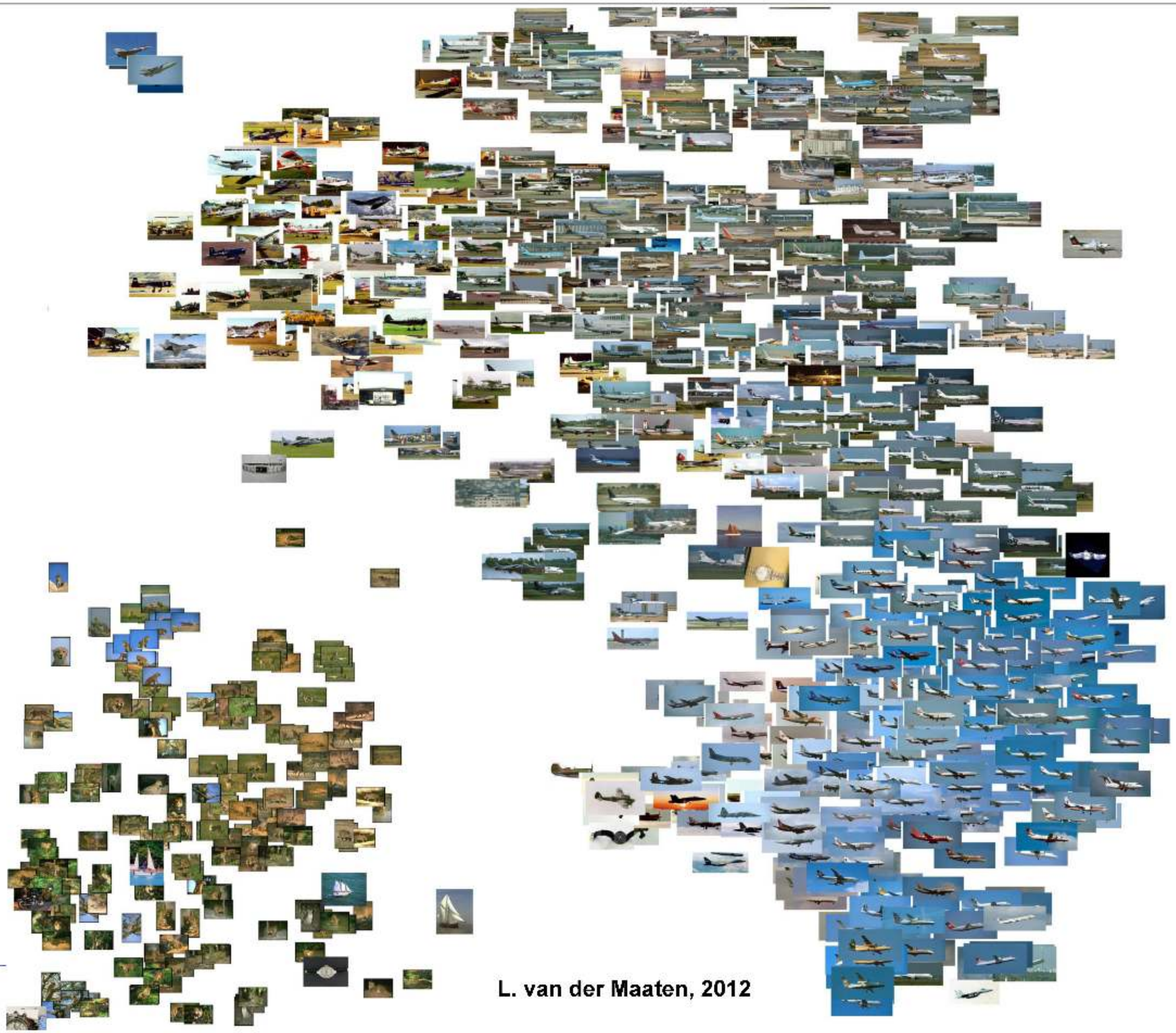
Unsupervised dimensionality reduction: from principal component analysis to modern nonlinear techniques

John A. Lee
Michel Verleysen

Machine Learning Group,
Université catholique de Louvain
Louvain-la-Neuve, Belgium

Image banks



L. van der Maaten, 2012

Image banks (zoom)

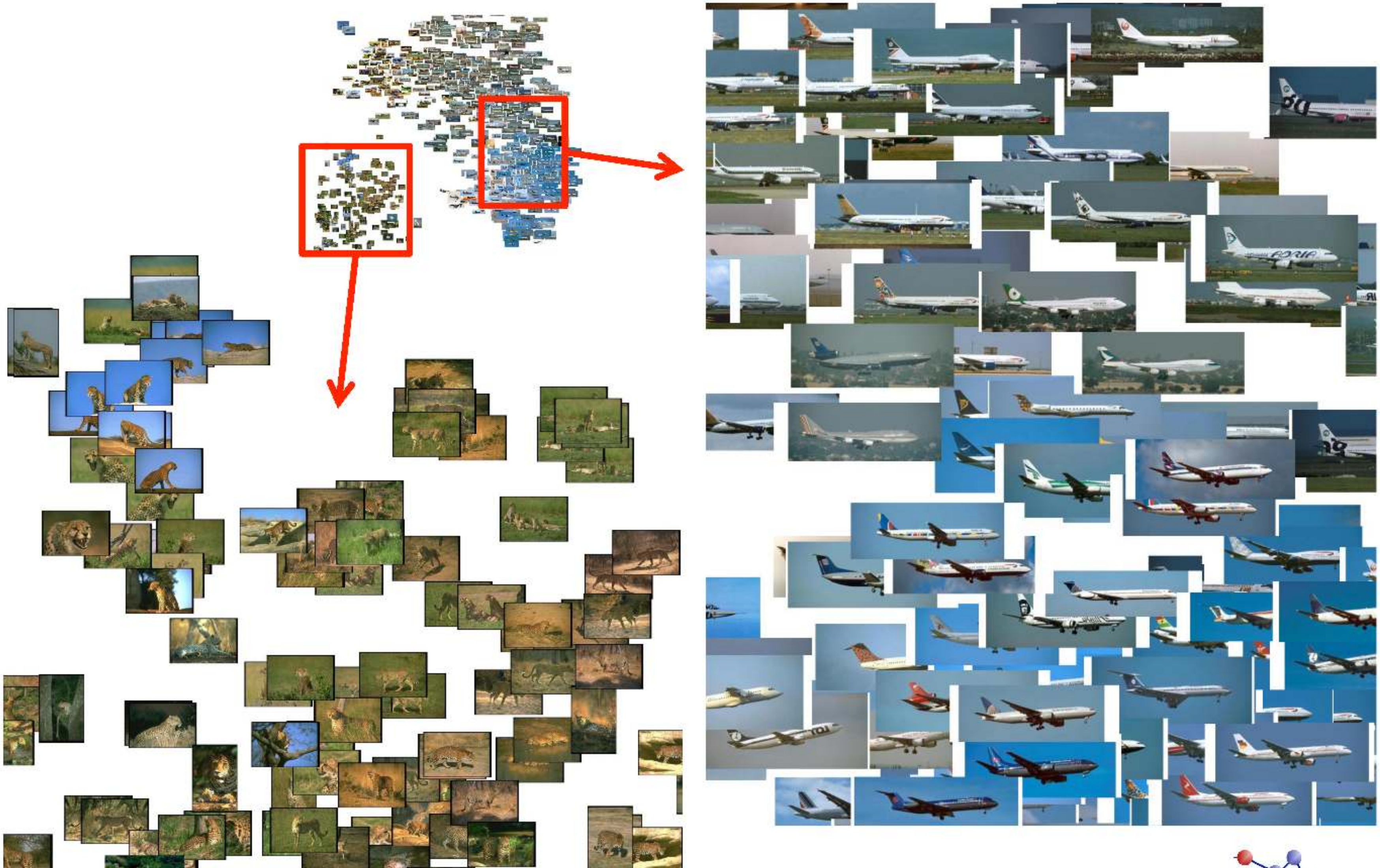
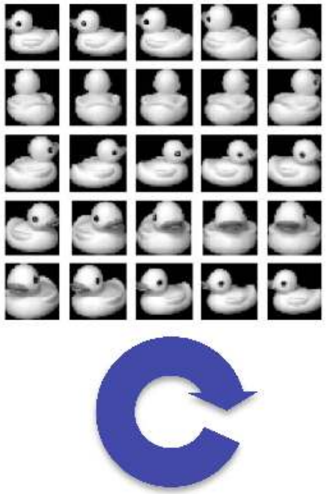


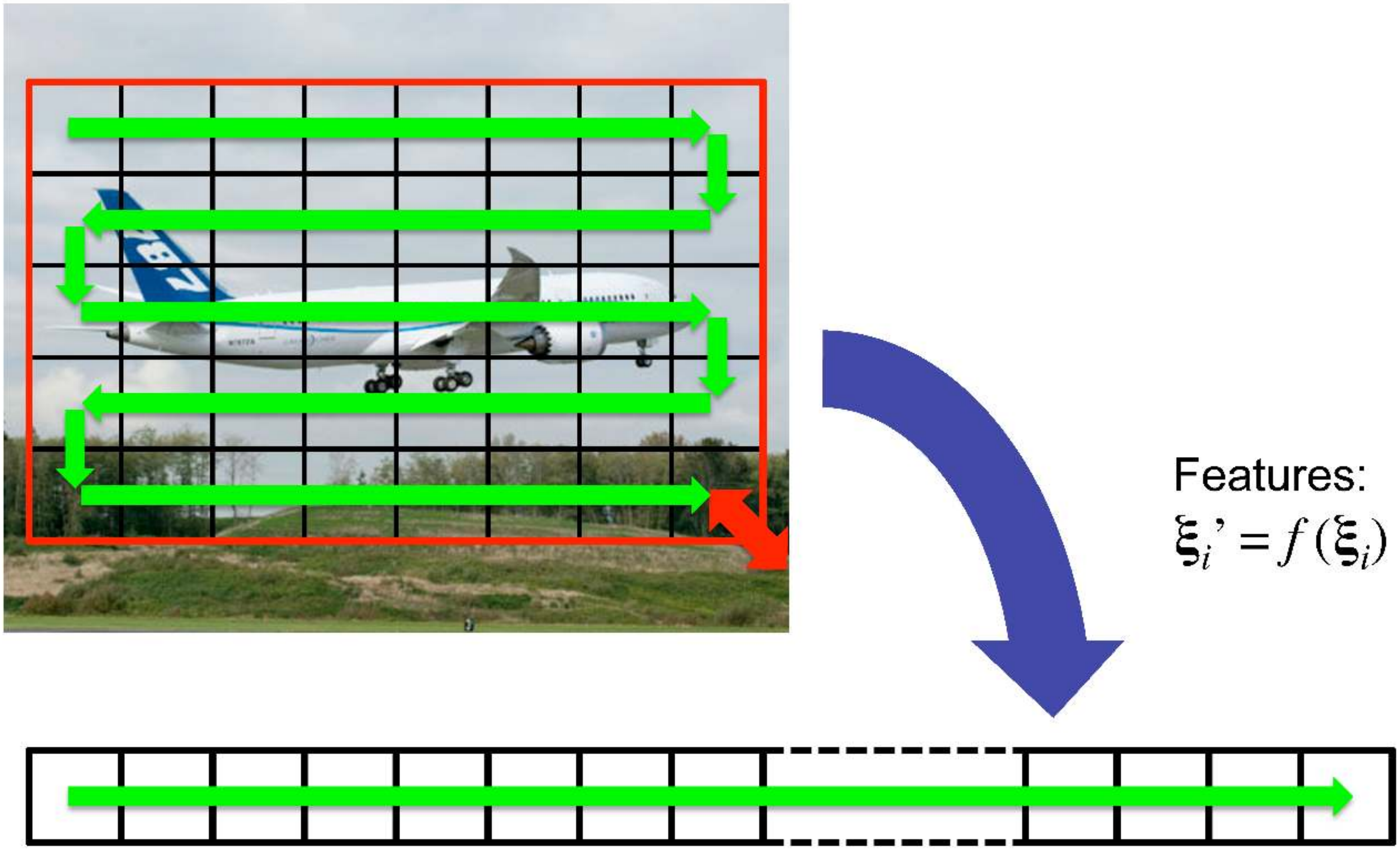
Image banks



COIL data set
(pictures of rotated objects)

