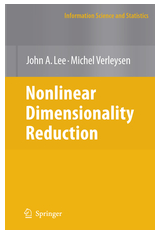


# Visualisation et réduction de dimension

Stéphane Canu

[asi.insa-rouen.fr/enseignants/~scanu](http://asi.insa-rouen.fr/enseignants/~scanu)  
[scanu@insa-rouen.fr](mailto:scanu@insa-rouen.fr)



MLA

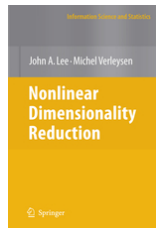
Vendredi 5 décembre 2025

# The 3 main kinds of machine learning



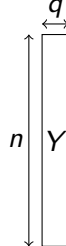
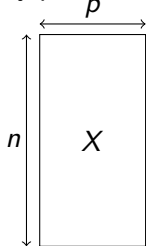
# Lecture road map

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

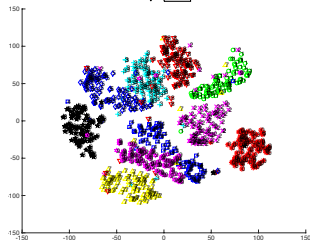


# A dimensionality reduction problem

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



$p = 784$



$q = 2$

# Dimensionality reduction: what for ?

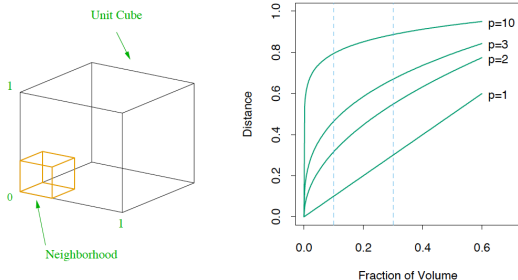
- Visualize ( $q = 2$  ou  $3$ )
  - ▶ validate data coding
  - ▶ detect outliers and miss labeled data
  - ▶ visualize classes
- Represent ( $q < p$ ) (or  $q > p$  cf. kernel representation)
  - ▶ summarize (remove noise)
  - ▶ preprocessing: brings statistic and computation efficiency
  - ▶ the hidden variable hypothesis

Coding/decoding functions

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

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 we have on distances in low dimension (2 or 3) no longer apply.

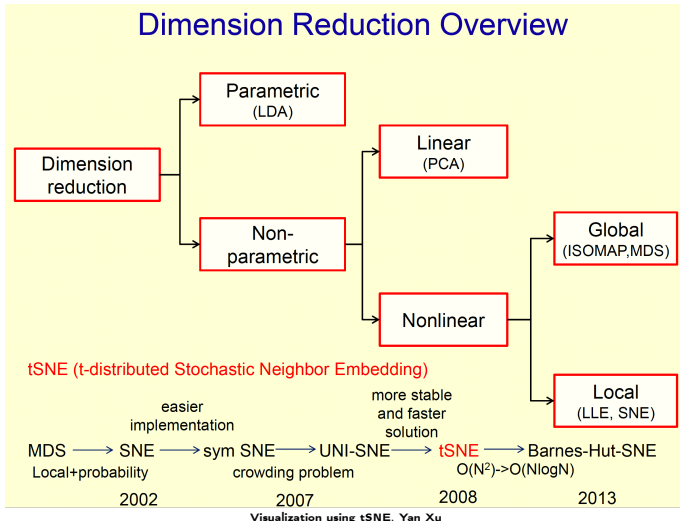
# Dimensionality reduction: the big picture

parametric (= supervised ) vs. non parametric (= unsupervised)

linear vs. non linear

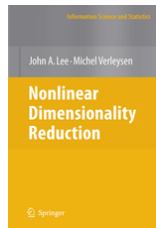
global metric vs. local metric

## Dimension Reduction Overview



# 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

$$X = YV^T + B$$

linear coding:  $cod : \mathbb{R}^p \longrightarrow \mathbb{R}^q$  ,  $X \longmapsto Y = XV$   
 $dec : \mathbb{R}^q \longrightarrow \mathbb{R}^p$  ,  $Y \longmapsto YV^T$

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

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

or maximize the variance of the projection

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

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

$$\min_{y \in \mathbb{R}^n} \|XX^T - yy^T\|^2$$

# PCA computation

## Theorem (Eckart & Young, 1936)

The unique solution of

$$\min_{y,v} J(y,v) \quad \text{with} \quad J(y,v) = \|X - yv^T\|_F^2$$

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

proof

$$\begin{cases} \nabla_y J(y,v) = -2Xv + 2\|v\|^2 y = 0 \\ \nabla_v J(y,v) = -2X^T y + 2\|y\|^2 v = 0 \end{cases}$$

