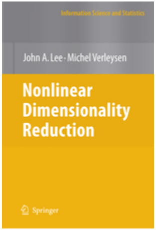
Visualisation et réduction de dimension

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



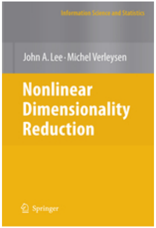
MLA

9 décembre 2024

The 3 main kinds of machine learning

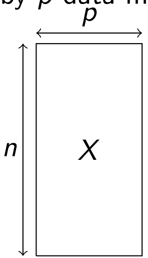
Lecture road map

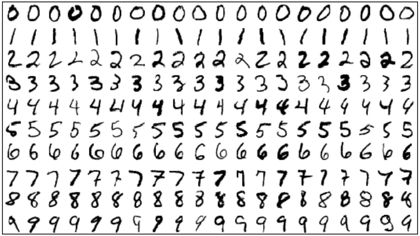
1. Introduction to hidden variables
2. PCA: principal component analysis
3. Distance preservation approaches (global methods)
4. Local approaches

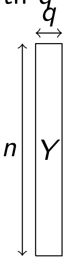


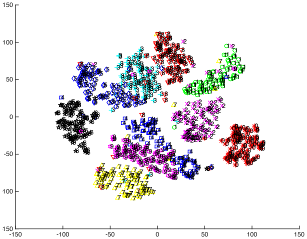
A dimensionality reduction problem

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









$p = 784$

$q = 2$

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 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

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

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

The curse of dimensionality

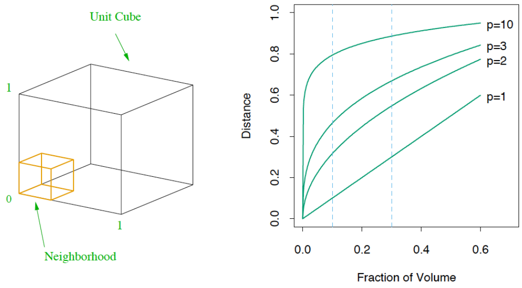


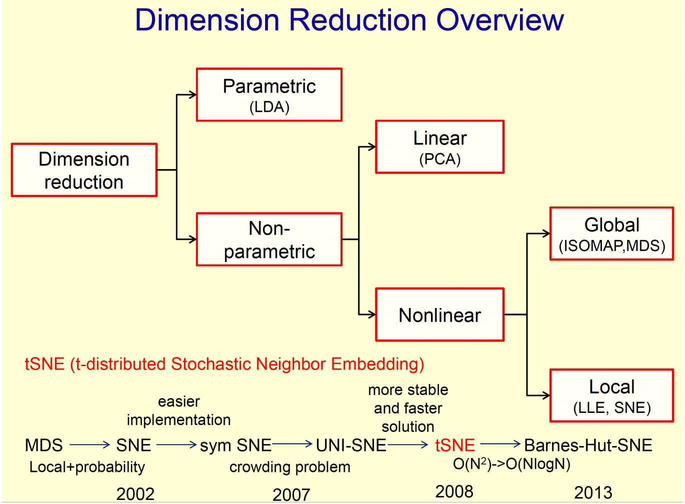
FIGURE 2.6. *The curse of dimensionality is well illustrated by a subcubical neighborhood for uniform data in a unit cube. The figure on the right shows the side-length of the subcube needed to capture a fraction r of the volume of the data, for different dimensions p. In ten dimensions we need to cover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

In large dimension, intuitions we have 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

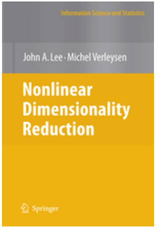
parametric (= supervised) vs. non parametric (= unsupervised)
linear vs. non linear
global metric vs. local metric



