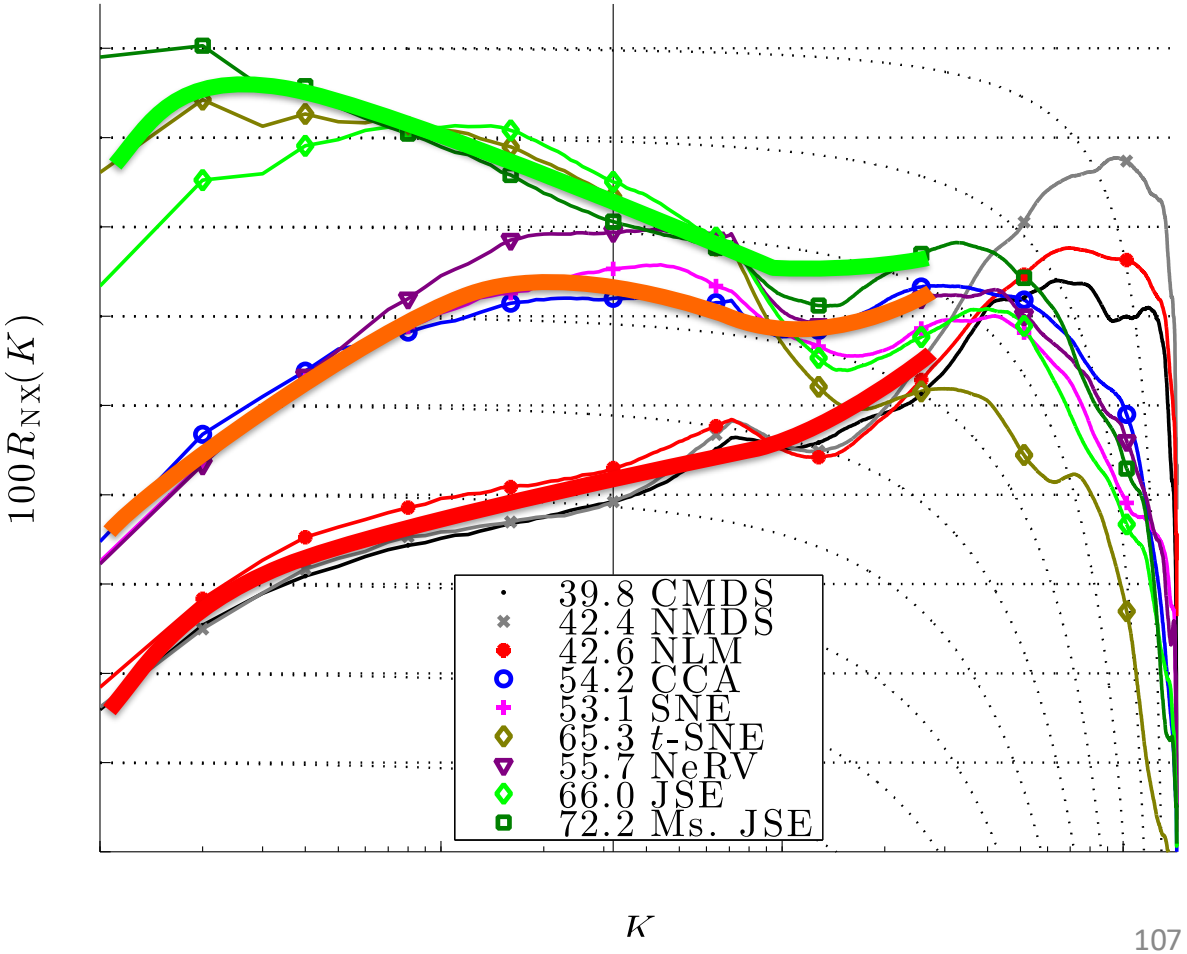
$$R_{\mathrm{NX}}(K) = \frac{(N-1)Q_{\mathrm{NX}}(K) - K}{N-1-K}$$

$$\mathrm{AUC} = \frac{\sum_{K=1}^{N-2} R_{\mathrm{NX}}(K)/K}{\sum_{K=1}^{N-2} 1/K}$$



