MITx: Statistics, Computation & Applications

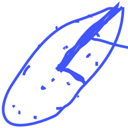
Genomics and High-Dimensional Data Module

Lecture 1: Visualization of Hig-Dimensional Data

$x^{(1)}, \dots, x^{(n)} \in \mathbb{R}^p \longrightarrow y^{(1)}, \dots, y^{(n)} \in \mathbb{R}^q$

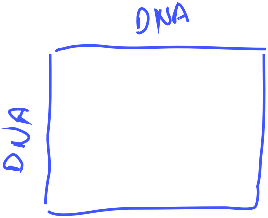
$n \approx 100'000 \qquad p \approx 20'000 \qquad q = 2, 3$

Σ : covariance matrix



eigenvector

$\boxed{\text{PCA}}$



contact frequency matrix $\longrightarrow$ Distance matrix $D \in \mathbb{R}^{n \times n}$

$y^{(1)} \dots y^{(n)} \in \mathbb{R}^3$

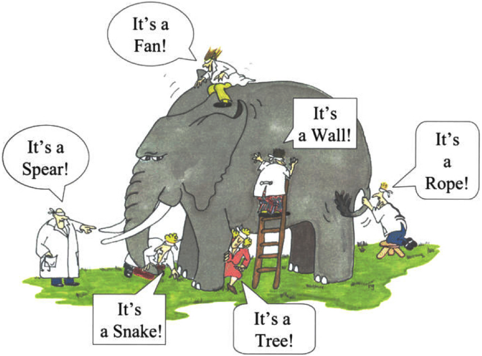
$$\underset{y^{(i)} \in \mathbb{R}^3}{\text{minimize}} \sum_{i \neq j} \left(D_{ij}^2 - \|y^{(i)} - y^{(j)}\|_2^2 \right)^2$$

$\boxed{\text{MDS}}$

ESNE

3 different approaches

- Principle component analysis: projection that spreads data as much as possible
- Multidimensional scaling: projection that retains original distances as much as possible
- Stochastic neighbor embedding: non-linear embedding that tries to keep close-by points close



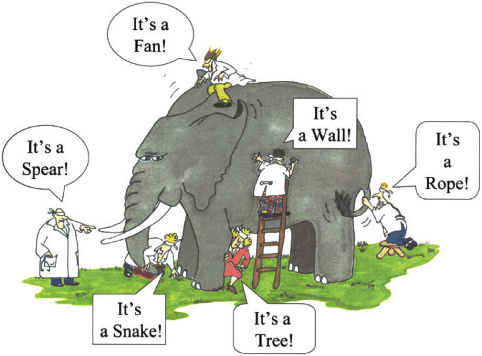
MITx: Statistics, Computation & Applications

Genomics and High-Dimensional Data Module

Lecture 1: Visualization of Hig-Dimensional Data

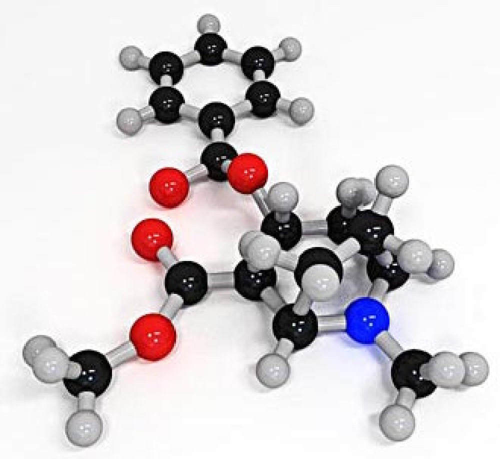
3 different approaches

- Principle component analysis: projection that spreads data as much as possible
- Multidimensional scaling: projection that retains original distances as much as possible
- Stochastic neighbor embedding: non-linear embedding that tries to keep close-by points close



