法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

关注 小象学院

Machine Learning

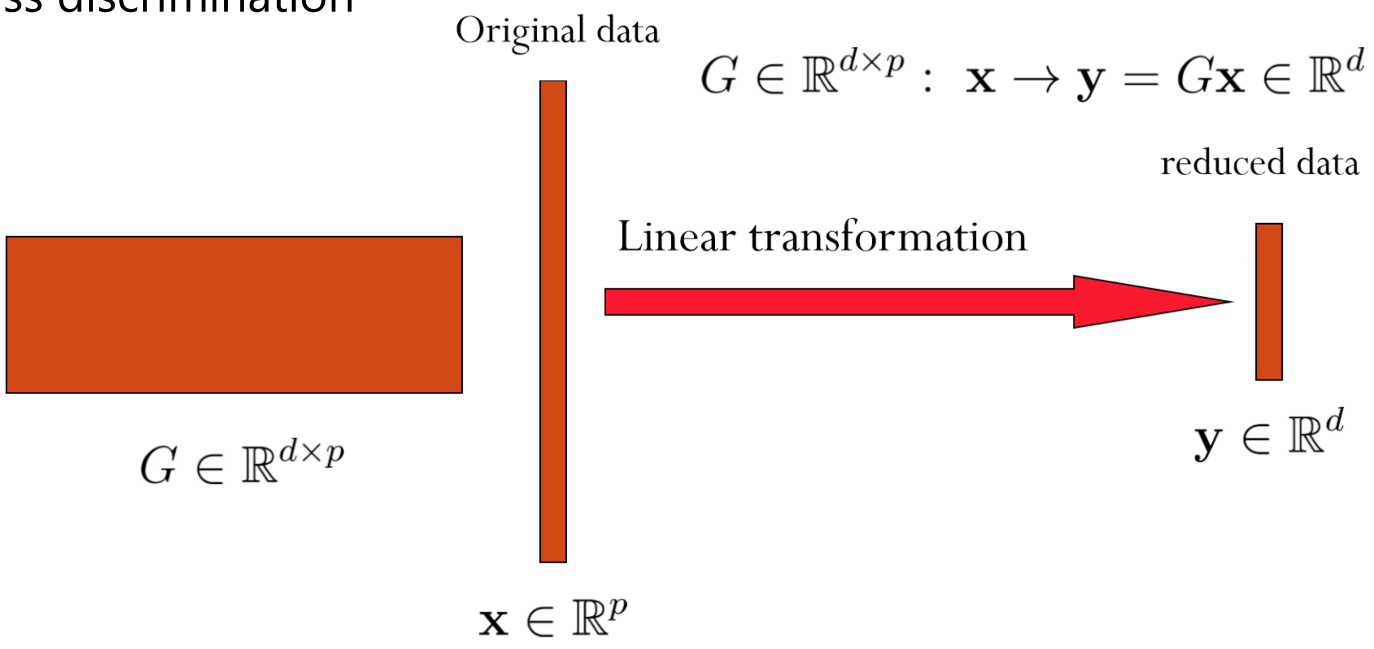
Part 6: Unsupervised Learning and Dimension Reduction

Dimension Reduction

Why Dimension Reduction

- Dimension reduction refers to the mapping of the original high-dim data onto a lower-dim space
- Criterion for dimension reduction can be different based on different problem settings

✓ Unsupervised setting: minimize the information loss
✓ Supervised setting: maximize the class discrimination



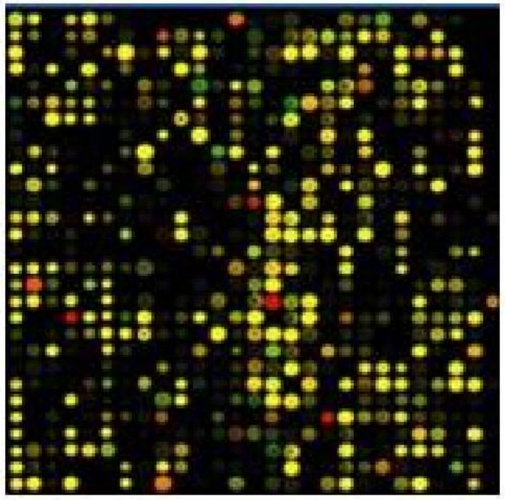
Why Dimension Reduction

Most machine learning and data mining techniques may not be effective for high-dimensional data

Curse of Dimensionality
Query accuracy and efficiency degrade rapidly as the dimension increases.

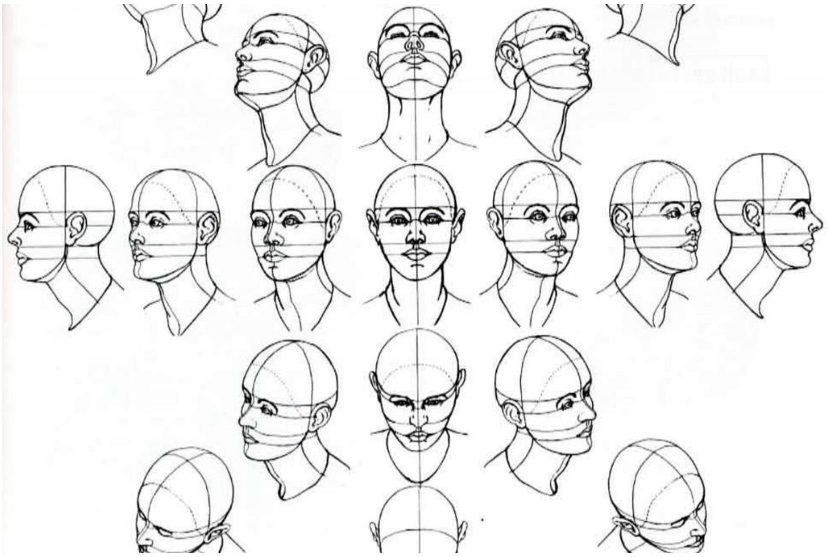
The **intrinsic** dimension may be small.
For example, the number of genes responsible for a certain type of disease may be small.

Visualization: projection of high-dimensional data onto 2D or 3D.
Data compression: efficient storage and retrieval.
Noise removal: positive effect on query accuracy.

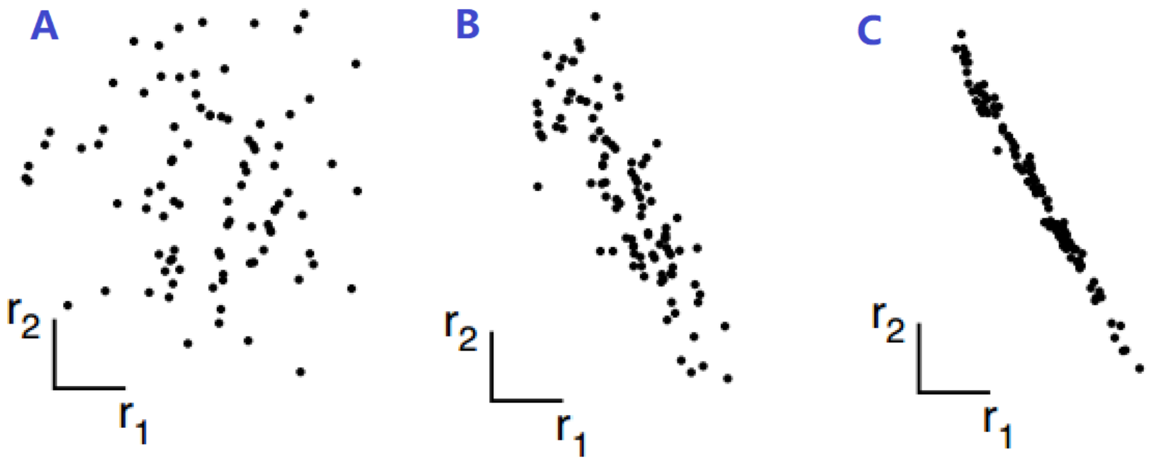


Redundancy

Representation: a high-dimensional vector (e.g., 20 x 28 = 560) where each dimension represents the brightness of one pixel.



