Lecture 3

Dimensionality Reduction

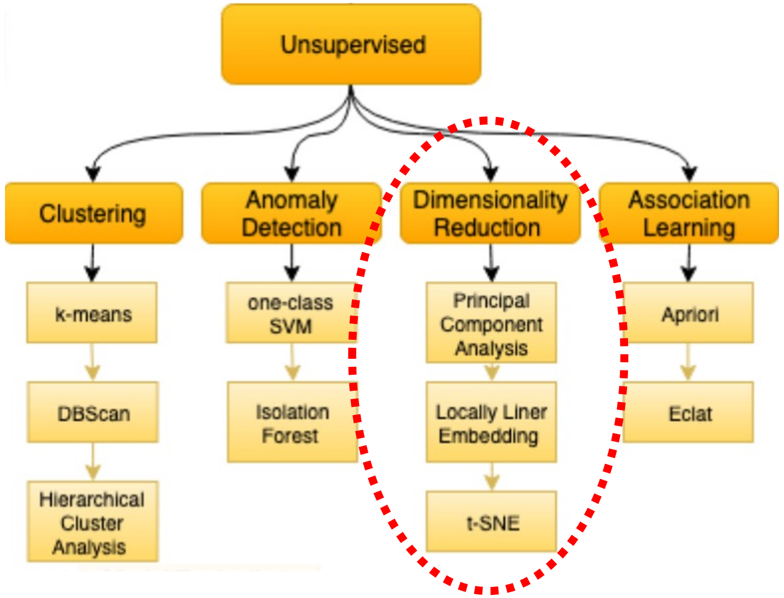
Ioanna Miliou, PhD

Senior Lecturer, Stockholm University

Stockholms universitet

Unsupervised learning

Experience: objects for which **no class labels** have been given
Performance: typically concerns the ability to output useful **characterizations** (or groupings) of objects



Stockholms universitet

Dimensions

Features (attributes)

Class label

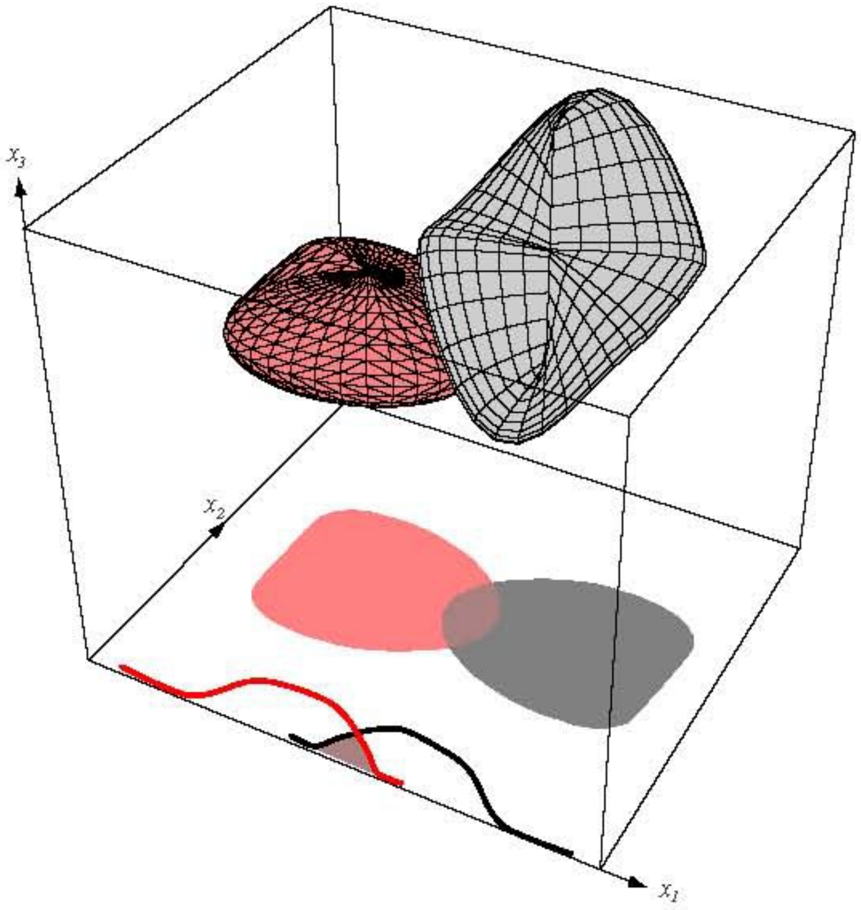
Examples (observations)

Email	All caps	No. excl. marks	Missing date	No. digits in From:	Image fraction	Spam
e1	yes	0	no	3	0	yes
e2	yes	3	no	0	0.2	yes
e3	no	0	no	0	1	no
e4	no	4	yes	4	0.5	yes
e5	yes	0	yes	2	0	no
e6	no	0	no	0	0	no

Stockholms universitet

Data Dimensionality

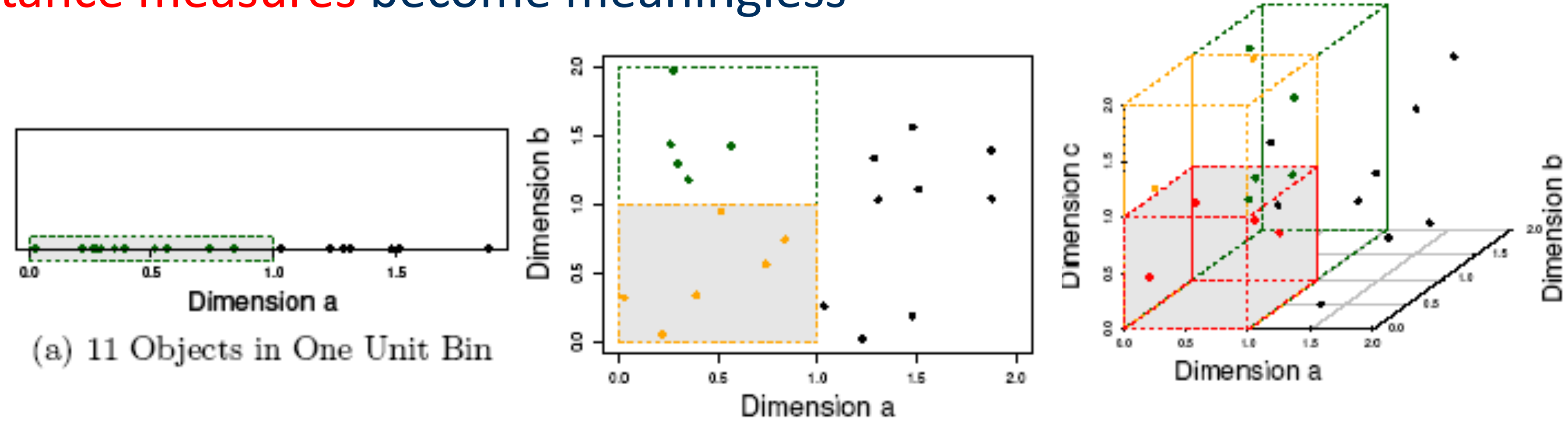
- From a theoretical point of view, increasing the number of features should lead to better performance
- In practice, the inclusion of more features leads to worse performance (i.e., the curse of dimensionality)
- Need an exponential number of training examples as dimensionality increases
- Index structures fail as the dimensionality of the data increases



Stockholms universitet

The Curse of Dimensionality

- Data in only one dimension is relatively packed
- Adding a dimension “stretches” the points across that dimension, making them further apart
- Adding more dimensions will make the points further apart—high dimensional data is extremely sparse
- Distance measures become meaningless



(a) 11 Objects in One Unit Bin

(b) 6 Objects in One Unit Bin

(c) 4 Objects in One Unit Bin

Data Dimensionality

- Significant improvements can be achieved by first mapping the data into a lower-dimensional space:

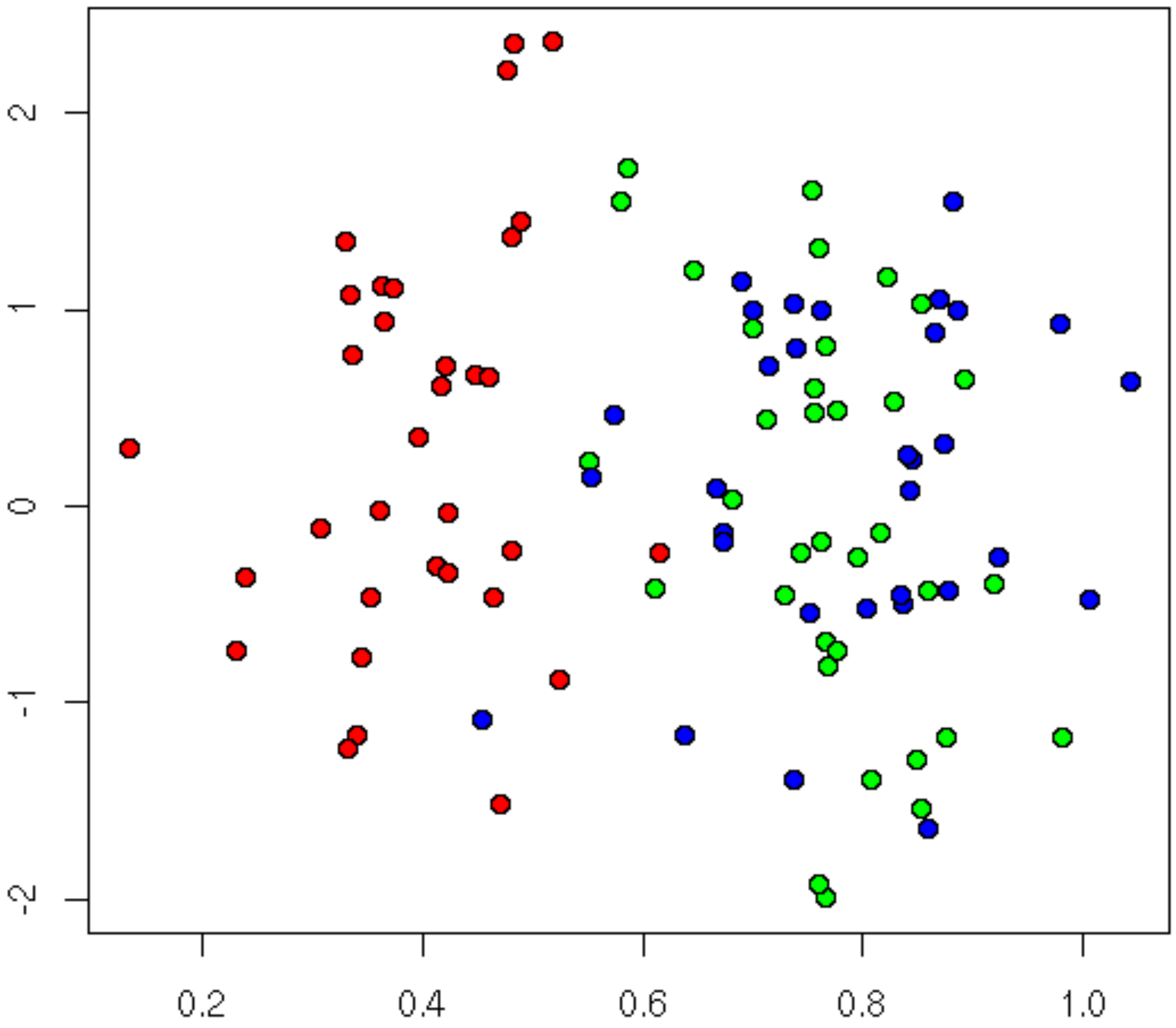
$$x = \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_N \end{bmatrix} --> reduce\ dimensionality --> y = \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_K \end{bmatrix} \quad (K << N)$$