3 different ways to get  $Y$

$$\text{svd}(X), \text{eig}(X^T X), \text{eig}(XX^T)$$

# PCA as an encoding/decoding mechanism

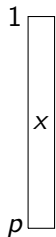
Coding/decoding functions

$$\begin{aligned} \text{cod} : \mathbb{R}^p &\longrightarrow \mathbb{R}^q, & x &\longmapsto y = \text{cod}(x) \\ \text{dec} : \mathbb{R}^q &\longrightarrow \mathbb{R}^p, & y &\longmapsto x_r = \text{dec}(y) \end{aligned}$$

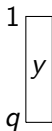
Let  $V$  be the  $p \times q$  matrix

$$\text{cod}(x) = V^T x = y$$

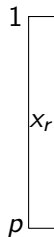
$$\text{dec}(y) = Vy = V^T Vx$$



$V^T$



$V$



$\text{cod}(x)$

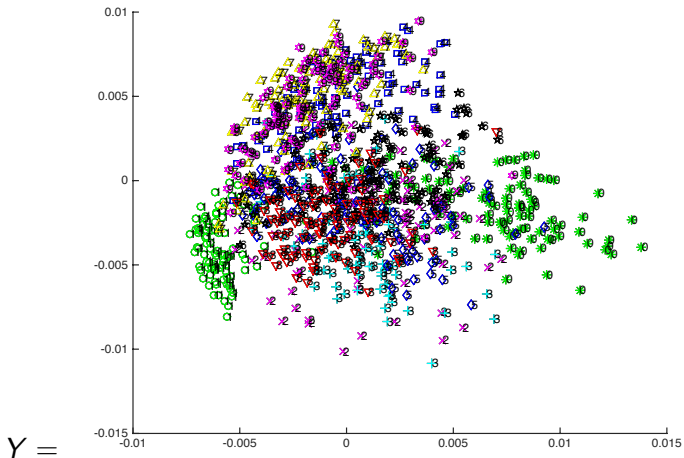
$\text{dec}(y)$

Autoencoders

# 2d PCA on the MNIST data

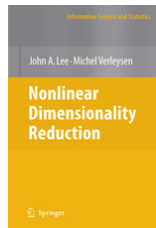
$p = 784$

$q = 2$



# Lecture road map

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



## Distances

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

Example: the euclidian distance

$$d(s, t) = \|s - t\| = \sqrt{\sum_{i=1}^n (s_i - t_i)^2}$$

Distance and dot product

$$d(s, t)^2 = \|s\|^2 + \|t\|^2 - 2s^T t$$

hypermetrics and quasi distances

distances and probabilities

Gram matrix

$$G = XX^T \quad \text{avec} \quad G_{ij} = x_i^T x_j = 1 - \frac{1}{2}d(x_i, x_j)$$

# Multidimensional scaling (MDS)

Given

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

Distances conservation

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

Related optimization problems (many variants):

- classical MDS (Torgerson, 1958)
- Kruskal -Shepard method (Kruskal, 1964)
- Sammon projection (Sammon, 1969)
- MDS INDSCAL (Carroll et Chang, 1970)
- ...

# Classical MDS

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

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

- 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

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

- 2 if  $X_c$  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.

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

# MDS and PCA

MDS = PCA

- if the data is lying on an hyperplane  
→ 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  $X$ ,

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

let  $X^c$  be the centered data matrix

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

recall the distance and scalar product formula  $d_X(i, j)^2 = \|x_i - x_j\|^2 = \|x_i\|^2 + \|x_j\|^2 - 2x_i^T x_j$

let  $G$  be the Gram matrix  $G = XX^T$  and  $\delta = \text{diag}(G)$  with  $\delta_i = \|x_i\|^2$   
we have, with  $D_X$  the distances matrix of general term  $d_X(i, j)^2$

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

and

$$HD_X H = -2X^c X^{cT}$$

Classical MDS aims at minimizing

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

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

# Weighted MDS: Sammon's projection

Reinforce closed neighbors (. . . and penalized distant ones)

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

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

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

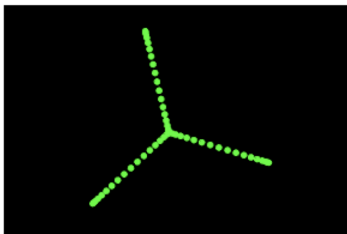
## Sammon's projection Quasi Newton

$$J_S(Y) = \sum_{i=1}^n \sum_{j=1}^n w_{i,j} (d_X(i,j) - d_Y(i,j))^2 \quad \text{with} \quad d_Y(i,j) = \|y_i - y_j\|$$

$$\begin{aligned} \nabla_{y_i} J_S(Y) &= -2 \sum_{j=1, j \neq i}^n w_{i,j} (d_X(i,j) - d_Y(i,j)) \nabla_{y_i} d_Y(i,j) \\ &= -2 \sum_{j=1, j \neq i}^n \frac{d_X(i,j) - d_Y(i,j)}{d_X(i,j) d_Y(i,j)} (y_i - y_j) \end{aligned}$$

