High-dimensional data and dimensionality reduction

Pardis Noorzad

Amirkabir University of Technology

Farvardin 1390

We'll be talking about…

I. Data analysis
II. Properties of high-dimensional data
III. Some vocabulary
IV. Dimensionality reduction methods
V. Examples
VI. Discussion

Era of massive data collection

- Usual data matrix: **D** rows, **N** cols
- Rows give different attributes/measurements
- Columns give different observations

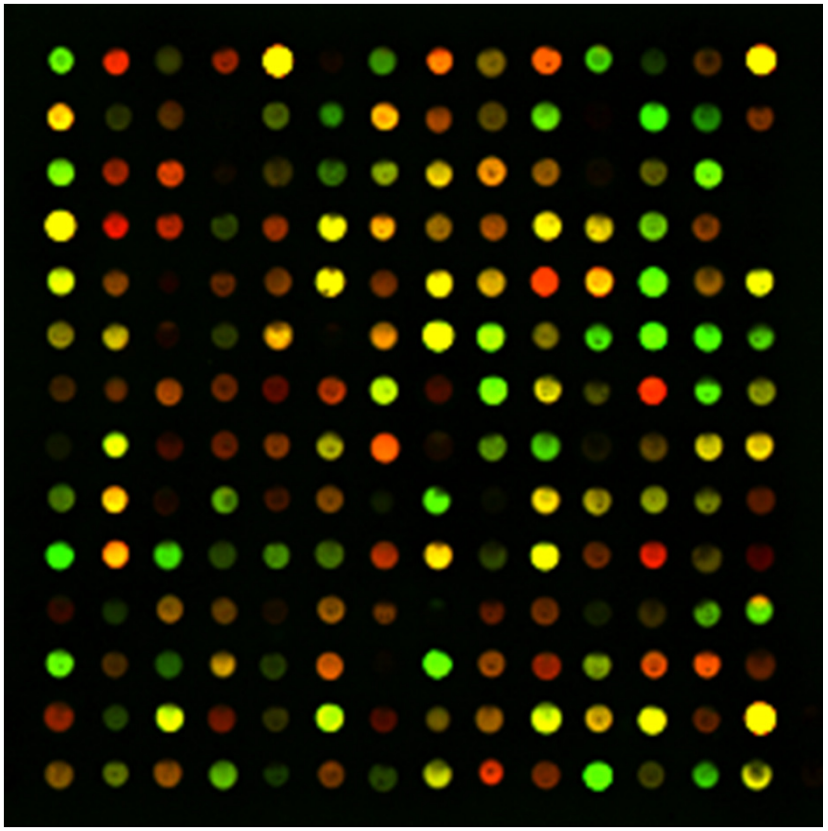
N points in a D-dimensional space

- Term-document data
  - **N** is the number of documents ~ millions
  - **D** is the number of terms ~ thousands
- Consumer preference data (Netflix, Amazon)
  - **N** is the number of individuals ~ millions
  - **D** is the number of products ~ thousands

Problem

- Assumption: **D < N** and **N → ∞**
- Many results fail if **D > N**
- We might have **D → ∞, N** fixed
- Very large number of measurements
  - relatively few instances of the event
- a.k.a. the **large p, small n** problem
  - a.k.a. **High Dimension Low Sample Size (HDLSS)** problem

Example



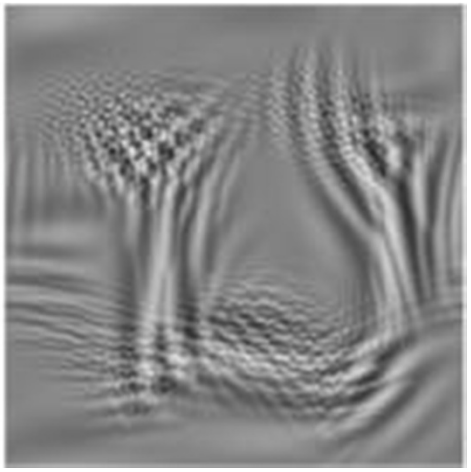
- Breast cancer gene expression data
- Number of measured genes:
  - **D = 6,128**
- Number of tumor samples:
  - **N = 49**

Another example

- Biscuit dough data
- Number of NIR reflectance measures:
  - **D = 23,625**
- Number of dough samples:
  - **N = 39**

Another example: computer vision

- Scene recognition
- Raw Gist feature dimension:
  - **D ~ 300 - 500**
- Number of color image samples:
  - **N ~ 2600**



And another:

- Video concept detection
- Multimedia feature vector:
  - **D ~ 2896**
- Number of video samples:
  - **N ~ 136**

So what?

- **D** is high in our data analysis problems…
- Properties of high dimensional data should be considered
  - Hughes phenomenon
  - Empty space phenomenon
  - Concentration phenomenon

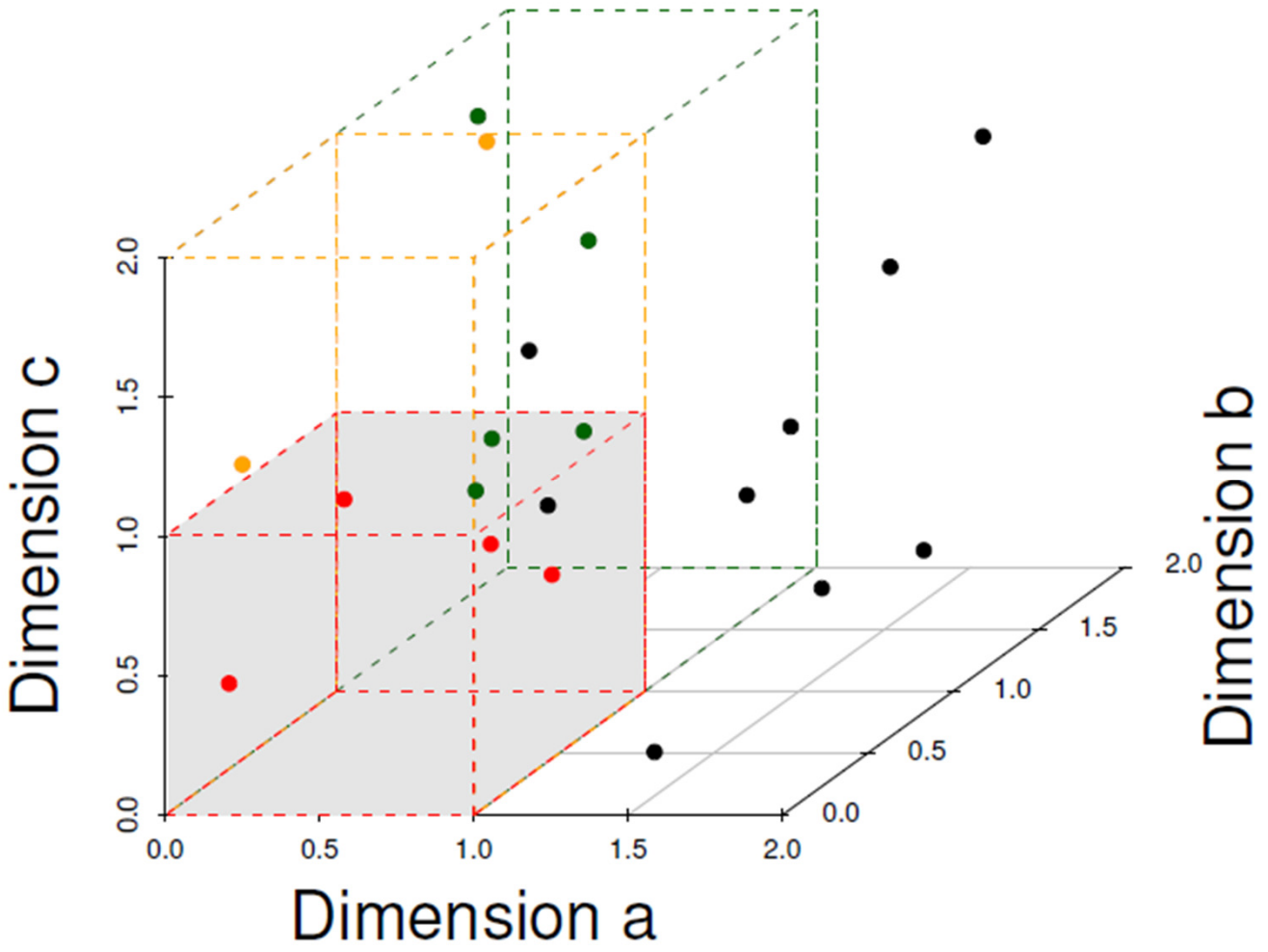
Hughes phenomenon

- a.k.a. **the curse of dimensionality** (R. Bellman, 1961)
- Unit cube in 10 dimensions, discretized with 1/10 spacing → **10^{10}**
- Unit cube in 20 dimensions, same accuracy → **10^{20}** points
- **Number of samples** needed **grows exponentially** with dimension

Empty space phenomenon

- Follows from **COD** and the fact that:
- amount of available data is **limited**
- → high-dimensional space is **sparse**
- You expect an increase in discrimination power (by employing more features)
  - but you lose accuracy
  - due to overfitting

Empty space phenomenon



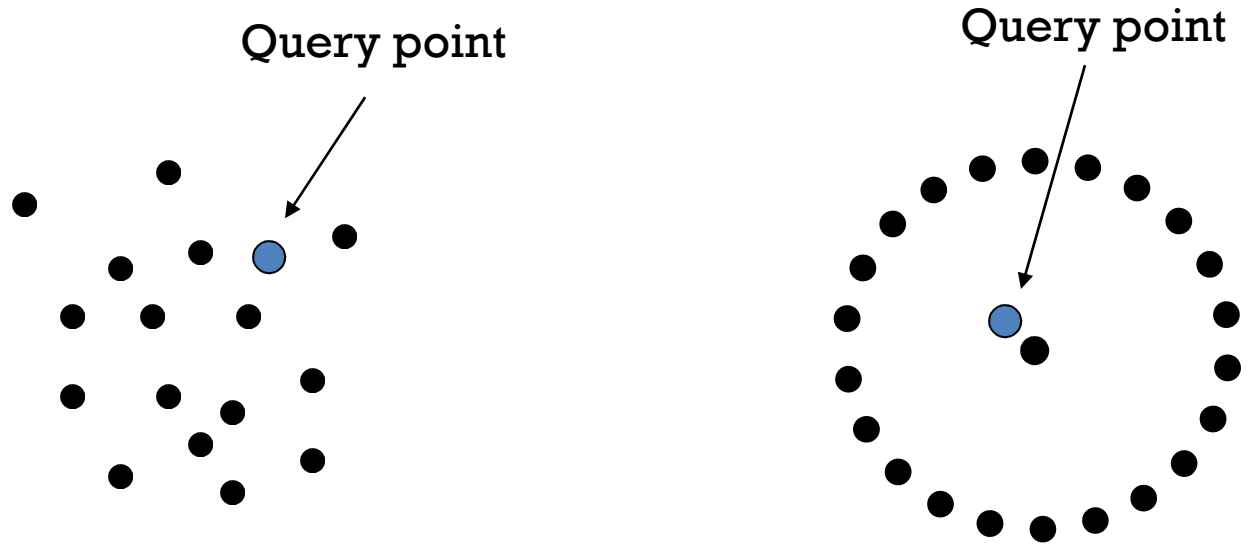
(L. Parsons et al., 2004)

Concentration phenomenon

- "When is nearest neighbor meaningful", (Beyer et al., 1999)
- In high dimensions, under certain conditions,
  - **distance to nearest neighbor** approaches **distance to farthest neighbor**
  - contrast in distances is lost

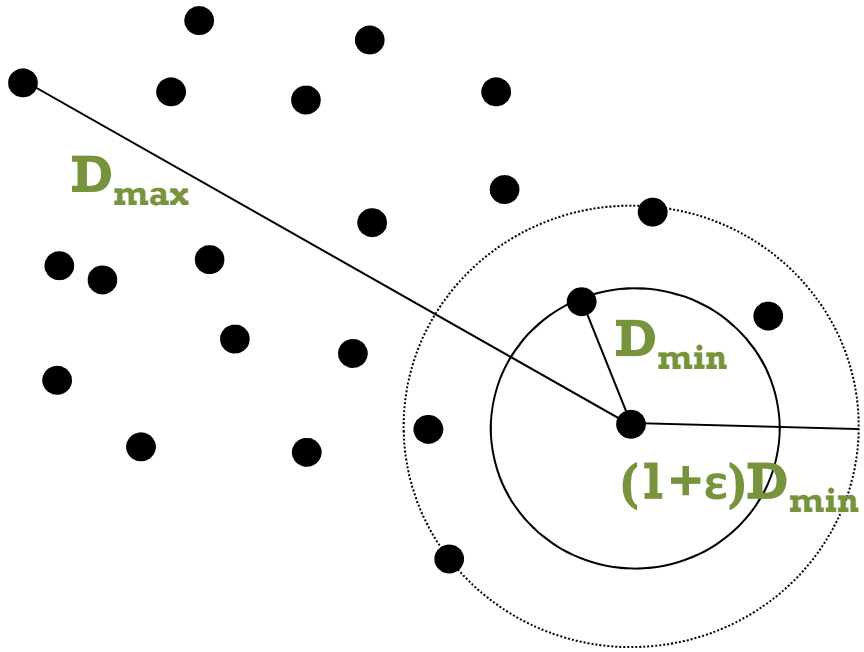
Concentration: continued

- When this happens, there is **no utility** in finding the "nearest neighbor"



Concentration: continued

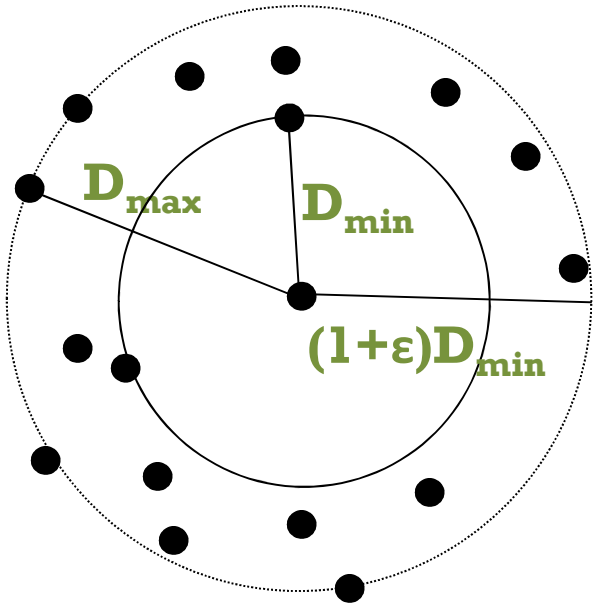
- Definition. **Stable query**



Stable query

Concentration: continued

- Definition. **Unstable query**



$$DMAX \leq (1 + \varepsilon)\ DMIN$$

Concentration: continued

- It is shown that (**under some conditions**), for any fixed $\varepsilon > 0$,
  - as dimensionality rises,
  - the probability that a query is unstable
  - approaches 1

$$\lim_{D \to \infty} \Pr[DMAX_D \leq (1 + \varepsilon)\, DMIN_D] = 1$$

Concentration: i.i.d. case

- Here are some results for i.i.d. dimensions
- Assume:
  - random vector $\mathbf{y} = [y_1, \ldots, y_D]^T$
  - y_i's are i.i.d.
- We'll show:
  - successful drawings of such random vectors yield almost the same norm

Concentration: continued

$$\mu_{\|\mathbf{y}\|} = \sqrt{aD - b} + \mathcal{O}(D^{-1})$$

$$\sigma^2_{\|\mathbf{y}\|} = b + \mathcal{O}(D^{-1/2})$$

The norm of random vectors grows proportionally to $\mathbf{D^{1/2}}$, but the variance remains constant for sufficiently large **D**.