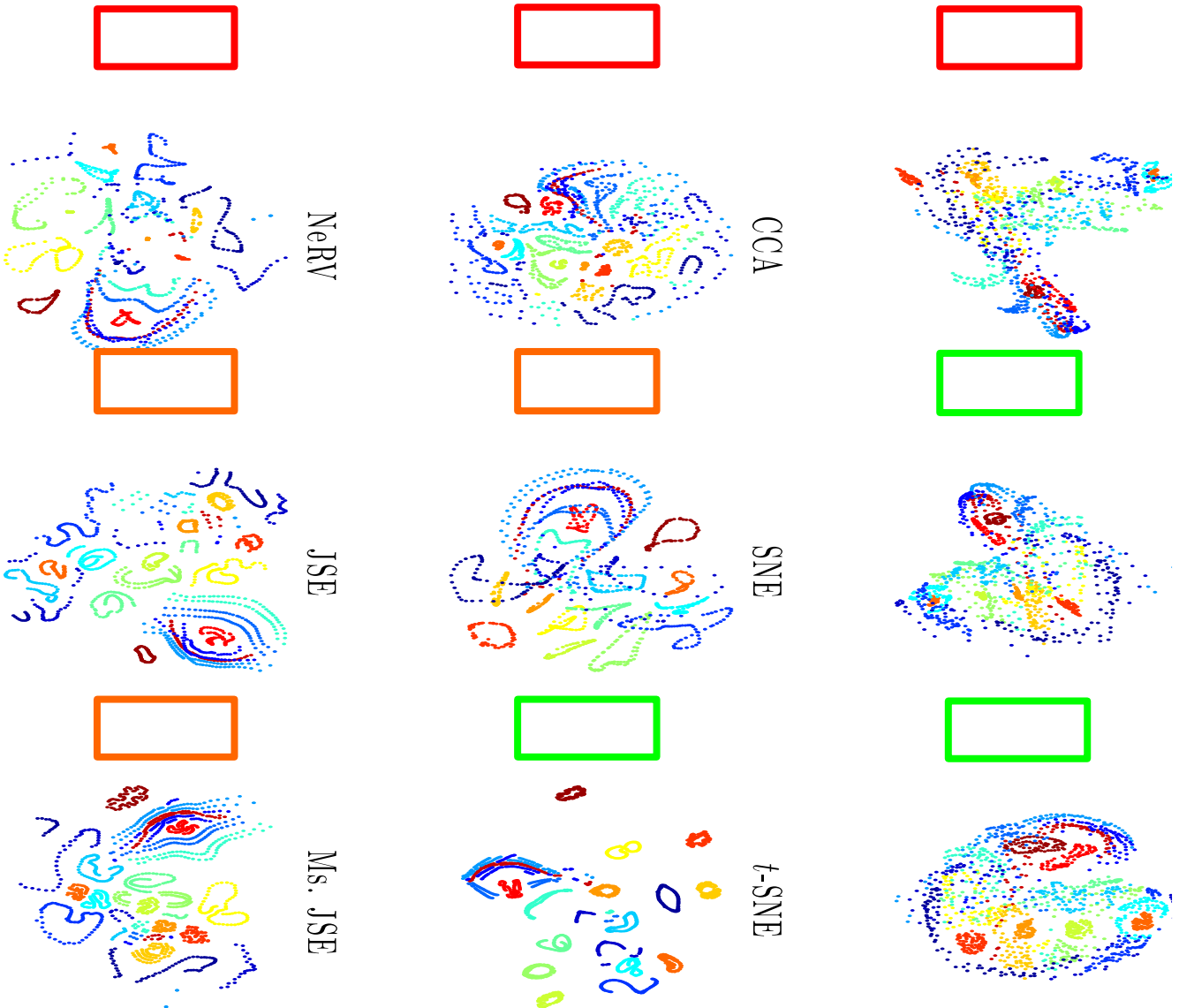
Results for COIL-20

Quality Assessment



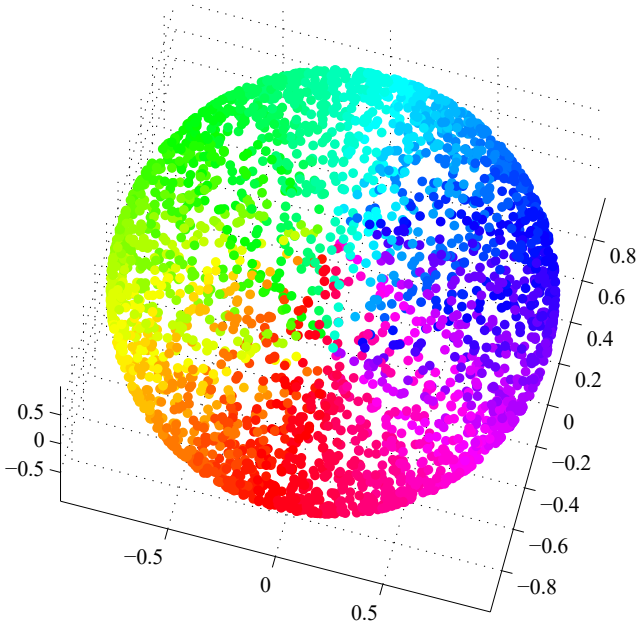
Results for COIL-20

Embeddings

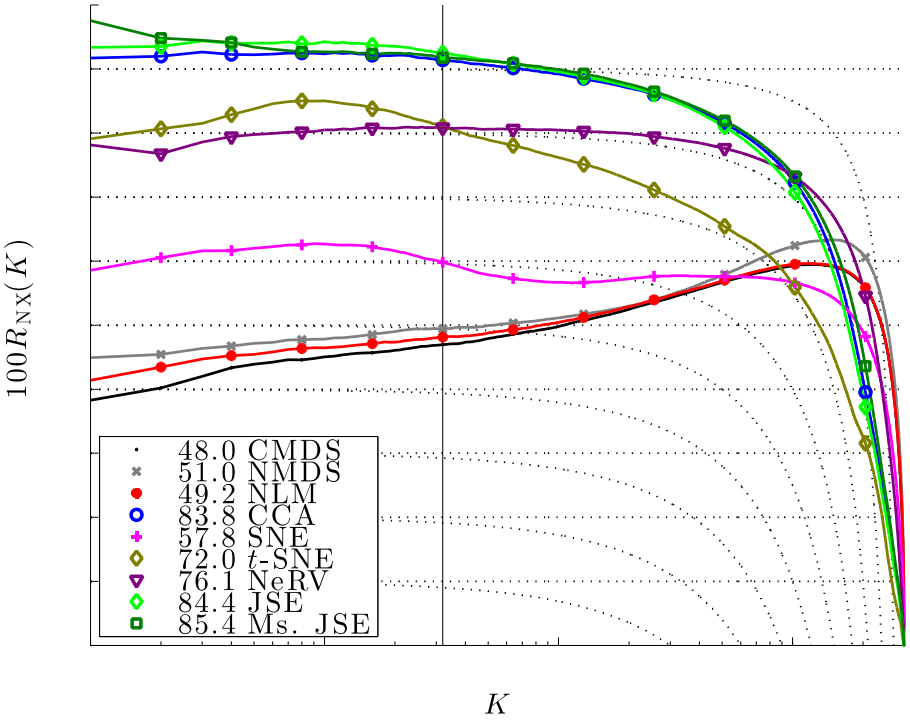


Results for 3D Sphere

quality assessment

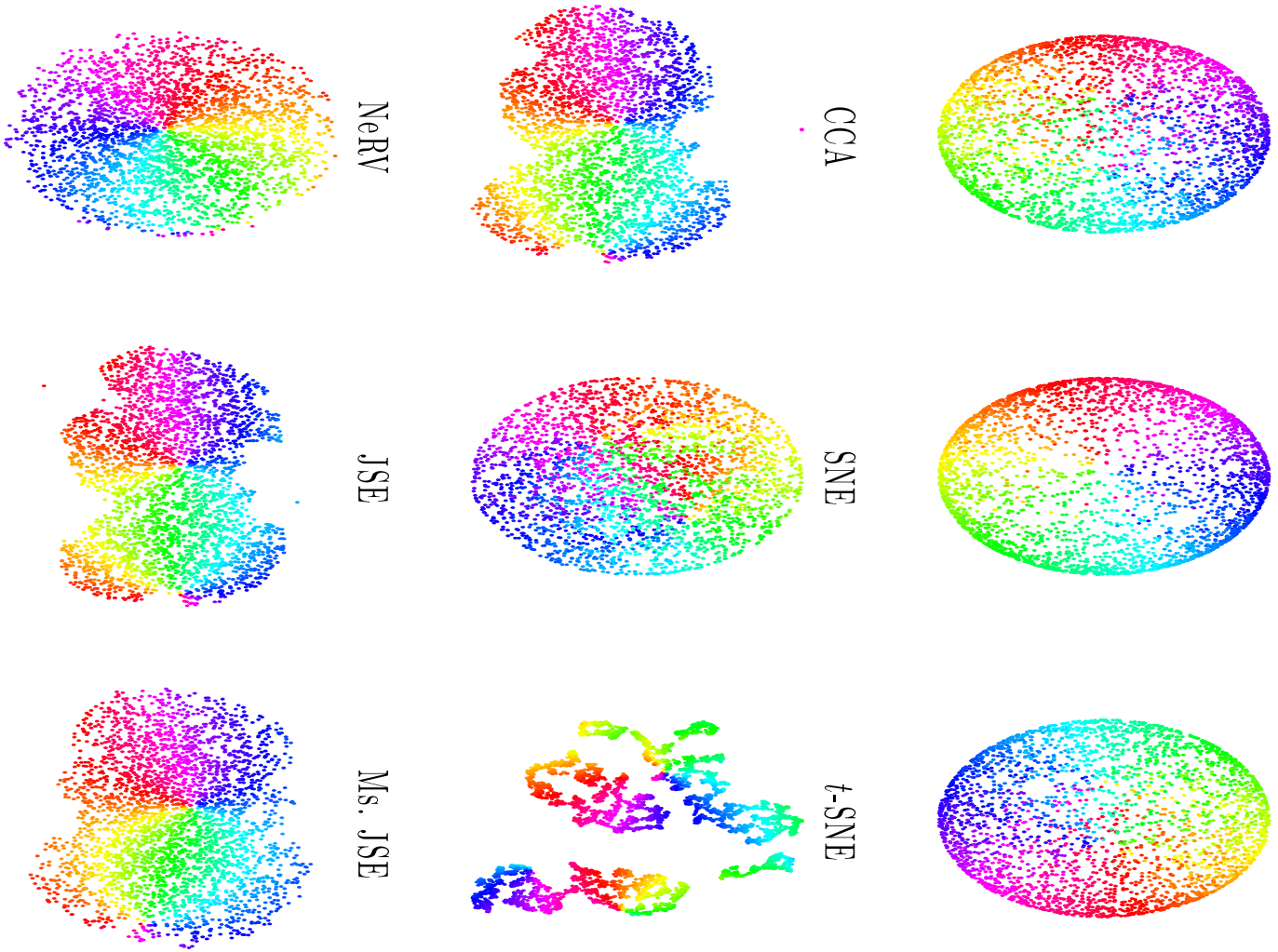


