Data Transformation & Principal Component Analysis (PCA)

Presented by Vikram Pudi
vikram@iiit.ac.in
IIIT Hyderabad

Data Science / KDD Life-cycle

- **Domain Understanding**
- **Data Preprocessing**
  - Data discovery / integration
  - Clean: Noise, inconsistency, missing values
  - Transform: Discretize, normalize, generalize
  - **Feature selection & engineering**
- ***Data Mining / Analytics***
  - Pick model
  - Fix hyper-parameters
  - Train & optimize
- **Post Mining**
  - Presentation / Visualization
  - Evaluation
  - Decision making

Data Cleaning

Noise, inconsistency, missing vales

Handling noise (inaccuracies)

- Binning: Sort values of an attribute and partition into ranges/bins. Replace all values within a bin by its mean/median/…
- Regression: Fit the data into a function such as linear or non-linear regression.
- Outliers: Treat outliers as noise and ignore them.

Inconsistency

- Avoid by using good integrity constraints when designing database.
- If inconsistency still arises, cure using same approaches as for handling noise.

Missing Values

1. Ignore missing values
2. Most common value
3. Concept most common value
4. All possible values
5. Missing values as special values
6. Use classification techniques

Data Transformation

Format conversion, discretization, normalization, …

Data format conversion

- Needed usually when combining data from multiple sources.
- Needs major manual programming effort
- Remember Y2K problem!

Discretization (Numeric → Categorical)

- Sort values of numeric attribute
- Divide sorted values into ranges
  - Equi-depth
  - Clustering
  - Information gain

Normalization

- Some attributes may have large ranges.
- Bring all attributes to common range.
- Scale values to lie within, say, -1.0 to +1.0

Generalization

- Categorical attributes (like name, location) contain too many values.
- Attributes like name can be ignored.
- Attributes like location can be generalized (e.g. instead of using address, use only the city/town name).

Feature selection

- Which features are likely to be relevant for a given task?
- E.g. for detecting spam emails, some words such as “free university degree”, “easy loan”, etc. may be more relevant than others.

Approach: Find discriminating features.

Data Representation

- Objective:
  - Given a real entity (a transaction, an image, a web page, a patient, etc.) come up with a representation.
  - “Features”, “Embedding”, “Measurements”, etc.
- Text: Eg. nGrams, “BoW”, “Word2Vec”

Principal Component Analysis

Simplifying representations
for numeric features

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

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

New Features as linear combination of old Features.

$$X' = AX$$

PCA Toy Example

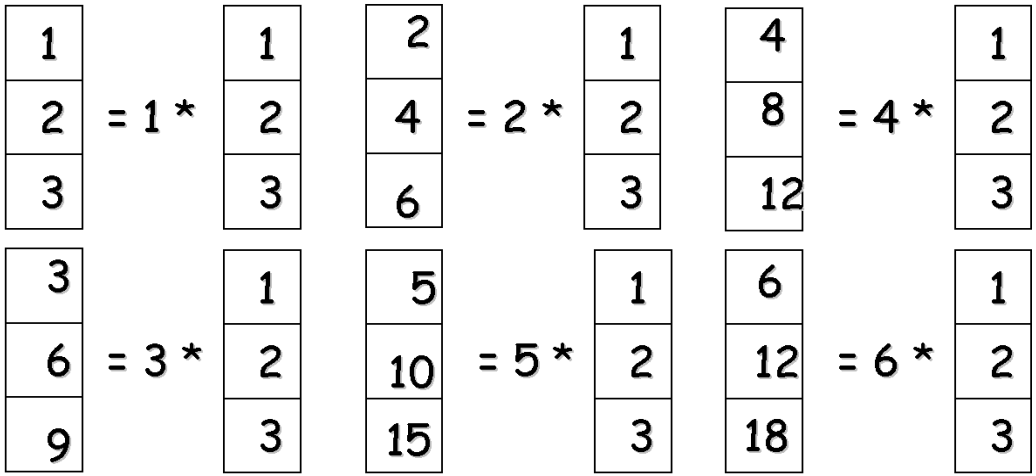
Consider the following 3D points

1	2	4	3	5	6
2	4	8	6	10	12
3	6	12	9	15	18

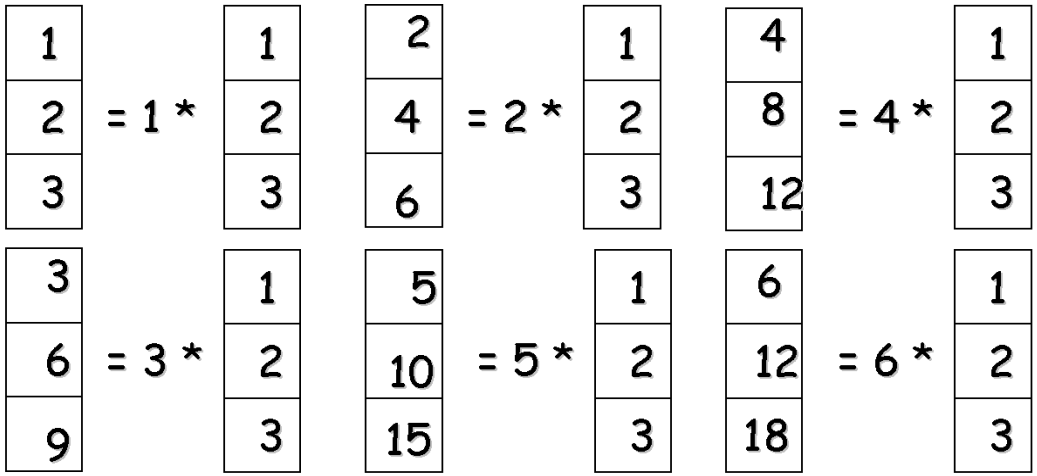
If each component is stored in a byte,
we need 18 = 3 x 6 bytes

PCA Toy Example

Looking closer, we can see that all the points are related geometrically: they are all the same point, scaled by a factor:



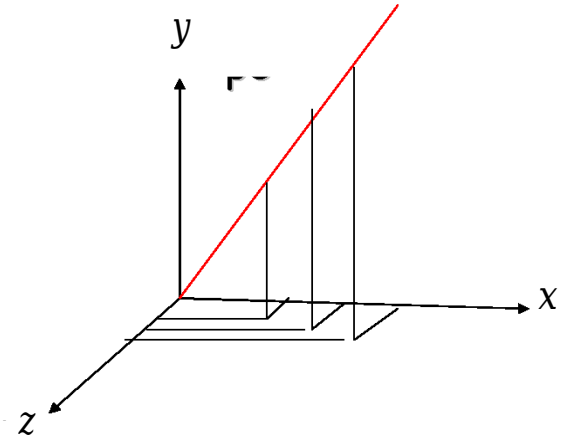
PCA Toy Example



They can be stored using only 9 bytes (50% savings!):
Store one point (3 bytes) + the multiplying constants (6 bytes)

Geometrical Interpretation:

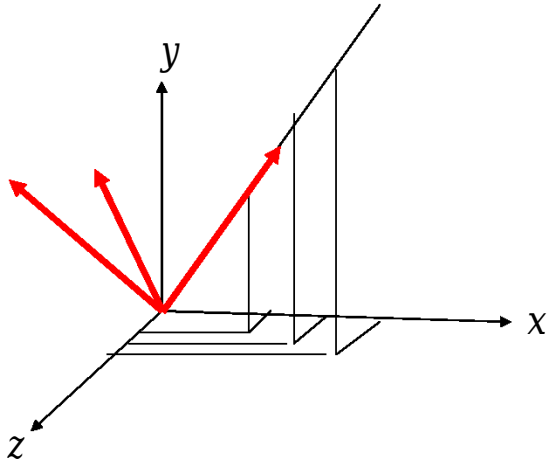
View each point in 3D space.



