Course: MCA

Semester: IV

Paper Code/Name: DSE4T2 (Introduction to Machine Learning)

Topic: Principal Component Analysis (PCA)

Faculty Name: Dr. Tapan Kant

Email: tapan.kant@gmail.com

Principal Component Analysis (PCA)

- Proposed by Karl Pearson, 1901
- Find projections that capture the largest amount of variation in data
- Project data in the directions of maximum variance
- Find the principal vectors from the data
- PCA finds the most accurate data representation in a lower dimensional space

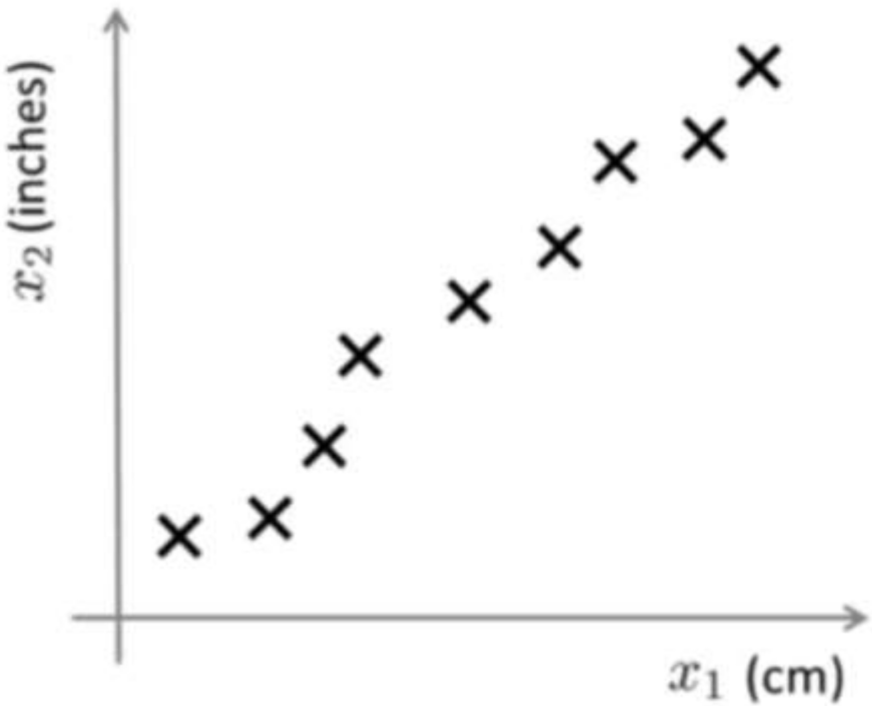
The problem

- Many modern data domains involve huge number of features / dimensions
  - Documents: thousands of words, millions of bigrams
  - Images: thousands to millions of pixels
  - Genomics: thousands of genes, millions of DNA polymorphisms