3 dimensions, $N = 3000$



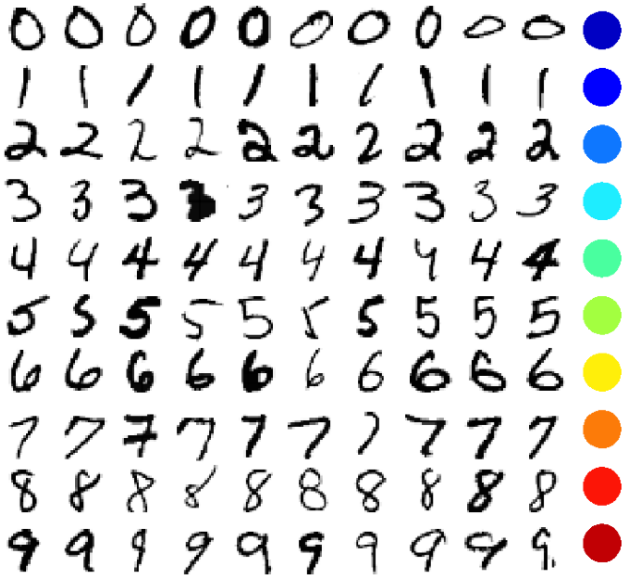
Results for 3D Sphere

embeddings

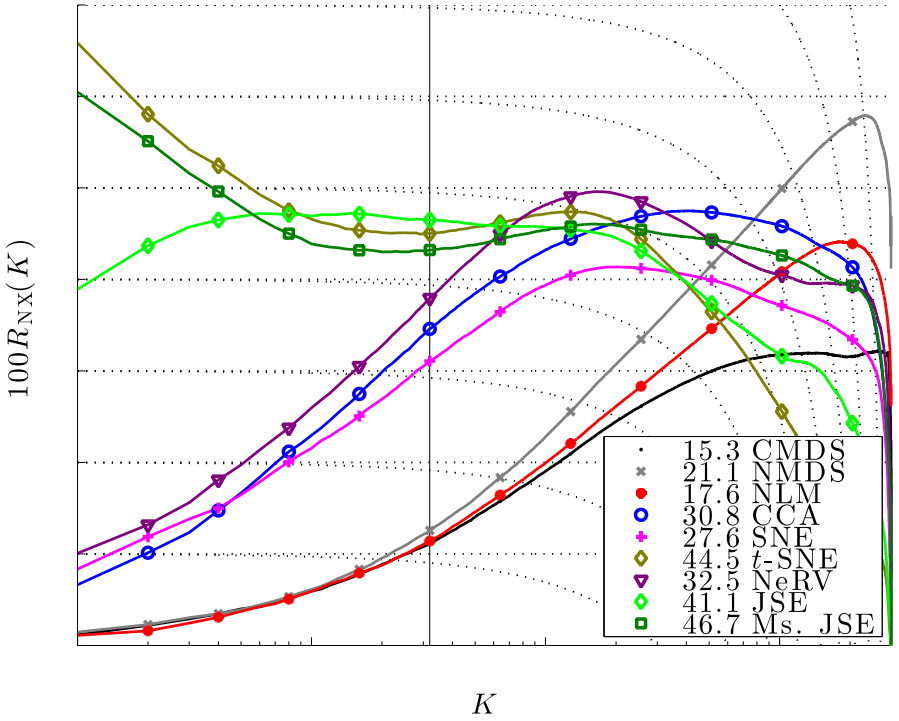


Results for MNIST digits

quality assessment

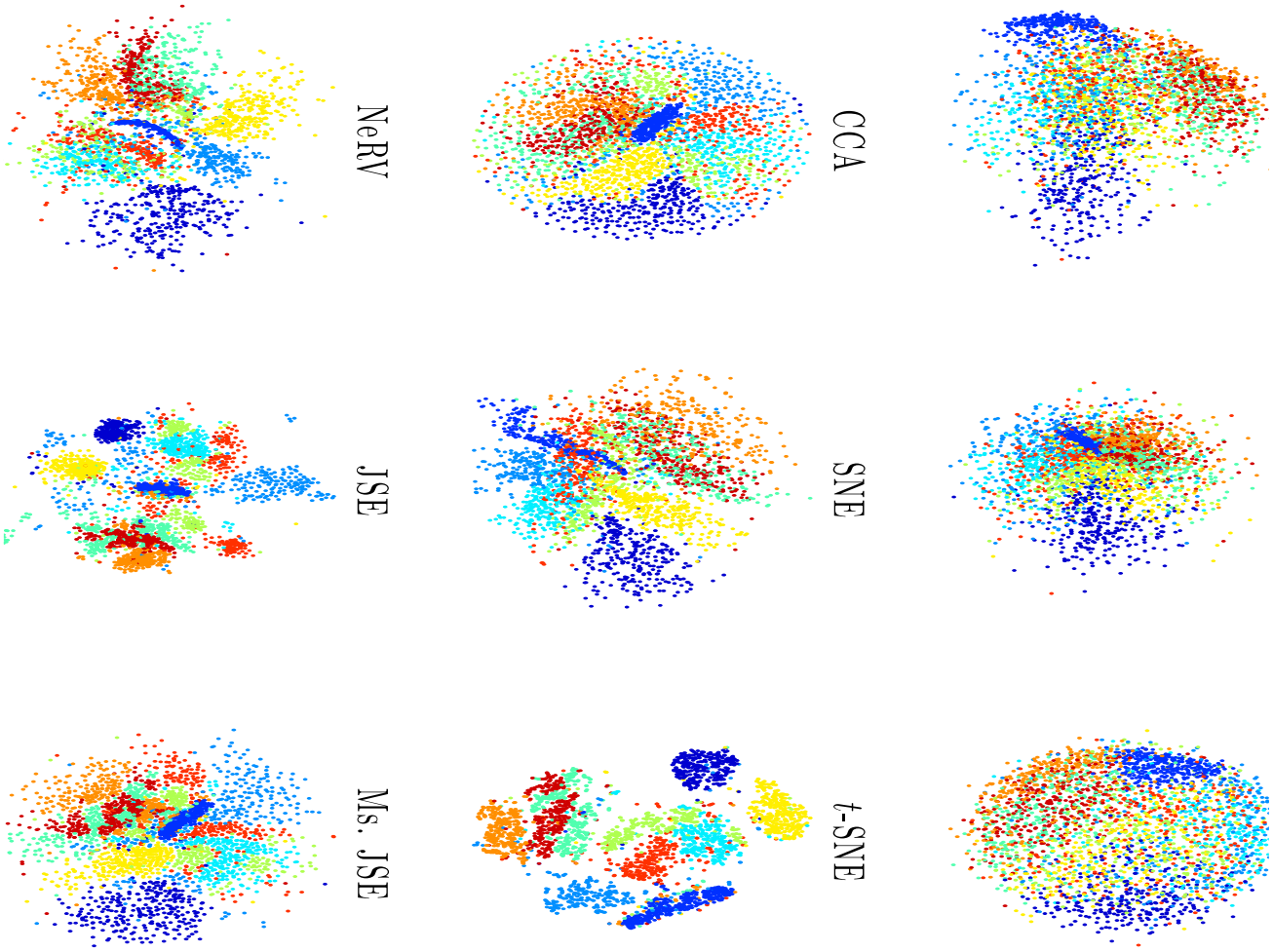


784 dimensions, $N = 3000$



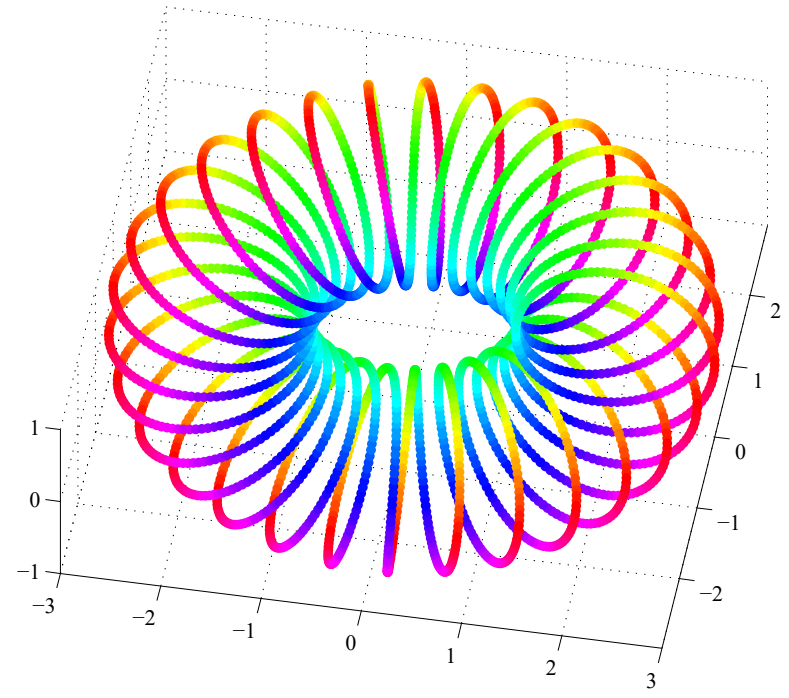