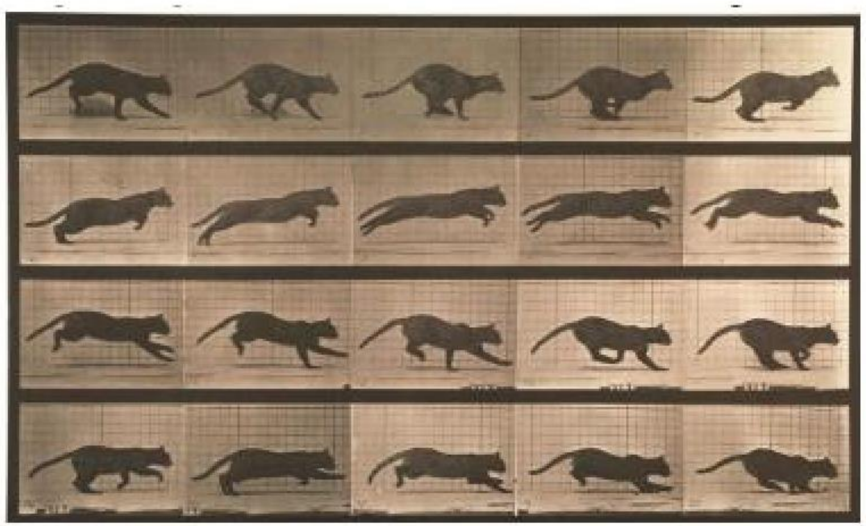
(**Face Recognition**) Underlying structure parameters: different camera angles, pose and lighting condition, face expression, etc.

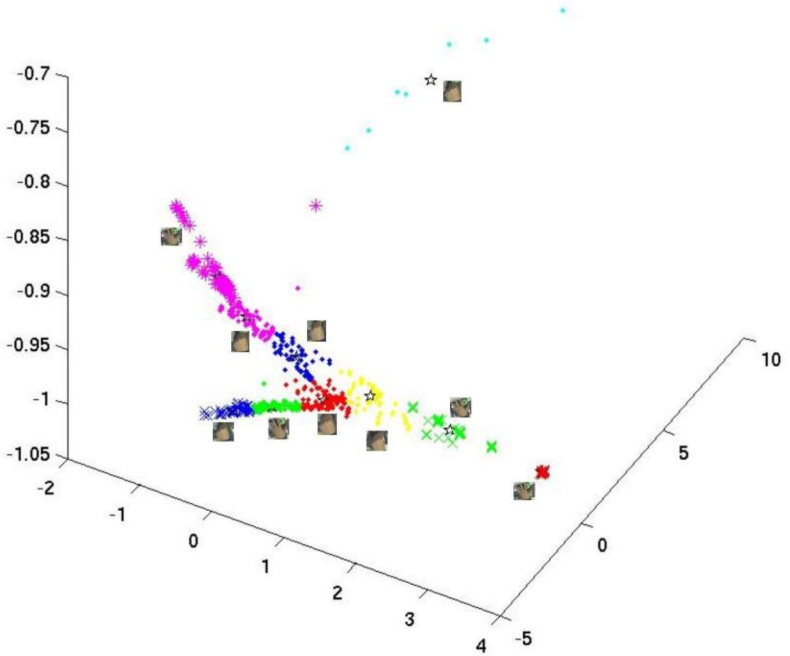


Highly redundant data are likely to be compressible -- essential idea for dimension reduction

Other Redundant Information

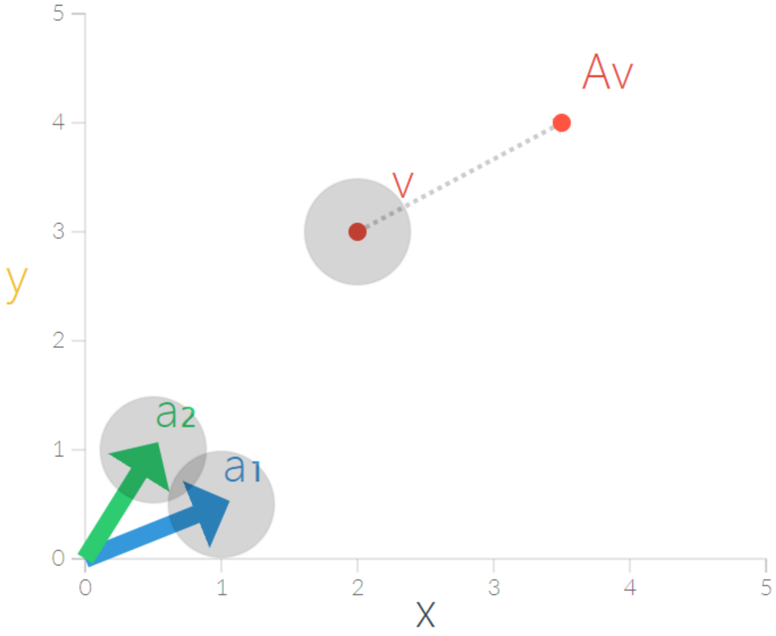
Representation: pose is determined, e.g., by the 3D coordinates of multiple points on the body





Underlying structure parameters: pose type
Motion can be viewed as a trajectory on the manifold

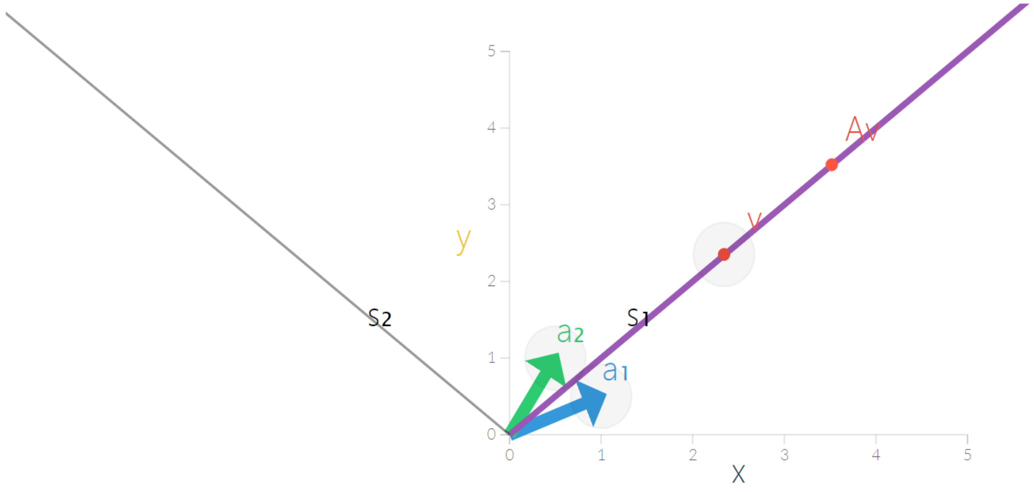
Eigenvectors and Eigenvalues



$$A = \begin{bmatrix} a_{1,x} & a_{2,x} \\ a_{1,y} & a_{2,y} \end{bmatrix} = \begin{bmatrix} 1.00 & 0.50 \\ 0.50 & 1.00 \end{bmatrix}$$

