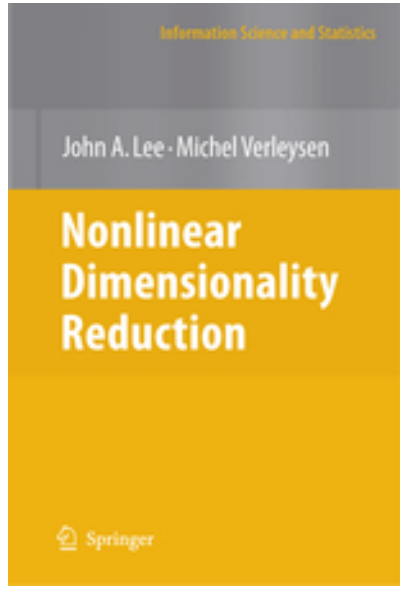
Data visualization and dimensionality reduction

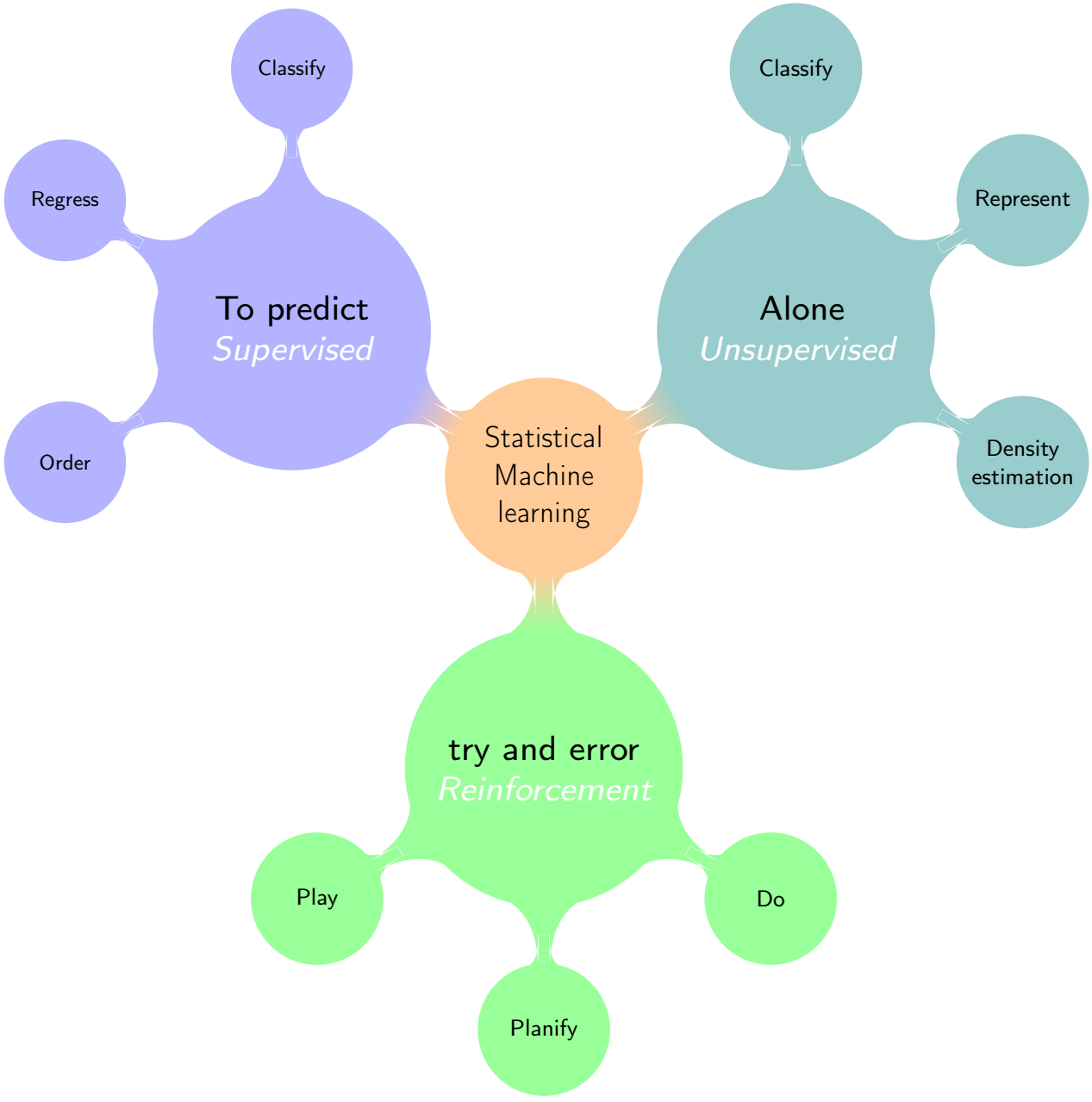
Stéphane Canu
asi.insa-rouen.fr/enseignants/~scanu
scanu@insa-rouen.fr



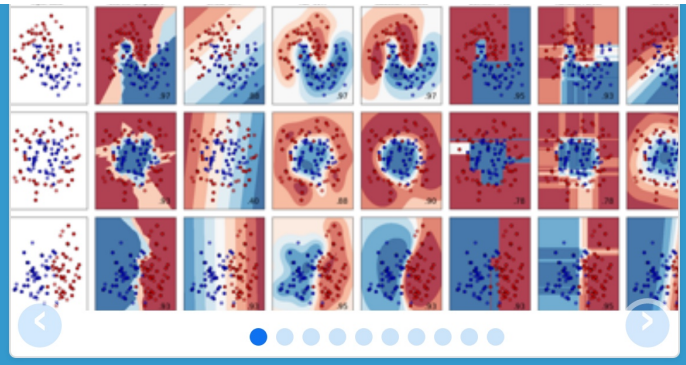
Workshop on Machine Learning

February 15, 2019

The 3 main different kinds of machine learning



Scikit-learn



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, … — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, … — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, … — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

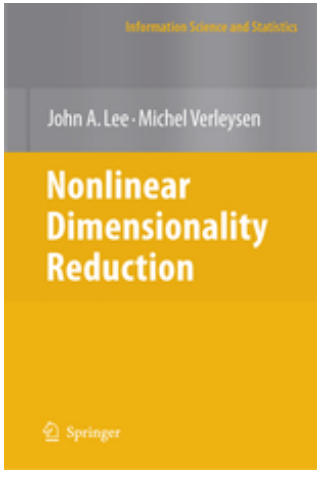
Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

https://scikit-learn.org/

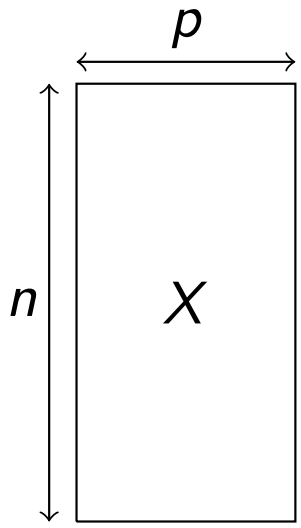
Lecture road map

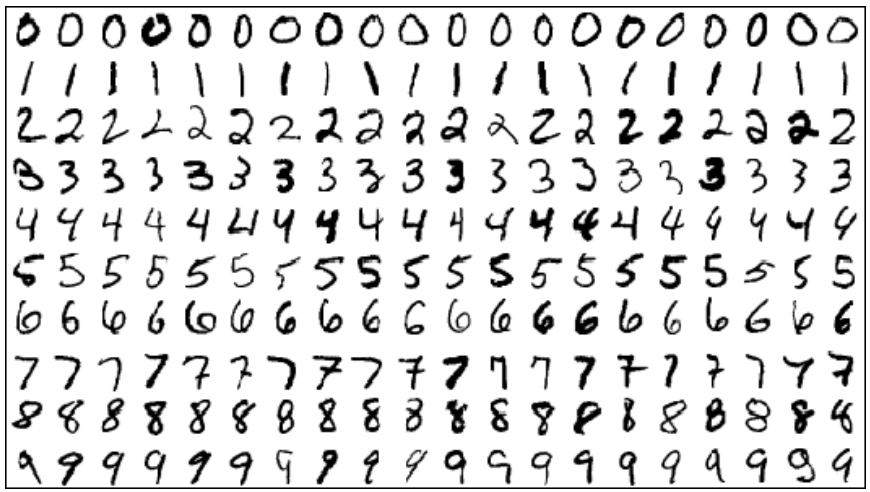
1. Introduction to hidden variables
2. PCA: principal component analysis
3. Distance preservation approaches (global methods)
4. Local approaches



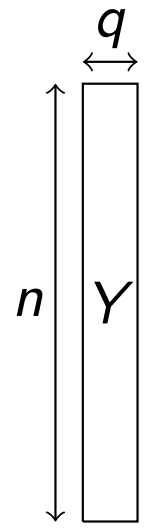
A dimensionality reduction problem

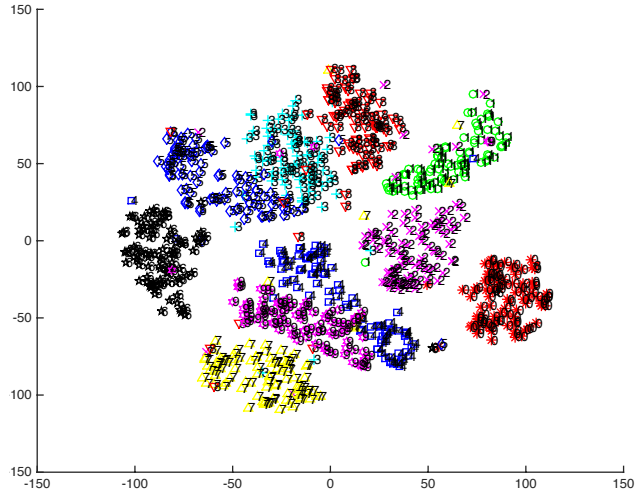
Given X a n by p data matrix , find a Y $n \times q$ matrix with $q < p$





$p = 784$





$q = 2$

Crédit : CSC 2535: 2013 Lecture 11 Non-linear dimensionality reduction Geoffrey Hinton
Nonlinear dimensionality reduction JA Lee, M Verleysen, Springer, 2007

Dimensionality reduction: what for ?