Results for MNIST digits

embeddings

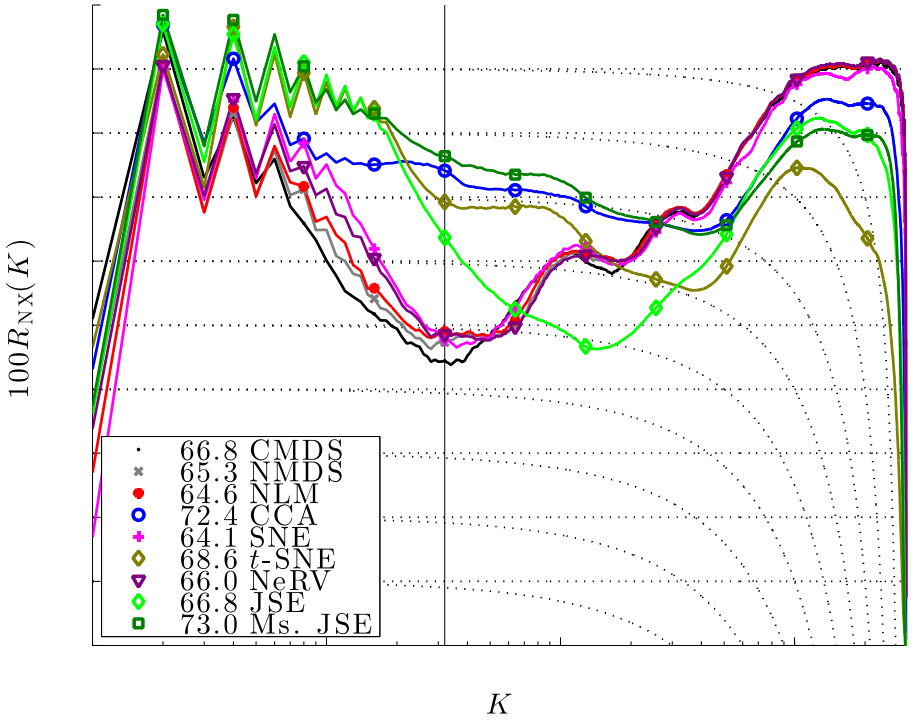


Results for Toroidal String

quality assessment

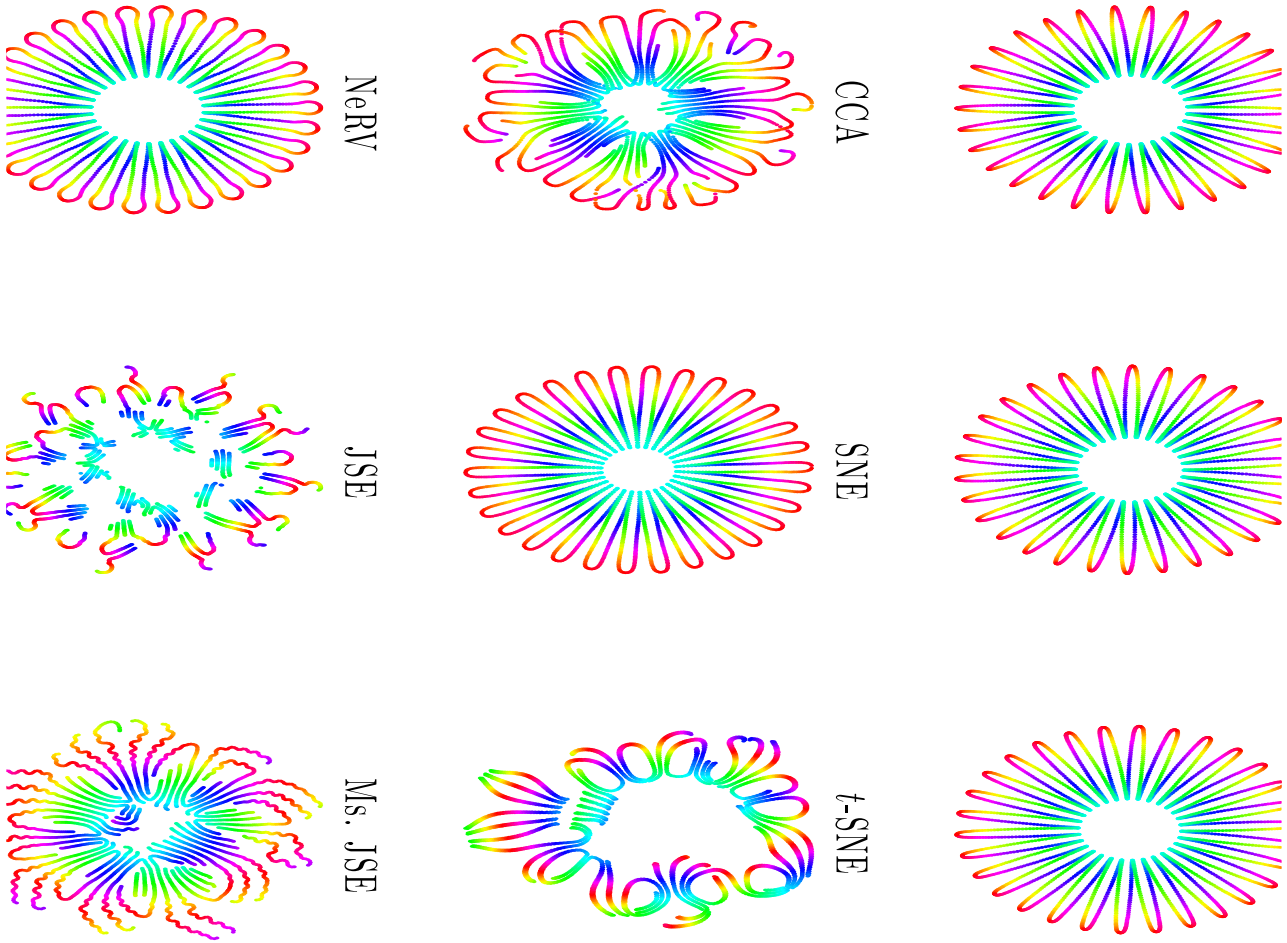


3 dimensions, $N = 3000$



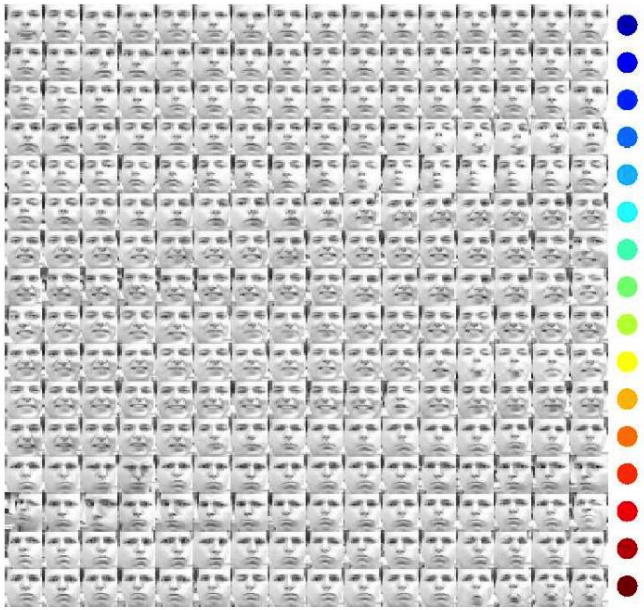
Results for Toroidal String

embeddings

