25.11.2024

BITS F464: Machine Learning (1st Sem 2024-25)

UNSUPERVISED LEARNING: K-MEANS CLUSTERING, GAUSSIAN MIXTURE MODEL (GMM), PRINCIPAL COMPONENT ANALYSIS (PCA)

Chittaranjan Hota, Sr. Professor
Dept. of Computer Sc. and Information Systems
hota@hyderabad.bits-pilani.ac.in

Supervised Vs. Un-supervised

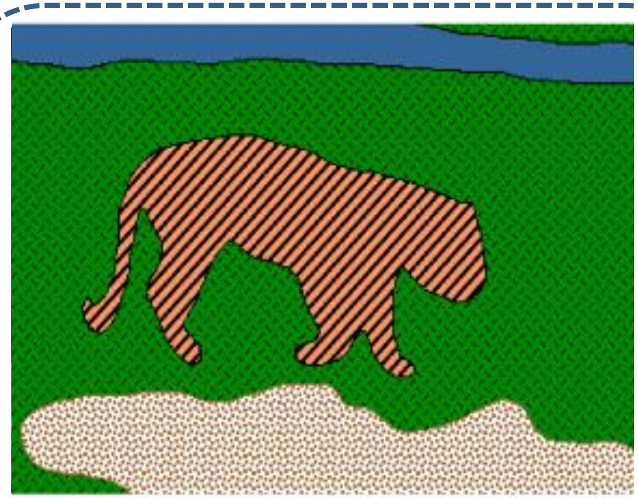
- Supervised: Learning from labelled data
  - Train data: (X, Y) for Input X, Y is the label
  - (Sunny, Evening, Moderate_Temp: Play)

  ⇦ Classification/ Regression.

- Unsupervised: Learning from un-labeled data
  - Train data: X

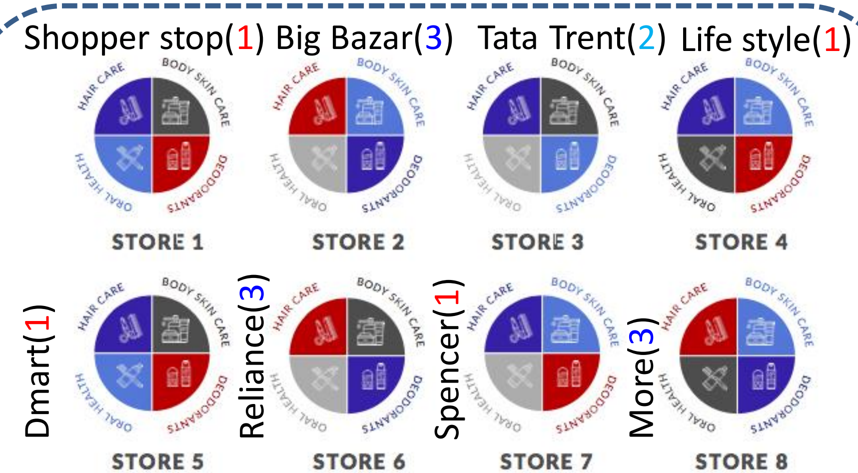
  ⇦ Clustering, Dimensionality reduc., Anomaly detection.

- Clustering: Its primary goal is to group similar data points together into clusters based on their intrinsic characteristics or features.



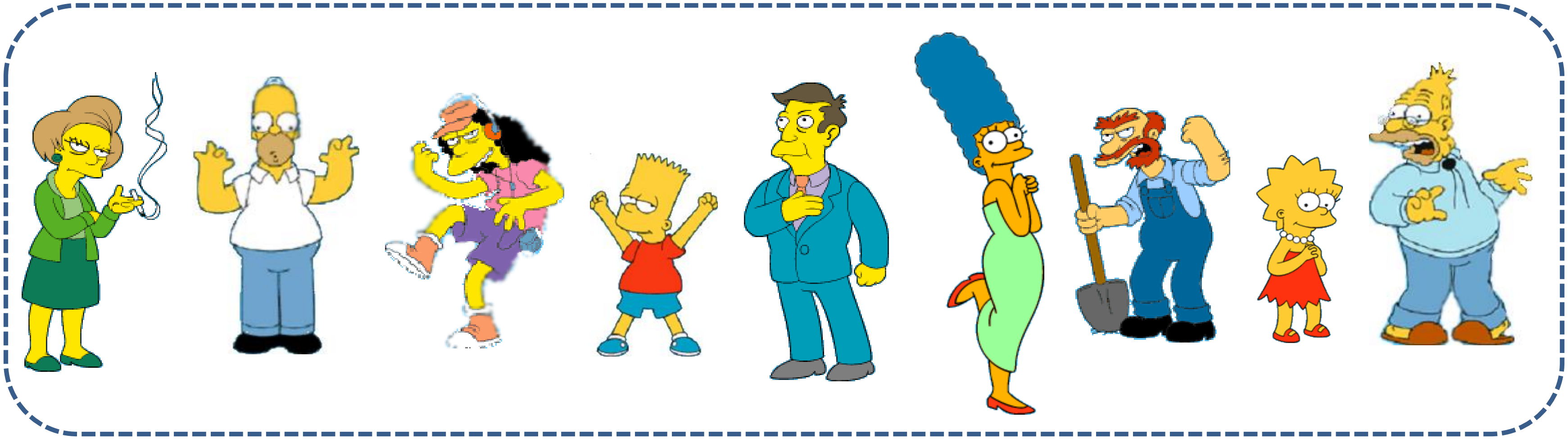
clusters?