Principle Component Analysis

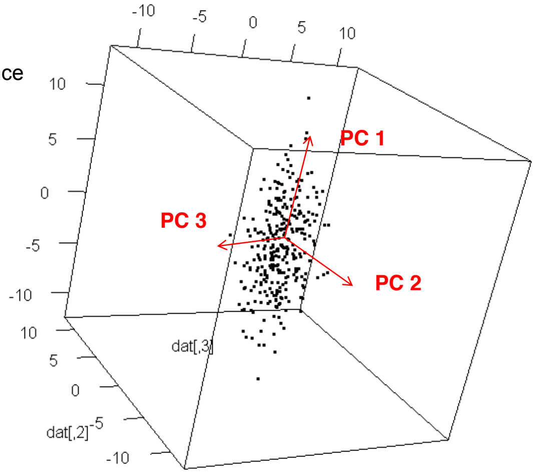
- **Goal:** Dimension reduction to a few dimensions
- **Intuition:** Find low-dimensional projection with largest spread



Definition 1: Maximize projection variance

Start with centered data $X \in \mathbb{R}^{n \times p}$

- PC 1 is direction of largest variance
- PC 2 is
  - perpendicular to PC 1
  - again largest variance
- PC 3 is
  - perpendicular to PC 1, PC 2
  - again largest variance
- etc.



Definition 2: Minimize projection residuals

- PC 1: Straight line with smallest orthogonal distance to all points
- PC 1 & PC 2: Plane with with smallest orthogonal distance to all points
- etc.

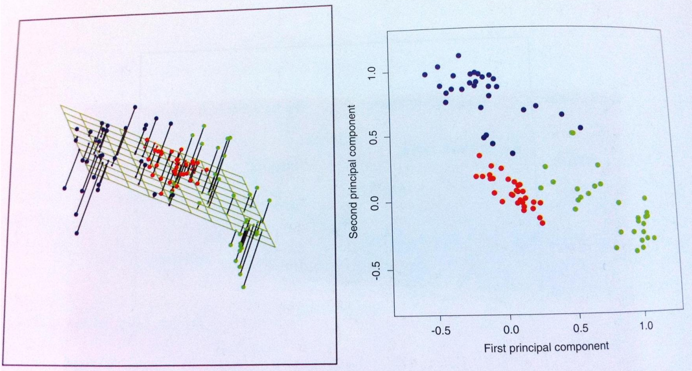


Figure from *Elements of Statistical Learning*by Hastie and Tibshirani

Definition 3: Spectral decomposition

- Covariance matrix (or correlation matrix) $R = \frac{1}{n}X^T X$ is symmetric and positive semidefinite

  $x^T R x = \lambda$

- **Spectral Decomposition Theorem:** Every real symmetric matrix R can be decomposed as

  $$R = V\Lambda V^T,$$

  where Λ is diagonal and V is orthogonal

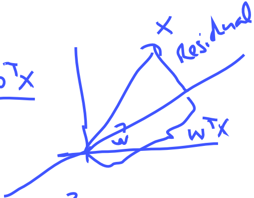
  $V = \begin{bmatrix} v_1 \dots v_p \end{bmatrix}$

  $\Lambda \begin{bmatrix} \lambda_1 & \dots & 0 \\ 0 & \dots & \lambda_p \end{bmatrix}$

- Columns of V (= eigenvectors of R) are the PCs

- Diagonal entries of Λ (= eigenvalues of R) are variances along PCs w: unit vector

length of projection of x onto w: $\underline{w^Tx}$



residuals (squared):

$$\|x-(w^Tx)w\|_2^2 = \|x\|_2^2 - 2(w^Tx)^2 + (w^Tx)^2 \underbrace{w^Tw}_{=1}$$

$$= \|x\|_2^2 - (w^Tx)^2$$

$$\underset{w\in\mathbb{R}^p,\ \|w\|=1}{\text{minimize}} \left(\underbrace{\sum_{i=1}^n \|x_i\|^2}_{\text{const}} - (w^Tx_i)^2 \right)$$

minimize residuals

$$\iff \underset{w\in\mathbb{R}^p,\ \|w\|=1}{\text{maximize}} \sum_{i=1}^n (w^Tx_i)^2$$

maximize variance

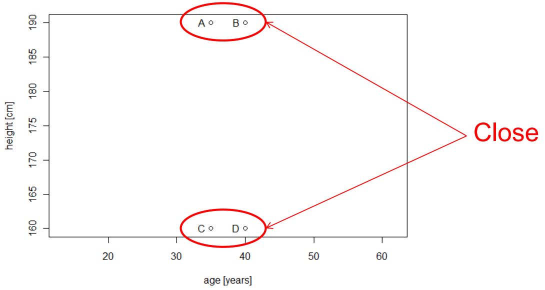
$$\iff \underset{w\in\mathbb{R}^p,\ \|w\|=1}{\text{maximize}} \; w^T \underbrace{\left(\frac{1}{n} \sum_{i=1}^n x_i x_i^T \right)}_{\text{Cov}(X)} w$$

