06.09.2024

Statistical Methods in AI (CS7.403)

Lecture-10: Feature Selection, Principal Component Analysis (PCA) - 1

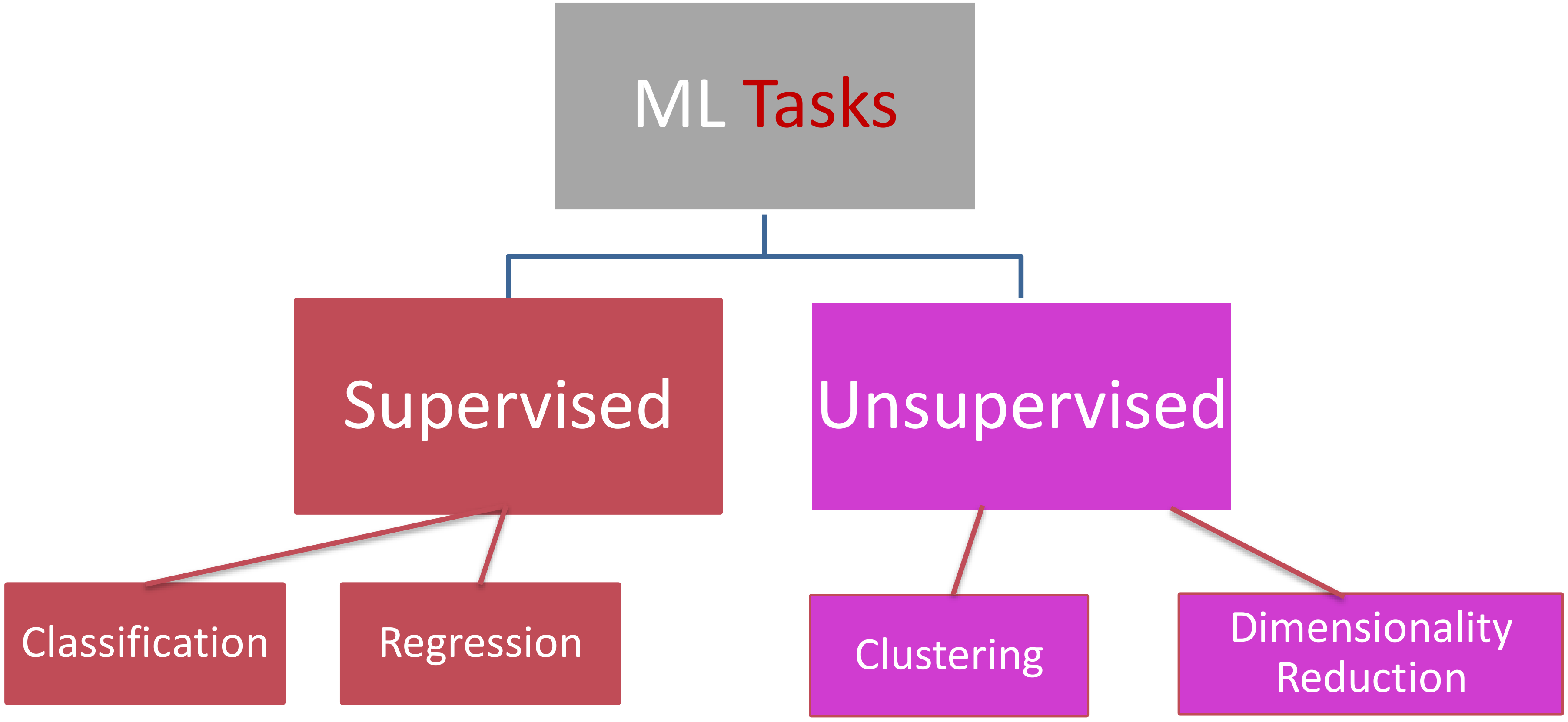
Ravi Kiran (ravi.kiran@iiit.ac.in)

https://ravika.github.io

@vikatravi

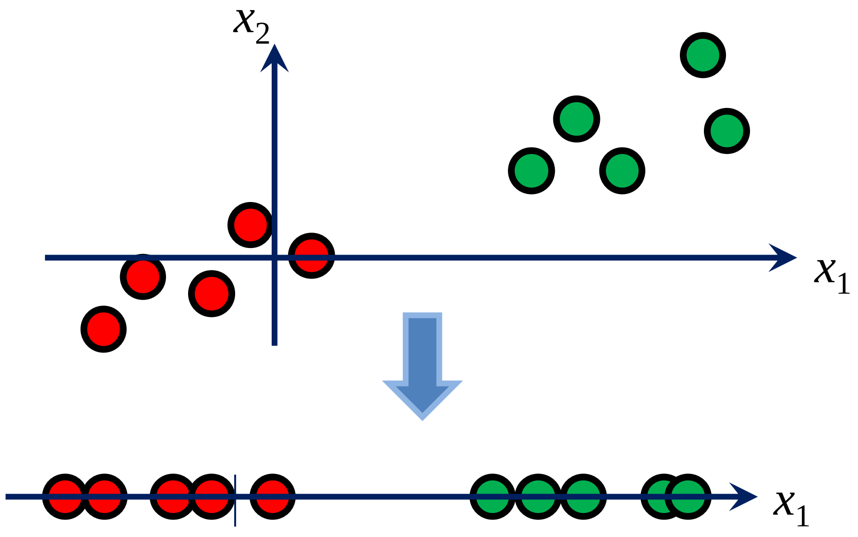
Center for Visual Information Technology (CVIT)

IIIT Hyderabad


ML Tasks
Supervised
Unsupervised
Classification
Regression
Clustering
Dimensionality Reduction

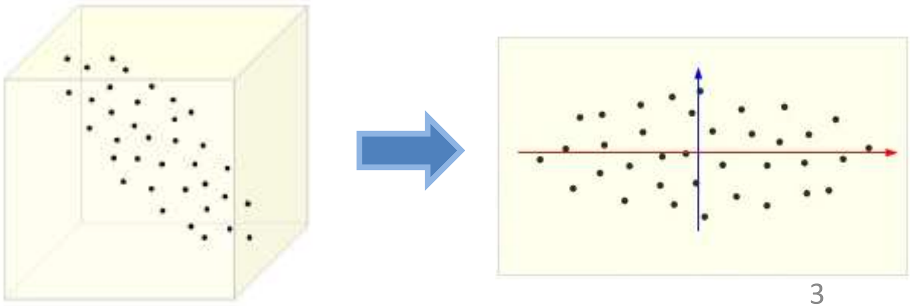
Reducing Dimensions

- Feature Selection:
  - Choose the "best" features from your data



- Feature Extraction:
  - Build derived features intended to be informative and non-redundant

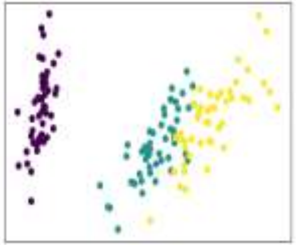
- Feature Visualization:
  - How are the 'best' features distributed in 1D/2D/3D ?



Selecting and Extracting Features

$$\begin{bmatrix} x_1 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Selecting first and third feature



$$\begin{bmatrix} x_1 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Selecting first and fourth feature

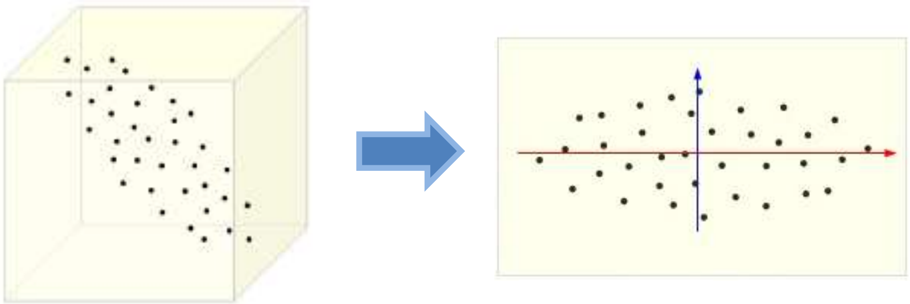
NOTE: Data samples are color-coded by their class label. But label info is <u>not used</u> for feature selection.

Selecting and Extracting Features

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.0 & 0.4 & 0.2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

New Features as linear combination of old Features

$$X' = AX$$

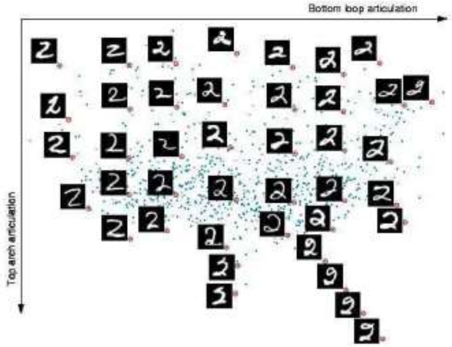


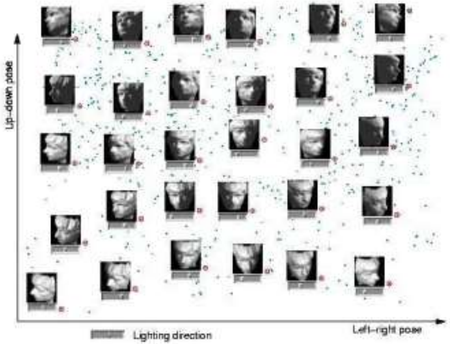
Applications for Dimensionality Reduction

- To compress data by reducing dimensionality. E.g., representing each image in a large collection as a linear combination of a small set of "template" images
  - Also sometimes called dictionary learning (can also be used for other types of data, e.g., speech signals, text-documents, etc.)

Applications for Dimensionality Reduction

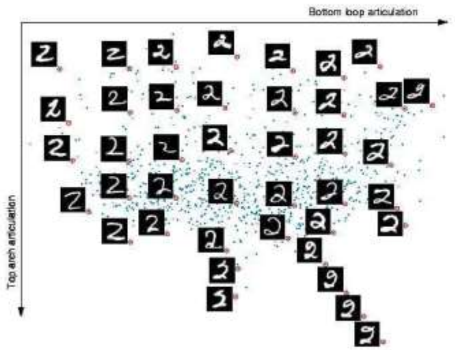
- To compress data by reducing dimensionality. E.g., representing each image in a large collection as a linear combination of a small set of "template" images
    - Also sometimes called dictionary learning (can also be used for other types of data, e.g., speech signals, text-documents, etc.)
- Visualization (e.g., by projecting high-dim data to 2D or 3D)

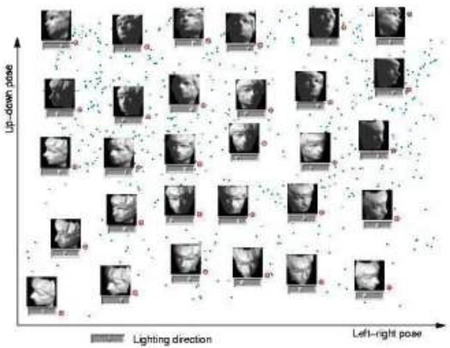




Applications for Dimensionality Reduction

- To compress data by reducing dimensionality. E.g., representing each image in a large collection as a linear combination of a small set of "template" images
    - Also sometimes called dictionary learning (can also be used for other types of data, e.g., speech signals, text-documents, etc.)
- Visualization (e.g., by projecting high-dim data to 2D or 3D)