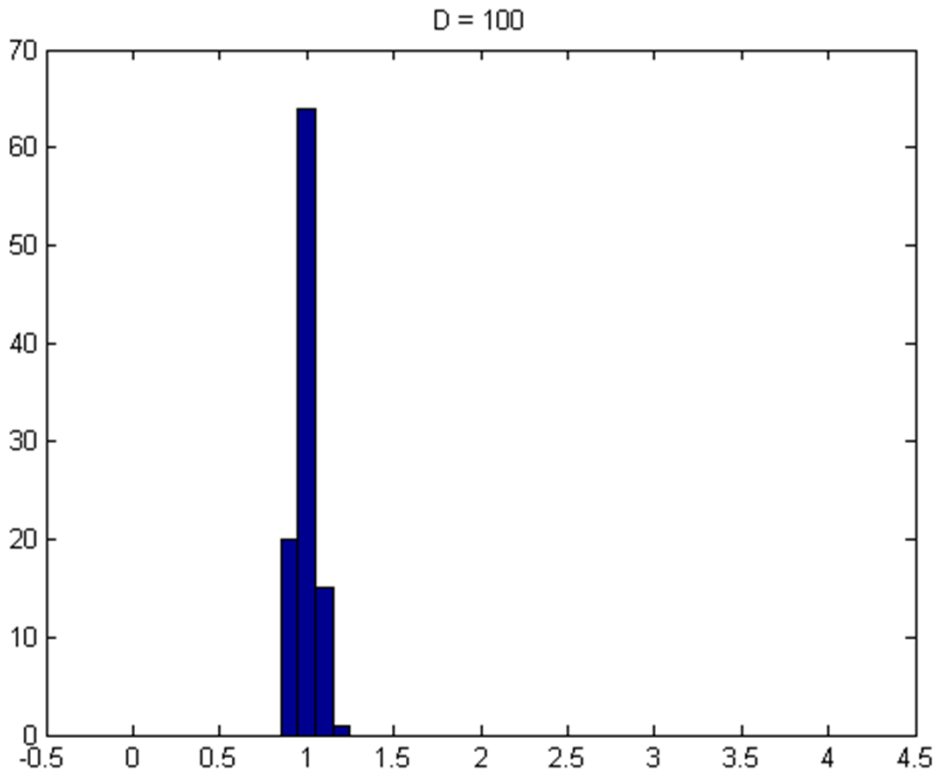
Concentration: continued

$$P\left(\left|\|\mathbf{y}\| - \mu_{\|\mathbf{y}\|}\right| \geq \varepsilon\right) \leq \frac{\sigma^2_{\|\mathbf{y}\|}}{\varepsilon^2}$$

Chebyshev's inequality

(D-1)-sphere

Concentration: simulation results



The relative error tends to zero, meaning that the normalized norm concentrates.

Concentration: continued

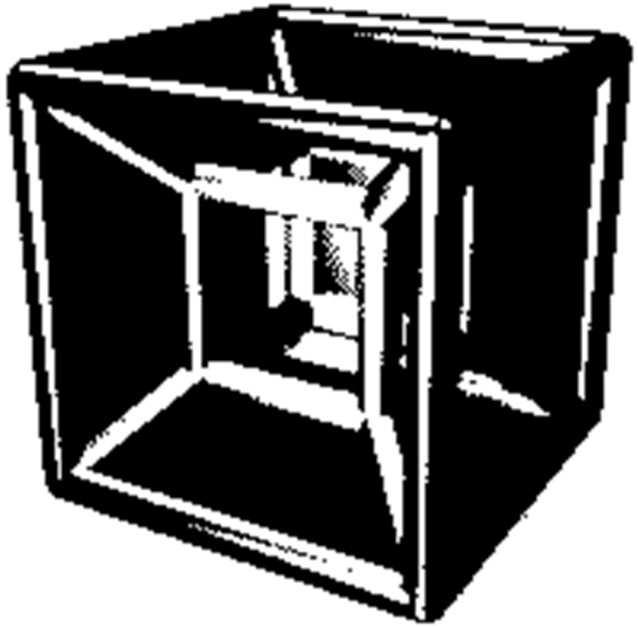
- Where is concentration an issue?
- **NN search**: collection of data points, and query point, find data point closest to query point
  - e.g. used in kNN classification
- Particular interest from vision community
  - each image is approximated with high-dimensional feature vector

Concentration: questions

1. How restrictive are the conditions?
   - sufficient but not necessary
2. When the conditions are satisfied, at what dimensionality do distances become meaningless?
   - about 10-15 (depends on dataset)
3. How can we tell if NN is not meaningful for our dataset?
   - statistical tests? (Casey and Slaney, 2008)
   - how can we fight it? (Houle et al., 2010)

Visualization

- Can be done for up to **4** Ds.



Two approaches to reduce D

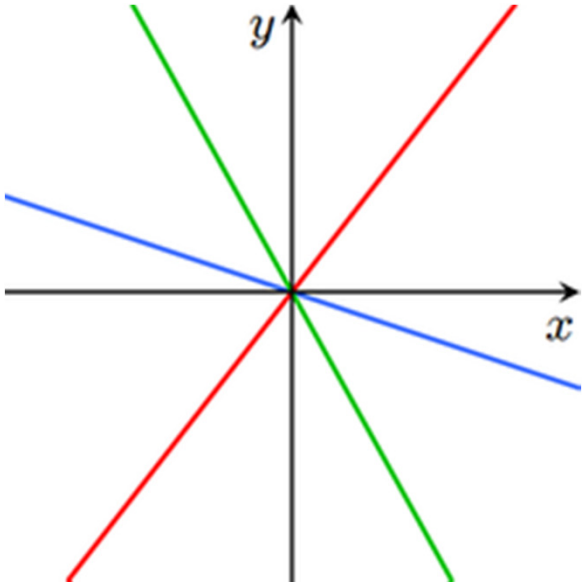
- Feature selection
  - a subset of variables chosen
  - techniques are usually supervised
    - those not correlated with output are eliminated
- Feature extraction
  - even when assuming all variables are relevant
  - detect and eliminate dependencies

the focus of this talk

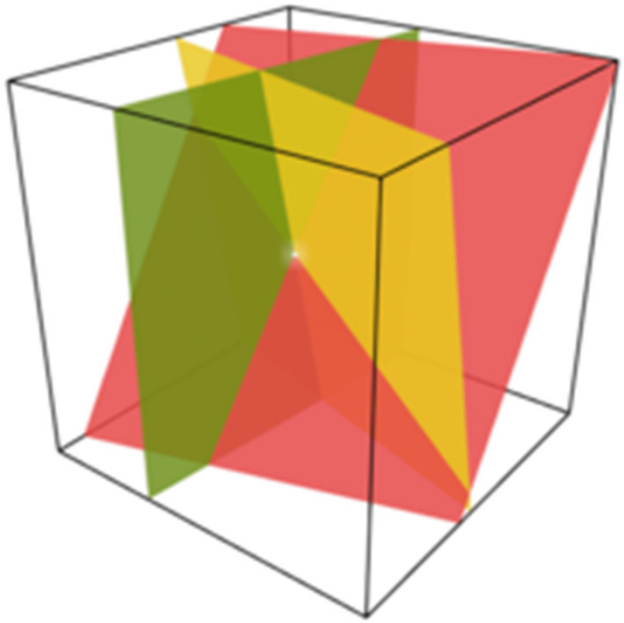
Vocabulary

- So you can develop an intuition about some of the words in the literature
  - Subspace
  - Manifold
  - Embedding
  - Intrinsic dimensionality

Subspace



three 1D subspaces of $\mathbb{R}^2$



three 2D subspaces of $\mathbb{R}^3$

Manifold-- but first some topology

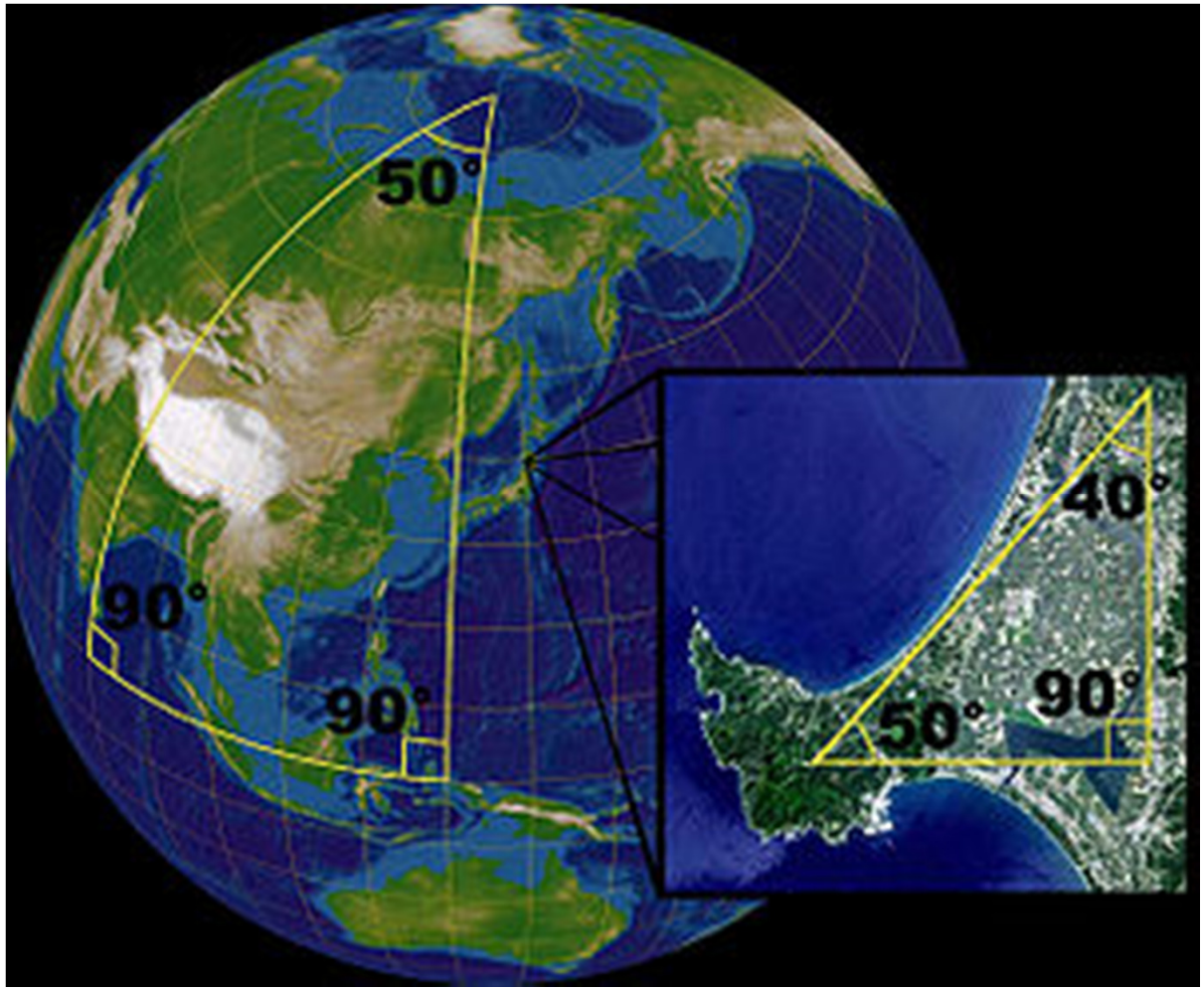
Spatial properties that are preserved under continuous deformations of object

- **Twisting, stretching**
- ~~**Tearing, gluing**~~

"a topologist can't distinguish a coffee mug from a doughnut"