→ eigenvector corresponding to largest eigenvalue

Covariance versus correlation - to scale or not to scale

- Using covariance will find the variable with largest spread as 1. PC
- Use correlation, if different units are compared

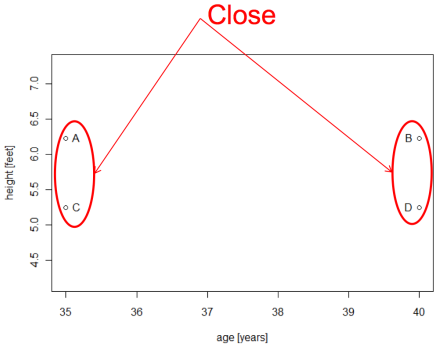
Person	Age (years)	Height (cm)
A	35	190
B	40	190
C	35	160
D	40	160



Covariance versus correlation - to scale or not to scale

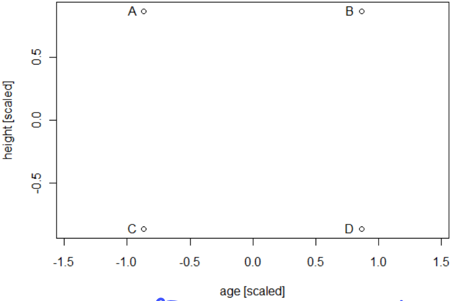
- Using covariance will find the variable with largest spread as 1. PC
- Use correlation, if different units are compared

Person	Age (years)	Height (feet)
A	35	6.232
B	40	6.232
C	35	5.248
D	40	5.248



- Using covariance will find the variable with largest spread as 1. PC
- Use correlation, if different units are compared

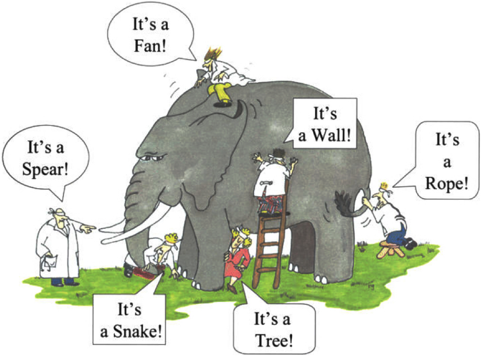
Person	Age (years)	Height (feet)
A	-0.87	0.87
B	0.87	0.87
C	-0.87	-0.87
D	0.87	-0.87



$\sum_{i=1}^{p} \lambda_i = \text{trace} = p$ keep PCs with $\lambda_i \geq 1$.

3 different approaches

- Principle component analysis: projection that spreads data as much as possible
- Multidimensional scaling: projection that retains original distances as much as possible
- Stochastic neighbor embedding: non-linear embedding that tries to keep close-by points close



Distance and dissimilarity

- $D \in \mathbb{R}^{n \times n}$ is a distance matrix if

$$D_{ii} = 0, \quad D_{ij} \geq 0, \quad D_{ij} = D_{ji}, \quad D_{ij} \leq D_{ik} + D_{jk} \quad \text{for all } i, j, k$$

  - **Ex:** Euclidean distance, Manhattan distance, maximum distance, ...