Visualization using tSNE, Yan Xu

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

$$X = YV^\top + B$$

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

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

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

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

proof

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

3 different ways to get Y

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

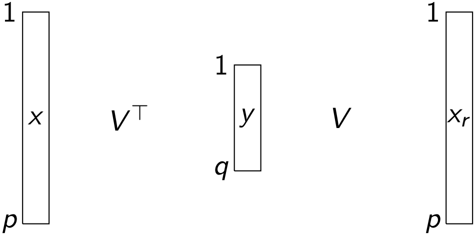
PCA as an encoding/decoding mechanism

Coding/decoding functions

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

Let V be the $p \times q$ matrix

$$cod(x) = V^\top x = y \qquad\qquad dec(y) = Vy = V^\top V x$$

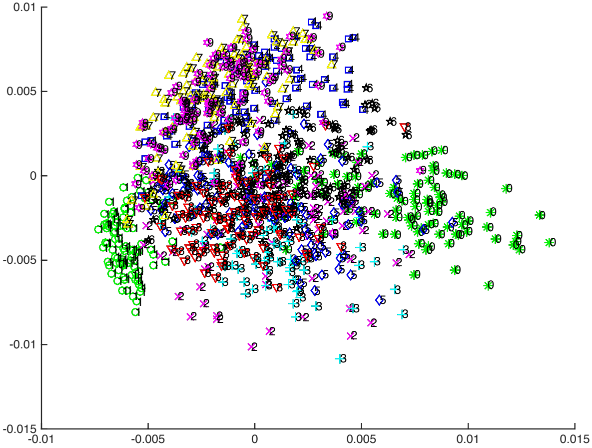


$cod(x)$ $\qquad$ $dec(y)$

2d PCA on the MNIST data

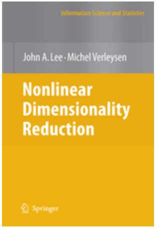
$p = 784$ $q = 2$

$Y =$



Lecture road map

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



Distances

- symmetric : $d(s,t) = d(t,s)$
- 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

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

hypermetrics and quasi distances
distances and probabilities
Gram matrix

$$G = XX^\top \qquad \text{avec} \qquad G_{ij} = \mathbf{x}_i^\top \mathbf{x}_j = 1 - \frac{1}{2} d(\mathbf{x}_i, \mathbf{x}_j)$$

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 (Carrol et Chang, 1970)
- ...

Classical MDS

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

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

1. given the distance matrix D_X
   - Y columns are the eigen vectors of $HD_X H$ 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,

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

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

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

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

Reinforce closed neigbors (…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).

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

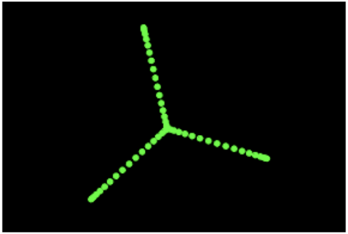
Sammon's projection Quasi Newton

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

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

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

http://hisee.sourceforge.net/about.html
http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV0910/henderson.pdf

2 issues with Sammon's projection

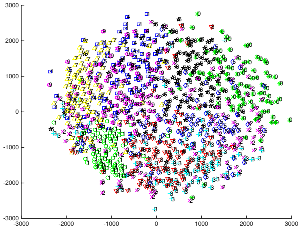
- $d = 0 \rightarrow w = \infty$ because

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

  the criterion preserve all small distances.
  Alternatives:

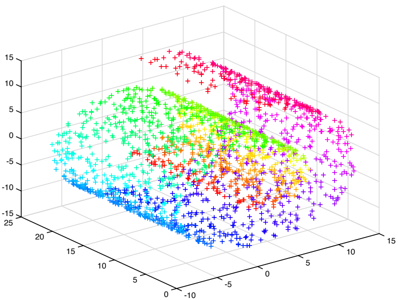
$$w_{i,j} = \begin{cases} 1 \text{ si } \|x_i - x_j\| \leq \varepsilon \\ 0 \text{ sinon} \end{cases}$$

- Y are uniformly distributed in a circle

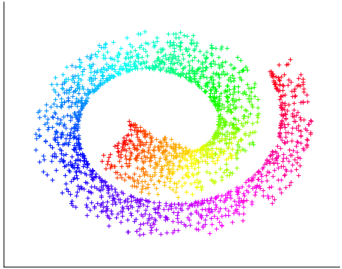


an example of MDS limitation

X



Y

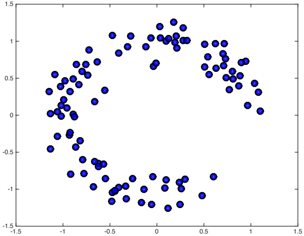


Solution: *metric learning* of $d(i,j)$

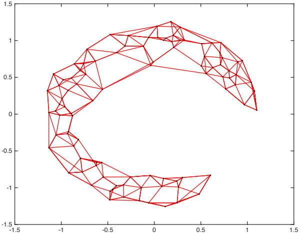
$$d_X(i,j) \text{ euclidean} \quad \rightarrow \quad d_g(i,j) \text{ geodesic}$$