(Image segmentation)

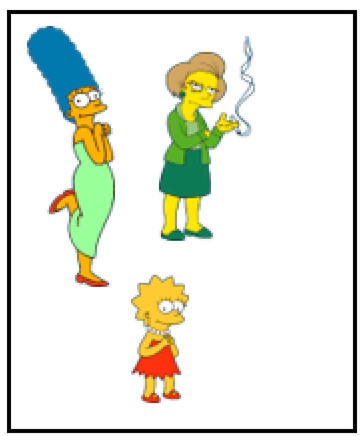


(Retail clustering)

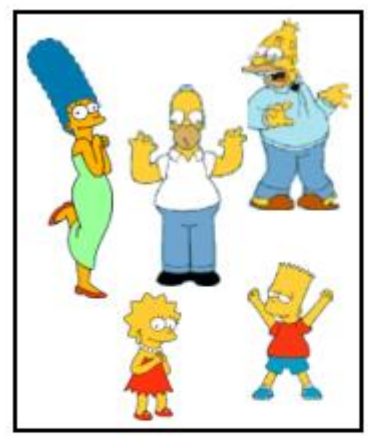
Clustering is Subjective: How to group?



Male



Female



A family

School employees

Distance metrics: Euclidean distance, Manhattan distance, Cosine similarity etc.

K-Means Algorithm

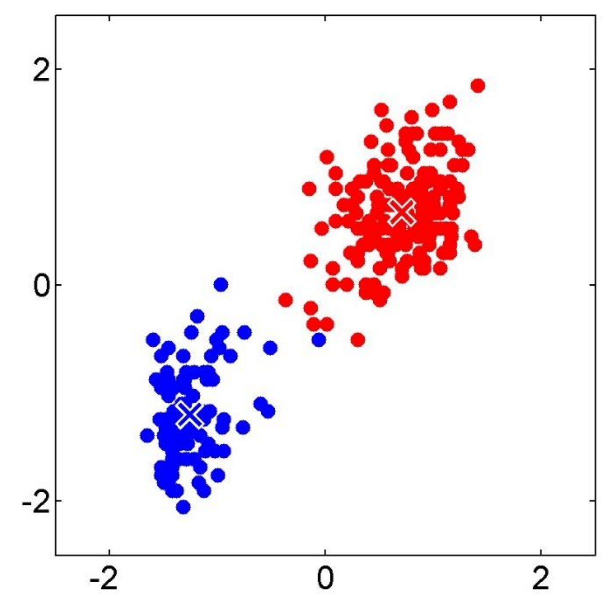
- Goal: represent a data set in terms of K clusters each of which is summarized by a prototype $\boldsymbol{\mu}_k$

- Initialize prototypes, then iterate between two phases:
  - E-step: assign each data point to nearest prototype
  - M-step: update prototypes to be the cluster means

- Responsibilities assign data points to clusters: $r_{nk} \in \{0, 1\}$ such that:

$$\sum_k r_{nk} = 1 \qquad (r_{nk}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

- Example 5 data points and 3 clusters:



Distortion measure (Eq.1)

K-Means Cost Function:

data

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

responsibilities

prototypes

Sum of the squares of the distances of each data point to its μ_k .

Continued…

- How to determine r_{nk} in Eq. (1) keeping μ_k fixed ?
  - As J is a linear function of r_{nk}, $r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$
- How to determine μ_k in Eq. (1) keeping r_{nk} fixed ?
  - As J is a quadratic function of μ_k, it can be minimized by setting its derivative to 0:
  - $$\boxed{2\sum_{n=1}^{N} r_{nk}(\mathbf{x}_n - \boldsymbol{\mu}_k) = 0 \quad \Rightarrow \quad \boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}}$$
- The two phases of re-assigning data points to clusters and re-computing the cluster means are repeated in turn until there is no further change in the assignments.