$$v = \begin{bmatrix} 2.00 \\ 3.00 \end{bmatrix}$$

$$Av = \begin{bmatrix} 3.50 \\ 4.00 \end{bmatrix}$$

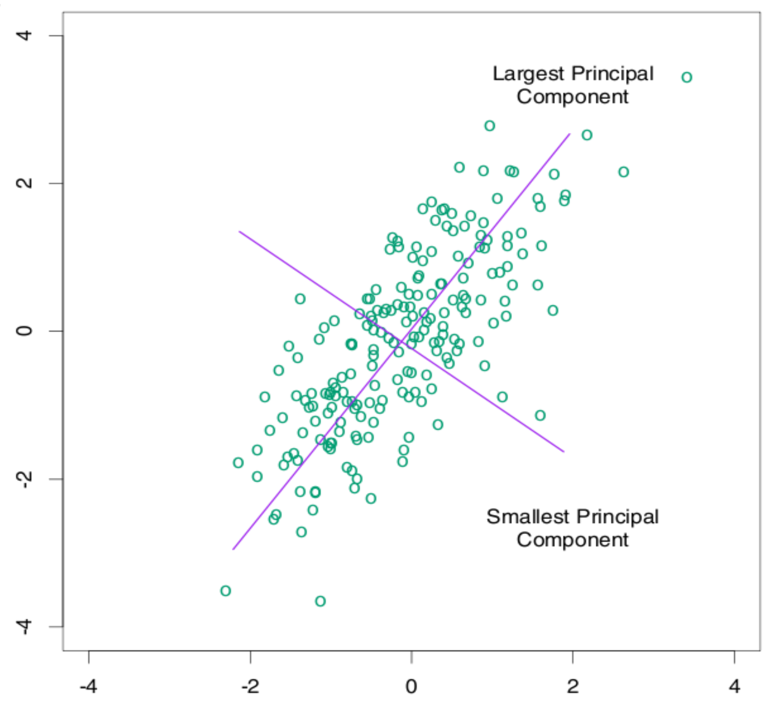


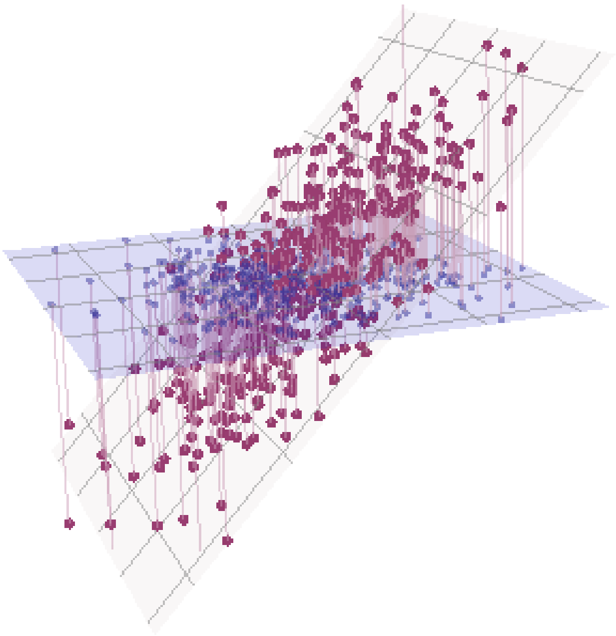
$\lambda_1 = 1.5$
$\lambda_2 = 0.5$

http://setosa.io/ev/eigenvectors-and-eigenvalues/

Principal Component Analysis - PCA

Probably the most widely-used and well-known of the **"standard"** multivariate methods
invented by Karl Pearson (1901) and independently developed by Harold Hotelling (1933)
Karl Person founded the world's first university statistics department at University College London in 1911.



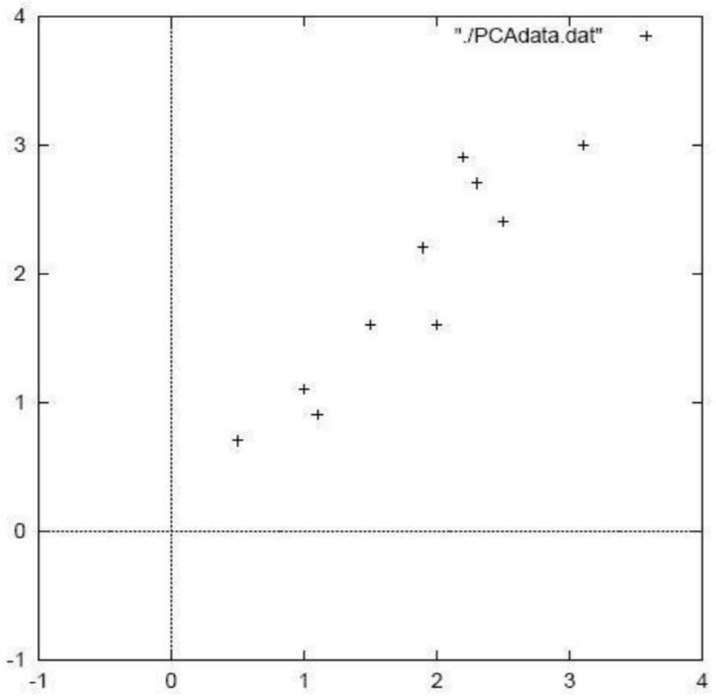


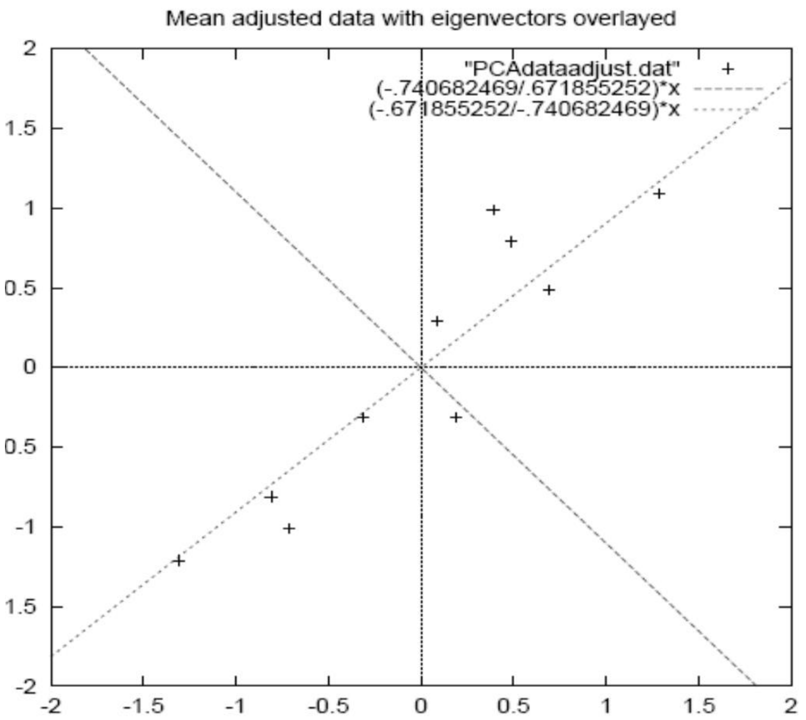
PCA

X : the data matrix with N=11 objects and d=2 dimensions.