Example of geodesic distance

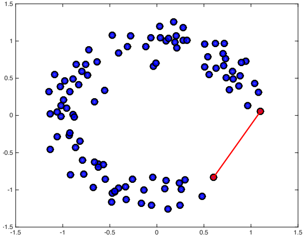
cloud of points



proximity graph

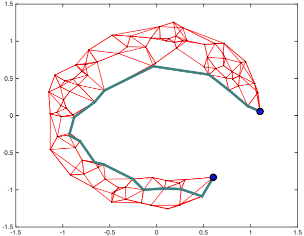


euclidean distance



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

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

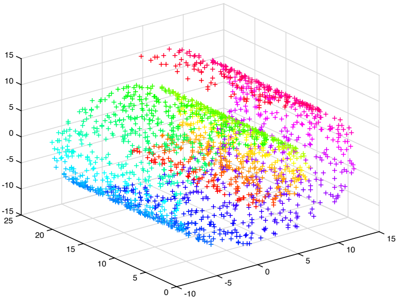
3. compute Y with MDS (or Sammon's proj.) using d_G instread of d_X,

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

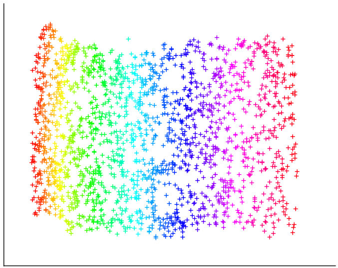
http://web.mit.edu/cocosci/isomap/isomap.html https://github.com/gpeyre/matlab-toolboxes/tree/master/toolbox_dimreduc
A Global Geometric Framework for Nonlinear Dimensionality Reduction J. B. Tenenbaum, et al. Science 290, 2000

Isometric Feature Mapping (ISOMAP)

X

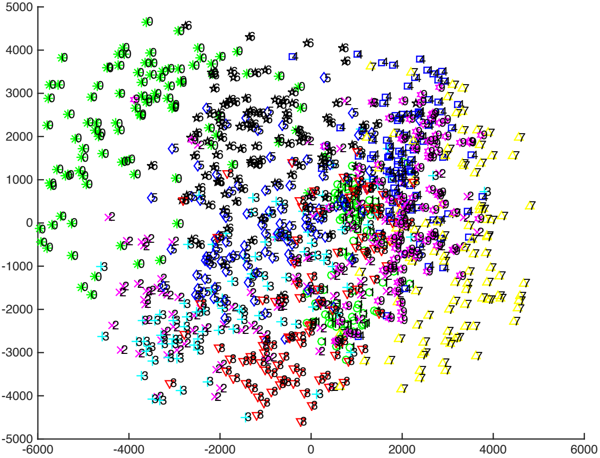


Y



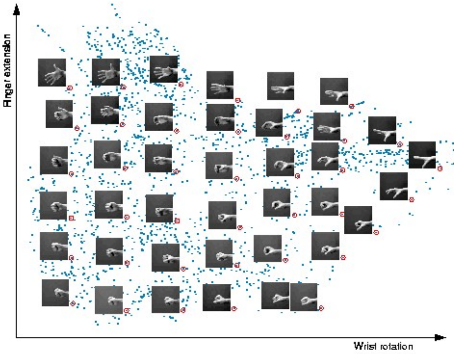
http://www.numerical-tours.com/matlab/shapes_7_isomap/

ISOMAP on MNIST



Problem with ISOMAP

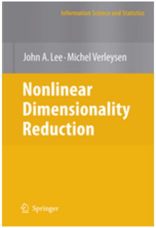
- its a global non sparse method
- doesn't scale $\mathcal{O}(n^3)$
- given a new Y the decoding function is not known X.
  The pre image problem