K-Means Example

Let there be 6 data points: (1,2), (2,3), (3,4), (5,6), (7,8), (9,10) and K=2.

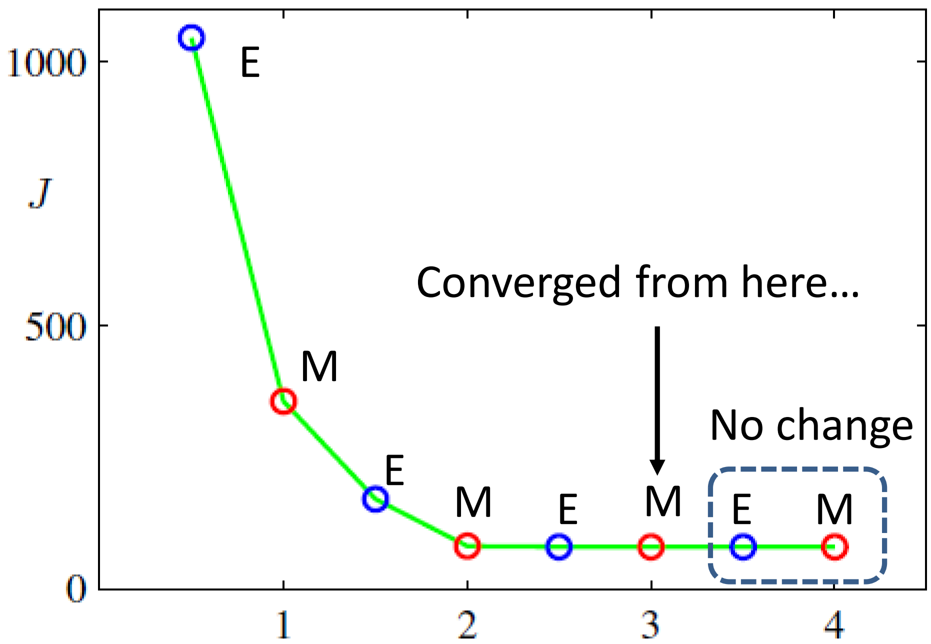
Point	Dist. to Centroid 1	Dist. to Centroid 2	Assigned Cluster
(1,2)	0	11.31	1
(2,3)	1.41	10.05	?
(3,4)	2.83	8.49	?
(5,6)	5.66	5.66	?
(7,8)	8.49	2.83	?
(9,10)	11.31	0	?

—> Cluster 1: (1,2), (2,3) and (3,4) and Cluster 2: (5,6), (7,8), (9,10)

Step 3: Compute New Centroids Centroid 1:{[(1+2+3)/3], [(2+3+4)/3]} = (2,3)
Centroid 2:{[(5+7+9)/3], [(6+8+10)/3]} = (7,8)

Step 4: Repeat until Convergence:Reassign points,Recalculate centroids,Stop when no change.