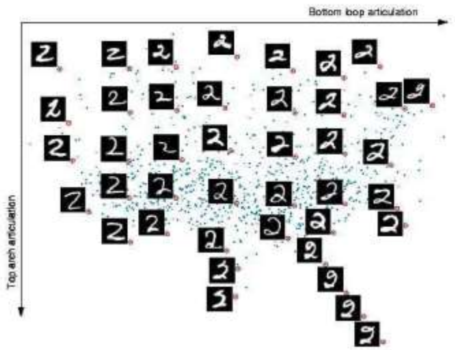


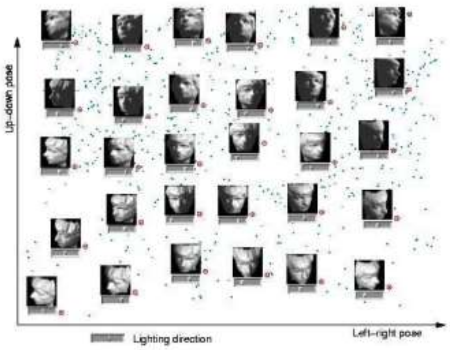


- To make learning algorithms run faster

Applications for Dimensionality Reduction

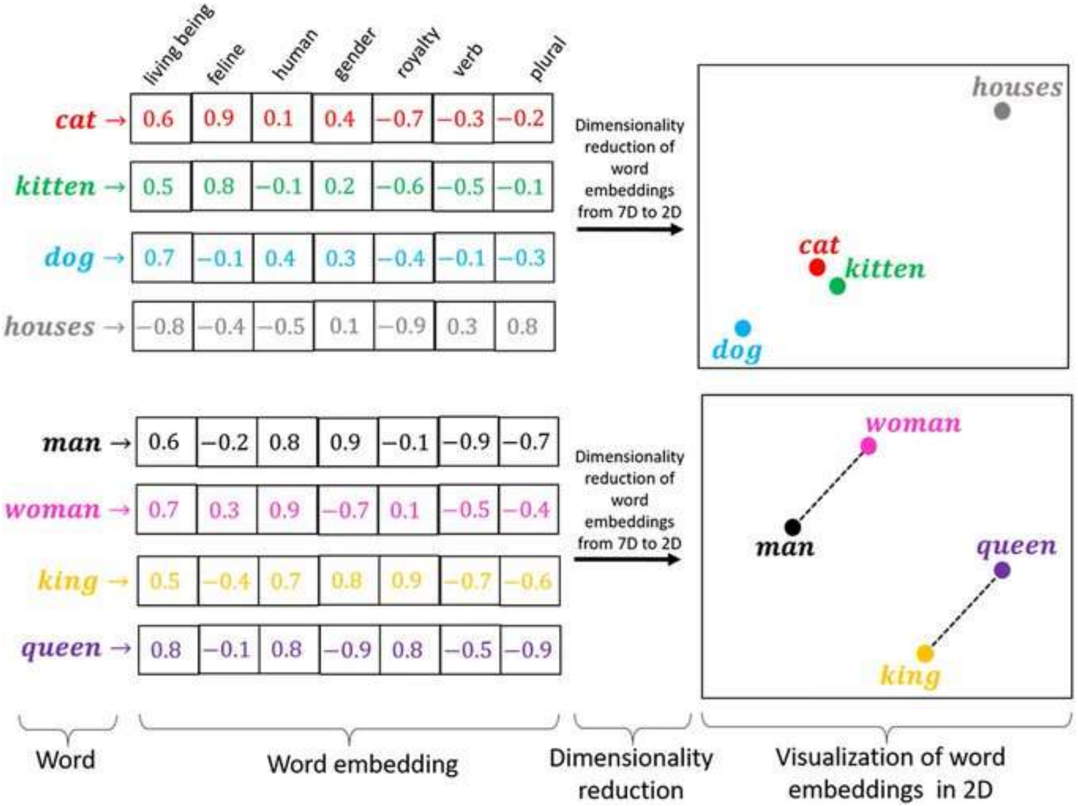
- To compress data by reducing dimensionality. E.g., representing each image in a large collection as a linear combination of a small set of "template" images
  - Also sometimes called dictionary learning (can also be used for other types of data, e.g., speech signals, text-documents, etc.)
- Visualization (e.g., by projecting high-dim data to 2D or 3D)





- To make learning algorithms run faster
- To reduce overfitting problem caused by high-dimensional data

Visualization using dimensionality reduction



https://pbs.twimg.com/media/F2uVj40aQAALw6c?format=jpg&name=small

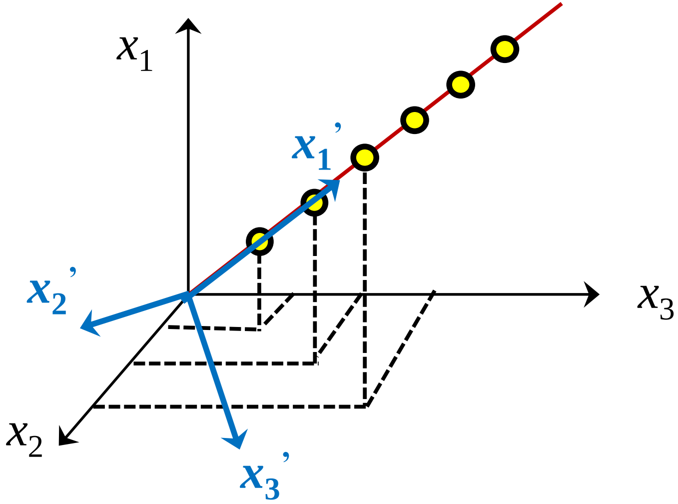
Intro to Principal Components Analysis (PCA)

Finding informative feature axes

PCA: A Toy Example

- Consider a new co-ordinate system with one axis along the line
- All co-ordinates except the first one are zeros now.

3.7	7.5	11.2	15	18.7	22.4
0	0	0	0	0	0
0	0	0	0	0	0

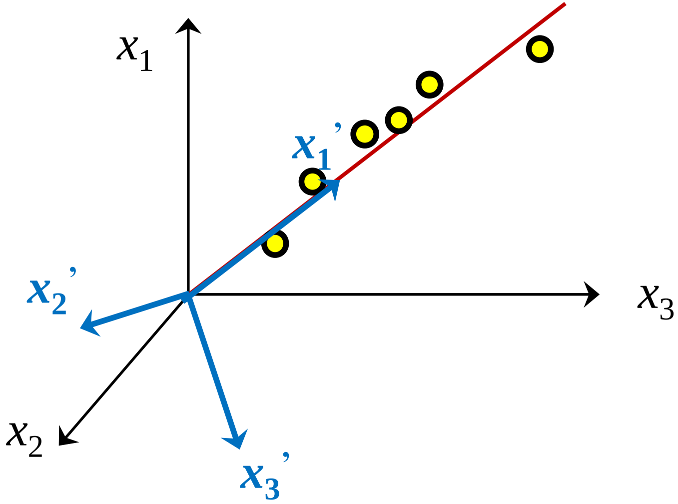


PCA: Toy Example - 2

1	4	3	5.7	5.1	2.2
2	7.9	5.8	12	9.9	4.1
3.1	12	9	18	15	6.3

3.61	7.4	11.1	15.0	18.4	22.4
0.2	0.4	0.9	0.7	0.8	0.3
0.1	0.1	0.1	0.1	0.1	0.1

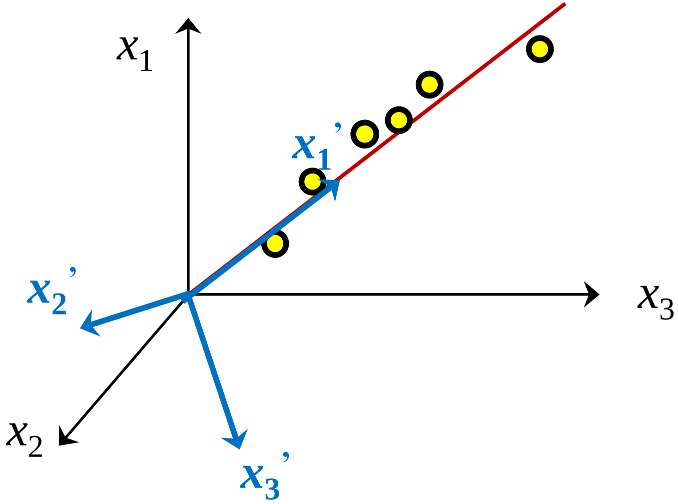
NOTE: These values are made up. Not exact.



PCA: Toy Example - 2

1	4	3	5.7	5.1	2.2
2	7.9	5.8	12	9.9	4.1
3.1	12	9	18	15	6.3

3.61	7.4	11.1	15.0	18.4	22.4

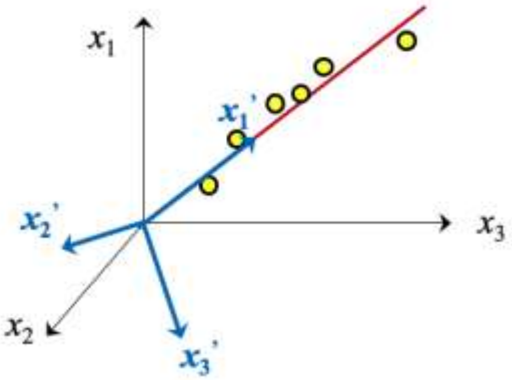


PCA strategy

- Construct new features that are **good** alternative representation of the original features

1	4	3	5.7	5.1	2.2
2	7.9	5.8	12	9.9	4.1
3.1	12	9	18	15	6.3

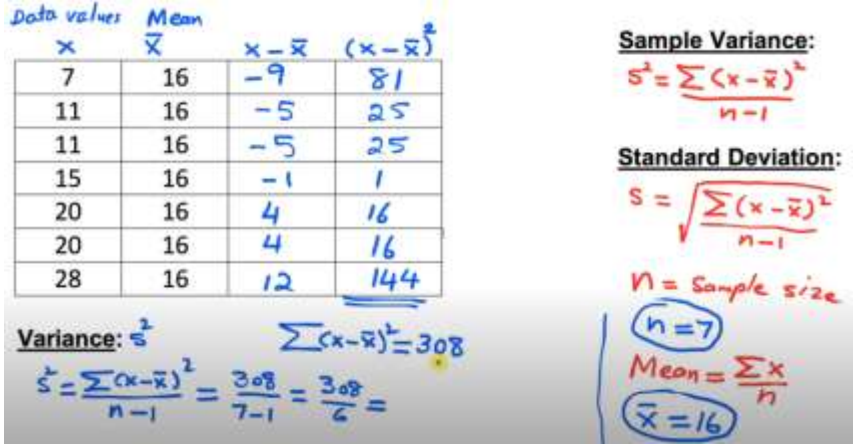
3.61	7.4	11.1	15.0	18.4	22.4



PCA strategy

- Construct new features that are **good** alternative representation of the original features
  - Good => Capture as much of original variation as possible

Variance



Data values x	Mean $\bar{x}$	$x-\bar{x}$	$(x-\bar{x})^2$
7	16	-9	81
11	16	-5	25
11	16	-5	25
15	16	-1	1
20	16	4	16
20	16	4	16
28	16	12	144

Variance: s^2 $\sum(x-\bar{x})^2 = 308$

$$s^2 = \frac{\sum(x-\bar{x})^2}{n-1} = \frac{308}{7-1} = \frac{308}{6} =$$

Sample Variance:

$$s^2 = \frac{\sum(x-\bar{x})^2}{n-1}$$

Standard Deviation:

$$s = \sqrt{\frac{\sum(x-\bar{x})^2}{n-1}}$$

n = Sample size

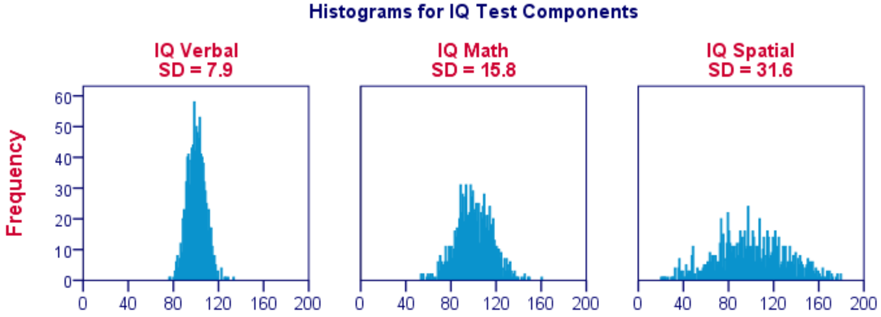
$n = 7$

$$\text{Mean} = \frac{\sum x}{n}$$

$\bar{x} = 16$

Mean = ‘Average’ value

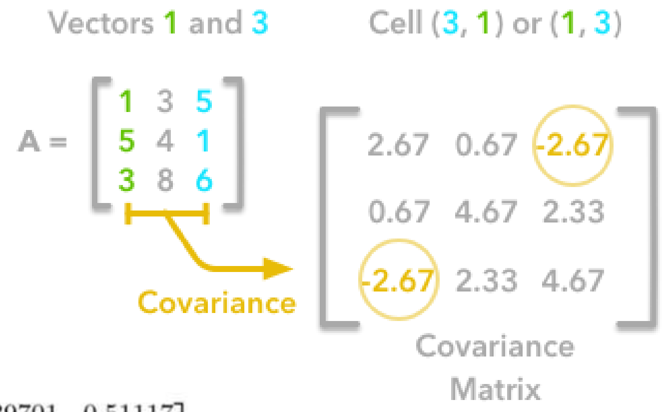
S.D = Average deviation of samples from mean



Reference: https://youtu.be/179ce7ZzFA8

Ref: https://www.spss-tutorials.com/img/standard-deviation-example-4.png

Covariance : m samples, n features



$$\begin{bmatrix} 0.39701 & 0.51117 \\ 0.55582 & 0.93003 \\ 0.59403 & 0.96645 \\ 0.51544 & 0.29759 \\ 0.85313 & 0.18118 \\ 0.88564 & 0.69114 \end{bmatrix}$$

Variance:

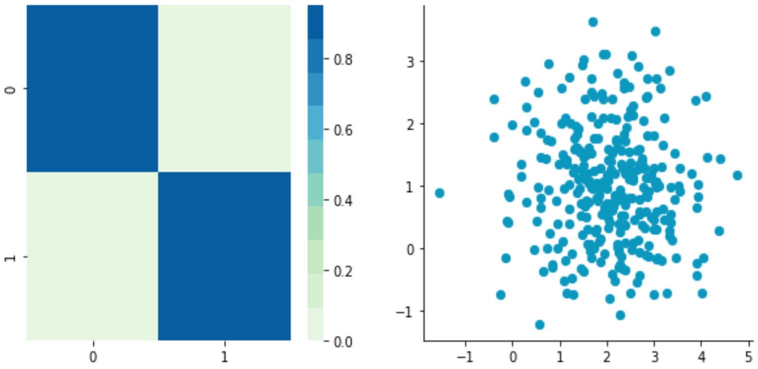
$$s^2 = \frac{\sum \left(\overline{X} - X_i\right)^2}{N}$$