But in this example, all the points happen to belong to a line: a 1D subspace of the original 3D space.

Geometrical Interpretation:

Consider a new coordinate system where one of the axes is along the direction of the line:

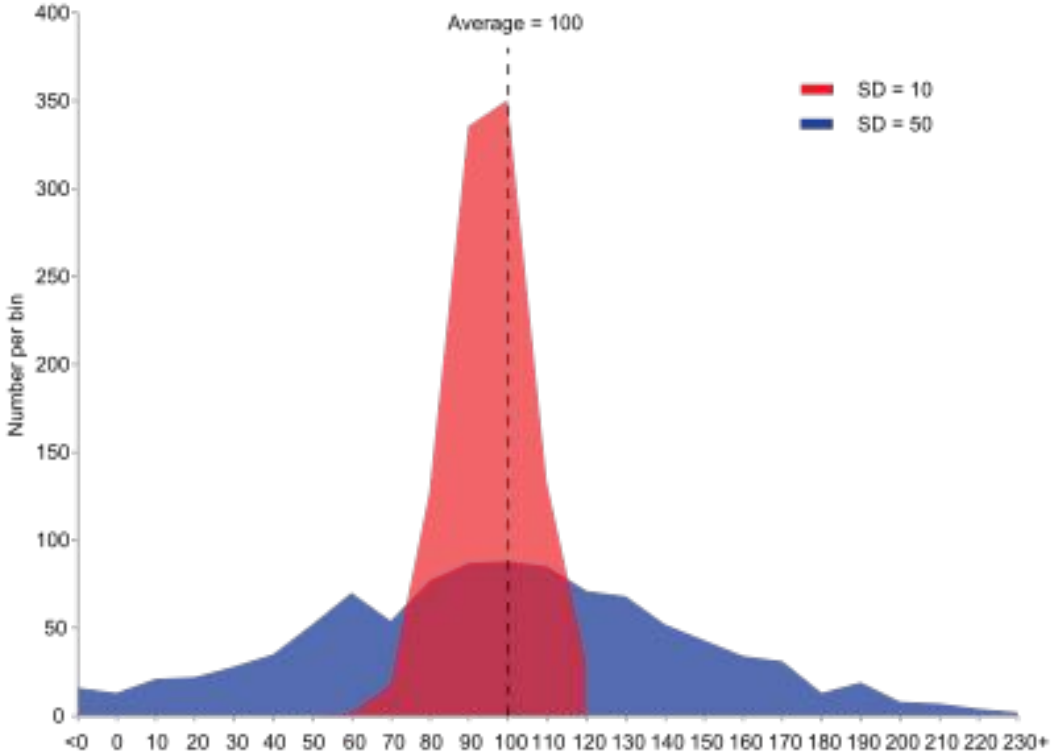


In this coordinate system, every point has only one non-zero coordinate: we *only* need to store the direction of the line (a 3 bytes image) and the non-zero coordinate for each of the points (6 bytes).

Principal Component Analysis (PCA)

- Given a set of points, how do we know if they can be compressed like in the previous example?
  - The answer is to look into the correlation between the points
  - The tool for doing this is called PCA

Variance



Population

Variance:

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

Standard Deviation:

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

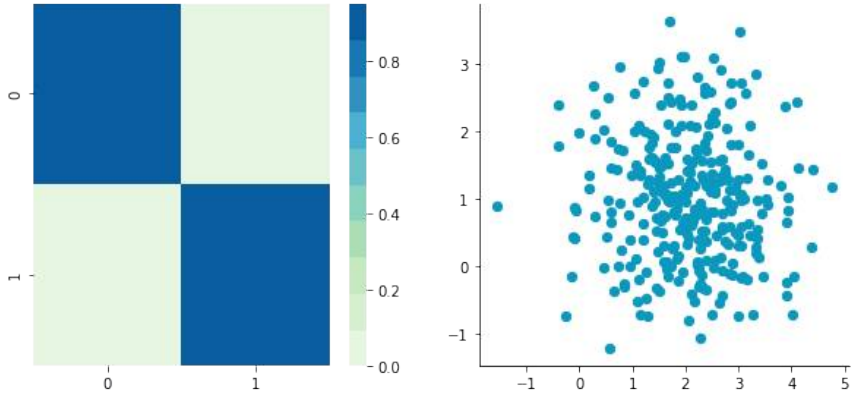
Sample Variance

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

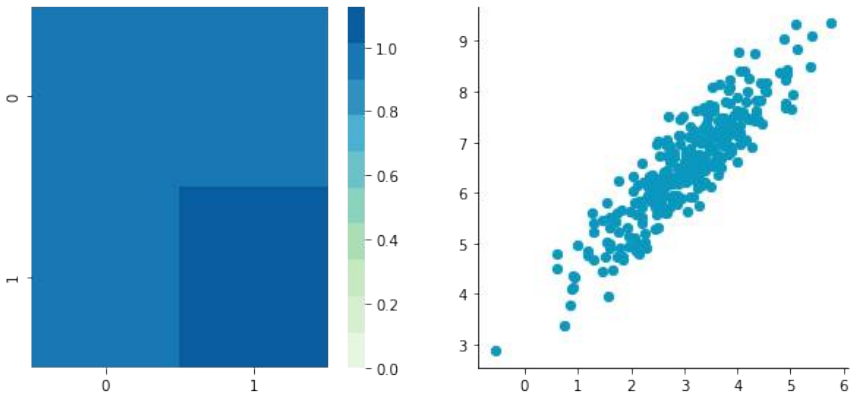
Sample Standard Deviation

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

Covariance



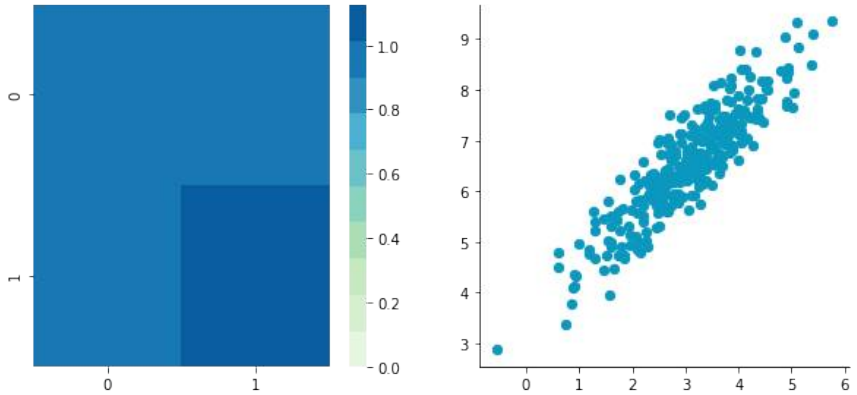
$$C = \begin{bmatrix} +0.95 & -0.04 \\ -0.04 & +0.87 \end{bmatrix}$$