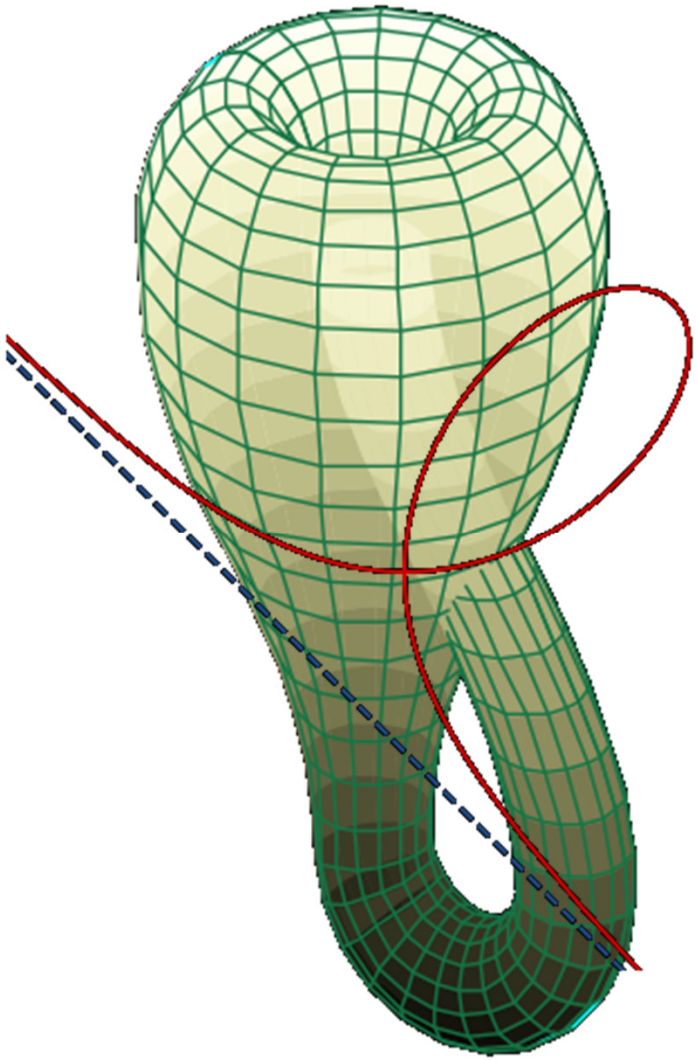
Manifold

locally isomorphic to Euclidean space



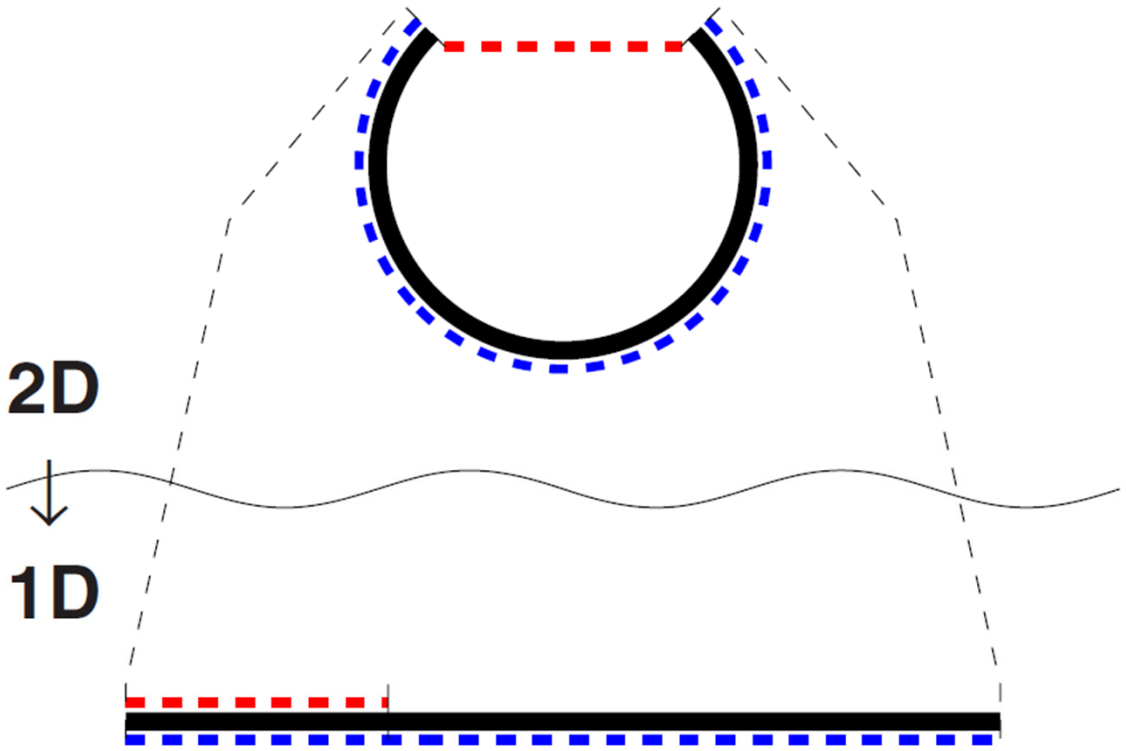
Embedding and p-manifold

- Embedding
- P-manifold
- Every **curve** is a 1-manifold
- Every **surface** is a 2-manifold

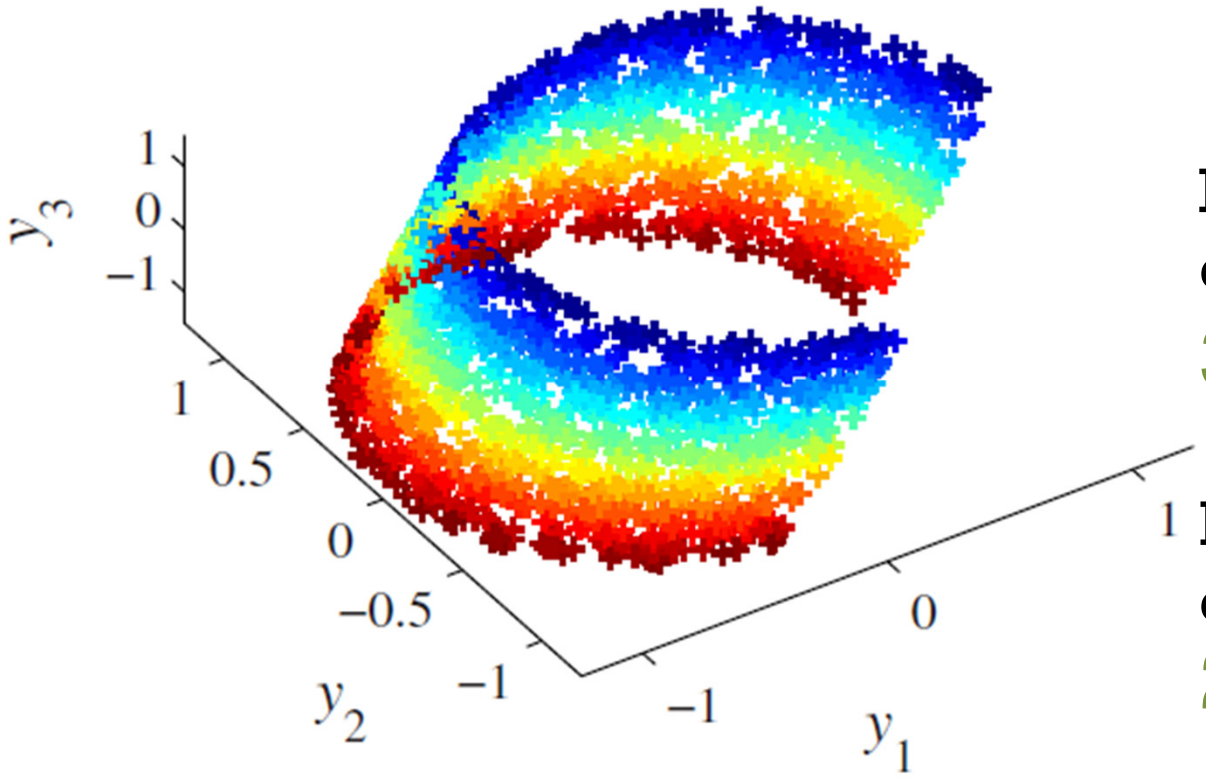


Dimensionality reduction (DR)

- Re-embedding a manifold from a high-dimensional space to a low-dimensional one
- **s.t.** manifold structure is preserved (**connectivity** and **local relationships**)
- one-to-one mapping



Data dimension and intrinsic dimension



Data dimensionality =

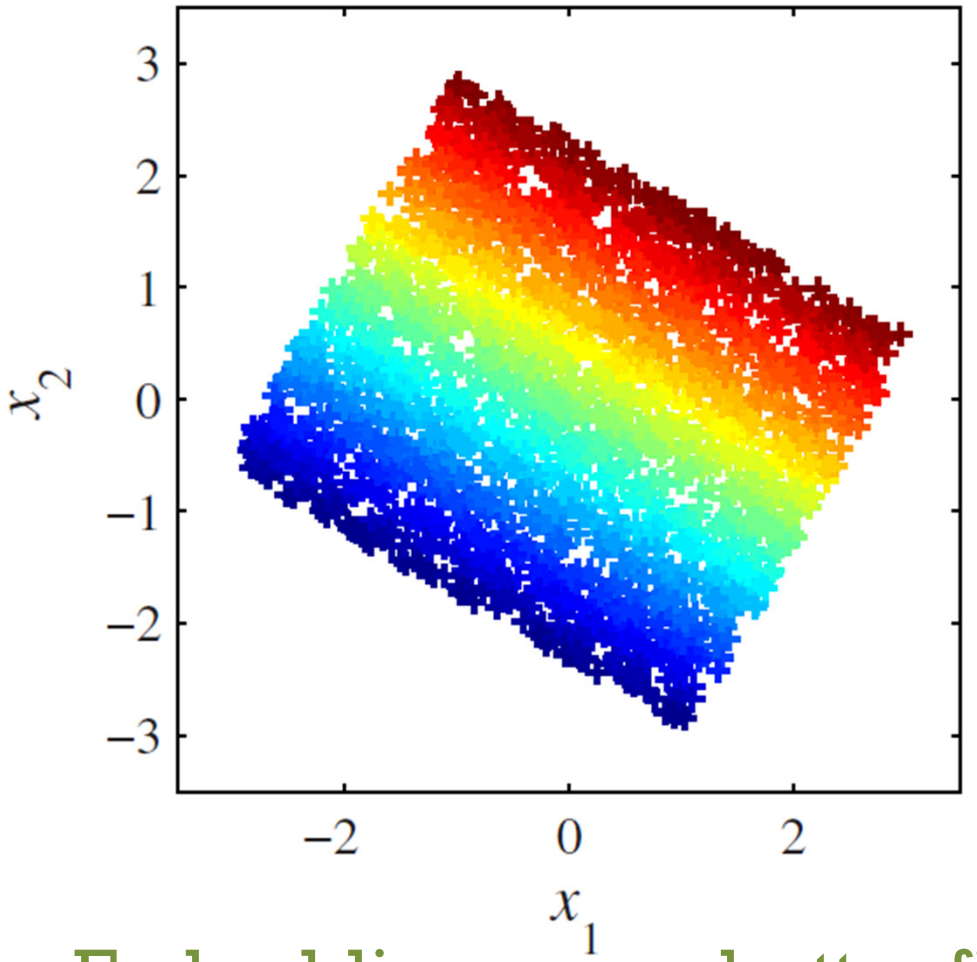
3

Intrinsic dimensionality =

2

Data does not completely fill the embedding space.

A new embedding

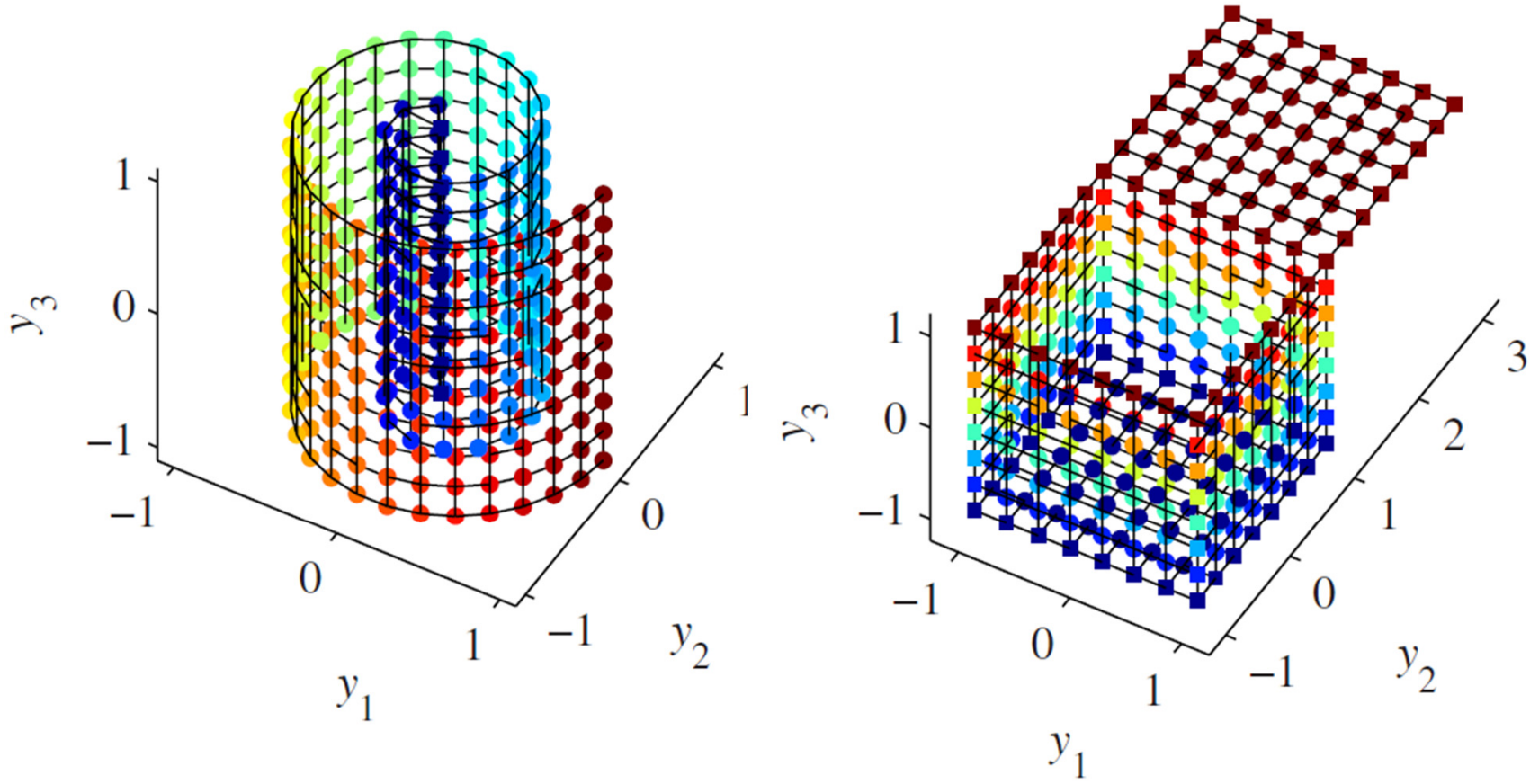


Data dimensionality =
2

Intrinsic dimensionality =
2

Embedding space better filled.