# an illustration of Sammon's projection

Data: 3 mutually perpendicular circles in 6 dimensional space



(a) Projection by PCA does not preserve the structure of the dataset — it is unclear that it consists of three circles



(b) The Sammon mapping preserves the topological structure — while the circles become distorted, there are still three closed loops meeting at a single point

## 2 issues with Sammon's projection

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

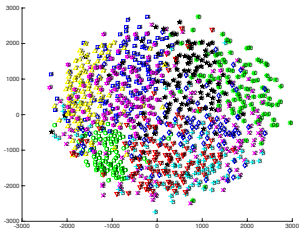
$$w_{i,j} = \frac{1}{\|x_i - x_j\|}$$

the criterion preserve all small distances.

Alternatives:

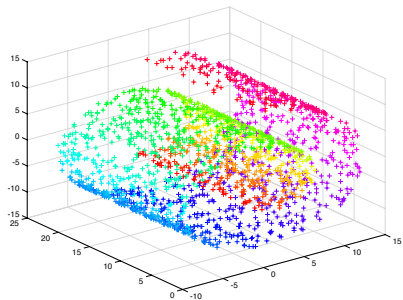
$$w_{i,j} = \begin{cases} 1 & \text{si } \|x_i - x_j\| \leq \varepsilon \\ 0 & \text{sinon} \end{cases}$$

- $Y$  are uniformly distributed in a circle

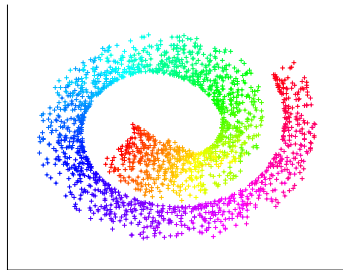


# an example of MDS limitation

X



Y

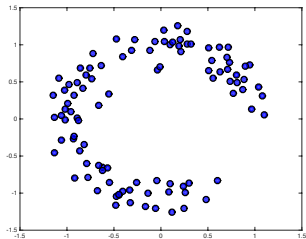


Solution: *metric learning* of  $d(i, j)$

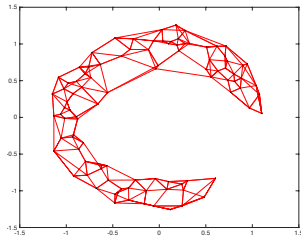
$$d_X(i, j) \text{ euclidean} \quad \rightarrow \quad d_g(i, j) \text{ geodesic}$$

# Example of geodesic distance

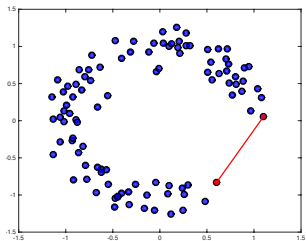
cloud of points



proximity graph

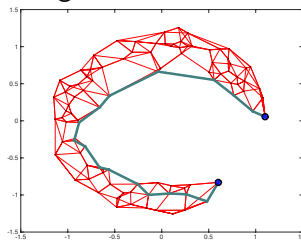


euclidean distance



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 :  $\mathcal{O}(kn^2 \log n)$ )

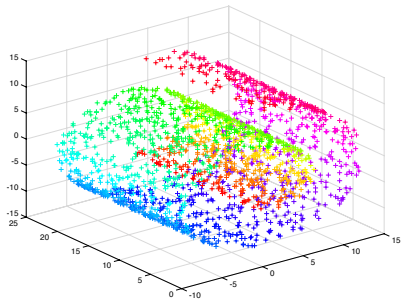
$$d_g(i, j) = \begin{cases} \sum_{\ell=1}^{n_{ij}} d_X(x_{\phi(\ell)}, x_{\phi(\ell+1)}) & \text{with } \phi \text{ the shortest path} \\ \infty & \text{else} \end{cases} \text{ connecting on } V, x_i \text{ to } x_j$$

- 3 compute  $Y$  with MDS (or Sammon's proj.) using  $d_G$  instead of  $d_X$ ,

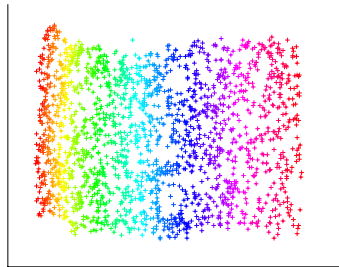
$$\min_{Y \in \mathbb{R}^{n \times q}} \sum_{i=1}^n \sum_{j=1}^{i-1} w_{i,j} (d_g(i, j) - d_Y(i, j))^2$$