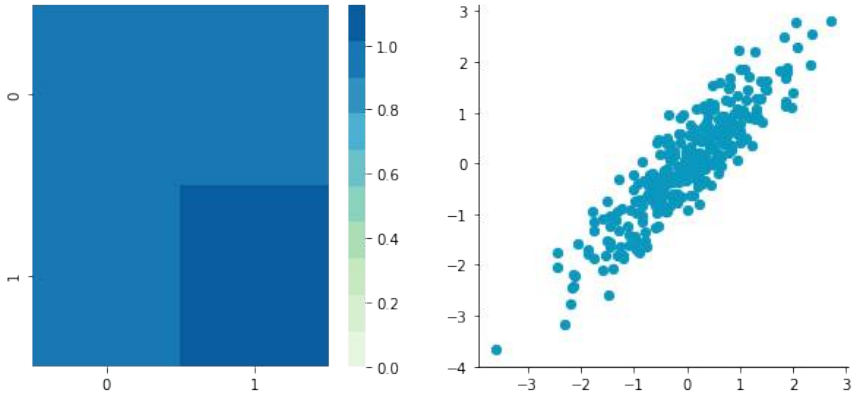


$$C = \begin{bmatrix} +0.95 & +0.92 \\ +0.92 & +1.12 \end{bmatrix}$$

Mean Normalization

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



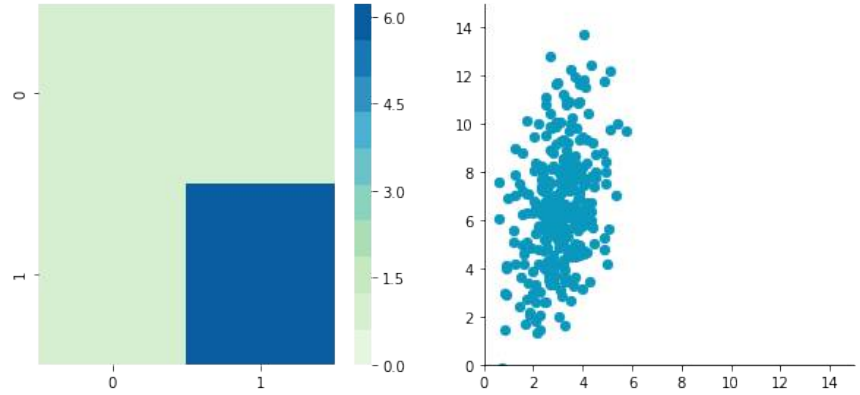


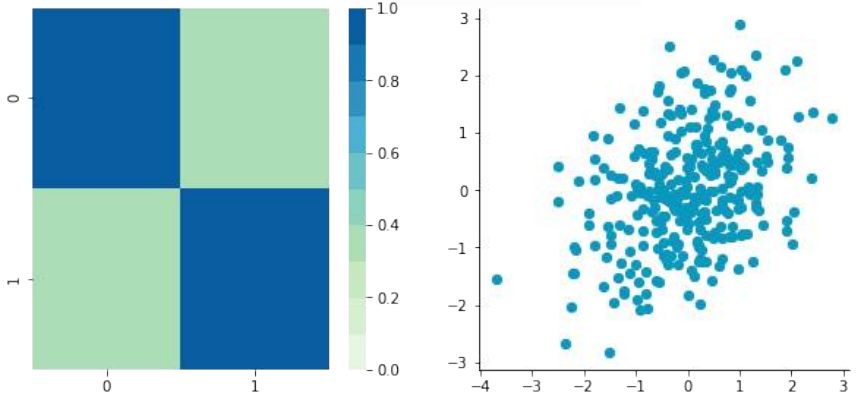
$$C = \begin{bmatrix} +0.95 & +0.92 \\ +0.92 & +1.12 \end{bmatrix}$$

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

Standardization

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





$$C = \begin{bmatrix} +0.95 & +0.83 \\ +0.83 & +6.22 \end{bmatrix}$$

$$C = \begin{bmatrix} +1.00 & +0.34 \\ +0.34 & +1.00 \end{bmatrix}$$

Covariance : m samples, n features

Vectors 1 and 3

Cell (3, 1) or (1, 3)

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 5 & 4 & 1 \\ 3 & 8 & 6 \end{bmatrix}$$

Covariance

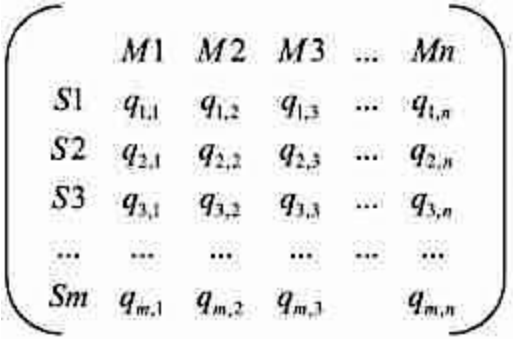
$$\begin{bmatrix} 2.67 & 0.67 & -2.67 \\ 0.67 & 4.67 & 2.33 \\ -2.67 & 2.33 & 4.67 \end{bmatrix}$$

Covariance Matrix

Variance:

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

$$cov(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

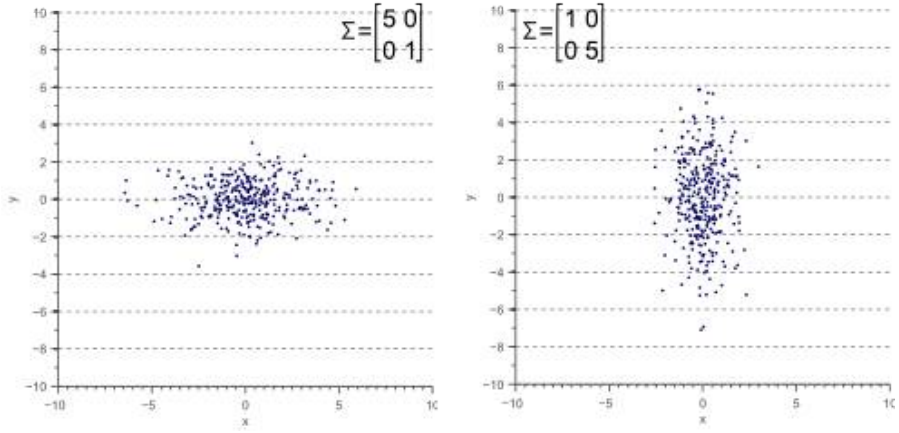


$$Cov(M_a, M_b) = \frac{1}{m}\sum_{i=1}^{m}(q_{i,a} - \overline{q_a})(q_{i,b} - \overline{q_b})$$