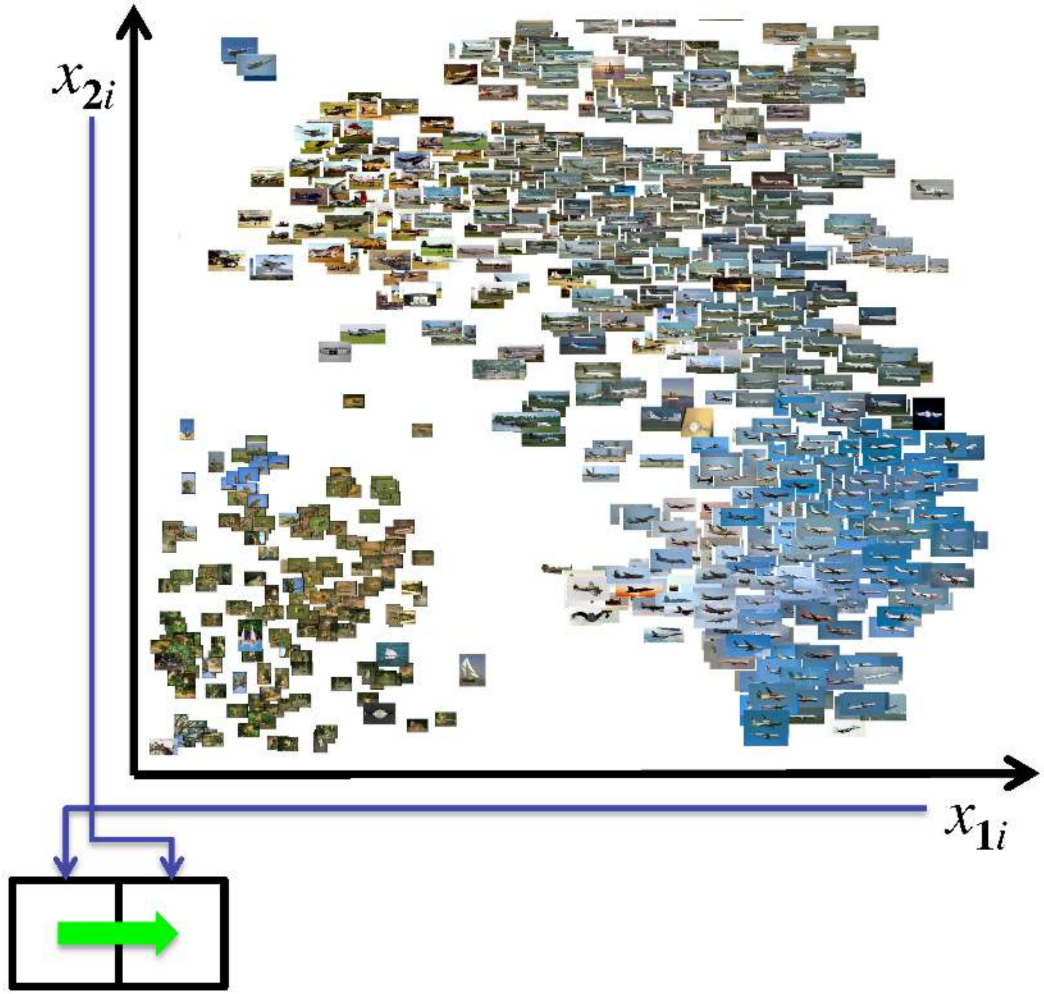
MNIST data set
(scanned handwritten digits)

How to encode images?



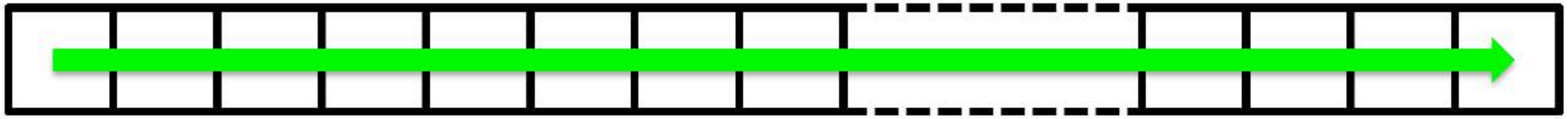
M-dimensional vectors: $\Xi = [\xi_i]_{1 \leq i \leq N}$

How to encode the representation?

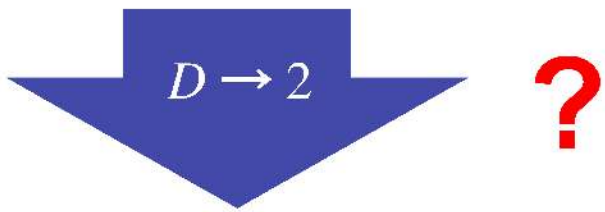


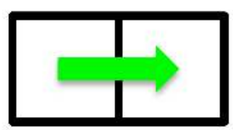
2-dimensional vectors: $\mathbf{X} = [\mathbf{x}_i]_{1 \leq i \leq N}$

From the image to the representation…



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

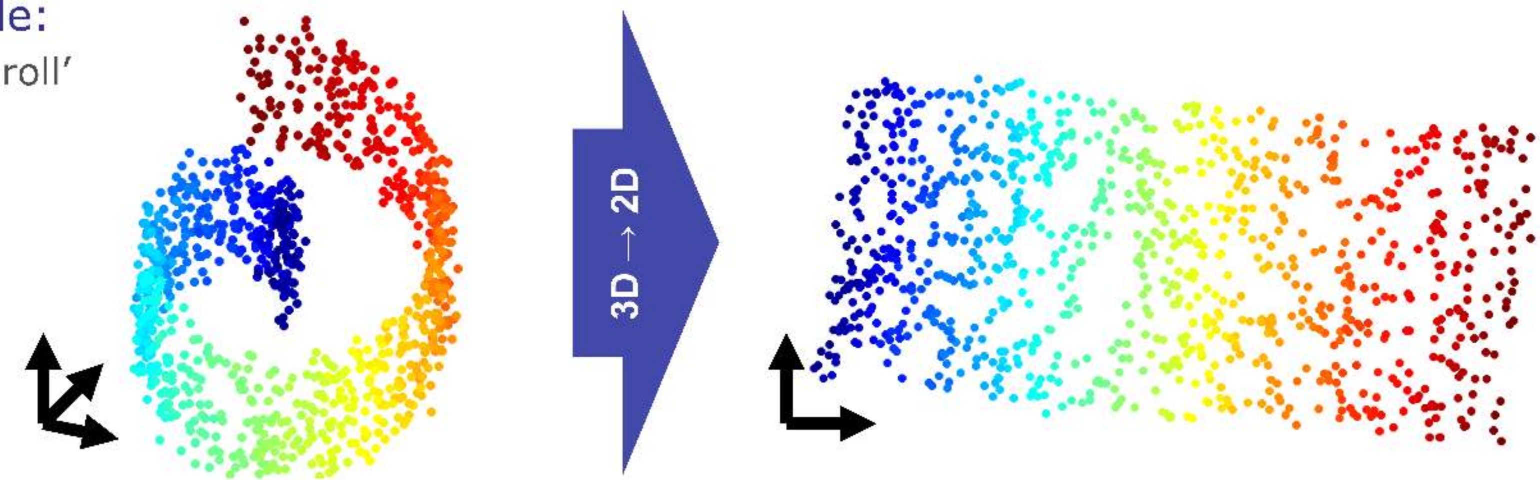




2-dimensional vectors: $\mathbf{X} = [\mathbf{x}_i]_{1\leq i\leq N}$

Dimensionality reduction

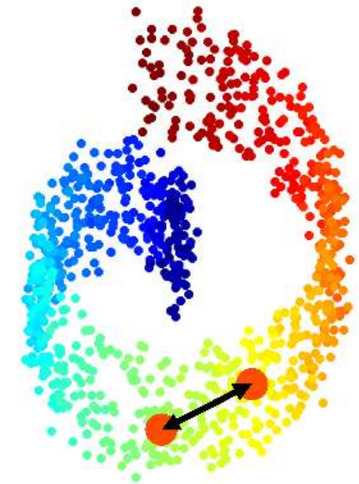
- (NL)DR is a.k.a.
  Manifold learning, embedding, scaling, (nonlinear) projection, feature extraction, etc.
- Purpose:
  Faithful low-dimensional representation of high-dimensional data
- Typical paradigms/models:
  - Autoassociation with bottleneck
  - Preservation of 'spatial' properties (dot products, distances, similarities, etc.)
  - Linear/nonlinear
  - Generative/discriminative
- Example:
  'Swiss roll'

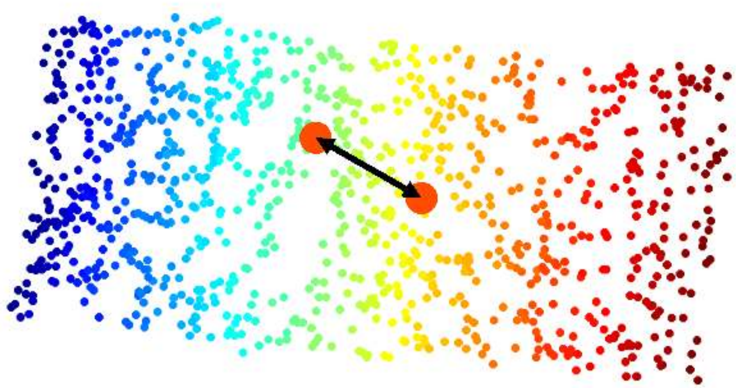


$\Xi = [\boldsymbol{\xi}_i]_{1\leq i\leq N}$ $\mathbf{X} = [\mathbf{x}_i]_{1\leq i\leq N}$

Faithful representation?

For all pairs
of points…





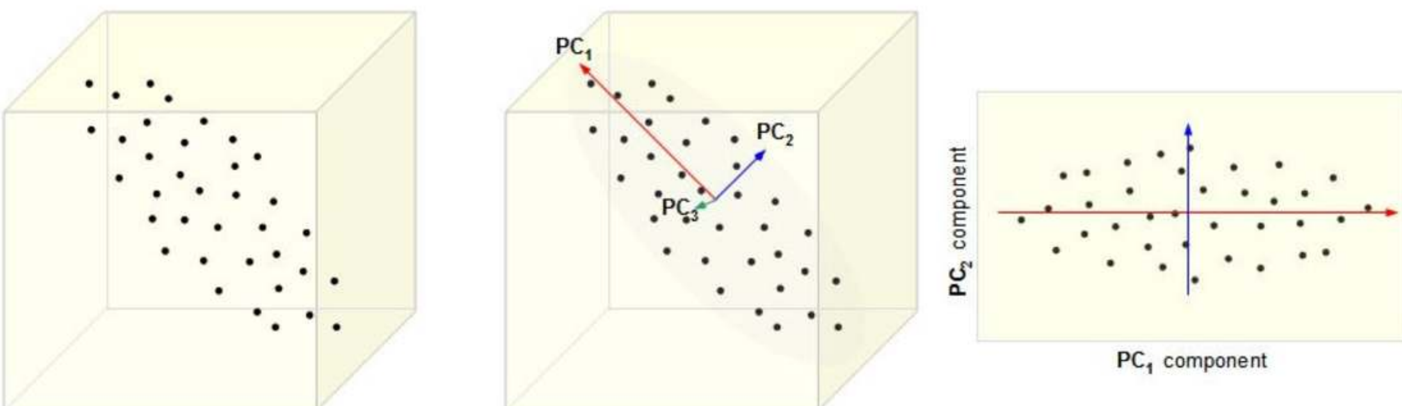
		HD	
		Near	Far
LD	Near	☺	☹
	Far	😐	☺

DR through the ages…

- **Principal component analysis (PCA)** 1901
- **Classical metric multidimensional scaling (CM MDS)** 1938
- Stress-based MDS 1952
- Nonmetric MDS 1962
- Sammon mapping 1969
- Self-organizing map 1982
- Principal curves 1984
- Auto-encoder (bottleneck FFN) 1991
- Curvilinear component analysis 1993
- Spectral methods (space transf. + CM MDS)
  - Kernel PCA 1996
  - Isomap 1998
  - Locally linear embedding 2000
  - Laplacian eigenmaps 2002
  - Maximum variance unfolding 2004
- Deep auto-encoder 2006
- Similarity-based embedding
  - (*t*-distributed) stochastic neighbor embedding 2008
  - Neighbor retrieval and visualization (NeRV) 2010

Linear dimensionality reduction

- Principal component analysis (PCA)
  - Minimal error after forward/backward linear transformation
  - Covariance preservation
  - Best-fit linear subspace
  - (2^{nd} order) decorrelation
- Classical metric multidimensional scaling (CMMDS)
  - Inner product preservation
- PCA and CMMDS are dual
  - PCA: eigenvalue decomposition of covariance matrix ($\Xi\,\Xi^{\mathrm{T}}/N$)
  - CMMDS: eigenvalue decomposition of Gram matrix ($\Xi^{\mathrm{T}}\,\Xi$)