K-Means Convergence

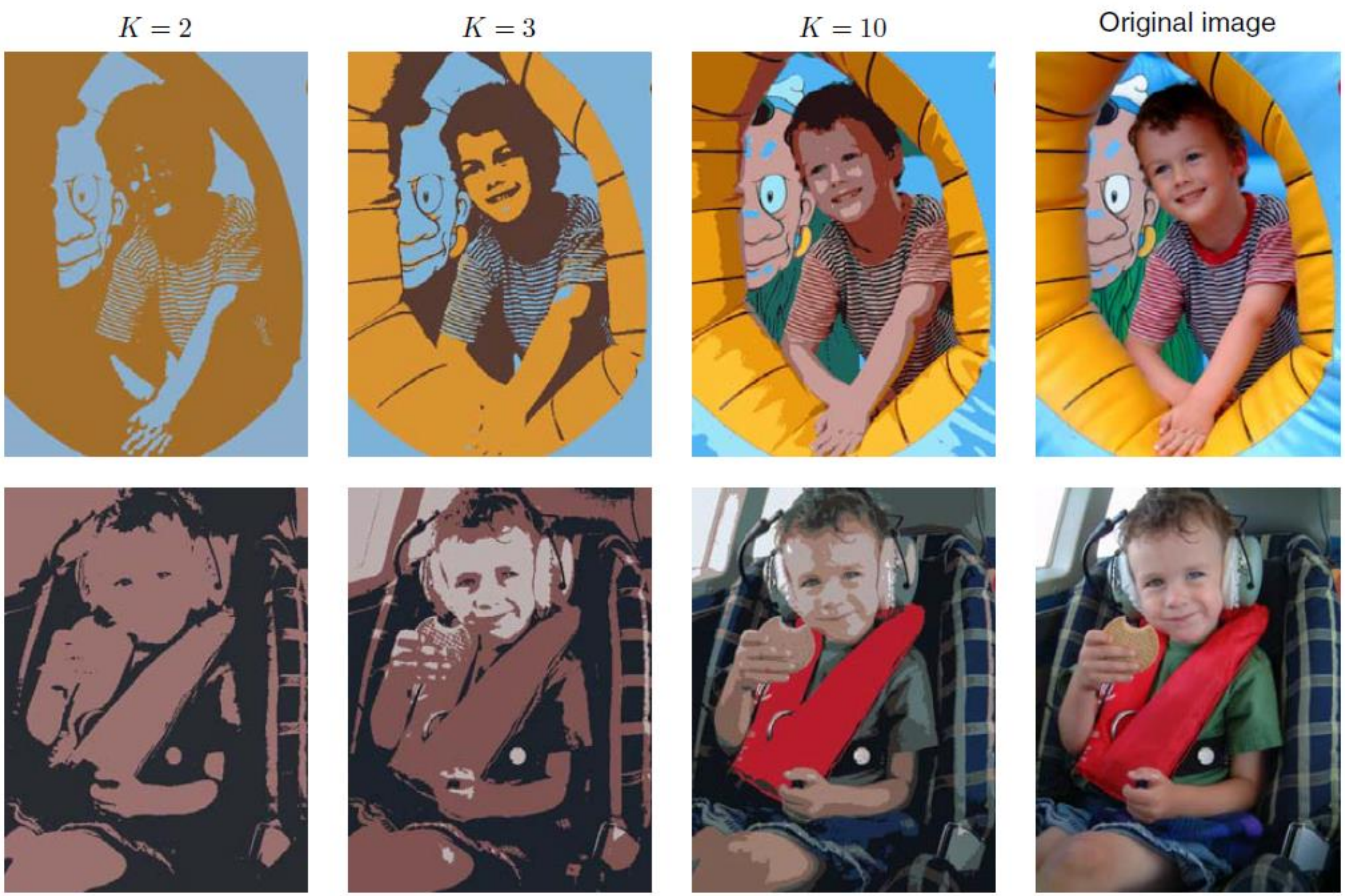


Each E and M successively minimize J, hence algorithm will converge.

How to choose a good value of K: Start with K=1. Then increase the value of K (up to a certain upper limit). Usually, the **variance** (the summation of the square of the distance from the "owner" center for each point) will decrease rapidly. After a certain point, it will decrease slowly. When you see such a behavior, you know you've overshot the K-value. Stop it there and that is the final value of K.

K-Means can converge to a local minima: Solution: K-Means++ initialization

An Application of K-Means: Segmentation



-(Problem) Hard assignments of data points to clusters: small shift of a data point can flip it to a different cluster.

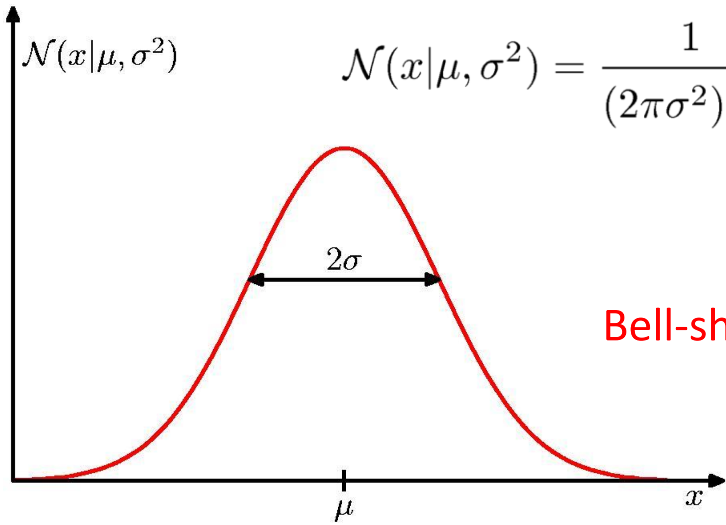
Solution: Replace 'hard' clustering of K-means with 'soft' probabilistic assignments

(*Gaussian Mixture Model*)

(Image source: Bishop's Text)

The Gaussian Distribution

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



Bell-shaped

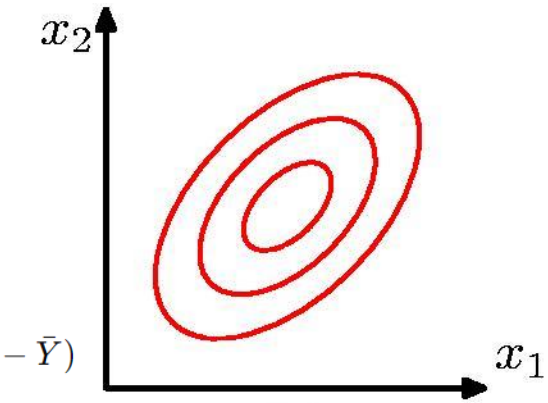
Maximum likelihood

$$\hat{\mu} = \frac{1}{N}\sum_i x^{(i)}$$

$$\hat{\Sigma} = \frac{1}{N}\sum_i (x^{(i)} - \hat{\mu})^T (x^{(i)} - \hat{\mu})$$