Step 1: subtract the mean and calculate the covariance matrix C.

$$C = \begin{pmatrix} 0.716 & 0.615 \\ 0.615 & 0.616 \end{pmatrix}$$





PCA -Example

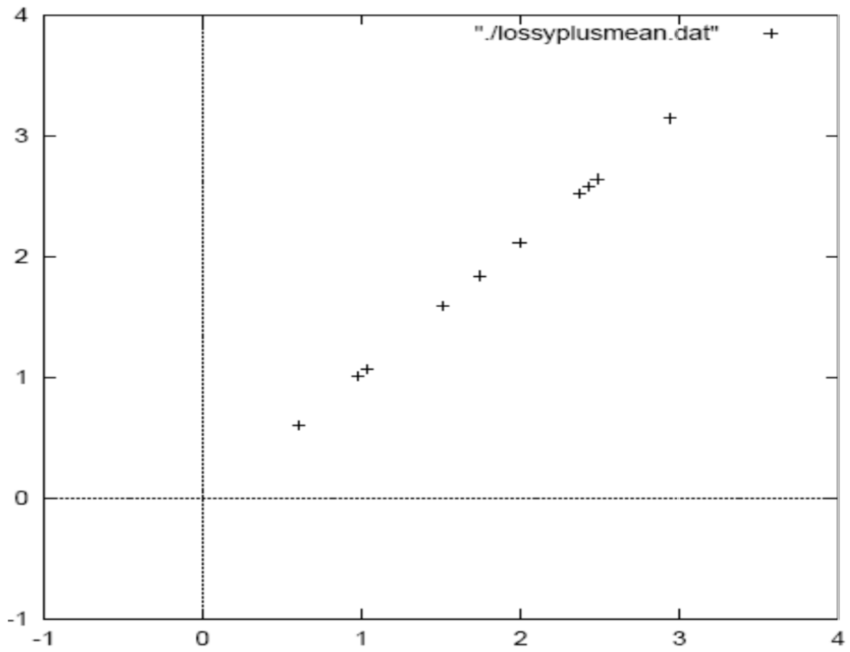
Step 2: Calculate the eigenvectors and eigenvalues of the covariance matrix:

$\lambda_1 \approx 1.28$, $v_1 \approx [-0.677 \;\; -0.735]^T$, $\lambda_2 \approx 0.49$, $v_2 \approx [-0.735 \;\; 0.677]^T$

Notice that v1 and v2 are orthonormal: |v1 |=1 |v2 |=1 v1 . v2 = 0

Step 3: project the data Let $V = [v1 , \cdots vm]$ is $d \times m$ matrix where the columns vi are the eigenvectors corresponding to the largest m eigenvalues The projected data: $Y = X\,V$ is $N \times m$ matrix. If m=d (more precisely rank(X)), then there is no loss of information!

The eigenvector with the highest eigenvalue is the principle component of the data. if we are allowed to pick only one dimension, the principle component is the best direction (retain the maximum variance). Our PC is v1 $\approx$ [-0.677 -0.735]



PCA

PCA finds a linear projection of high dimensional data into a lower dimensional subspace such as:

- ✓ The variance retained is maximized.
- ✓ The least square reconstruction error is minimized.

$$\mathrm{cov}(X, Y) = \mathbf{E}\left[(X - \mathbf{E}[X])(Y - \mathbf{E}[Y])\right],$$

PCA steps:

- ✓ transform an $N \times d$ matrix X into an $N \times m$ matrix Y:
  - Centralized the data (subtract the mean).
  - Calculate the $d \times d$ covariance matrix: $\mathrm{C} = \frac{1}{N-1} X^T X$
    - o $C_{i,j} = \frac{1}{N-1} \sum_{q=1}^{N} X_{q,i} . X_{q,j}$
    - o Ci,i (diagonal) is the variance of variable i.
    - o Ci,j (off-diagonal) is the covariance between variables i and j.
  - Calculate the eigenvectors of the covariance matrix (orthonormal).
  - Select *m* eigenvectors that correspond to the largest *m* eigenvalues to be the new basis.

PCA Pseudo-Code and Example

Input: $x_1, \ldots, x_n$ d length vector, k

Output: Transform matrix R

$X \Leftarrow n \times d$ data matrix with x_i in each row;

$\overline{x} \Leftarrow \frac{1}{n} \sum_{i=1}^{n} x_i$;

$X \Leftarrow$ subtract $\overline{x}$ from each row x_i in X;

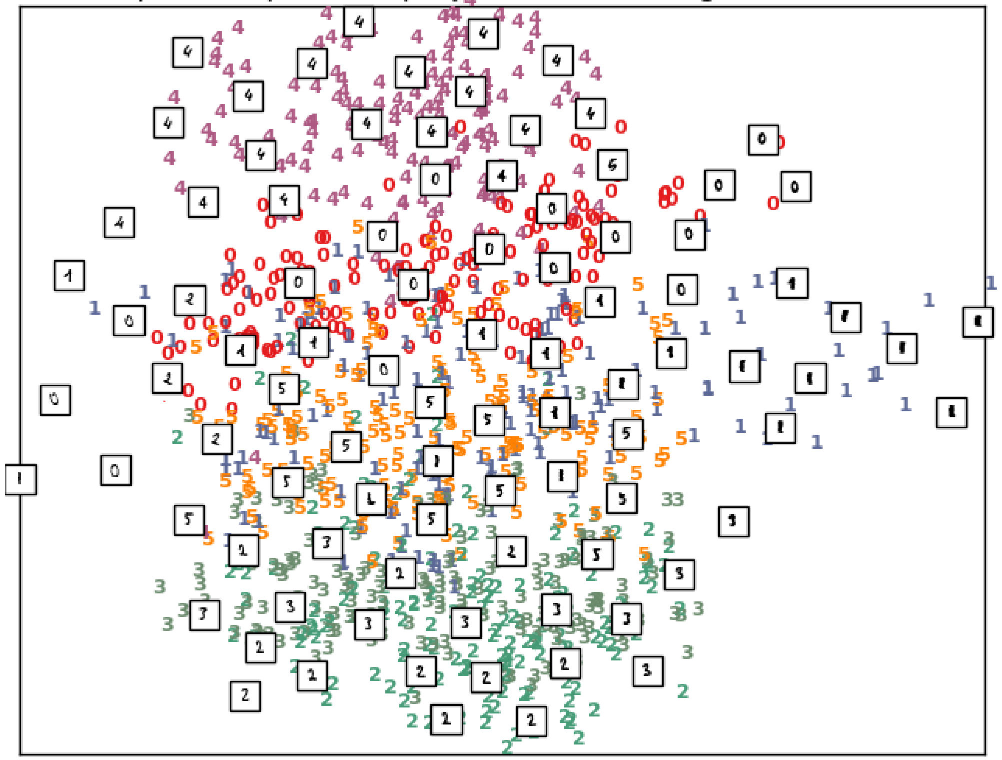
$COV \Leftarrow \frac{1}{n-1} X^{\mathrm{T}} \times X$ Compute eigenvalue $e_1, \ldots, e_d$ of COV, and sort them;