- $D \in \mathbb{R}^{n \times n}$ is a dissimilarity matrix if

$$D_{ii} = 0, \quad D_{ij} \geq 0, \quad D_{ij} = D_{ji} \quad \text{for all } i, j, k$$

  - More flexible than distances, works e.g. for rankings

Multidimensional scaling (MDS)

Given a matrix $D \in \mathbb{R}^{n \times n}$, determine points $y_1, \ldots, y_n \in \mathbb{R}^q$ such that:

$q = 2$ or 3

- Classical MDS: minimize $\sum_{i=1}^{n} \sum_{j=1}^{n} (D_{ij} - \|y_i - y_j\|_2)^2$

  assuming D is a Euclidean distance matrix

Multidimensional scaling (MDS)

Given a matrix $D \in \mathbb{R}^{n \times n}$, determine points $y_1, \ldots, y_n \in \mathbb{R}^q$ such that:

- Classical MDS: minimize $\sum_{i=1}^n \sum_{j=1}^n (D_{ij} - \|y_i - y_j\|_2)^2$
  assuming D is a Euclidean distance matrix
- Weighted MDS: minimize $\sum_{i=1}^n \sum_{j=1}^n w_{ij}(D_{ij} - \|y_i - y_j\|_2)^2$
  assuming D is a distance matrix and w_{ij} are non-negative weights
  - solved iteratively using stress majorization

Multidimensional scaling (MDS)

Given a matrix $D \in \mathbb{R}^{n\times n}$, determine points $y_1, \ldots, y_n \in \mathbb{R}^q$ such that:

- Classical MDS: minimize $\sum_{i=1}^n \sum_{j=1}^n (D_{ij} - \|y_i - y_j\|_2)^2$
  assuming D is a Euclidean distance matrix
- Weighted MDS: minimize $\sum_{i=1}^n \sum_{j=1}^n w_{ij}(D_{ij} - \|y_i - y_j\|_2)^2$
  assuming D is a distance matrix and w_{ij} are non-negative weights
  - solved iteratively using stress majorization
- Non-metric MDS: minimize $\sum_{i=1}^n \sum_{j=1}^n (\theta(D_{ij}) - \|y_i - y_j\|_2)^2$
  assuming D is a dissimilarity matrix
  - also optimize over increasing function θ
  - finds low-dimensional embedding that respects ranking of dissimilarities
  - solved numerically (isotonic regression); very time-consuming

Classical MDS

- First convert a distance matrix D, with $D_{ij} = \|x_i - x_j\|_2$ into a positive semidefinite matrix XX^T, namely

$$XX^T = -\frac{1}{2}(I - \frac{1}{n}ee^t)D^2(I - \frac{1}{n}ee^t), \quad \text{where } e \text{ is vector of ones}$$

- **Note:** $(XX^T)_{ij} = -\frac{1}{2}(D_{ij}^2 - D_{i\cdot}^2 - D_{\cdot j}^2 + D_{\cdot\cdot}^2)$ (doubly centered matrix)

$$X = \begin{bmatrix} - x_1 - \\ \vdots \\ - x_n - \end{bmatrix} \in \mathbb{R}^{n \times p} \qquad Y = \begin{bmatrix} - y_1 - \\ \vdots \\ - y_n - \end{bmatrix} \in \mathbb{R}^{q}$$

$$D_{ij}^2 = \|x_i\|^2 + \|x_j\|^2 - 2x_i^T x_j$$
$$= (XX^T)_{ii} + (XX^T)_{jj} - 2(XX^T)_{ij}$$

Classical MDS

- First convert a distance matrix D, with $D_{ij} = \|x_i - x_j\|_2$ into a positive semidefinite matrix XX^T, namely

$$XX^T = -\frac{1}{2}(I - \frac{1}{n}ee^t)D^2(I - \frac{1}{n}ee^t), \quad \text{where } e \text{ is vector of ones}$$

- **Note:** $(XX^T)_{ij} = -\frac{1}{2}(D_{ij}^2 - D_{i\cdot}^2 - D_{\cdot j}^2 + D_{\cdot\cdot}^2)$ (doubly centered matrix)

- $\min_Y \ \sum_{i=1}^n \sum_{j=1}^n (D_{ij}^2 - \|y_i - y_j\|_2^2)^2$ is equivalent to

$$\min_{Y} \ \text{trace}(XX^T - YY^T)^2$$

rk p matrix

rk q matrix.