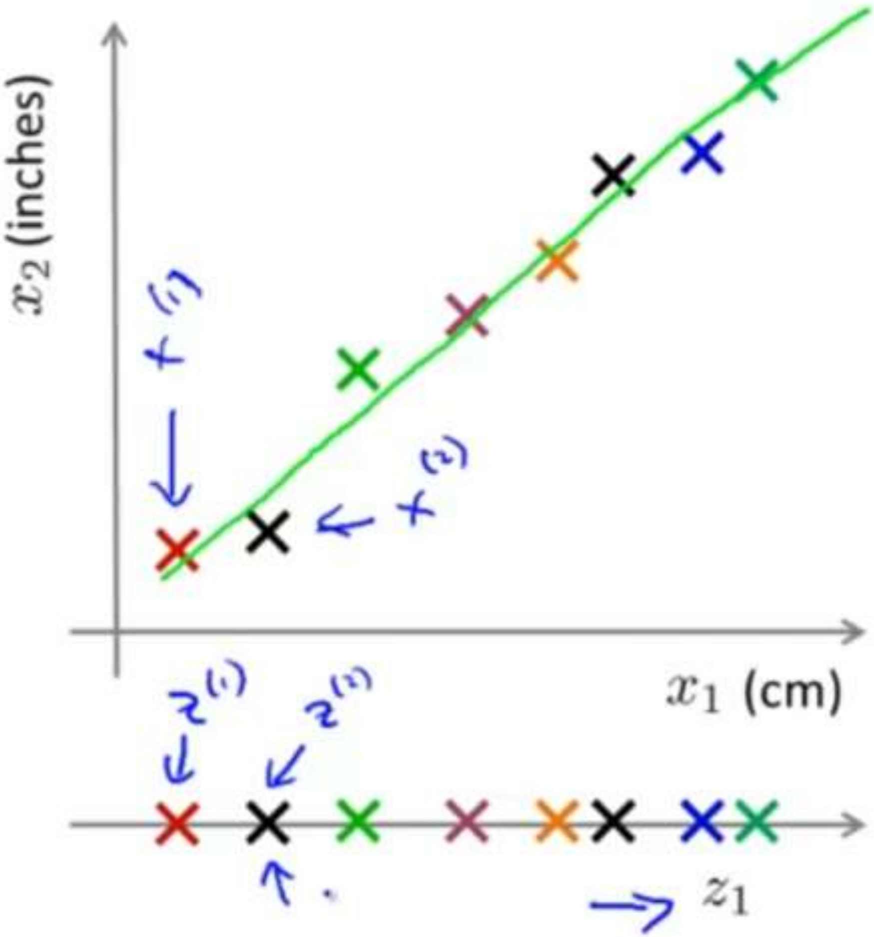
The problem contd…

- High dimensionality data has many costs
  - Redundant and irrelevant features degrade performance of some ML algorithms
  - Difficulty in interpretation and visualization
  - Computation may become infeasible (e.g. O(n3))
  - Curse of dimensionality