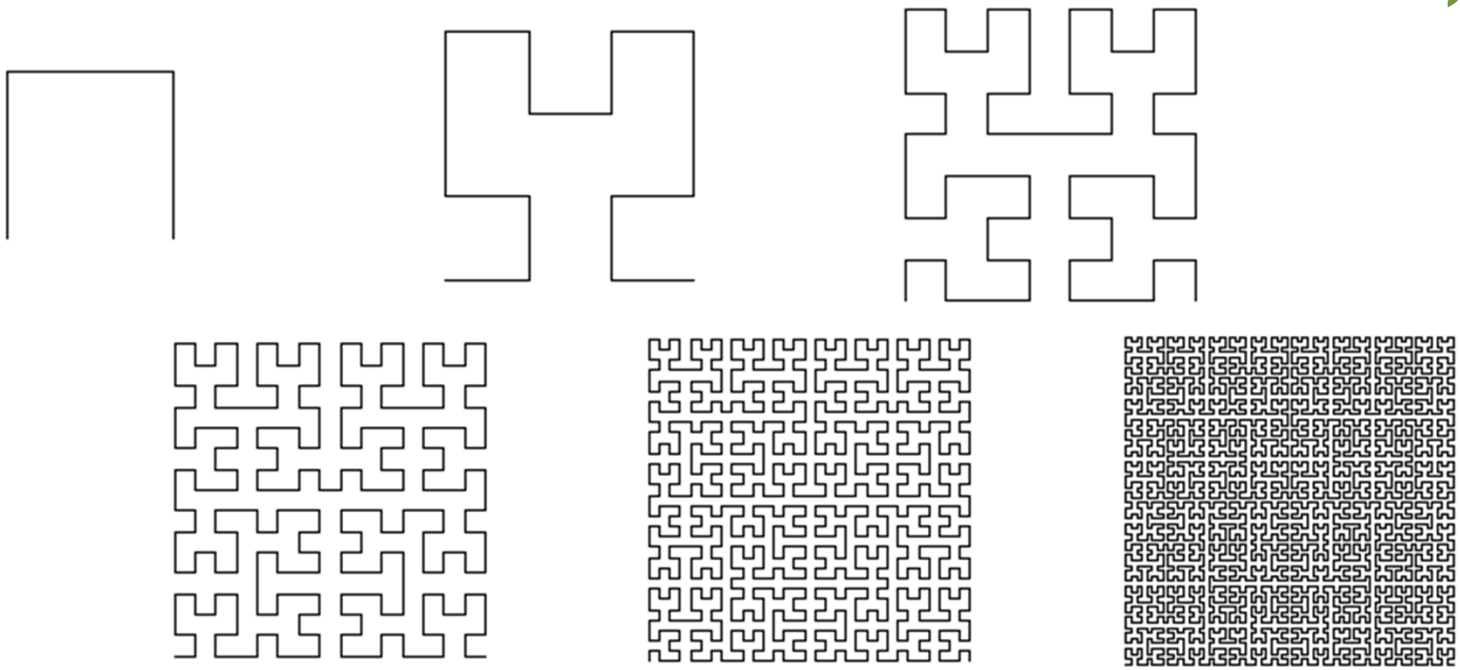
Datasets



(J. A. Lee, and M. Verleysen, 2007)

Datasets: intrinsic dimensionality

- It depends how you define the information content of data
- Several algorithms have been proposed to estimate it (J. A. Lee, and M. Verleysen, 2007)



Subspace learning (linear DR)

- Assume a linear model of data
  - (a linear relation between observed and latent variables)
- We'll look at
  - **PCA (Principal Component Analysis)**
  - **classical metric MDS (Multidimensional Scaling)**
  - **RP (Random Projections)**

PCA (H. Hotelling, 1933)

$$\mathbf{y} = [y_1, \ldots, y_d, \ldots, y_D]^T$$

$$\mathbf{x} = [x_1, \ldots, x_p, \ldots, x_P]^T$$

$$\mathbf{y} = \mathbf{W}\mathbf{x} \qquad \mathbf{W}^T\mathbf{W} = \mathbf{I}_P$$

$$E_{\mathbf{y}}\{\mathbf{y}\} = \mathbf{0}_D$$

$$E_{\mathbf{x}}\{\mathbf{x}\} = \mathbf{0}_P$$

Given: data coordinates

$$\mathbf{Y} = [\mathbf{y}(1), \ldots, \mathbf{y}(n), \ldots, \mathbf{y}(N)]$$

PCA: continued

$$\mathbf{C}_{\mathbf{yy}} = E\{\mathbf{y}\mathbf{y}^T\}$$

$$\mathbf{C}_{\mathbf{yy}} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$$

$$\mathbf{W} = \mathbf{V}\mathbf{I}_{D\times P}$$

$$\hat{\mathbf{x}} = \mathbf{I}_{P\times D}\mathbf{V}^T\mathbf{y}$$

MDS (I. Borg and P. Groenen, 1997)

$$s_{\mathbf{y}}(i,j) = s(\mathbf{y}(i), \mathbf{y}(j)) = \langle \mathbf{y}(i) \cdot \mathbf{y}(j) \rangle$$

$$\begin{aligned}
\mathbf{S} = [s_{\mathbf{y}}(i,j)]_{1 \le i,j \le N} &= \mathbf{Y}^T \mathbf{Y} \\
&= (\mathbf{W}\mathbf{X})^T (\mathbf{W}\mathbf{X}) \\
&= \mathbf{X}^T \mathbf{W}^T \mathbf{W} \mathbf{X} \\
&= \mathbf{X}^T \mathbf{X} \ .
\end{aligned}$$

Given: matrix of scalar products

MDS: continued

$$\begin{aligned}\mathbf{S} &= \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \\ &= (\mathbf{U}\mathbf{\Lambda}^{1/2})(\mathbf{\Lambda}^{1/2}\mathbf{U}^T) \\ &= (\mathbf{\Lambda}^{1/2}\mathbf{U}^T)^T(\mathbf{\Lambda}^{1/2}\mathbf{U}^T)\end{aligned}$$

$$\hat{\mathbf{X}} = \mathbf{I}_{P\times N}\mathbf{\Lambda}^{1/2}\mathbf{U}^T$$

MDS: discussion

- Widely used and developed in human sciences
  - particularly **psychometrics**
- People are asked to give qualitative separation between objects
- So each object is characterized by **distances** to other objects

Neither **S** nor **Y** but **distances** available

A technique called **double centering**

$$
\begin{aligned}
d^2_{\mathbf{y}}(i,j) &= \|\mathbf{y}(i) - \mathbf{y}(j)\|_2^2 \\
&= \langle \mathbf{y}(i) - \mathbf{y}(j) \cdot \mathbf{y}(i) - \mathbf{y}(j) \rangle \\
&= \langle \mathbf{y}(i) \cdot \mathbf{y}(i) \rangle - 2\langle \mathbf{y}(i) \cdot \mathbf{y}(j) \rangle + \langle \mathbf{y}(j) \cdot \mathbf{y}(j) \rangle \\
&= s_{\mathbf{y}}(i,i) - 2s_{\mathbf{y}}(i,j) + s_{\mathbf{y}}(j,j) \ ,
\end{aligned}
$$

squared Euclidean distance

$$
s_{\mathbf{y}}(i,j) = -\frac{1}{2}(d^2_{\mathbf{y}}(i,j) - \langle \mathbf{y}(i) \cdot \mathbf{y}(i) \rangle - \langle \mathbf{y}(j) \cdot \mathbf{y}(j) \rangle)
$$

Double centering: continued

$$\mathbf{S} = -\frac{1}{2}(\mathbf{D} - \frac{1}{N}\mathbf{D}\mathbf{1}_N\mathbf{1}_N^T - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\mathbf{D} + \frac{1}{N^2}\mathbf{1}_N\mathbf{1}_N^T\mathbf{D}\mathbf{1}_N\mathbf{1}_N^T)$$

$$\mu_j(d^2_{\mathbf{y}}(i,j)) =$$

$$\langle \mathbf{y}(i) \cdot \mathbf{y}(i) \rangle + \mu_j(\langle \mathbf{y}(j) \cdot \mathbf{y}(j) \rangle)$$

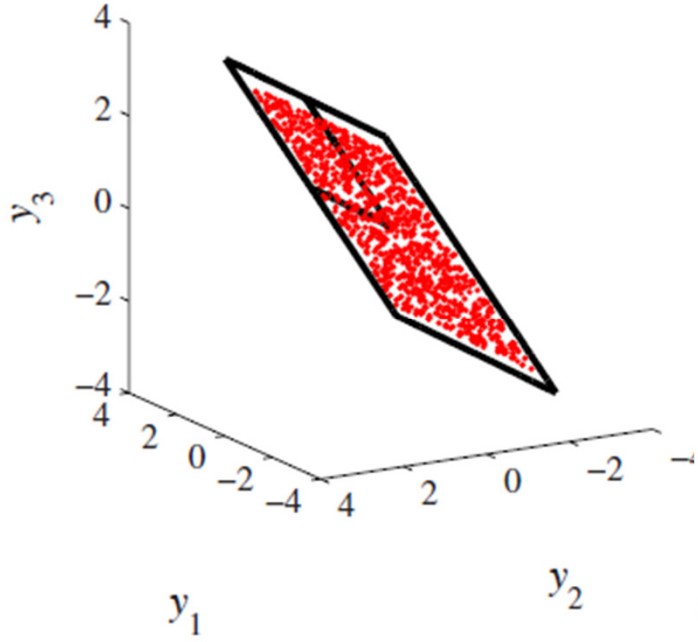
$$\mu_i(d^2_{\mathbf{y}}(i,j)) =$$

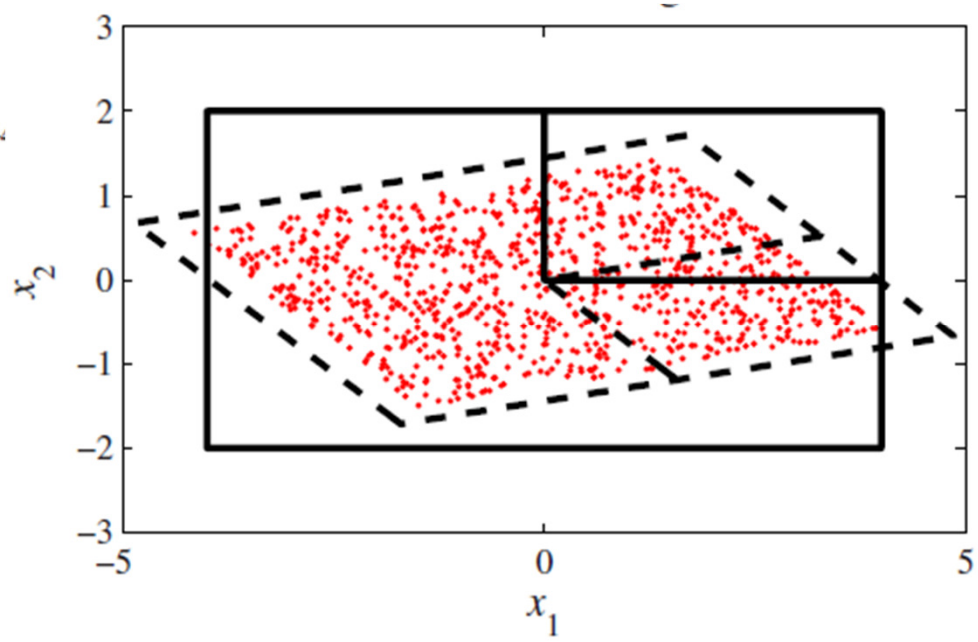
$$\mu_i(\langle \mathbf{y}(i) \cdot \mathbf{y}(i) \rangle) + \langle \mathbf{y}(j) \cdot \mathbf{y}(j) \rangle$$

$$\mu_{i,j}(d^2_{\mathbf{y}}(i,j)) =$$

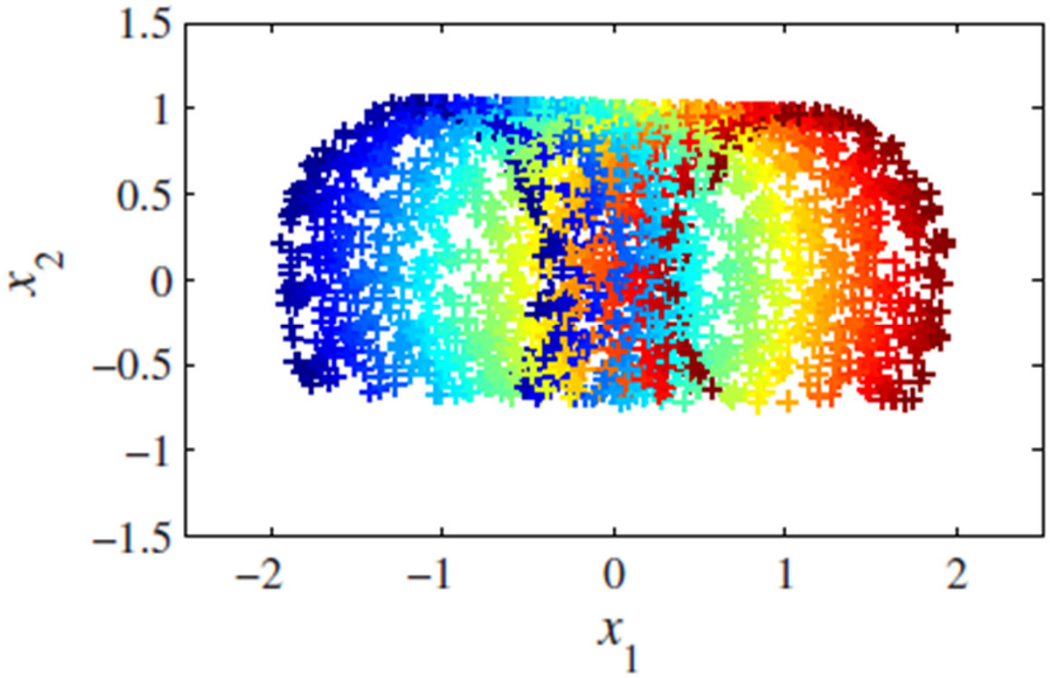
$$\mu_i(\langle \mathbf{y}(i) \cdot \mathbf{y}(i) \rangle) + \mu_j(\langle \mathbf{y}(j) \cdot \mathbf{y}(j) \rangle)$$

PCA/MDS: results





PCA/MDS: results



PCA/MDS: discussion

- Metric MDS and PCA give the same solution
- Both focus mainly on retaining **large pairwise distances**, instead of small pairwise distances which is more important
- Both may consider two points as near points, whereas their distance over the manifold is much larger

Random projections
(W. B. Johnson and J. Lindenstrauss, 1984)

- A **linear** method
- Simple yet powerful
- Randomly chosen low-dimensional subspace
  - the projection doesn't depend on the data
  - a "**data-oblivious**" method