(Univariate: probability distribution of a single random variable: Single dimension. Characterized by mean, and variance.)

$$\mathrm{Cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{X})(y_i - \bar{Y})$$



covariance

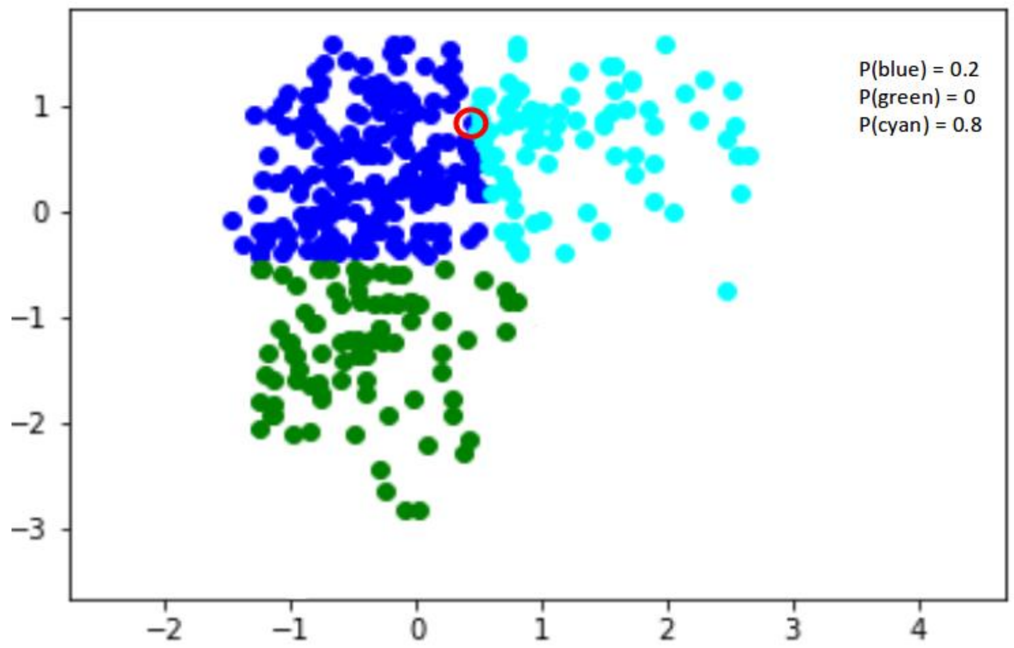
mean

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

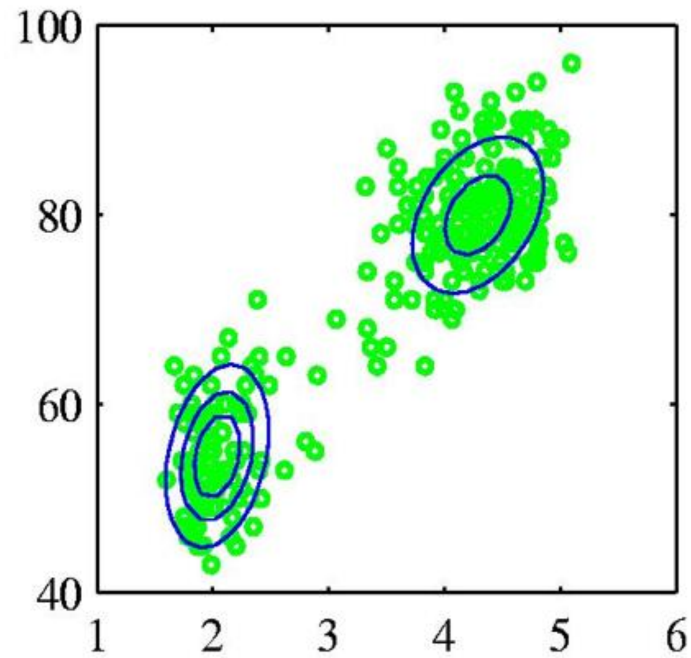
(Multi-variate: joint-probability distribution of multiple random variables. Ellipsoidal surface in n-dimensional space. Characterized by mean vector and co-variance matrix.)

Gaussian Mixture Model (GMM)

- Clusters modeled by Gaussians and not by their Means. EM algorithm assigns data point to a cluster with some probability.