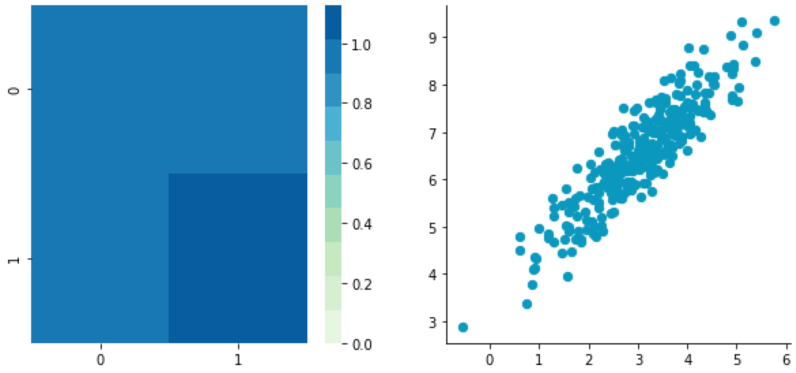
Covariance:

$$\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^\intercal$$

$$\Sigma = \frac{\sum_{i=1}^{n} x_i x_i^T}{n} - \mu\mu^T \text{ where } \mu = \frac{\sum_{i=1}^{n} x_i}{n}$$

Covariance Matrix



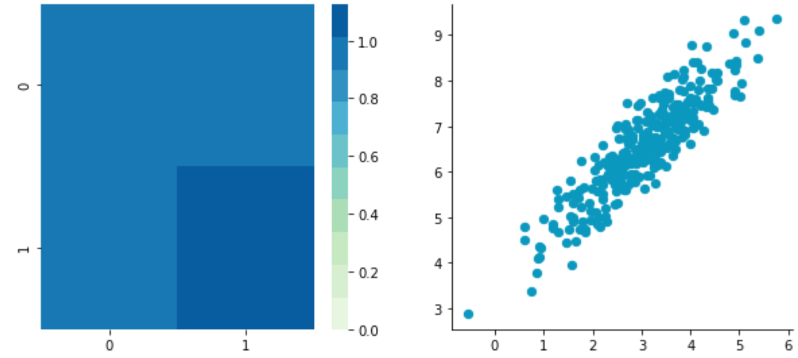


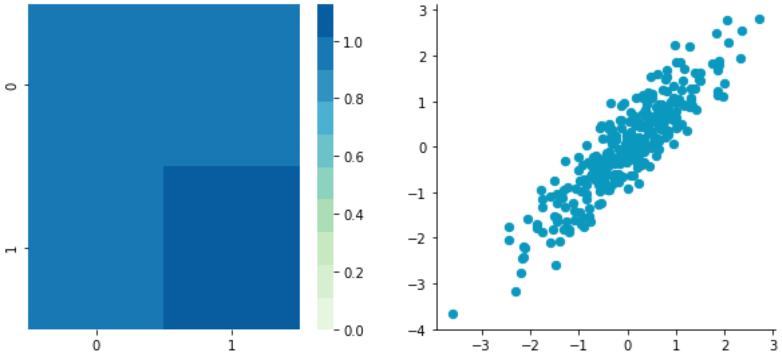
$$C = \begin{bmatrix} +0.95 & -0.04 \\ -0.04 & +0.87 \end{bmatrix}$$

$$C = \begin{bmatrix} +0.95 & +0.92 \\ +0.92 & +1.12 \end{bmatrix}$$

Mean Normalization

$$X' = X - \bar{x}$$

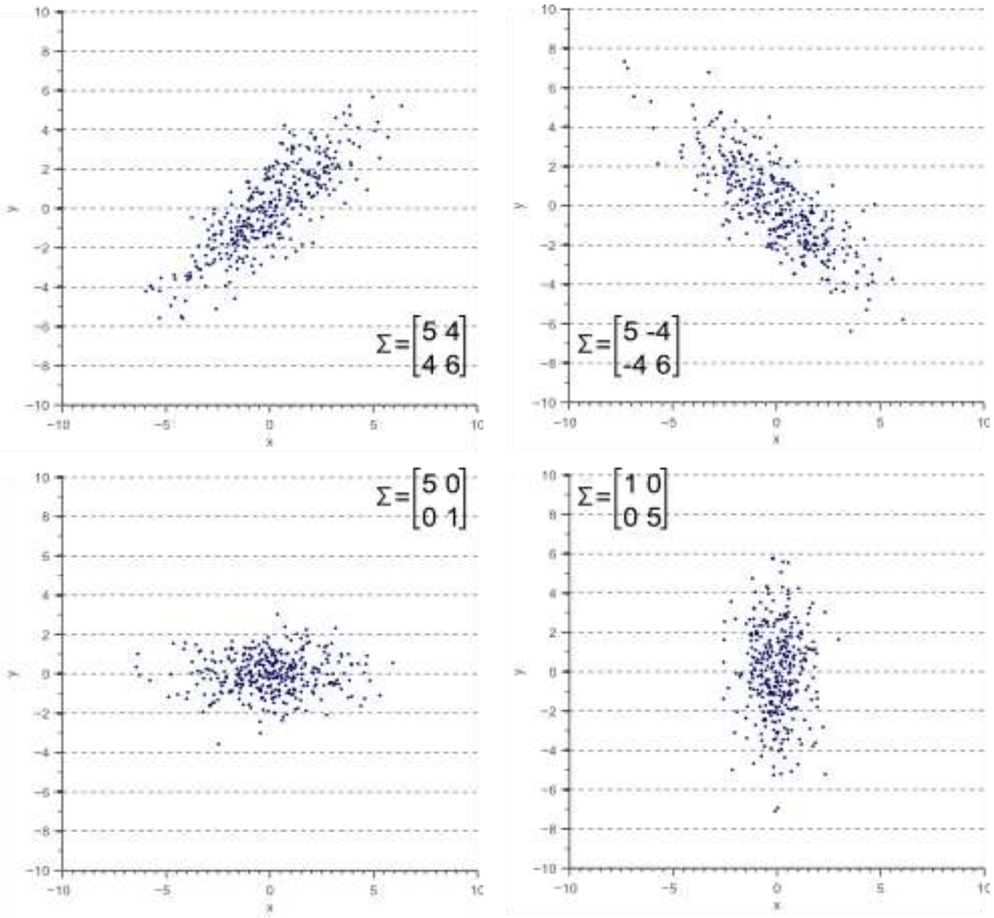




$$C = \begin{bmatrix} +0.95 & +0.92 \\ +0.92 & +1.12 \end{bmatrix}$$

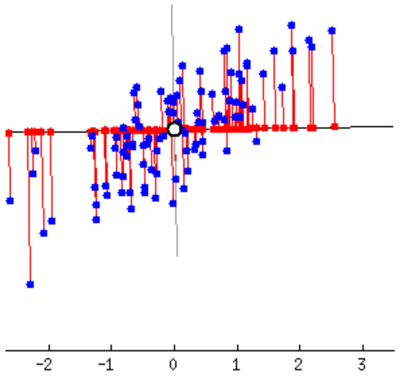
$$C = \begin{bmatrix} +0.95 & +0.92 \\ +0.92 & +1.12 \end{bmatrix}$$

Covariance Matrix encodes spread and orientation of data



PCA strategy

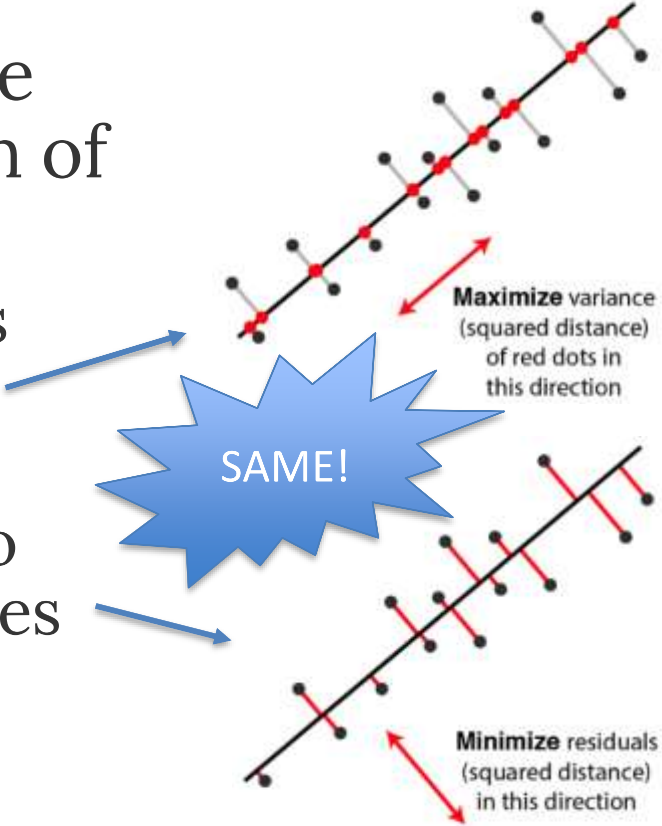
- Construct new features that are **good** alternative representation of the original features
  - Good => Capture as much of original variation as possible



http://ethen8181.github.io/machine-learning/dim_reduct/PCA.html

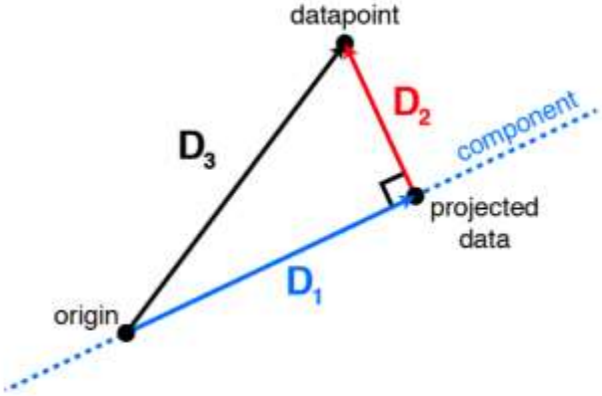
PCA strategy

- Construct new features that are **good** alternative representation of the original features
  - **good** => new features capture as much of original variation as possible
  - **good** => new features allow us to "reconstruct" the original features



http://ethen8181.github.io/machine-learning/dim_reduct/PCA.html

Maximizing variance = Minimizing reprojection error

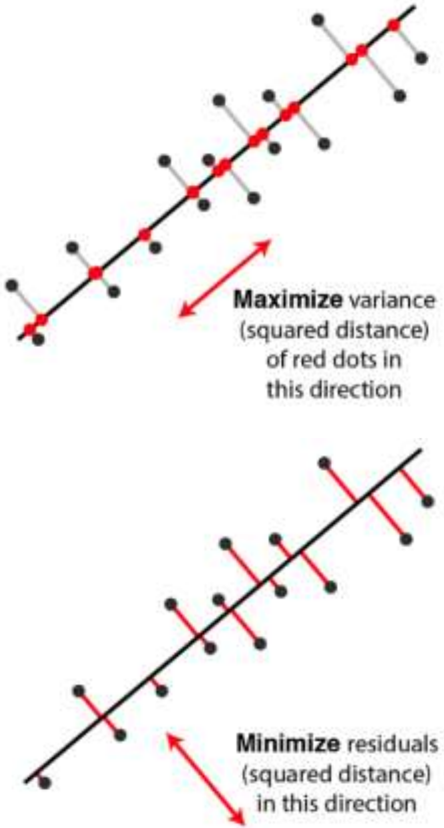


$$D_3^2 = D_1^2 + D_2^2$$

initial variance = remaining variance + lost variance

$$\|\mathbf{a}_i\|^2 = \|w_i \mathbf{c}\|^2 + \|\mathbf{a}_i - w_i \mathbf{c}\|^2$$

this is constant | maximize this **or** minimize this



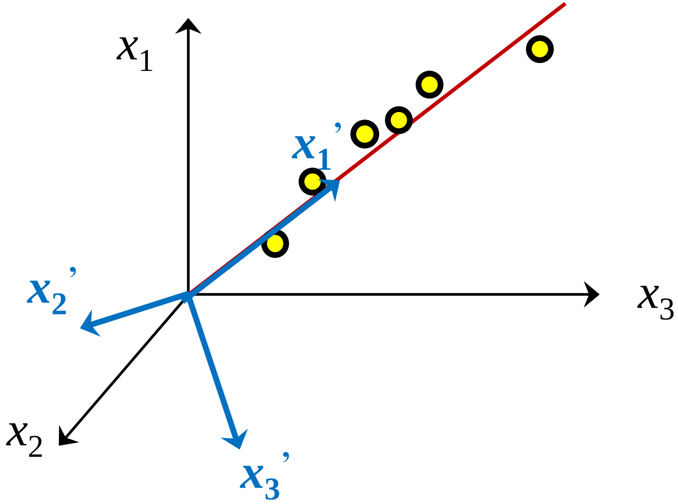
http://ethen8181.github.io/machine-learning/dim_reduct/PCA.html

PCA: How to find the PC ?

