Dimensionality reduction

Victor Kitov

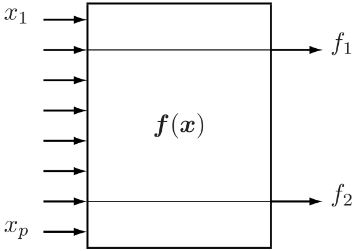
Yandex School of Data Analysis

Table of Contents

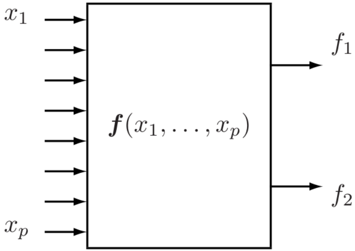
1. Feature extraction
2. Principal component analysis
3. SVD decomposition
4. Non-linear dimensionality reduction

Definition

Feature selection / Feature extraction



(a) feature selector



(b) feature extractor

Feature extraction: find transformation of original data which extracts most relevant information for machine learning task.

We will consider unsupervised dimensionality reduction methods, which try to preserve geometrical properties of the data.

Applications of dimensionality reduction

Applications:

- visualization in 2D or 3D
- reduce operational costs (less memory, disc, CPU usage on data transfer)
- remove multi-collinearity to improve performance of machine-learning models

PCA vs. regularization.

Categorization

Supervision in dimensionality reduction:

- supervised (such as Fisher's direction)
- unsupervied

Mapping to reduced space:

- linear
- non-linear

Table of Contents

1. Feature extraction
2. Principal component analysis
    - Definition
3. SVD decomposition
4. Non-linear dimensionality reduction

2 Principal component analysis
- Definition

Definition

Linear transformation of data, using orthogonal matrix $A = [a_1; a_2; ...a_D] \in \mathbb{R}^{DxD}$, $a_i \in \mathbb{R}^D$:

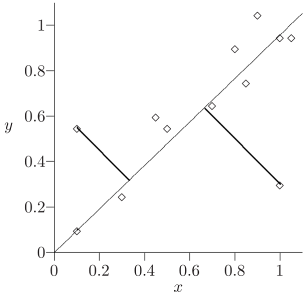
$$\xi = A^T x$$

Equivalent ways to derive PCA:

1. Find line of best fit, plane of best fit, etc.
   - fit is the sum of squares of perpendicular distances.
2. Find line, plane, etc. preserving most of the variability of the data.
   - variability is a sum of squared projections
3. Find orthogonal transform A yielding new variables ξ_i having stationary values for their variance and uncorrelated ξ_j

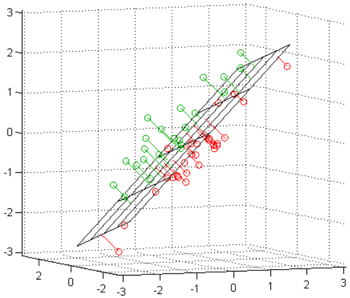
Example: line of best fit

- In PCA sum of squared of perpendicular distances to line is minimized
    - compare with regression



- Not invariant to scale - features should be standardized.
- Method works for $\mathbb{E}x = 0$.

Best hyperplane fit



Subspace L_k or rank k best fits points $x_1, x_2, ...x_D$ if sum of squared distances of these points to this plane is maximized over all planes of rank k.

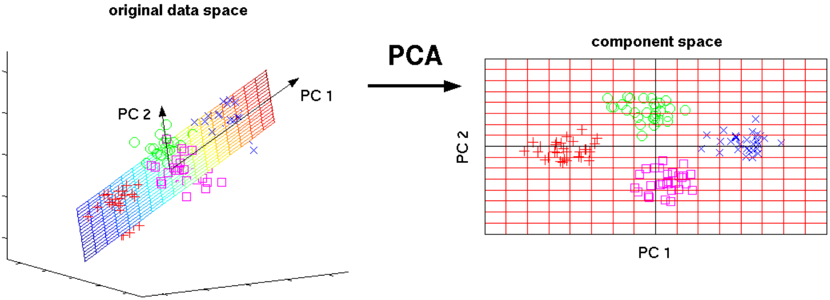
Best hyperplane fit

For point x_i denote p_i the projection on plane L_k and h_i - orthogonal component. Then $\|x_i\|^2 = \|p_i\|^2 + \|h_i\|^2$.
For set of points:

$$\sum_i \|x_i\|^2 = \sum_i \|p_i\|^2 + \sum_i \|h_i\|^2$$

Since sum of squares is constant, minimization of $\sum_i \|h_i\|^2$ is equivalent to maximization of $\sum_i \|p_i\|^2$.

PCA for visualization



Covariance matrix properties

$\Sigma = cov[x] \in \mathbb{R}^{DxD}$ is symmetric positive semidefinite matrix

- has $\lambda_1, \lambda_2, ...\lambda_D$ eigenvalues, satisfying: $\lambda_i \in \mathbb{R}$, $\lambda_i \geq 0$.
- if eigenvalues are unique, corresponding eigenvectors are also unique
- always exists a set of orthogonal eigenvectors $z_1, z_2, ...z_D$: $\Sigma z_i = \lambda_i z_i$.

later we will assume that $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_D \geq 0$.

Derivation

1-st component:

$$\begin{cases} \mathrm{Var}\xi_1 \to \max_a \\ |a_1|^2 = a_1^T a_1 = 1 \end{cases}$$

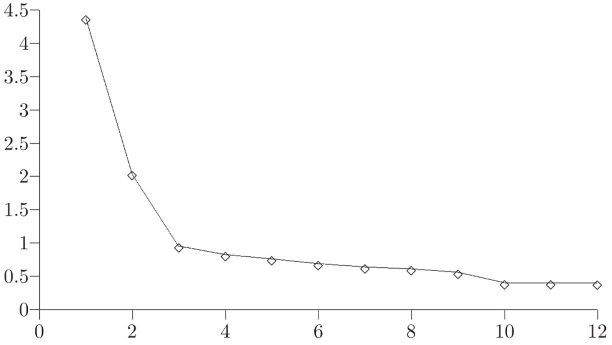
2-nd component:

$$\begin{cases} \mathrm{Var}[\xi_2] = a_2^T \Sigma a_2 \to \max_{a_2} \\ a_2^T a_2 = |a_2|^2 = 1 \\ cov[\xi_1, \xi_2] = a_2^T \Sigma a_1 = \lambda_1 a_2^T a_1 = 0 \end{cases}$$

...

Number of components

- Data visualization: 2 or 3 components.
- Take most significant components until their variance falls sharply down:



Number of components

Remind that $A = [a_1|a_2|...|a_D],\ A^T A = I,\ \xi = A^T x.$
Denote $S_k = [\xi_1, \xi_2, ...\xi_k, 0, 0, ..., 0] \in \mathbb{R}^D$

$$\mathbb{E}[\|S_k\|^2] = \mathbb{E}[\xi_1^2 + \xi_2^2 + ... + \xi_k^2] = \sum_{i=1}^{k} \text{var}\, \xi_i = \sum_{i=1}^{k} \lambda_i$$

$$\begin{aligned} \mathbb{E}[\|S_D\|^2] &= \mathbb{E}[\xi^T \xi] = \\ &= \mathbb{E} x^T A A^T x = \mathbb{E}\left[x^T x\right] = \mathbb{E}[\|x\|^2] \end{aligned}$$

Select such k^* that

$$\frac{\mathbb{E}[\|S_k\|^2]}{\mathbb{E}[\|x\|^2]} = \frac{\mathbb{E}[\|S_k\|^2]}{\mathbb{E}[\|S_D\|^2]} = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{D} \lambda_i} > \textit{threshold}$$

We may select k^* to account for 90%, 95% or 99% of total variance.