Img. Source: https://www.analyticsvidhya.com/



Mixture (Superposition) of two Gaussians

Continued…

- Combine simple models into a complex model:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \underbrace{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}_{\text{Normal/ Gaussian}}$$

Mixing coefficient: Relative importance of each component 'k' in the mixture.

Mixture of Gaussians

$$\forall k : \pi_k \geqslant 0 \qquad \sum_{k=1}^{K} \pi_k = 1$$

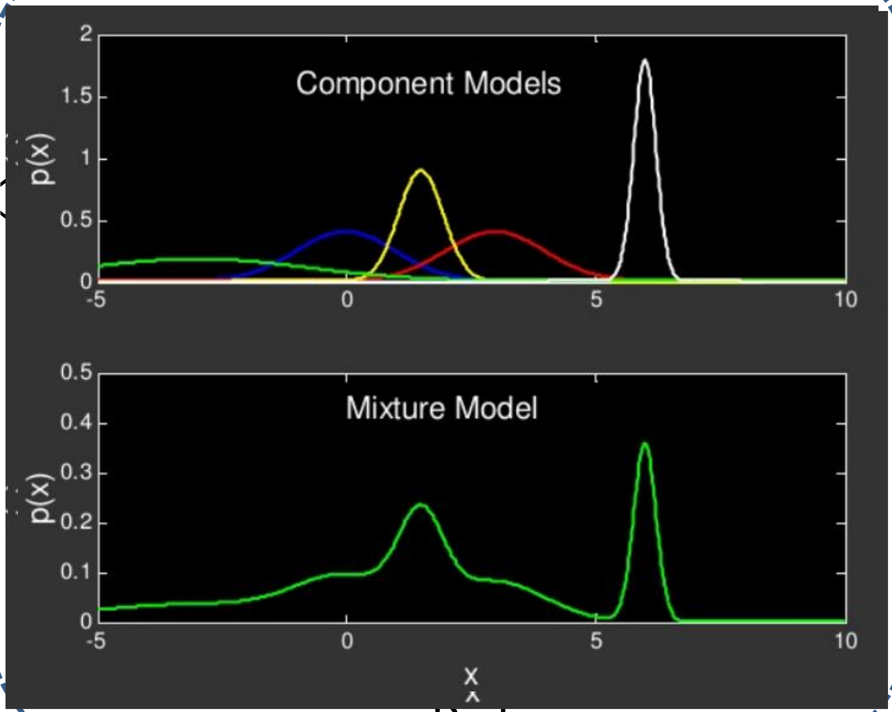
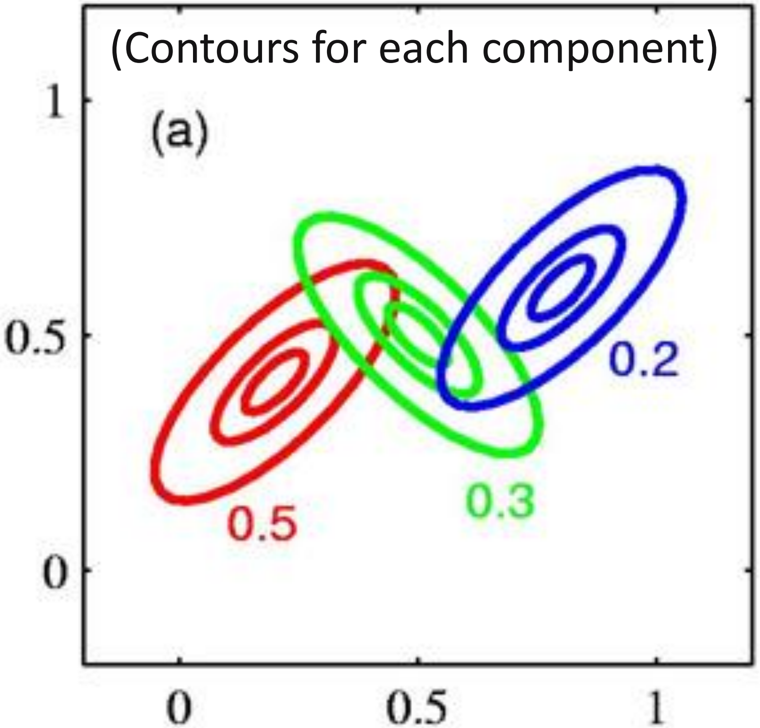
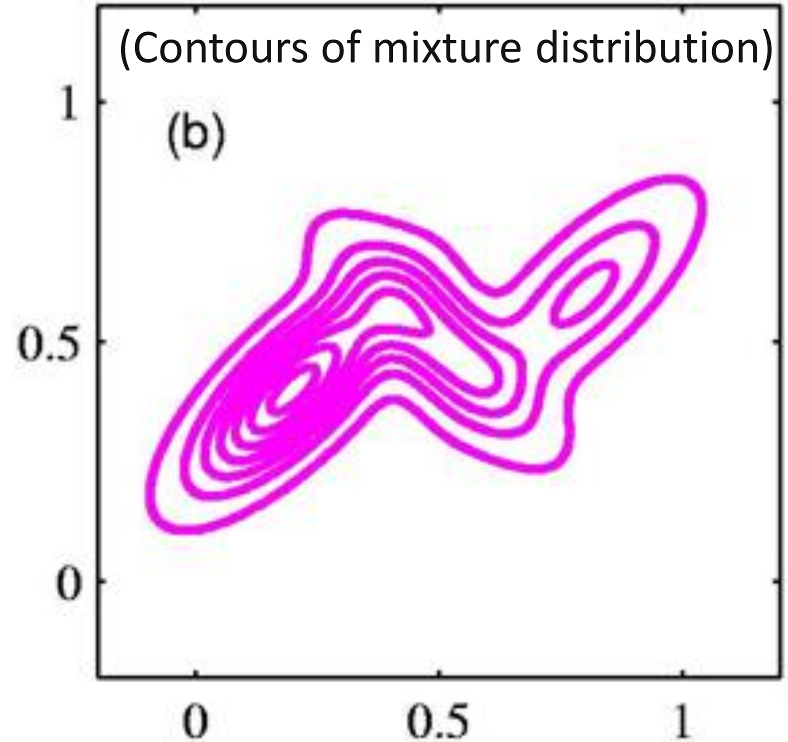