RP: algorithm

- Here's how to obtain the **P×D** linear transform **R** (Dasgupta, 2000)
  1. set each entry of **R** to an i.i.d. ~**N(0,1)** value
  2. make the **P** rows of the matrix orthogonal using the **Gram-Schmidt** algorithm
  3. normalize rows to unit length

RP: the theory behind the algorithm

- **JL Thm.** A set of points of size **n** in a high-dimensional Euclidean space, can be mapped into a **q**-dimensional space, $q \geq O(\log(n)/\varepsilon^2)$, such that the distance between any two points changes by only a factor of $1 \pm \varepsilon$.
  - (W. B. Johnson and J. Lindenstrauss, 1984)

RP theory: continued

(S. Dasgupta and A. Gupta, 1999)

- A matrix whose entries are normally distributed represents such a mapping with probability at least **1/n**, therefore doing **O(n)** projections will result in an arbitrarily high probability of preserving distances.
- Tighter bound obtained:
  - $q \geq 4*(\varepsilon^2/2 - \varepsilon^3/3)^{-1} \ln(n)$

RP: discussions

- It is shown that RP **underperforms** PCA as a preprocessing step in classification (but still remains comparable)
  - (D. Fradkin and D. Madigan, 2003)
- **But**, it is **computationally more attractive** than PCA and can replace it
  - e.g. when initial dimension is ~6000
  - PCA is $O(D^3)$ vs. RP which is $O(P^2D)$
  - with some loss of accuracy, even faster versions of RP have been proposed

Manifold learning

- Manifold assumption
  - **"data lies on a low-dimensional manifold in the high-dimensional space"**
- It's an assumption that helps reduce the hypothesis space
  - *A priori* information on **the support** of the **data distribution**

Manifold learning (nonlinear DR)

- First we'll look at
  - **Isomap (Isometric feature map)** and
  - **LLE (Locally Linear Embedding)**
- Easy to understand/explain
- Both build a graph **G** using **K**-rule: $O(N^2)$
  - A discretized approximation of the manifold, sampled by the input
- Both published in **Science** in **2000**, and lead to the rapid development of **spectral methods for NLDR**
- **~3840** and **~3735** citations, respectively

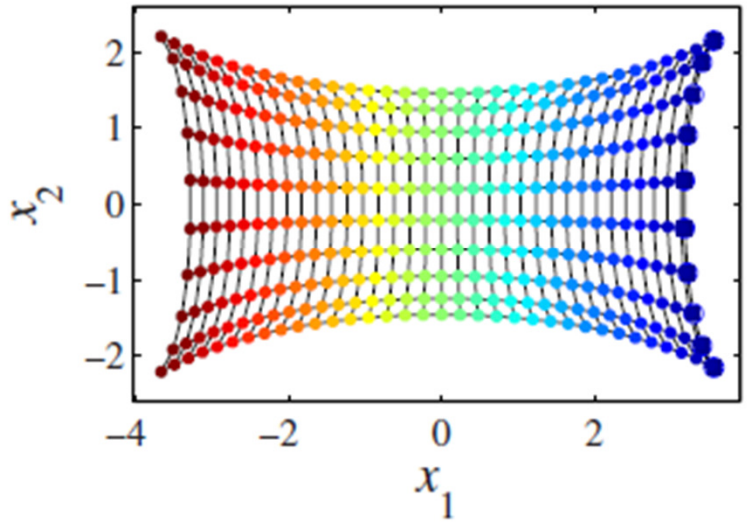
Nonlinear DR: continued

- We'll also be looking at
  - **Autoassociative neural networks** and
  - **Autoencoders**
    - use a new technique for training autoassociative neural nets
- and an overview of some other NLDR algorithms
- so hang on to your seats!

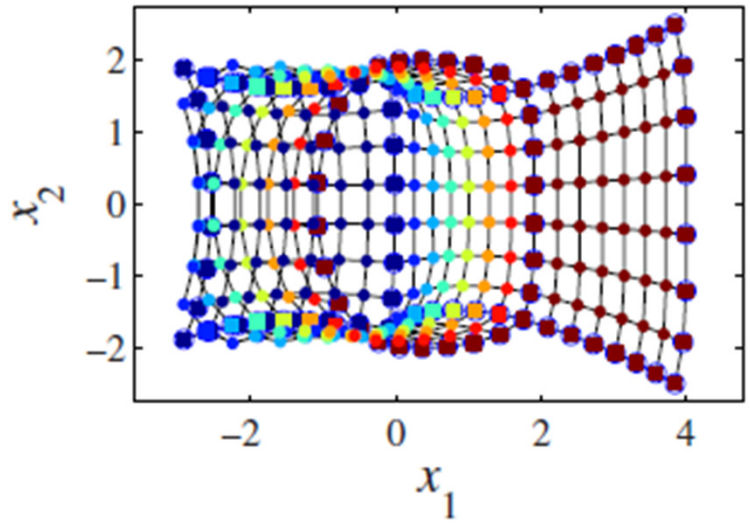
Isomap (J. Tenenbaum et al., 2000)

1) Build graph **G** with **K**-rule
2) Weigh each edge by its Euclidean length **(weighted graph)**
3) Perform **Dijkstra**'s algorithm, store square of pairwise distances in Δ
4) Perform **MDS** on Δ

Isomap: results



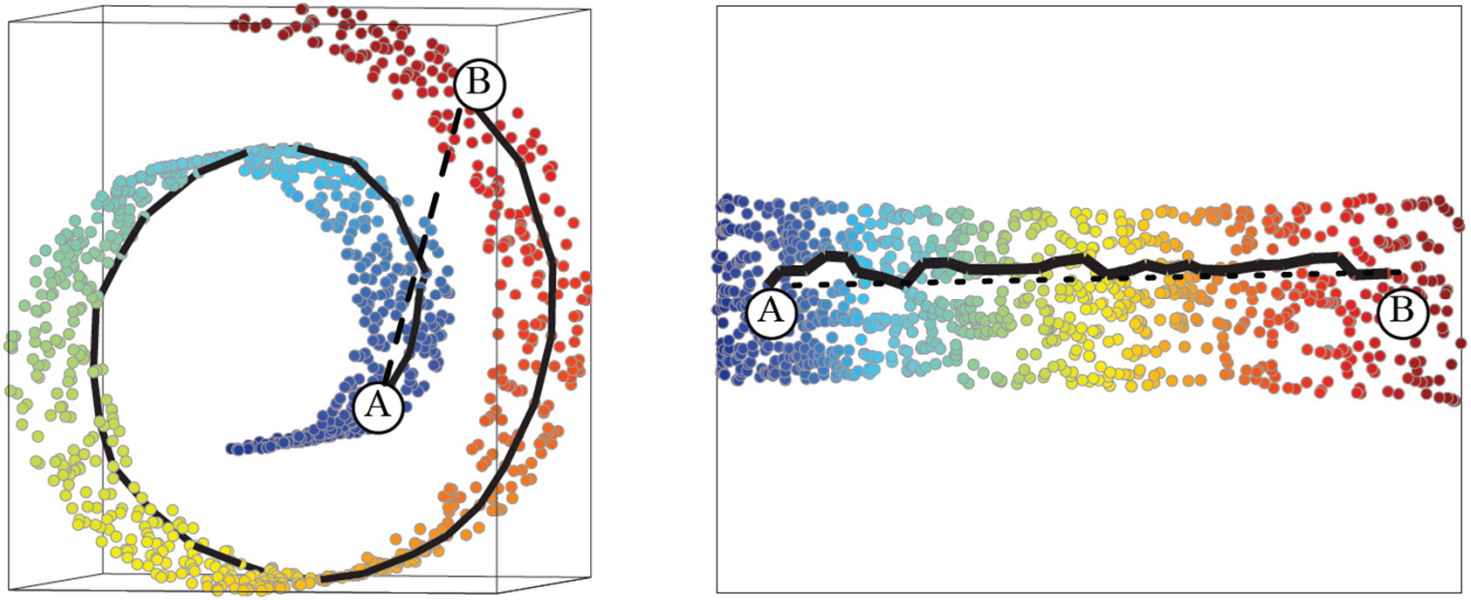
A.



N.A.

Isomap: discussion

- A variant of MDS
  - estimates of **geodesic distances** are substituted for Euclidean distances



(L. K. Saul et al., 2005)

Isomap: discussion

- A variant of MDS
  - Nonlinear capabilities brought by graph distances and **not by** inherent nonlinear models of data
- Computation time dominated by calculation of shortest paths
- Guaranteed convergence for **developable manifolds** only
  - Pairwise geodesic distances computed between points of the P-manifold, can be mapped to pairwise Euclidean distances measured in a P-dimensional Euclidean space

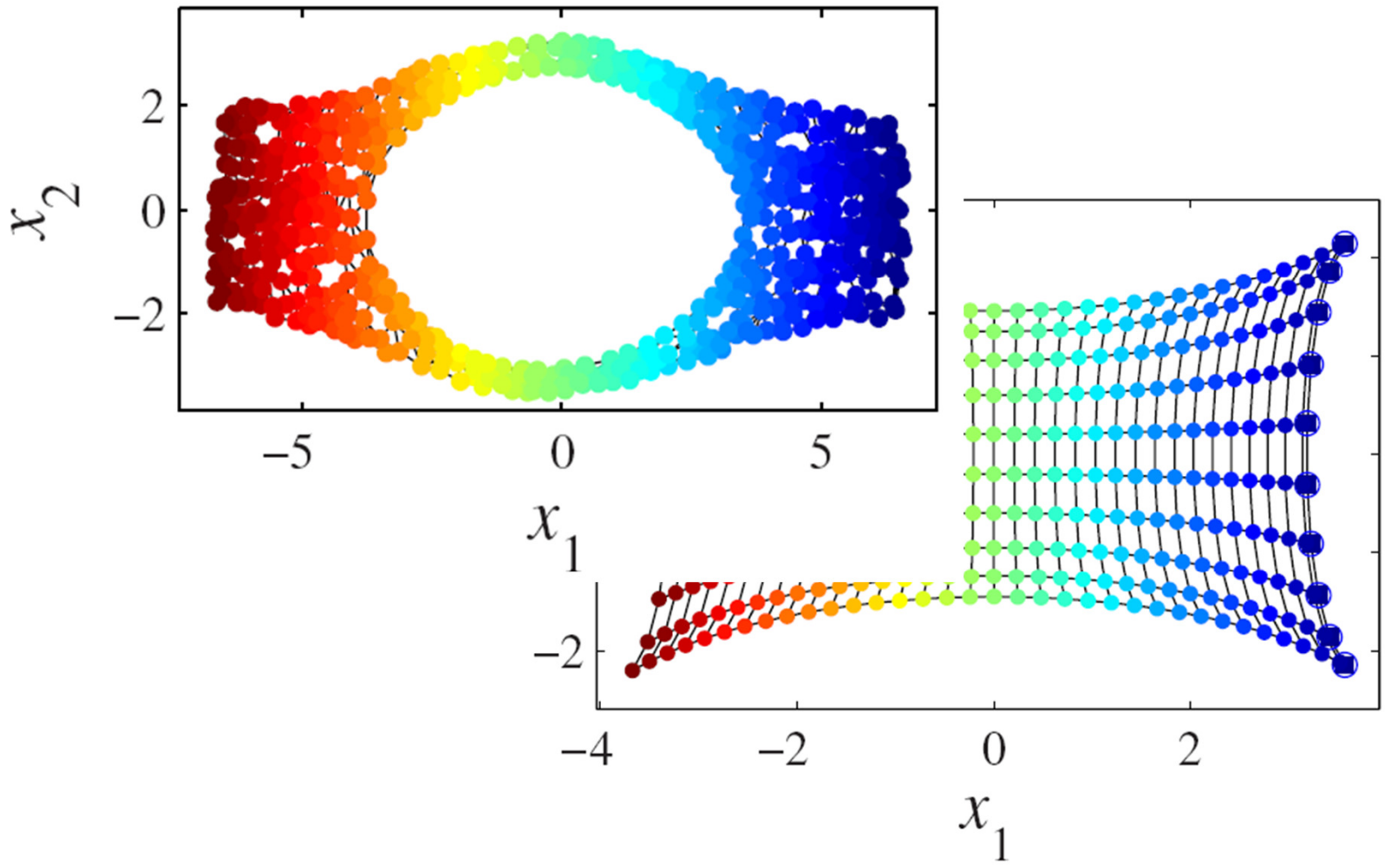
Isomap: discussion

- Dijkstra's algorithm solves the **single-source shortest path** problem
- So we need to run Dijkstra for each vertex
- More efficient than Floyd-Warshall because graph is sparse

Isomap: discussion

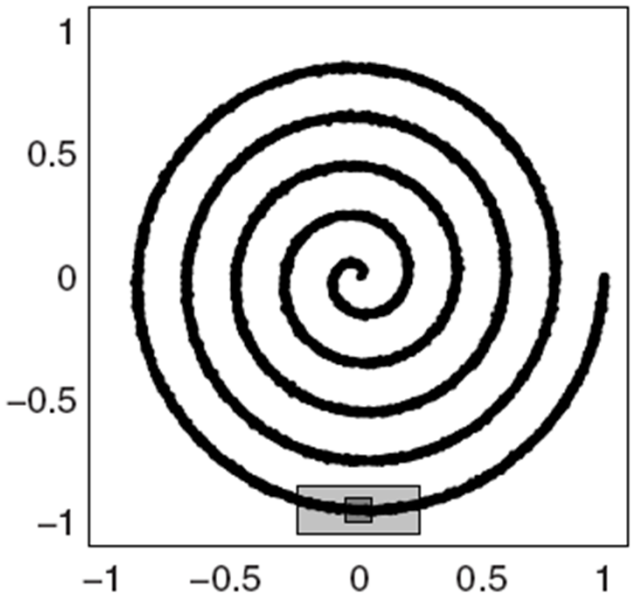
- Results of Isomap strongly depend on the quality of the estimation of geodesic distances
- If **data set is sparse**, (and no shortcuts take place)
  - graph distances are likely to be overestimations
- If **data manifold contains holes**, paths need to go around holes
  - graph distances are overestimations

Isomap: discussion



Isomap: estimation of intrinsic dimension

- A single run of **PCA**, **MDS**, or **Isomap**
- Gap in eigenvalues



-PCA, MDS: **2**

-Isomap: **1**

LLE (S. Roweis and L. Saul, 2000)

1) Build graph **G** with **K**-rule
2) Find the weight matrix **W** for reconstructing each point from its **K** neighbors
3) Find the low-dimensional coordinates **X**, that are reconstructed from weights **W** with minimum error