1	4	3	5.7	5.1	2.2
2	7.9	5.8	12	9.9	4.1
3.1	12	9	18	15	6.3

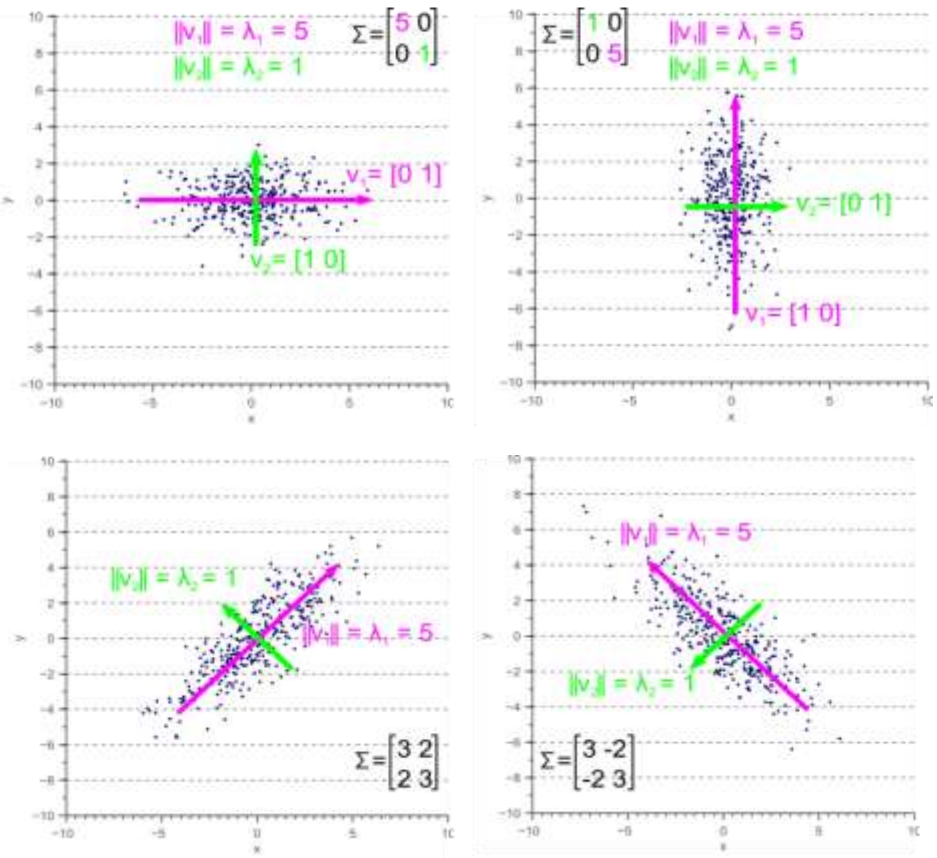
3.61	7.4	11.1	15.0	18.4	22.4
0.2	0.4	0.9	0.7	0.8	0.3
0.1	0.1	0.1	0.1	0.1	0.1

NOTE: These values are made up. Not exact.



Eigen-analysis of Covariance Matrix

$v_1, v_2 : Principal\ Components$

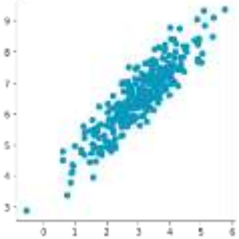


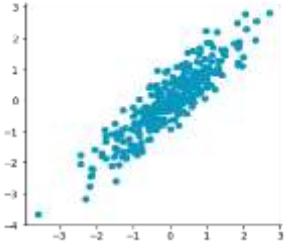
$$\Sigma\vec{v} = \lambda\vec{v}$$

$Value\ of\ \lambda\ indicates\ `variance'(spread)$
$in\ direction\ of\ \ eigenvector\ v\ associated\ with\ \lambda$

The PCA Recipe

1. Center the data





$$\boldsymbol{X}' = \boldsymbol{X} - \bar{x}$$

2. Compute the covariance matrix of X'

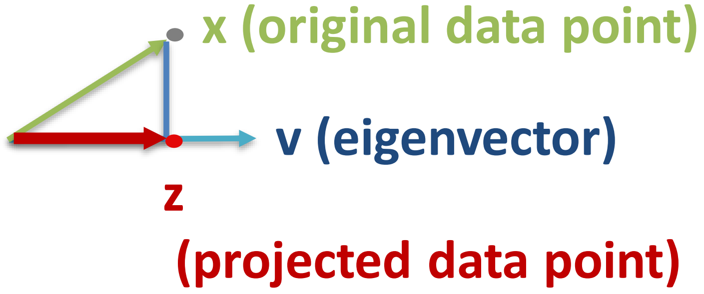
$$\left(\begin{array}{cccccc} & M1 & M2 & M3 & \dots & Mn \\ S1 & q_{1,1} & q_{1,2} & q_{1,3} & \dots & q_{1,n} \\ S2 & q_{2,1} & q_{2,2} & q_{2,3} & \dots & q_{2,n} \\ S3 & q_{3,1} & q_{3,2} & q_{3,3} & \dots & q_{3,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ Sm & q_{m,1} & q_{m,2} & q_{m,3} & & q_{m,n} \end{array}\right) \Rightarrow C = \begin{pmatrix} \mathrm{cov}(M_1,M_1) & \mathrm{cov}(M_1,M_2) & \cdots & \mathrm{cov}(M_1,M_n) \\ \mathrm{cov}(M_2,M_1) & \mathrm{cov}(M_2,M_2) & \cdots & \mathrm{cov}(M_2,M_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{cov}(M_n,M_1) & \mathrm{cov}(M_n,M_2) & \cdots & \mathrm{cov}(M_n,M_n) \end{pmatrix}_{n \times n}$$

N-dimensional Covariance Matrix

The PCA Recipe

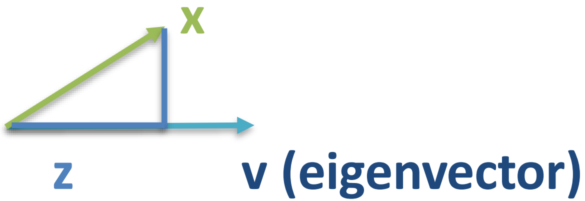
3. Compute Eigenvectors and Eigenvalues of Covariance Matrix Σ

4. Project data onto eigenvectors to obtain new coordinates



$$\mathbf{z} = (\mathbf{x}^T \mathbf{v})\mathbf{v}$$

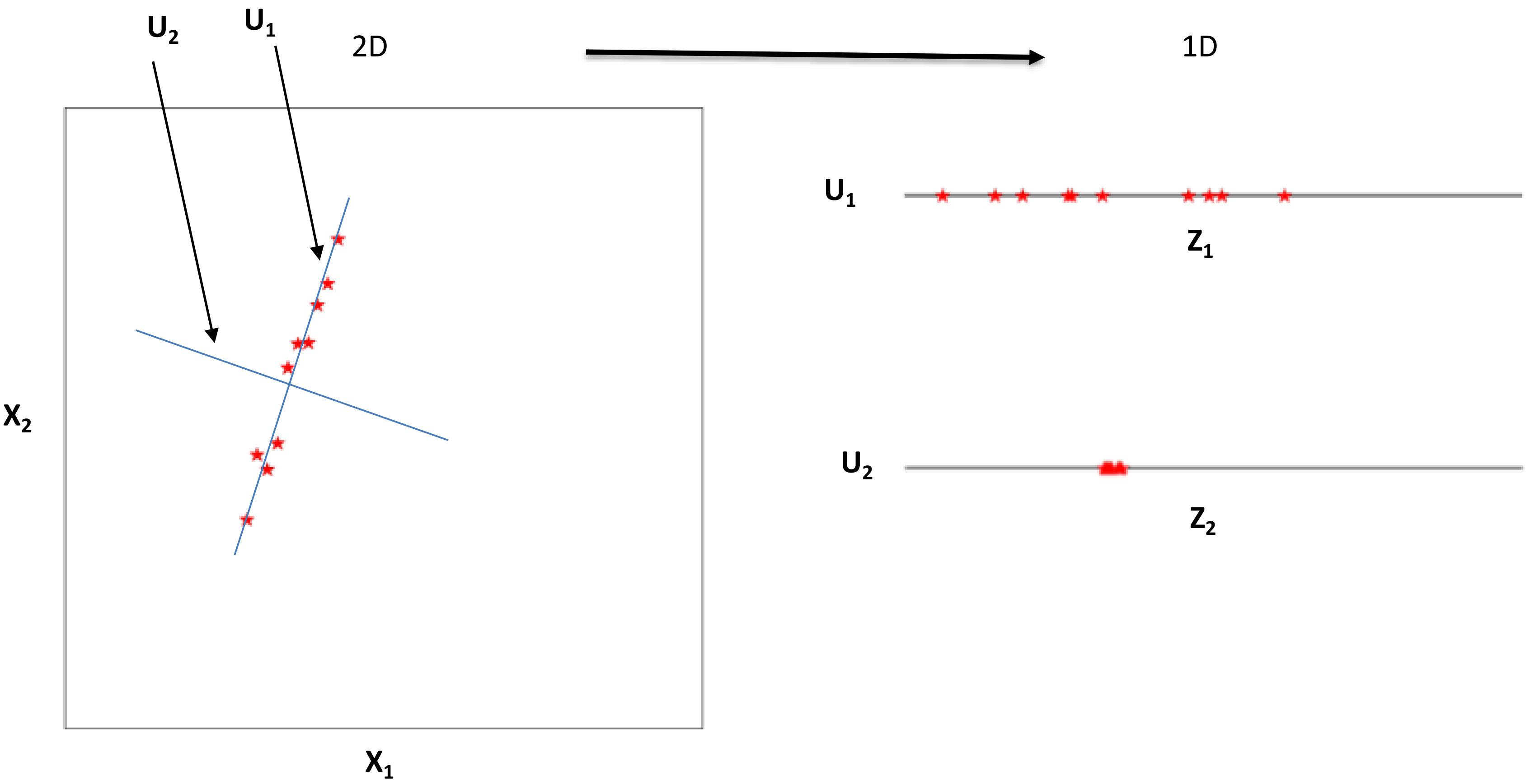
x



$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & v_1^T & \cdot & \cdot \\ \cdot & \cdot & v_2^T & \cdot & \cdot \\ \cdot & \cdot & v_3^T & \cdot & \cdot \\ \cdot & \cdot & v_4^T & \cdot & \cdot \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

New coordinates

Old coordinates

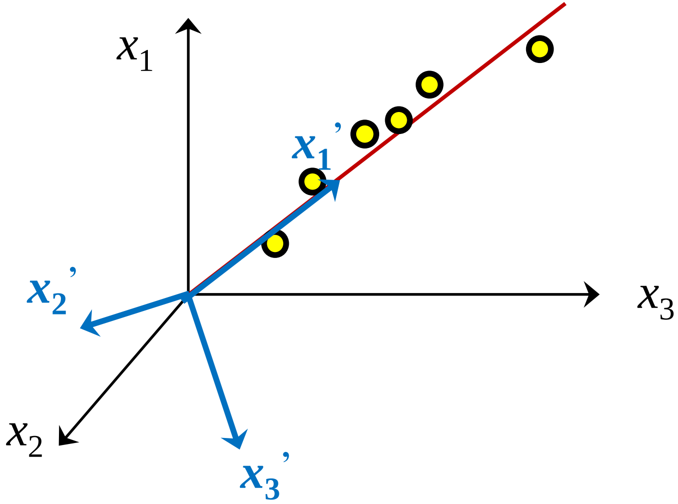

U2
U1
2D
1D
X2
X1
U1
Z1
U2
Z2

PCA: Toy Example - 2

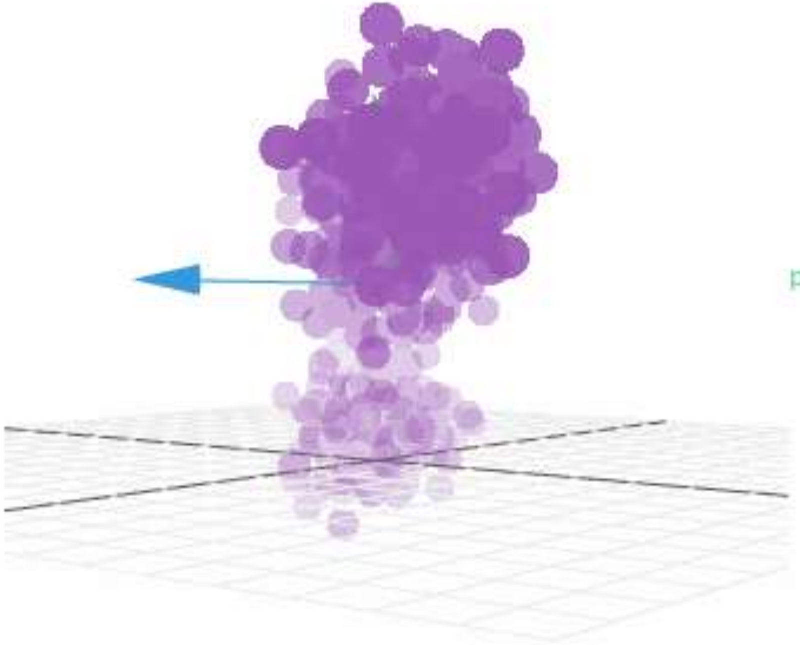
1	4	3	5.7	5.1	2.2
2	7.9	5.8	12	9.9	4.1
3.1	12	9	18	15	6.3

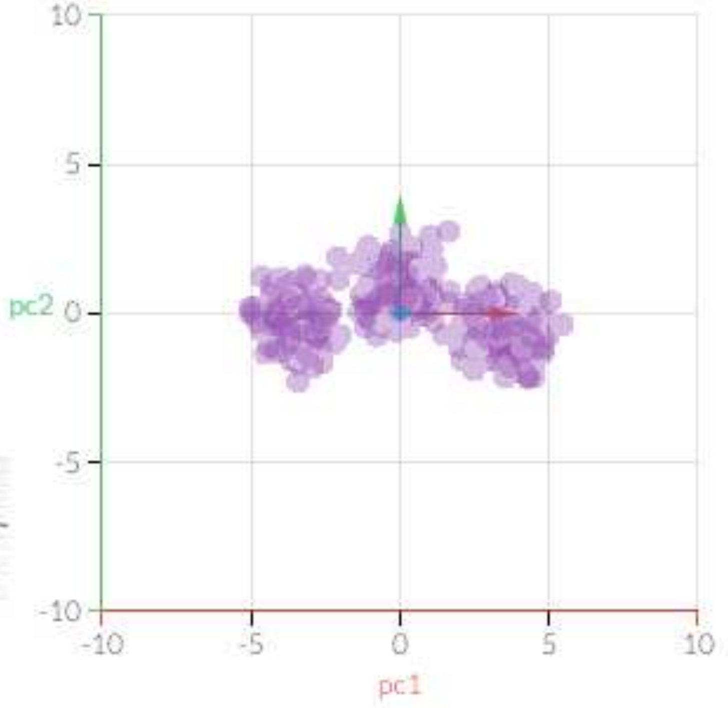
3.61	7.4	11.1	15.0	18.4	22.4
0.2	0.4	0.9	0.7	0.8	0.3
0.1	0.1	0.1	0.1	0.1	0.1

NOTE: These values are made up. Not exact.