Transformation $\xi \rightleftarrows x$

Dependence between original and transformed features:

$$\xi = A^T(x - \mu),\ x = A\xi + \mu,$$

where μ is the mean of the original non-shifted data.
Taking first r components - $A_r = [a_1|a_2|...|a_r]$, we get the image of the reduced transformation:

$$\xi_r = A_r^T(x - \mu)$$

ξ_r will correspond to

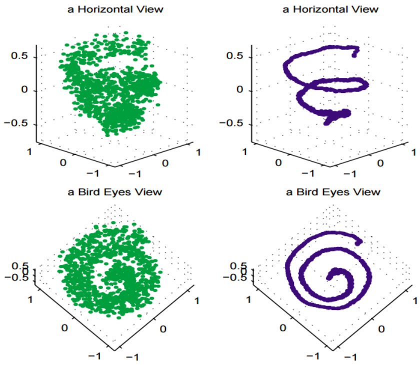
$$x_r = A\begin{pmatrix} \xi_r \\ 0 \end{pmatrix} + \mu = A_r\xi_r + \mu$$

$$x_r = A_r A_r^T(x - \mu) + \mu$$

$A_r A_r^T$ is projection matrix with rank r.

Application - data filtering

Local linear projection method:



X. Huo and Jihong Chen (2002). Local linear projection (LLP). First IEEE Workshop on Genomic Signal Processing and Statistics (GENSIPS), Raleigh, NC, October. http://www.gensips.gatech.edu/proceedings/.

Properties of PCA

- Depends on scaling of individual features.
- Assumes that each feature has zero mean.

- Covariance matrix replaced with sample-covariance.
- Does not require distribution assumptions about x.

Example

Faces database:

Eigenfaces

Table of Contents

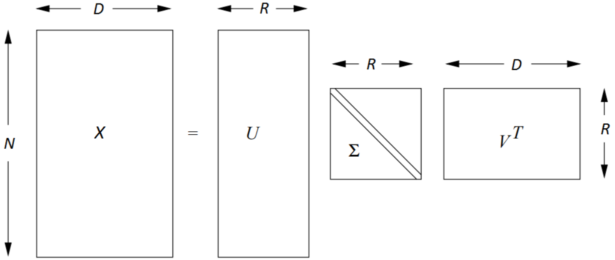
1. Feature extraction
2. Principal component analysis
3. SVD decomposition
4. Non-linear dimensionality reduction

SVD decomosition

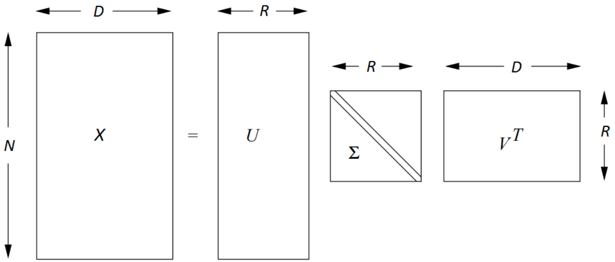
Every matrix $X \in \mathbb{R}^{NxD}$ of rank R can be decomposed into the product of three matrices:

$$X = U\Sigma V^T$$

where $U \in \mathbb{R}^{NxR}$, $\Sigma \in \mathbb{R}^{RxR}$, $V^T \in \mathbb{R}^{RxD}$, and $\Sigma = diag\{\sigma_1, \sigma_2, ...\sigma_R\}$, $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_R \geq 0$, $U^T U = I$, $V^T V = I$. I denotes identity matrix.



Interpretation of SVD



For X_{ij} let i denote objects and j denote properties.

- U represents standardized coordinates of concepts
- V^T represents standardized concepts representations
- Σ shows the magnitudes of presence of standardized concepts in X.