Compute matrix V which satisfy $V^{-1} \times COV \times V = D$, D is the diagonal matrix of eigenvalue of COV;

$R \Leftarrow$ the first k column of V

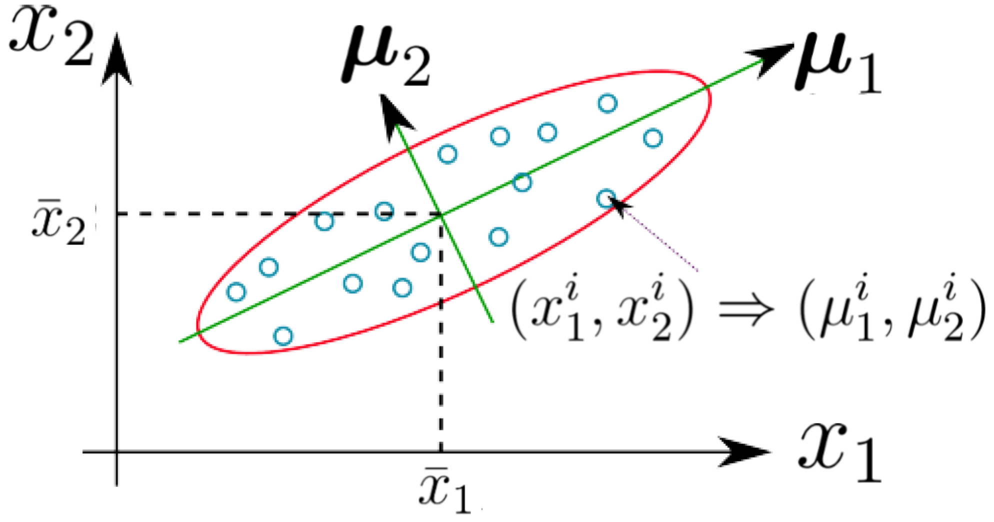
Principal Component Analysis

Principal Components projection of the digits (time 0.01s)



PCA

http://setosa.io/ev/principal-component-analysis/



1. Let $\mu = \frac{1}{m}\sum_{i=1}^m x^{(i)}$
2. Replace each $x^{(i)}$ with $x^{(i)} - \mu$
3. Let $\sigma_j^2 = \frac{1}{m}\sum_i (x_j^{(i)})^2$
4. Replace each $x_j^{(i)}$ with $x_j^{(i)}/\sigma_j$

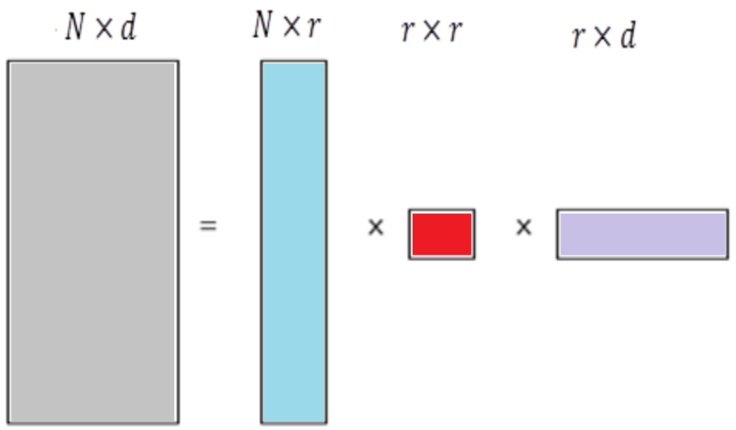
$$\begin{aligned}\frac{1}{m}\sum_{i=1}^m ({x^{(i)}}^T u)^2 &= \frac{1}{m}\sum_{i=1}^m u^T x^{(i)} {x^{(i)}}^T u \\ &= u^T \left(\frac{1}{m}\sum_{i=1}^m x^{(i)} {x^{(i)}}^T\right) u\end{aligned}$$

We easily recognize that the maximizing this subject to $\|u\|_2 = 1$ gives the principal eigenvector

$\Sigma = \frac{1}{m}\sum_{i=1}^m x^{(i)} {x^{(i)}}^T$ which is empirical covariance matrix of the data.

Singular Value Decomposition (SVD)

Any $N \times d$ matrix X can be uniquely expressed as:



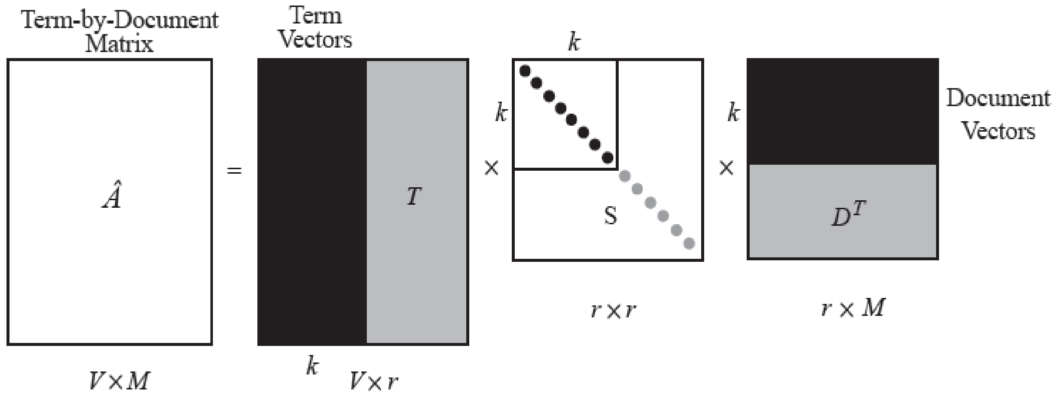
- r is the rank of the matrix X (# of linearly independent columns/rows).
- U is a column-orthonormal $N \times r$ matrix.
- Σ is a diagonal $r \times r$ matrix where the singular values σi are sorted in descending order.
- V is a column-orthonormal $d \times r$ matrix

Given the Term-by-Document Matrix, we can see two categories of documents in which rock has different semantic meanings.

	D1	D2	D3	D4	D5	D6	Q1
rock	2	1	0	2	0	1	1
granite	1	0	1	0	0	0	0
marble	1	2	0	0	0	0	1
music	0	0	0	1	2	0	0
song	0	0	0	1	0	2	0
band	0	0	0	0	1	0	0

SVD for LSI

Given the Term-Document Matrix, we can use SVD to lower the dimension. We choose k = 2 in this case, all documents can be represented in 2-dimensional space in which the distance can be measured angles.