3D to 2D





X1, X2, X3

Z1, Z2

PCA: Dimensionality Reduction

r x 1   r x d   d x 1

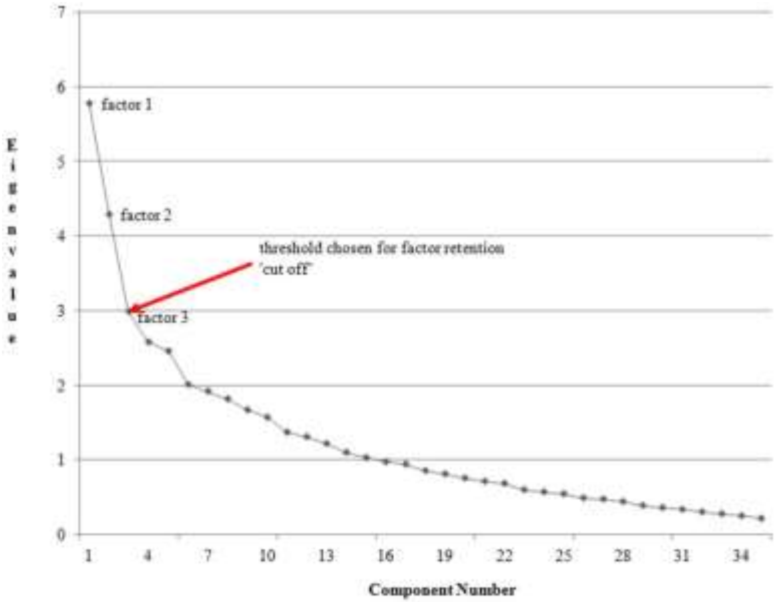
$$\underbrace{\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ . \\ . \\ z_r \end{bmatrix}}_{Z} = \underbrace{\begin{bmatrix} u_{11} & u_{12} & u_{13} & . & . & . & . & . & u_{1d} \\ u_{21} & u_{22} & u_{23} & . & . & . & . & . & u_{2d} \\ u_{31} & u_{32} & u_{33} & . & . & . & . & . & u_{3d} \\ . & & & & & & & & \\ u_{r1} & u_{r2} & u_{r3} & . & . & . & . & . & u_{rd} \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ x_d \end{bmatrix}}_{X}$$

Each row in U is an eigen vector of covariance matrix

Dimensionality reduction => r < d

PCA: Two Questions

- How many Eigen vectors to select? Eg. $\frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{d} \lambda_i} > 0.90$
  - Ans: Eigen Vectors corresponding to the larger Eigen values
  - Link to variance and trace



https://www.researchgate.net/profile/Collins_Ouma/publication/263548435/figure/fig1/AS:202705835433990@1425340245309/A-Scree-plot-of-eigenvalues-of-the-unrotated-factors-displaying-an-elbow-of-the-plot.png

PCA: Two Questions

- How much information is lost? Can we recover the old data/information from the new?

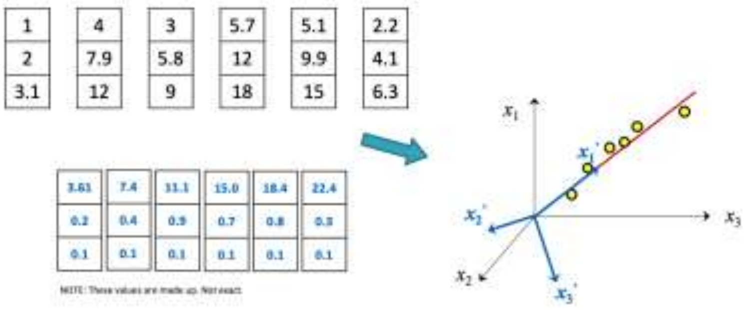
$$\mathbf{x} = z_1\mathbf{u_1} + z_2\mathbf{u_2} + z_3\mathbf{u_3} + z_4\mathbf{u_4}$$

$$\mathbf{x} = z_1\mathbf{u_1} + z_2\mathbf{u_2} + {\color{red} z_3\mathbf{u_3} + z_4\mathbf{u_4}}$$

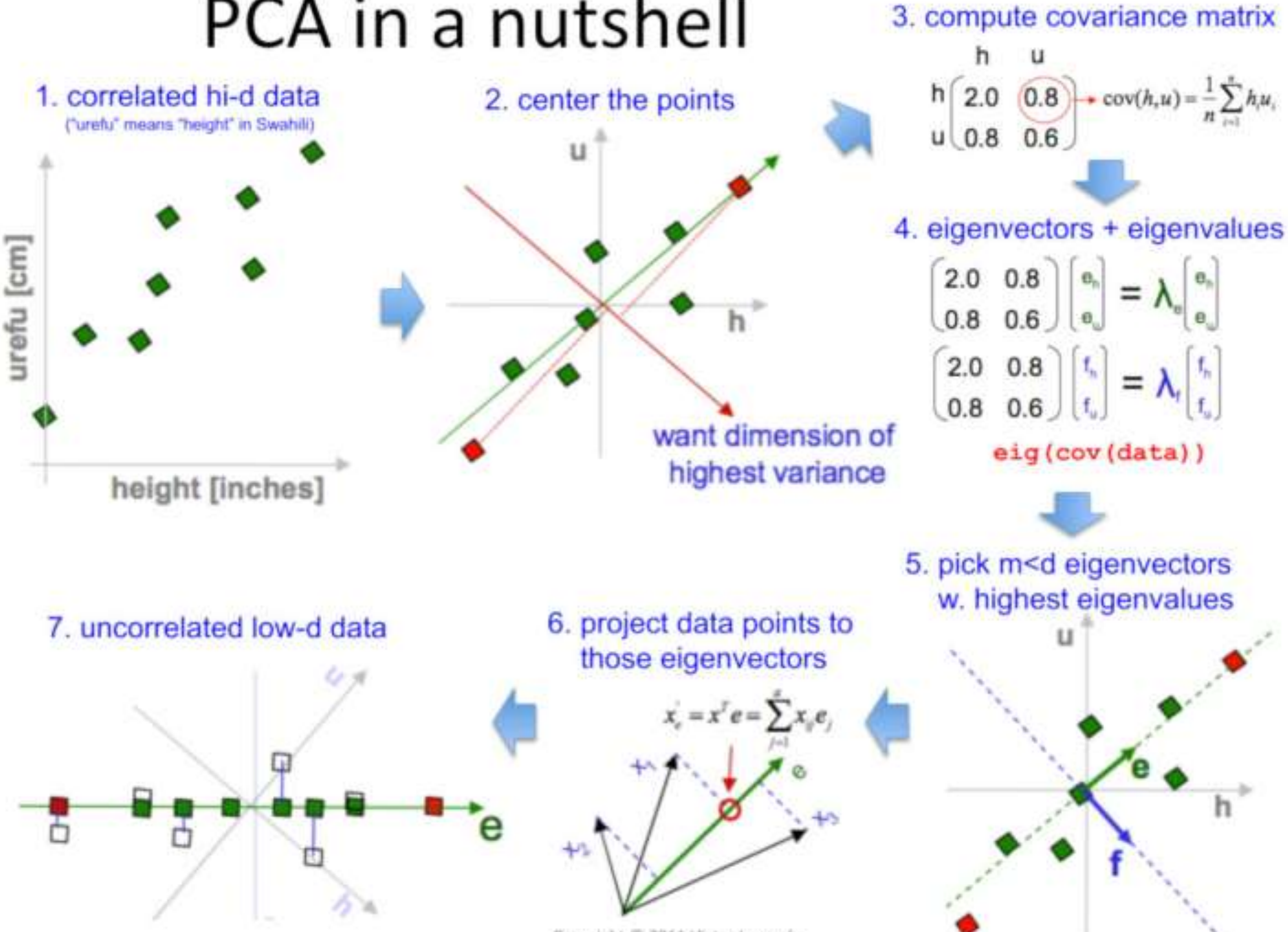
$$\mathbf{x}' = z_1\mathbf{u_1} + z_2\mathbf{u_2}$$

$$\text{Loss in Information} = ||\mathbf{x} - \mathbf{x}'||$$

Note: z_3 and z_4 are small and also λ_3 and λ_4 are small



PCA in a nutshell



Principal Component Analysis (PCA)

- Methodology
  - Suppose x_1, x_2, ..., x_M are N x 1 vectors

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

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

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

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

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

Step 4: compute the eigenvalues of C: $\lambda_1 > \lambda_2 > \cdots > \lambda_N$

Step 5: compute the eigenvectors of C: $u_1, u_2, \ldots, u_N$

Principal Component Analysis (PCA)

- Methodology
  - Suppose $x_1, x_2, ..., x_M$ are N x 1 vectors

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

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

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

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

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

Step 4: compute the eigenvalues of C: $\lambda_1 > \lambda_2 > \cdots > \lambda_N$

Step 5: compute the eigenvectors of C: $u_1, u_2, \ldots, u_N$

Principal Component Analysis (PCA)

- Linear transformation implied by PCA
  - The linear transformation $R^N \rightarrow R^K$ that performs the dimensionality reduction is:

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

(i.e., simply computing coefficients of linear expansion)

Selecting and Extracting Features

$$\begin{bmatrix} x_1 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Selecting first and third feature

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.2 & 0.3 & 0.4 \\ 0.0 & 0.4 & 0.2 & 1.7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

New Features as linear combination of old Features

$$X' = AX$$

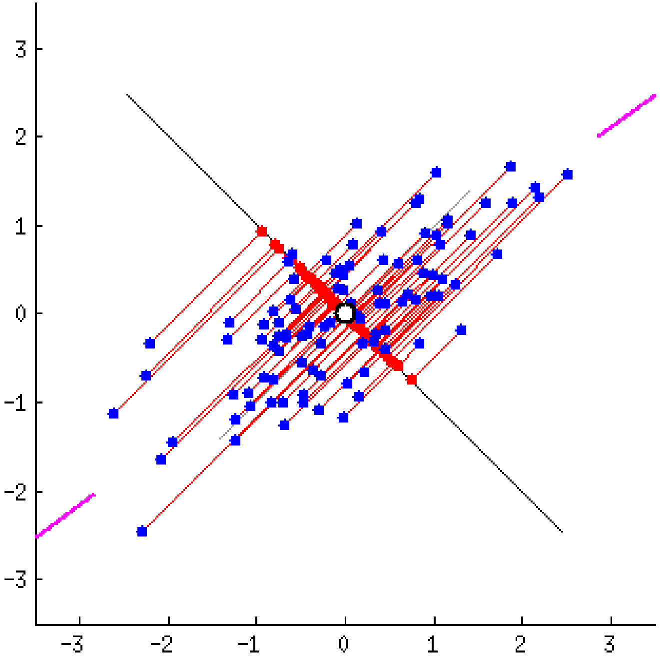
For PCA: Rows are Eigen vectors of the covariance matrix.