Example

	The lord of the rings	Harry Potter	Avatar	Titanic	Love story	A walk to remember
Andrew	4	5	5	0	0	0
John	4	4	5	0	0	0
Matthew	5	5	4	0	0	0
Anna	0	0	0	5	5	5
Maria	0	0	0	5	5	4
Jessika	0	0	0	4	5	4

Example

$$U=\begin{pmatrix}0. & 0.6 & -0.3 & 0. & 0. & -0.8\\ 0. & 0.5 & -0.5 & 0. & 0. & 0.6\\ 0. & 0.6 & 0.8 & 0. & 0. & 0.2\\ 0.6 & 0. & 0. & -0.8 & -0.2 & 0.\\ 0.6 & 0. & 0. & 0.2 & 0.8 & 0.\\ 0.5 & 0. & 0. & 0.6 & -0.6 & 0.\end{pmatrix}$$

$$\Sigma=\text{diag}\{\begin{pmatrix}14. & 13.7 & 1.2 & 0.6 & 0.6 & 0.5\end{pmatrix}\}$$

$$V^T=\begin{pmatrix}0. & 0. & 0. & 0.6 & 0.6 & 0.5\\ 0.5 & 0.6 & 0.6 & 0. & 0. & 0.\\ 0.5 & 0.3 & -0.8 & 0. & 0. & 0.\\ 0. & 0. & 0. & -0.2 & 0.8 & -0.6\\ -0. & -0. & -0. & 0.8 & -0.2 & -0.6\\ 0.6 & -0.8 & 0.2 & 0. & 0. & 0.\end{pmatrix}$$

Example (excluded insignificant concepts)

$$U_2 = \begin{pmatrix} 0. & 0.6 \\ 0. & 0.5 \\ 0. & 0.6 \\ 0.6 & 0. \\ 0.6 & 0. \\ 0.5 & 0. \end{pmatrix}$$

$$\Sigma_2 = \text{diag}\{\begin{pmatrix} 14. & 13.7 \end{pmatrix}\}$$

$$V_2^T = \begin{pmatrix} 0. & 0. & 0. & 0.6 & 0.6 & 0.5 \\ 0.5 & 0.6 & 0.6 & 0. & 0. & 0. \end{pmatrix}$$

Concepts may be

- patterns among movies (along j) - fantasy/romance
- patterns among people (along i) - boys/girls

Dimensionality reduction case: patterns along j axis.

Applications

- Example: new movie rating by new person

$$x = \begin{pmatrix} 5 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- **Dimensionality reduction:** map x into concept space:

$$y = V_2^T x = \begin{pmatrix} 0 & 2.7 \end{pmatrix}$$

- **Recommendation system:** map y back to original movies space:

$$\widehat{x} = y V_2^T = \begin{pmatrix} 1.5 & 1.6 & 1.6 & 0 & 0 & 0 \end{pmatrix}$$

Fronebius norm

- Fronebius norm of matrix X is $\|X\|_F \stackrel{df}{=} \sqrt{\sum_{n=1}^{N} \sum_{d=1}^{D} x_{nd}^2}$
- Using properties $||X||_F = \text{tr}\, XX^T$ and $\text{tr}\, AB = \text{tr}\, BA$, we obtain:

$$\begin{aligned} \|X\|_F &= \text{tr}[U\Sigma V^T V\Sigma U^T] = \text{tr}[U\Sigma^2 U^T] = \\ &= \text{tr}[\Sigma^2 U^T U] = \text{tr}[\Sigma^2] = \sum_{r=1}^{R} \sigma_r^2 \end{aligned} \quad (1)$$

Matrix approximation

Consider approximation $X_k = U\Sigma_k V^T$, where
$\Sigma_k = \text{diag}\{\sigma_1, \sigma_2, ...\sigma_k, 0, 0, ..., 0\} \in \mathbb{R}^{RxR}$.

Theorem 1

X_k is the best approximation of X retaining k concepts.