- Visualize ($q = 2$ ou 3)
  - validate data coding
  - detect outliers and liss labeled data
  - visualize classes

- Represent ($q < p$)
  - summarize (remove noise)
  - preprocessing: brings statistic and computation efficiency
  - the hidden variable hypothesis

Coding/decoding functions

$$\begin{array}{rccccccc} cod: & \mathbb{R}^p & \longrightarrow & \mathbb{R}^q & , & \mathbf{x} & \longmapsto & \mathbf{y} = cod(\mathbf{x}) \\ dec: & \mathbb{R}^q & \longrightarrow & \mathbb{R}^p & , & \mathbf{y} & \longmapsto & \mathbf{x} = dec(\mathbf{y}) \end{array}$$

This problem is ill posed: what is the criteria to be optimized?

The curse of dimensionality

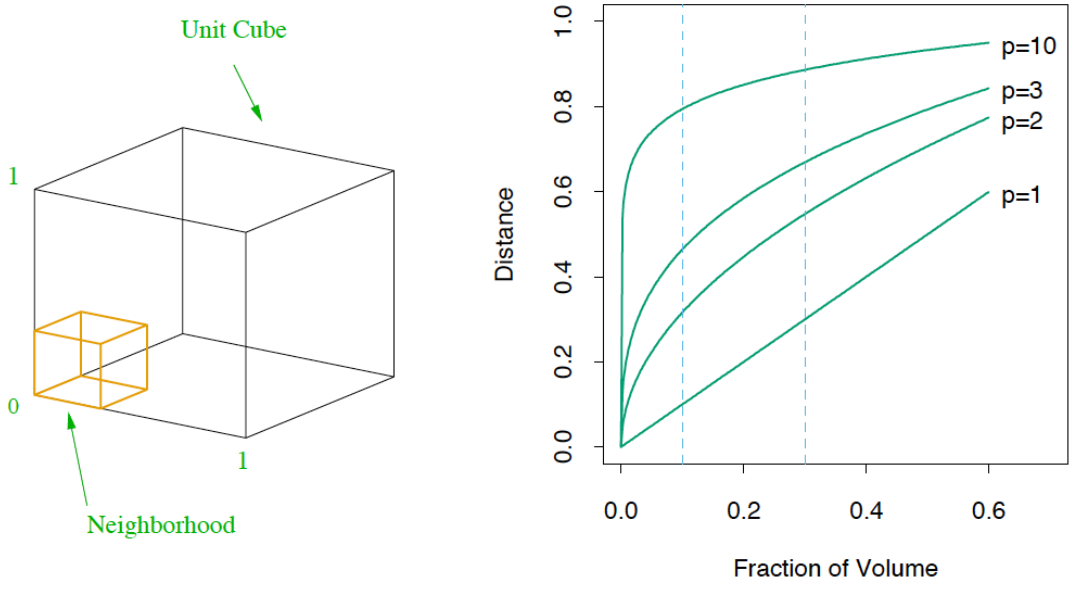
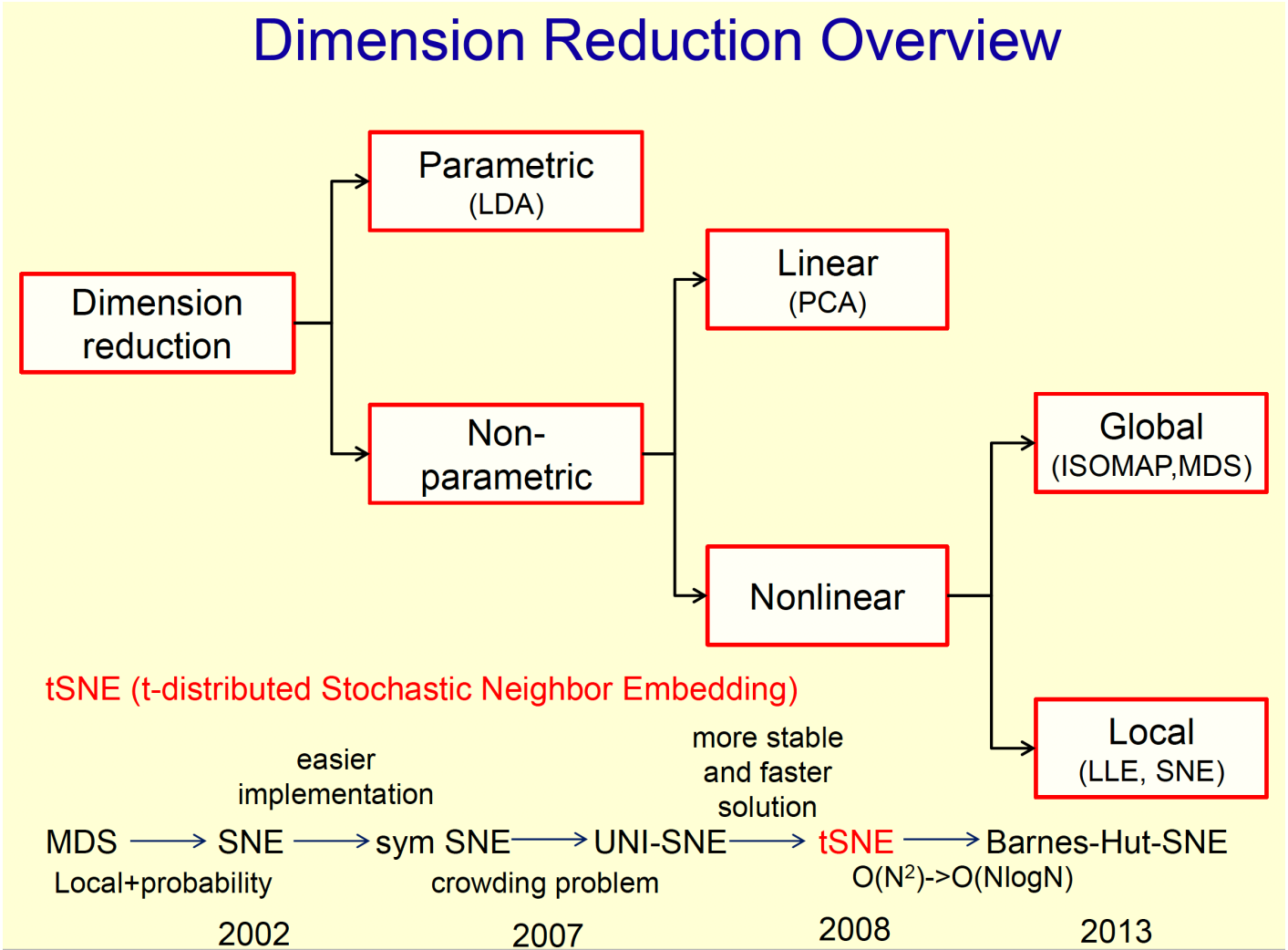


FIGURE 2.6. *The curse of dimensionality is well illustrated by a subcubical neighborhood for uniform data in a unit cube. The figure on the right shows the side-length of the subcube needed to capture a fraction r of the volume of the data, for different dimensions p. In ten dimensions we need to cover 80% of the range of each coordinate to capture 10% of the data.*

In large dimension, intuitions on distances in low dimension (2 or 3) no longer apply.

image from the Elements of Statistical Learning, chapter 2 (pp 23):

Dimensionality reduction: the big picture

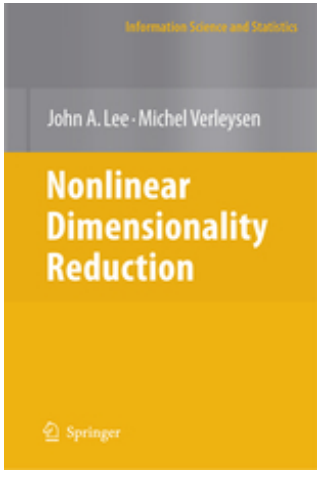


and then UMAP, 2018

Nonlinear idmensionality reduction = manifold learning

Lecture road map

1. Introduction to hidden variables
2. **PCA: principal component analysis**
3. Distance preservation approaches (global methods)
4. Local approaches



PCA: principal component analysis

Model: data = information + noise $\qquad X = YV^\top + B$

Linear coding: find V orthogonal such that

$$\begin{array}{rccccccc} cod: & \mathbb{R}^p & \longrightarrow & \mathbb{R}^q & , & X & \longmapsto & Y = XV \\ dec: & \mathbb{R}^q & \longrightarrow & \mathbb{R}^p & , & Y & \longmapsto & \widehat{X} = YV^\top \end{array}$$

Objective: min. the reconstruction error between X et $dec(cod(X))$

$$\min_{Y \in \mathbb{R}^{n \times q}, V} \|X - YV^\top\|_F^2$$

or maximize the variance of the projection

$$\max_{v \in \mathbb{R}^p} \|Xv\|^2 \qquad \text{with } \|v\|^2 = 1 \text{ and } y = Xv$$

or minimize the reconstruction error of the covariance (Gram) matrix

$$\min_{y \in \mathbb{R}^n} \|XX^\top - \mathbf{y}\mathbf{y}^\top\|^2$$

PCA computation

Theorem (Eckart & Young, 1936)

The unique solution of

$$\min_{\mathbf{u},\mathbf{v}} J(\mathbf{u},\mathbf{v}) \qquad \text{with} \quad J(\mathbf{u},\mathbf{v}) = \|X - \mathbf{u}\mathbf{v}^\top\|_F^2$$

with $\|\mathbf{v}^\star\| = 1$, is given by: $\mathbf{v}^\star$ and $\mathbf{u}^\star = X\,\mathbf{v}^\star$, where $\mathbf{v}^\star$ is the normalized eigen vector associated with λ the largest eigen value of $X^\top X$. Furthermore, we have: $\|\mathbf{u}^\star\| = \sqrt{\lambda}$.