- Dimensionality can be reduced by:
  - Combining features (linearly or non-linearly)
  - Selecting a subset of features (i.e., feature selection)
- We will focus on combining features

Data Understanding

Name	Solubility	No. C atoms	Fraction of rotatable bonds	Topol. diam.	Geom. diam.	LogP	No. heavy bonds	…
methylpentane	good	6	0.40	4	3.46	2.44	5	…
methylcyclohexene	good	7	0	4	3.00	2.51	7	…
nonene	med.	9	0.75	8	6.93	3.53	8	…
hexadiene	good	6	0.60	5	4.36	2.14	5	…
butadiene	good	4	0.33	3	2.65	1.36	3	…
naphthalene	good	10	0	5	3.61	2.84	11	…
acenaphthylene	good	12	0	5	3.58	3.32	14	…
pyrene	poor	16	0	7	5.00	4.58	19	…
dimethylanthracene	poor	16	0	7	5.29	4.61	18	…
hexahydropyrene	med.	16	0	7	5.00	3.82	19	…
triphenylene	poor	18	0	7	5.00	5.15	21	…
benzo(e)pyrene	poor	20	0	7	5.29	5.64	24	…

Plot the Data: two variables



Stockholms universitet

Dimensionality Reduction

- Statistical methods that provide information about point scatters in multivariate space
- Simplify complex relationships between cases and/or variables
- Make it easier to identify patterns
- Remove correlation between features

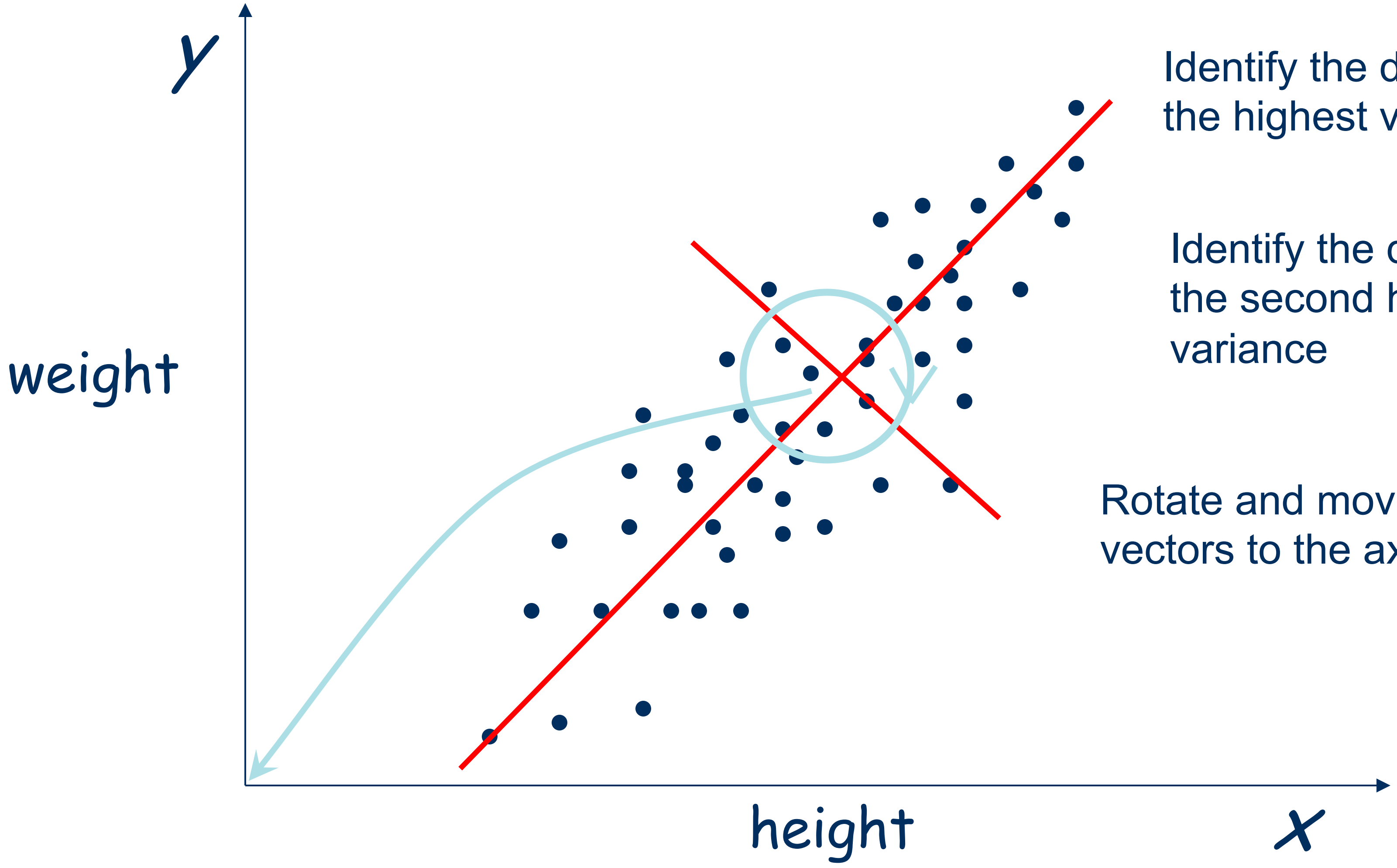
Why reduce the dimensionality?

- Better presentation than ordinal axes
- Do we need a 100-dimensional space to view the data?
- **Question:** How to find the “best” low dimensional space that conveys maximum useful information?
- **One answer:** Find “Principal Components”

The goal

- We wish to explain/summarize the underlying variance-covariance structure of a large set of variables
- Use a few linear combinations of these variables

Imagine a two-dimensional scatter of points that show a high degree of correlation …



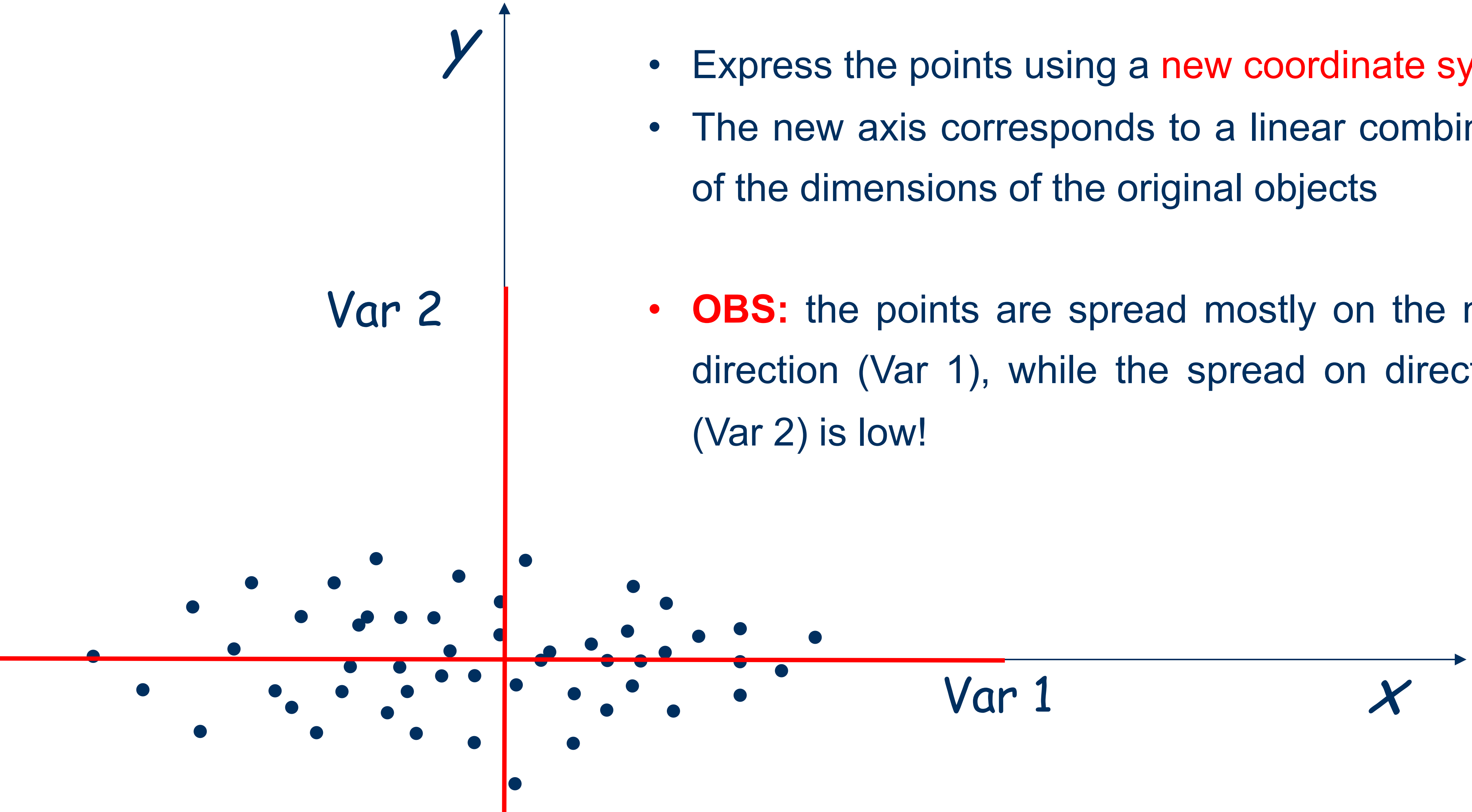
Identify the direction of the highest variance