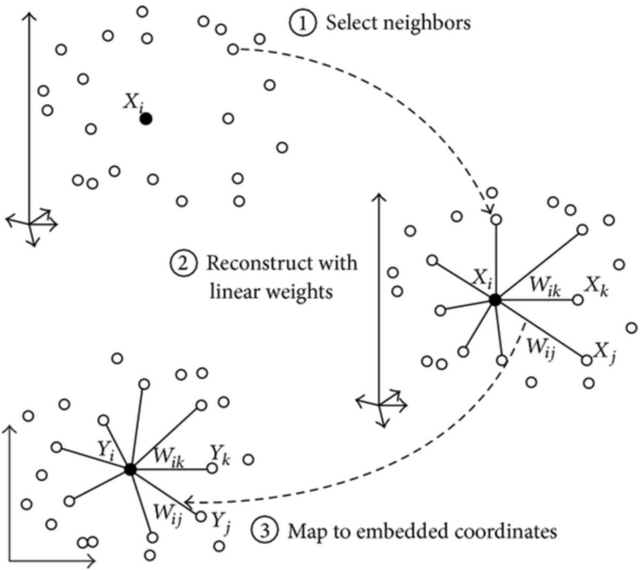
Isomap ($k = 6$) applied to $n = 2000$ images (64 pixels by 64 pixels) of a hand in different configurations. The images were generated by making a series of opening and closing movements of the hand at different wrist orientations, designed to give rise to a two-dimensional manifold.
http://web.mit.edu/cocosci/isomap/handfig.html

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

Locally Linear Embedding (LLE)

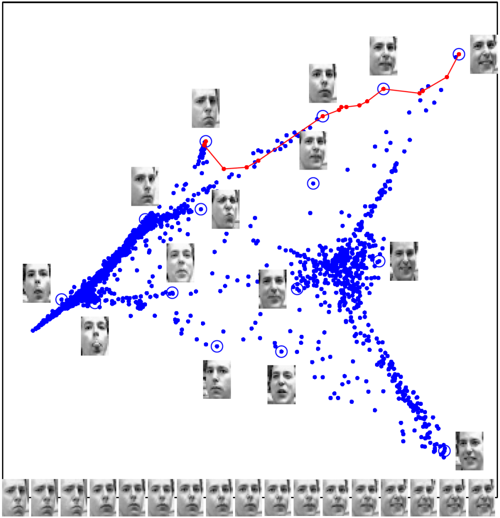
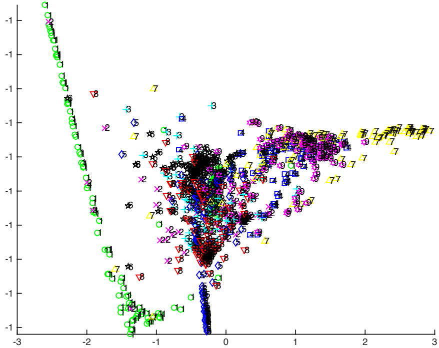


FIGURE 14.45. *Images of faces mapped into the embedding space described by the first two coordinates of LLE. Next to the circled points, representative faces are shown in different parts of the space. The images at the bottom of the plot correspond to points along the top right path (linked by solid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

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

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

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

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 $\mathbb{P}_X(j|i) = \mathbb{P}_X(i|j)$

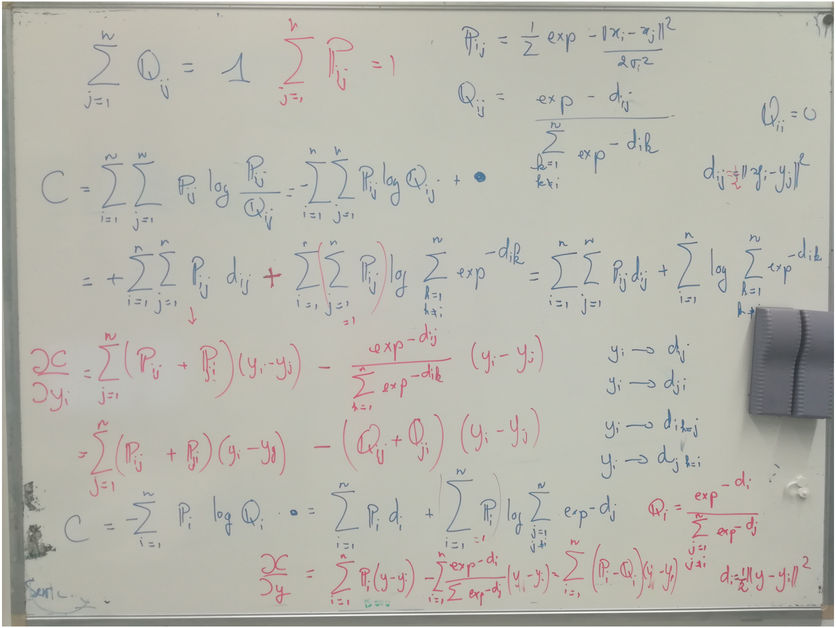
$$\frac{\mathbb{P}_X(j|i) + \mathbb{P}_X(i|j)}{2}$$

in that case:

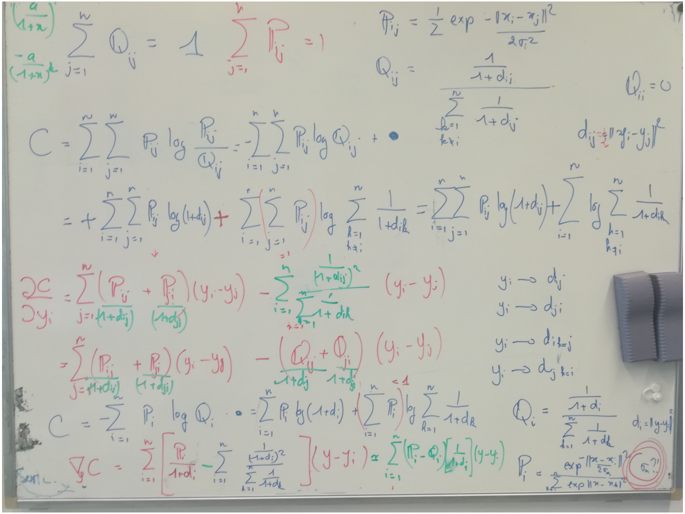
$$\nabla_{Y(i)} KL\big(\mathbb{P}_X || \mathbb{P}_Y\big) = 2 \sum_{j=1}^{n} \underbrace{(\mathbf{y}_i - \mathbf{y}_j)}_{\text{similarity}} \big(\underbrace{\mathbb{P}_X(j|i) - \mathbb{P}_Y(j|i)}_{\text{rigidity}}\big)$$

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

Derivation of the SNE gradient

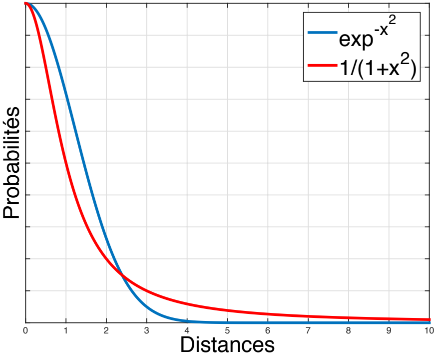


Derivation of the tSNE gradient



t-Stochastic Neighbor Embedding (t-SNE)

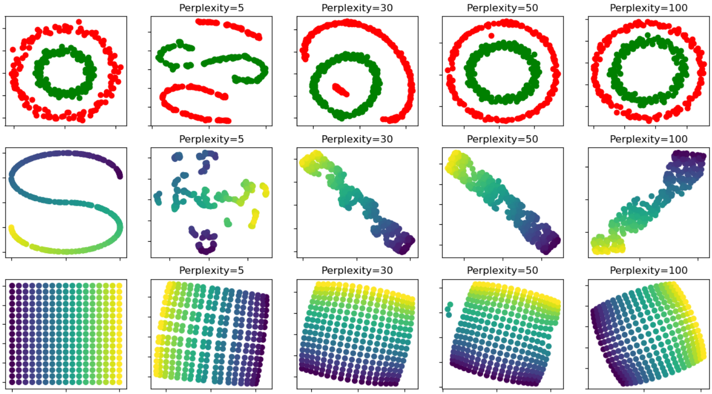
$$\mathbb{P}_Y(y_j|y_i) = \frac{1}{Z} \frac{1}{1 + \|y - y_i\|^2}$$



- $\mathbb{P}_X = \mathbb{P}_Y$ 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 practical optimization