q < p

Classical MDS

- First convert a distance matrix D, with $D_{ij} = \|x_i - x_j\|_2$ into a positive semidefinite matrix XX^T, namely

$$XX^T = -\frac{1}{2}(I - \frac{1}{n}ee^t)D^2(I - \frac{1}{n}ee^t), \quad \text{where } e \text{ is vector of ones}$$

- **Note:** $(XX^T)_{ij} = -\frac{1}{2}(D^2_{ij} - D^2_{i\cdot} - D^2_{\cdot j} + D^2_{\cdot\cdot})$ (doubly centered matrix)

- $\min_Y \ \sum_{i=1}^n \sum_{j=1}^n (D^2_{ij} - \|y_i - y_j\|_2^2)^2$ is equivalent to

$$\min_Y \ \operatorname{trace}(XX^T - YY^T)^2$$

- Eigenvalue decomposition: $XX^T = V\Lambda V^T$, where columns of V are eigenvectors of XX^T, Λ is diagonal containing eigenvalues of XX^T

Classical MDS

- First convert a distance matrix D, with $D_{ij} = \|x_i - x_j\|_2$ into a positive semidefinite matrix XX^T, namely

$$XX^T = -\frac{1}{2}(I - \frac{1}{n}ee^t)D^2(I - \frac{1}{n}ee^t), \quad \text{where } e \text{ is vector of ones}$$

- **Note:** $(XX^T)_{ij} = -\frac{1}{2}(D^2_{ij} - D^2_{i\cdot} - D^2_{\cdot j} + D^2_{\cdot\cdot})$ (doubly centered matrix)
- $\min_Y \sum_{i=1}^n \sum_{j=1}^n (D^2_{ij} - \|y_i - y_j\|_2^2)^2$ is equivalent to

$$\min_Y \ \text{trace}(XX^T - YY^T)^2$$

- Eigenvalue decomposition: $XX^T = V\Lambda V^T$, where columns of V are eigenvectors of XX^T, Λ is diagonal containing eigenvalues of XX^T
- Best rank q approximation of XX^T is given by choosing q largest eigenvalues and corresponding eigenvectors, i.e. $YY^T = V_1\Lambda_1 V_1^T$, or equivalently, $Y = V_1\Lambda_1^{1/2}$

$V_1 = \begin{bmatrix} v_1 & \cdots & v_q \end{bmatrix}$ $\Lambda_1 = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_q \end{bmatrix}$

Classical MDS

- First convert a distance matrix D, with $D_{ij} = \|x_i - x_j\|_2$ into a positive semidefinite matrix XX^T, namely

$$XX^T = -\frac{1}{2}(I - \frac{1}{n}ee^t)D^2(I - \frac{1}{n}ee^t), \quad \text{where } e \text{ is vector of ones}$$

- **Note:** $(XX^T)_{ij} = -\frac{1}{2}(D_{ij}^2 - D_{i\cdot}^2 - D_{\cdot j}^2 + D_{\cdot\cdot}^2)$ (doubly centered matrix)
- $\min_Y \ \sum_{i=1}^n \sum_{j=1}^n (D_{ij}^2 - \|y_i - y_j\|_2^2)^2$ is equivalent to

$$\min_Y \ \mathrm{trace}(XX^T - YY^T)^2$$

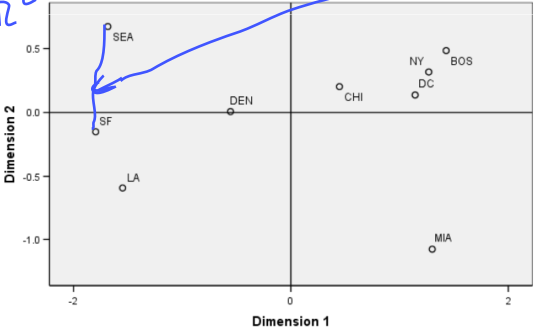
- Eigenvalue decomposition: $XX^T = V\Lambda V^T$, where columns of V are eigenvectors of XX^T, Λ is diagonal containing eigenvalues of XX^T
- Best rank q approximation of XX^T is given by choosing q largest eigenvalues and corresponding eigenvectors, i.e. $YY^T = V_1\Lambda_1 V_1^T$, or equivalently, $Y = V_1\Lambda_1^{1/2}$
- Classical MDS is PCA on $B = \overset{n\times n}{XX^T}$; classical PCA operates on $\overset{p\times p}{X^TX}$