Identify the direction of the second highest variance

Rotate and move the two vectors to the axis center

Stockholms universitet

Imagine a two-dimensional scatter of points that show a high degree of correlation …



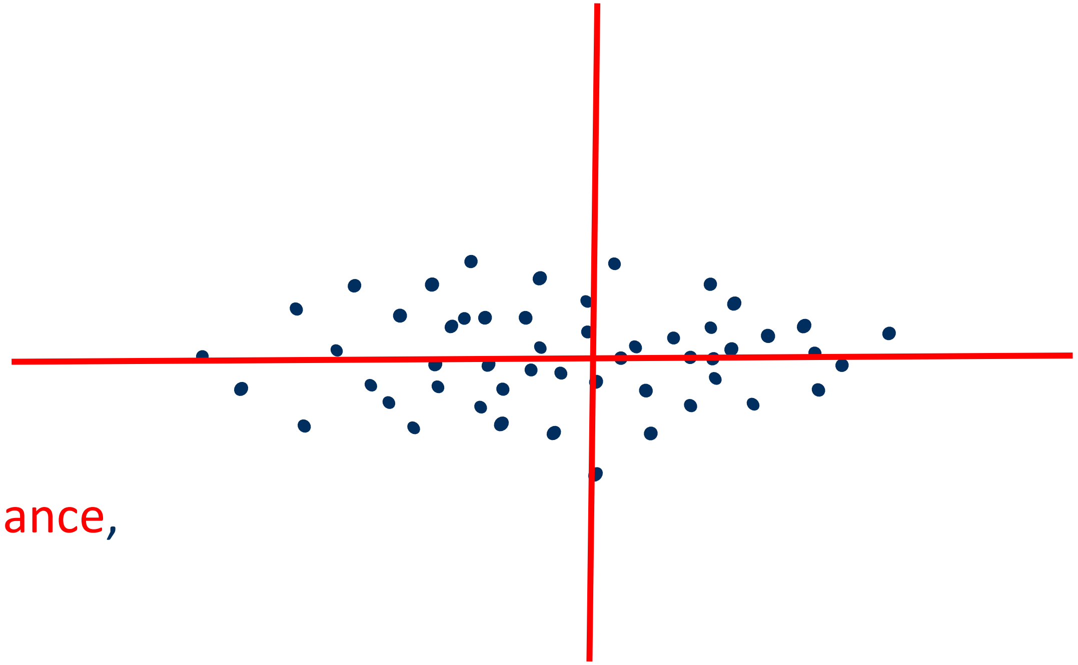
- Express the points using a new coordinate system
- The new axis corresponds to a linear combination of the dimensions of the original objects

- **OBS:** the points are spread mostly on the new x direction (Var 1), while the spread on direction y (Var 2) is low!

Stockholms universitet

What does this mean?

- More “efficient” description
  - 1st Var captures max. variance
  - 2nd Var captures the max. amount of residual variance, at right angles (orthogonal) to the first

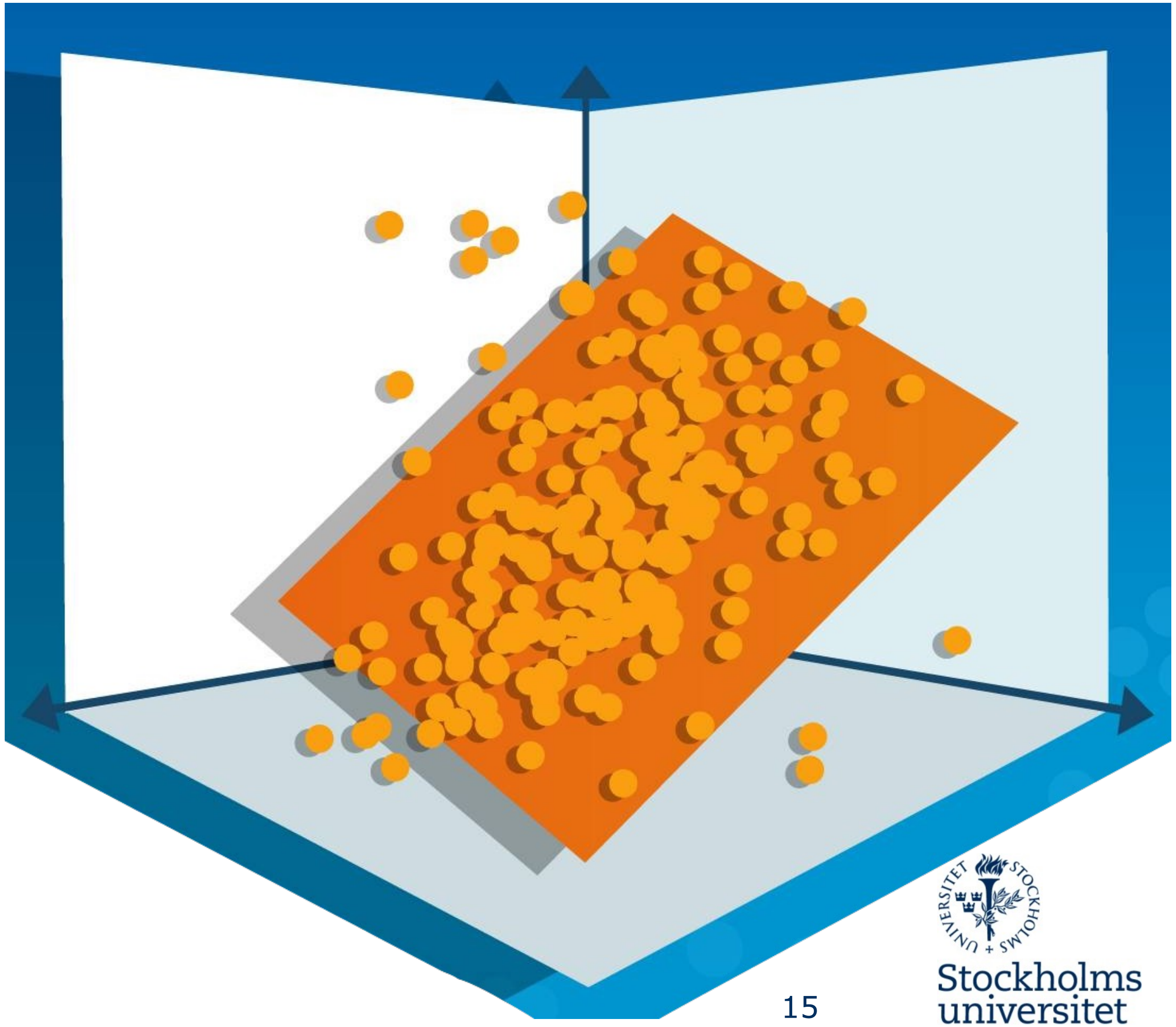


- The 1st Var may capture so much of the information content in the original data set that we can ignore the remaining axis

Stockholms universitet

Principal Components Analysis (PCA)

- Why:
  - clarify relationships among variables
  - clarify relationships among objects
- When:
  - significant correlations exist among variables
- How:
  - define new axes (components)
  - examine **correlation** between axes and variables
  - find **scores** of objects on new axes



Stockholms universitet

Philosophy of PCA

- We typically have a data matrix of M observations on N correlated variables

$$\boldsymbol{x} = \ \boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_N}$$

- PCA looks for a transformation of x_i into N new variables $\boldsymbol{y} = \boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_N}$ that are uncorrelated
- Dimensionality reduction implies information loss
- PCA preserves as much information as possible, that is, it minimizes the error $||\mathbf{x} - \mathbf{y}||$

Stockholms universitet

PCA: output

- New variables b_i that are linear combination of the original variables (x_i):

$$b_i = u_{i1}x_1 + u_{i2}x_2 + \ldots u_{iN}x_N \quad , i=1\ldots N$$

- The new variables b_i are derived in decreasing order of importance

- They are called "Principal Components"

PCA: more formally

From N original variables: $x_1,x_2,...,x_N$:

Produce N new variables: $b_1,b_2,...,b_N$:

$$b_1 = u_{11}x_1 + u_{12}x_2 + ... + u_{1N}x_N$$

$$b_2 = u_{21}x_1 + u_{22}x_2 + ... + u_{2N}x_N$$

...

$$b_N = u_{N1}x_1 + u_{N2}x_2 + ... + u_{NN}x_N$$

such that:

u_N's are uncorrelated (orthogonal)

u_1 explains as much as possible of the original variance in the data set

u_2 explains as much as possible of the remaining variance

...

Stockholms universitet

PCA: Covariance Matrix

- **Question:** How should we determine the “best” lower dimensional space?

- **Answer:**

The “best” low-dimensional space can be determined by the “best” eigenvectors of the covariance matrix of the data (i.e., the eigenvectors corresponding to the “largest” eigenvalues – also called “Principal components”)

Stockholms universitet

PCA: Covariance Matrix

$$\mathbf{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}$$

$$\operatorname{cov}(X_i, X_j) = \mathrm{E}\big[(X_i - \mu_i)(X_j - \mu_j)\big]$$

$$\Sigma = \begin{bmatrix} \mathrm{E}[(X_1-\mu_1)(X_1-\mu_1)] & \mathrm{E}[(X_1-\mu_1)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_1-\mu_1)(X_n-\mu_n)] \\ \mathrm{E}[(X_2-\mu_2)(X_1-\mu_1)] & \mathrm{E}[(X_2-\mu_2)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_2-\mu_2)(X_n-\mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{E}[(X_n-\mu_n)(X_1-\mu_1)] & \mathrm{E}[(X_n-\mu_n)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_n-\mu_n)(X_n-\mu_n)] \end{bmatrix}.$$

$$\operatorname{cov}(X, Y) = \frac{1}{n} \sum_{i=1}^{n} (x_i - E(X))(y_i - E(Y))$$