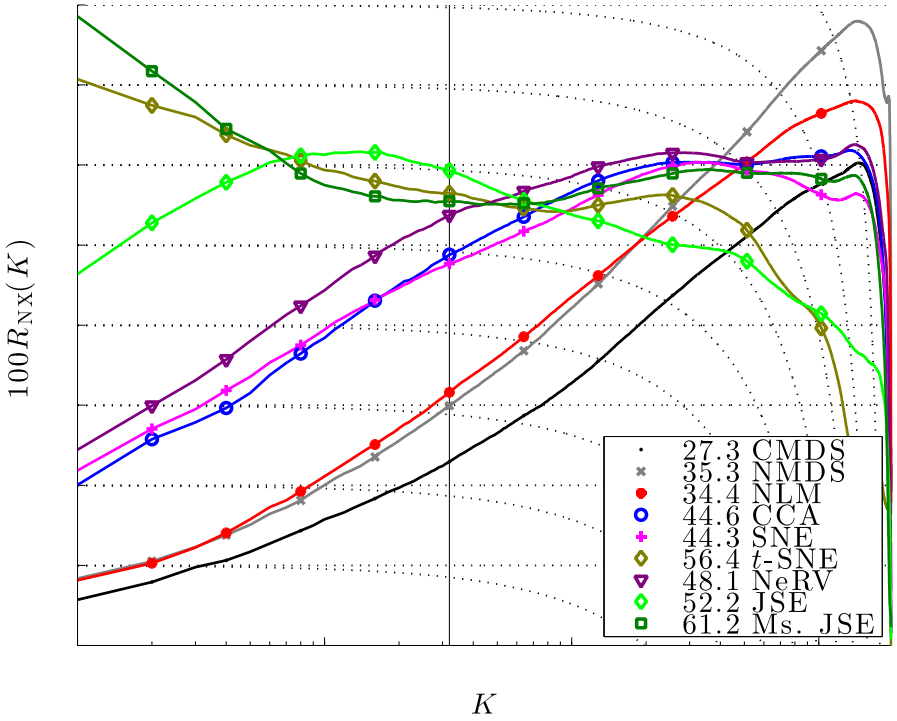


Results for B. Frey's faces

quality assessment

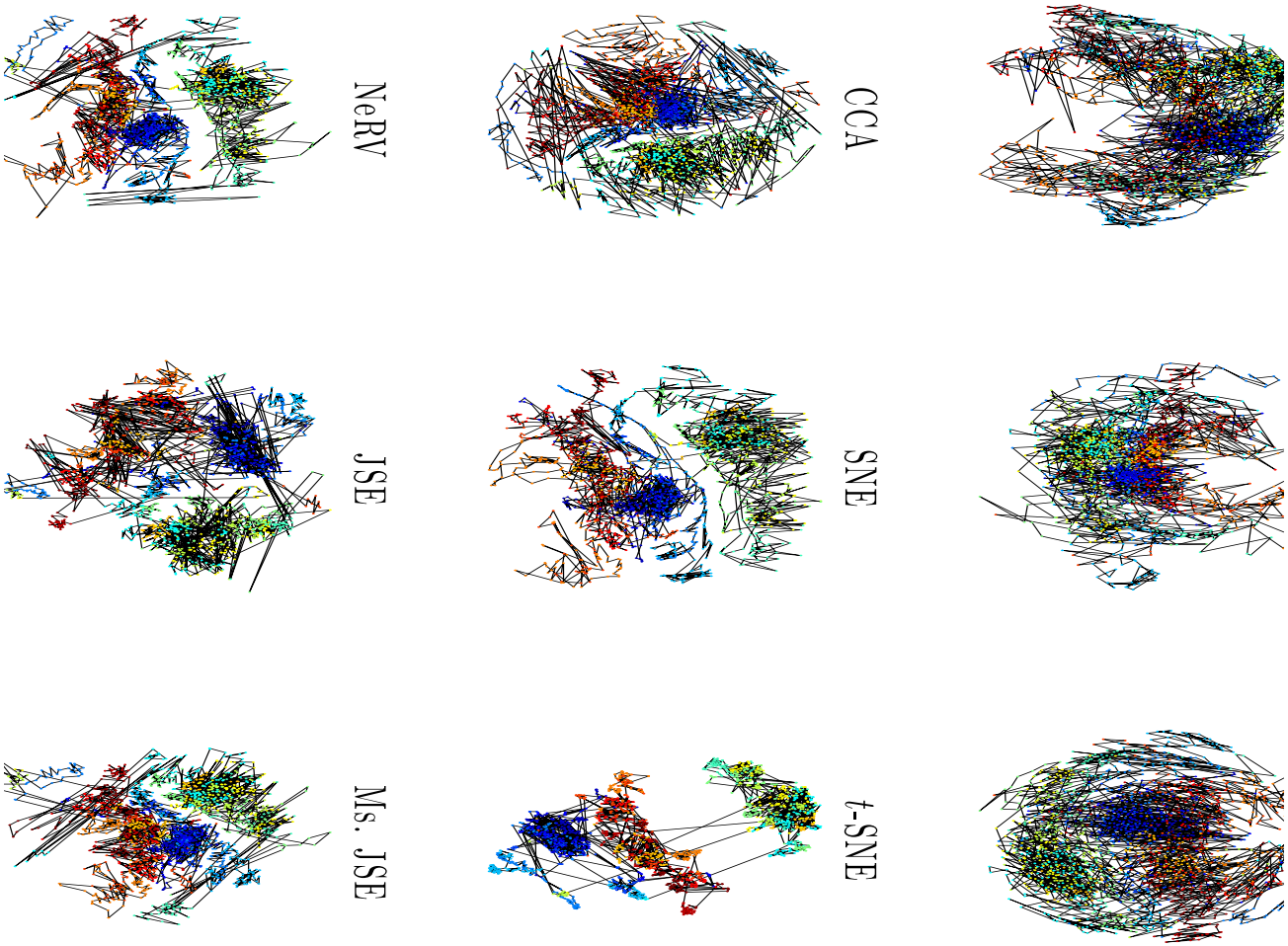


560 dimensions, $N = 1965$



Results for B. Frey's faces

embeddings



DR and Big Data…

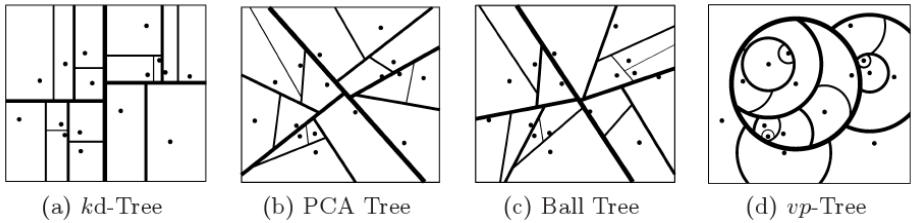
David vs. Goliath?