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cdots & u_1^T & \cdots \\ \cdots & u_2^T & \cdots \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

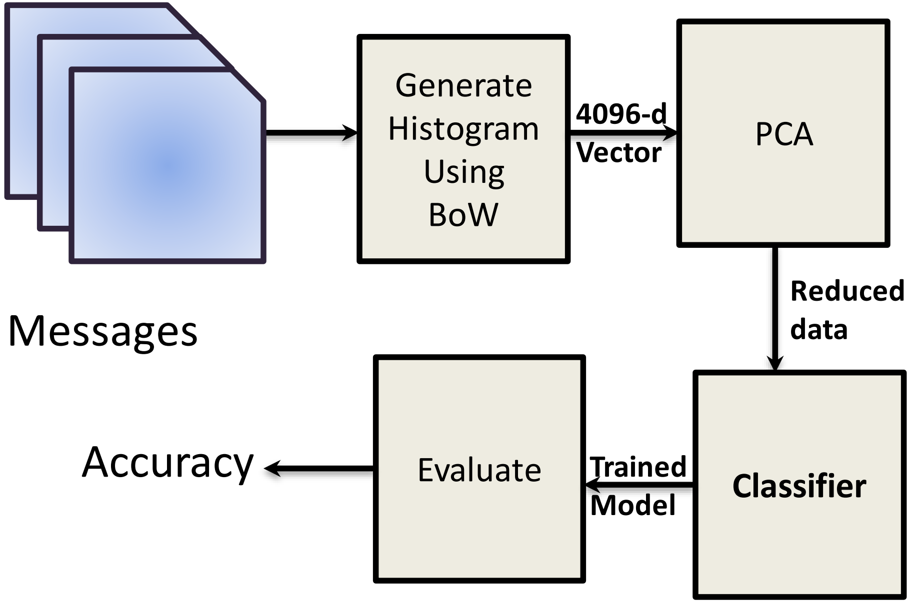
Selecting first and fourth feature

PCA - a graphical/energy explanation



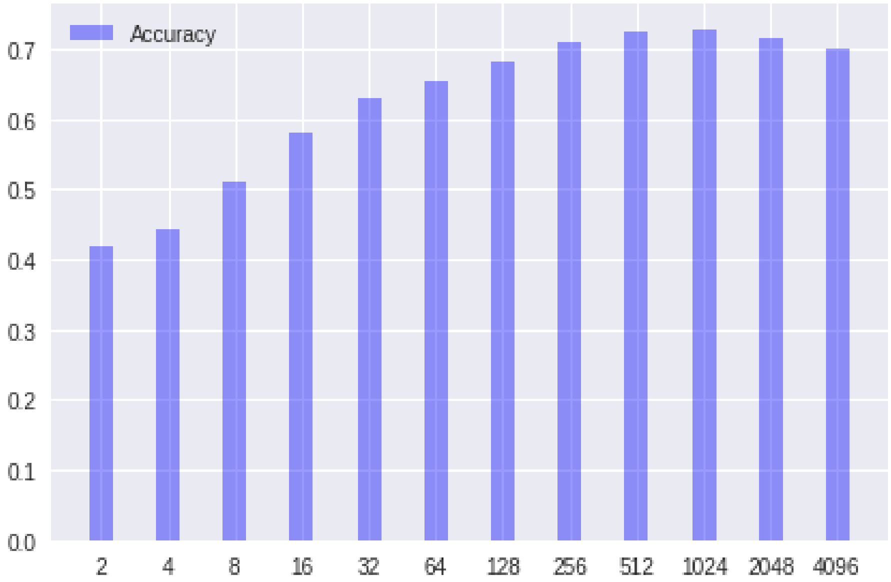
Case Study: PCA and Classification

- Text data with 20 classes
- Preprocessing:
  - Find the Histograms for Each Document using Bag of words
  - Apply PCA to reduce the dimensions
- Train the classifier on the reduced data
- Find the Accuracy to Evaluate the model



Effect of PCA on the Accuracy

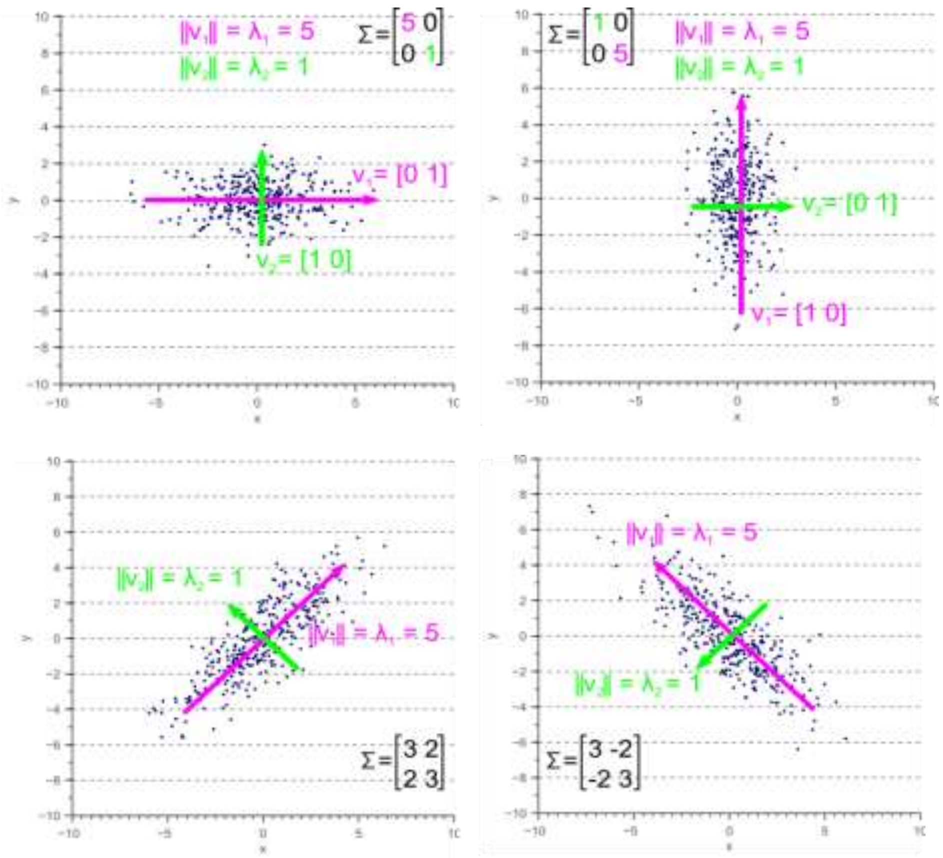
- Change r (dimensions in projected space) to 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096
- With just 3% (32) of the total dimensions (4096), comparable accuracies are obtained



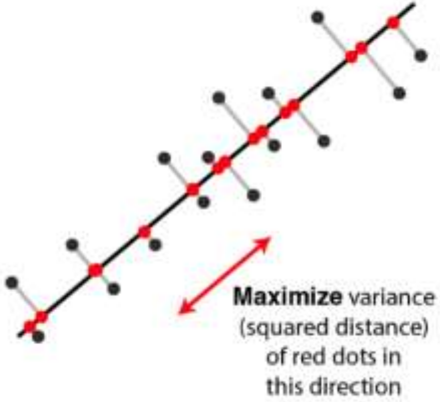
Eigen-analysis of Covariance Matrix

v_1, v_2: *Principal Components*

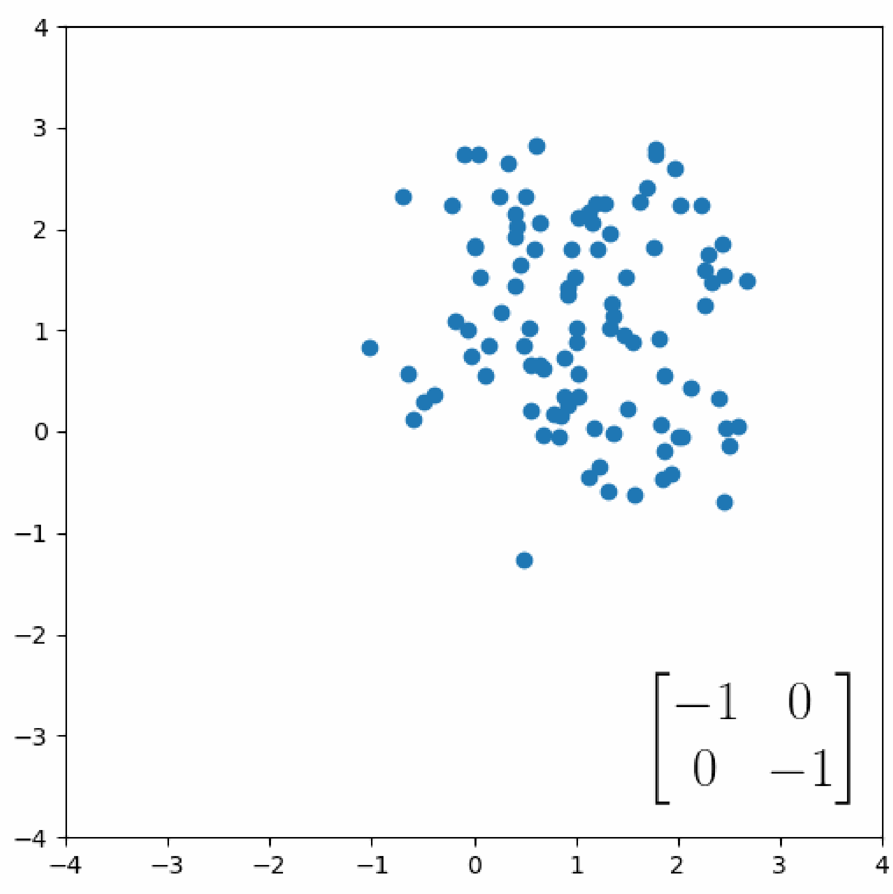
$$\Sigma\vec{v} = \lambda\vec{v}$$



Value of λ indicates `variance' (spread) in direction of eigenvector v associated with λ

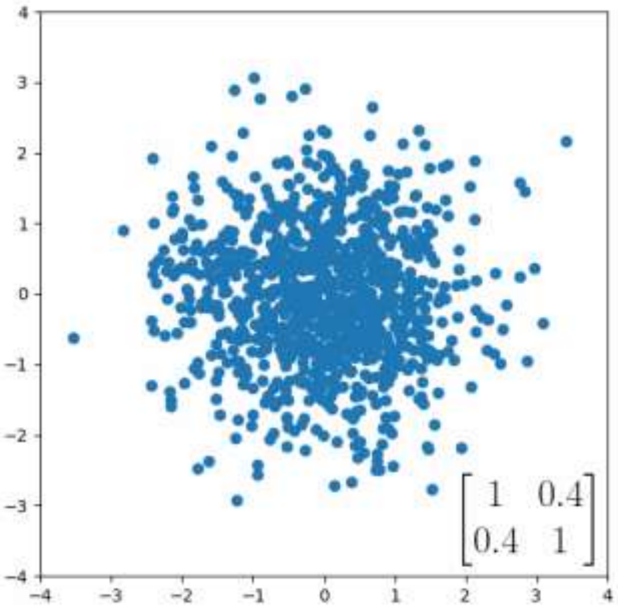


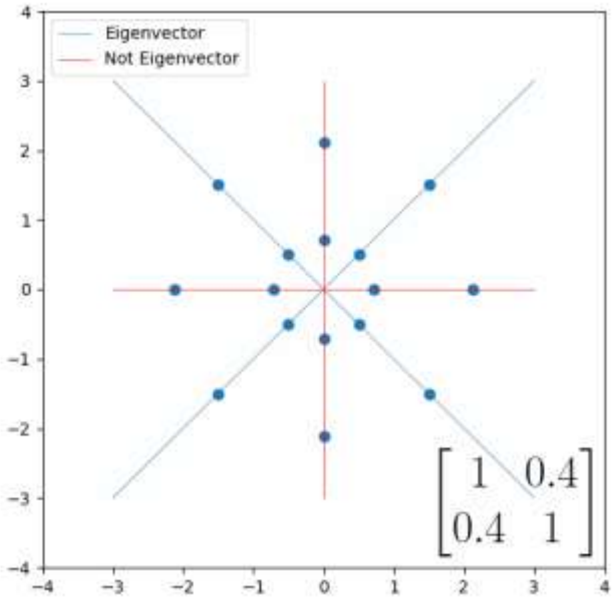
Visualizing matrices / linear transformations



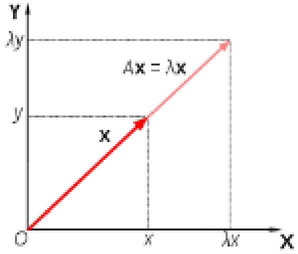
http://www.billconnelly.net/?p=697

Visualizing matrices / linear transformations





Eigenvectors = “Directions” of the matrix



http://www.billconnelly.net/?p=697

https://upload.wikimedia.org/wikipedia/commons/thumb/5/58/Eigenvalue_equation.svg/250px-Eigenvalue_equation.svg.png

Principal Component Analysis (PCA)

- Lower dimensionality basis
  - Approximate vectors by finding a basis in an appropriate lower dimensional space.