# Isometric Feature Mapping (ISOMAP)

X

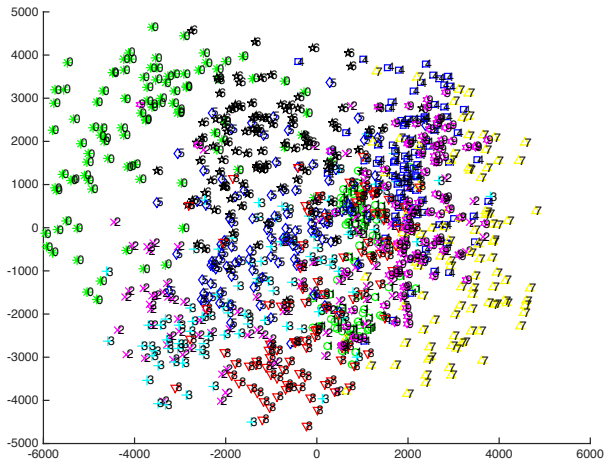


Y



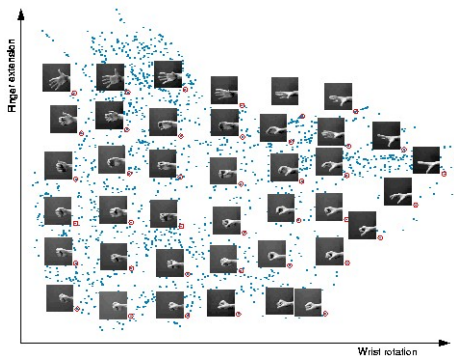
[http://www.numerical-tours.com/matlab/shapes\\_7\\_isomap/](http://www.numerical-tours.com/matlab/shapes_7_isomap/)

# ISOMAP on MNIST



# Problem with ISOMAP

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

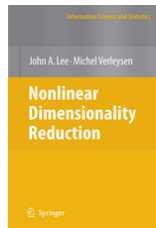


Isomap ( $k = 6$ ) applied to  $n = 2000$  images (64 pixels by 64 pixels) of a hand in different configurations. The images were generated by making a series of opening and closing movements of the hand at different wrist orientations, designed to give rise to a two-dimensional manifold.

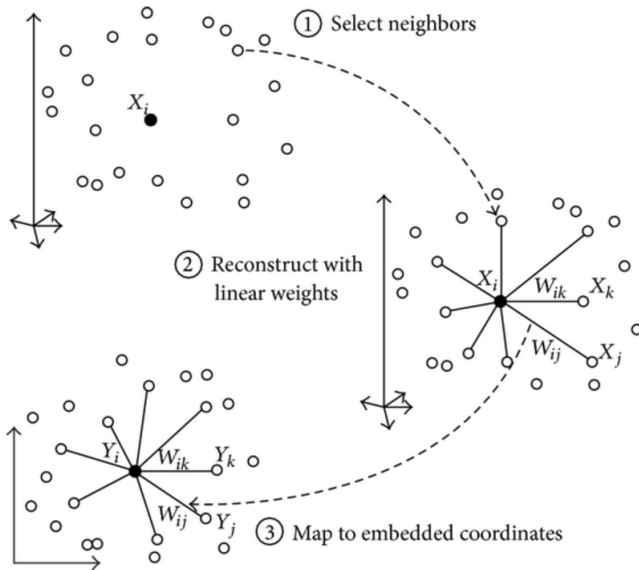
<http://web.mit.edu/cocosci/isomap/handfig.html>

# 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)



# 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 neighbors
- 2 compute the non zero weight: only for  $V_{i,j} \neq 0$

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

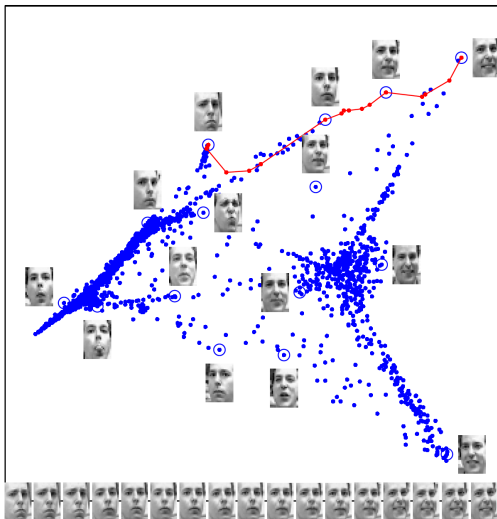
→  $n$  least square problems

- 3 how to get  $Y$

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

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

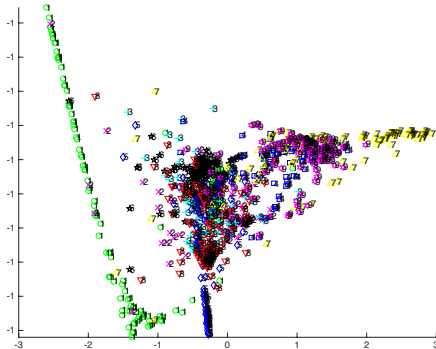
# Locally Linear Embedding (LLE)



**FIGURE 14.45.** Images of faces mapped into the embedding space described by the first two coordinates of LLE. Next to the circled points, representative faces are shown in different parts of the space. The images at the bottom of the plot correspond to points along the top right path (linked by solid line), and illustrate one particular mode of variability in pose and expression.