MDS example: Distances between US cities

	BOS	CHI	DC	DEN	LA	MIA	NY	SEA	SF
BOS	0	963	429	1,949	2,979	1,504	206	2,976	3,095
CHI	963	0	671	996	2,054	1,329	802	2,013	2,142
DC	429	671	0	1,616	2,631	1,075	233	2,684	2,799
DEN	1,949	996	1,616	0	1,059	2,037	1,771	1,307	1,235
LA	2,979	2,054	2,631	1,059	0	2,687	2,786	1,131	379
MIA	1,504	1,329	1,075	2,037	2,687	0	1,308	3,273	3,053
NY	206	802	233	1,771	2,786	1,308	0	2,815	2,934
SEA	2,976	2,013	2,684	1,307	1,131	3,273	2,815	0	808
SF	3,095	2,142	2,799	1,235	379	3,053	2,934	808	0

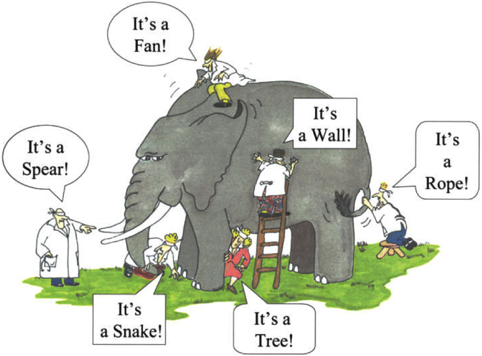
$n = 9,\ q = 2$

$y_1, \dots, y_9 \in \mathbb{R}^2$



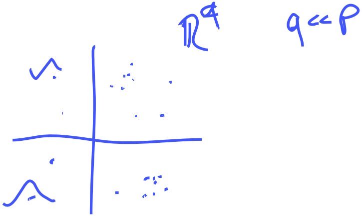
3 different approaches

- Principle component analysis: projection that spreads data as much as possible
- Multidimensional scaling: projection that retains original distances as much as possible
- Stochastic neighbor embedding: non-linear embedding that tries to keep close-by points close



Stochastic neighbor embedding (SNE)

- probabilistic approach to place objects from high-dimensional space into low-dimensional space so as to preserve the identity of neighbors
- center a Gaussian on each object in high-dimensional space
- find embedding so that resulting high-dimensional distribution is approximated well by resulting low-dimensional distribution

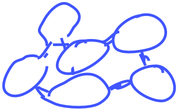


Stochastic neighbor embedding (SNE)

- probabilistic approach to place objects from high-dimensional space into low-dimensional space so as to preserve the identity of neighbors
- center a Gaussian on each object in high-dimensional space
- find embedding so that resulting high-dimensional distribution is approximated well by resulting low-dimensional distribution
- determine low-dimensional distribution by minimizing Kullback-Leibler divergence $KL(p \| q) := \sum_i p_i \log\left(\frac{p_i}{q_i}\right)$

Stochastic neighbor embedding (SNE)

- probabilistic approach to place objects from high-dimensional space into low-dimensional space so as to preserve the identity of neighbors
- center a Gaussian on each object in high-dimensional space
- find embedding so that resulting high-dimensional distribution is approximated well by resulting low-dimensional distribution
- determine low-dimensional distribution by minimizing Kullback-Leibler divergence
- allows ambiguous objects like “bank”, to be close to “river” and “finance” without forcing all outdoor concepts to be located close to corporate concepts



(Symmetric) SNE

- given dissimilarity matrix D, for each object i compute probability of picking j as neighbor:

$$p_{ij} = \frac{\exp(-D_{ij}^2)}{\sum_{k \neq \ell} \exp(-D_{k\ell}^2)}$$

- in low-dimensional space, for each point y_i compute probability of picking y_j as neighbor:

$$q_{ij} = \frac{\exp(-\|y_i - y_j\|_2^2)}{\sum_{k \neq \ell} \exp(-\|y_k - y_\ell\|_2^2)}$$