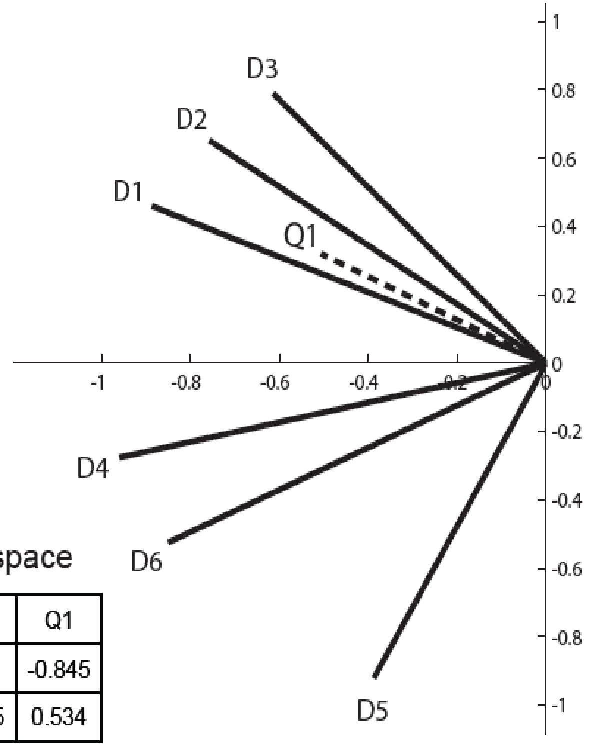


Original term-by-document matrix

	D1	D2	D3	D4	D5	D6	Q1
rock	2	1	0	2	0	1	1
granite	1	0	1	0	0	0	0
marble	1	2	0	0	0	0	1
music	0	0	0	1	2	0	0
song	0	0	0	1	0	2	0
band	0	0	0	0	1	0	0

Documents projected into 2D semantic space

	D1	D2	D3	D4	D5	D6	Q1
Dim. 1	-0.888	-0.759	-0.615	-0.961	-0.388	-0.851	-0.845
Dim. 2	0.460	0.652	0.789	-0.276	-0.922	-0.525	0.534



Another Example

doc-to-concept similarity matrix

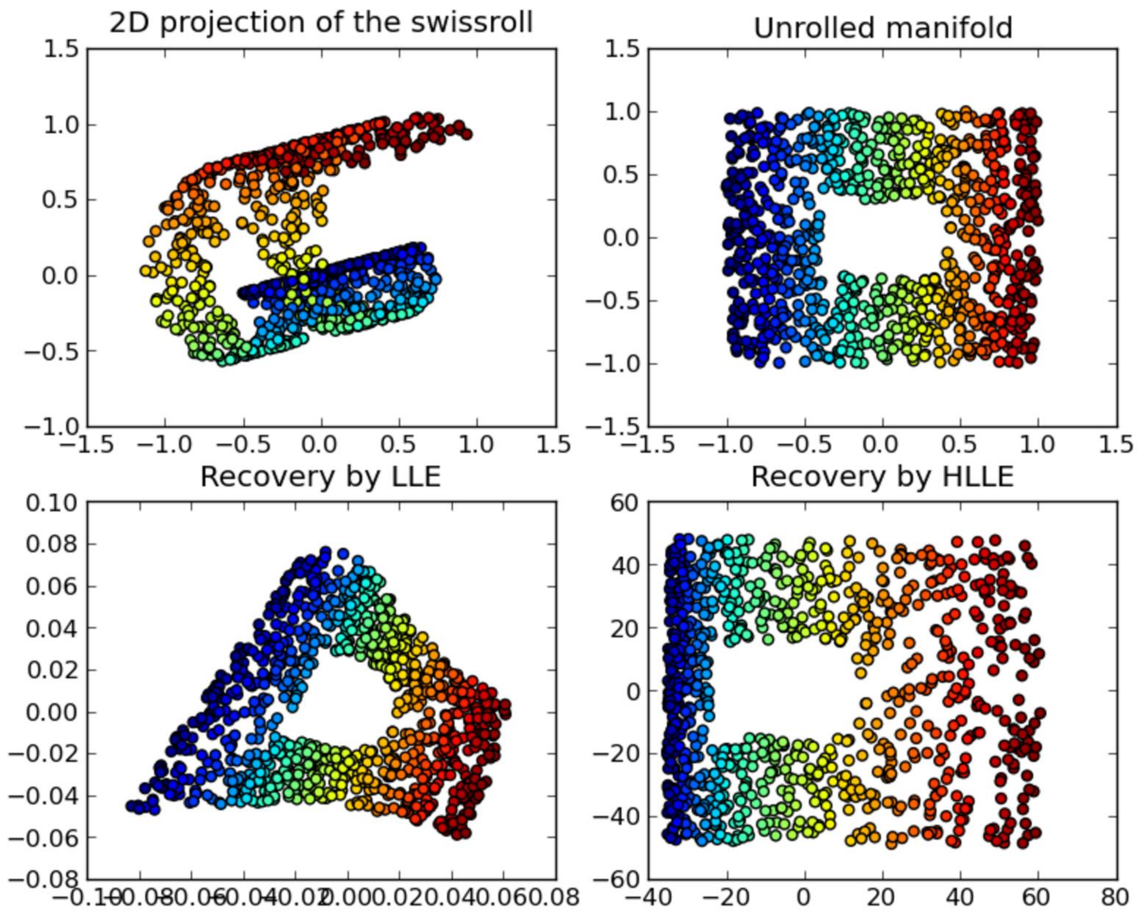
concepts strengths

term-to-concept similarity matrix

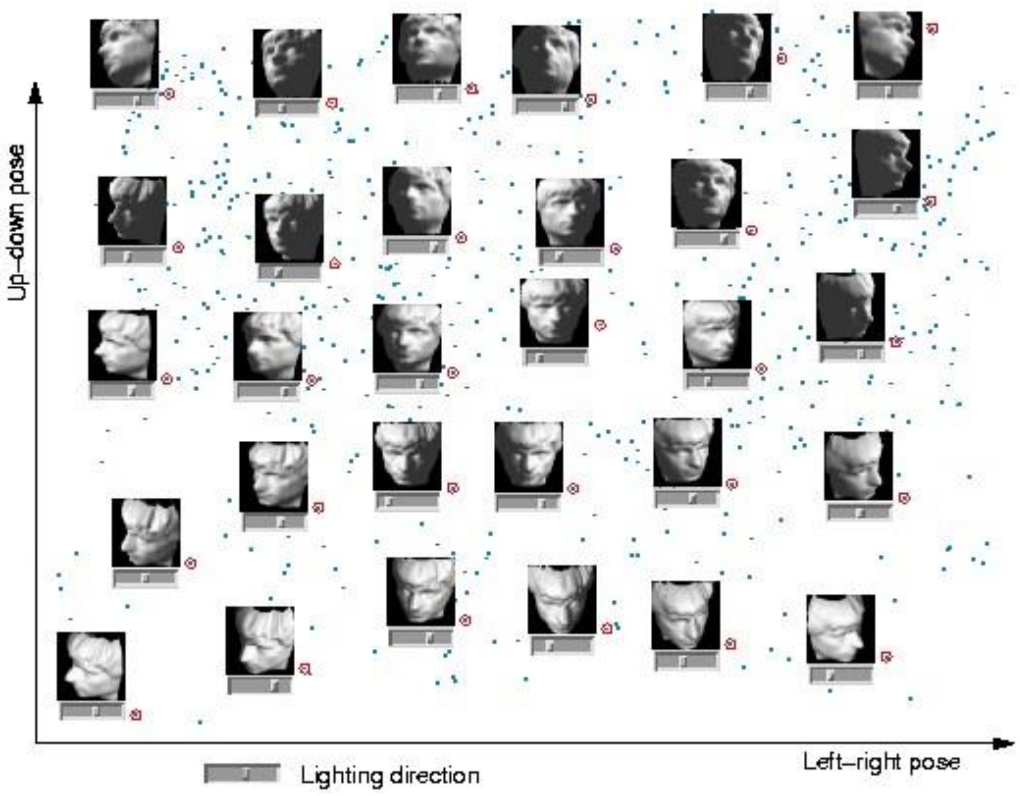
$$
\begin{array}{c} \\ \text{CS} \\ \\ \\ \text{MD} \\ \\ \end{array}
\begin{array}{c} \begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}
=
\begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix}
\times
\begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix}
\times
\begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}
$$