LLE: step 2)

2) Find the weight matrix **W** for reconstructing each point from its **K** neighbours

$$\mathcal{E}(\mathbf{W}) = \sum_{i=1}^{N} \left\| \mathbf{y}(i) - \sum_{j \in \mathcal{N}(i)} w_{i,j} \mathbf{y}(j) \right\|^2$$

LLE: step 3)

3) Find the low-dimensional coordinates **X**, that are reconstructed from weights **W** with minimum error

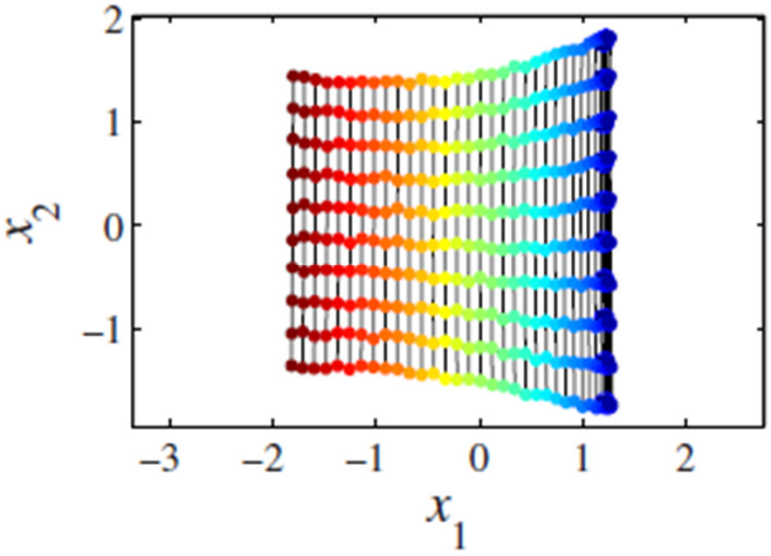
$$\Phi(\hat{\mathbf{X}}) = \sum_{i=1}^{N} \left\| \hat{\mathbf{x}}(i) - \sum_{j \in \mathcal{N}(i)} w_{i,j} \hat{\mathbf{x}}(j) \right\|^2$$

LLE step 3): discussion

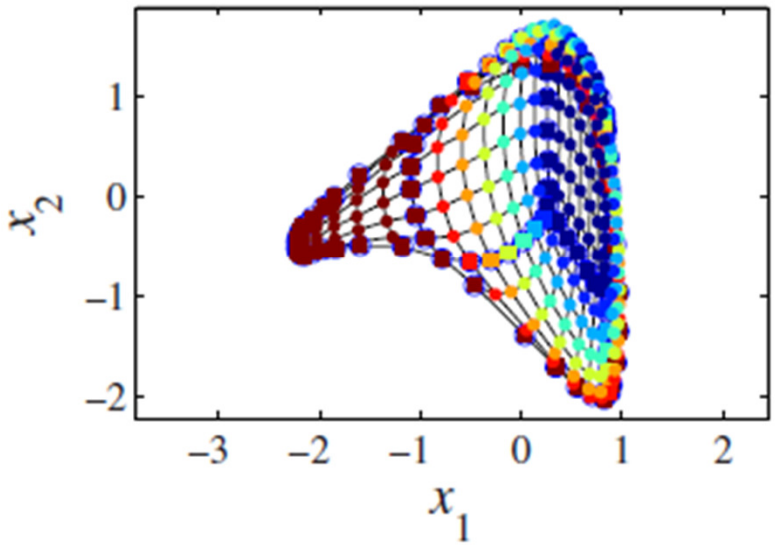
- Optimal embedding is found by computing the bottom **P+1** eigenvectors of **M**

$$\mathbf{M} = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W})$$

LLE: results



A.



A.

LLE: discussion

- Like MDS, LLE uses **EVD**, which is purely linear
  - Nonlinear capabilities of LLE come from the computation of nearest neighbors (**thresholding**)
- Unlike MDS, cannot estimate **intrinsic dimensionality** (no telltale gap in **M**)
- Works for non-convex manifolds, but not ones that contain holes
- Very sensitive to its parameter values

Discussion: "local manifold learning" (Y. Bengio and M. Monperrus, 2005)

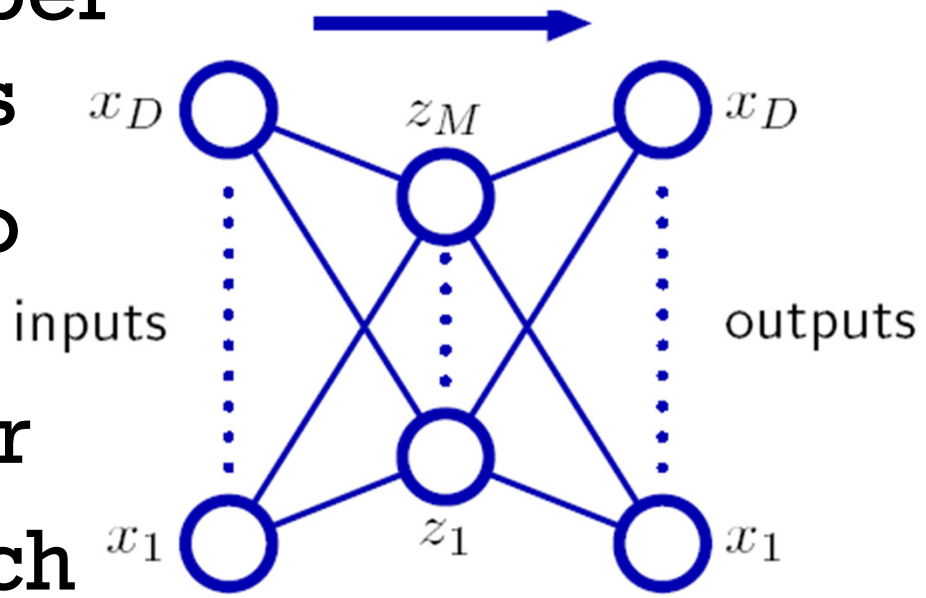
- LLE, Isomap are **local learning** methods
- They could fail when
  - Noise around manifold
  - High curvature of the manifold
  - High intrinsic dimension of the manifold
  - Presence of multiple manifolds with little data per manifold

Autoassociative neural nets
(M. A. Kramer, 1991)

- Nonlinear capabilities of Isomap and LLE were not brought by inherent **nonlinear models of data**
- Also, both methods use **'local'** generalization
- Apart from **supervised** learning for **classification**, neural nets have been used in the context of **unsupervised** learning for **dimensionality reduction**

Autoassociative NN: continued

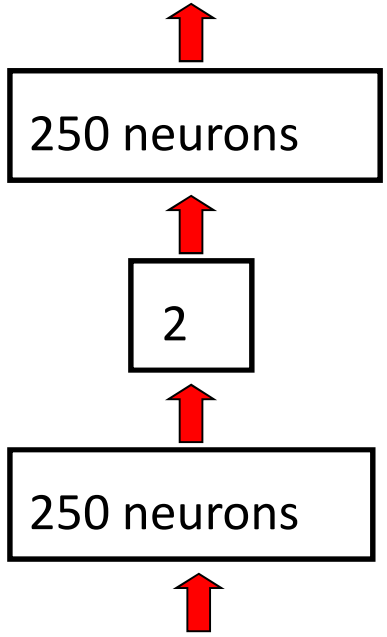
- DR achieved by using net with same number of input and outputs
- Optimize weights to minimize reconstruction error
- Net tries to map each input vector onto itself



(C. M. Bishop, 2006)

Autoassociative NN: the intuition

- Net is trained to reproduce its input at the output
- So it packs as much information as possible into the central bottleneck



Autoassociative NN: continued

- Number of hidden units is smaller than number of inputs
  - there exists a **reconstruction error**
- Determine network weights by minimizing the reconstruction sum-of-squares error:

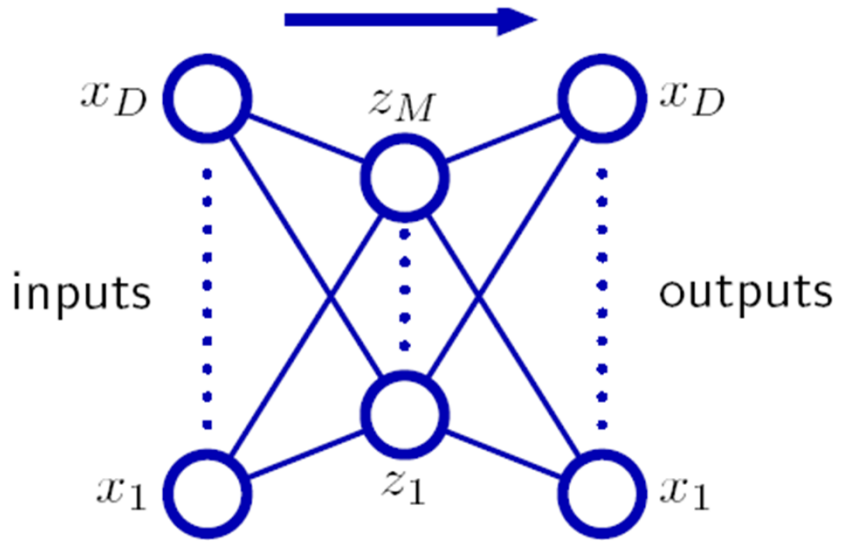
$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N} \|\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{x}_n\|^2$$

Autoassociative NN and PCA

- Here's an interesting fact:
- If hidden units have linear activation functions,
- It can be shown that error function has a **unique global minimum**
- At this minimum, the network performs a projection onto an **M**-dimensional subspace
  - spanned by the **first M PCs** of the data!

Autoassociative NN and PCA: continued

- Vector of weights leading into zi's from a basis set which spans the principal subspace
- These vectors need not be orthonormal

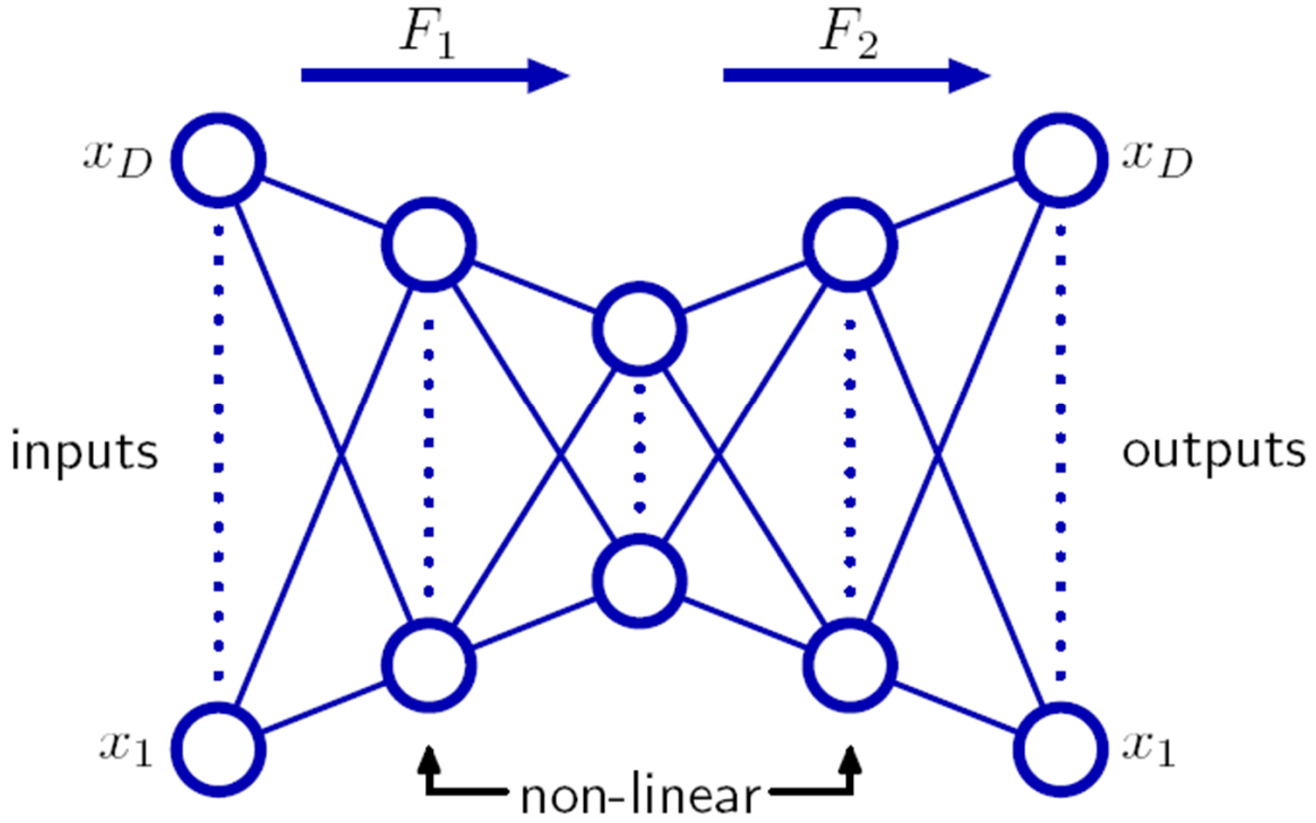


Autoassociative NN and PCA: continued

- Even with nonlinear activation functions for the hidden units,
  - the min error solution is again the projection onto the PC subspace
  - so there is no advantage in using 2-layer NNs to perform DR
  - standard PCA techniques based on SVD are better

