Dimensionality Reduction

Elena Trunz

September 21, 2023

Uni Bonn

Overview

Introduction

Motivation

Working with high-dimensional data (e.g. images) comes with some difficulties:

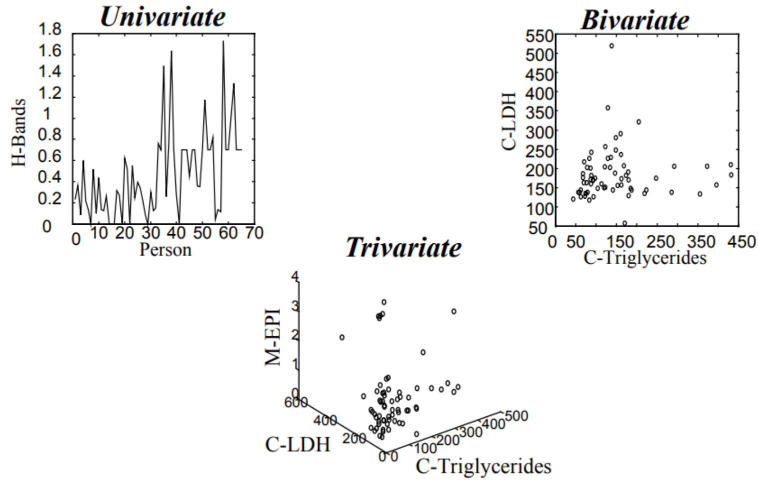
- hard to analize
- interpretaion is difficult
- visualization is almost impossible
- computationally expensive
- needs high amount of storage

Example: visualization

Table: Part of 53 blood and urine measurements from 65 people (33 alcoholics, 32 non-alcoholics)

	H-WBC	H-RBC	H-Hgb	H-Hct	H-MCV	H-MCH	H-MCHC
A1	8.0	4.82	14.1	41	85	29	34
A2	7.3	5.02	14.7	43	86	29	34
A3	4.3	4.48	14.1	41	91	32	35
A4	7.5	4.47	14.9	45	101	33	33
A5	7.3	5.52	15.4	46	84	28	33
A6	6.9	4.86	16.0	47	97	33	34
A7	7.8	4.68	14.7	43	92	31	34
A8	8.6	4.82	15.8	42	88	33	37
A9	5.1	4.71	14.0	43	92	30	32

Example: visualization (2)



Example: visualization (3)

- Is there a better data presentation than ordinate axes?
- Do we need a 53-dimensional space to view the data?
- How to find the ”best” low-dimensional space that conveys maximum useful informaiton?

Dimensionality reduction

High-dimensional data often comes with properties that we can exploit:

- often overcomplete, i.e. many dimensions are redundant and can be explaned by a combination of other dimensions
- dimensions are often correlated, so that the data possesses an intrinsic low-dimensional structure

Dimensionality reduction exploits structure and correlation and allows us to work with a more compact representation of the data, ideally without loosing information!

$\rightarrow$ Compression technique

Managing high-dimensional data

- Database of face scans (3D geometry + texture)
- 10,000 points in each scan
- x, y, z, r, g, b - 6 numbers in each point

$\rightarrow$ each scan is a 60,000 dimensional vector!

There is hope: faces are likely to be governed by small set of parameters (much less than 60,000)

PCA

Goal

Find the "best" low-dimensional space that conveys the maximum useful information by looking for *principal components*

- Which components (features) are important to keep?
- Importance (significance) = variance
- Artificial intelligence = recognizing the significance

Retaining most information after data compression is equivalent to capturing the largest amount of variance in the low-dimensional code.