With

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

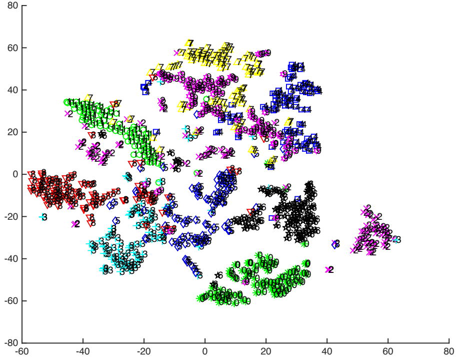
The gradient components are

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

With an exageration factor $\rho = 12$

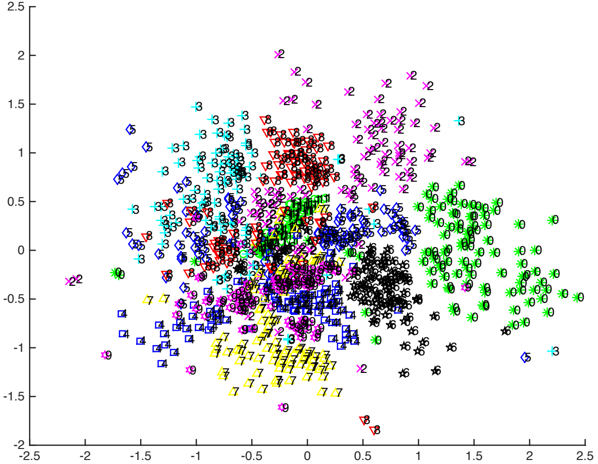
$$\frac{\partial KL_\rho\big(\mathbb{P}_X||\mathbb{P}_Y\big)}{\partial y_i} = 2\sum_{j=1}^{n} v_{ij} w_{ij}(\mathbf{y}_i - \mathbf{y}_j) - 2\frac{n}{Z\rho}\sum_{j=1}^{n} w_{ij}^2(\mathbf{y}_i - \mathbf{y}_j)$$

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.

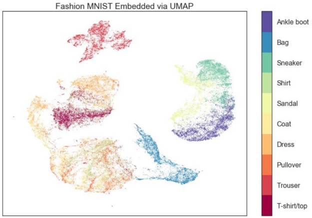
http://dx.doi.org/10.1016/j.neucom.2014.12.095

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



11 199 513

What the UMAP algorithm actually does

$$d_x(x_i, x_j) = \|x_i - x_j\|^2 \qquad d_y(y_i, y_j) = \|y_i - y_j\|^2$$

$$v_i(x_i, x_j) = \exp^{\frac{-d_x(x_i,x_j)+\mathbf{m}_i}{\sigma_i}} \qquad w_i(x_i, x_j) = \exp^{\frac{-d_y(y_i,y_j)+\mathbf{m}_i}{\sigma_i}}$$

1. estimate $\mathbf{m}_i$ and σ_i using k neighbors of point x_i
   - $\mathbf{m}_i$ the distance between x_i and its nearest neigbor
   - σ_i the diameter of the neighborhood of x_i
2. symetrize v and w

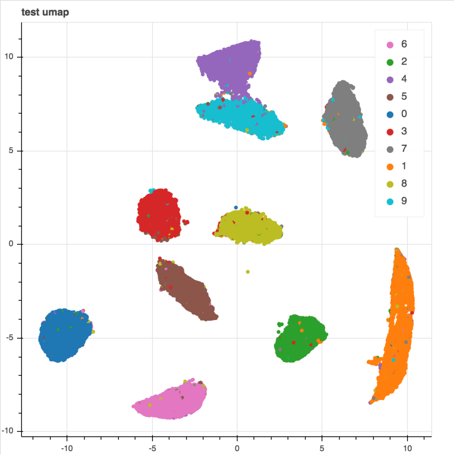
$$v_s(i,j) = v_i(x_i, x_j) + v_j(x_i, x_j) - v_i(x_i, x_j)v_j(x_i, x_j)$$

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

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

https://www.math.upenn.edu/~jhansen/2018/05/04/UMAP/
https://umap-learn.readthedocs.io/en/latest/how_umap_works.html