Eigenvectors

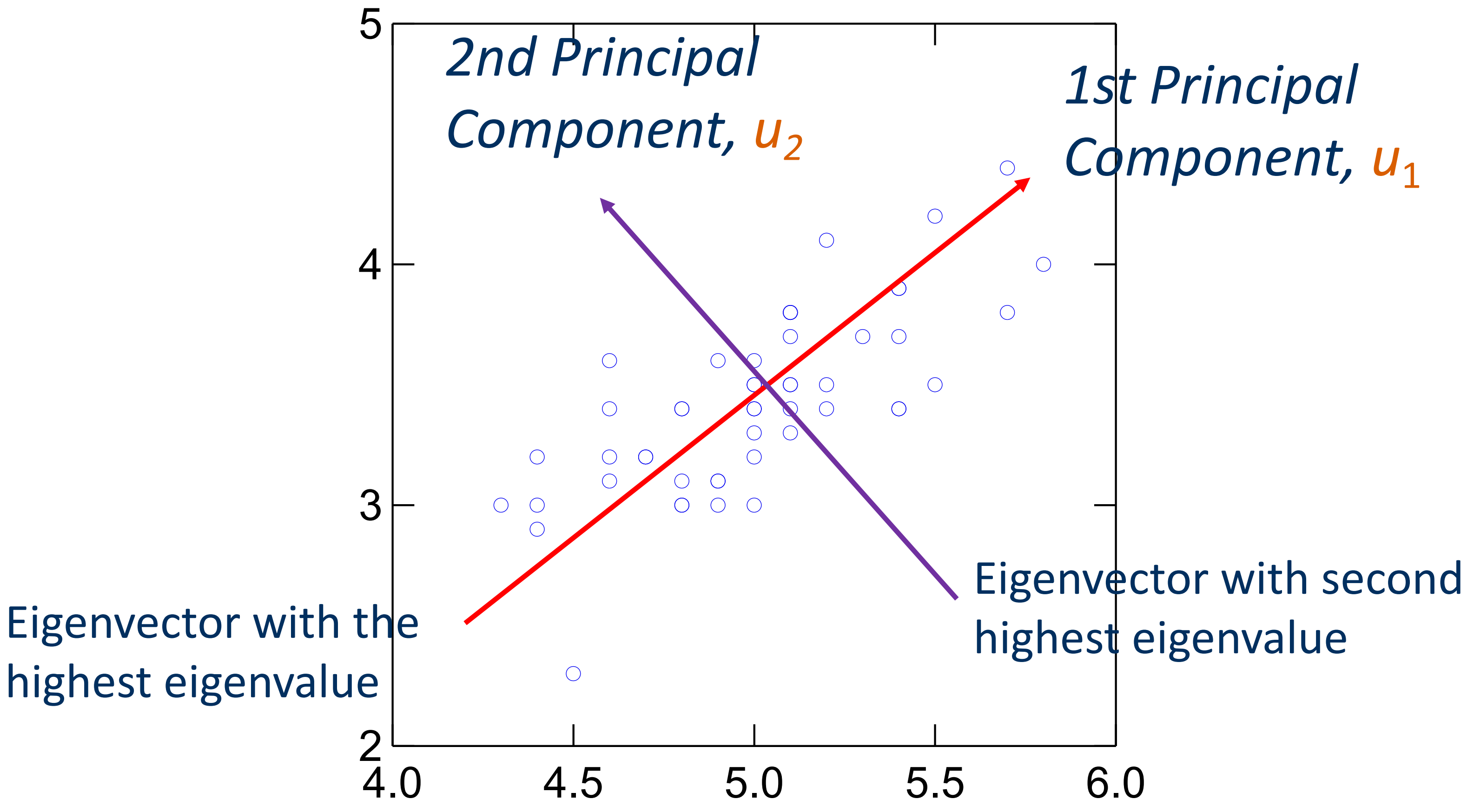
$\{u_{11}, u_{12}, \ldots, u_{1N}\}$: 1st **eigenvector** of the covariance matrix, and **coefficients** of the 1st principal component

$\{u_{21}, u_{22}, \ldots, u_{2N}\}$: 2nd **eigenvector** of the covariance matrix, and **coefficients** of the 2nd principal component

…

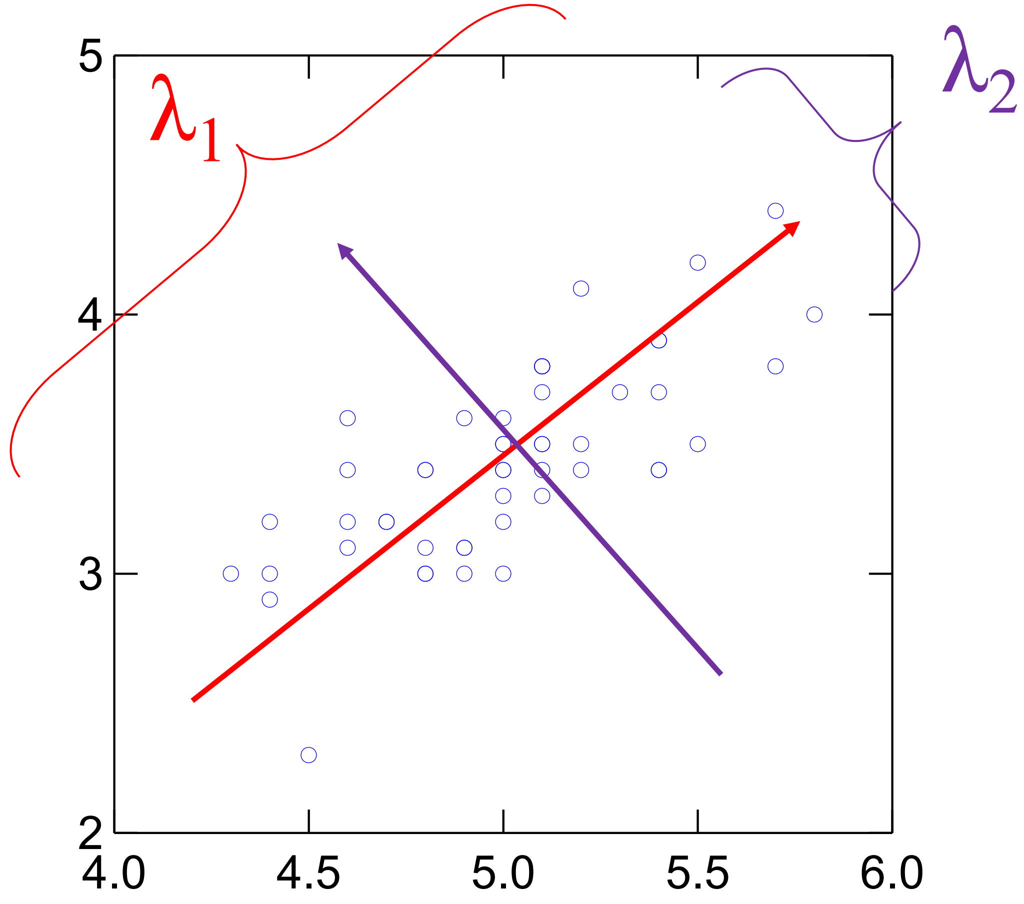
$\{u_{N1}, u_{N2}, \ldots, u_{NN}\}$: Nth **eigenvector** of the covariance matrix, and **coefficients** of the Nth principal component

Singular Value Decomposition (SVD)



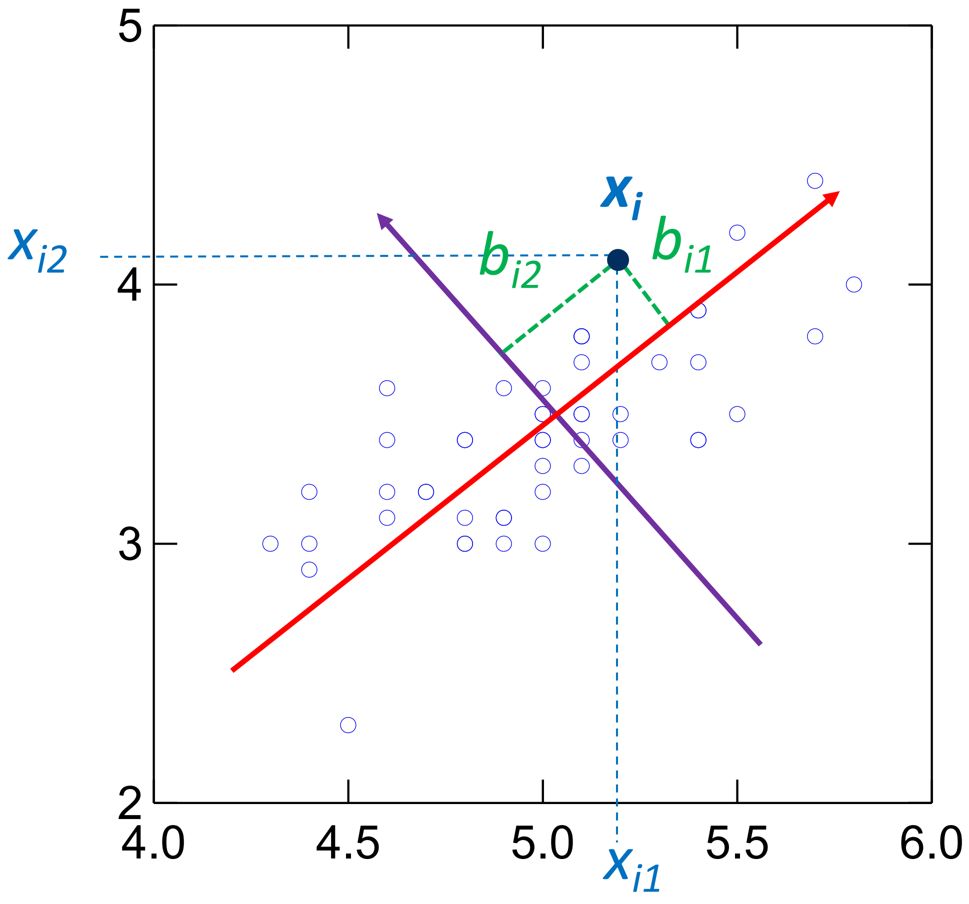
Stockholms universitet

PCA Eigenvalues



Stockholms universitet

PCA Scores



Stockholms universitet

PCA Steps

<u>Step 1:</u> $\bar{x} = \frac{1}{M} \sum_{i=1}^{M} x_i$

<u>Step 2:</u> subtract the mean: $\Phi_i = x_i - \bar{x}$ (i.e., center at zero)

<u>Step 3:</u> form the matrix $A = [\Phi_1 \ \Phi_2 \cdots \Phi_M]$ (NxM matrix), then compute:

$$C = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T = AA^T$$

(sample **covariance** matrix, NxN, characterizes the *scatter* of the data)

<u>Step 4:</u> compute the eigenvalues of C: $\boldsymbol{\lambda}_1 > \boldsymbol{\lambda}_2 > \cdots > \boldsymbol{\lambda}_N$