- Most DR/embedding techniques
  - Are nonparametric
  - Involve pairwise (dis)similaritities

  $\longrightarrow$ Space & time complexity $\geq O(N^2)$
  $\longrightarrow$ Poor scalability
- Workarounds
  - Use parametric DR methods (e.g. auto-encoders)
  - Aggregate/Approximate pairwise interactions
    - Favour locality/sparsity
    - Vantage point trees (HD) & quad trees (2D) $\longrightarrow O(N \log N)$
    - Fast multipole methods, fast Gauss transform $\longrightarrow O(N)$

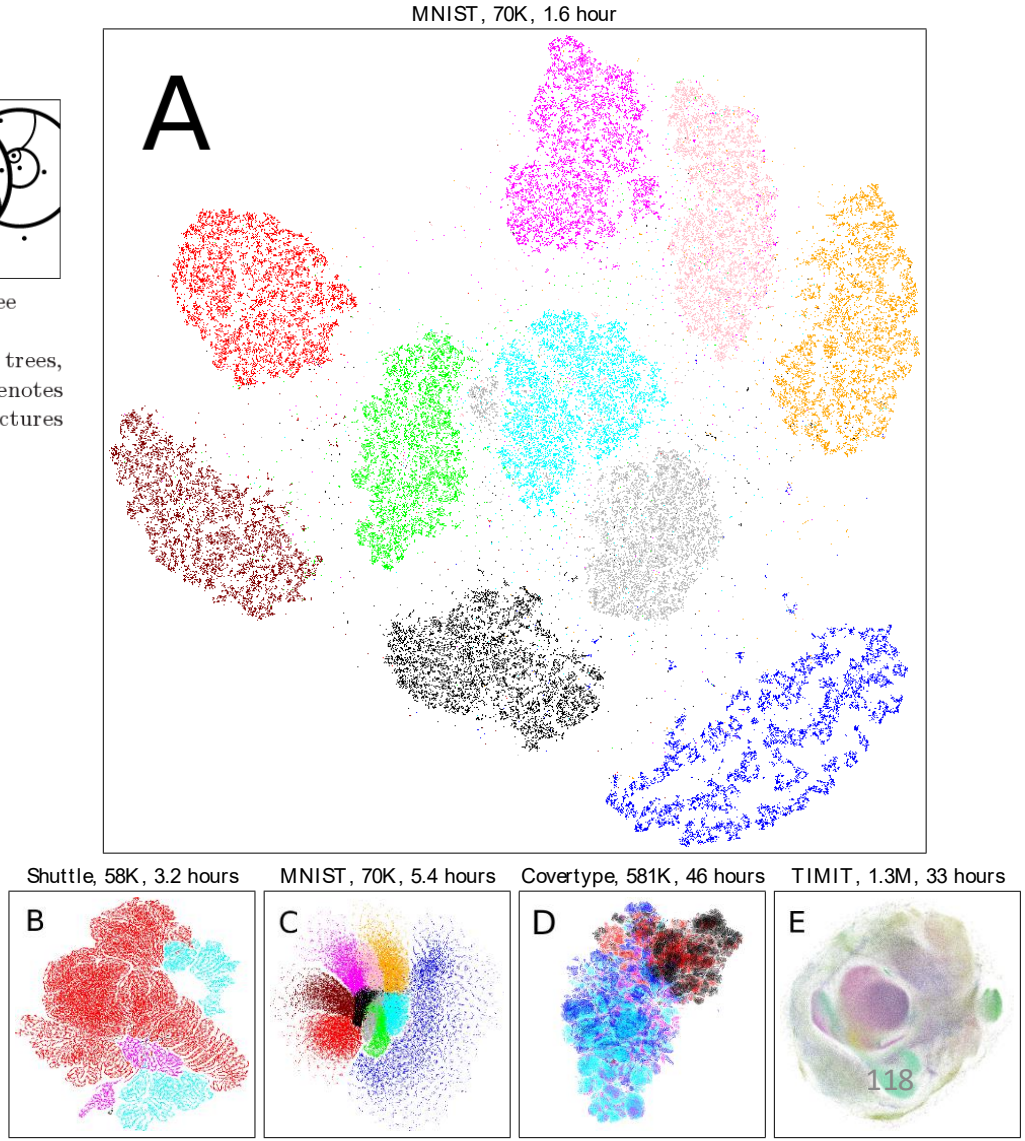
DR and Big Data…

Vantage point trees (HD) & quad trees (2D)



(a) kd-Tree (b) PCA Tree (c) Ball Tree (d) vp-Tree

Fig. 2: The partitioning of a 2D point set using different types of nearest neighbor trees, all with a maximum leaf size of 1 and a branching factor of 2. Line thickness denotes partition order (thicker lines were partitioned first). Note the very different structures created by the methods, which result in very different search speeds.



Original figure from Yang, 2013.

DR and Big Data…

Fast multipole methods, fast Gauss transform

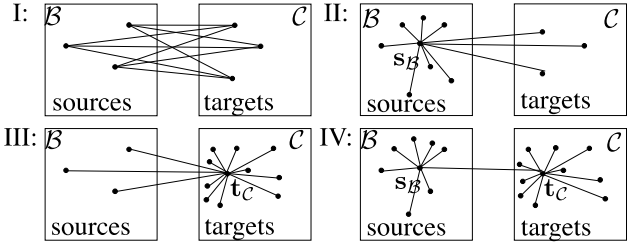
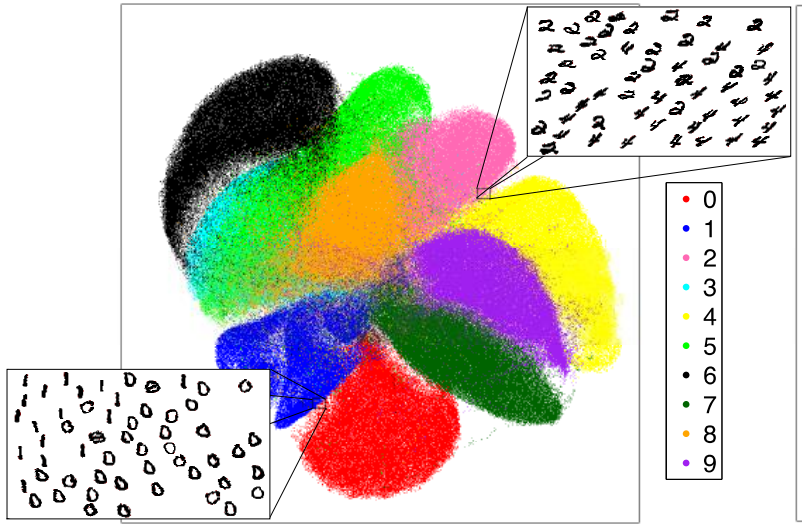
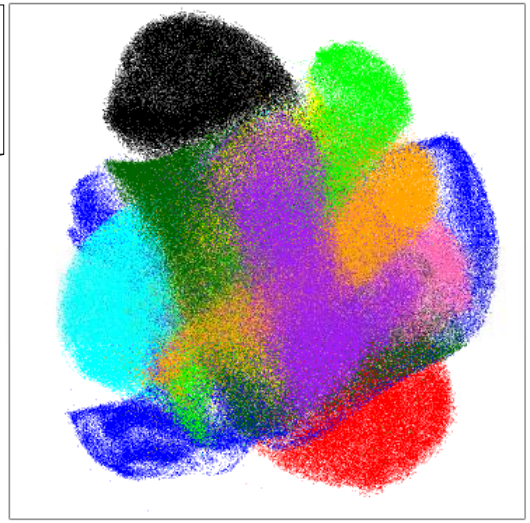


Figure 2: Different FGT approximations. I: exact interaction (4) (few points in both boxes); II: expansion around $\mathbf{s}_{\mathcal{B}}$ (many source points); III: expansion around $\mathbf{t}_{\mathcal{C}}$ (many target points); IV: expansion around $\mathbf{s}_{\mathcal{B}}$, then Taylor expansion to the Hermite functions (many points in both boxes).

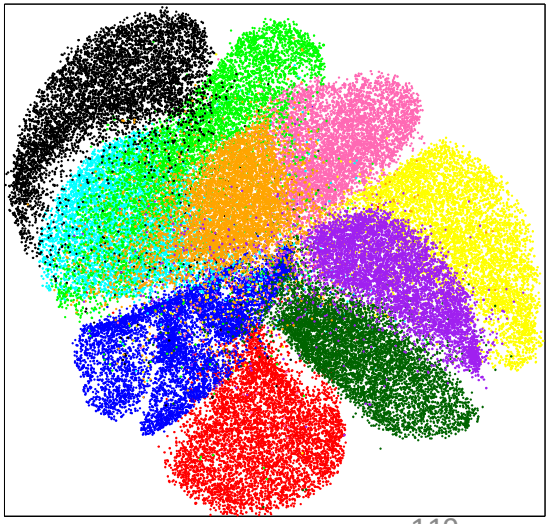
FGT using L-BFGS after 3 hours
$E = 521\,666, \quad 221$ iter.