Data Compression



Reduce data from 2D to 1D



Reduce data from 2D to 1D

$x^{(1)} \in \mathbb{R}^2 \rightarrow z^{(1)} \in \mathbb{R}$

$x^{(2)} \in \mathbb{R}^2 \rightarrow z^{(2)} \in \mathbb{R}$

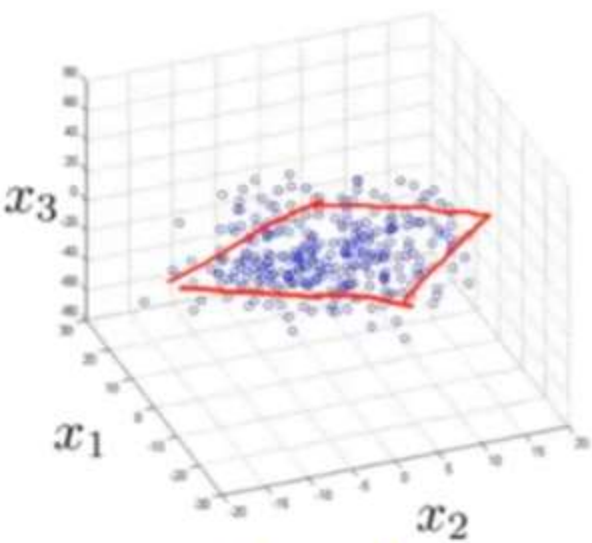
$\vdots$

$x^{(m)} \rightarrow z^{(m)}$

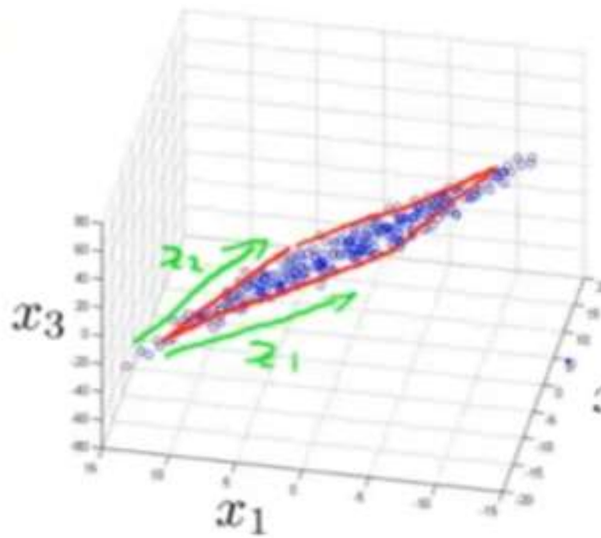
Data Compression