proof

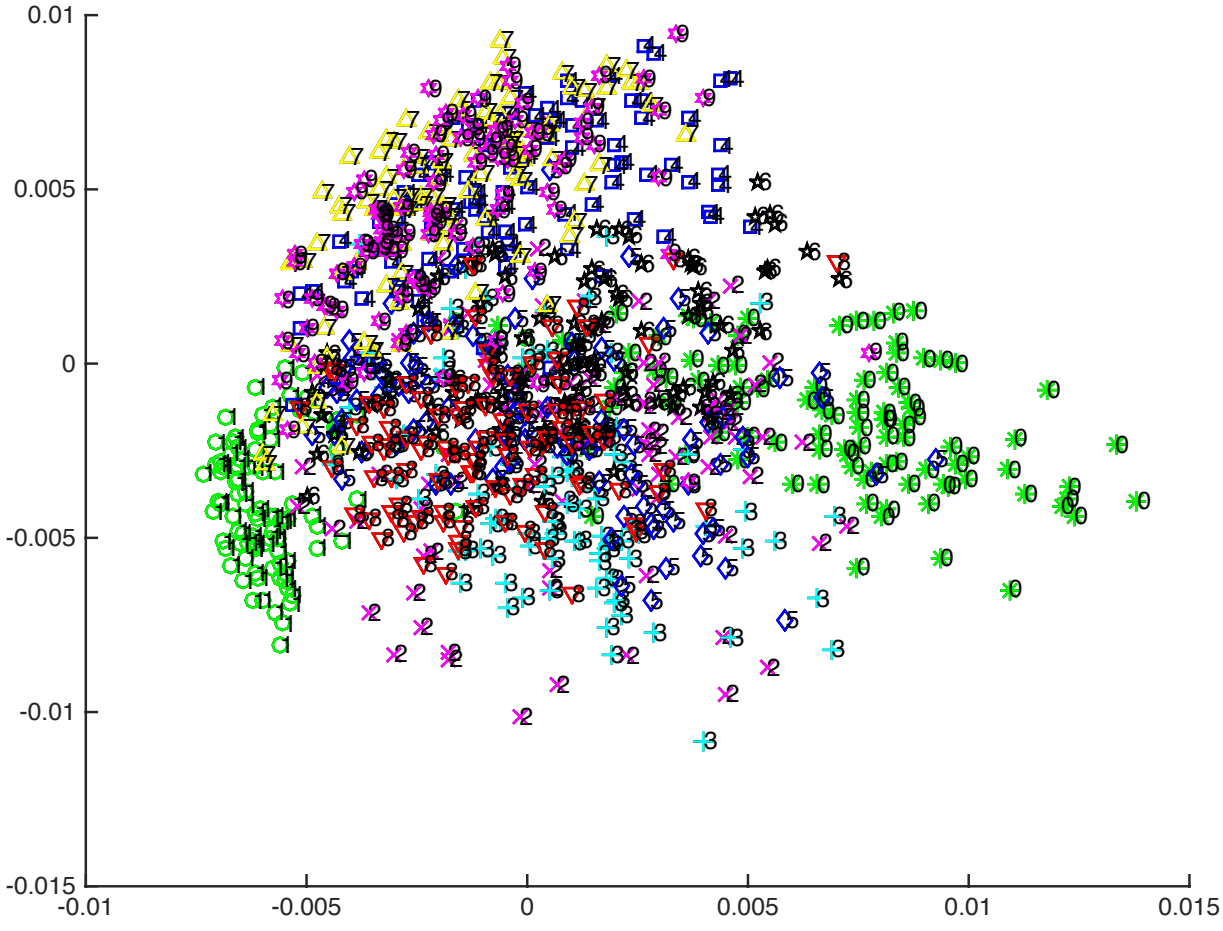
$$\begin{cases} \nabla_{\mathbf{u}} J(\mathbf{u},\mathbf{v}) = \quad -2X\mathbf{v} + 2\|\mathbf{v}\|^2\mathbf{u} = 0 \\ \nabla_{\mathbf{v}} J(\mathbf{u},\mathbf{v}) = -2X^\top\mathbf{u} + 2\|\mathbf{u}\|^2\mathbf{v} = 0 \end{cases}$$

3 different ways to get Y

$$svd(X),\ eig(X^\top X),\ eig(XX^\top)$$

2d PCA on the MNIST data

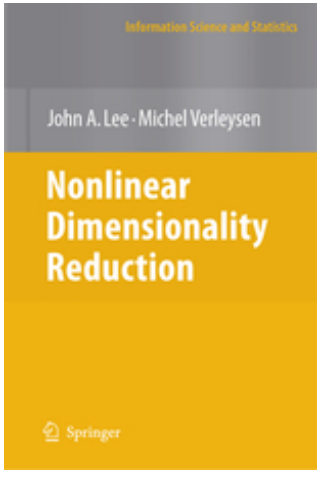
$p = 784$ $q = 2$



$Y =$

Lecture road map

1. Introduction to hidden variables
2. PCA: principal component analysis
3. **Distance preservation approaches (global methods)**
4. Local approaches



Distances

- symmetric : $d(x, y) = d(y, x)$
- separation : $d(x, y) = 0 \Leftrightarrow x = y$
- triangular inequality: $d(x, y) \leq d(x, z) + d(z, y)$

Example: the euclidian distance

$$d(x, y) = \|x - y\| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Distance and dot product

$$d(x, y)^2 = \|x\|^2 + \|y\|^2 - 2x^\top y$$

Gram matrix

$$G = XX^\top$$

pseudometric ($d(x, y) = 0$ and $x \neq y$)) quasi distances (no symetry), ...

Dimensionality reduction: MDS

Given

$$d_X(i,j) = \|x_i - x_j\|$$

Distances conservation

$$\min_{Y \in \mathbb{R}^{n \times q}} \sum_{i=1}^{n} \sum_{j=1}^{i-1} \left(d_X(i,j)^2 - \underbrace{\|y_i - y_j\|^2}_{d_Y(i,j)^2} \right)^2$$

Related ptimization problems (many variants):

- classical MDS (Torgerson, 1958)
- Kruskal -Shepard method (Kruskal, 1964)
- Sammon projection
- MDS INDSCAL (Carrol et Chang, 1970)
- ...

Classical MDS

Let H be the $n \times n$ centering projection matrix

$$H = I - \frac{1}{n} ee^\top \quad \text{avec} \quad e = (1, 1, \ldots, 1, \ldots, 1)^\top \in \mathbb{R}^n$$

1. given the distance matrix D_X
   - Y columns are the eigen vectors of HD_XH multiplied by the square root of their corresponding eigen values

$$Y_j = \sqrt{\lambda_i} u_i, \qquad i = 1, q$$

2. if X, is known, it is the PCA of the centered data matrix
   - Y columns are singular values of $X^c = HX$ multiplied by their singular values.

$$Y_j = \mu_i u_i, \qquad i = 1, q$$

MDS and PCA

MDS = PCA

- if the data is lying on an hyperplane
  $\rightarrow$ in that case, distances are preserved

- if X is centered
  and if D_X is doubly centered

MDS and PCA

let c be the col mean vector of X,

$$c = \frac{1}{n} X^\top e \quad \text{avec} \quad e = (1, 1, \ldots, 1, \ldots, 1)^\top \in \mathbf{R}^n$$

let X^c be the centered data matrix

$$\begin{aligned} X^c &= X - ec^\top \\ &= X - \frac{1}{n} ee^\top X = HX \quad \text{with} \quad H = I - \frac{1}{n} ee^\top \end{aligned}$$

recall the distance and scalar product formula $d_X(i,j)^2 = \|x_i - x_j\|^2 = \|x_i\|^2 + \|x_j\|^2 - 2x_i^\top x_j$
let G be the Gram matrix $G = XX^\top$ and $\delta = diag(G)$ with $\delta_i = \|x_i\|^2$
we have, with D_X the distances matrix of general term $d_X(i,j)^2$)

$$D_X = \delta e^\top + e\delta^\top - 2XX^\top$$

and

$$HD_X H = -2X^c X^{c\top}$$

Classical MDS aims at minimizing

$$\min_{Y \in \mathbf{R}^{n \times q}} \|HD_X H - HD_Y H\|^2 = \|X^c X^{c\top} - Y^c Y^{c\top}\|^2$$

Y^c is the eigen matrix of $X^c X^{c\top}$ that is the result of the SVD of X^c

Weighted MDS: Sammon's projection

Reinfore closed neigbors (. . . and penalized distant ones)

$$\min_{Y \in \mathbb{R}^{n \times q}} \sum_{i=1}^{n} \sum_{j=1}^{i-1} w_{i,j} \left(d_X(i,j) - \underbrace{\|y_i - y_j\|}_{d_Y(i,j)} \right)^2$$

$$w_{i,j} = \frac{1}{\|x_i - x_j\|}$$

Optimization via an iterative descent algorithm (slow) Quasi Newton (L-BFGS).

https://www.codeproject.com/Articles/43123/Sammon-Projection

Sammon, John W. Jr., "A Nonlinear Mapping for Data Structure Analysis", IEEE T. on Computers, vol. C-18, no. 5, 1969

Sammon's projection Quasi Newton

$$J_S(Y) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j}\left(d_X(i,j) - d_Y(i,j)\right)^2 \quad \text{with} \quad d_Y(i,j) = \|\mathbf{y}_i - \mathbf{y}_j\|$$

$$\begin{aligned}
\nabla_{\mathbf{y}_i} J_S(Y) &= -2\sum_{j=1, j\neq i}^{n} w_{i,j}\left(d_X(i,j) - d_Y(i,j)\right)\nabla_{\mathbf{y}_i} d_Y(i,j) \\
&= -2\sum_{j=1, j\neq i}^{n} \frac{d_X(i,j) - d_Y(i,j)}{d_X(i,j)d_Y(i,j)}(\mathbf{y}_i - \mathbf{y}_j)
\end{aligned}$$

2 issues with Sammon's projection

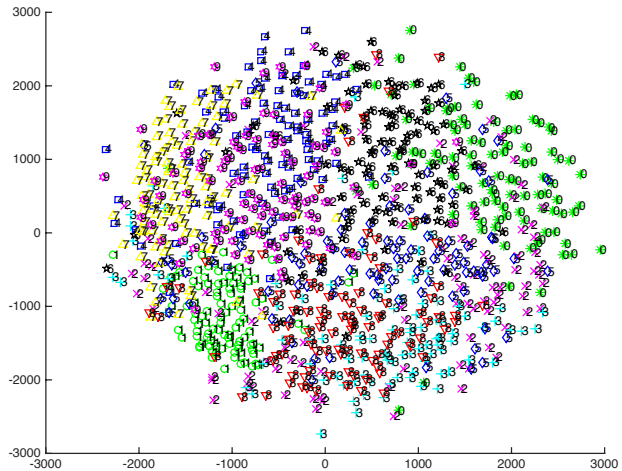
- $d = 0 \rightarrow w = \infty$ because

$$w_{i,j} = \frac{1}{\|x_i - x_j\|}$$

the criterion preserve all small distances.
Alternatives:

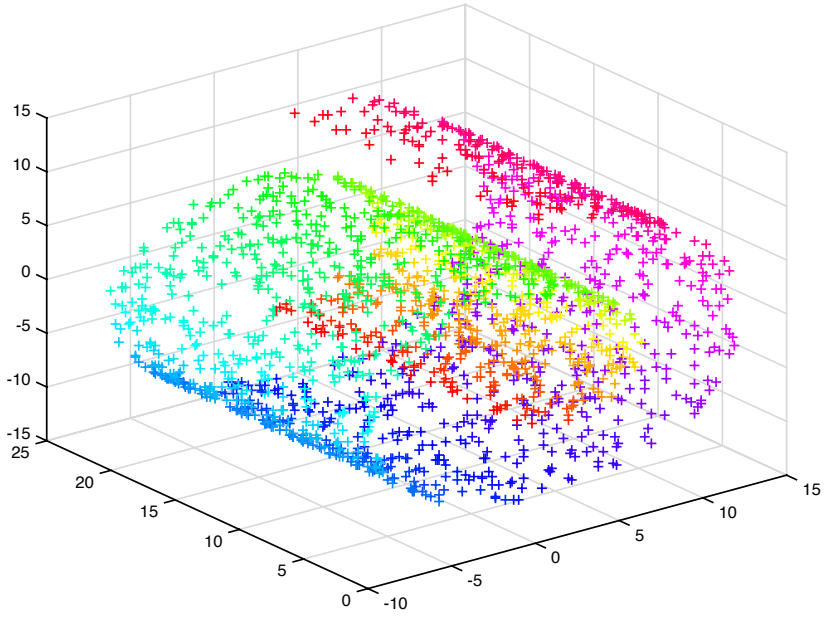
$$w_{i,j} = \frac{1}{\|x_i - x_j\| + \varepsilon} \qquad \text{or} \qquad w_{i,j} = \begin{cases} 1 \text{ if } \|x_i - x_j\| \leq \varepsilon \\ 0 \text{ else} \end{cases}$$

- Y are uniformly distributed in a circle

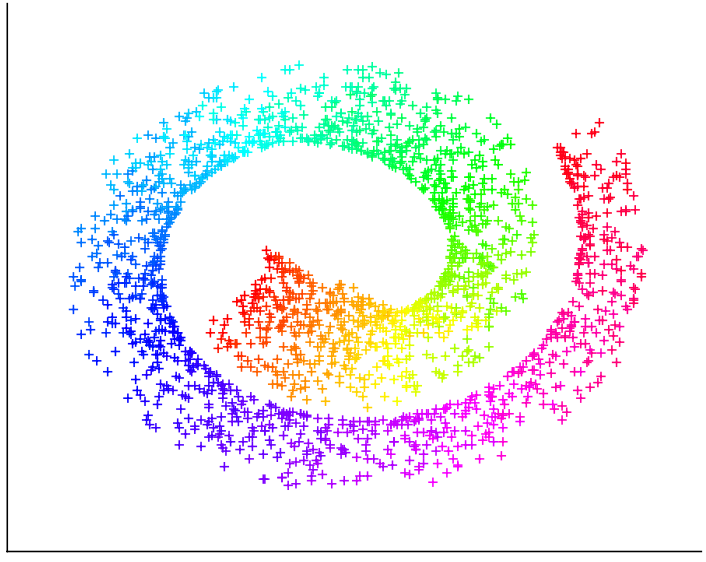


an example of MDS limitation

X



Y

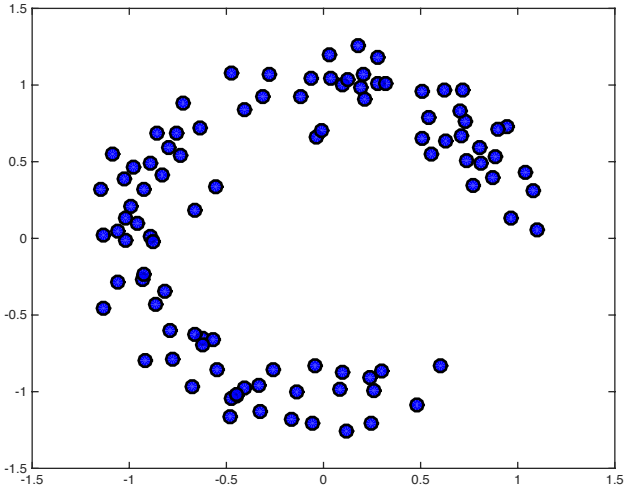


Solution: *metric learning* of $d(i,j)$

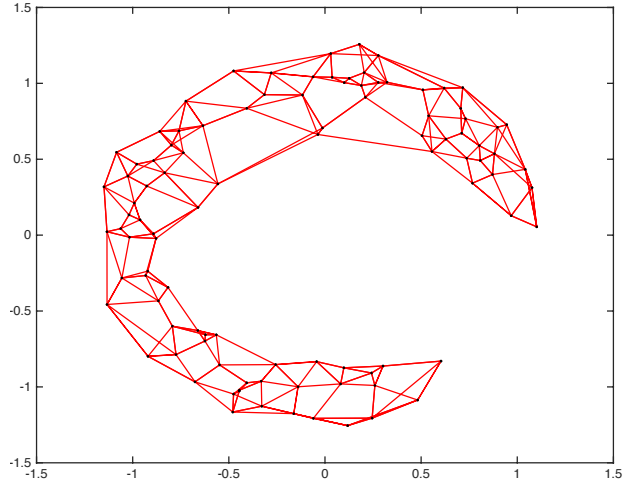
$$d_X(i,j) \text{ euclidean} \quad \rightarrow \quad d_g(i,j) \text{ geodesic}$$

Example of geodesic distance

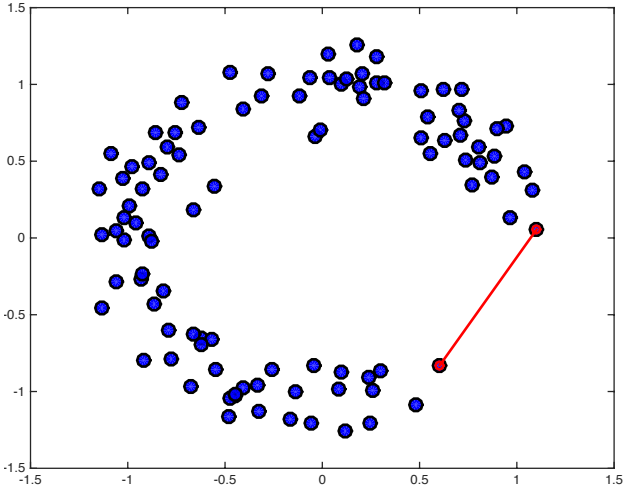
cloud of points



proximity graph

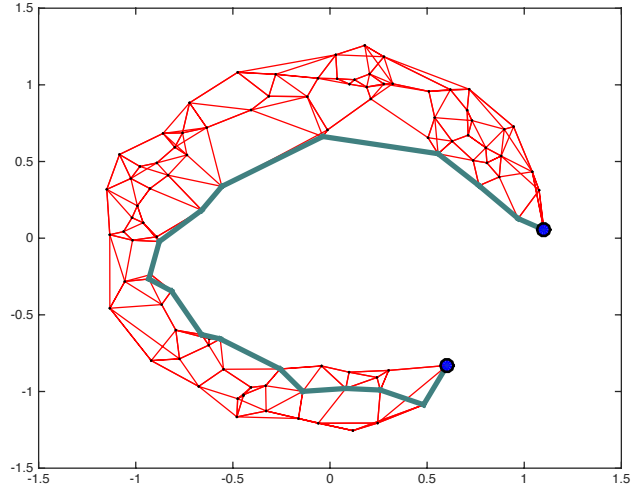


euclidean distance



small $d_X(i,j)$

geodesic distance



large $d_g(i,j)$

Isometric Feature Mapping (ISOMAP)

1. build a neighbor graph V
   - create the graph of the k nearest neighbors for each data point x_i
   - connect x_i with x_j if $\|x_i - x_j\| \leq \varepsilon$
2. find the shortest path in the graph (Dijkstra : $\mathcal{O}(kn^2 \log n)$)

$$d_g(i,j) = \begin{cases} \sum_{\ell=1}^{n_{ij}} d_X(x_{\phi(\ell)}, x_{\phi(\ell+1)}) & \text{with } \phi \text{ the shorstest path} \\ \infty \text{ else} & \text{connecting on } V,\ x_i \text{ to } x_j \end{cases}$$

3. compute Y with MDS using d_G instread of d_X,

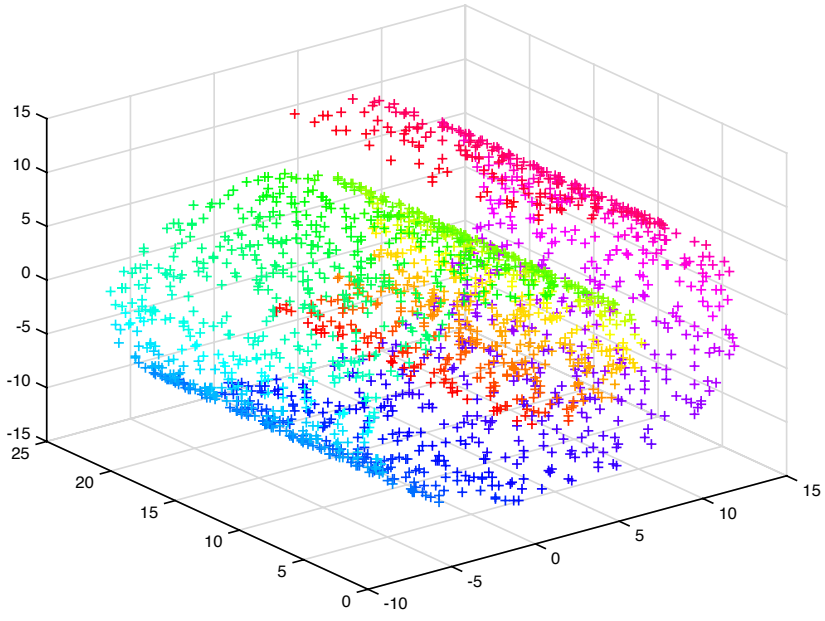
$$\min_{Y \in \mathbb{R}^{n \times q}} \sum_{i=1}^{n} \sum_{j=1}^{i-1} w_{i,j} (d_g(i,j) - d_Y(i,j))^2$$

http://web.mit.edu/cocosci/isomap/isomap.html https://github.com/gpeyre/matlab-toolboxes/tree/master/toolbox_dimreduc