# 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  $x_i$

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

- 1 tune  $\sigma_i$  so that each point  $x_i$  have  $k$  neighbors
  - ▶ or to have the same perplexity  $p$  at each point

$$p = \text{entropy}(P_i) = \log(k)$$

- 2 Minimize the Kullback-Leibler divergence between both distributions

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

## SNE optimization

Symmetric 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:

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

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

# Derivation of the SNE gradient

$$\sum_{j=1}^n Q_{ij} = 1 \quad \sum_{j=1}^n P_{ij} = 1$$

$$P_{ij} = \frac{1}{Z} \exp \frac{-\|x_i - x_j\|^2}{2\sigma^2}$$

$$Q_{ij} = \frac{\exp -d_{ij}}{\sum_{k=1}^n \exp -d_{ik}}$$

$$Q_{ii} = 0$$

$$C = \sum_{i=1}^n \sum_{j=1}^n P_{ij} \log \frac{P_{ij}}{Q_{ij}} = - \sum_{i=1}^n \sum_{j=1}^n P_{ij} \log Q_{ij} + \bullet$$

$$d_{ij} = \frac{1}{2} \|x_i - x_j\|^2$$

$$= + \sum_{i=1}^n \sum_{j=1}^n P_{ij} d_{ij} + \sum_{i=1}^n \left( \sum_{j=1}^n P_{ij} \right) \log \sum_{k=1}^n \exp -d_{ik} = \sum_{i=1}^n \sum_{j=1}^n P_{ij} d_{ij} + \sum_{i=1}^n \log \sum_{k=1}^n \exp -d_{ik}$$

$$\frac{\partial C}{\partial y_i} = \sum_{j=1}^n (P_{ij} + P_{ji}) (y_i - y_j) - \frac{\exp -d_{ij}}{\sum_{k=1}^n \exp -d_{ik}} (y_i - y_j)$$

$$y_i \rightarrow d_{ij}$$

$$y_i \rightarrow d_{ji}$$

$$= \sum_{j=1}^n (P_{ij} + P_{ji}) (y_i - y_j) - (Q_{ij} + Q_{ji}) (y_i - y_j)$$

$$y_i \rightarrow d_{i \neq j}$$

$$y_i \rightarrow d_{j \neq i}$$

$$C = - \sum_{i=1}^n P_i \log Q_i = \sum_{i=1}^n P_i d_i + \sum_{i=1}^n P_i \log \sum_{j=1}^n \exp -d_{ij}$$

$$Q_i = \frac{\exp -d_i}{\sum_{j=1}^n \exp -d_{ij}}$$

$$\frac{\partial C}{\partial y} = \sum_{i=1}^n P_i (y_i - y_j) - \frac{\exp -d_i}{\sum_{j=1}^n \exp -d_{ij}} (y_i - y_j) - \sum_{j=1}^n (P_i - Q_{ij}) (y_i - y_j)$$

$$d_i = \frac{1}{2} \|y_i - y_j\|^2$$

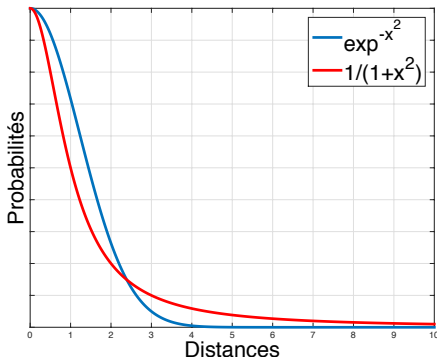
Sort...