- Minimize the KL-divergence

$$\mathrm{KL}(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

D_{ij} small $\Rightarrow p_{ij}$ large. if q_{ij} small $\Rightarrow$ KL-div large
D_{ij} large $\Rightarrow p_{ij}$ small; if q_{ij} large $\Rightarrow$ negative KL-div

(Symmetric) SNE

- given dissimilarity matrix D, for each object i compute probability of picking j as neighbor:

$$p_{ij} = \frac{\exp(-D_{ij}^2)}{\sum_{k \neq \ell} \exp(-D_{k\ell}^2)}$$

- in low-dimensional space, for each point y_i compute probability of picking y_j as neighbor:

$$q_{ij} = \frac{\exp(-\|y_i - y_j\|_2^2)}{\sum_{k \neq \ell} \exp(-\|y_k - y_\ell\|_2^2)}$$

- Minimize the KL-divergence

$$\mathrm{KL}(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

- by modeling p_{ij} by $q_{ij} = p_{ij} + x$ you gain less than you lose by choosing $q_{ij} = p_{ij} - x$
- keeps nearby objects nearby and separated objects relatively far

tSNE

- SNE (non-convex) is optimized using gradient descent from an initial configuration

tSNE

- SNE (non-convex) is optimized using gradient descent from an initial configuration
- problem with many embedding methods: points often get crowded in the middle

tSNE

- SNE (non-convex) is optimized using gradient descent from an initial configuration

- problem with many embedding methods: points often get crowded in the middle

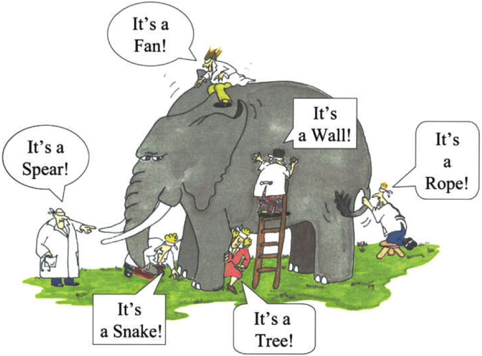
- t-SNE reduces this by using t-distribution with 1 degree of freedom for y's:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|_2^2)^{-1}}{\sum_{k \neq \ell}(1 + \|y_i - y_j\|_2^2)^{-1}}$$

- reduces crowding: moderate distance in high-dim. space can be faithfully modeled by much larger distance in low-dim. space

3 different approaches

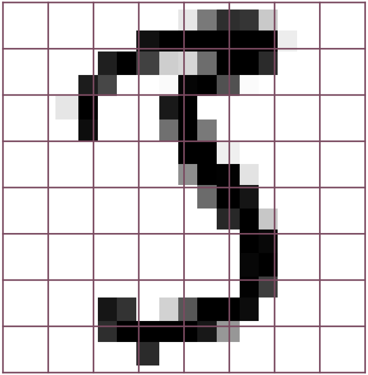
- Principle component analysis: projection that spreads data as much as possible
- Multidimensional scaling: projection that retains original distances as much as possible
- Stochastic neighbor embedding: non-linear embedding that tries to keep close-by points close



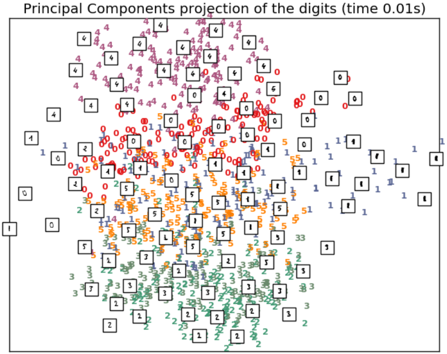
Example: Digit recognition

- $\sim$ 1800 hand-written digits (i.e., $n \approx 180$ for each class label)
- each (centered) digit was put in a 8×8-grid (i.e., $d = 64$)
- measure grey value in each part of the grid, i.e. 64 grey values