<u>Step 5:</u> compute the eigenvectors of C: $u_1, u_2, \ldots, u_N$

PCA Steps (cont'd)

- Since C is symmetric, $u_1, u_2, \ldots, u_N$ form a basis, (i.e., any vector x or actually $(x - \bar{x})$, can be written as a linear combination of the eigenvectors):

$$x - \bar{x} = b_1 u_1 + b_2 u_2 + \cdots + b_N u_N = \sum_{i=1}^{N} b_i u_i \quad \text{, where} \quad b_i = \frac{(x - \bar{x}).u_i}{(u_i.u_i)}$$

Step 6: (**dimensionality reduction step**) keep only the terms corresponding to the K largest eigenvalues:

$$\hat{x} - \bar{x} = \sum_{i=1}^{K} b_i u_i \text{ where } K << N$$

- The representation of $\hat{x} - \bar{x}$ into the basis $u_1, u_2, \ldots, u_K$ is thus

$$\begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_K \end{bmatrix}$$

PCA: Linear Transformation

- Every object **x** in the original space is mapped as follows:

$$\begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_K \end{bmatrix} = \begin{bmatrix} u_1^T \\ u_2^T \\ \dots \\ u_K^T \end{bmatrix} (x - \bar{x}) = U^T (x - \bar{x})$$

Step 4: Singular Value Decomposition (SVD)*

$$A = U \quad S \quad V^T = \begin{bmatrix} \vec{u}_1 & \vec{u}_2 & \cdots & \vec{u}_r \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_r \end{bmatrix} \begin{bmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_r \end{bmatrix}$$

$$[M \times r][r \times r]\,[r \times N]$$

- **r** : rank of matrix A
- $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r$: singular values (square roots of eig-values AA^T, A^TA)
- $\vec{u}_1, \vec{u}_2, \cdots, \vec{u}_r$: left singular vectors (eig-vectors of AA^T)
- $\vec{v}_1, \vec{v}_2, \cdots, \vec{v}_r$: right singular vectors (eig-vectors of A^TA)

$$A = \lambda_1 \vec{u}_1 \vec{v}_1^T + \lambda_2 \vec{u}_2 \vec{v}_2^T + \cdots + \lambda_r \vec{u}_r \vec{v}_r^T$$

* Compact SVD

Example SVD*

- Initial matrix:

$$A = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}$$

- Transpose:

$$A^T = \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix}$$

- Then compute:

$$AA^T = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix}$$

- Find the eigenvectors and eigenvalues!

* Find the step-by-step solution here: https://datajobs.com/data-science-repo/SVD-Tutorial-%5BKirk-Baker%5D.pdf

Example SVD

- Matrix:

$$AA^T = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix}$$

- Finding eigenvectors and eigenvalues:

$$\begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Solve the above system of equations!

Example SVD

- System of equations:

$$\begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- After solving it, we get:

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- After normalizing:

$$U = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$$

Example SVD

- In a similar way we get **V**:

$$V^T = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{-5}{\sqrt{30}} \end{bmatrix}$$

- We have also recorded the eigenvalues of **U** and **V** (they are the same!)

$$S = \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \end{bmatrix}$$

Example SVD

- So finally, we have:

$$A_{mn} = U_{mm} S_{mn} V_{nn}^T = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{-5}{\sqrt{30}} \end{bmatrix} =$$

$$\begin{bmatrix} \frac{\sqrt{12}}{\sqrt{2}} & \frac{\sqrt{10}}{\sqrt{2}} & 0 \\ \frac{\sqrt{12}}{\sqrt{2}} & \frac{-\sqrt{10}}{\sqrt{2}} & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & 0 \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{30}} & \frac{-5}{\sqrt{30}} \end{bmatrix} = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}$$

More about SVD:

https://datajobs.com/data-science-repo/SVD-Tutorial-%5BKirk-Baker%5D.pdf

How to choose K?

- Choose **K** using the following criterion:

$$\frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{N} \lambda_i} > Threshold \quad \text{(e.g., 0.9 or 0.95)}$$

- In this case, we say that we "preserve" 90% or 95% of the information in the data

- If K=N, then we "preserve" 100% of the information in the data

Normalization

- The principal components are dependent on the *units* used to measure the original variables as well as on the *range* of values they assume

- Data should always be normalized prior to using PCA

- A common normalization method is to transform all the data to have zero mean and unit standard deviation:

$$\frac{x_i - \mu}{\sigma}$$

(μ *and* σ *are the mean and standard deviation of* x_i*'s*)

Stockholms universitet

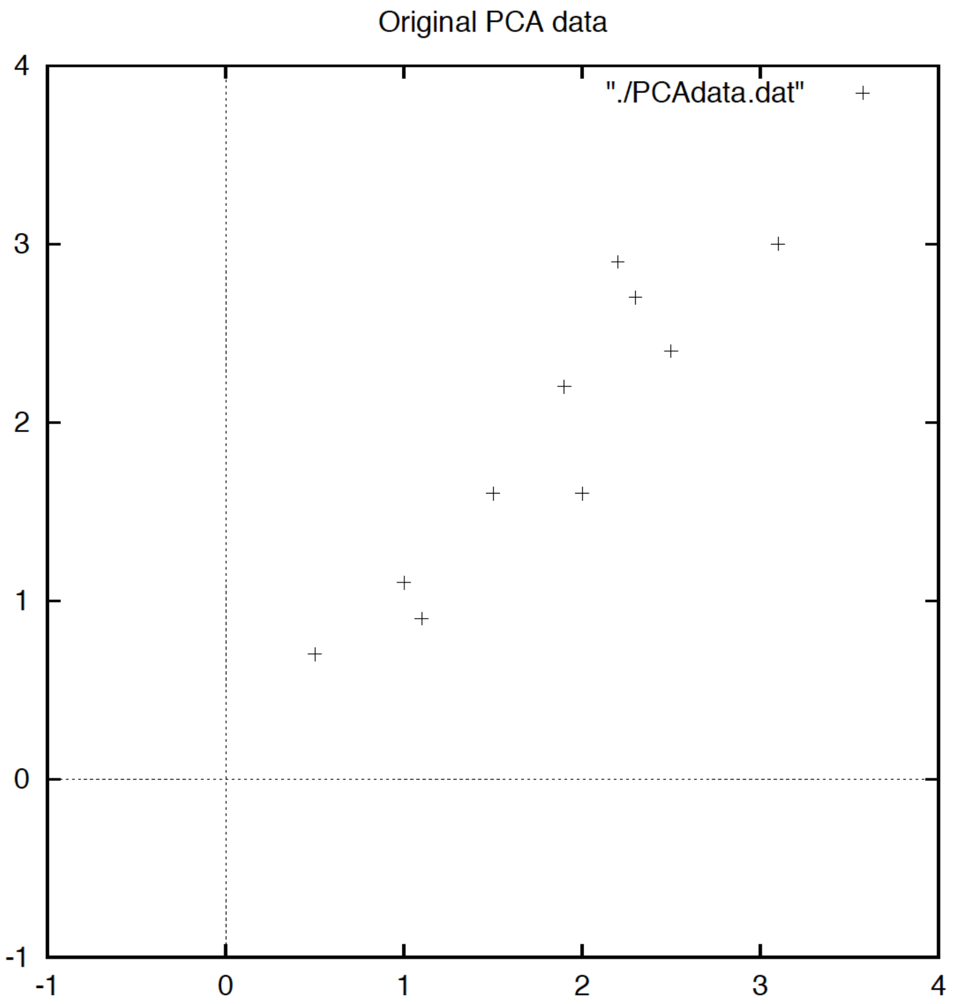
PCA Summary

- Rotates a multivariate dataset into a new configuration which is easier to interpret

- Purpose:
  - simplify data
  - look at relationships between variables
  - identify patterns in the correlated variables

Example PCA

$$\text{Data} = \begin{array}{c|c} 2.5 & 2.4 \\ 0.5 & 0.7 \\ 2.2 & 2.9 \\ 1.9 & 2.2 \\ 3.1 & 3.0 \\ 2.3 & 2.7 \\ 2 & 1.6 \\ 1 & 1.1 \\ 1.5 & 1.6 \end{array}$$



Example PCA

- Subtract the mean (can also divide by standard deviation)

Data =

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

DataAdjust =

x	y
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01

Example PCA

- Compute the covariance matrix:

$$cov = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

- Compute the eigenvalues and eigenvectors:

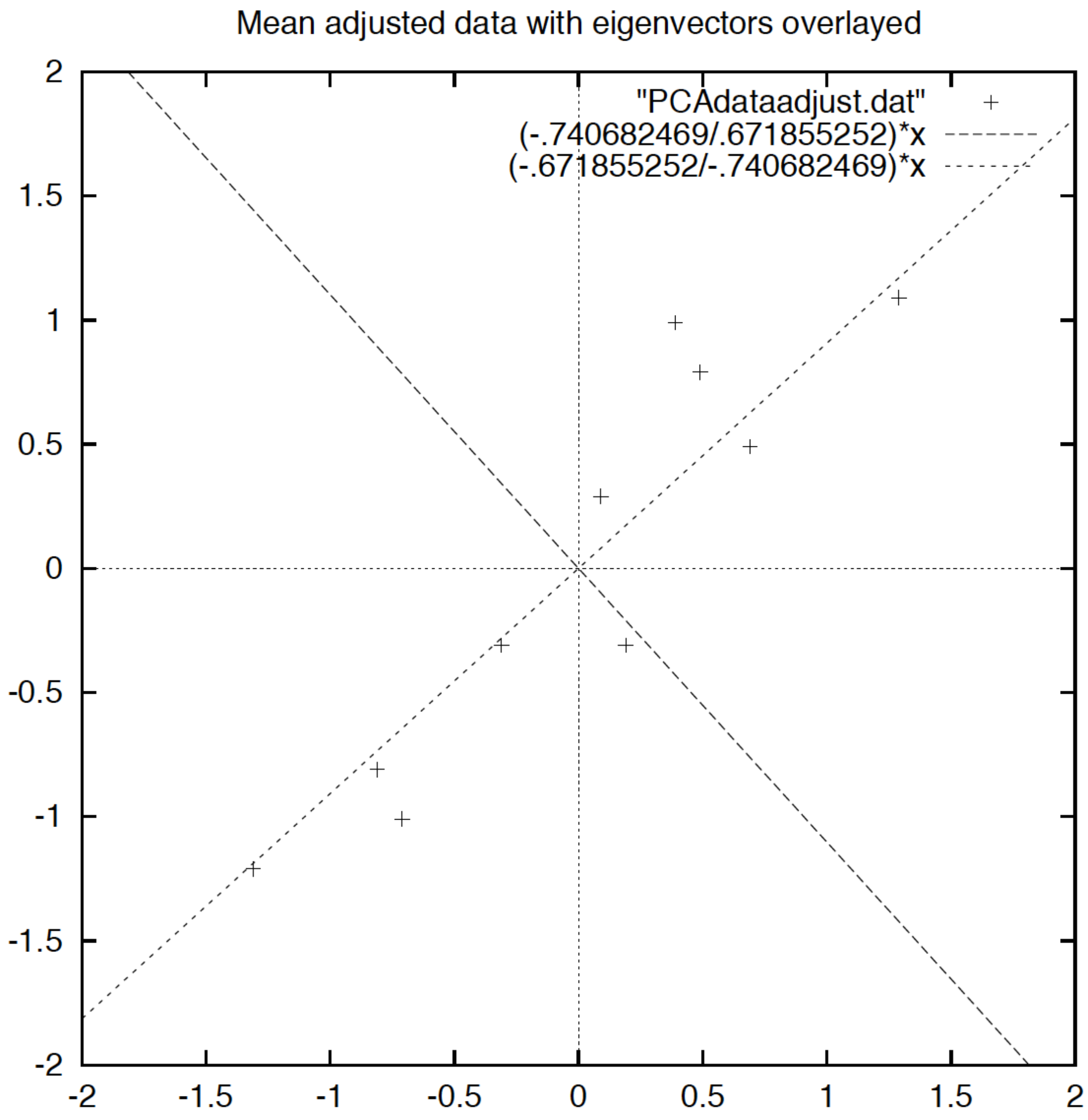
$$eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