BH using L-BFGS after 3 hours
$E = 1\,079\,357, \quad 32$ iter.



Out-of-sample using FGT, 11 min



Original figures from Vladymyrov, 2014.

DR and Big Data…

Latest developments and take-home messages…

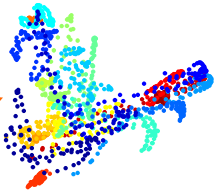
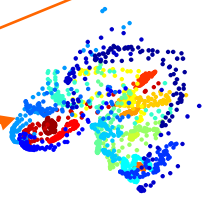
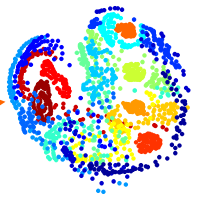
- Fast approximate versions of single-scale methods like (*t*-)SNE, JSE, NeRV, etc.
  —> space-partitioning trees, multipole methods
- UMAP (Uniform Manifold Approximation & Projection): similar to *t*-SNE
  - Same principle of shift invariance (simplified/hardcoded)
  - Pretends to be global but controversial (the global structure would heavily depend on initialization)
  - Faster than BH *t*-SNE, not faster than FI*t*-SNE (Fourier Interpolation t-SNE); linear-time iterations thanks to 'negative sampling' (gradient terms subsampled on all close neighbors but just a few random non-neighbours)
- [Fast] multiscale NE: Ms (*t*-)SNE, JSE, t-SNE, NeRV
  - Average similarities for exponentially-growing perplexities
  - Avoids the burden of manually choosing a perplexity
  - Local ***and*** global (covers all scales) at slightly higher complexity

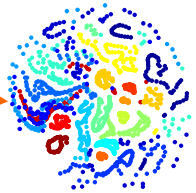
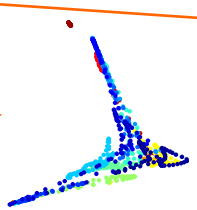
.

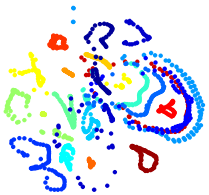
	Original	Fast/approximate
Single scale	$O(N^2)$	$O(N \log(N))$
Multiscale	$O(N^2 \log(N))$	$O(N \log^2(N))$

(NL)DR through time…

1901 Principal component analysis (PCA)
1938 Classical multidimensional scaling (CMDS)
1962 Nonmetric MDS (NMDS)
1969 Sammon's nonlinear mapping (NLM)
1982 Self-organising maps (SOMs)
1991 Auto-encoder (back prop.)
1993 Curvilinear component analysis (CCA)
1996 Kernel PCA
1998 Isomap
2000 Locally linear embedding (LLE)
2002 Laplacian eigenmaps (LE)
2002 Stochastic neighbour embedding (SNE)
2006 Auto-encoder (deep learning)
2008 Student-distributed SNE (*t*-SNE)
2010 Neighbour retrieval & vis. (NeRV)
2012 Jensen-Shannon Embedding (JSE)
2014 Multiscale JSE (Ms JSE)
2018 UMAP, *tt*-SNE, Ms *t*-SNE
2019 FI*t*-SNE, NE with missing data
2022 Fast Multiscale NE

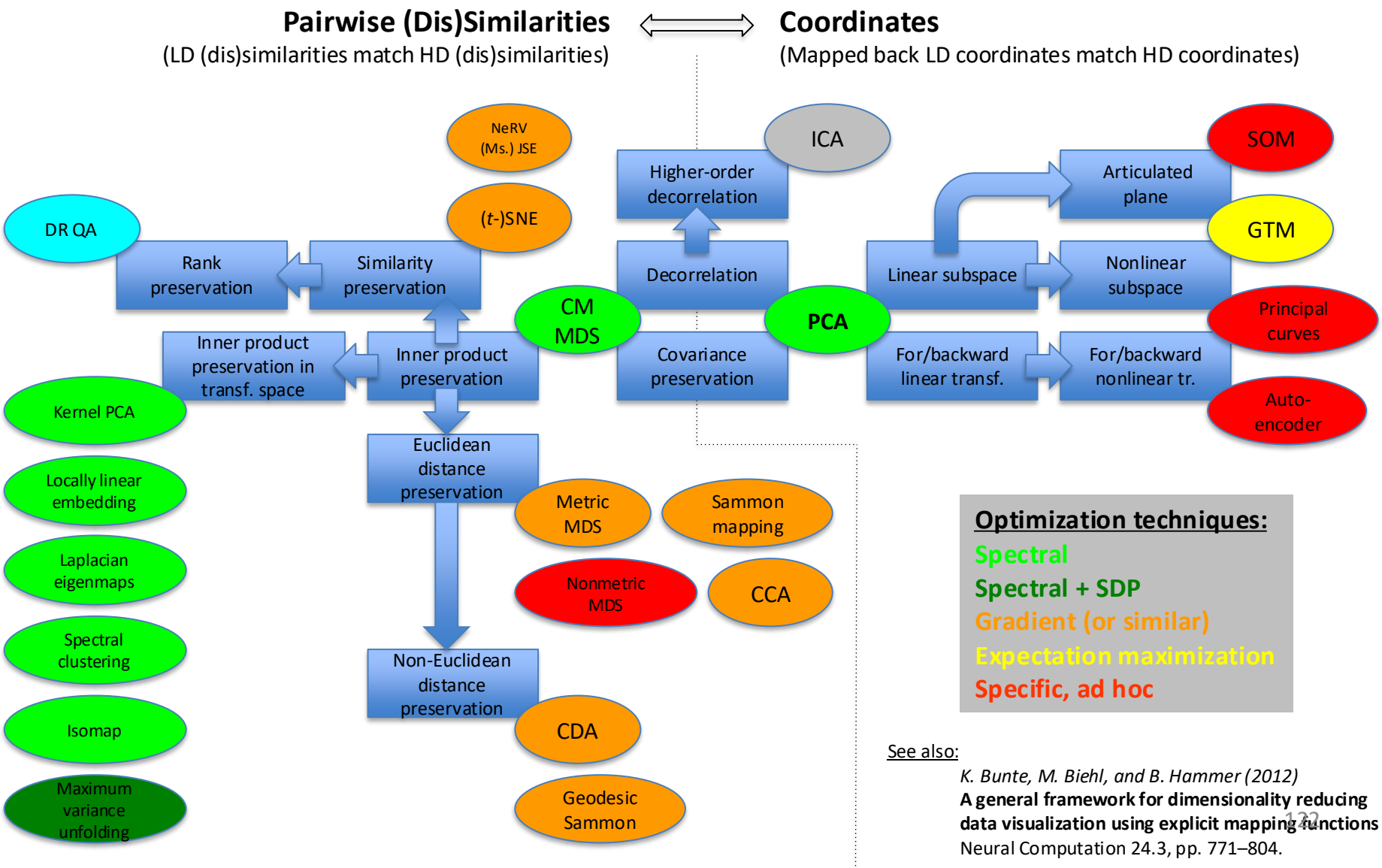




Spectral
Neural network
Reconstruction error
Distance preservation
Similarity preservation

Method taxonomy



See also:

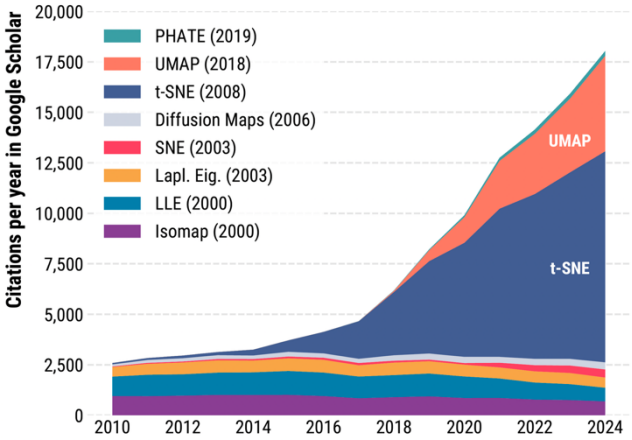
K. Bunte, M. Biehl, and B. Hammer (2012)
A general framework for dimensionality reducing data visualization using explicit mapping functions
Neural Computation 24.3, pp. 771–804.