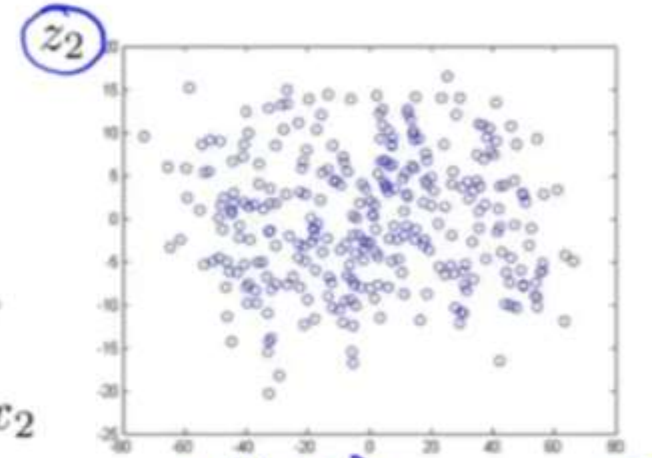
10000 → 1000

Reduce data from 3D to 2D



$x^{(i)} \in \mathbb{R}^3$

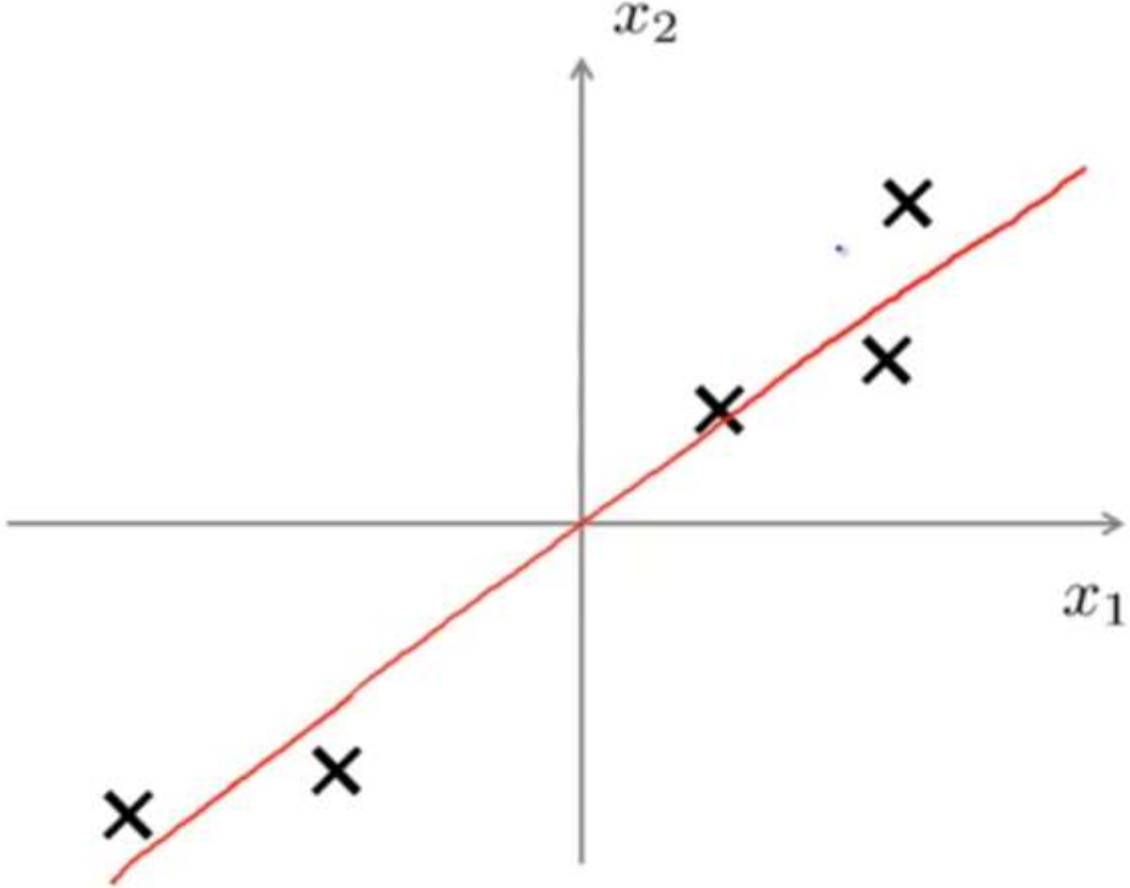




$z^{(i)} \in \mathbb{R}^2$

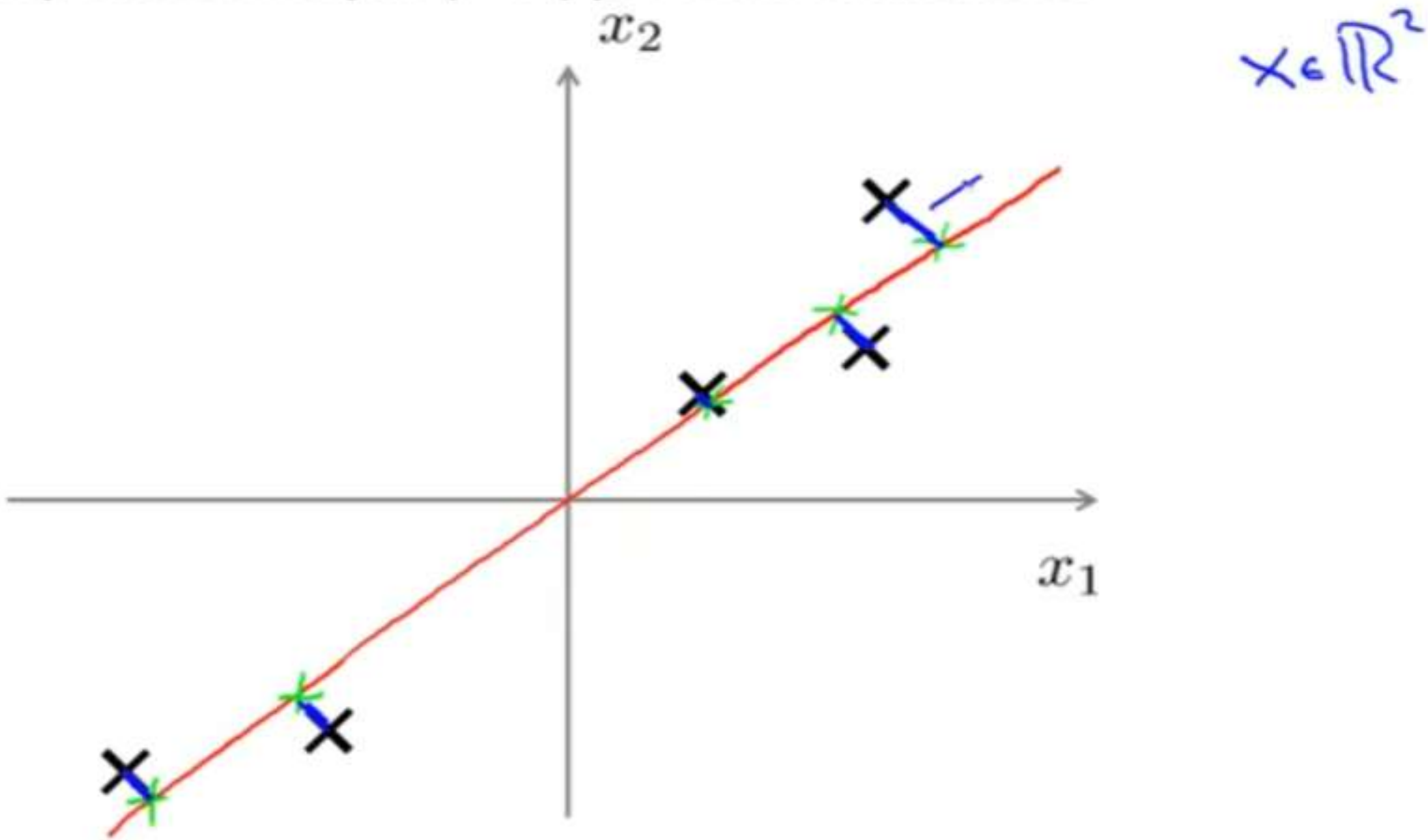
$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \qquad z^{(i)} = \begin{bmatrix} z_1^{(i)} \\ z_2^{(i)} \end{bmatrix}$$