Proof: consider matrix $Y_k = U\Sigma' V^T$, where Σ' is equal to Σ except some $R-k$ elements set to zero:
$\sigma'_{i_1} = \sigma'_{i_2} = ... = \sigma'_{i_{R-k}} = 0$. Then, using (1)

$$\|X - Y_k\|_F = \left\| U(\Sigma - \Sigma')V^T \right\|_F = \sum_{p=1}^{R-k} \sigma_{i_p}^2 \le \sum_{p=1}^{R-k} \sigma_p^2 = \|X - X_k\|_F$$

since $\sigma_1 \ge \sigma_2 \ge ... \ge \sigma_R \ge 0$.

Matrix approximation

How many components to retain?

General case: Since

$$\|X - X_k\|_F = \left\|U(\Sigma - \Sigma_k)V^T\right\|_F = \sum_{i=k+1}^{R} \sigma_i^2$$

a reasonable choice is k^* such that

$$\frac{\|X - X_{k^*}\|_F}{\|X\|_F} = \frac{\sum_{i=k^*+1}^{R} \sigma_i^2}{\sum_{i=1}^{R} \sigma_i^2} \geq threshold$$

Visualization: 2 or 3 components.

Theorem 2

For any matrix Y_k *with* rank $Y_k = k$: $\|X - X_k\|_F \leq \|X - Y_k\|_F$

Finding U and V

- **Finding** V
  $X^TX = \left(U\Sigma V^T\right)^T U\Sigma V^T = (V\Sigma U^T)U\Sigma V^T = V\Sigma^2 V^T$. It follows that
  $$X^TXV = V\Sigma^2 V^T V = V\Sigma^2$$
  So V consists of eigenvectors of X^TX with corresponding eignvalues $\sigma_1^2, \sigma_2^2, ...\sigma_R^2$.

- **Finding** U**:**
  $XX^T = U\Sigma V^T \left(U\Sigma V^T\right)^T = U\Sigma V^T V\Sigma U^T = U\Sigma^2 U^T$. So
  $$XX^TU = U\Sigma^2 U^T U = U\Sigma^2.$$
  So U consists of eigenvectors of $XX^T$with corresponding eigenvalues $\sigma_1^2, \sigma_2^2, ...\sigma_R^2$.

V concepts are principal components

- Denote the average $\bar{X} \in \mathbb{R}^D : \bar{X}_j = \sum_{i=1}^{N} x_{ij}$
- Denote the n-th row of X be $X_n \in \mathbb{R}^D : X_{nj} = x_{nj}$
- For centered X sample covariance matrix $\widehat{\Sigma}$ equals:

$$\begin{aligned}\widehat{\Sigma} &= \frac{1}{N}\sum_{n=1}^{N}(X_n - \bar{X})(X_n - \bar{X})^T = \frac{1}{N}\sum_{n=1}^{N} X_n X_n^T \\ &= \frac{1}{N}X^T X\end{aligned}$$

- V **consists of principal components** since
  - V consists of eigenvectors of $X^T X$,
  - principal components are eignevectors of $\widehat{\Sigma}$ and
  - $\widehat{\Sigma} \propto X^T X$.

Table of Contents

1. Feature extraction
2. Principal component analysis
3. SVD decomposition
4. Non-linear dimensionality reduction
    - Global methods
    - Local methods

4 Non-linear dimensionality reduction
- Global methods
- Local methods

Multi-dimensional scaling

Multi-dimensional scaling

Map $x \to y$ preserving distances as much as possible.

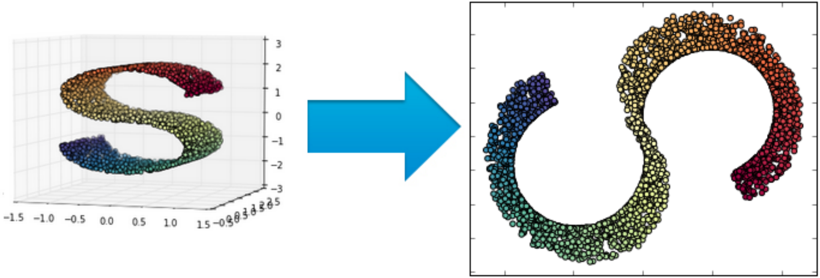
- Approaches:
  - absolute difference

$$\sum_{i,j} \left(\|x_i - x_j\| - \|y_i - y_j\| \right)^2 \to \min_Y$$

  - relative difference (more attention to small distances)

$$\sum_{i,j} \frac{\left(\|x_i - x_j\| - \|y_i - y_j\| \right)^2}{\|x_i - x_j\|} \to \min_Y$$