DR through the ages…

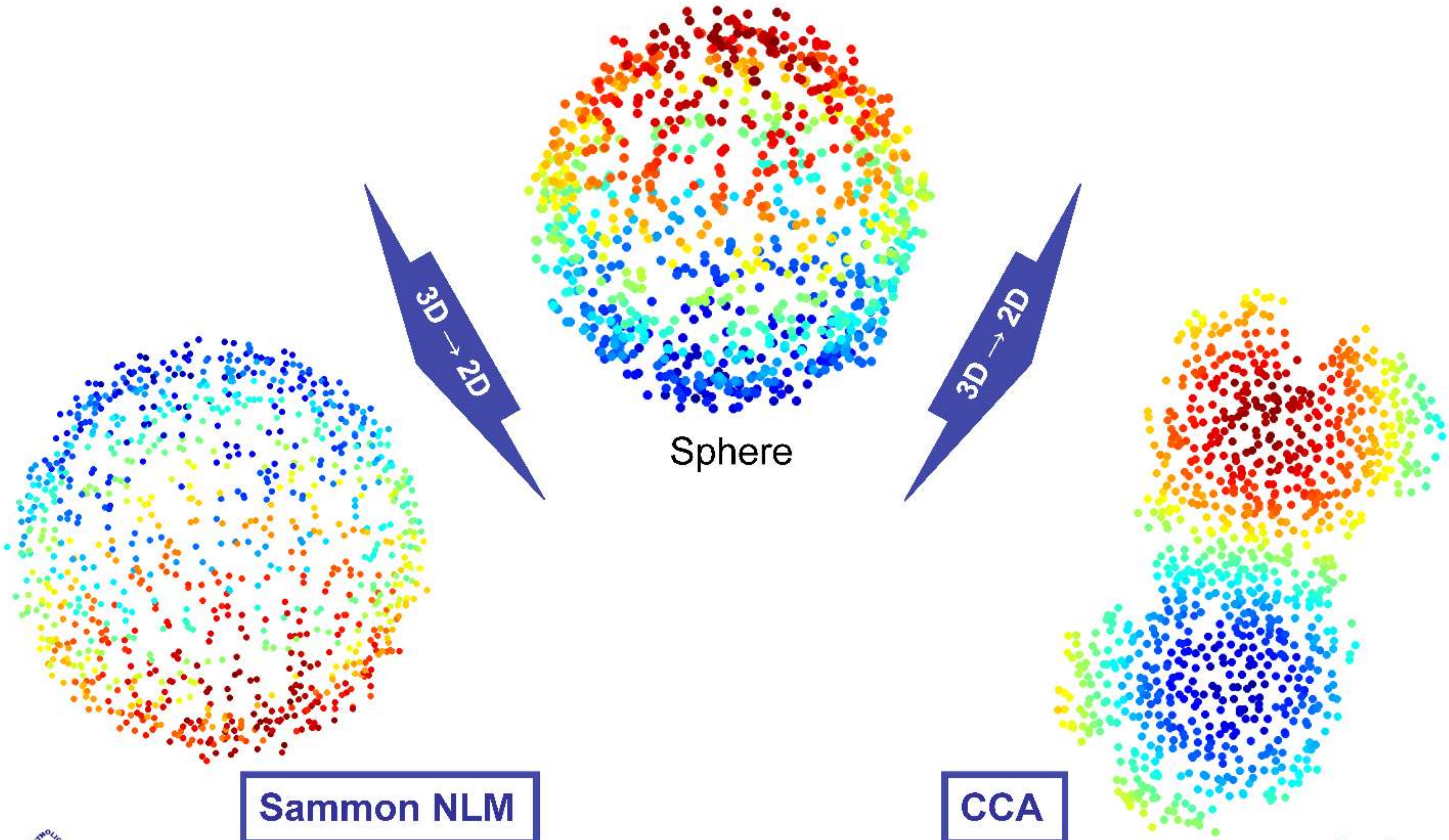
- Principal component analysis (PCA) — 1901
- Classical metric multidimensional scaling (CM MDS) — 1938
- **Stress-based MDS** — 1952
- **Nonmetric MDS** — 1962
- **Sammon mapping** — 1969
- Self-organizing map — 1982
- Principal curves — 1984
- Auto-encoder (bottleneck FFN) — 1991
- **Curvilinear component analysis** — 1993
- Spectral methods (space transf. + CM MDS)
  - Kernel PCA — 1996
  - Isomap — 1998
  - Locally linear embedding — 2000
  - Laplacian eigenmaps — 2002
  - Maximum variance unfolding — 2004
- Deep auto-encoder — 2006
- Similarity-based embedding
  - (*t*-distributed) stochastic neighbor embedding — 2008
  - Neighbor retrieval and visualization (NeRV) — 2010

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)

CCA can tear manifolds



DR through the ages…

- Principal component analysis (PCA) 1901
- Classical metric multidimensional scaling (CM MDS) 1938
- Stress-based MDS 1952
- Nonmetric MDS 1962
- Sammon mapping 1969
- **Self-organizing map** 1982
- Principal curves 1984
- **Auto-encoder (bottleneck FFN)** 1991
- Curvilinear component analysis 1993
- Spectral methods (space transf. + CM MDS)
  - Kernel PCA 1996
  - Isomap 1998
  - Locally linear embedding 2000
  - Laplacian eigenmaps 2002
  - Maximum variance unfolding 2004
- **Deep auto-encoder** 2006
- Similarity-based embedding
  - (*t*-distributed) stochastic neighbor embedding 2008
  - Neighbor retrieval and visualization (NeRV) 2010

Self-organizing map

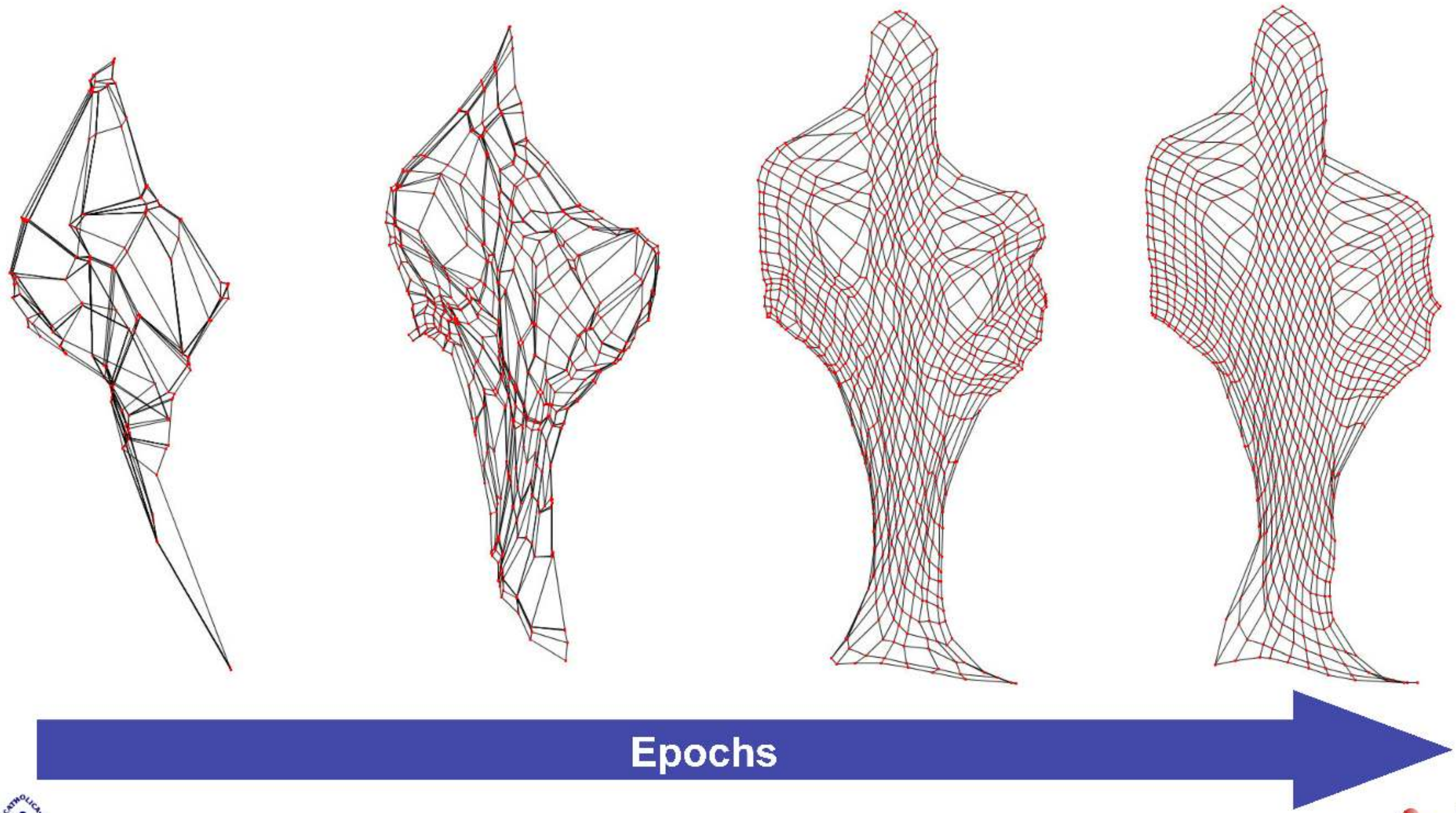
- von der Malsburg, 1973; Kohonen, 1982.
- Idea
  - Biological inspiration (brain cortex)
  - Nonlinear version of PCA
    - Replace PCA plane with an articulated grid
    - Fit the grid through the data cloud

  (≈ K-means with a priori topology and 'winner takes most' rule)
- Details
  - A grid is defined in the low-dim space: $\mathbf{G} = [\mathbf{g}_i]_{1 \leq i \leq N}$ and $d(\mathbf{g}_i, \mathbf{g}_j)$
  - Grid nodes have high-dim coordinates as well: $\mathbf{\Gamma} = [\boldsymbol{\gamma}_i]_{1 \leq i \leq N}$
  - 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 $\mathbb{R}^+$

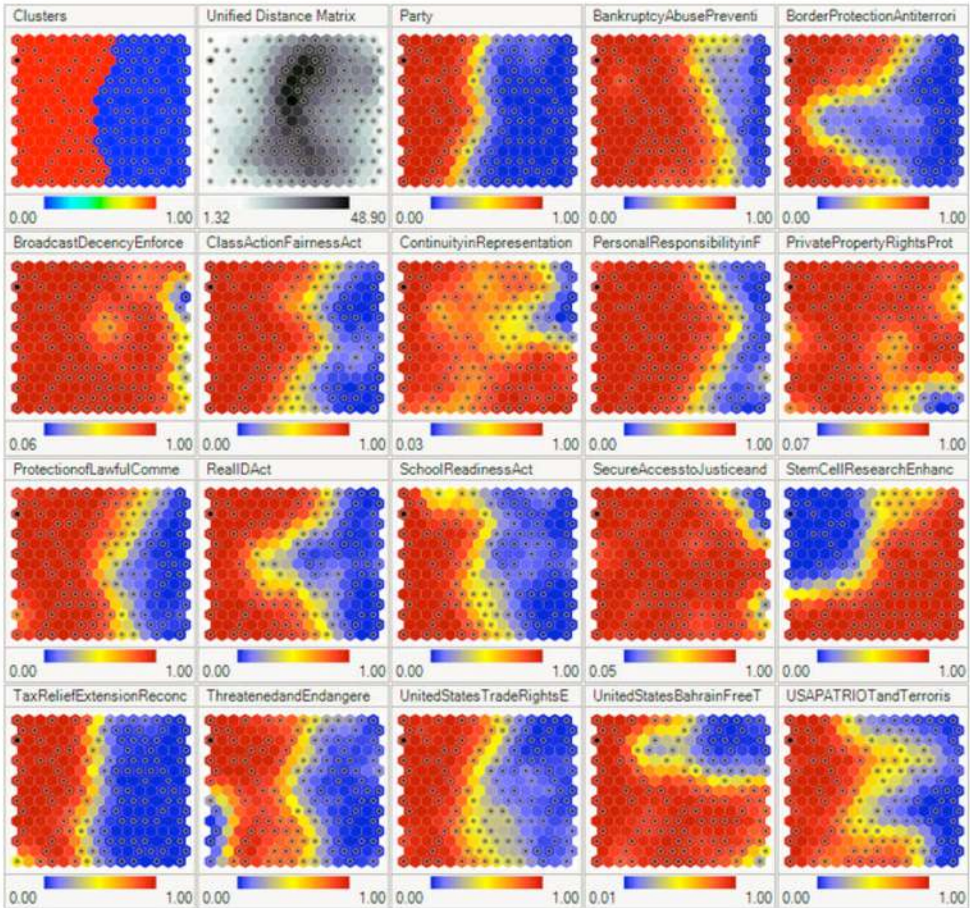
Self-organizing maps

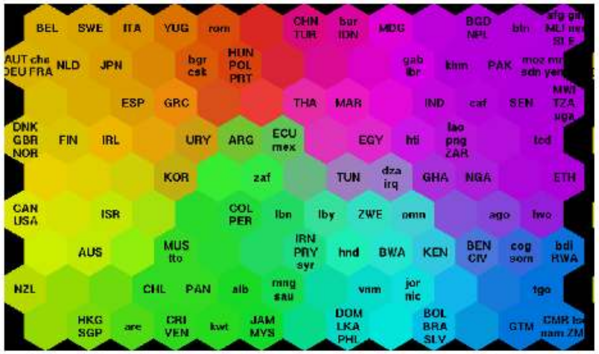
Articulated grid attracted by the data cloud (cactus-shaped here)

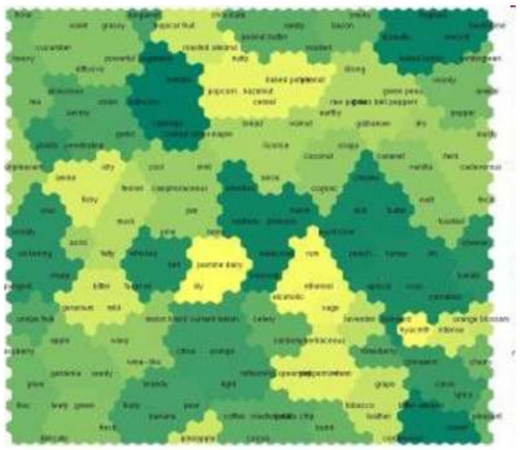


Self-organizing map

Visualisations in the grid space







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)

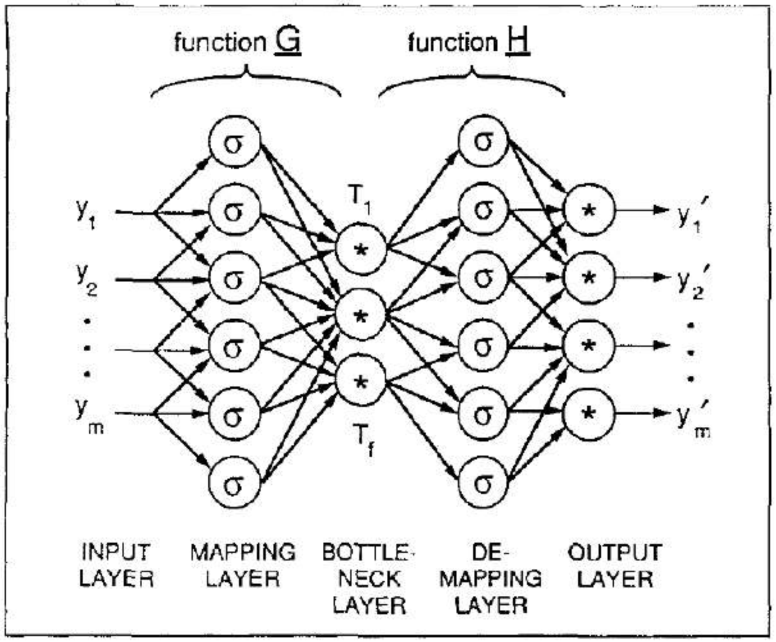


Figure 2. Network architecture for simultaneous determination of *f* nonlinear factors using an autoassociative network.