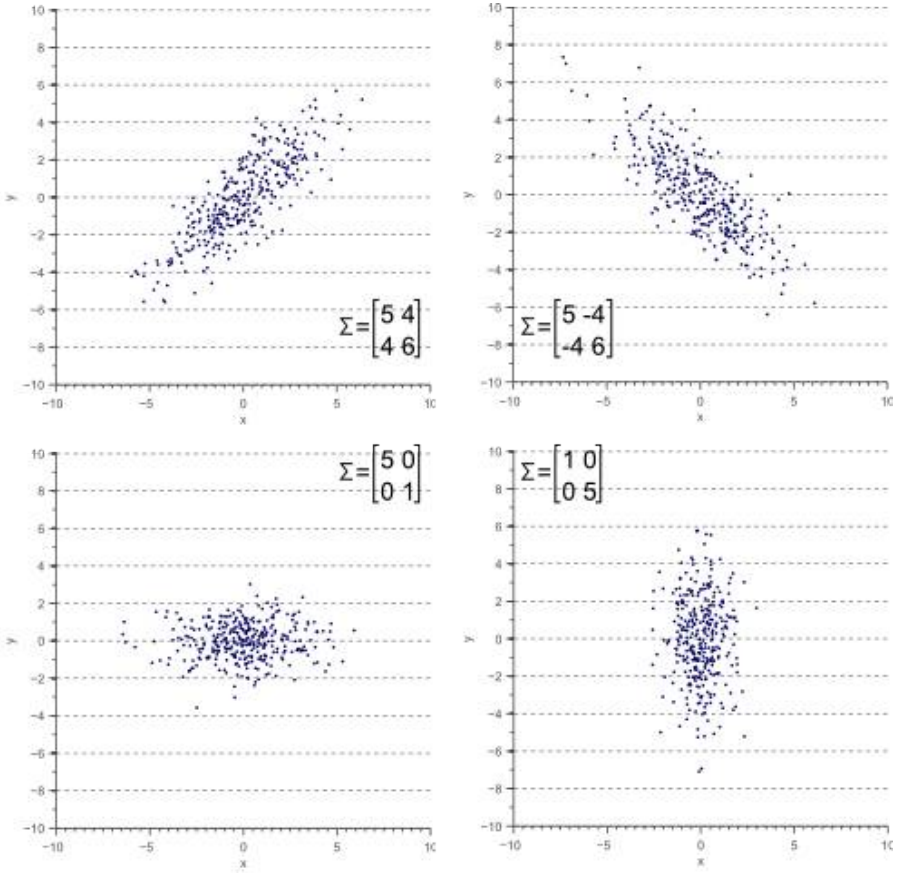
$$C = \begin{pmatrix} \mathrm{cov}(M_1, M_1) & \mathrm{cov}(M_1, M_2) & \cdots & \mathrm{cov}(M_1, M_n) \\ \mathrm{cov}(M_2, M_1) & \mathrm{cov}(M_2, M_2) & \cdots & \mathrm{cov}(M_2, M_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{cov}(M_n, M_1) & \mathrm{cov}(M_n, M_2) & \cdots & \mathrm{cov}(M_n, M_n) \end{pmatrix}_{n \times n}$$

n-dimensional Covariance Matrix

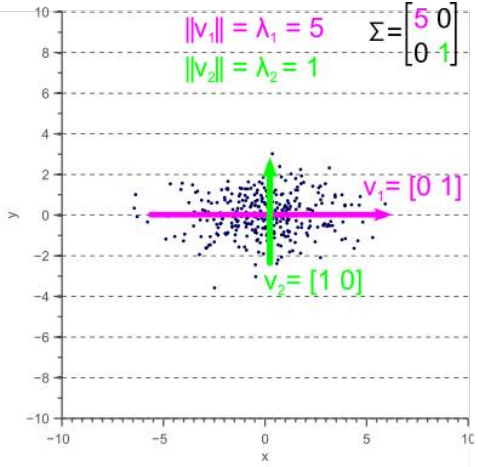
Covariance Matrix encodes spread and orientation of data

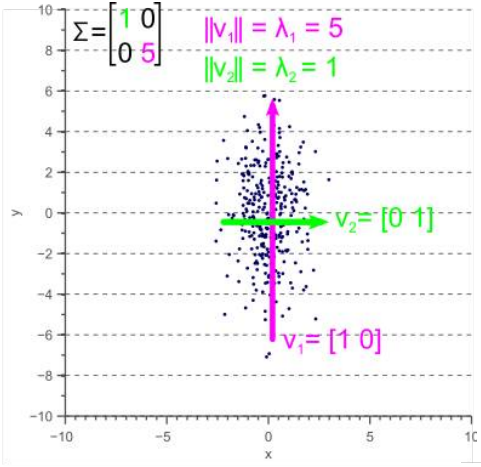


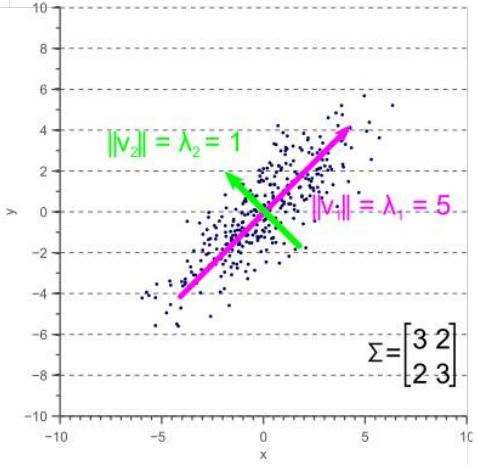
Covariance Matrix encodes spread and orientation of data