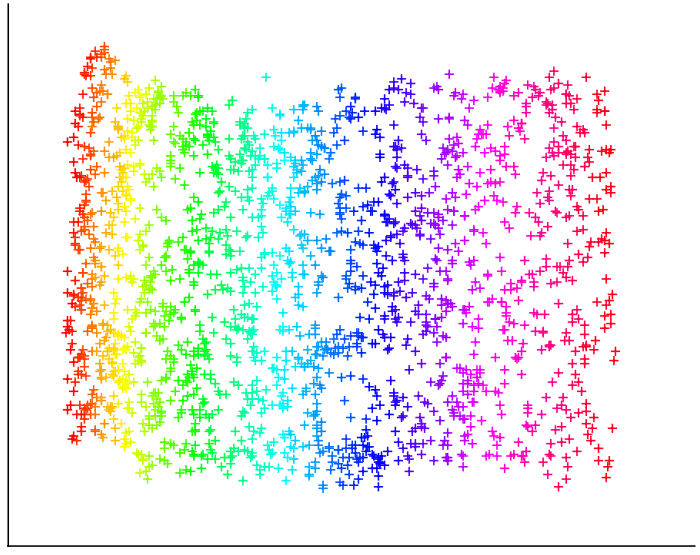
A Global Geometric Framework for Nonlinear Dimensionality Reduction J. B. Tenenbaum, et al. Science 290, 2000

Isometric Feature Mapping (ISOMAP)

X

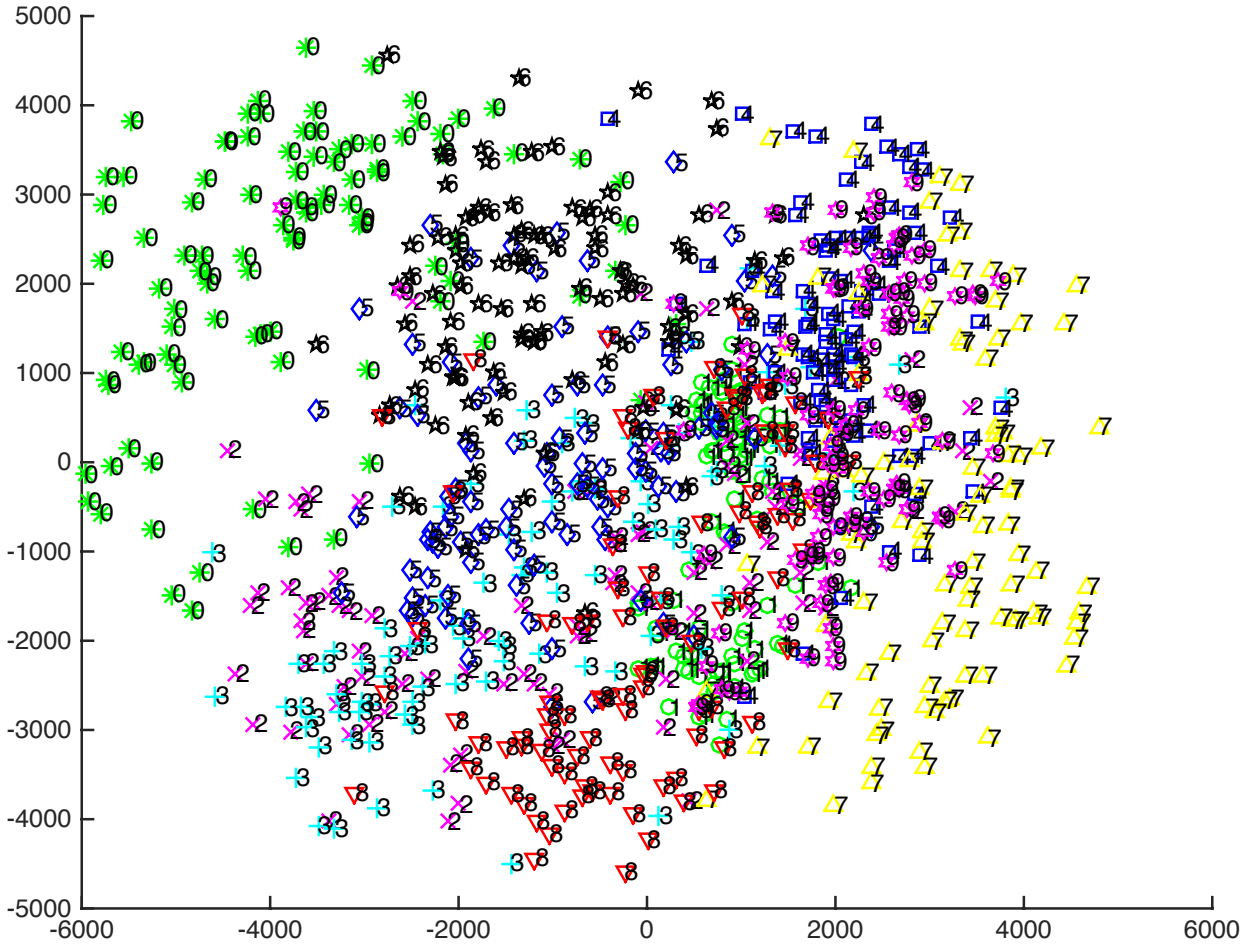


Y



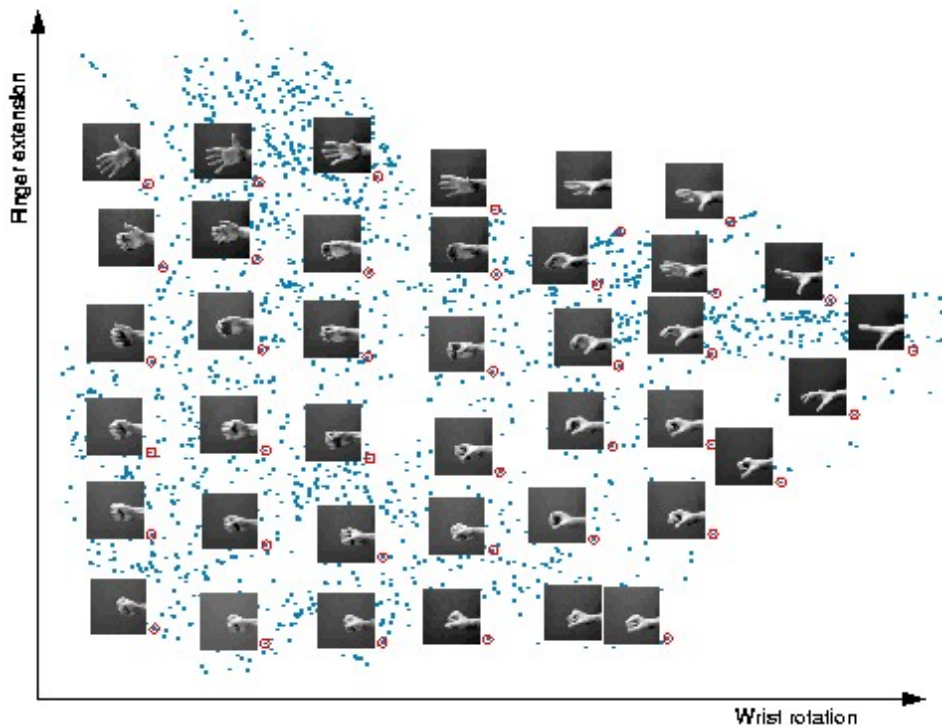
http://www.numerical-tours.com/matlab/shapes_7_isomap/

ISOMAP on MNIST



Problem with ISOMAP

- its a global non sparse method
- doesn't scale $\mathcal{O}(n^3)$
- given a new Y ithe decoding function is not known X. The pre image problem

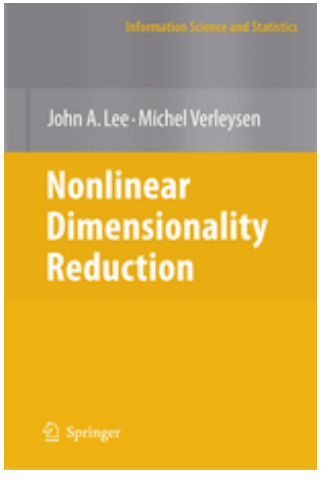


Isomap ($k = 6$) applied to $n = 2000$ images (64 pixels by 64 pixels) of a hand in different configurations. The images were generated by making a series of opening and closing movements of the hand at different wrist orientations, designed to give rise to a two-dimensional manifold.

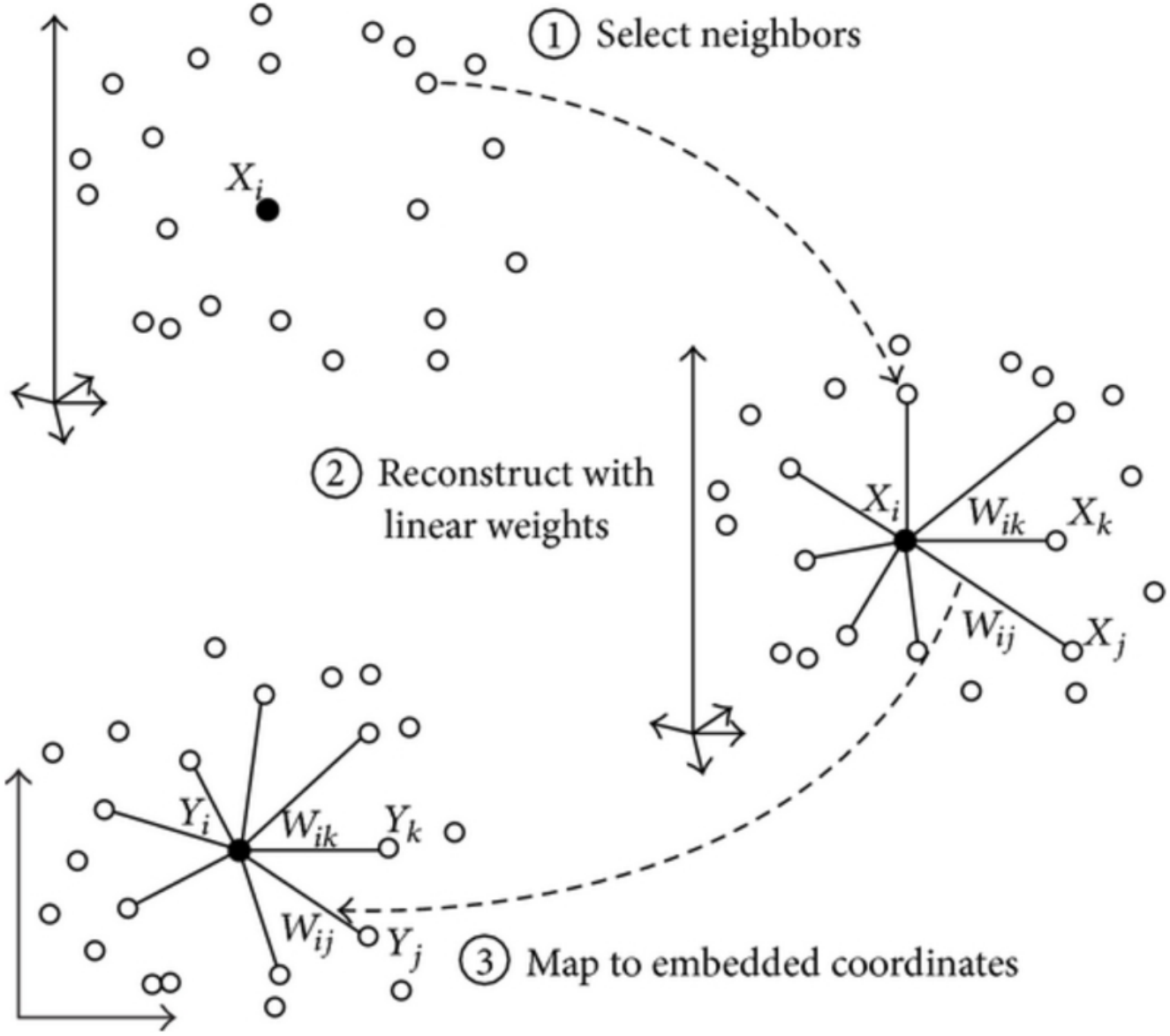
http://web.mit.edu/cocosci/isomap/handfig.html