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

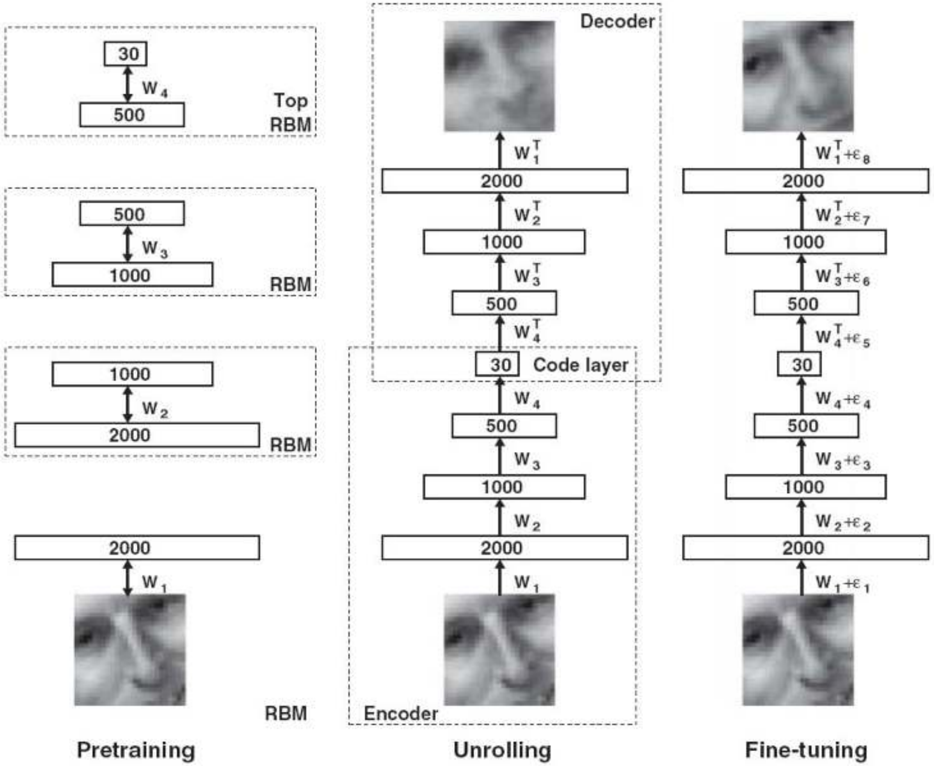


Fig. 1. Pretraining consists of learning a stack of restricted Boltzmann machines (RBMs), each having only one layer of feature detectors. The learned feature activations of one RBM are used as the "data" for training the next RBM in the stack. After the pretraining, the RBMs are "unrolled" to create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

Original figure from Kramer, 1991.

Original figure from Salakhutdinov, 2006.

DR through the ages…

- Principal component analysis (PCA) 1901
- **Classical metric multidimensional scaling (CM MDS)** 1938
- **Stress-based MDS** 1952
- **Nonmetric MDS** 1962
- **Sammon mapping** 1969
- Self-organizing map 1982
- Principal curves 1984
- Auto-encoder (bottleneck FFN) 1991
- **Curvilinear component analysis** 1993
- Spectral methods (space transf. + **CM MDS**)
  - Kernel PCA 1996
  - Isomap 1998
  - Locally linear embedding 2000
  - Laplacian eigenmaps 2002
  - Maximum variance unfolding 2004
- Deep auto-encoder 2006
- **Similarity-based embedding**
  - **(*t*-distributed) stochastic neighbor embedding** 2008
  - **Neighbor retrieval and visualization (NeRV)** 2010

Similarity preservation

- Examples
  - Stochastic neighbor embedding (SNE, 2002)
  - t-distributed SNE (t-SNE, 2008)
- Ingredients (replace distances with decreasing fun. of dist.)
  - Softmax similarities:

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

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

  - Similarity preservation:

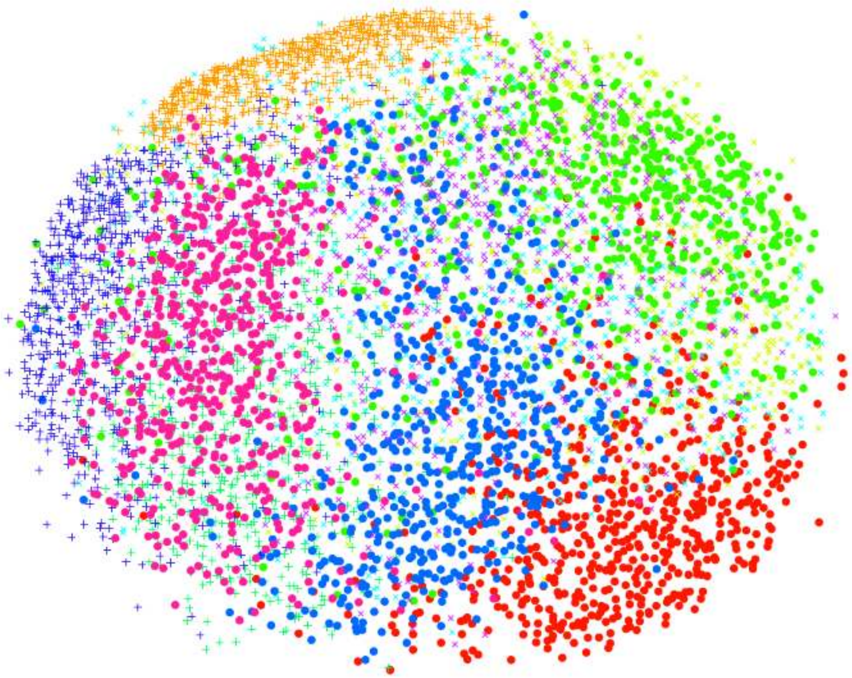
$$E(\mathbf{X};\boldsymbol{\Xi},\boldsymbol{\Lambda}) = \sum_{i=1}^{N} D_{\mathrm{KL}}(\boldsymbol{\sigma}_i\|\mathbf{s}_i)$$

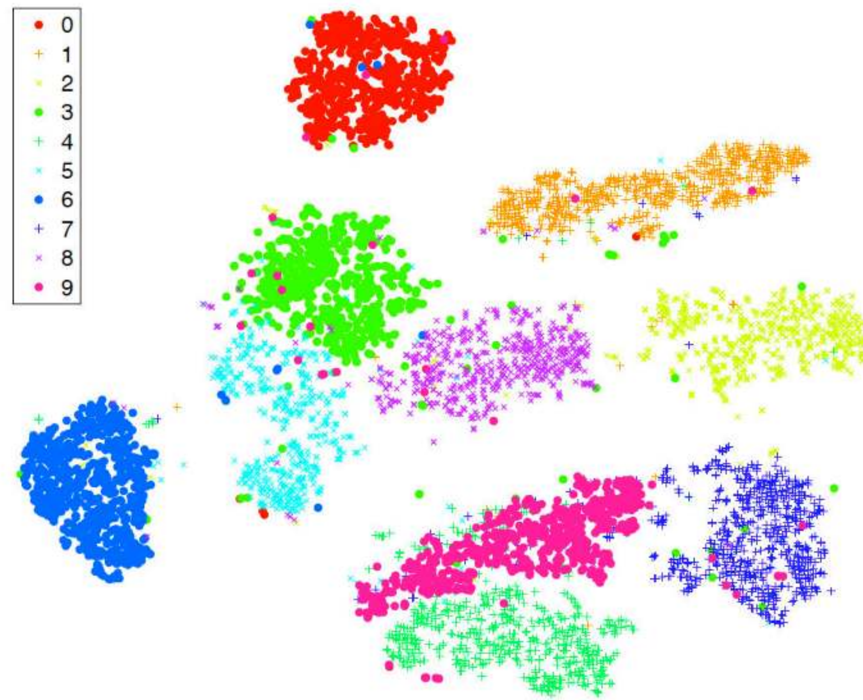
$$D_{\mathrm{KL}}(\boldsymbol{\sigma}_i\|\mathbf{s}_i) = \sum_{j=1}^{N}\sigma_{ij}\log(\sigma_{ij}/s_{ij})$$

Distance vs similarity preservation

Sammon's nonlinear mapping
(distance preservation)

t-SNE
(similarity preservation)

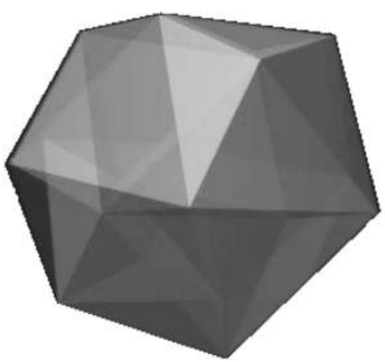
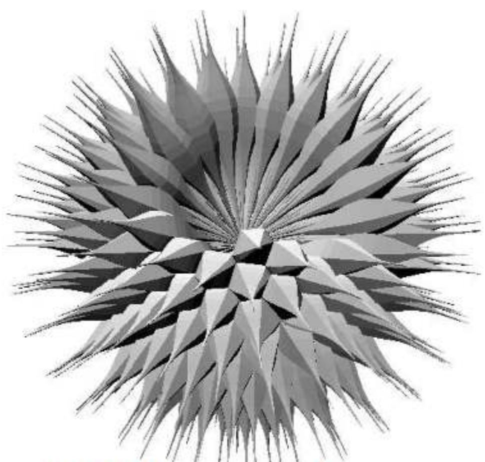




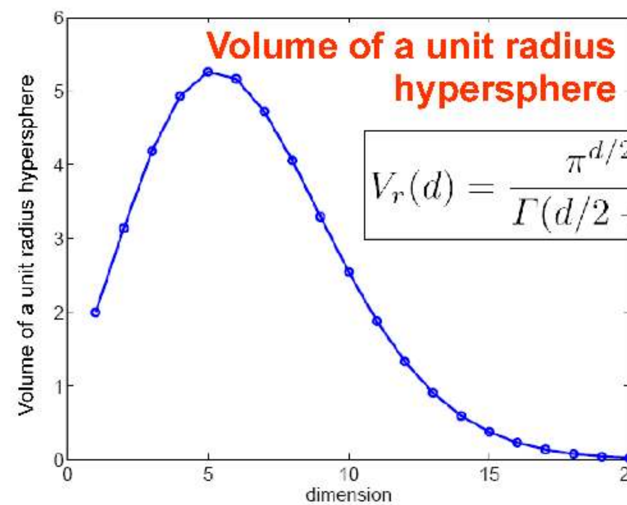
MNIST database of handwritten digits, pictures from Van der Maaten 2008

Having many dimensions: is it a blessing?