Covariance of data matrix

- Each sample has m features:

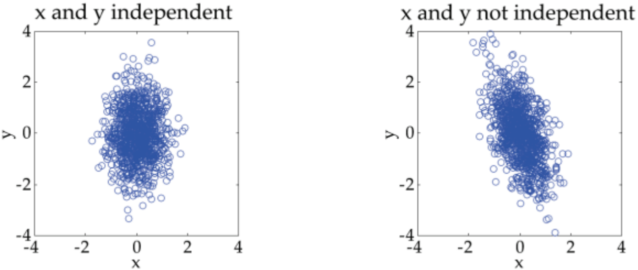
$$\mathbf{x} = (x_1, \dots, x_m)$$

- Take N such samples, subtract the mean over features and combine them in a $n \times m$ matrix X
- Then the covariance matrix of features C is $d \times d$, positive-semidefinite and symmetric:

$$C = \mathrm{Cov}(X) = \mathrm{E}(X^\top X) = \frac{X^\top X}{N-1} = \begin{pmatrix} \sigma_{11}^2 & \cdots & \sigma_{1d}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{d1}^2 & \cdots & \sigma_{dd}^2 \end{pmatrix}$$

Covariance matrix

$$C = \begin{bmatrix} c_{xx} & c_{xy} \\ c_{xy} & c_{yy} \end{bmatrix}$$



$$C = \begin{bmatrix} 0.35 & 0 \\ 0 & 1 \end{bmatrix}$$

covariance diagonal

$$C = \begin{bmatrix} 0.35 & -0.35 \\ -0.35 & 1.35 \end{bmatrix}$$

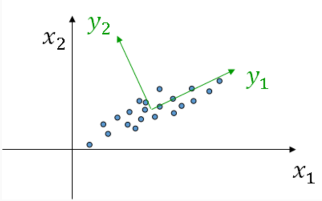
covariance has nonzero off diagonal elements

Principal component analysis

Find an orthonormal matrix P as basis transform $Y = XP$, such that

- C_Y becomes diagonal (remove correlation between dimensions)
- Dimensions are ordered by importance (decreasing order of variance)

The columns of P are the principal components $\mathbf{p}_1, \ldots, \mathbf{p}_d$ of X.



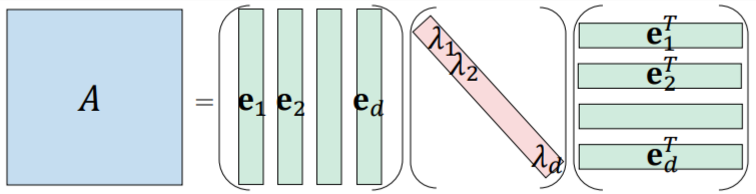
Eigendecomposition

It can be shown that

- the principal components of data matrix X are the eigenvectors of its covariance matrix C_X
- if a matrix is symmetric, then there's always an orthogonal eigenbasis

$\rightarrow$ Perform eigendecomposition of C_X

Eigendecomposition (2)



PCA for dimensionality reduction

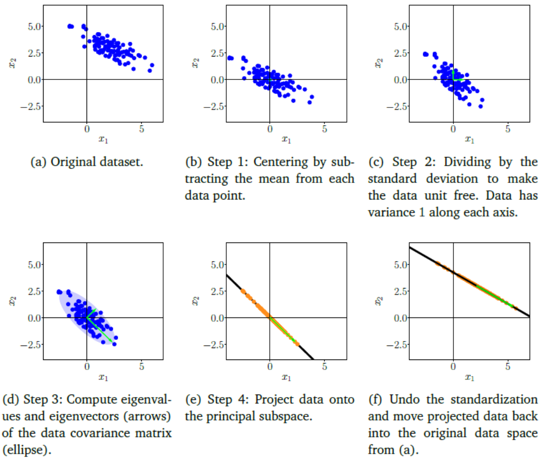
- Center your data by shifting data points to have zero mean
- Project data to eigenspace
  - Compute covariance matrix $C = Cov(X_{centered})$
  - Find eigenvectors and eigenvalues of C
  - Sort eigenvectors according to eigenvalues
- Neglect dimensions (eigenvectors) with small eigenvalues

Eigenvectors that correspond to large eigenvalues are the directions in which the data has strong components (=large variance)

After dimensionality reduction

- Data compression: store only k largest eigenvectors and the mean vector
- Pre-processing: operate on principal components as features or project back and use as preprocessed lower-dimensional dataset for further operations
  - Clustering
  - Denoising
  - Training
  - Etc...
- Visualization: use first 1 to 3 components to visualize your data

Example



(a) Original dataset.

(b) Step 1: Centering by subtracting the mean from each data point.