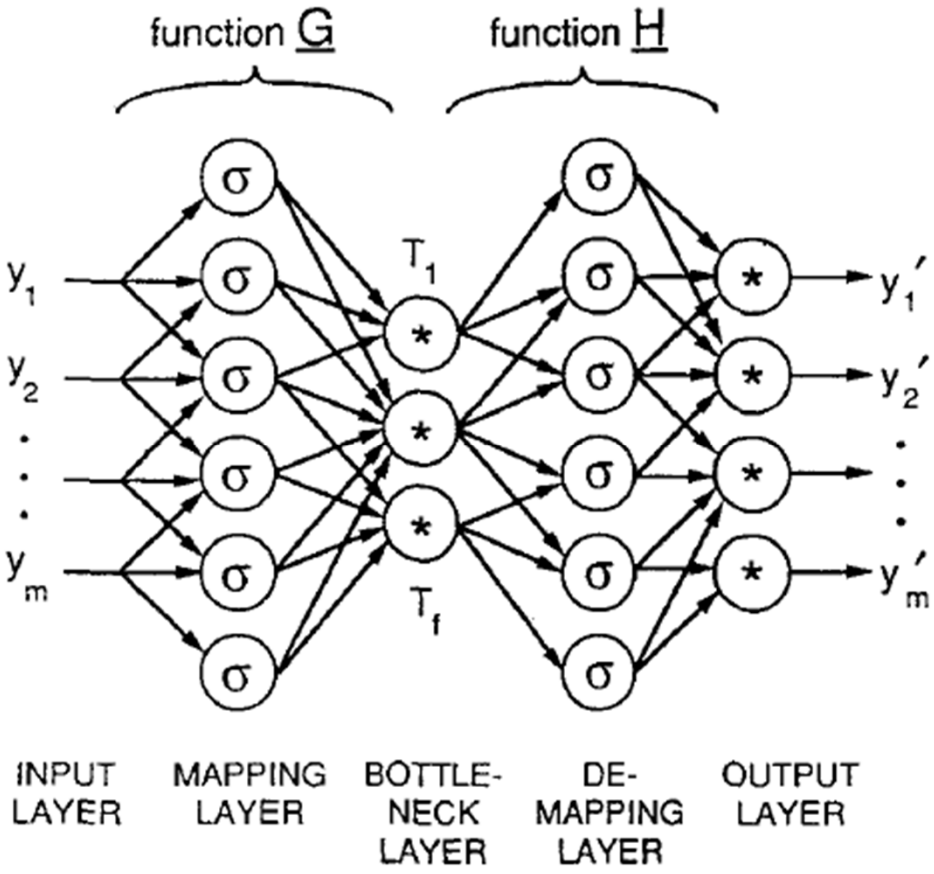
Autoassociative NN: nonlinear PCA

- What we need is additional hidden layers -- consider the 4-layer net below



Autoassociative NN: NLPCA

- Training to learn the identity mapping is called
  - **self-supervised backpropagation** or
  - **Autoassociation**
- After training, the combined net has no utility
  - And is divided into two single-hidden layer nets **G** and **H**



NLPCA: discussion

- Start with random weights, the two nets (**G** and **H**) can be trained together by minimizing the discrepancy between the original data and its reconstruction
- Error function as before (sum-of-squares)
  - but no longer a quadratic function of net params.
  - risk of falling into local minima of err. func.
  - and burdensome computations
- Dimension of subspace must be specified before training

Autoencoder
(G. E. Hinton and R. R. Salakhutdinov, 2006)

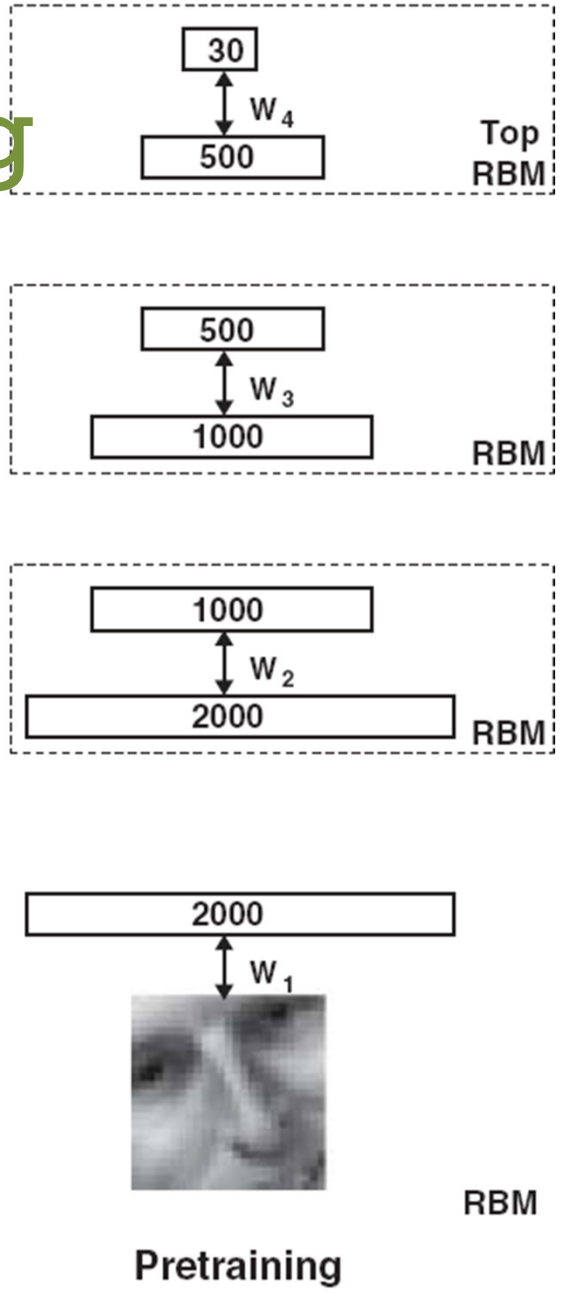
- It was known since the 1980s that **backpropagation** through **deep neural nets** would be very effective for **nonlinear dimensionality reduction** -- subject to:
  - fast computers … OK
  - big data sets … OK
  - **good initial weights** …

Autoencoder: continued

- BP = backpropagation (CG methods, steepest descent, …)
- Fundamental problems in training **nets with many hidden layers** (**"deep" nets**) with BP
  - learning is slow, results are poor
- But, results can be improved significantly if **initial weights** are close to solution

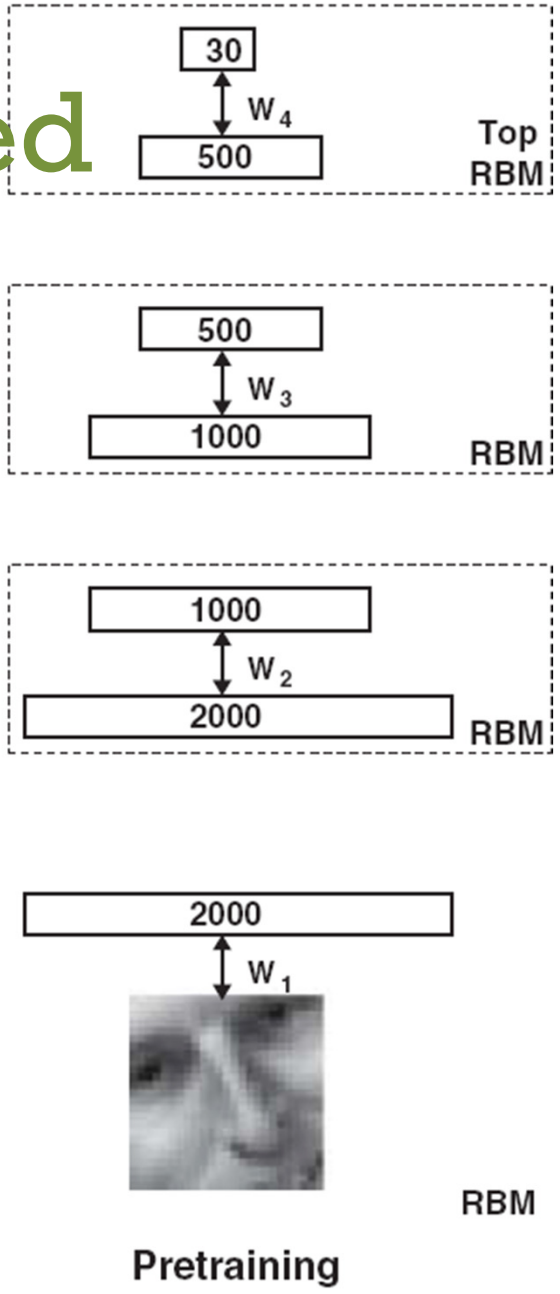
Autoencoder: pretraining

- Treating each neighboring set of layers like an RBM
  - to approximate a good solution
- **RBM = Restricted Boltzmann Machine**
  - will be the topic of an upcoming talk



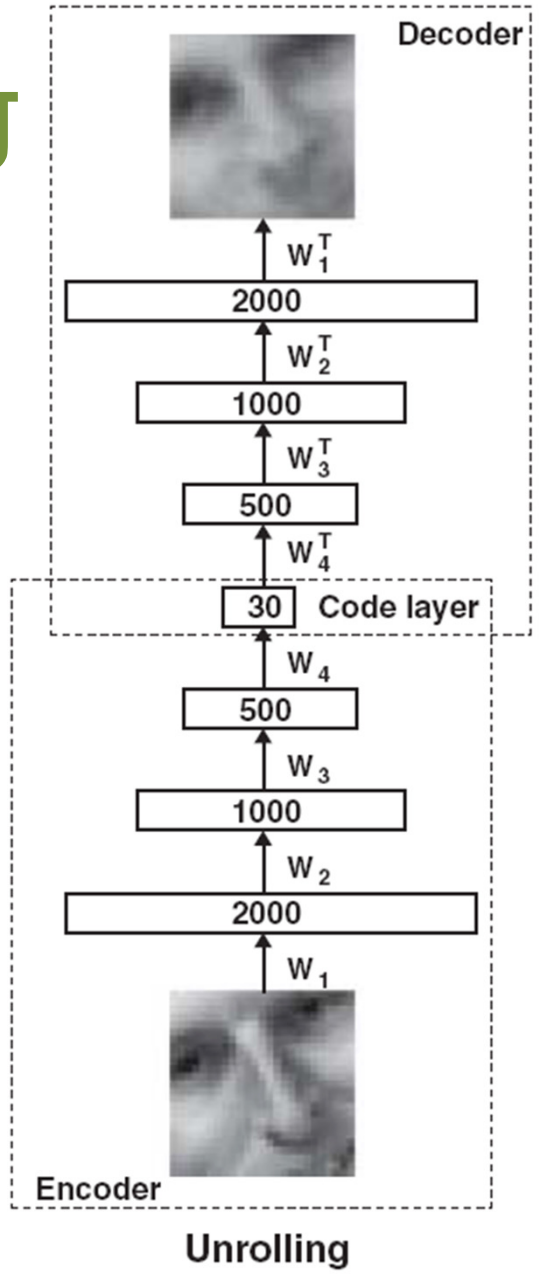
Autoencoder: continued

- The learned features of one RBM are used as data for training the next RBM in the stack
- The learning is unsupervised.



Autoencoder: unrolling

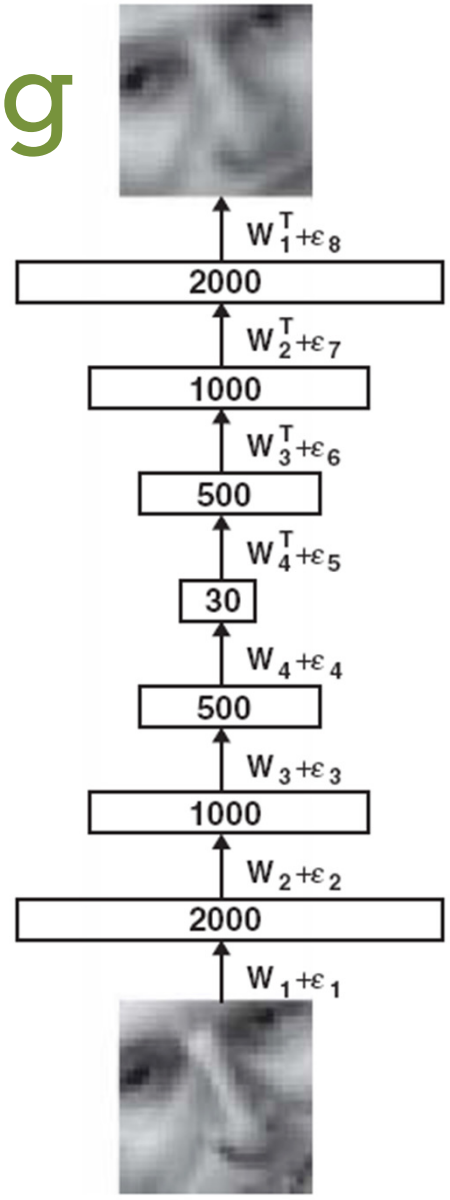
- After pretraining, the model is unfolded
- Produces encoder and decoder networks that use the same weights
- Now, we'll go on to the global fine-tuning stage



Unrolling

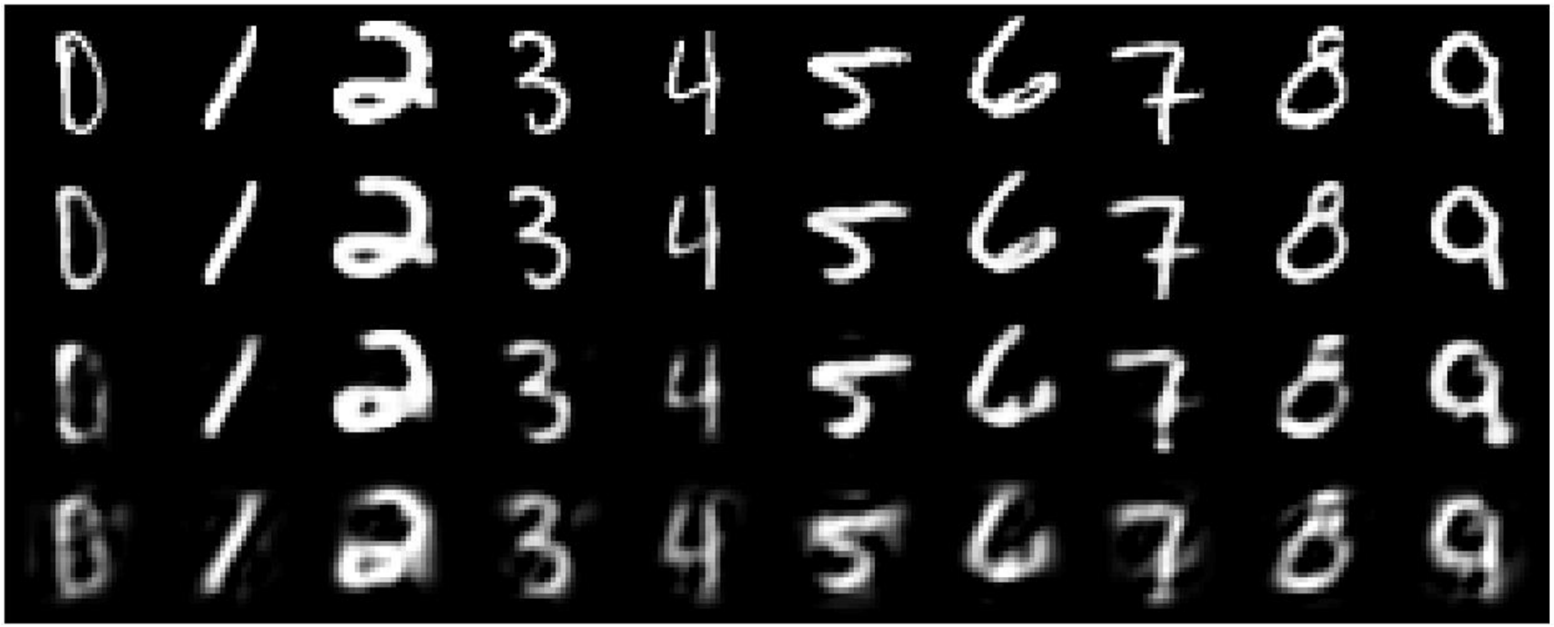
Autoencoder: fine-tuning

- Now use BP of error derivatives to fine-tune ☺
- So we don't run BP until we have good initial weights
- With good initial weights, BP need only perform local search