- The curse of dimensionality consists of
  - The *empty space* phenomenon
    (function approximation requires an exponential number of points)
  - The *norm concentration* phenomenon
    (Euclidean norms in a normal distribution have a chi distribution)
- It has 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

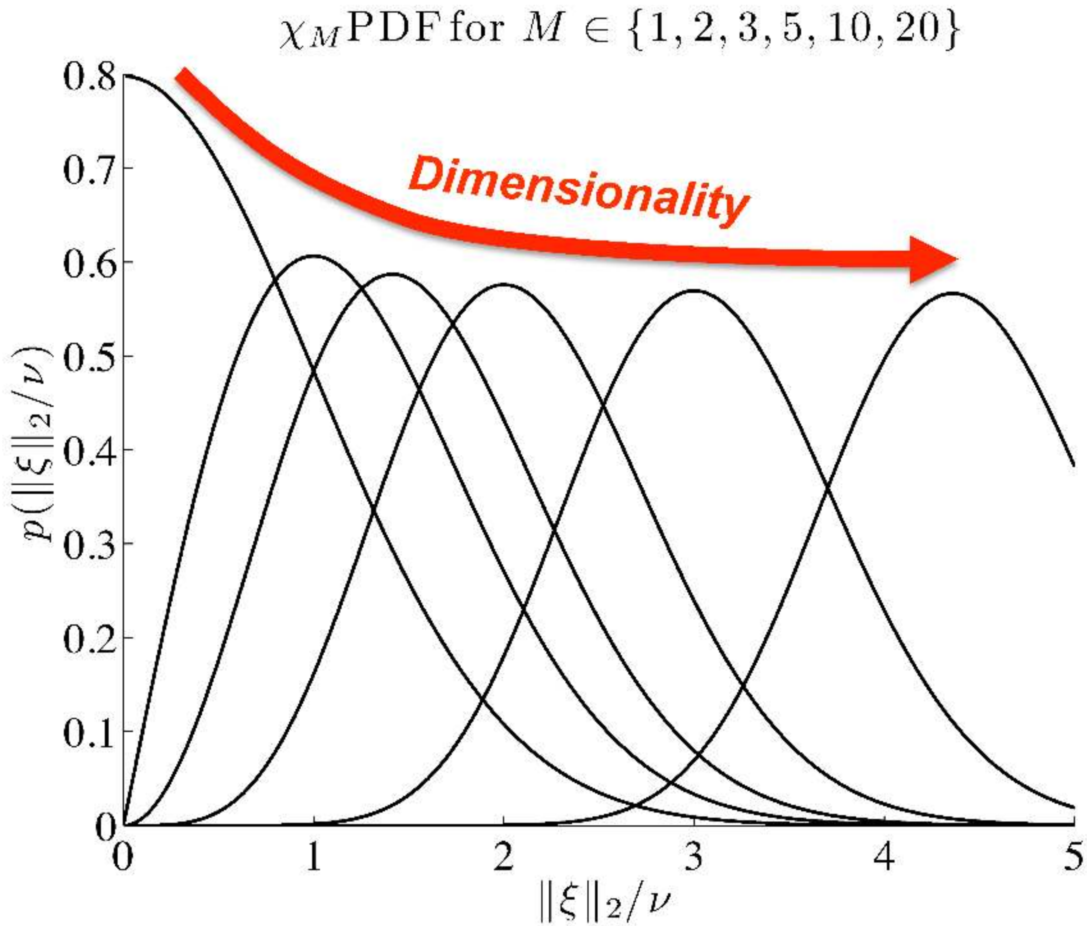



3D views of 4D and 8D hypercubes



$$V_r(d) = \frac{\pi^{d/2}}{\Gamma(d/2+1)} \cdot r^d$$

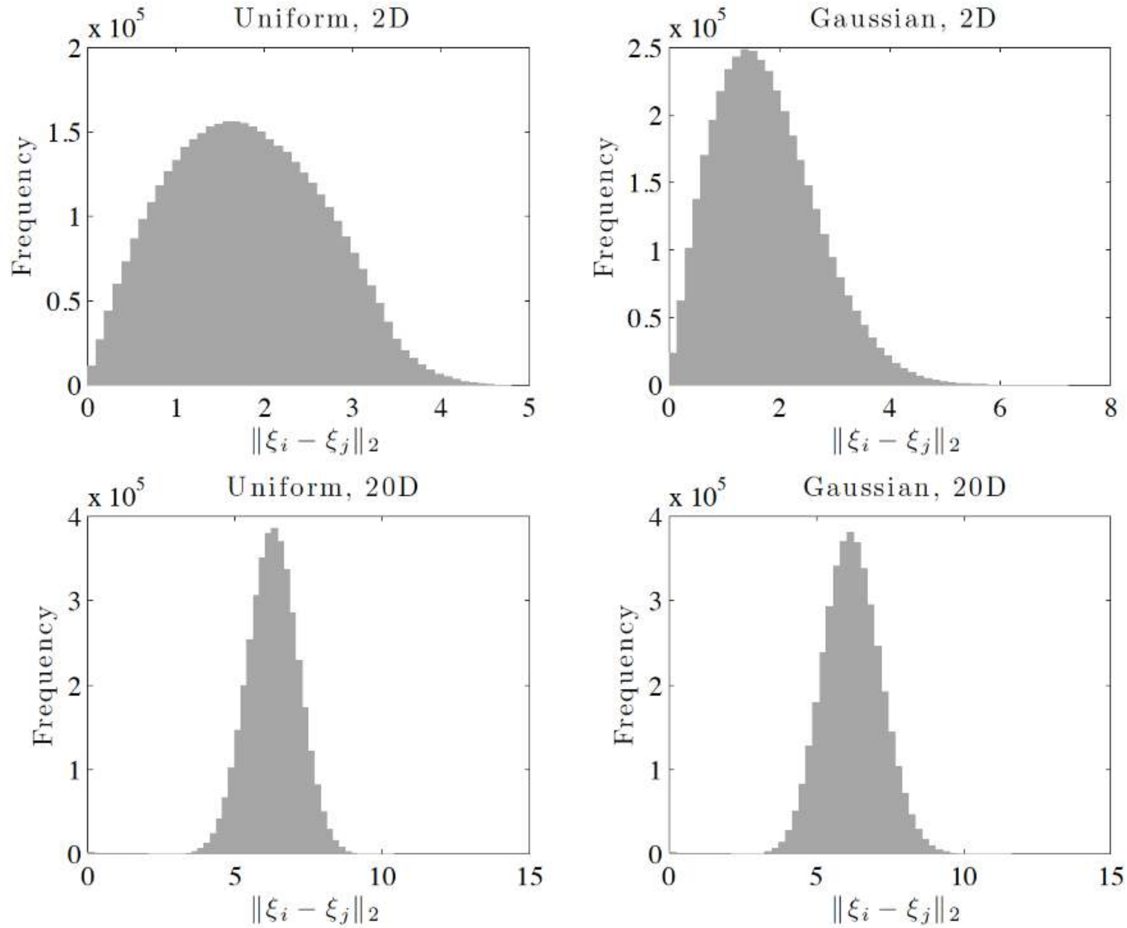
Distributions of Euclidean norms & distances



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

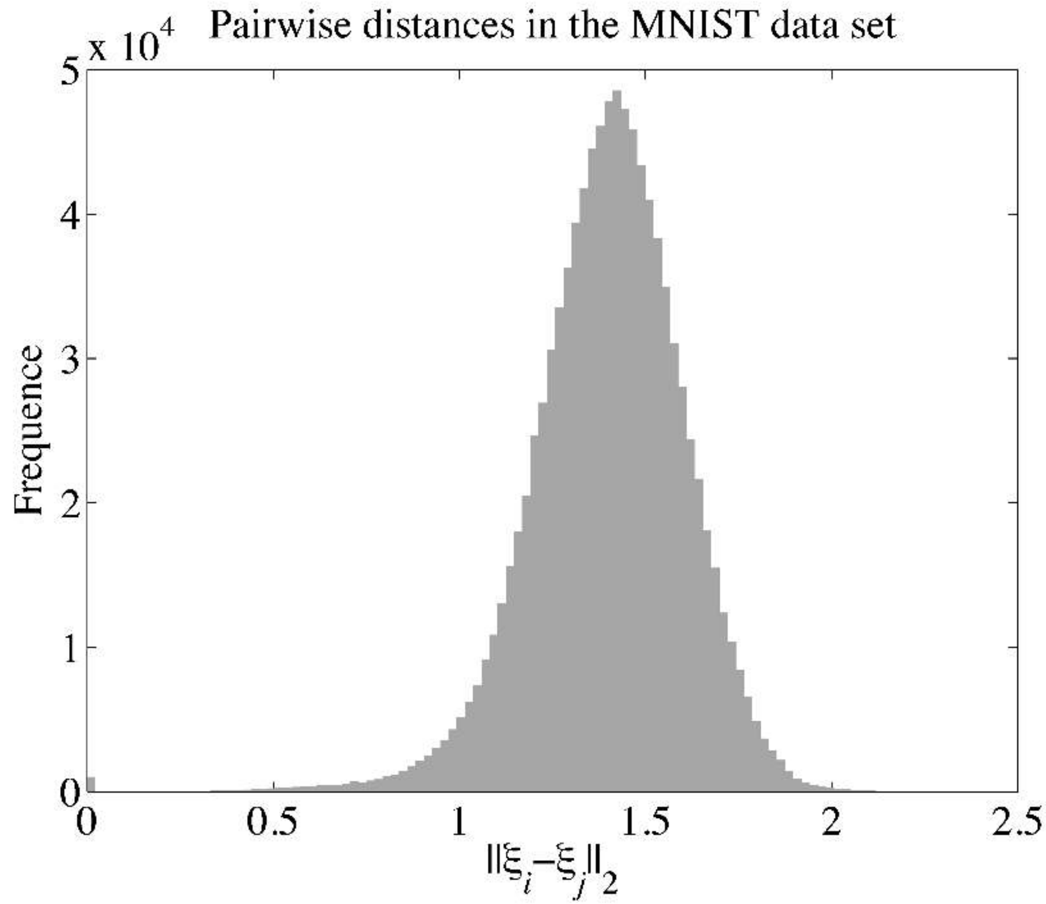
- Shape of Euclidean distance distribution marginally affected by shape of vector distribution



Distributions of Euclidean norms & distances

- Shape of Euclidean distance distribution marginally affected by shape of vector distribution

28-by-28 images
=> 784 dimensions



Distance preservation is hopeless

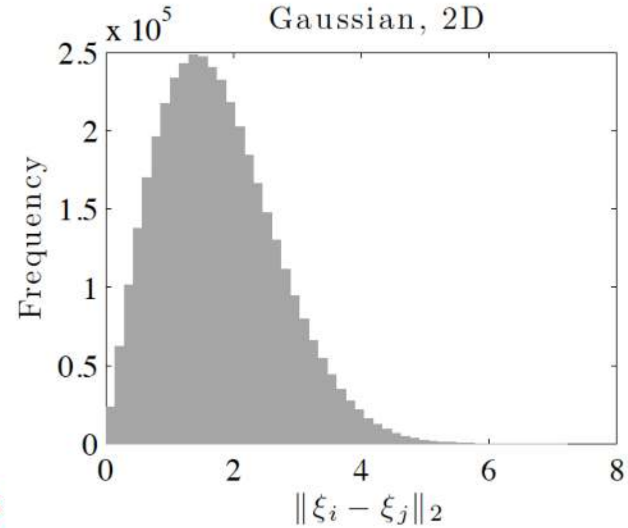
- How can we match this and that?

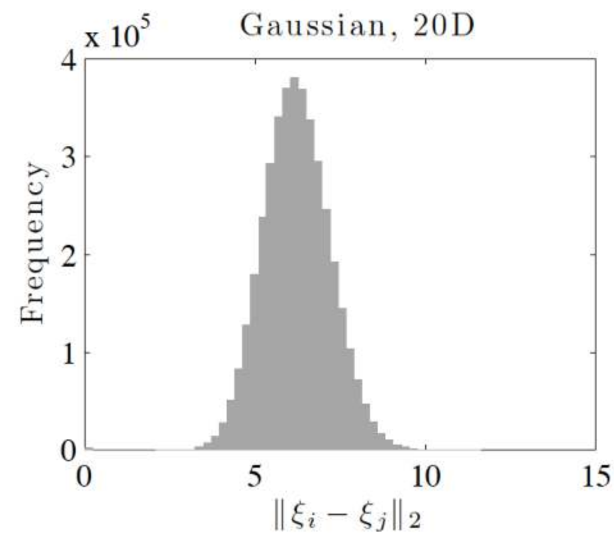
Low-dimensional

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

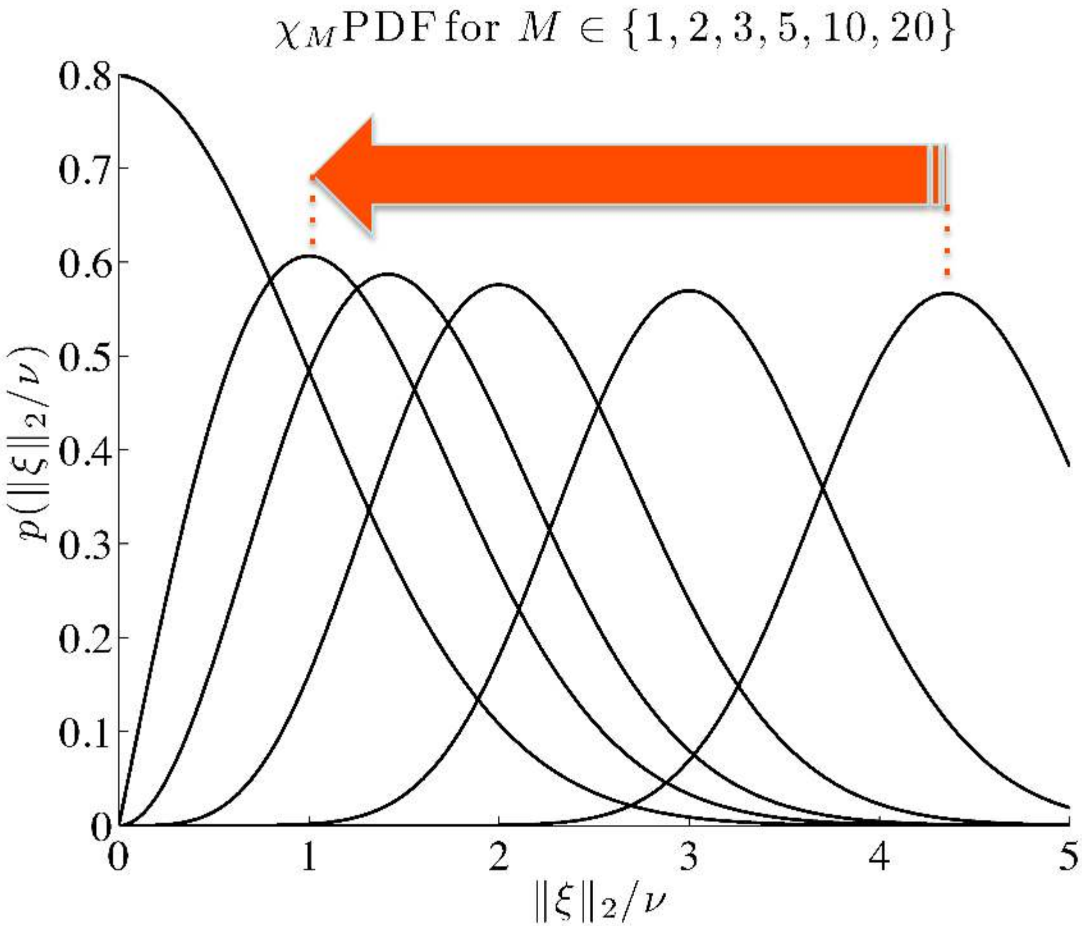
?