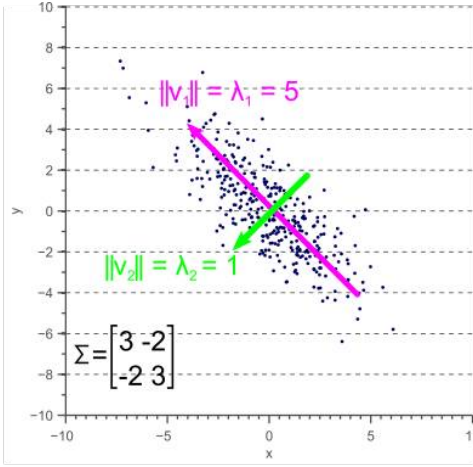


$$\Sigma\vec{v} = \lambda\vec{v}$$

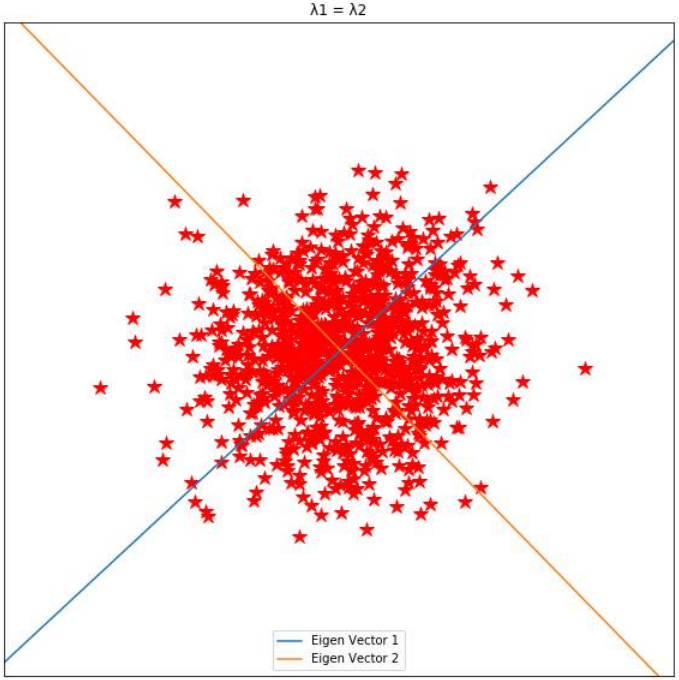




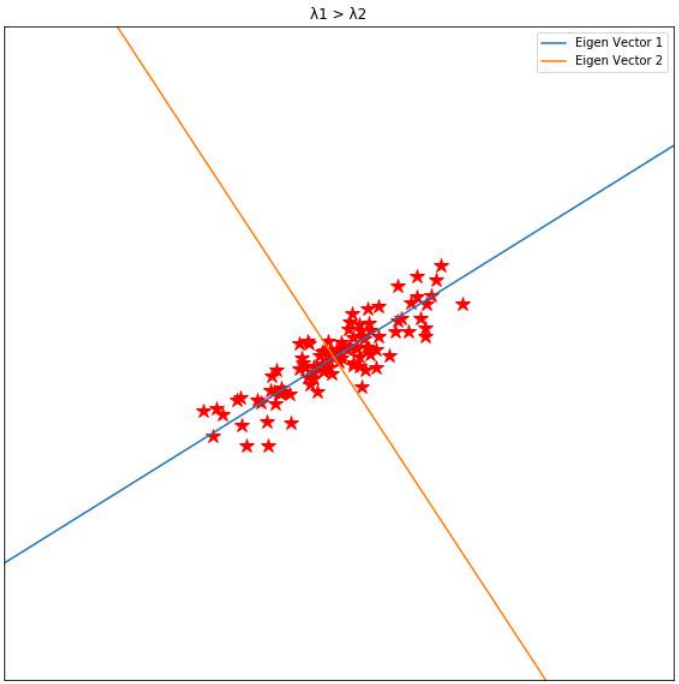




Covariance, Eigen Values and Vectors

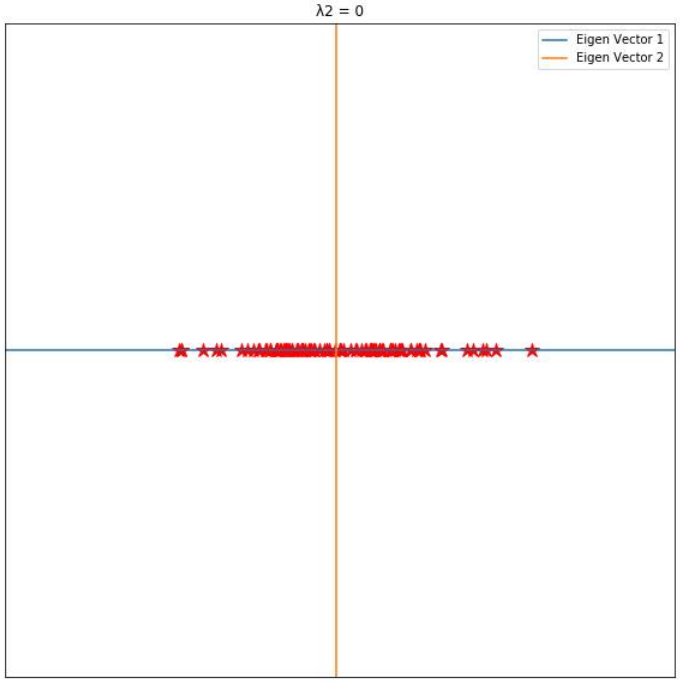


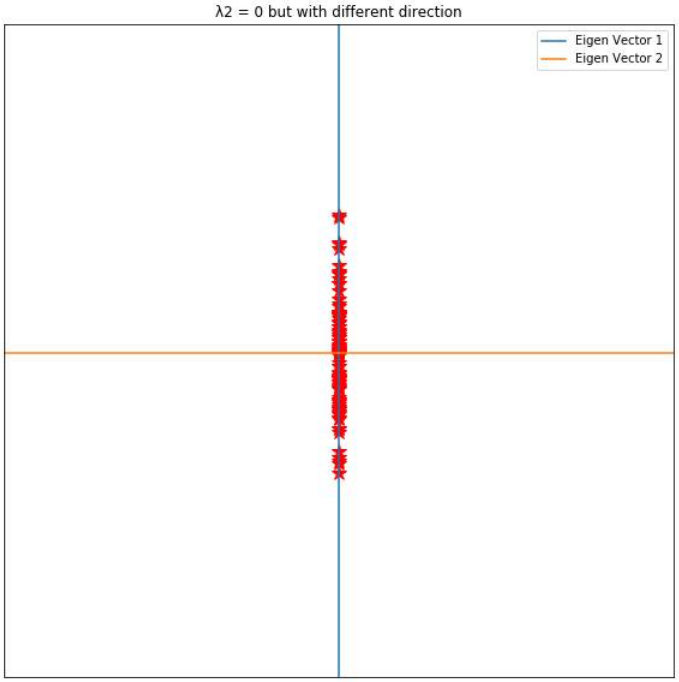
Both the eigen values are equal
(Distribution is circular)



One eigen value is greater than the other
(Distribution is elongated in the direction of that eigen vector)

Covariance, Eigen Values and Vectors





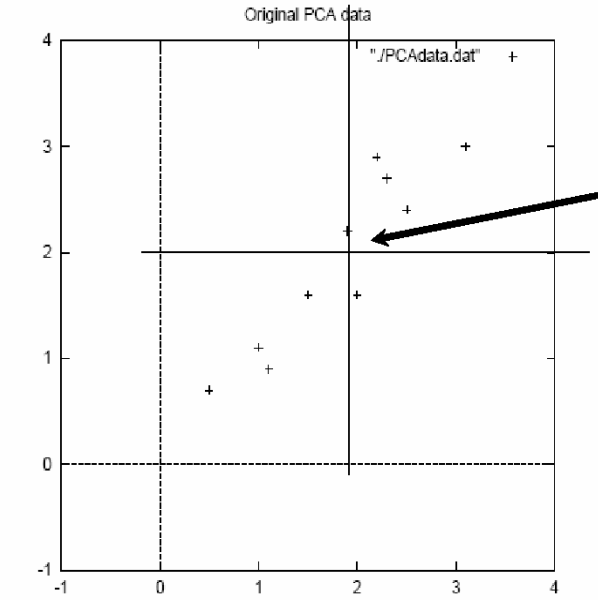
Only 1 Eigen value is non-zero, distribution of data will align on that Eigen vector

PCA Example –STEP 1

http://kybele.psych.cornell.edu/~edelman/Psych-465-Spring-2003/PCA-tutorial.pdf

- 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



mean

this becomes the new origin of the data from now on

Figure 3.1: PCA example data, original data on the left, data with the means subtracted on the right, and a plot of the data

PCA Example –STEP 2

- Calculate the covariance matrix

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

- since the non-diagonal elements in this covariance matrix are positive, we should expect that both the x and y variable increase together.