Lecture road map

1. Introduction to hidden variables
2. PCA: principal component analysis
3. Distance preservation approaches (global methods)
4. Local approaches



Locally Linear Embedding (LLE)



Nonlinear dimensionality reduction by locally linear embedding. S. Roweis & L. Saul. Science, v.290 no.5500 , Dec.22, 2000.

Locally Linear Embedding (LLE)

Topology conservation: define a local metric

1. $V_{i,j} = 0$ if i and j are not among the $k < p$ nearest neigbors
2. compute the non zero weight: only for $V_{i,j} \neq 0$

$$\min_{V \in \mathbb{R}^{n \times n}} \sum_{i=1}^{n} \|x_i - \sum_{j=1}^{n} v_{i,j} x_j\|^2 \qquad \text{avec} \qquad \sum_{j=1}^{n} v_{i,j} = 1, \quad i = 1 : n$$

$\rightarrow$ n least square problems

3. how to get Y

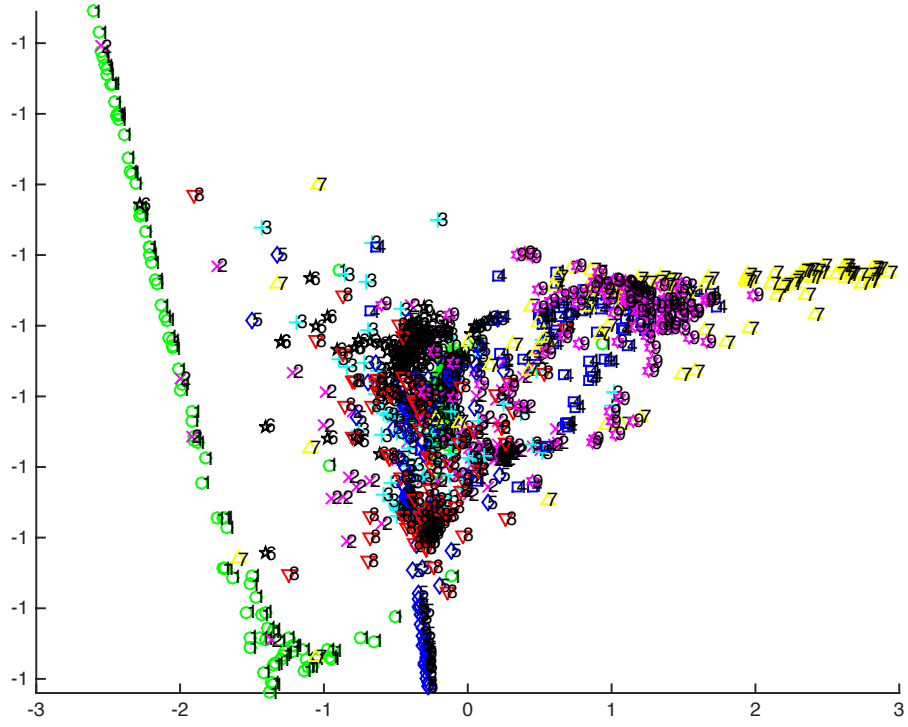
$$\min_{Y \in \mathbb{R}^{n \times q}} \sum_{i=1}^{n} \|y_i - \sum_{j=1}^{n} v_{i,j} y_j\|^2 = \|Y - VY\|_F^2$$

$\rightarrow$ SVD$(I - V)$, the 2 smallest non zero singular values.

https://cs.nyu.edu/~roweis/lle/algorithm.htm

Problems with LLE

- Most of the data points concentrate at the center
- A few points are far from the center to satisfy the unit variance constraint.



Stochastic Neighbor Embedding (SNE)

Model the conditional probability of a point x conditionally to our position in x_i

$$\mathbb{P}_X(x|x_i) = \frac{1}{Z_x} \exp^{\frac{-\|x-x_i\|^2}{2\sigma_i^2}} \qquad \mathbb{P}_Y(y|y_i) = \frac{1}{Z_y} \exp^{-\|y-y_i\|^2}$$

1. tune σ_i so that each point x_i have k neighbors
   - or to have the same perplexity p at each point

$$p = entropy(P_i) = log(k)$$

2. Minimize the Kullback-Leibler divergence between both distributions

$$\min_Y \sum_{i=1}^{n} KL(\mathbb{P}_X(i))||\mathbb{P}_Y(i)) = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{P}_X(j|i) \log \frac{\mathbb{P}_X(j|i)}{\mathbb{P}_Y(j|i)}$$

Visualizing data using t-SNE, Maaten, L.v.d. and Hinton, G., 2008. JMLR, Vol 9, pp. 2579–2605.

SNE optimization

Symetric case: build $\mathbb{P}_X$ so that $\mathbb{P}_X(j|i) = \mathbb{P}_X(i|j)$

$$\frac{\mathbb{P}_X(j|i) + \mathbb{P}_X(i|j)}{2}$$

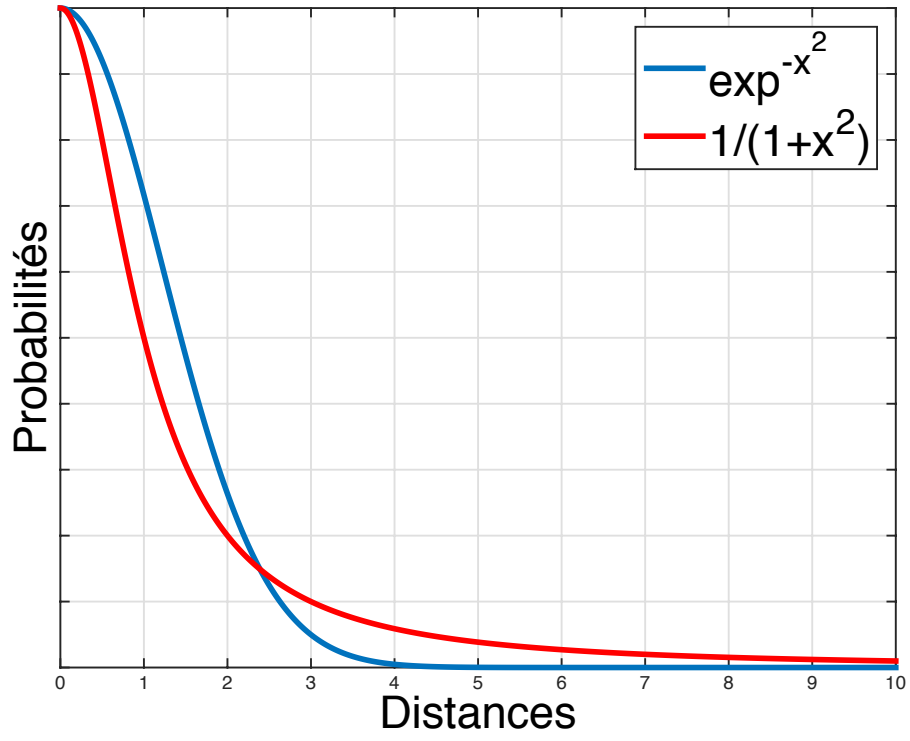
in that case:

$$\nabla_{Y(i)} KL\big(\mathbb{P}_X)||\mathbb{P}_Y\big) = 2\sum_{j=1}^{n} \underbrace{(\mathbf{y}_i - \mathbf{y}_j)}_{\text{similarity}} \underbrace{\big(\mathbb{P}_X(j|i) - \mathbb{P}_Y(j|i)\big)}_{\text{rigidity}}$$

Possible acceleration thanks to the Barnes-Hut-SNE $\mathcal{O}(n \log n)$

t-Stochastic Neighbor Embedding (t-SNE)