Example



Issue: small $\|x_i - x_j\|$ should not always imply small $\|y_i - y_j\|$, such as in case of red and yellow points.

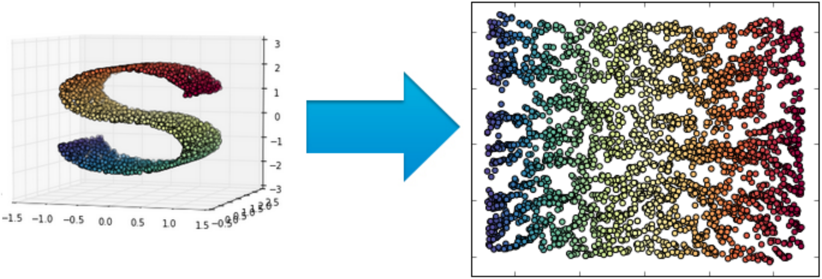
Isomap

Isomap

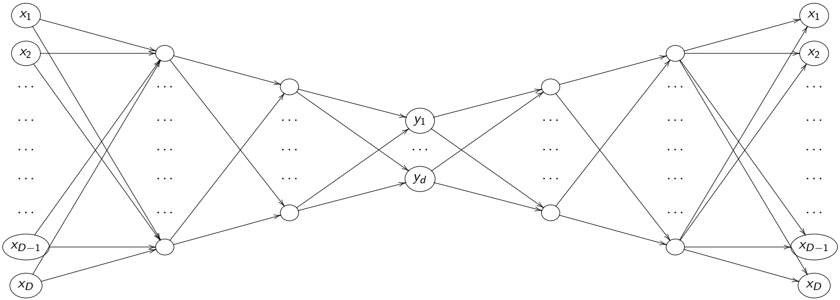
Map $x \to y$ preserving correspondence between distance in transformed space and "geodesic" distance along the surface in original space.

- This apprach solves the previous issue of MDS.
- Geodesic distance calculation:
  1. for each x_n find its K nearest neighbours $x_{n_1, n_2, \ldots n_K}$
  2. build the pairwise distance matrix, filling distance between samples and their k-NN.
  3. calculate all pairwise distances using shortest-path algorithm of Dijkstra or Floyd.
- Finally usual MDS is applied to match $\|x_i - x_j\|_G$ and $\|y_i - y_j\|$, where $\|\cdot\|_G$ is geodesic distance.

Example of ISOMAP



Autoencoders



- feed-forward neural network, tranined to reproduce input with MSE loss.
- D input and D output nodes
- d nodes in the central layer
- User-defined number of layers and nodes

Autoencoders

- Benefits: can map new points to reduced space
- Issues:
    - optimization may get stuck in local optima
    - slow convergence (can be improved with specific starting weights)
    - unfeasible to apply to high d (too many connections).

4. Non-linear dimensionality reduction
   - Global methods
   - Local methods

Local linear embedding

Local linear embedding

Method preserves reconstruction weights of objects through their nearest neighbors.

ALGORITHM:

for each x_i:

find its K nearest neighbours: $x_{i(1)}, x_{i(2)}, \dots x_{i(K)}$

find weights to reconstruct x_i using its neighbours:

$x_i \approx \sum_{k=1}^{K} w_{ik} x_{i(k)}$

solve optimization problem:

$\sum_{n=1}^{N} (y_i - \sum_{k=1}^{K} w_{ik} y_{ik})^2 \to \max_Y$