Example PCA

- Plot

$$\begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$



Stockholms universitet

Example PCA

- Choose eigenvectors from:

$$FeatureVector = (eig_1\ eig_2\ eig_3\ ....\ eig_n)$$

- Choose both eigenvectors:

$$\begin{pmatrix} -.677873399 & -.735178656 \\ -.735178656 & .677873399 \end{pmatrix}$$

- Or just the first:

$$\begin{pmatrix} -.677873399 \\ -.735178656 \end{pmatrix}$$

Example PCA

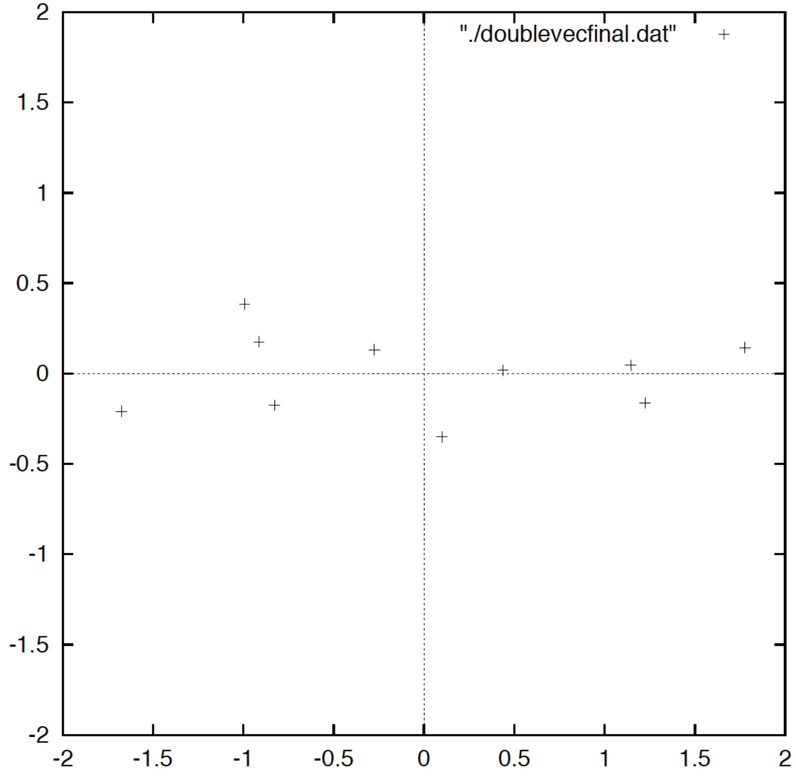
- Derive the final data:

$$FinalData = RowFeatureVector \times RowDataAdjust$$

- Hence, using both eigenvectors:

Transformed Data=

x	y
-.827970186	-.175115307
1.77758033	.142857227
-.992197494	.384374989
-.274210416	.130417207
-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287



Stockholms universitet

Example PCA

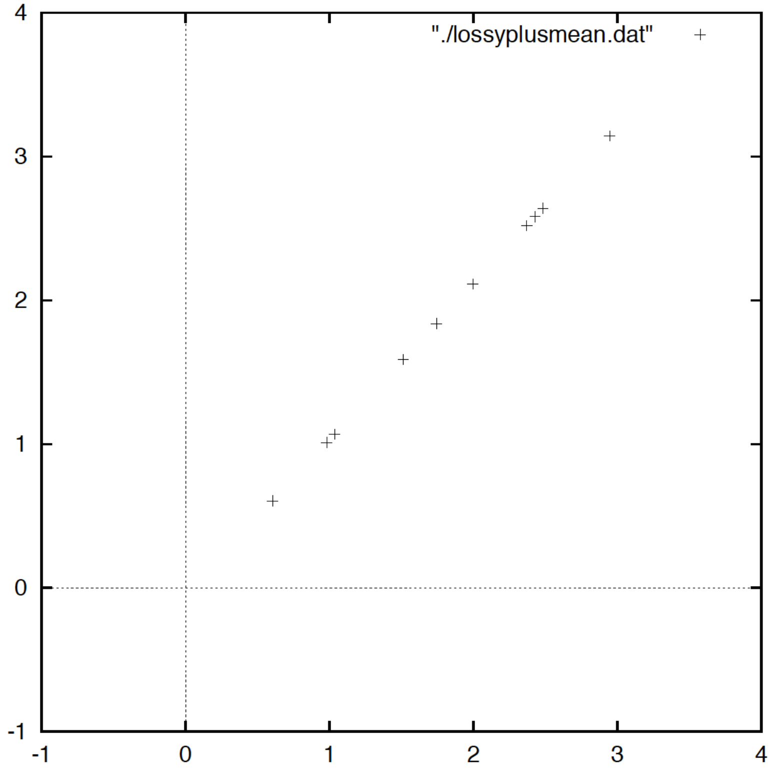
- Derive the final data:

$$FinalData = RowFeatureVector \times RowDataAdjust$$

- And using one eigenvector:

Transformed Data (Single eigenvector)

x
-.827970186
1.77758033
-.992197494
-.274210416
-1.67580142
-.912949103
.0991094375
1.14457216
.438046137
1.22382056