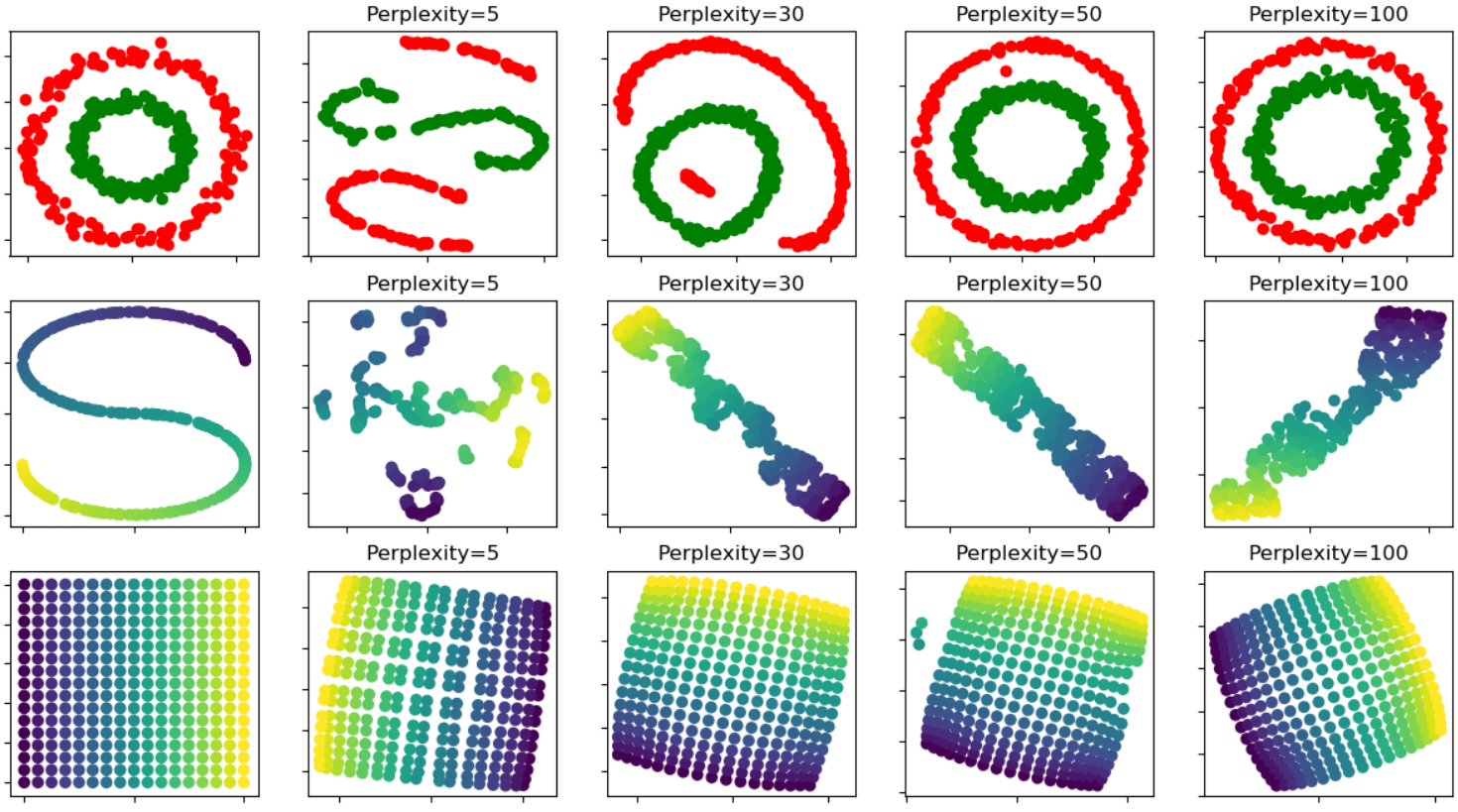
$$\mathbb{P}_Y(y_j|y_i) = \frac{1}{Z}\frac{1}{1+\|y-y_i\|^2}$$



- $\mathbb{P}_X = \mathbb{P}_Y$ large $\Rightarrow$ $d_Y < d_X$ (attraction)
- $\mathbb{P}_X = \mathbb{P}_Y$ small $\Rightarrow$ $d_Y > d_X$ (repulsion)

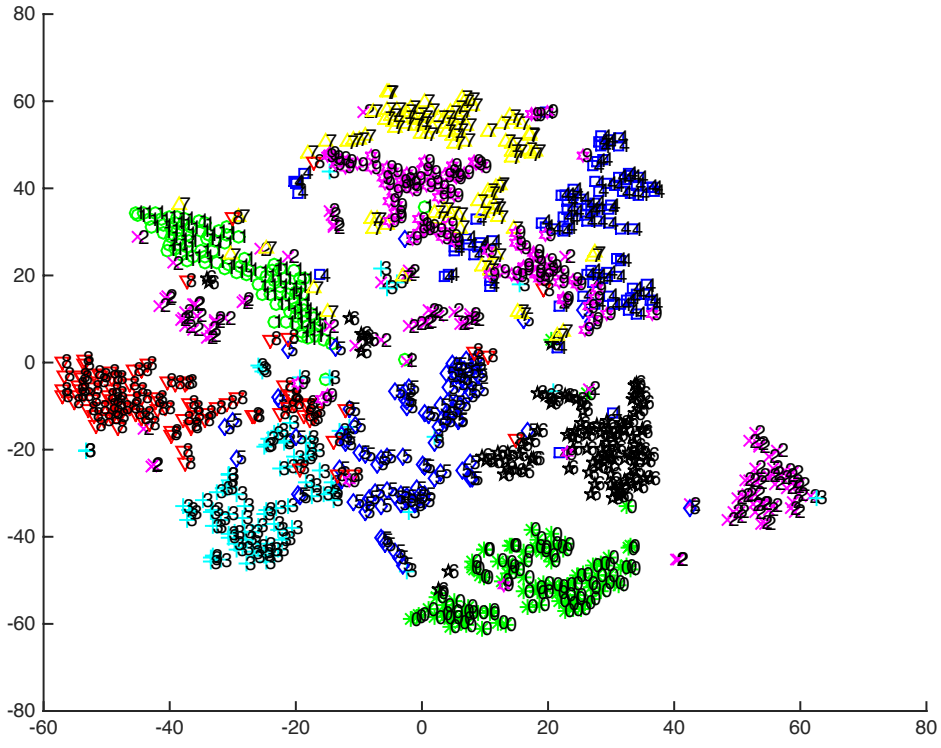
t-SNE: influence of the perplexity

"The perplexity can be interpreted as a smooth measure of the effective number of neighbors"



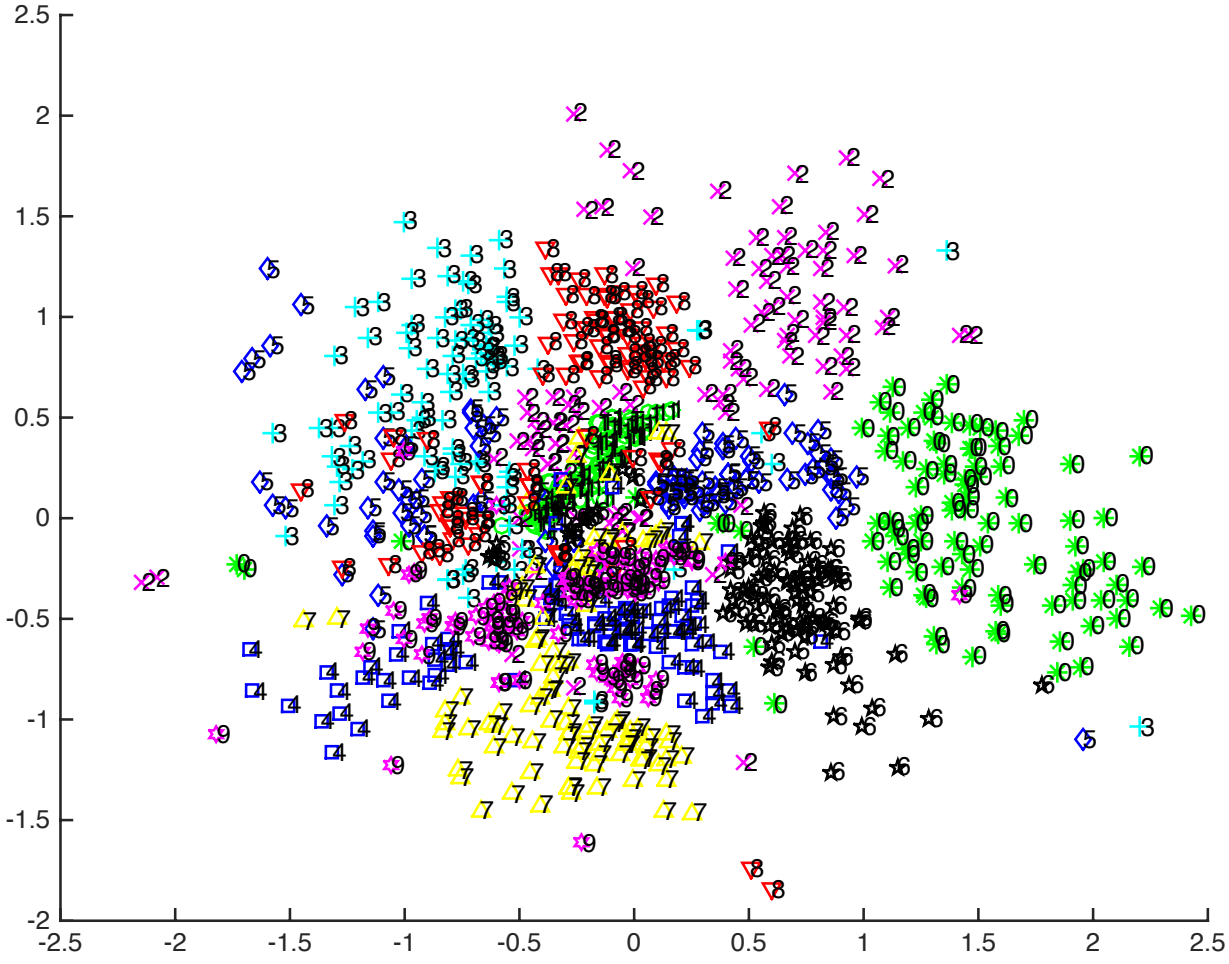
http://scikit-learn.org/stable/auto_examples/manifold/
https://distill.pub/2016/misread-tsne/

t-SNE on MNIST



https://lvdmaaten.github.io/tsne/

Multi-scale similarities in SNE



John A. Lee, Diego H. Peluffo, Michel Verleysen Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure Neurocomputing 2015, 169:246-261.