Image source: https://www.inf.u-szeged.hu/~tothl/

By increasing the number of components the curve defined by the mixture model can take basically any shape, so it is much more flexible than just one Gaussian.

Contour Plots of Mixture Models





$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(k)p(\mathbf{x}|k)$$

Maximum likelihood:

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Summation of ‘k’ inside the log is problematic. No closed-form maximum. We will use EM algorithm.

EM Algorithm to solve GMM

Start with parameters describing each cluster:

Mean 'μ_c', Covariance 'Σ_c', and size 'π_c'.

E-step (Expectation):

For each datum x_i:

Compute 'r_{ic}', the probability that it belongs to cluster 'c':

1. Compute its probability under model 'c'
2. Normalize to sum to one (over clusters 'c')

$$r_{ic} = \frac{\pi_c \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \; ; \; \mu_{c'}, \Sigma_{c'})}$$

If x_i is very likely under the c^{th} Gaussian, it gets high weight.

Denominator just makes the sum to one.

Continued…

Start with assignment probabilities: r_{ic}

Update parameters: mean μ_c , Covariance Σ_c, and 'size' π_c

M-step (Maximization):

For each cluster (Gaussian) x_c

Update its parameters using the (weighted) data points:

a. $N_c = \sum_i r_{ic}$ ... Total Responsibility allocated to cluster 'c'.