Fine-tuning

Autoencoder: results



real data

30-D deep auto

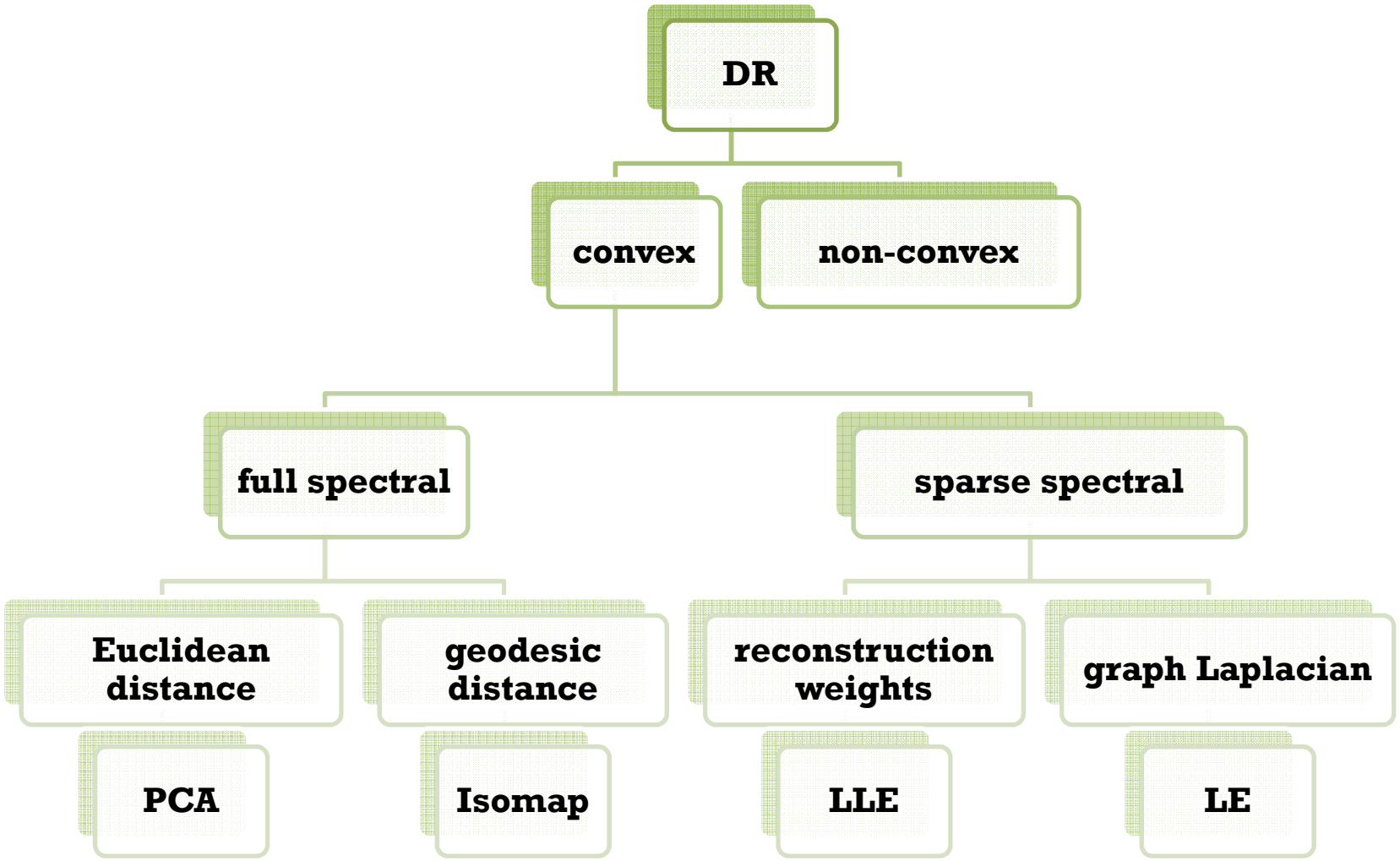
30-D logistic PCA

30-D PCA

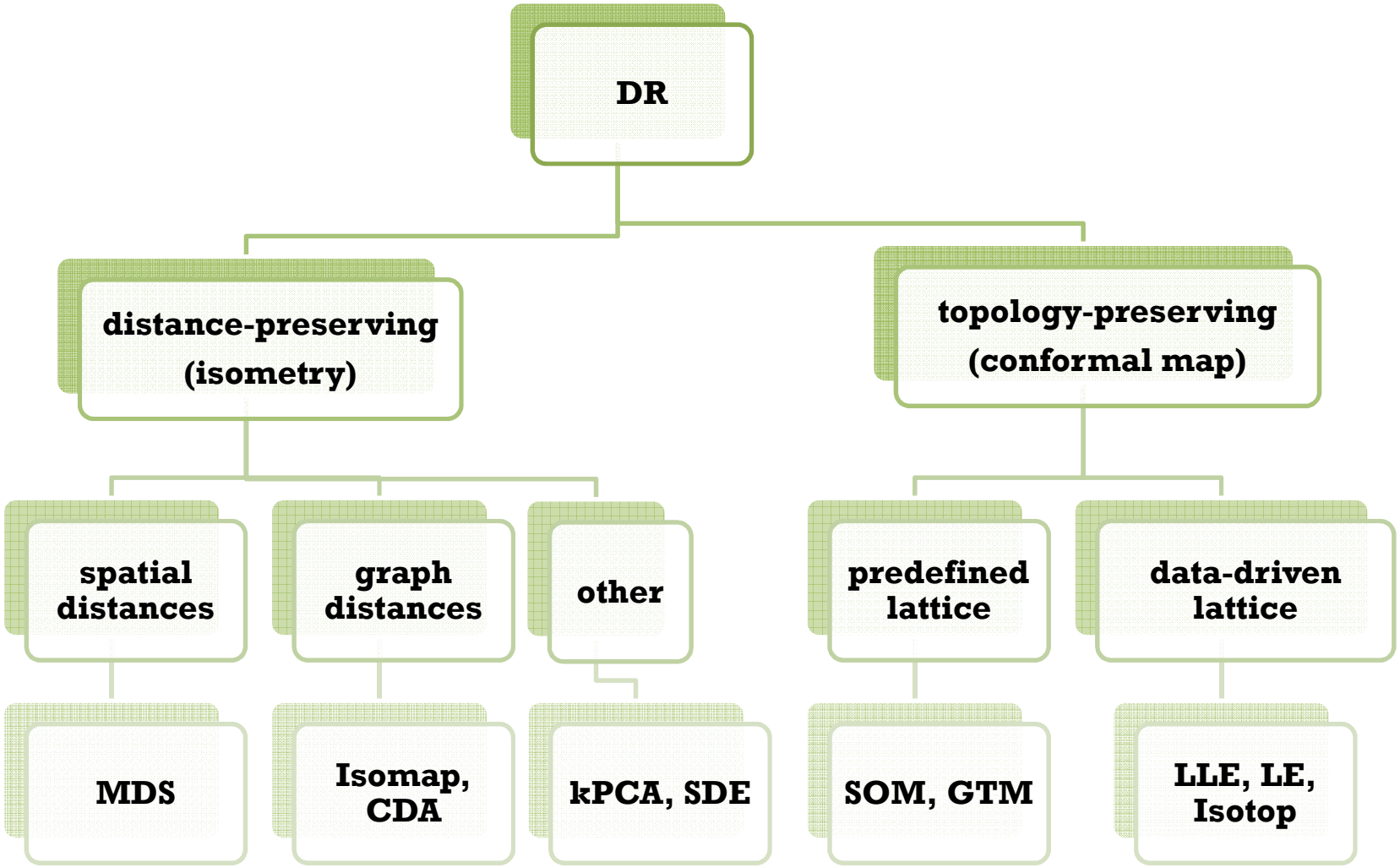
DR: taxonomy

- Here we considered
  - **linear** vs. **nonlinear** (model of data)
- There are many other possible categorizations, to name a few:
  - **local** vs. **non-local** (generalization)
  - **single** vs. **multiple** (coordinate system)
  - **unsupervised** vs. **supervised**
  - **data-aware** vs. **data-oblivious**
  - **exact** vs. **approximate** (optimization)

DR: taxonomy (L. van der Maaten, 2009)



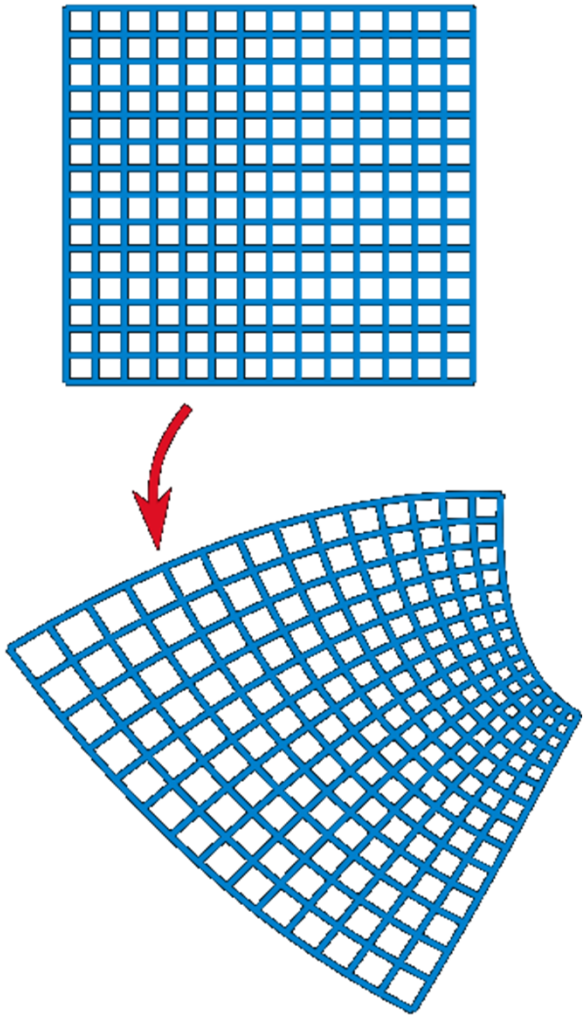
DR: taxonomy (J. A. Lee, and M. Verleysen, 2007)



Note: conformal map (Wikipedia)

A function that preserves angles.

Pairs of lines intersecting at 90° to pairs of curves intersecting at 90°.



Discussion: out-of-sample generalization (Y. Bengio et al., 2003)

- The model of PCA is continuous
  - An implicit mapping is defined:
    - $\mathbf{X} = \mathbf{W}^T\mathbf{Y}$
  - → generalization to new points is easy
- But, MDS, Isomap and LLE provide an explicit mapping
  - $(\mathbf{x}_n, \mathbf{y}_n)$

Discussion: dataset size

- Large datasets: **N>2000**
  - Time and space complexity of NLDR methods at least $O(N^2)$
  - Need to resample available data
    - using k-means for example
- Medium: **200<N≤2000**
  - OK
- Small: **N≤200**
  - Insufficient to identify parameters
  - Use PCA/MDS

Discussion: dataset dimensionality

- Very high: **D>50**
  - NLDR fails b/c of COD
  - First apply PCA/MDS/RP for hard DR
    - can provide robustness to noise
- High: **5<D≤50**
  - COD still exists, use at your own risk
- Low: **D≤5**
  - Apply with confidence

Discussion: dataset intrinsic dimensionality

- Target dim >> intrinsic dim
  - PCA/MDS/RP perform well
- Target dim ≥ intrinsic dim
  - NLDR provides good results
- Target dim < intrinsic dim
  - Use NLDR at your own risk
    - results are meaningless b/c forced
  - Nonspectral methods don't converge
    - spectral methods solve an eigenproblem irrespective of target dimensionality

Discussion: goal of DR

- DR is a **preprocessing** step
  - and some information is lost
- You want to preserve what is important for the next step
  - whether it's classification or clustering
- The **method** and **metric** you use should be in line with the next task

One final note

- Motivation behind DR was to remove COD
- But the mentioned NLDR methods fall prey to COD themselves
  - when intrinsic dimensionality is higher than **4** or **5**

Looking ahead: future sessions

- We'll be talking about
  - **kernel methods**
  - **SVM**
    - (sparse kernel machines)
  - **statistical learning theory**
    - (PAC learning and VC dimension)
- And after that, we'll talk about
  - **deep learning methods**
    - **as a feature extraction method that allows us to deal with the curse of dimensionality**
- We'll try to put it all in the context of information retrieval
  - specifically **multimedia information retrieval**
  - e.g. CBIR, MIR

References

Richard E. Bellman, Adaptive control processes - a guided tour, Princeton University Press, Princeton, New Jersey, U.S.A., 1961.

T. F. Cox and M. A. A. Cox, Multidimensional scaling, Chapman and Hall, 1994.

Luis Jimenez and David Landgrebe, Supervised classication in high dimensional space: Geometrical, statistical, and asymptotical properties of multivariate data, IEEE Transactions on System, Man, and Cybernetics 28 (1998), 39-54.

I. T. Jolliffe, Principal component analysis, Springer series in statistics, Springer, New York, 1986.

Verleysen Michel Lee, John A., Nonlinear Dimensionality Reduction, Information Science and Statistics, Springer, 2007.

Sam T. Roweis and Lawrence K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000), 2323-2326.

J. B. Tenenbaum, V. de Silva, and J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000), 2319-2323.

Thank you for your attention.