Principal Component Analysis (PCA) problem formulation

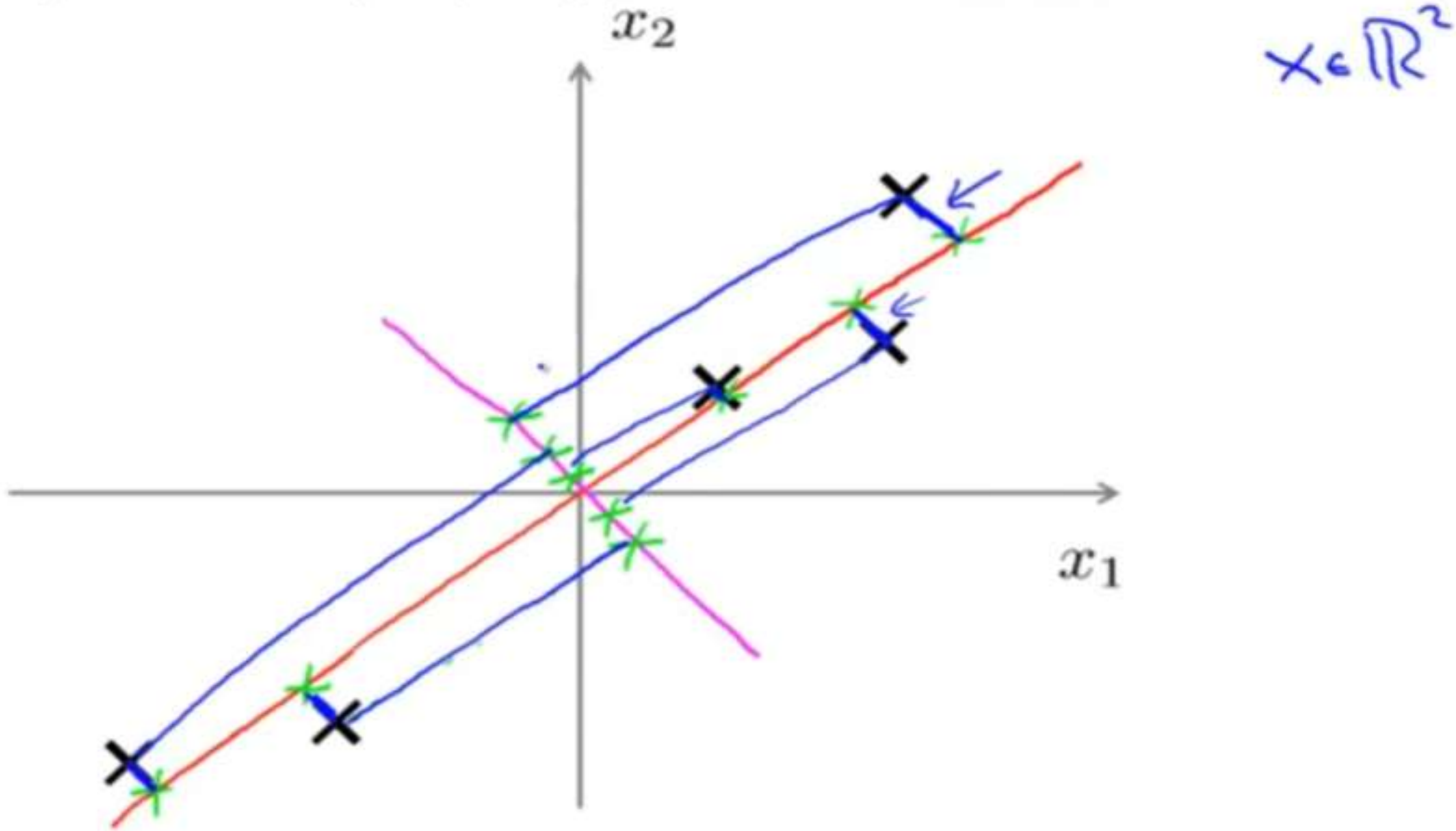


$x \in \mathbb{R}^2$

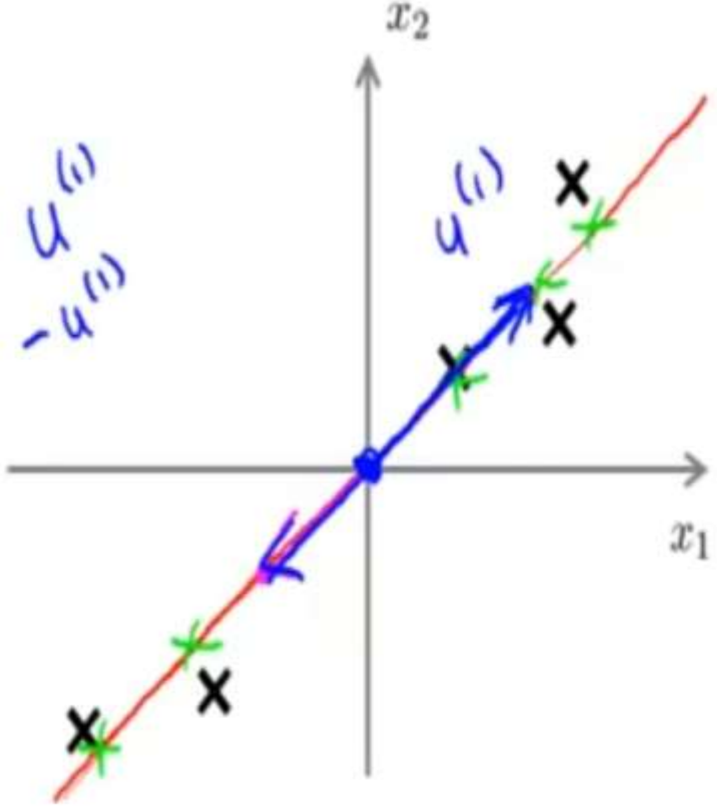
Principal Component Analysis (PCA) problem formulation



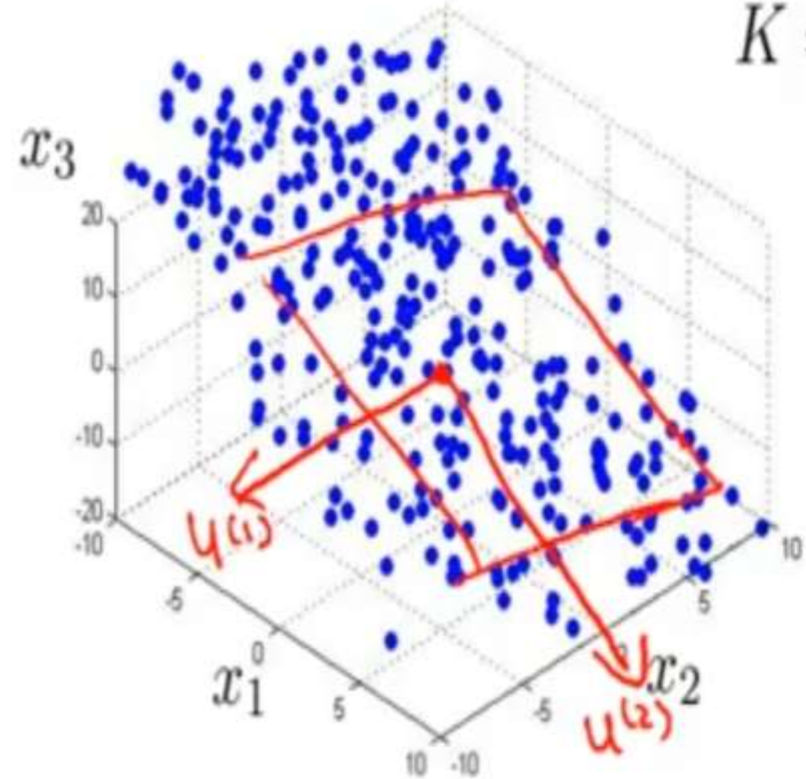
Principal Component Analysis (PCA) problem formulation



Principal Component Analysis (PCA) problem formulation



$3D \to 2D$
$K = 2$



Reduce from 2-dimension to 1-dimension: Find a direction (a vector $u^{(1)} \in \mathbb{R}^n$) onto which to project the data so as to minimize the projection error.

Reduce from n-dimension to k-dimension: Find k vectors $u^{(1)}, u^{(2)}, \ldots, u^{(k)}$ onto which to project the data, so as to minimize the projection error.

How to perform PCA

PCA Example – Data

- Original data

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

STEP 1

- Subtract the mean

- from each of the data dimensions. All the x values have average (x) subtracted and y values have average (y) subtracted from them. This produces a data set whose mean is zero.