PCA Example –STEP 3

- Calculate the eigenvectors and eigenvalues of the covariance matrix

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

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

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

PCA Example –STEP 3

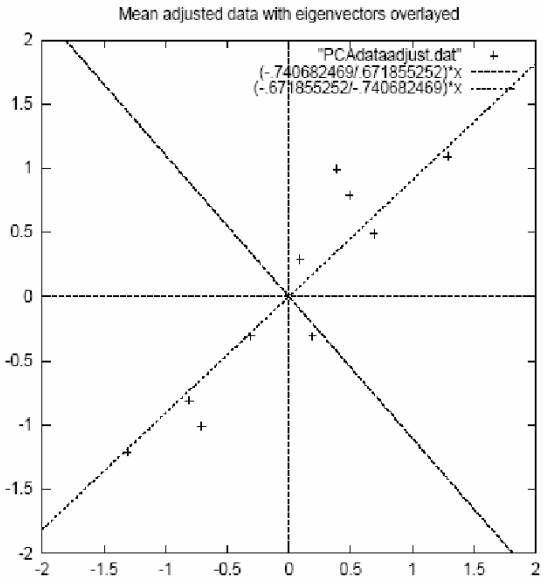


Figure 3.2: A plot of the normalised data (mean subtracted) with the eigenvectors of the covariance matrix overlayed on top.

- eigenvectors are plotted as diagonal dotted lines on the plot.
- Note they are perpendicular to each other.
- Note one of the eigenvectors goes through the middle of the points, like drawing a line of best fit.
- The second eigenvector gives us the other, less important, pattern in the data, that all the points follow the main line, but are off to the side of the main line by some amount.

PCA Example –STEP 4

- Feature Vector

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

  We can either form a feature vector with both of the eigenvectors:

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

  or, we can choose to leave out the smaller, less significant component and only have a single column:

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

PCA Example –STEP 5

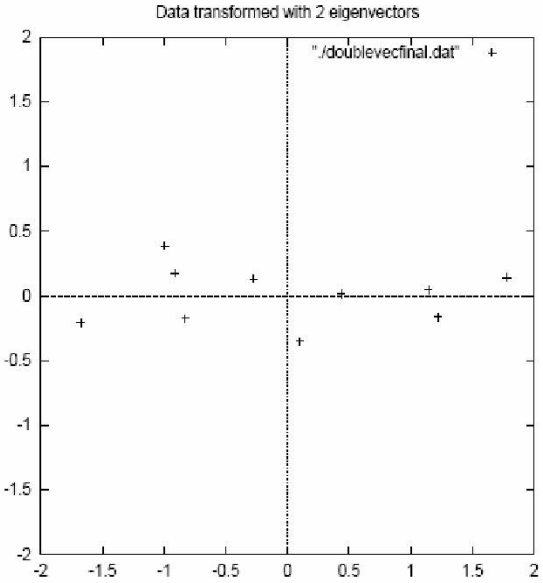
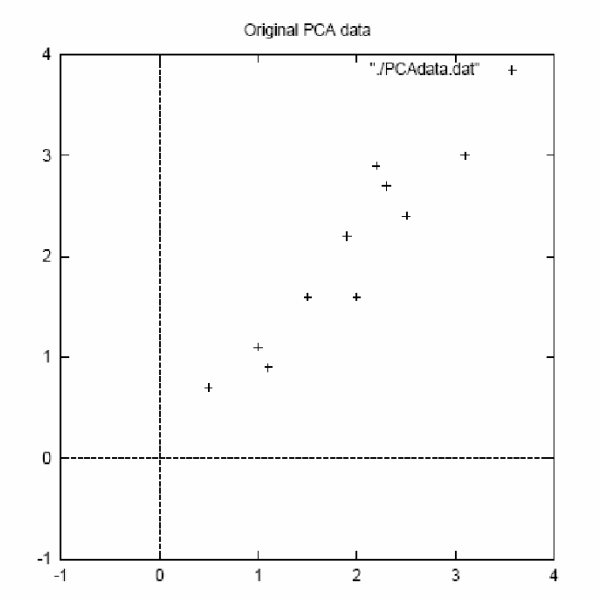


Figure 3.3: The table of data by applying the PCA analysis using both eigenvectors, and a plot of the new data points.

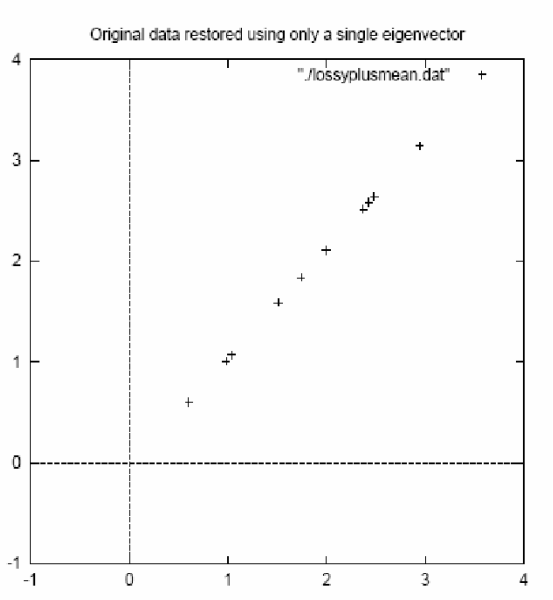
PCA Example : Approximation

- If we reduced the dimensionality, obviously, when reconstructing the data we would lose those dimensions we chose to discard. In our example let us assume that we considered only the x dimension…