U: document-to-concept similarity matrix
V: term-to-concept similarity matrix. Example:
U1,1 is the weight of CS concept in document d1 ,
σ1 is the strength of the CS concept.
V1,1 is the weight of ‘data’ in the CS concept. **V1,2=0** means ‘data’ has zero similarity with the 2nd concept (Medical).

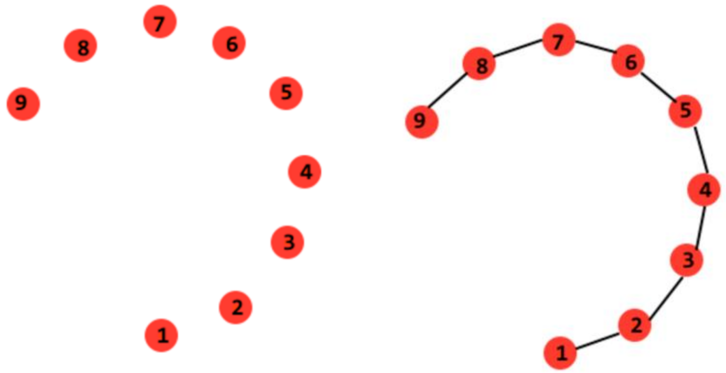
Nonlinear Dimension Reduction



Faces on manifold



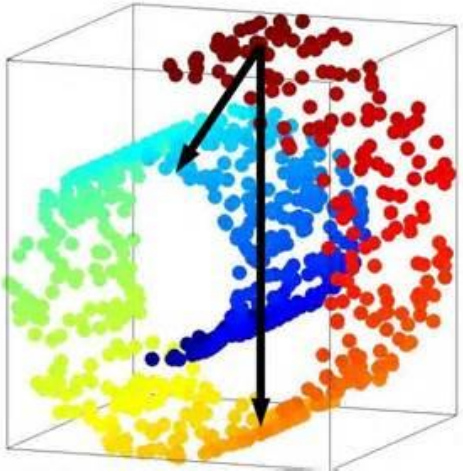
Isomap

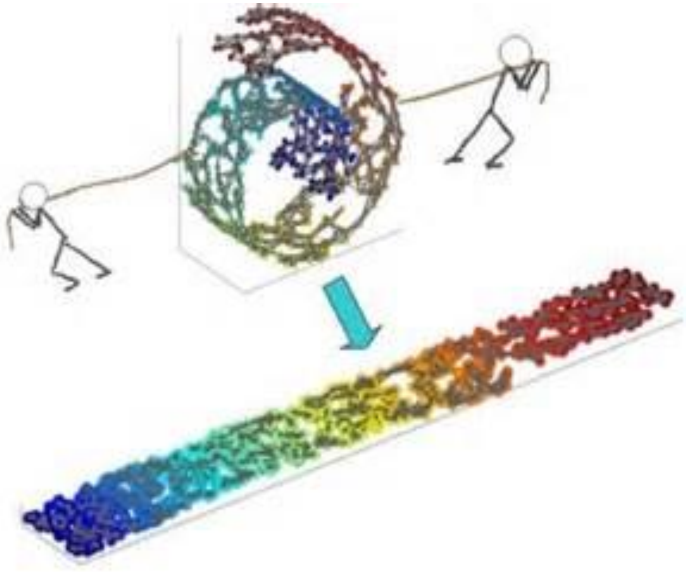


In Isomap, the distances between points are the weight of the shortest path in a point-graph (Dijkstra Algorithm).The point graph is constructed by placing an edge between two points if the Euclidean distance between them falls under a certain threshold or between a point and its top k-neighbors.

We then can map high-dimensional data to lower embedding dimension.

E.g.: $\boldsymbol{R}^3 \rightarrow \boldsymbol{R}^2$

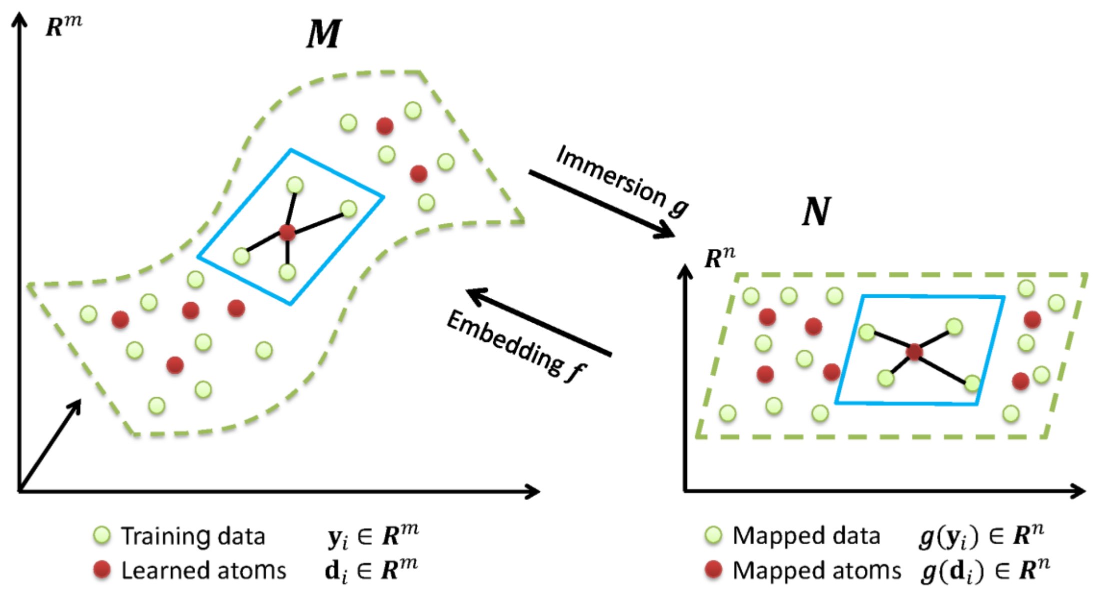




Local Linear Embedding (LLE)

LLE computes the barycentric coordinates of a point X_i based on its neighbors X_j. The original point is reconstructed by a linear combination, given by the weight matrix W_{ij}, of its neighbors. The reconstruction error is given by the cost function $E(W)$.