(c) Step 2: Dividing by the standard deviation to make the data unit free. Data has variance 1 along each axis.

(d) Step 3: Compute eigenvalues and eigenvectors (arrows) of the data covariance matrix (ellipse).

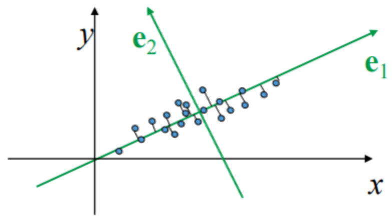
(e) Step 4: Project data onto the principal subspace.

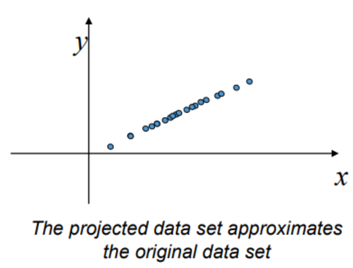
(f) Undo the standardization and move projected data back into the original data space from (a).

Image was taken from [DFO20]

PCA in Applications

Approximation (compression)

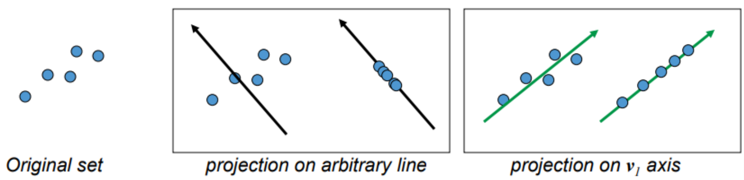




Optimality of approximation (compression)

The approximation is optimal in least squares sense

The projected points have maximal variance

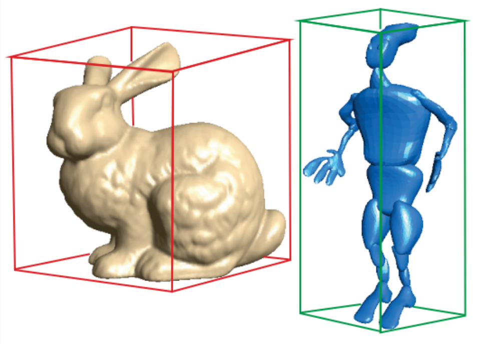


Application: tight bounding box

Bounding boxes are often served as a very simple approximation of objects

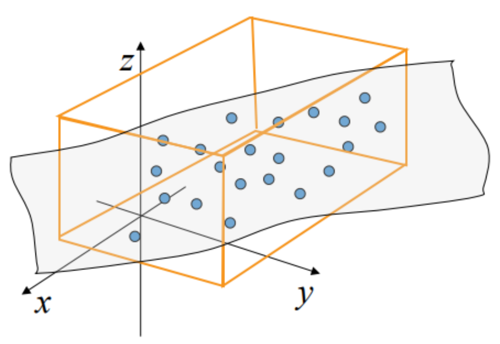
- Fast collision detection
- Fast computation of the size of object

The tighter the bounding box the better!



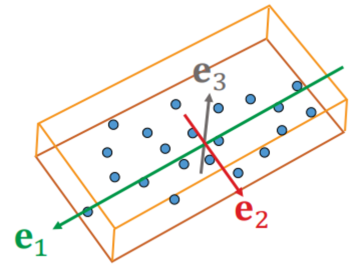
Axis-aligned bounding box

Not the optimal bounding box



Oriented bounding box

- Compute the bounding box with respect to the axis defined by the eigenvectors
- The origin is at the mean point



Data Pre-processing

Pre-processing the data

- Standartization (zero mean, unit standart deviation)
  - Many elements used in the objective function (e.g. RBF kernel of SVM or L1 and L2 regularizers of linear models) may assume that all features are centered around zero or have variance in the same order.
  - If a feature has a variance that is orders of magnitude larger than others, it might dominate the objective function and make the estimator unable to learn from other features correctly as expected.
- Dimensionality reduction (PCA)

Pre-processing the data (2)

- Perform your pre-processing transformation only on train data!
- When finished training, the transformation information (e.g. mean, standard deviation, eigenvectors) must be carried on together with the model to the tester
- Before testing perform the pre-processing transformation on the test data

→ Faster convergence, better results