PCA Example : Final Approximation

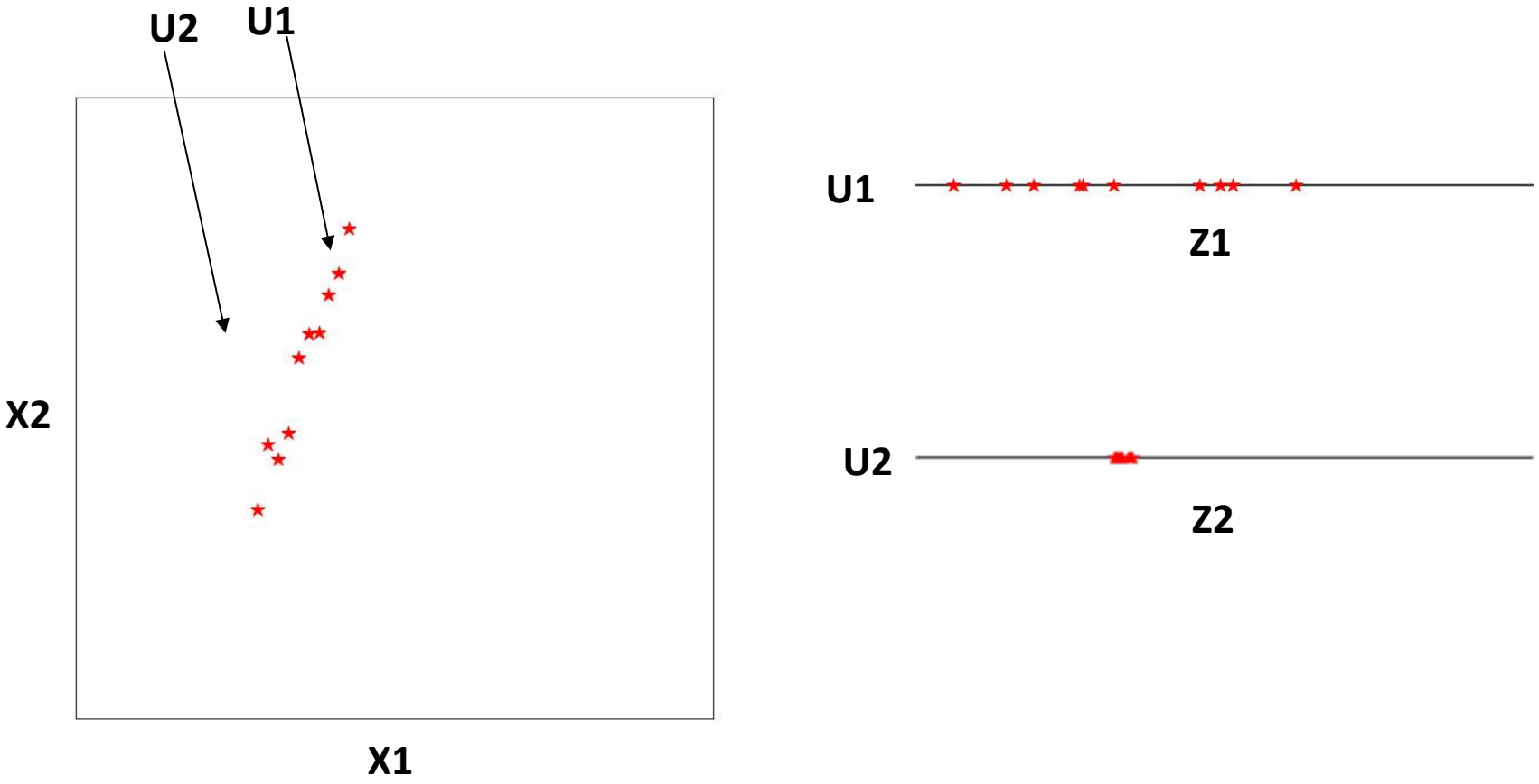


2D point cloud



Approximation using one eigenvector basis

Dimensionality and Representation



Principal Component Analysis

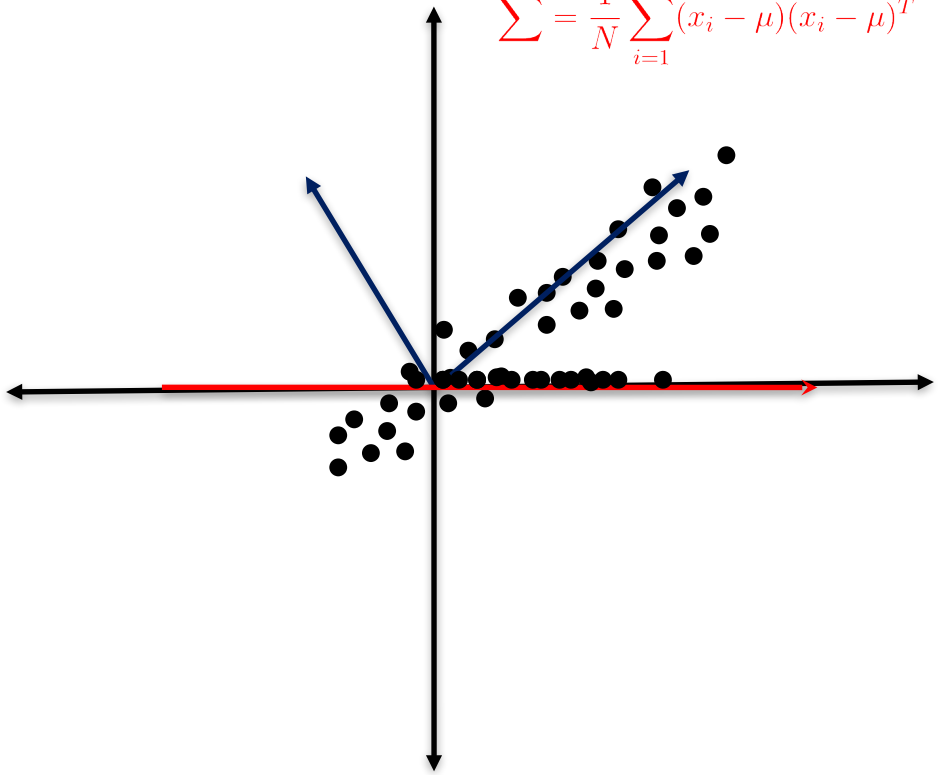
$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\sum = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)(x_i - \mu)^T$$

Goal: Find r-dim projection that best preserves variance

- Compute the Mean Vector μ and Covariance Matrix $\sum$
- Compute Eigen vectors and Eigen values $\sum \vec{v} = \lambda \vec{v}$
- Select top r-Eigen Vectors
- Project the points on the subspace spanned by them

$$z = U[x - \mu]$$



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

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

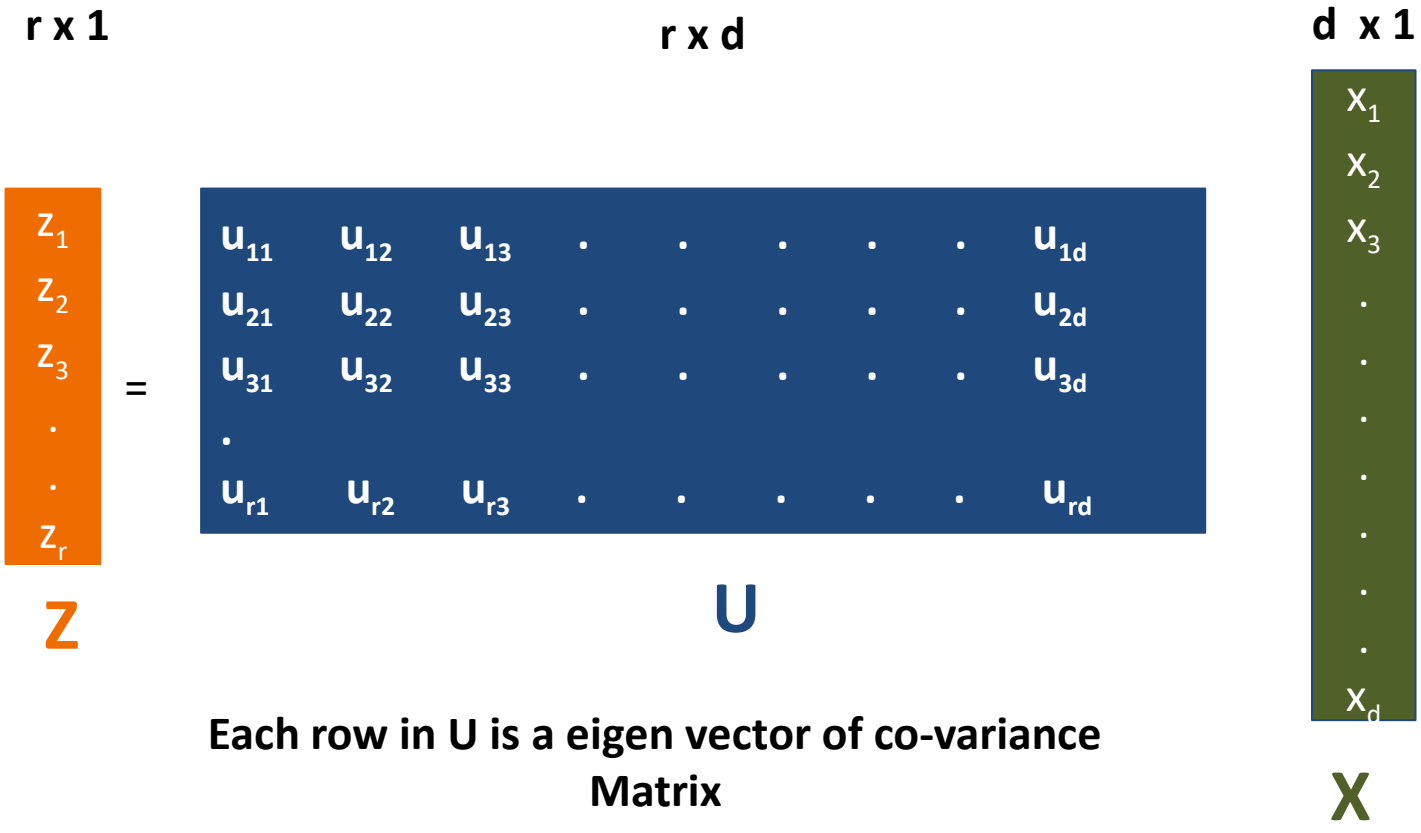
New Features as linear combination of old Features.