    (1) Higher-dimensional space representation:

    $$x = a_1 v_1 + a_2 v_2 + \cdots + a_N v_N$$

    $v_1, v_2, \ldots, v_N$ is a basis of the N-dimensional space

    (2) Lower-dimensional space representation:

    $$\hat{x} = b_1 u_1 + b_2 u_2 + \cdots + b_K u_K$$

    $u_1, u_2, \ldots, u_K$ is a basis of the K-dimensional space

    - *Note:* if both bases have the same size ($N = K$), then $x = \hat{x}$)

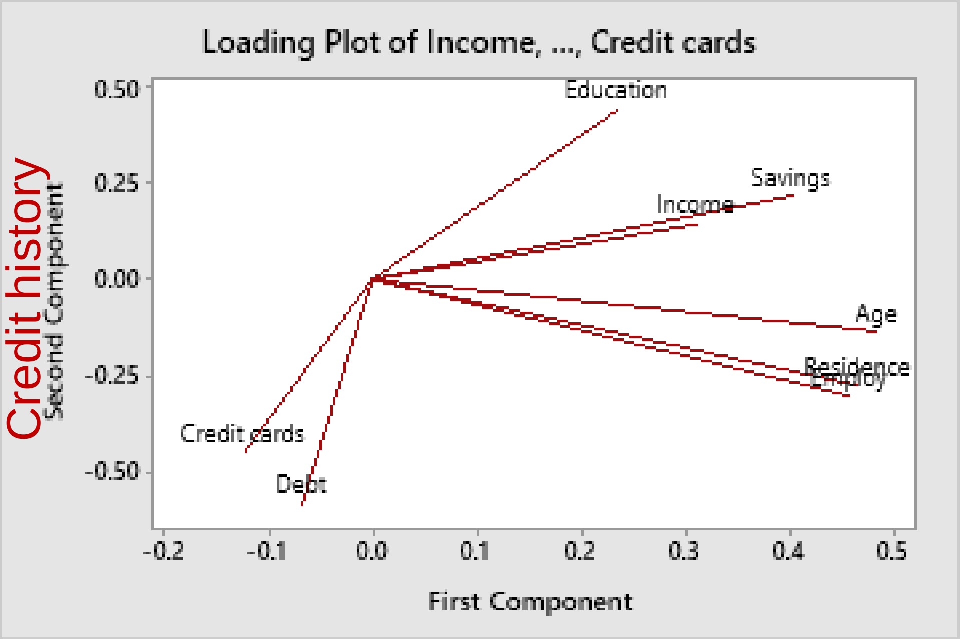
PCA – Loadings and Scores

Eigenanalysis of the Correlation Matrix

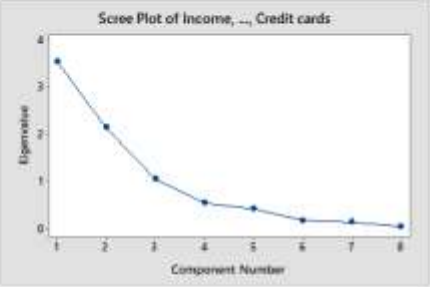
Eigenvalue	3.5476	2.1320	1.0447	0.5315	0.4112	0.1665	0.1254	0.04
Proportion	0.443	0.266	0.131	0.066	0.051	0.021	0.016	0.0
Cumulative	0.443	0.710	0.841	0.907	0.958	0.979	0.995	1.0

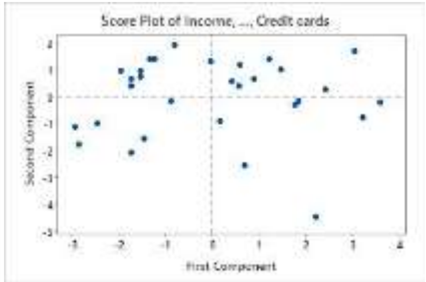
Eigenvectors

Variable	PC1	PC2	PC3	PC4	PC5	PC6	PC7	
Income	0.314	0.145	-0.676	-0.347	-0.241	0.494	0.018	-0
Education	0.237	0.444	-0.401	0.240	0.622	-0.357	0.103	0
Age	0.484	-0.135	-0.004	-0.212	-0.175	-0.487	-0.657	-0
Residence	0.466	-0.277	0.091	0.116	-0.035	-0.085	0.487	-0
Employ	0.459	-0.304	0.122	-0.017	-0.014	-0.023	0.368	0
Savings	0.404	0.219	0.366	0.436	0.143	0.568	-0.348	-0
Debt	-0.067	-0.585	-0.078	-0.281	0.681	0.245	-0.196	-0
Credit cards	-0.123	-0.452	-0.468	0.703	-0.195	-0.022	-0.158	0



Long-term Financial stability





https://support.minitab.com/en-us/minitab/21/help-and-how-to/statistical-modeling/multivariate/how-to/principal-components/interpret-the-results/all-statistics-and-graphs/#:~:text=Use%20the%20loading%20plot%20to,weak%20influence%20on%20the%20component.

https://cran.r-project.org/web/packages/LearnPCA/vignettes/Vig_04_Scores_Loadings.pdf

Singular Value Decomposition (SVD)
aka “billion-dollar algorithm”

$A = U \Sigma V^T$ - example: Users to Movies

SciFi-concept

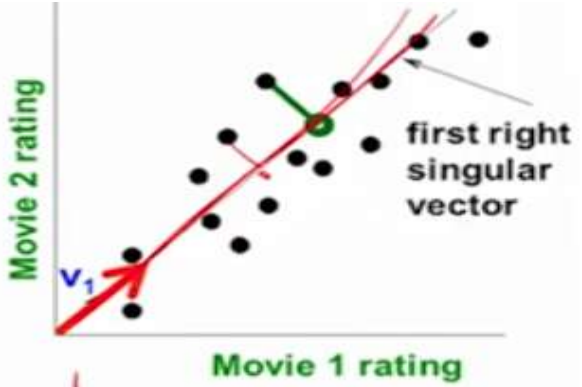
Romance-concept

$$
\begin{array}{c}
\begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
\mathbf{0.13} & 0.02 & -0.01 \\
\mathbf{0.41} & 0.07 & -0.03 \\
\mathbf{0.55} & 0.09 & -0.04 \\
\mathbf{0.68} & 0.11 & -0.05 \\
0.15 & \mathbf{-0.59} & \mathbf{0.65} \\
0.07 & \mathbf{-0.73} & \mathbf{-0.67} \\
0.07 & \mathbf{-0.29} & \mathbf{0.32}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{12.4} & 0 & 0 \\
0 & \mathbf{9.5} & 0 \\
0 & 0 & \mathbf{1.3}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\
0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\
0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09
\end{bmatrix}
$$

https://www.youtube.com/watch?v=P5mlg91as1c

$A = U\,\Sigma\,V^T$ - example:

- $U \cdot \Sigma$: Gives the coordinates of the points in the projection axis



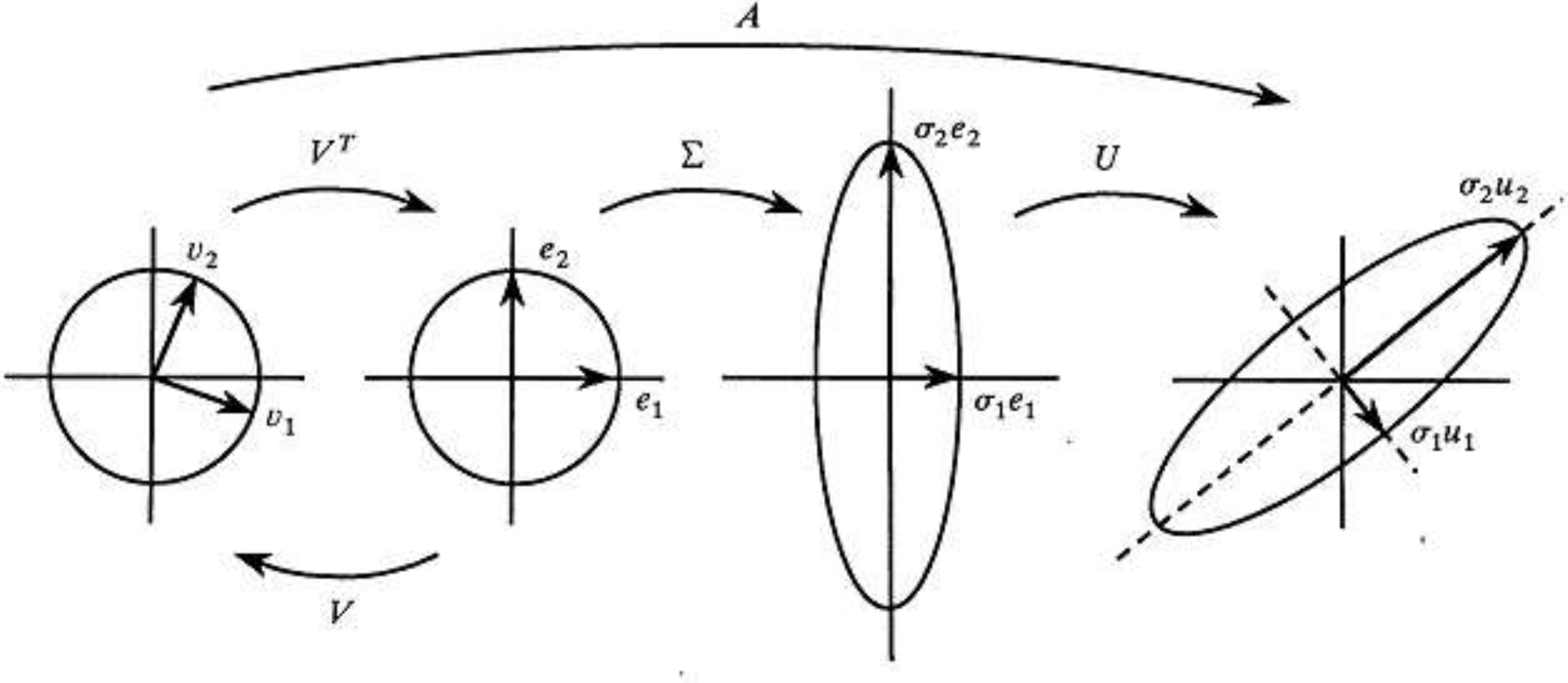
$$
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
$$

A

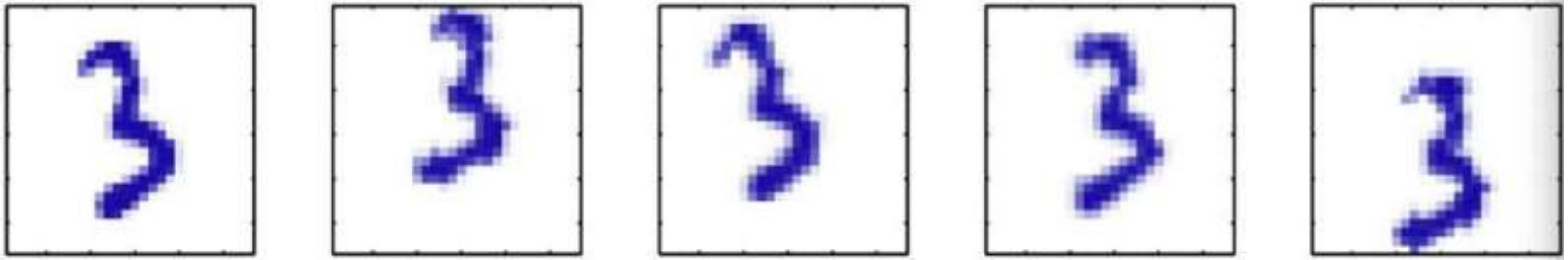
Projection of users on the "Sci-Fi" axis $((U\,\Sigma)^T)$:

$$
\begin{bmatrix}
\boxed{1.61} & 0.19 & -0.01 \\
\boxed{5.08} & 0.66 & -0.03 \\
\boxed{6.82} & 0.85 & -0.05 \\
\boxed{8.43} & 1.04 & -0.06 \\
\boxed{1.86} & -5.60 & 0.84 \\
\boxed{0.86} & -6.93 & -0.87 \\
\boxed{0.86} & -2.75 & 0.41
\end{bmatrix}
$$

$A = UDV^T$



- Take a *single* 64*64 digit and create a dataset by repeatedly
  - Move it to a 100*100 image
  - Shift by *x,y* and rotate by θ
- Dataset has 10,000 features but really only needs 3

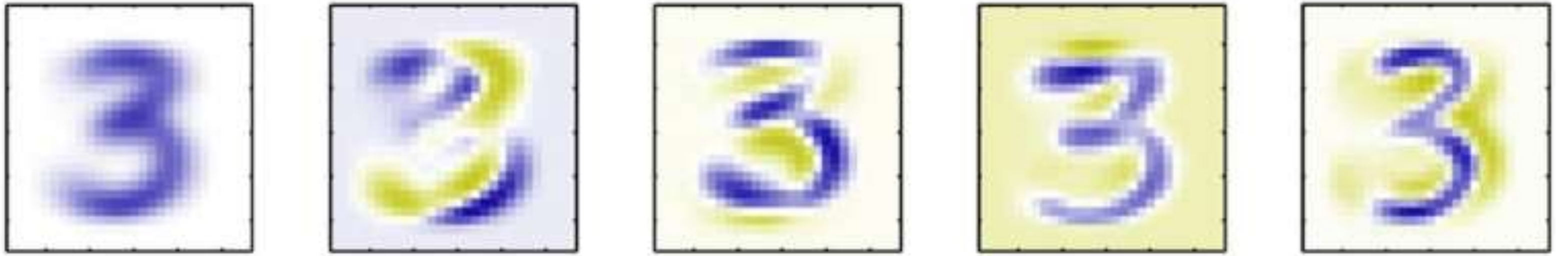


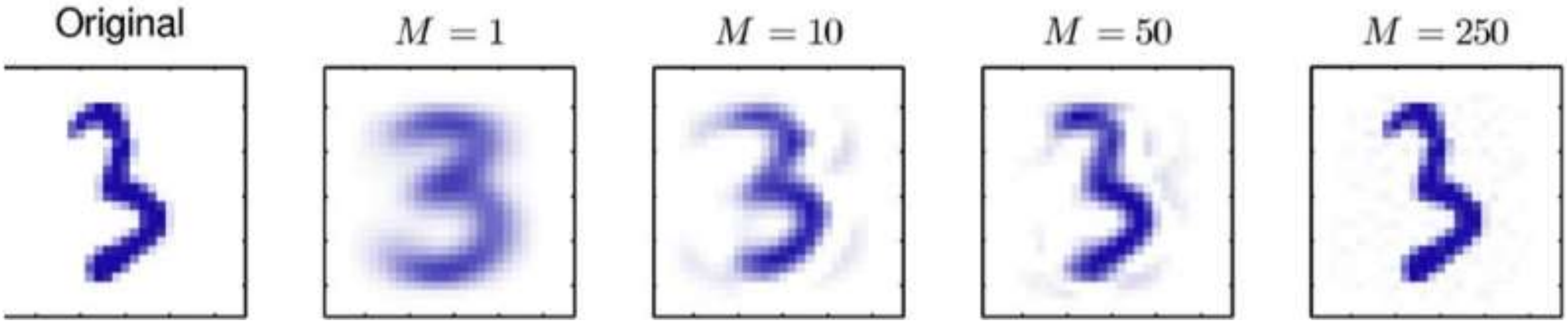
https://slideplayer.com/slide/12700209/

"prototype" = a vector of the same dimension as the instances

- PCA: reduces each instance to a linear combination of a few "prototypes" (blue+, green-). These are the first 5:

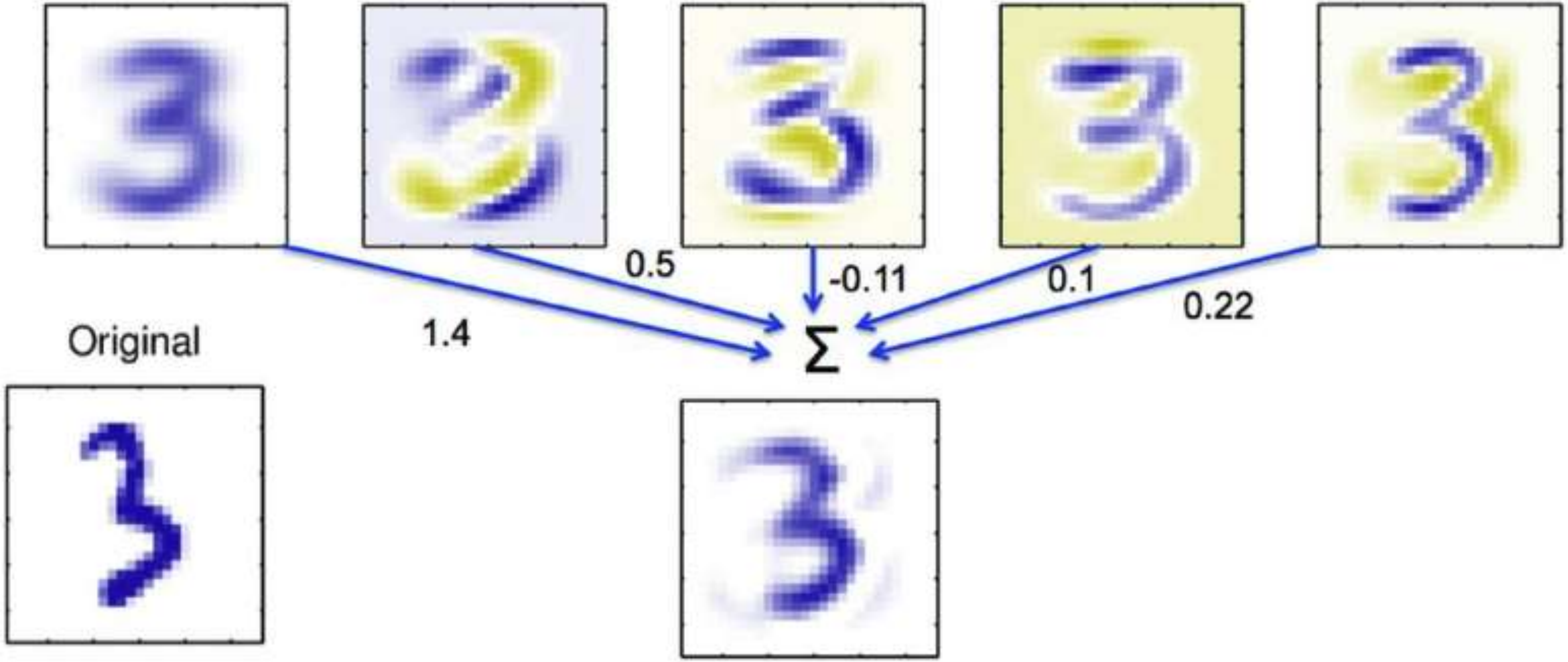
A specific choice of prototypes are the **principle components**



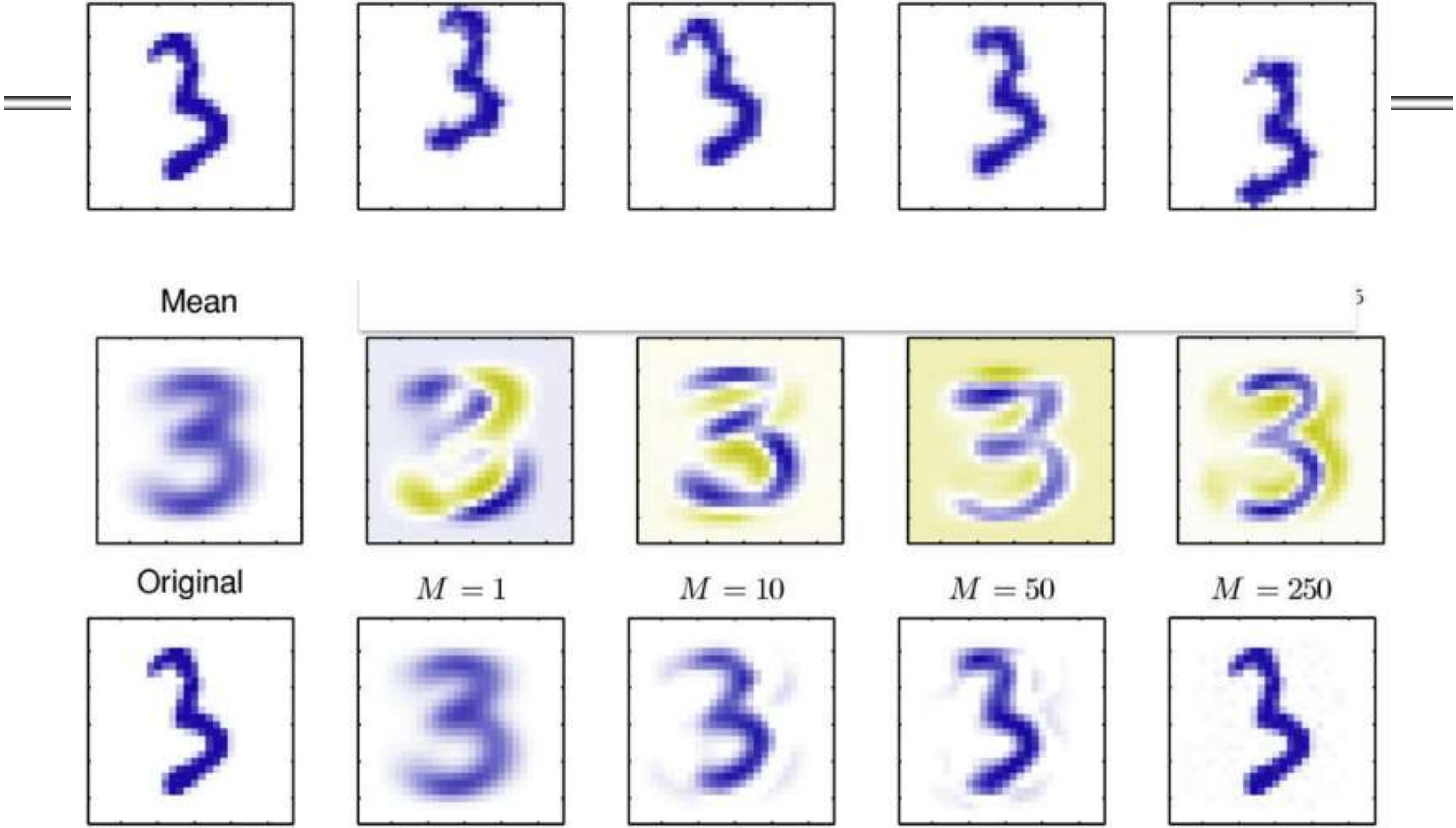


"prototype" = a vector of the same dimension as the instances

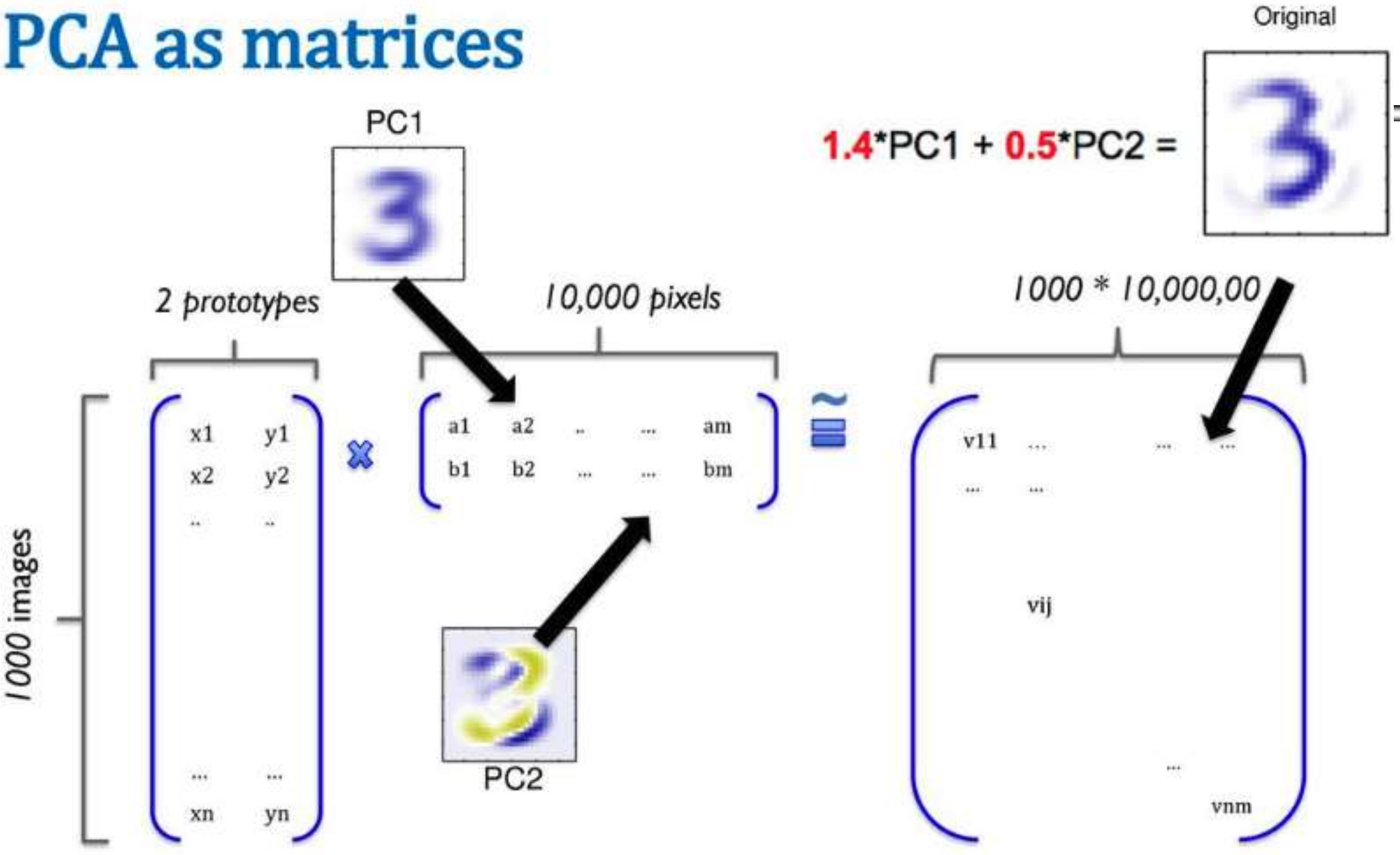
- PCA: reduces each instance to a linear combination of a few "prototypes" (blue+, green-). These are the first 5:



https://slideplayer.com/slide/12700209/


Mean
Original
$M = 1$
$M = 10$
$M = 50$
$M = 250$

PCA as matrices



V[i,j] = pixel j in image i

https://slideplayer.com/slide/12700209/

Assumptions when using PCA

- Variance is related to information content
- Data should be transformed in such a way that this variance is maximized
- High correlations between variables are a form of noise that should be minimized
- Correlations between variables are linear

References

- https://towardsdatascience.com/principal-component-analysis-3c39fbf5cb9d
- https://medium.com/swlh/interpreting-principal-components-fifa20-players-use-case-639fde373bac
- https://jeremy9959.net/Math-3094-UConn/published_notes/notes/PCA.pdf

- Bishop PRML, 12.1, 12.2, 12.4
- https://www.cse.iitk.ac.in/users/piyush/courses/pml_winter16/slides_lec10.pdf