High-dimensional



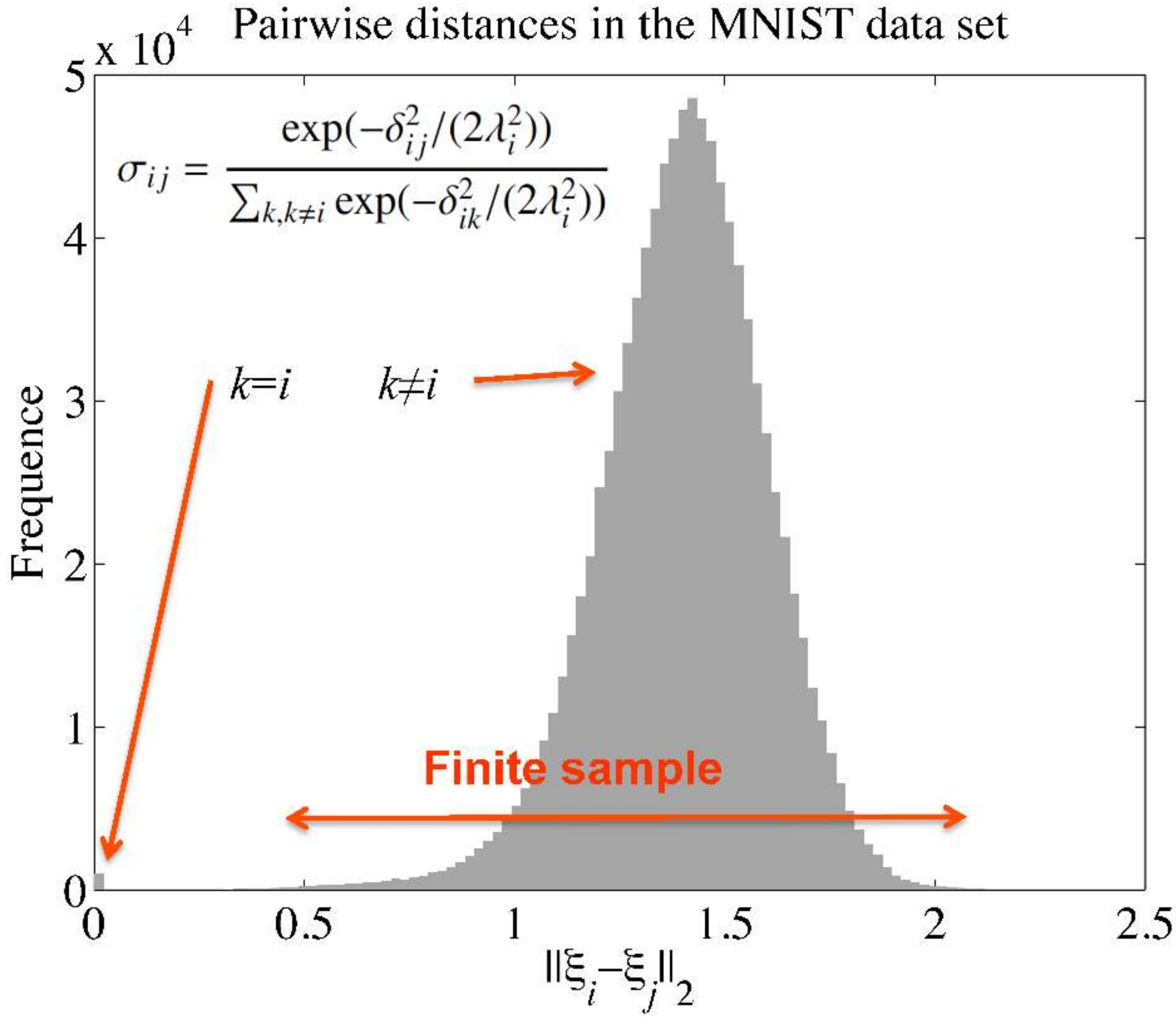


Curse of dimensionality: norm concentration



NLDR from HD to 2D requires a shift!

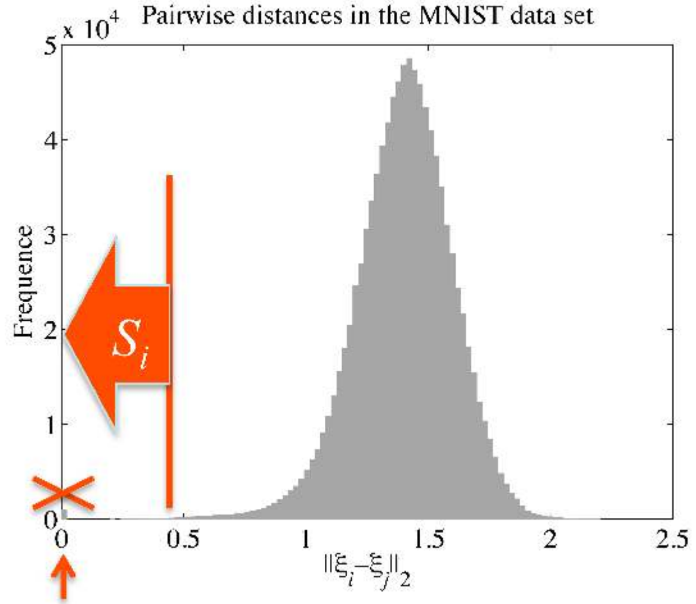
Curse of dimensionality: norm concentration



Shift-invariant similarities

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

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

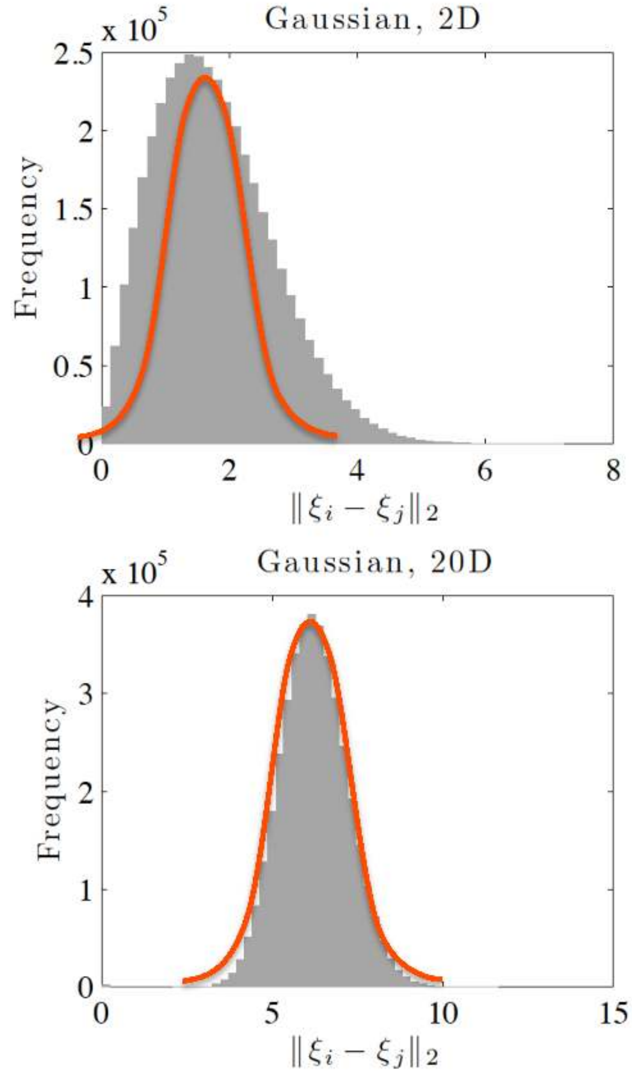


Similarity preservation is hopeful

- We can easily figure out how to match this and that with a shift…

Low-dimensional

High-dimensional



More flexible cost functions

- Starting point: single KL divergence
  - Asymmetric terms in $D_{\mathrm{KL}}(\boldsymbol{\sigma}_i \| \mathbf{s}_i) = \sum_{j=1}^{N} \sigma_{ij} \log(\sigma_{ij}/s_{ij})$
  - Weights of the log terms depend on σ_{ij} only, not on s_{ij}
- Type 1 mixture of KLs