$$X' = AX$$

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

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

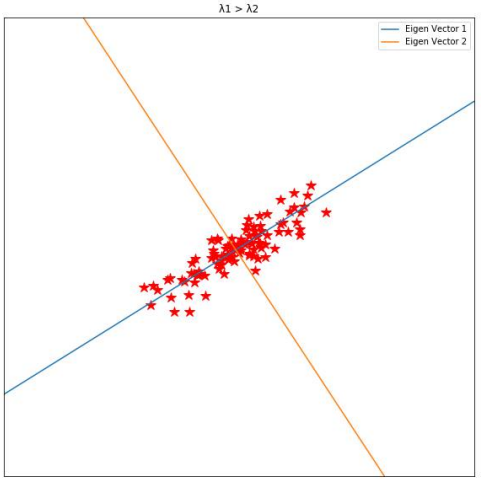
PCA based Feature Extraction



Each row in U is a eigen vector of co-variance Matrix

Appreciating PCA: “Rotation” and “Selection”

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z4 \end{bmatrix} = \begin{bmatrix} . & . & \mathbf{u_1}^T & . & . \\ . & . & \mathbf{u_2}^T & . & . \\ . & . & \mathbf{u_3}^T & . & . \\ . & . & \mathbf{u_4}^T & . & . \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$



Example in 2D

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

Old point in the New coordinate system

Appreciating PCA: Two Questions

- How many eigen vectors to select?
  - Ans: Eigen Vectors corresponding to the larger eigen values
- How much information is lost? Can we recover the old data/information from the new?

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

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

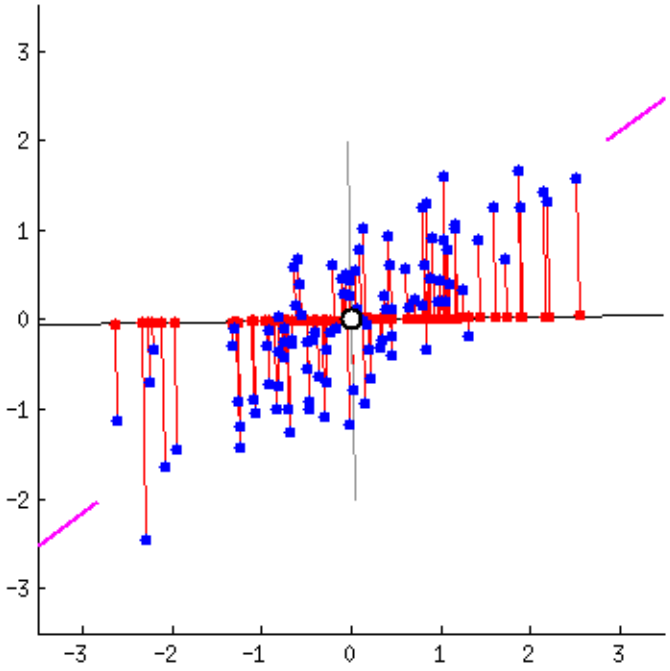
$$\mathbf{x}' = z_1\mathbf{u_1} + z_2\mathbf{u_2}$$

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

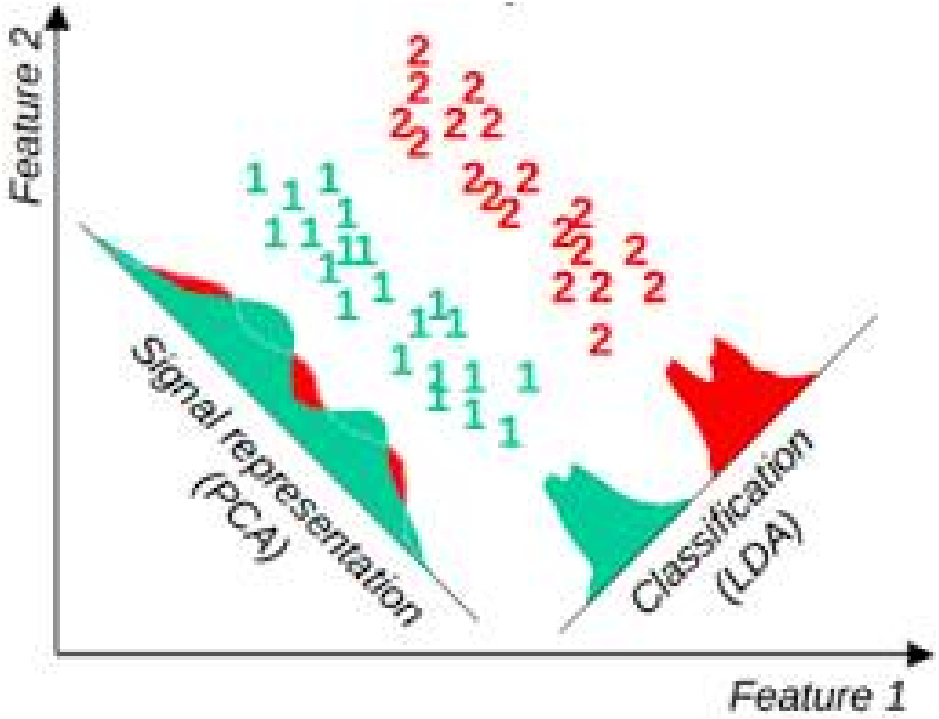
$$\text{Eg.} \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{d} \lambda_i} > 0.90$$

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

PCA - geometric intuition – preserving variance, reducing reconstruction error



Linear Discriminant Analysis (LDA)



Summary

- We often get raw data/logs/measurements
- Two problems:
  - Select good ones out of all
  - Define new ones as linear combination of existing
- Dimensionality reduction for
  - Compression/compaction
  - Classification/Discrimination