Theoretical method comparisons

- Purpose
  - Visualization / data preprocessing
  - Hard / soft dimensionality reduction
- Model characteristics
  - Backward (HD to LD) / forward (LD to HD, generative)
  - Linear / non-linear
  - Parametric / non-parametric (new data embedding possible or not)
  - With / without vector quantization (SOMs & CCA vs all others, cfr scalability)
- Algorithmic criteria
  - Spectral / non-spectral (soft-computings, ANN, etc.)
  - Among spectral methods: dense / sparse matrix
- Several unifying paradigms or framework
  - Distance preservation (stress-based MDS)
  - Neighborhood preservation (SOMs & NE, neighbor embedding)
  - Force-directed placement (analogy with physical N-body problems)
  - Rank preservation (mainly for QA, quality assessment)

Final thoughts & perspectives

- In practice, NLDR requires
  - Appropriate data preprocessing steps and a suitable metric/(dis)similarity
  - An estimator of the intrinsic dimensionality
  - A NLDR method
  - Method-independent quality criteria
- Main take-home messages
  - Carefully adjust your model complexity…
  - Beware of (hidden) metaparameters…
  - Convex methods are not a panacea…
  - But always try PCA first…
- Present & Future …
  - Local NLDR with neighbor embedding (NE) is booming in applicative fields (e.g., computational biology)
  - Reconciliating local and global DR is needed…
  - The interest for spectral methods seems to diminish…
  - Will auto-encoders emerge again thanks to ‘deep’ learning?
  - (Scalable) similarity-based NLDR is a hot and debated topic…
  - Tighter connections are expected with the domains of data mining, visualization, and graph embedding

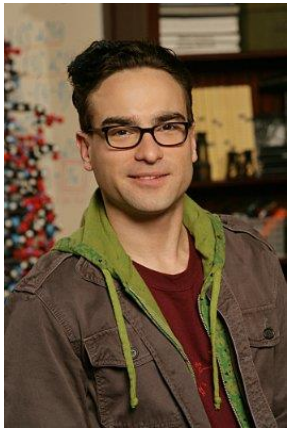


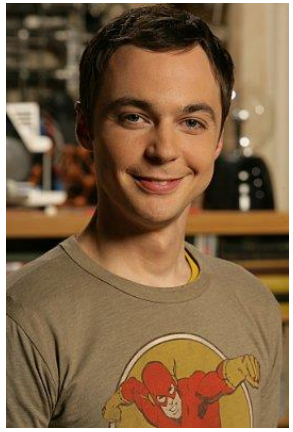
https://arxiv.org/pdf/2508.15929

Final thoughts & perspectives

Some of the world's best scientists talk about high-dimensional spaces on TV…

- *Leonard*: [*discussing Sheldon's work*] At least I didn't have to invent 26 dimensions just to make the math come out.
- *Sheldon*: I didn't invent them. They're there.
- *Leonard*: In what universe?
- *Sheldon*: In all of them, that is the point!



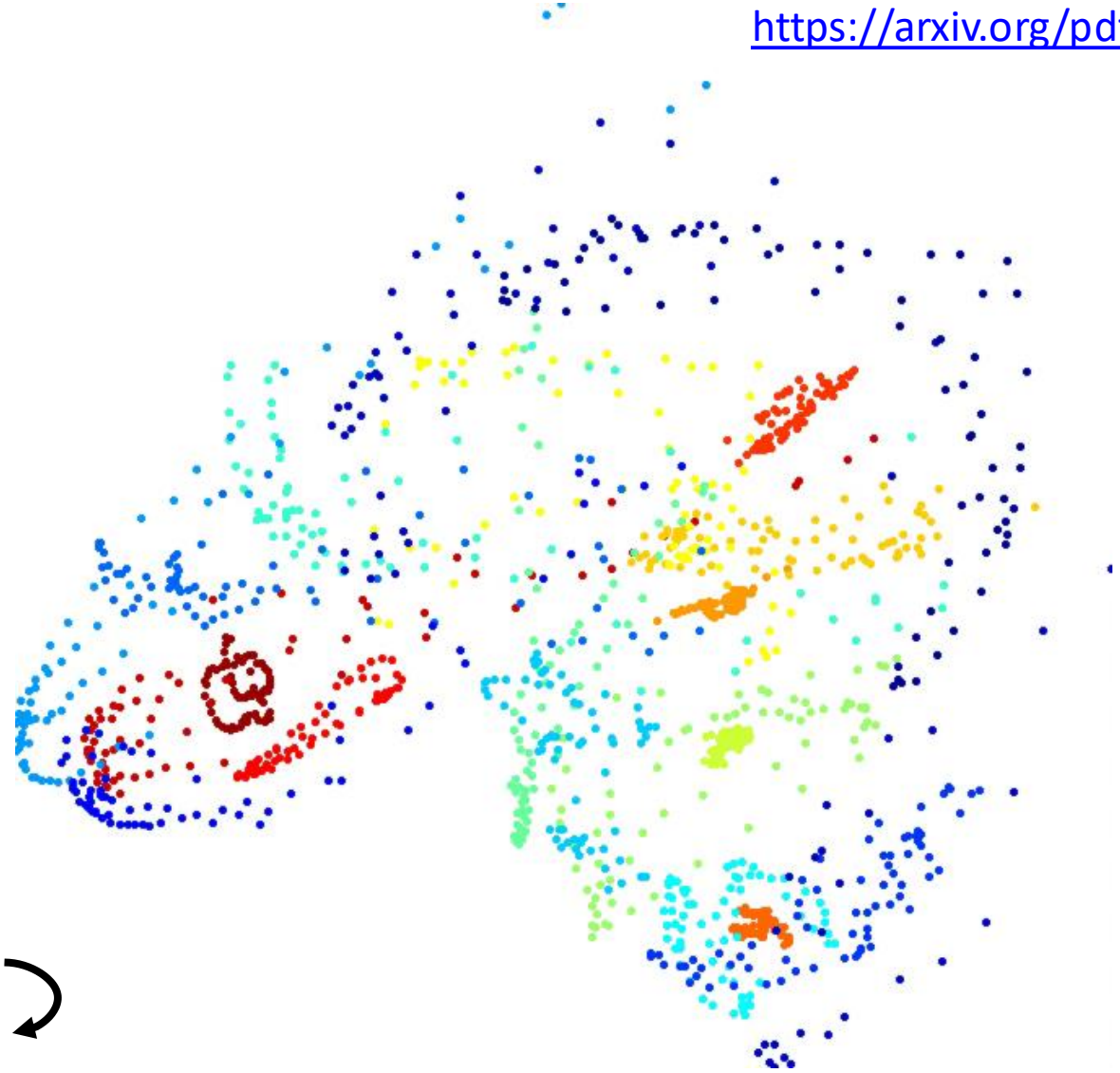


(The Big Bang Theory, pilot episode, 2007)

Thank you for your attention

Any question? Here or later… john.lee@uclouvain.be

https://arxiv.org/pdf/2508.15929



Nonlinear Dimensionality Reduction
J.A. Lee, M. Verleysen, Springer 2007
300 pp. ISBN: 978-0-387-39350-6

NMDS

Ms. JSE