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

- Type 2 mixture of KLs

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

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

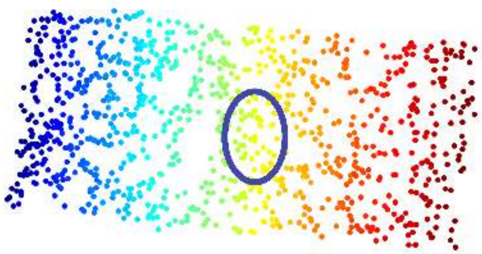
➔ Jensen-Shannon divergence

QA: neighbourhood agreement

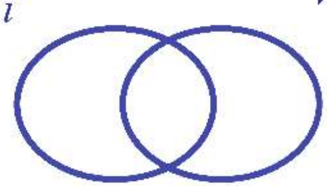
For all pairs of points…

	Near	**Far**
Near	☺	☹
Far	😐	☺

K-ary neighbourhood in LD

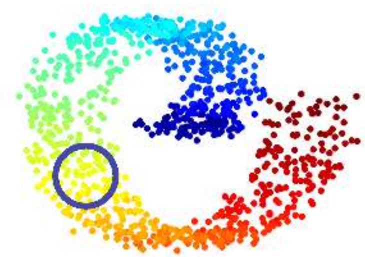


n_i^K



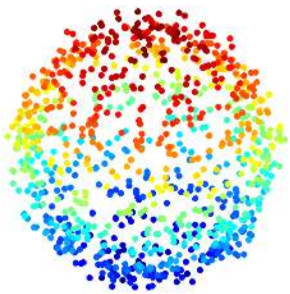
v_i^K

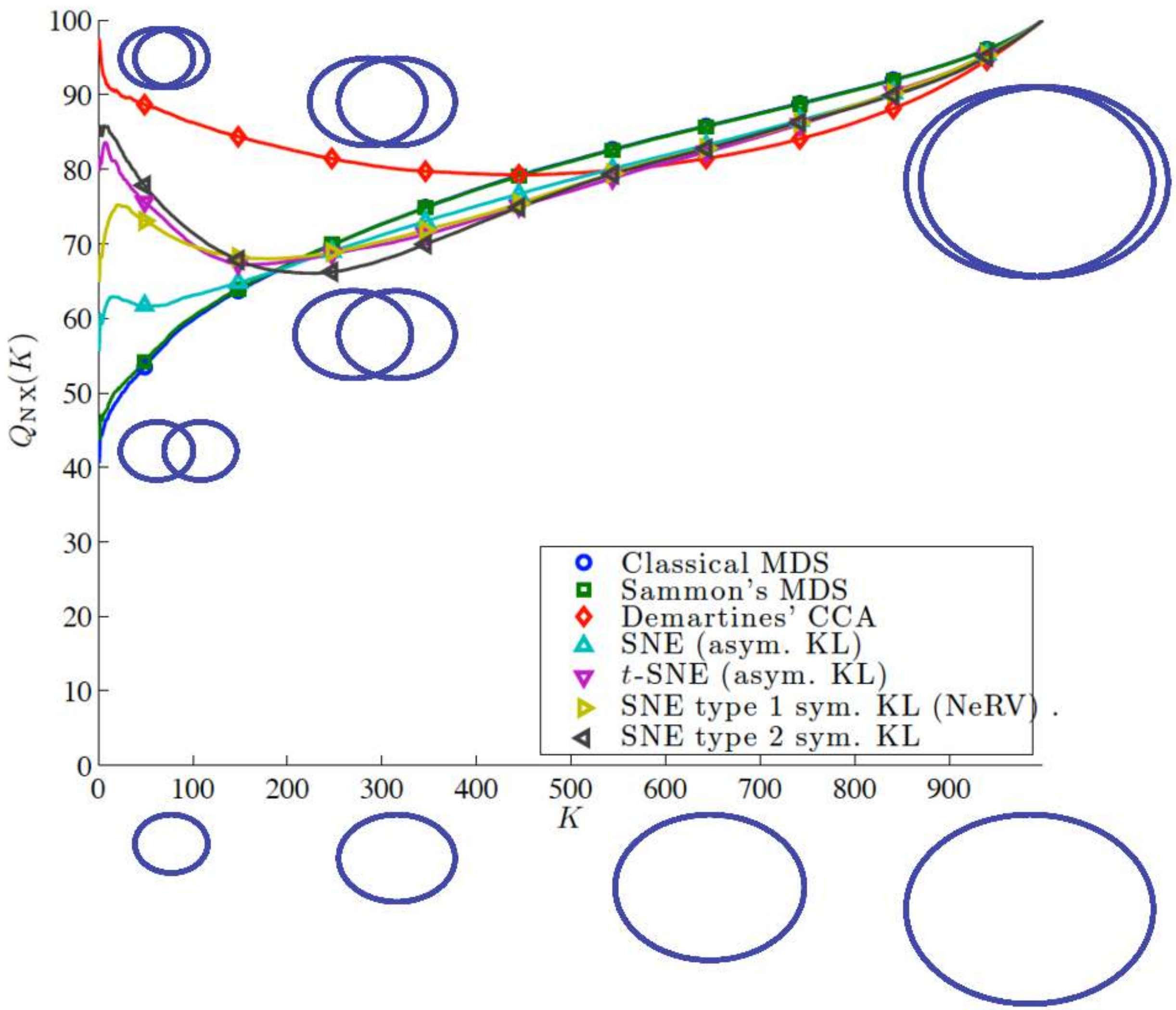
K-ary neighbourhood in HD



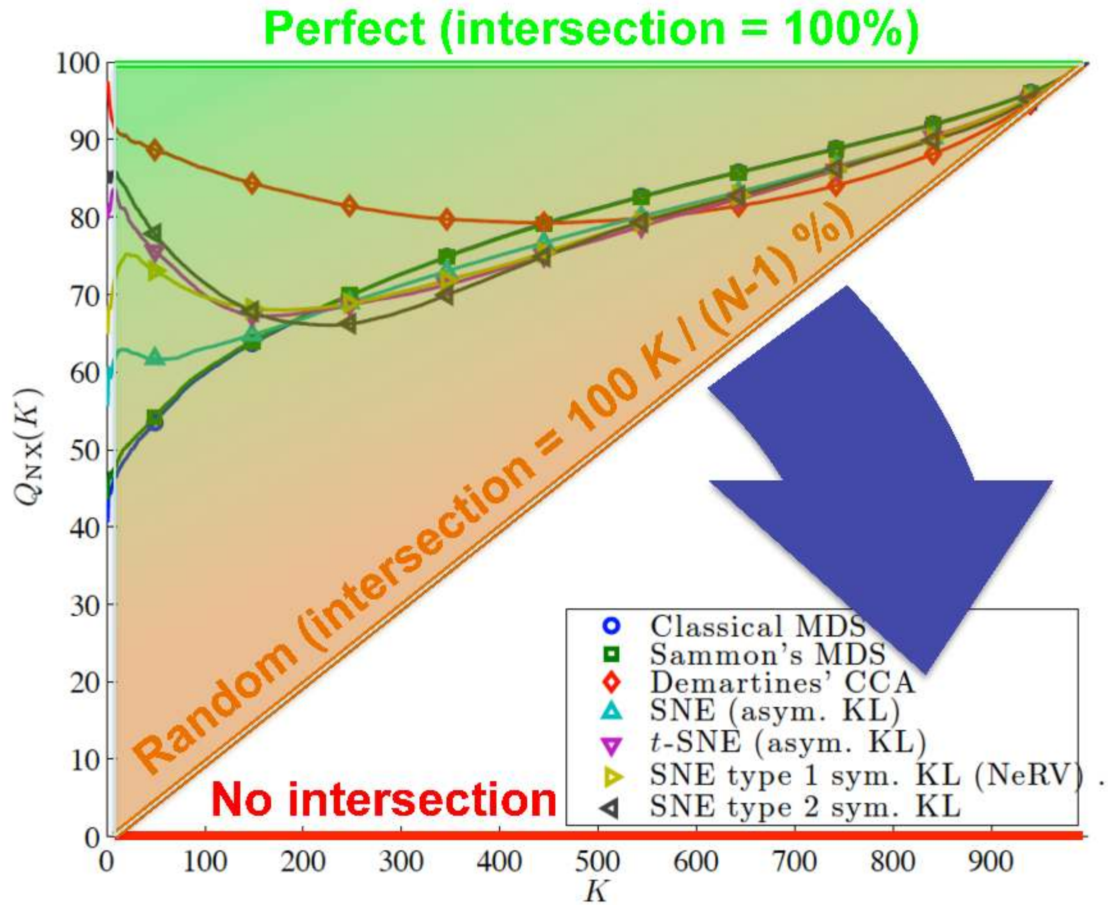
$$Q_{\mathrm{NX}}(K) = \sum_{i=1}^{N} \frac{\left|\nu_i^K \cap n_i^K\right|}{KN}$$

Neighbourhood agreement curve





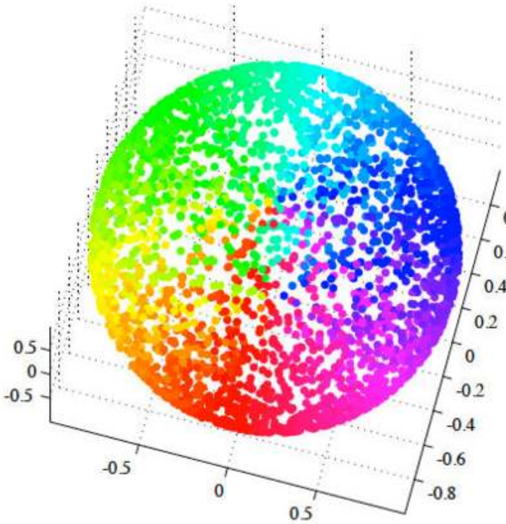
Improvement w.r.t. random embedding



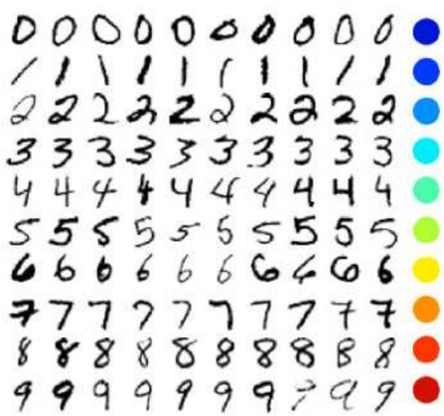
$$R_{\mathrm{NX}}(K) = 100\frac{N-1}{N-K}\left(Q_{\mathrm{NX}}(K) - \frac{K}{N-1}\right)$$