$$E(W) = \sum_i |\mathbf{X}_i - \sum_j \mathbf{W}_{ij}\mathbf{X}_j|^2 \quad \text{Where: } \sum_j \mathbf{W}_{ij} = 1$$



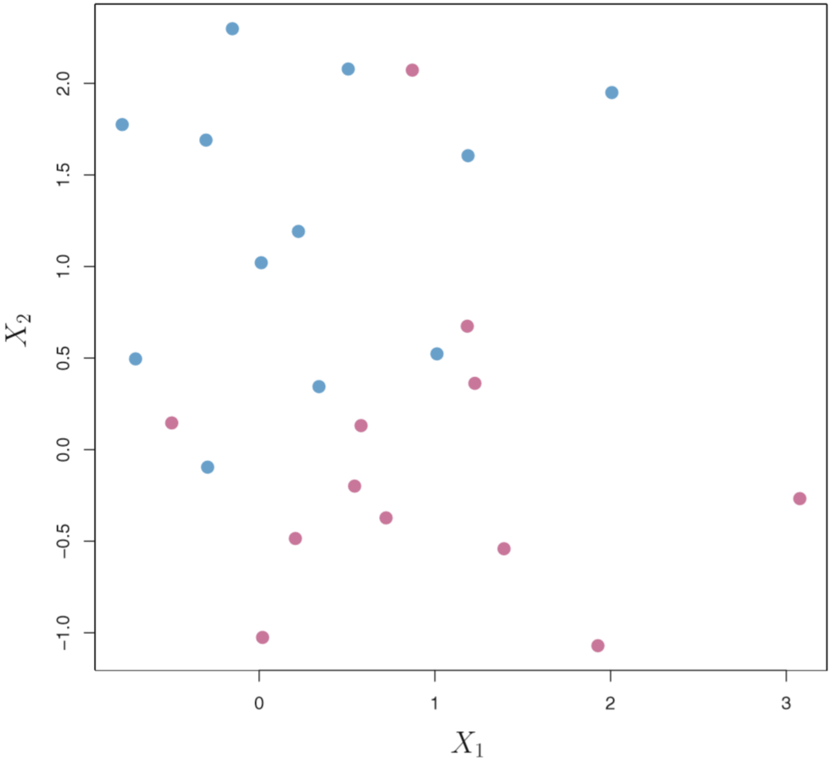
A neighborhood preserving map is created based on this idea. Each point X_i in the D dimensional space is mapped onto a point Y_i in the d dimensional space by minimizing the cost function:

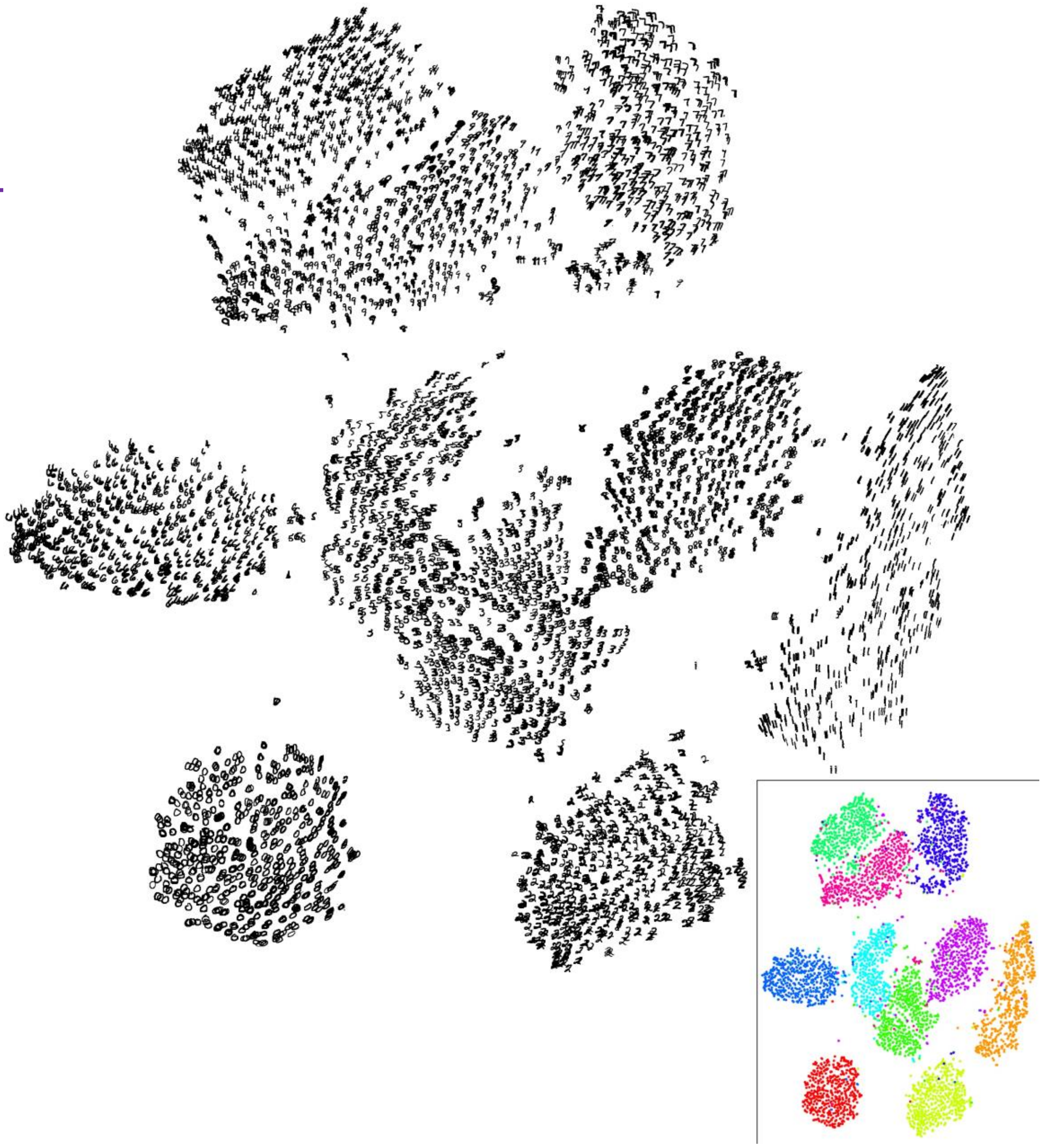
$$C(Y) = \sum_i |\mathbf{Y}_i - \sum_j \mathbf{W}_{ij}\mathbf{Y}_j|^2$$

Matlab Code: http://www.cs.nyu.edu/~roweis/lle/code.html

MINST Visualization

Stochastic Neighbor Embedding (SNE) starts by converting the high-dimensional Euclidean distances between datapoints into conditional probabilities that represent similarities.





References

Andew Ng, Lectures notes of Machine Learning, CS229 at Stanford.
T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning, 2001, Springer.
C. Bishop, Pattern Recognition and Machine Learning, 2006, Springer.
T. Mitchell, Machine Learning, McGraw Hill.
https://en.wikipedia.org/wiki/Kernel_density_estimation
http://setosa.io/ev/principal-component-analysis/
http://people.cs.pitt.edu/~iyad/PCA.pdf
http://sebastianraschka.com/Articles/2014_pca_step_by_step.html