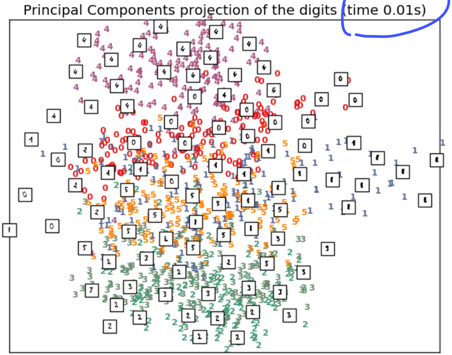
A selection from the 64-dimensional digits dataset

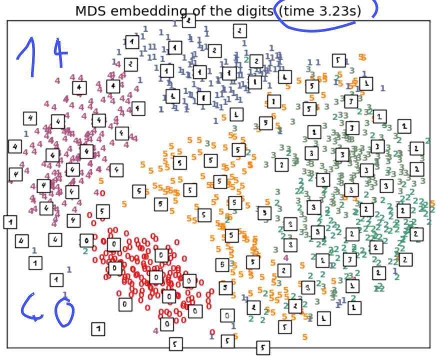


Example: Digit recognition



64x64 (6·180) x (6·180)

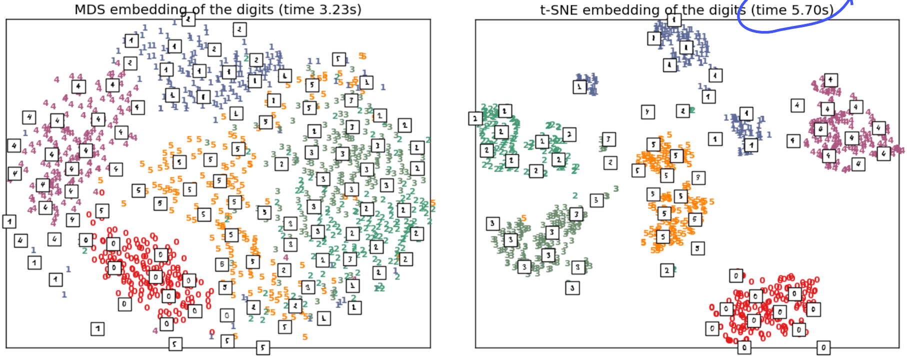




For code and figures see

`http:scikit-learn.orgstableauto_examplesmanifoldplot_lle_digits.html`

Example: Digit recognition

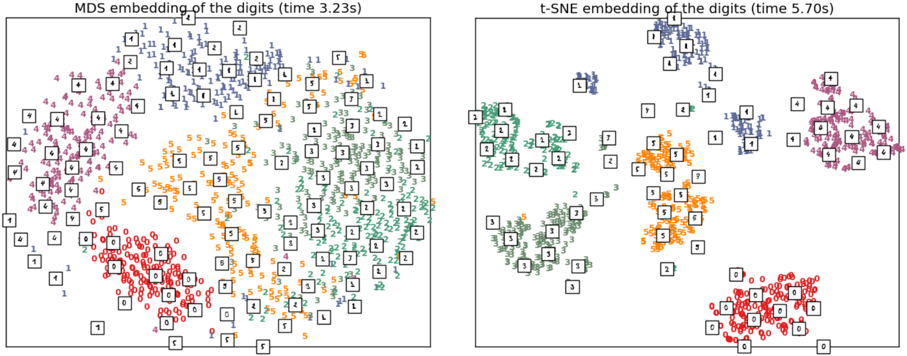


$x^{(1)}, \dots, x^{(n)} \in \mathbb{R}^p$

$c^{(1)}, \dots, c^{(n)} \in \mathcal{C}$ ← class labels

$x \in \mathbb{R}^p \xrightarrow{?} c \in \mathcal{C}$

Example: Digit recognition



- tSNE seems to find meaningful clusters
- But: This is the result of a non-convex optimization problem, which depends immensely on the starting configuration
- Axes of tSNE have NO meaning

References

- For PCA and MDS:
  - B. Everitt & T. Hothorn. *An Introduction to Applied Multivariate Analysis with R*. Springer, 2011.
  - T. Hastie, R. Tibshirani & J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.
- For tSNE:
  - L. van der Maaten & G. E. Hinton. *Visualizing Data using t-SNE*. JMLR, 2008.
  - G. E. Hinton & S. T. Roweis. *Stochastic Neighbor Embedding*. NIPS, 2002.