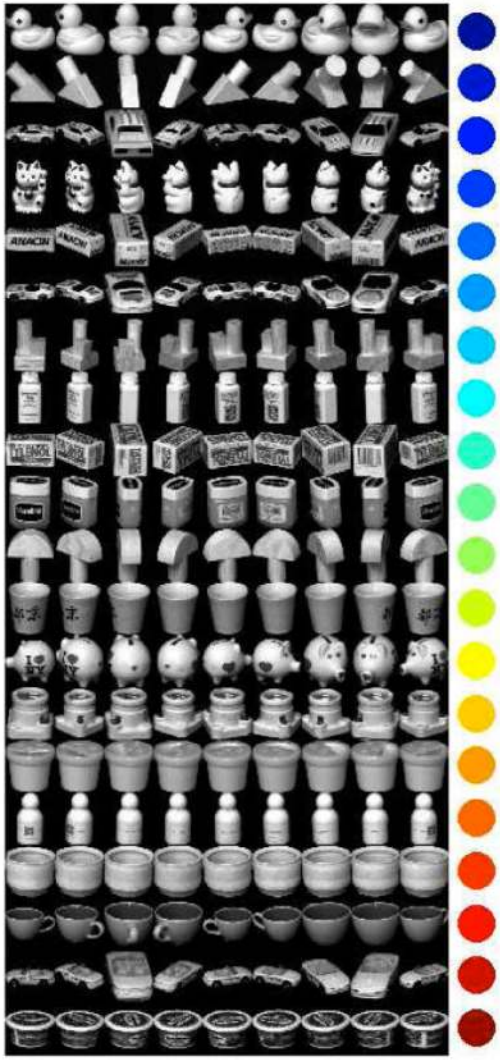
A few experiments and results…

N = 3000
3D to 2D



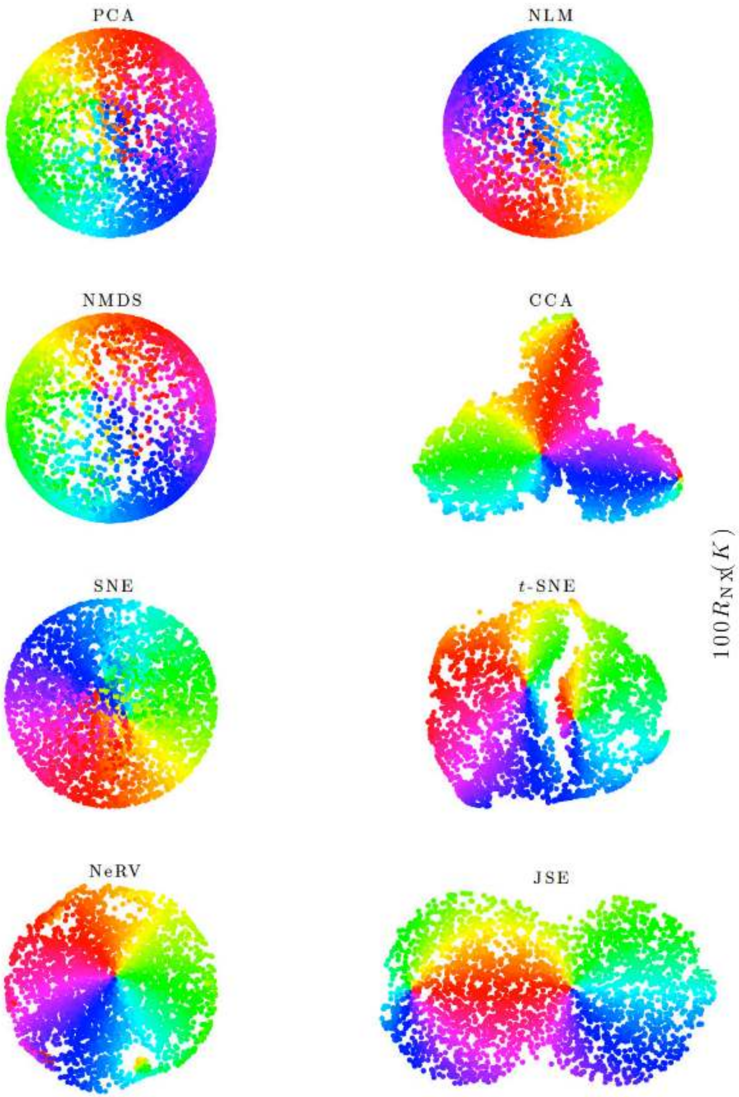
N = 6000
784D to 2D
10 digits



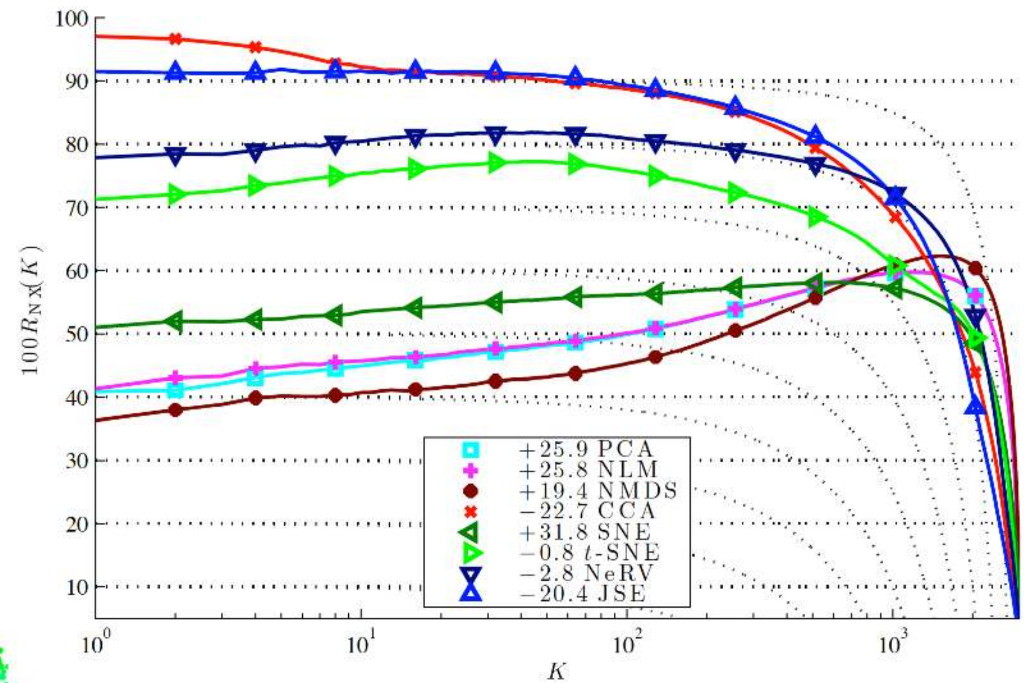


N = 3000
3D to 2D
20 objects

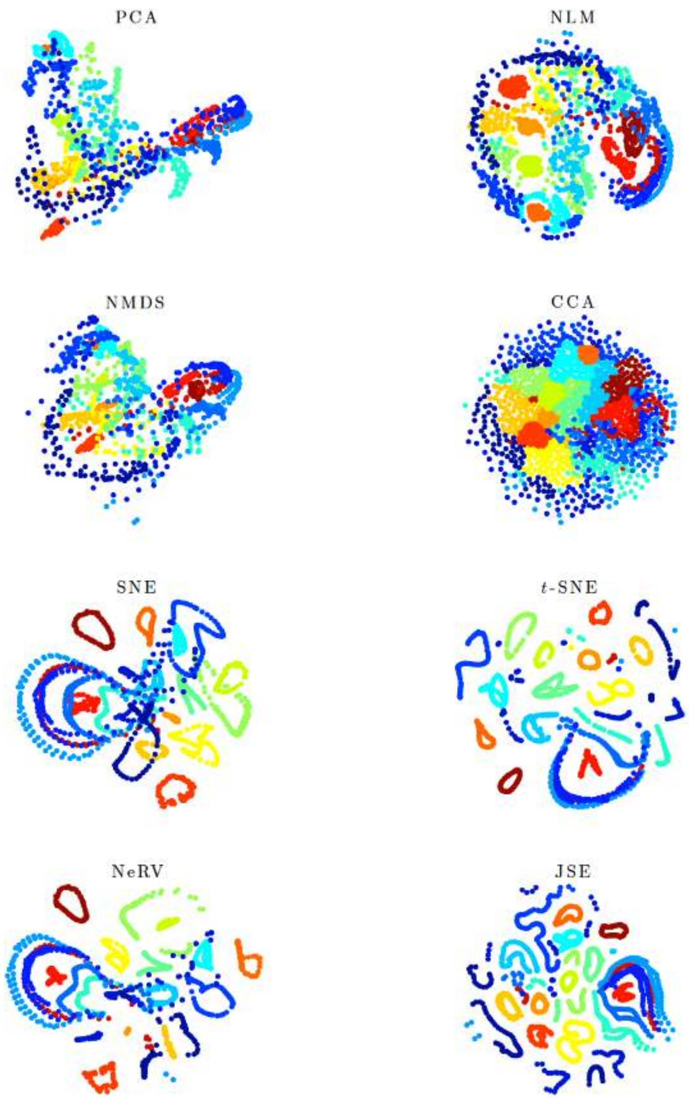
Spherical shell



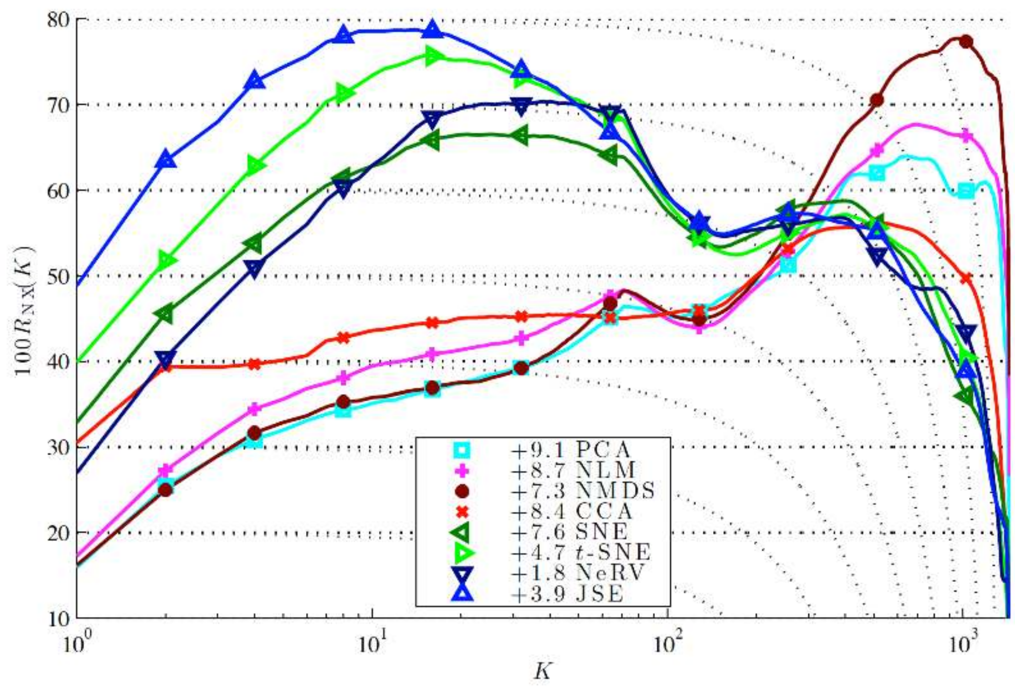
N = 3000, 3D to 2D



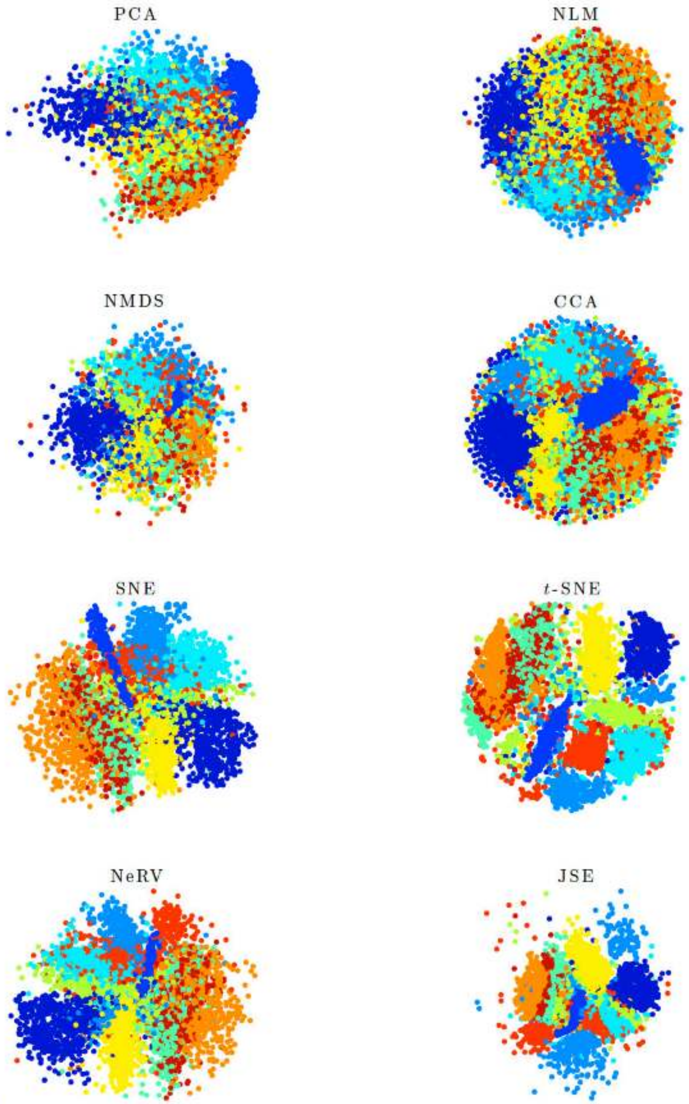
COIL-20 rotated objects



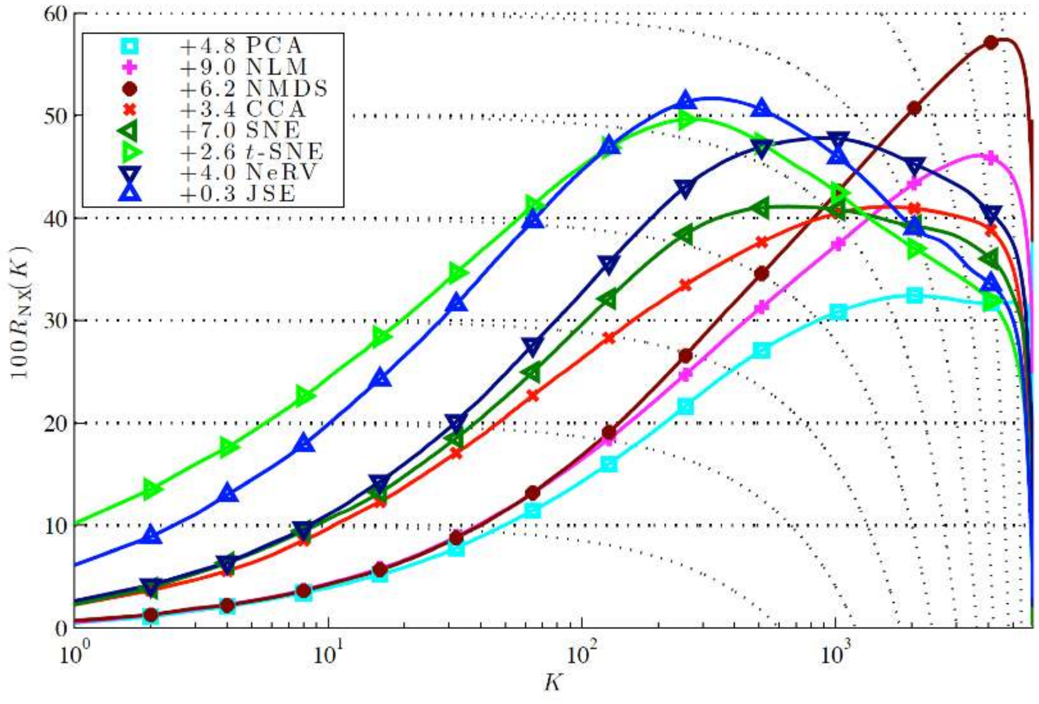
N = 1440, 16384D to 2D

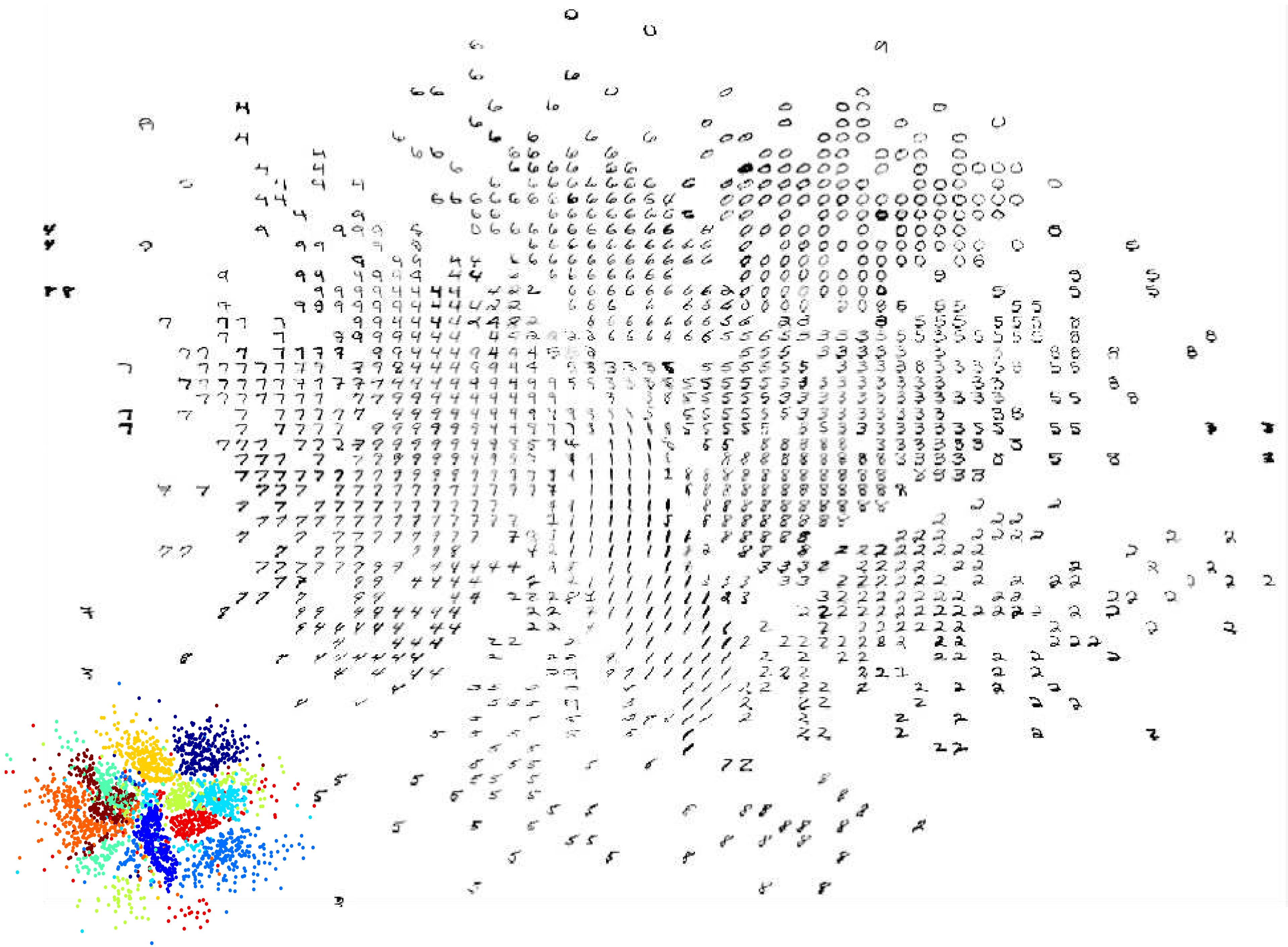


MNIST handwritten digits



N = 6000, 784D to 2D





Conclusions

- Images can be
  - Encoded in a HD (high-dim.) space: pixel space, feature space…
  - Displayed in a LD (low-dim.) space: paper sheet, computer screen…
- Dimensionality reduction
  - Faithful LD representation of HD data
- Distance preservation
  - Intuitive but flawed paradigm in case of very HD data (norm and distance concentration)
- Similarity preservation
  - New successful paradigm, less affected by concentration
- Dimensionality reduction also applies to
  - Graphs (social networks, authors/citations, collaborative filtering, …)
  - Text mining (author-word co-occurrence)
  - …

Thank you for your attention!

machine
learning
group @ UCL

If you have any question…

Please visit:
http://www.ucl.ac.be/mlg/