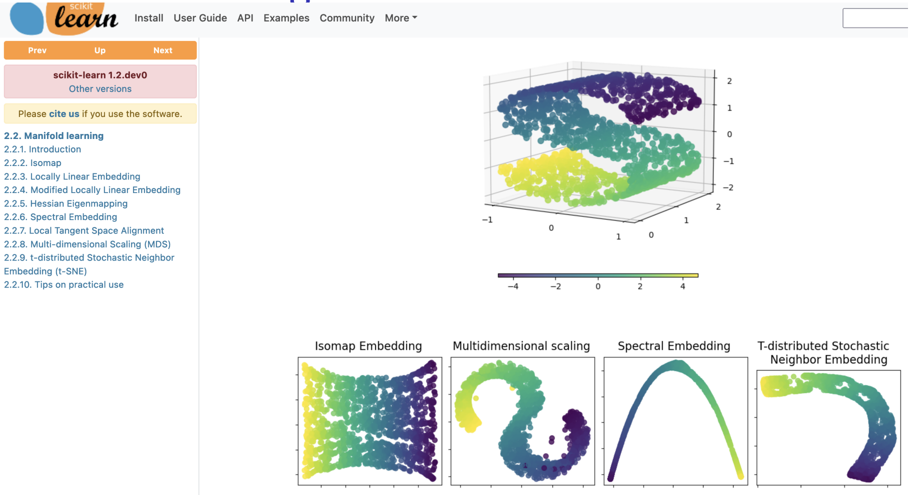
Uniform manifold approx. & projection (UMAP)



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

`https://github.com/lmcinnes/umap`

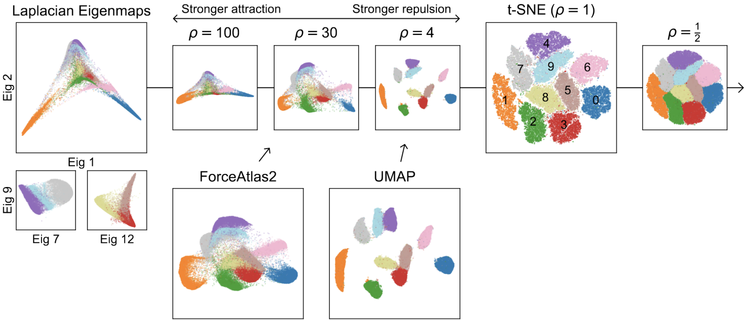
Manifold learning with Scikit-learn



New ones:

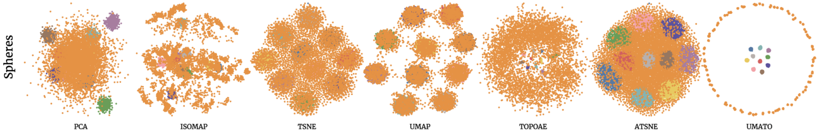
- 2.2.5. Hessian Eigenmapping,
- 2.2.6. Spectral Embedding (Laplacian Eigenmaps),
- 2.2.7. Local Tangent Space Alignment

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

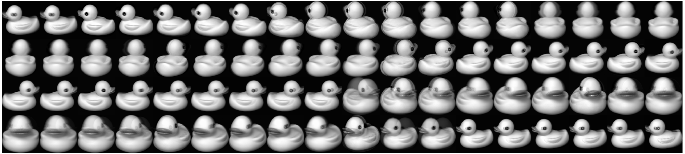


Uniform Manifold Approximation with Two-phase Optimization, Ko et al, 2021

https://github.com/hyungkwonko/umato

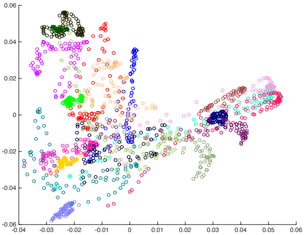
COIL 20

- 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

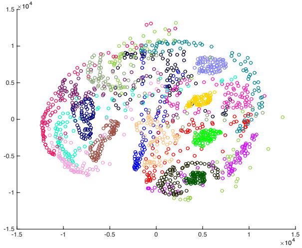


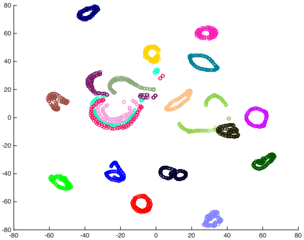
COIL 20

PCA

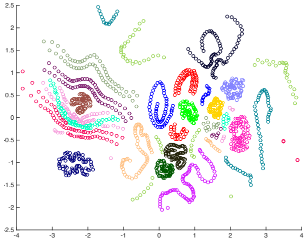


Sammon's projection (MDS)





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/`
   - Metric learning
   - ...

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