# Derivation of the tSNE gradient

$$\begin{aligned}
 & \sum_{j=1}^n Q_{ij} = 1 \quad \sum_{j=1}^n P_{ij} = 1 \quad P_{ij} = \frac{1}{\sum_{k=1}^n \exp\left(-\frac{\|x_i - x_k\|^2}{2\sigma_i^2}\right)} \\
 & Q_{ij} = \frac{1}{1+d_{ij}} \quad d_{ij} = \|y_i - y_j\|^k \quad Q_{ii} = 0 \\
 & C = \sum_{i=1}^n \sum_{j=1}^n P_{ij} \log \frac{P_{ij}}{Q_{ij}} = - \sum_{i=1}^n \sum_{j=1}^n P_{ij} \log Q_{ij} + \bullet \\
 & = + \sum_{i=1}^n \sum_{j=1}^n P_{ij} \log(1+d_{ij}) + \sum_{i=1}^n \left( \sum_{j=1}^n P_{ij} \right) \log \sum_{k=1}^n \frac{1}{1+d_{ik}} = \sum_{i=1}^n \sum_{j=1}^n P_{ij} \log(1+d_{ij}) + \sum_{i=1}^n \log \sum_{k=1}^n \frac{1}{1+d_{ik}} \\
 & \frac{\partial C}{\partial y_i} = \sum_{j=1}^n \left( P_{ij} + \frac{P_{ji}}{1+d_{ji}} \right) (y_i - y_j) - \sum_{k=1}^n \frac{\frac{1}{1+d_{ik}}}{\sum_{l=1}^n \frac{1}{1+d_{il}}} (y_i - y_j) \quad \begin{array}{l} y_i \rightarrow d_{ij} \\ y_i \rightarrow d_{ji} \end{array} \\
 & = \sum_{j=1}^n \left( P_{ij} + \frac{P_{ji}}{1+d_{ji}} \right) (y_i - y_j) - \frac{(Q_{ij} + Q_{ji})}{1+d_{ij}} (y_i - y_j) \quad \begin{array}{l} y_i \rightarrow d_{ik} \\ y_i \rightarrow d_{jk} \end{array} \\
 & C = \sum_{i=1}^n P_i \log Q_i = \sum_{i=1}^n P_i \log(1+d_i) + \sum_{i=1}^n P_i \log \sum_{k=1}^n \frac{1}{1+d_{ik}} \quad Q_i = \frac{1}{\sum_{k=1}^n \frac{1}{1+d_{ik}}} \quad d_i = \|y_i\|^k \\
 & \nabla C = \sum_{i=1}^n \left[ \frac{P_i}{1+d_i} - \sum_{k=1}^n \frac{\frac{1}{1+d_{ik}}}{\sum_{l=1}^n \frac{1}{1+d_{il}}} \right] (y_i - y_j) = \sum_{i=1}^n \left( P_i - Q_i \left[ \frac{1}{1+d_i} \right] \right) (y_i - y_j) \quad P_i = \frac{\exp\left(-\frac{\|x_i - x_i\|^2}{2\sigma_i^2}\right)}{\sum_{k=1}^n \exp\left(-\frac{\|x_i - x_k\|^2}{2\sigma_i^2}\right)} \quad \text{(?)}
 \end{aligned}$$

# t-Stochastic Neighbor Embedding (t-SNE)

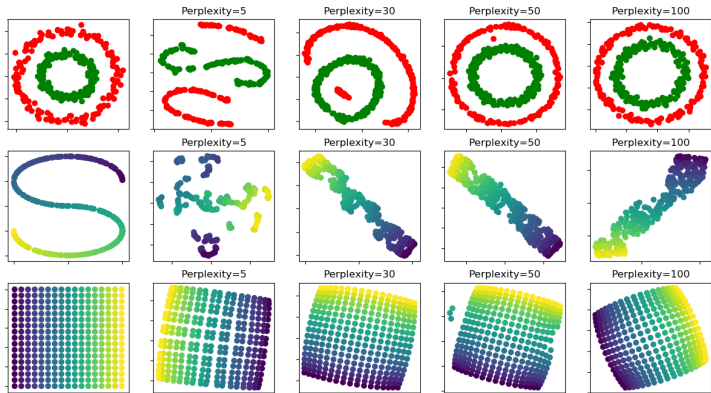
$$\mathbb{P}_Y(y_j|y_i) = \frac{1}{Z} \frac{1}{1 + \|y - y_i\|^2}$$



- $\mathbb{P}_X = \mathbb{P}_Y$  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"



## t-SNE practical optimization

With

$$v_{ij} = \frac{\mathbb{P}_X(j|i) + \mathbb{P}_X(i|j)}{2} \qquad w_{ij} = \frac{\mathbb{P}_Y(j|i) + \mathbb{P}_Y(i|j)}{2}$$

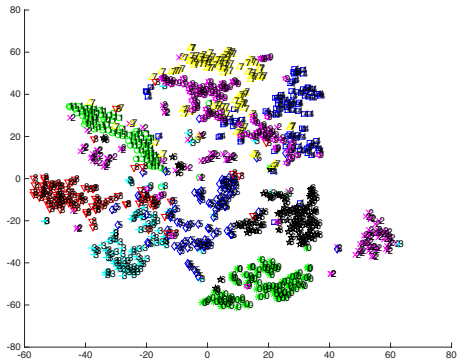
The gradient components are

$$\frac{\partial KL(\mathbb{P}_X || \mathbb{P}_Y)}{\partial y_i} = 2 \sum_{j=1}^n v_{ij} w_{ij} (y_i - y_j) - 2 \frac{n}{Z} \sum_{j=1}^n w_{ij}^2 (y_i - y_j)$$