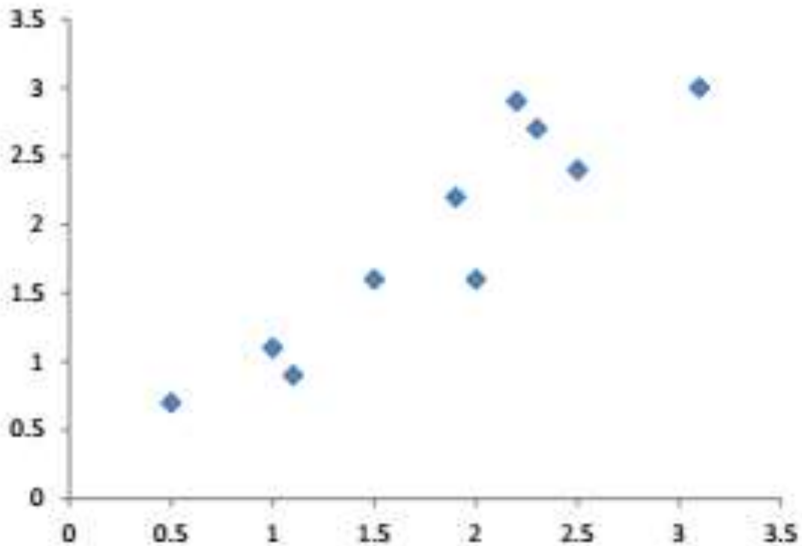
- Subtracting the mean makes variance and covariance calculation easier by simplifying their equations. The variance and co-variance values are not affected by the mean value.

STEP 1

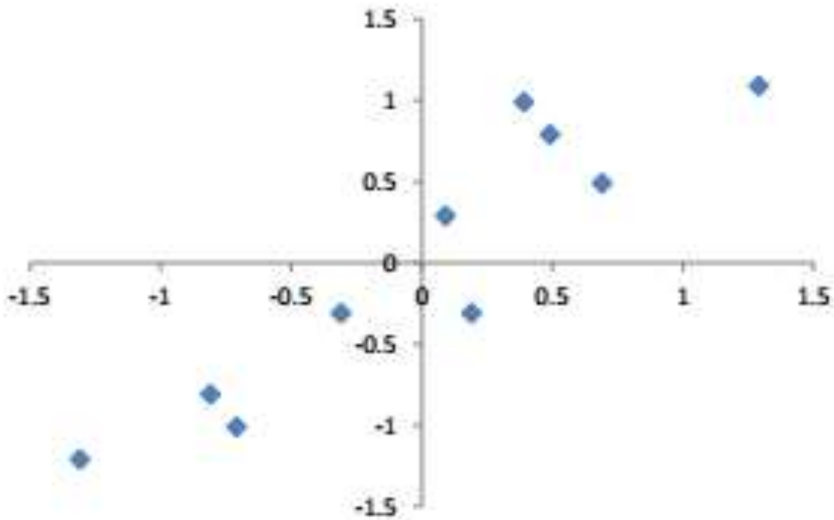
- Zero-mean data

0.69	0.49
-1.31	-1.21
0.39	0.99
0.09	0.29
1.29	1.09
0.49	0.79
0.19	-0.31
-0.81	-0.81
-0.31	-0.31
-0.71	-1.01

STEP 1



Original



Zero-mean

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.

STEP 3

- Calculate the eigenvectors and eigenvalues of the covariance matrix

eigenvalue indicates the percentage of transformation present along a particular direction

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

Eigenvector of a linear transformation is a non-zero vector that changes at most by the scalar factor when that linear transformation is applied to it.