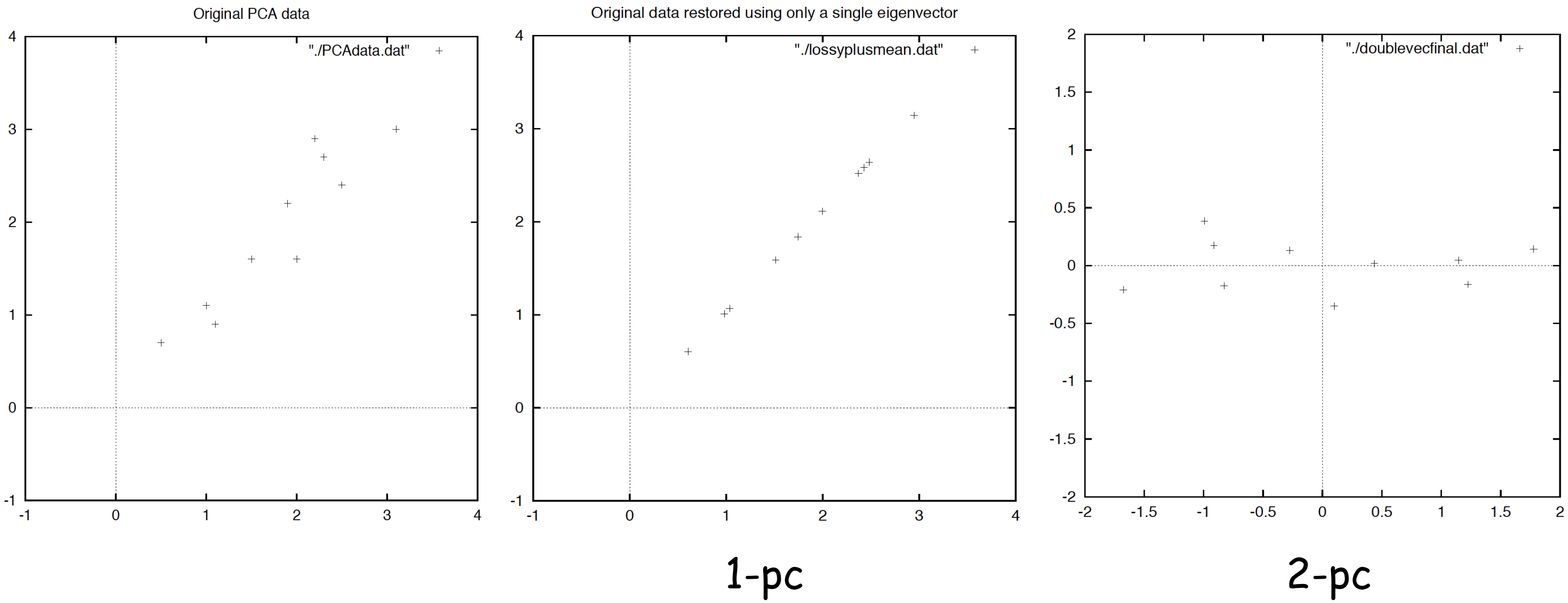


Stockholms universitet

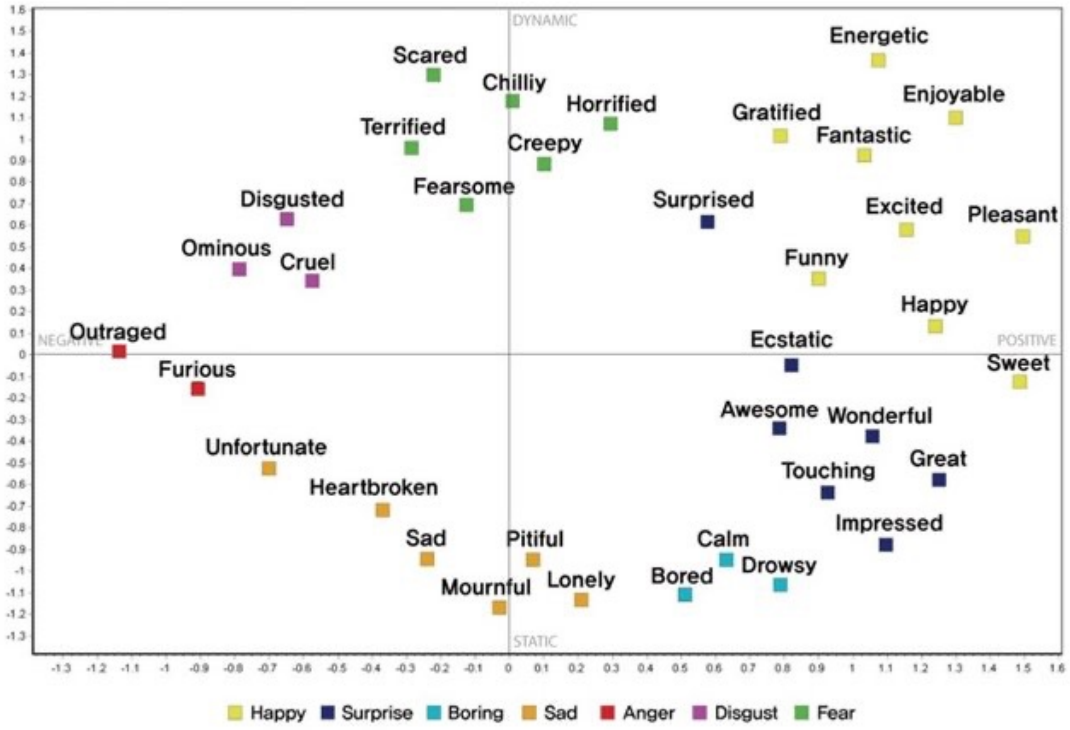
Example PCA

- Getting back the original data:

$$RowOriginalData = (RowFeatureVector^T \times FinalData) + OriginalMean$$



Multi-Dimensional Scaling (MDS)



- So far, we assumed that we know both data points **X** and distance matrix **D** between these points
- What if the original points **X** are not known but only distance matrix **D** is known?
- Can we reconstruct **X** or some approximation of **X**?

Stockholms universitet

Problem

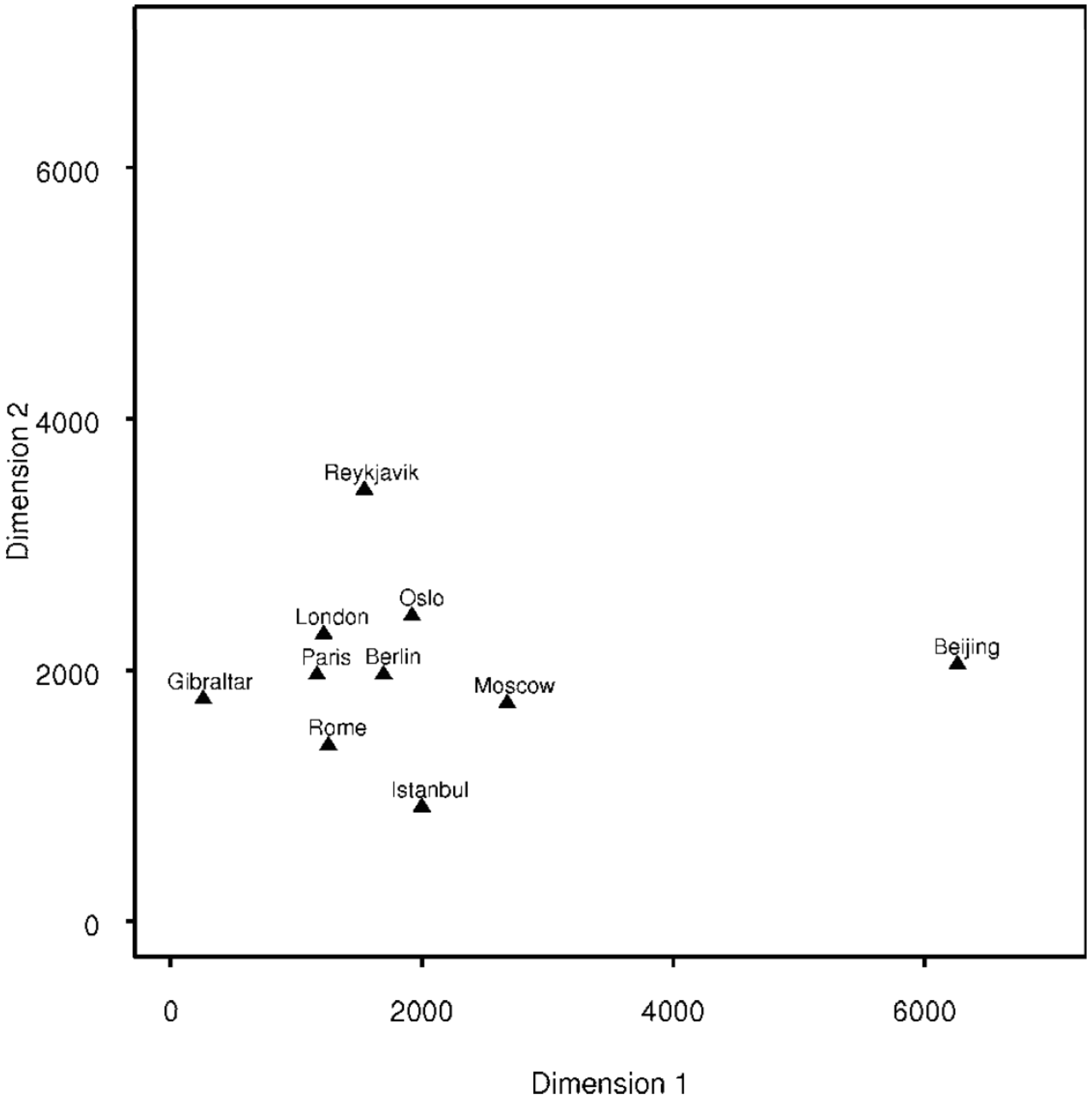
- Given distance matrix **D** between **n** points
- Find a **k**-dimensional representation of every x_i point **i**
- So that $d(x_i,x_j)$ is as close as possible to $D(i,j)$

Why do we want to do that?

Distances between 10 cities

	London	Berlin	Oslo	Moscow	Paris	Rome	Beijing	Istanbul	Gibraltar	Reykjavik
London	–									
Berlin	570	–								
Oslo	710	520	–							
Moscow	1550	1000	1020	–						
Paris	210	540	830	1540	–					
Rome	890	730	1240	1470	680	–				
Beijing	5050	4570	4360	3600	5100	5050	–			
Istanbul	1550	1080	1520	1090	1040	850	4380	–		
Gibraltar	1090	1450	1790	2410	960	1030	6010	1870	–	
Reykjavik	1170	1480	1080	2060	1380	2040	4900	2560	2050	–

Distances between 10 cities



Stockholms universitet

Financial Indicators of Countries

Country	Increase	Life	IMR	TFR	GDP
Albania	1.2	69.2	30	2.9	659.91
Argentina	1.2	68.6	24	2.8	4343.04
Australia	1.1	74.7	7	1.9	17529.98
Austria	1.0	73.0	7	1.5	20561.88
Benin	3.2	45.9	86	7.1	398.21
Bolivia	2.4	57.7	75	4.8	812.19
Brazil	1.5	64.0	58	2.9	3219.22
Cambodia	2.8	50.1	116	5.3	97.39
China	1.1	66.7	44	2.0	341.31
Colombia	1.7	66.4	37	2.7	1246.87
Croatia	−1.5	67.1	9	1.7	5400.66
El Salvador	2.2	63.9	46	4.0	988.58
France	0.4	73.0	7	1.7	21076.77
Greece	0.6	75.0	10	1.4	6501.23
Guatemala	2.9	62.4	48	5.4	831.81
Iran	2.3	67.0	36	5.0	9129.34
Italy	−0.2	74.2	8	1.3	19204.92
Malawi	3.3	45.0	143	7.2	229.01
Netherlands	0.7	74.4	7	1.6	18961.90
Pakistan	3.1	60.6	91	6.2	385.59
Papua New Guinea	1.9	55.2	68	5.1	839.03
Peru	1.7	64.1	64	3.4	1674.15
Romania	−0.5	66.6	23	1.5	1647.97
USA	1.1	72.5	9	2.1	21965.08
Zimbabwe	4.4	52.4	67	5.0	686.75