With an exaggeration factor  $\rho = 12$

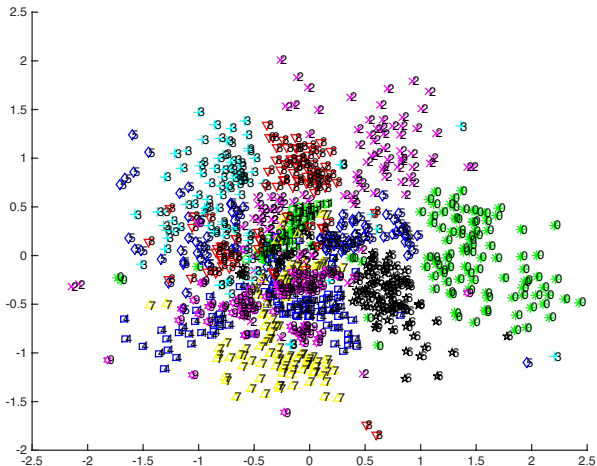
$$\frac{\partial KL_\rho(\mathbb{P}_X || \mathbb{P}_Y)}{\partial y_i} = 2 \sum_{j=1}^n v_{ij} w_{ij} (y_i - y_j) - 2 \frac{n}{Z^\rho} \sum_{j=1}^n w_{ij}^2 (y_i - y_j)$$

# t-SNE on MNIST



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

# Refinement: 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

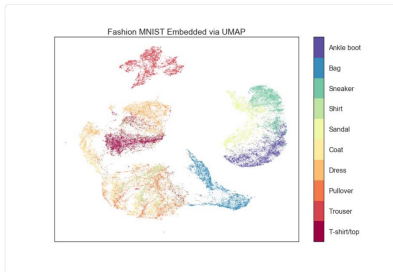
followers  
566

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](https://github.com/lmcinnes/umap)



11

199

513



# What the UMAP algorithm actually does

$$d_x(x_i, x_j) = \|x_i - x_j\|^2$$

$$d_y(y_i, y_j) = \|y_i - y_j\|^2$$

$$v_i(x_i, x_j) = \exp \frac{-d_x(x_i, x_j) + m_i}{\sigma_i}$$

$$w_j(x_i, x_j) = \exp \frac{-d_y(y_i, y_j) + m_j}{\sigma_j}$$

- 1 estimate  $m_i$  and  $\sigma_i$  using  $k$  neighbors of point  $x_i$ 
  - $m_i$  the distance between  $x_i$  and its nearest neighbor
  - $\sigma_i$  the diameter of the neighborhood of  $x_i$

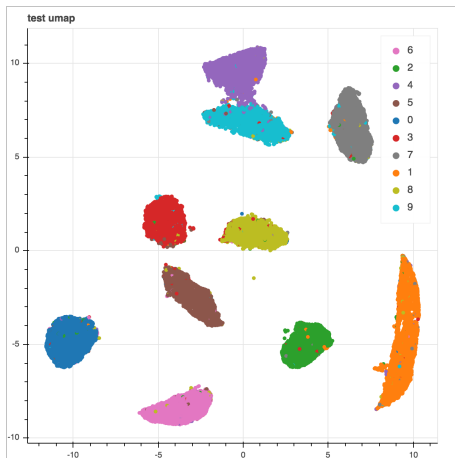
- 2 symmetrize  $v$  and  $w$

$$v_s(i, j) = v_i(x_i, x_j) + v_j(x_i, x_j) - v_i(x_i, x_j)v_j(x_i, x_j)$$

- 3 Minimize the cross entropy (close to the KL divergence) using SGD

$$\min_Y \sum_{i=1}^n \sum_{j=1}^n v_s(i, j) \log \frac{v_s(i, j)}{w_s(i, j)} + (1 - v_s(i, j)) \log \frac{1 - v_s(i, j)}{1 - w_s(i, j)}$$

# 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>

# Manifold learning with Scikit-learn



Install User Guide API Examples Community More ▾

Prev Up Next

scikit-learn 1.2.dev0

[Other versions](#)

Please [cite us](#) if you use the software.