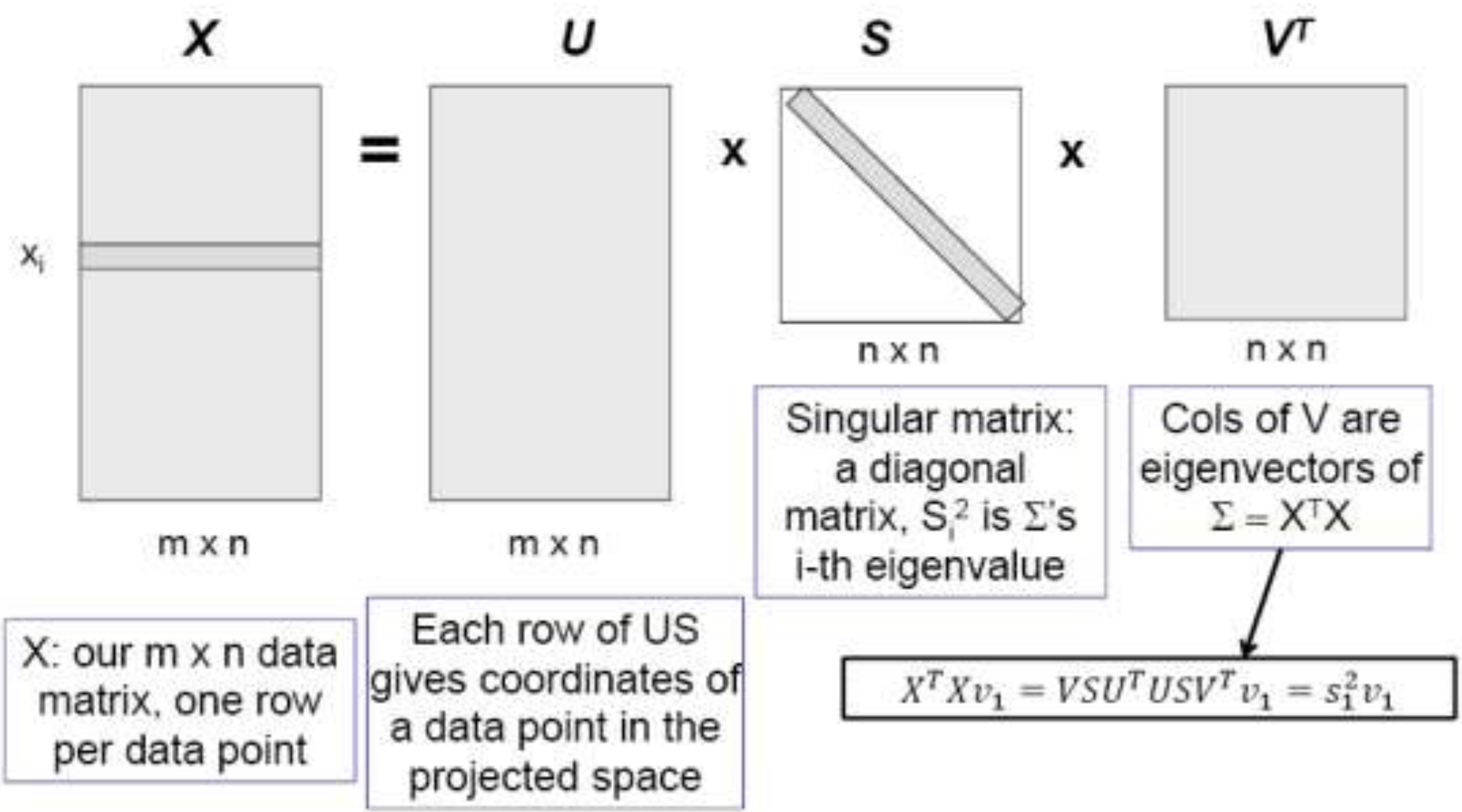
$$\text{eigenvectors} = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

$\boxed{\text{Def'n}}$ Let A be an $n \times n$ matrix. A scalar λ is called an eigenvalue of A if there is a nonzero vector $\bar{x}$ such that $A\bar{x} = \lambda\bar{x}$. Such a vector $\bar{x}$ is called an eigenvector of A corresponding to λ.

Show that $\bar{x} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ is an eigenvector of $A = \begin{pmatrix} 3 & 2 \\ 3 & -2 \end{pmatrix}$ corresponding to $\lambda = 4$

- Practical issue: covariance matrix is n x n.
  - E.g. for image data $\mathbf{\Sigma}$ = 32768 x 32768.
  - Finding eigenvectors of such a matrix is slow.

- Singular value decomposition (SVD) to the rescue!
  - Can be used to compute principal components.
  - Efficient implementations available, e.g. MATLAB svd.



X
U
S
V^T
=
x
x
x_i
m x n
m x n
n x n
n x n
Singular matrix: a diagonal matrix, S_i^2 is Σ's i-th eigenvalue
Cols of V are eigenvectors of $\Sigma = X^T X$
$X^T X v_1 = VSU^T USV^T v_1 = s_1^2 v_1$
X: our m x n data matrix, one row per data point
Each row of US gives coordinates of a data point in the projected space

- Create mean-centered data matrix $\mathbf{X}$.

- Solve SVD: $\mathbf{X} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T$.

- Columns of $\mathbf{V}$ are the eigenvectors of $\mathbf{\Sigma}$ sorted from largest to smallest eigenvalues.

- Select the first k columns as our k principal components.

Conclusion

- Many modern data domains involve huge number of features
- Irrelevant features degrade performance of some ML algorithms
- Difficulty in interpretation and visualization
- Data Compression
- PCA finds the most accurate data representation in a lower dimensional space

Thank you