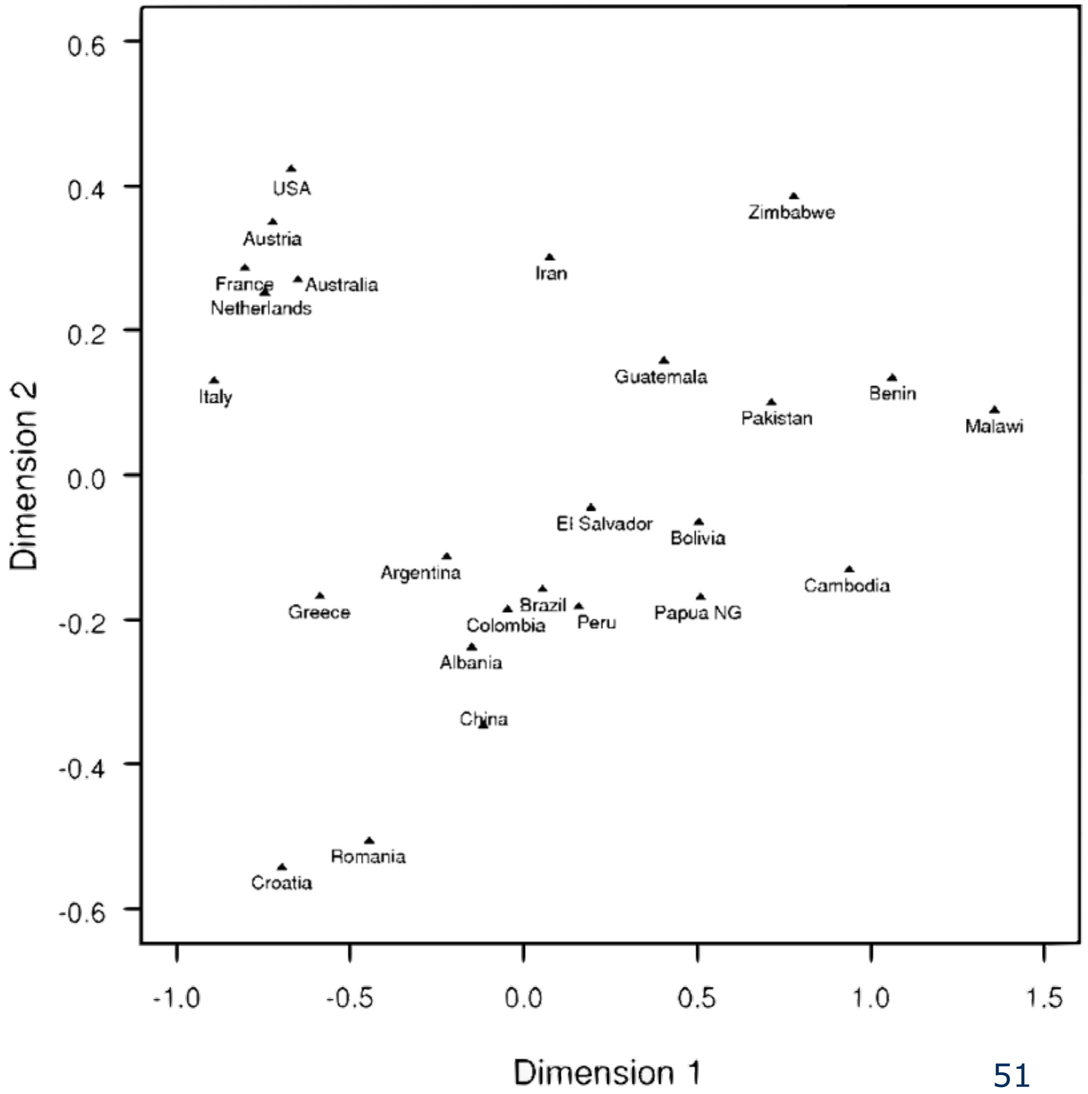
Financial Indicators of Countries: First coordinate

Dim1: Measure of overall development

Malawi	−2.027
Benin	−1.616
Cambodia	−1.414
Zimbabwe	−1.302
Pakistan	−1.133
Bolivia	−0.798
Papua New Guinea	−0.783
Guatemala	−0.706
El Salvador	−0.344
Peru	−0.277
Iran	−0.167
Brazil	−0.112
Colombia	0.036
China	0.188
Albania	0.220
Argentina	0.327
Romania	0.786
Greece	0.921
Australia	1.049
USA	1.105
Netherlands	1.158
Austria	1.164
Croatia	1.167
France	1.230

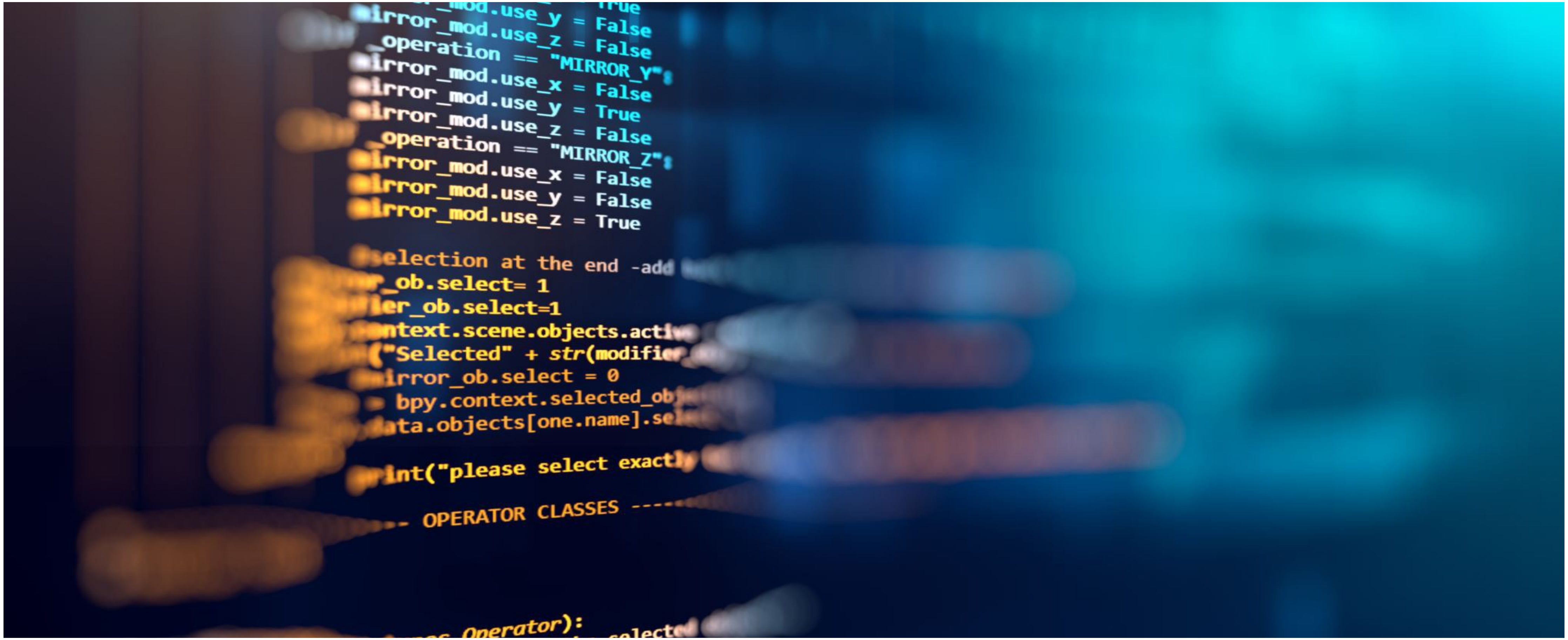
Financial Indicators of Countries: Two coordinates

Dim2: Measure GDP



Stockholms universitet

How can we do that? (Algorithm)



Stockholms universitet

High-level view of the MDS algorithm

- Randomly initialize the positions of **n** points in a **k**-dimensional space
- Compute pairwise distances **D'** for this placement
- Compare **D'** to **D**
- Move points to adjust their pairwise distances better (make **D'** closer to **D**)
- Repeat until **D'** is close to **D**

The MDS algorithm

- ***Input:*** n x n distance matrix **D**
- Random **n** points in the **k**-dimensional space **($x_1,\dots,x_n$)**
- **stop = false**
- **while not stop**
  - **totalerror = 0.0**
  - For every pair of points **i,j** compute
    - **$D'(i,j)=d(x_i,x_j)$**
    - **error = (D(i,j) - D'(i,j)) / D(i,j)**
    - **totalerror +=error**
    - **x_i = ((x_i - x_j) / D'(i,j))*error**
  - **If totalerror** small enough, **stop = true**

Stockholms universitet

Questions about MDS

- Running time of the MDS algorithm
  - **$O(n^2I)$**, where **I** is the number of iterations of the algorithm
- MDS does not guarantee that the metric property is maintained in **D'**
- Faster? Guarantee of metric property?

Today…

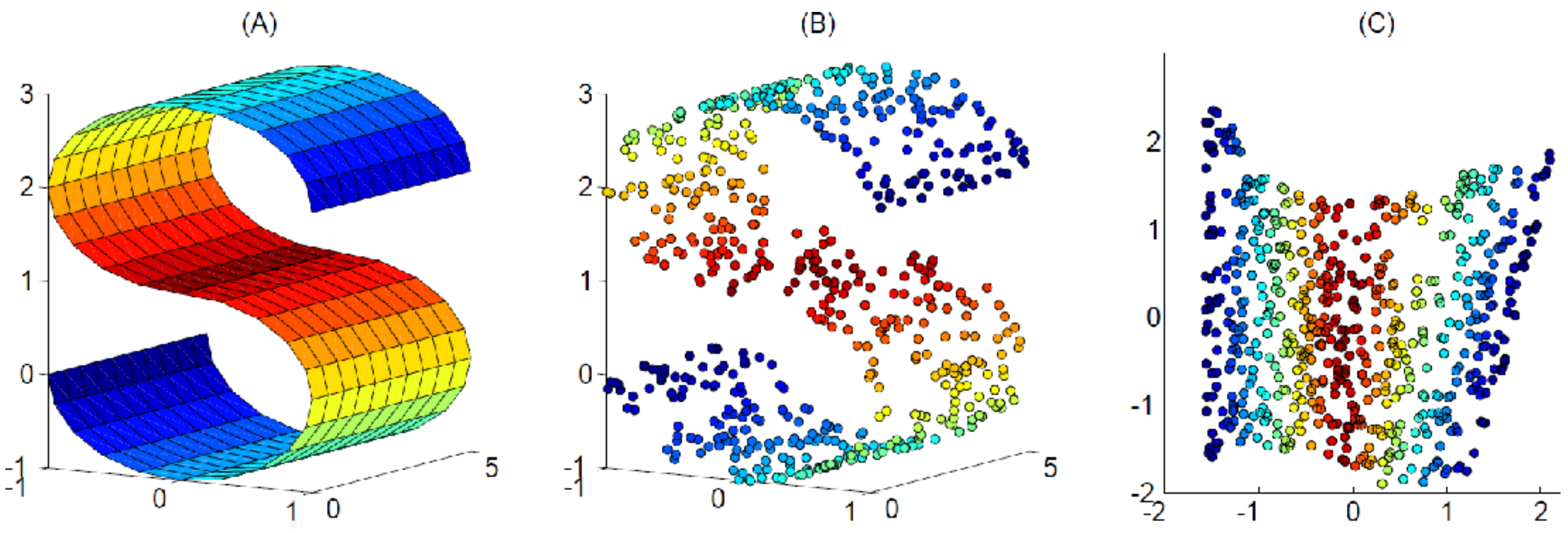
What is the **Curse of Dimensionality**?

The importance of efficient **Data Representation**

How do we **reduce** Data Dimensionality?

What is the **PCA algorithm**?

What is the **MDS algorithm**?



Stockholms universitet

TODOs

Reading:

Main course book: Chapter 6

(Sec. 6.1, 6.3, 6.5)

Lab 1

Recommended to complete the lab before the end of the week

Quiz 1

Stockholms universitet

Coming up next

Wednesday

Lab 1 – Data Exploration

Thursday

Lecture 4 – Clustering I

Next week

Sept 11

Lecture 5 – Clustering II

Sep 14

Lab 2 – Clustering

Stockholms universitet