## 2.2. Manifold learning

2.2.1. Introduction

2.2.2. Isomap

2.2.3. Locally Linear Embedding

2.2.4. Modified Locally Linear Embedding

2.2.5. Hessian Eigenmapping

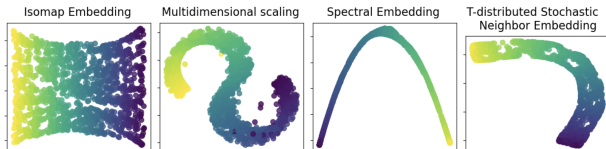
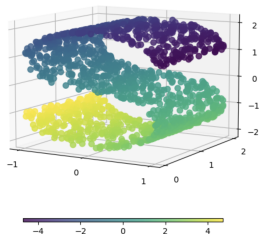
2.2.6. Spectral Embedding

2.2.7. Local Tangent Space Alignment

2.2.8. Multi-dimensional Scaling (MDS)

2.2.9. t-distributed Stochastic Neighbor Embedding (t-SNE)

2.2.10. Tips on practical use



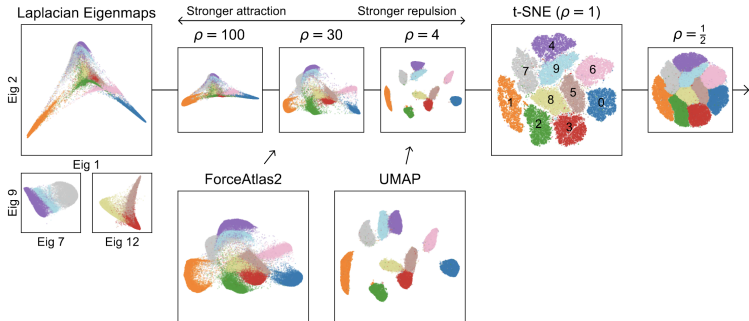
New ones:

2.2.5. Hessian Eigenmapping,

2.2.6. Spectral Embedding (Laplacian Eigenmaps),

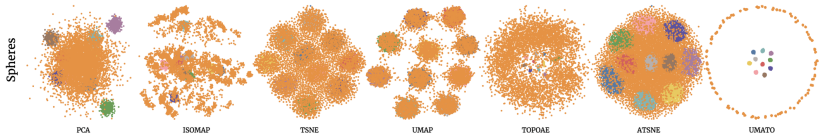
2.2.7. Local Tangent Space Alignment

# On going research field (manifold learning)



A Unifying Perspective on Neighbor Embeddings along the Attraction-Repulsion Spectrum, Böhm et al, JMLR, 2021

<https://github.com/berenslab/ne-spectrum>



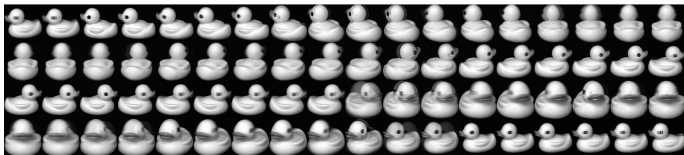
Uniform Manifold Approximation with Two-phase Optimization, Ko et al, 2021

<https://github.com/hyungkwonko/umato>

# COIL 20

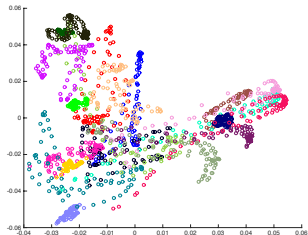


- Columbia University Image Library
- $n = 20 \times 72 = 1440$  images
- $p = 128 \times 128 = 16384$  pixels
- images for all of the objects in which the background has been discarded

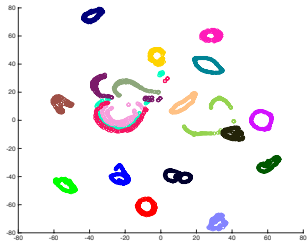
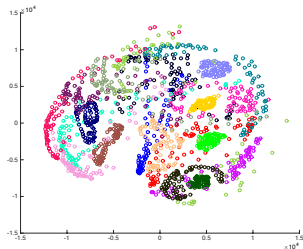


# COIL 20

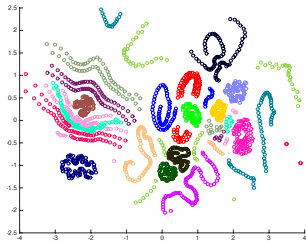
PCA



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
  - ▶ Autoencoder neural networks
  - ▶ Kohonen's maps
  - ▶ Laplacian Eigenmaps
  - ▶ Curvilinear distance analysis (CDA, Lee et al. 2004)
  - ▶ Semidefinite Embedding (SDE, Weinberger and Saul 2004)
  - ▶ Maximum Variance Unfolding (MVU)
  - ▶ Weighted t-SNE [research.cs.aalto.fi/pml/software/ne/](http://research.cs.aalto.fi/pml/software/ne/)
  - ▶ Metric learning
  - ▶ ...
- 3 to play with: [doc.gold.ac.uk/~lfedd001/three/demo.html](http://doc.gold.ac.uk/~lfedd001/three/demo.html)