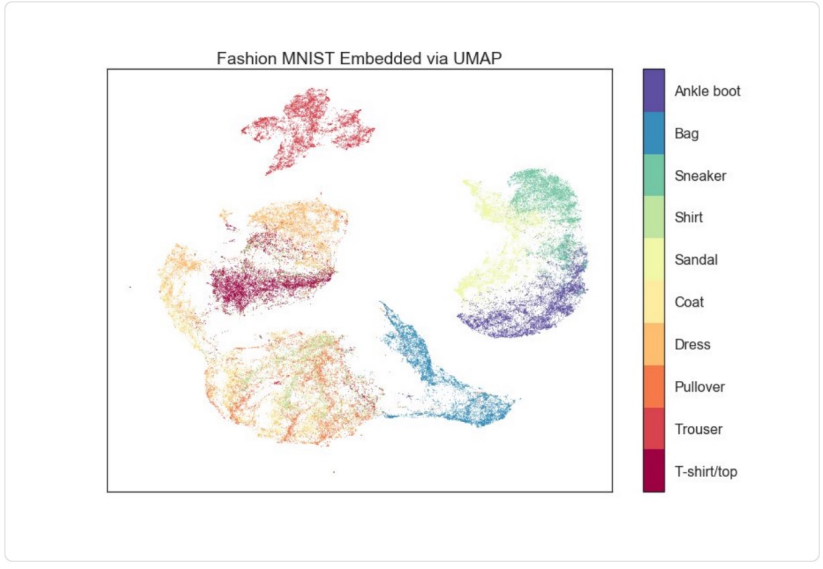
http://dx.doi.org/10.1016/j.neucom.2014.12.095

Uniform manifold approx. & projection (UMAP)

Tweets 524 Following 42 Followers 566 Likes 673

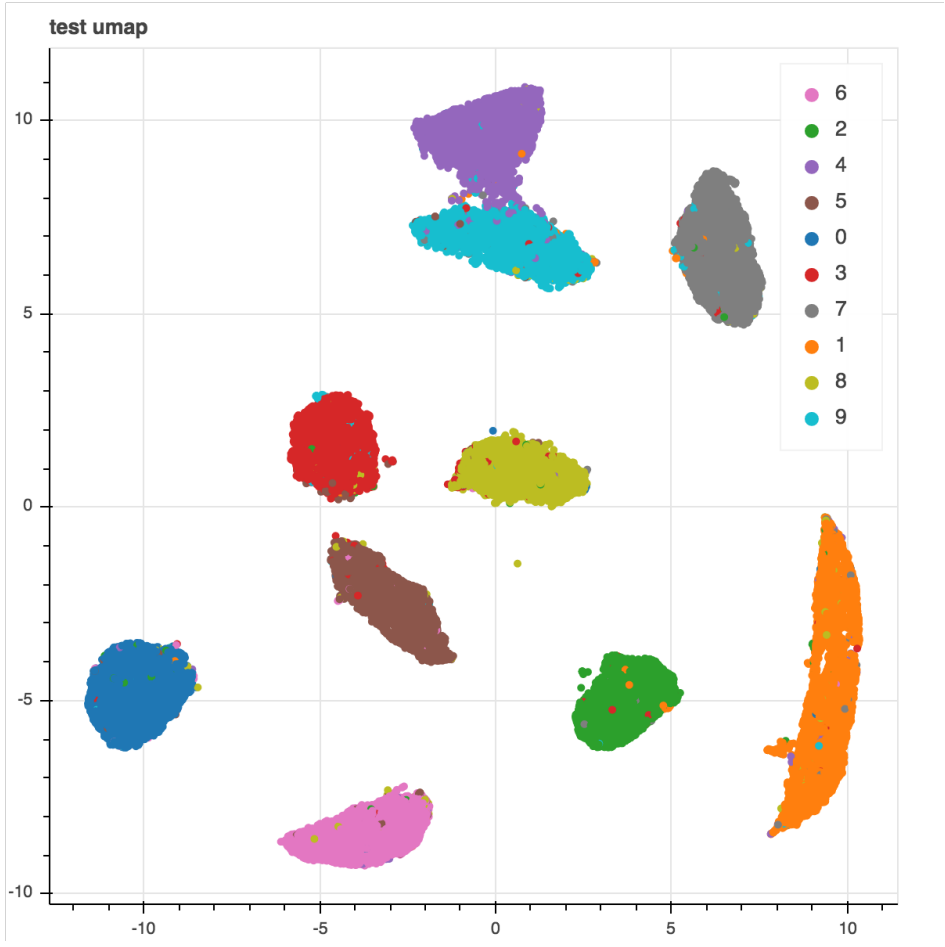
Leland McInnes @leland_mcinnes · 10 Nov 2017

The new numba based version of UMAP is out. Now faster than ever, it takes only 2.5 minutes to embed the full 70000 points of the 784-dimensional "Fashion MNIST" dataset. github.com/lmcinnes/umap



11 199 513

Uniform manifold approx. & projection (UMAP)

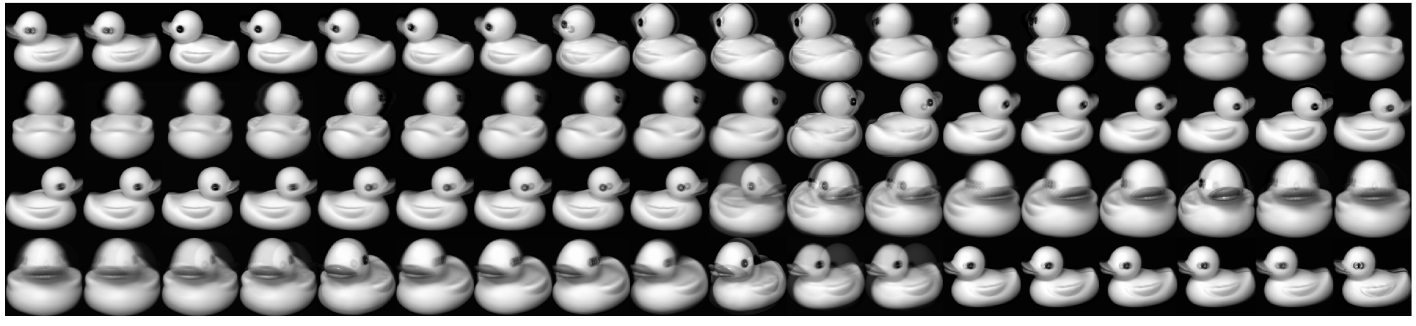


UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction Leland McInnes, John Healy (Submitted on 9 Feb 2018)

https://github.com/lmcinnes/umap

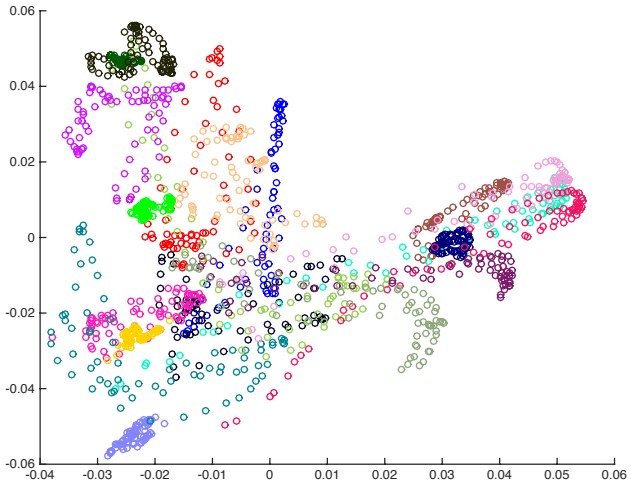
COIL 20

- Columbia University Image Library
- $n = 20 \times 72 = 1440$ images
- $p = 128 \times 128 = 16384$ pixels
- images for all of the objects in which the background has been discarded

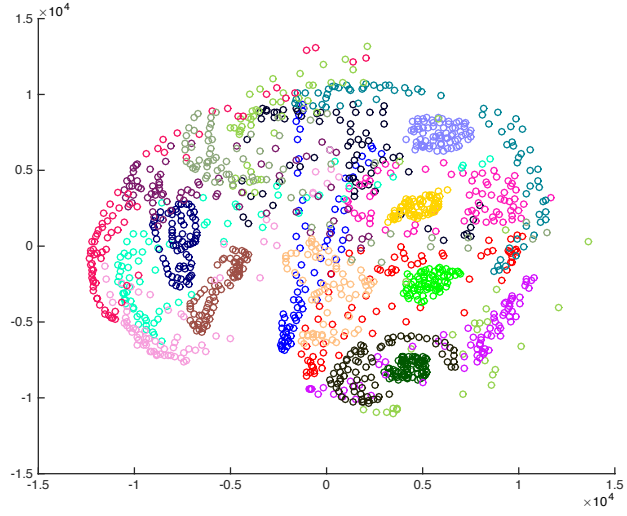


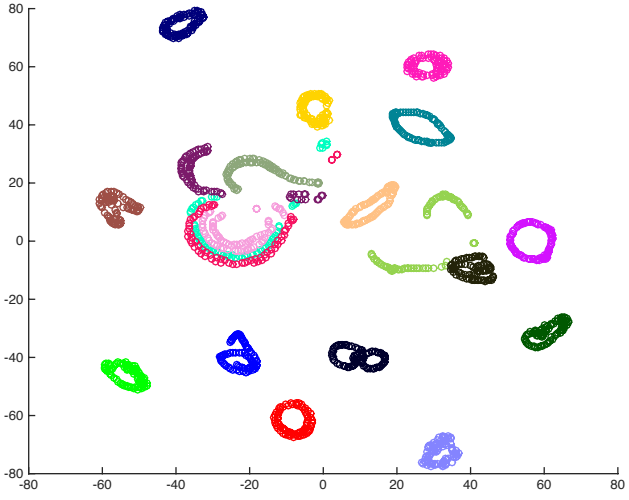
COIL 20

PCA

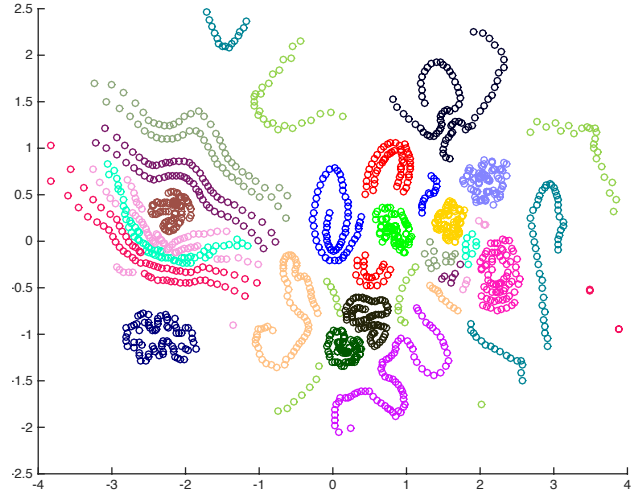


Sammon's projection (MDS)





t-SNE



Multi-scale similarities SNE

Conclusions

1. And the winner is UMAP . . . and t-SNE

2. other approaches. . .
   - Self organizing feature maps - SOM (Kohonen, 1974)
   - Curvilinear component analysis (CCA, Demartines & Hérault, 1995)
   - Kernel PCA
   - Laplacian Eigenmaps
   - Curvilinear distance analysis (CDA, Lee et al. 2004)
   - Semidefinite Embedding (SDE, Weinberger and Saul 2004)
   - Réseaux de neurones de type Autoencoder.
   - Maximum Variance Unfolding (MVU)
   - Weighted t-SNE `research.cs.aalto.fi/pml/software/ne/`
   - Metric learning
   - . . .

3. to play with: `doc.gold.ac.uk/~lfedd001/three/demo.html`