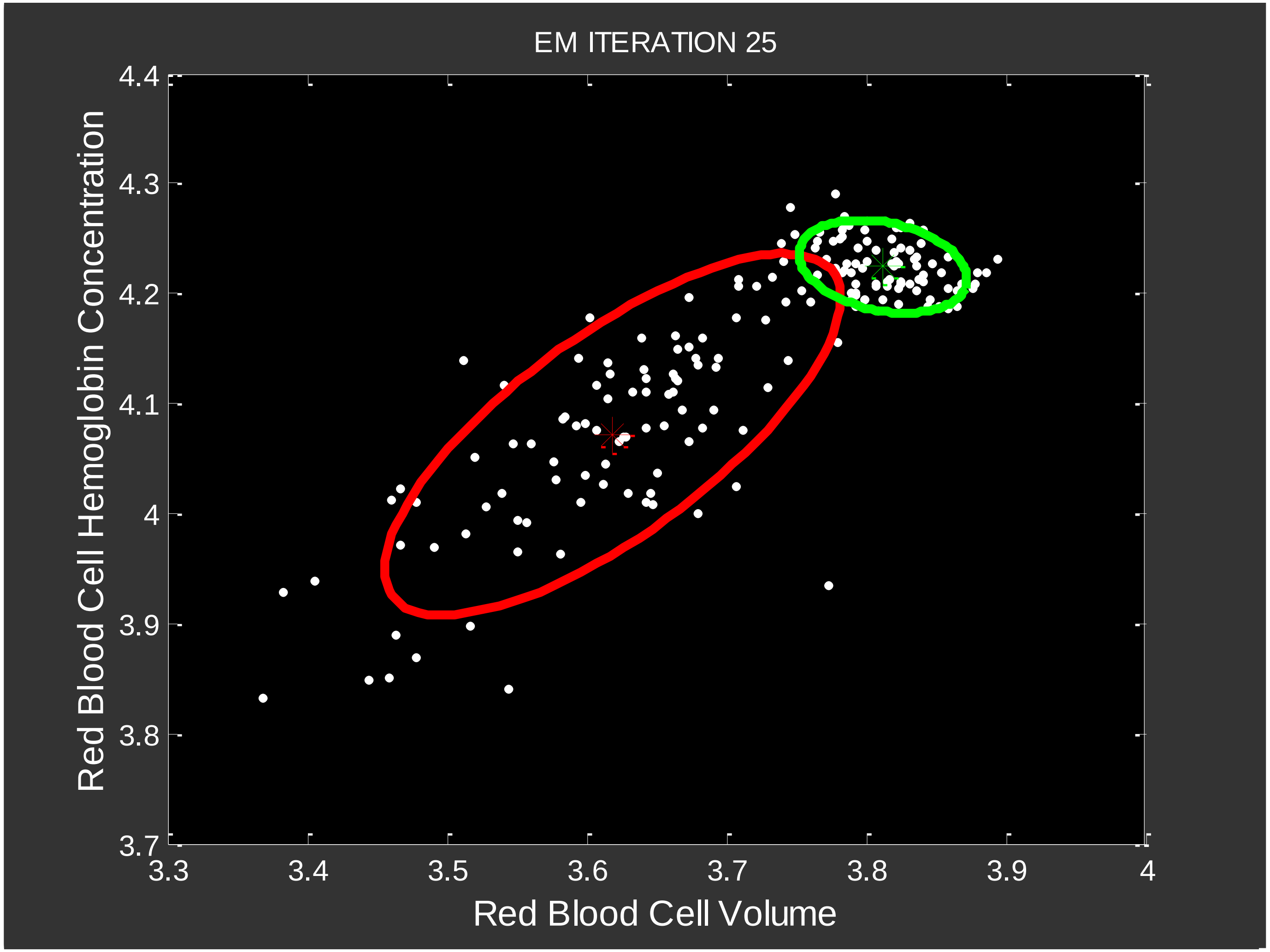
b. $\pi_c = \frac{N_c}{N}$ ... Fraction of total assigned to cluster 'c'.

c. $\mu_c = \frac{1}{N_c} \sum_i r_{ic} x_i$ ... Weighted mean of assigned data.

d. $\Sigma_c = \frac{1}{N_c} \sum_i r_{ic} (x_i - \mu_c)^T (x_i - \mu_c)$ ... Weighted covariance

Each 'E' and 'M' step increases the log likelihood: $\log p(\underline{X}) = \sum_i \log \left[\sum_c \pi_c \, \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c) \right]$

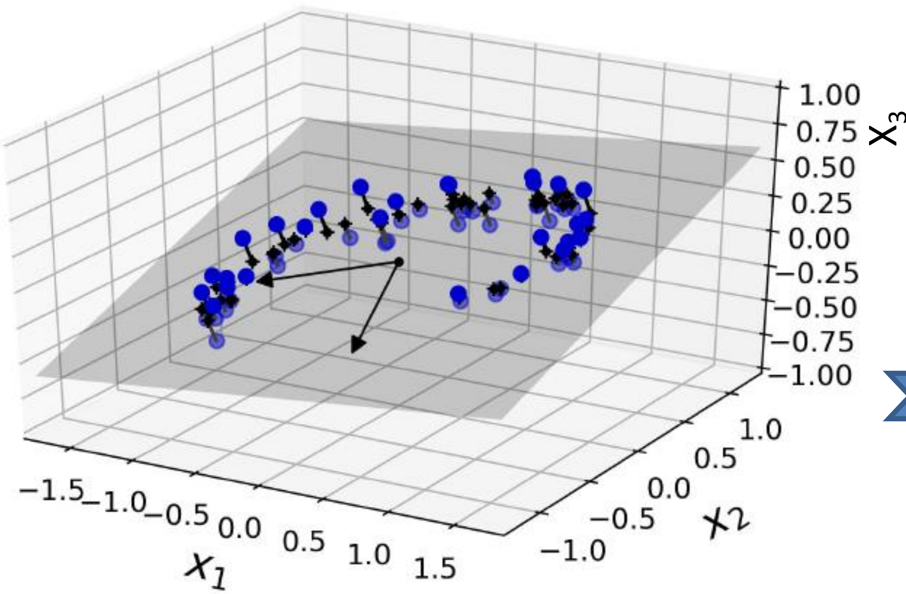
Expectation-Maximization in Action!



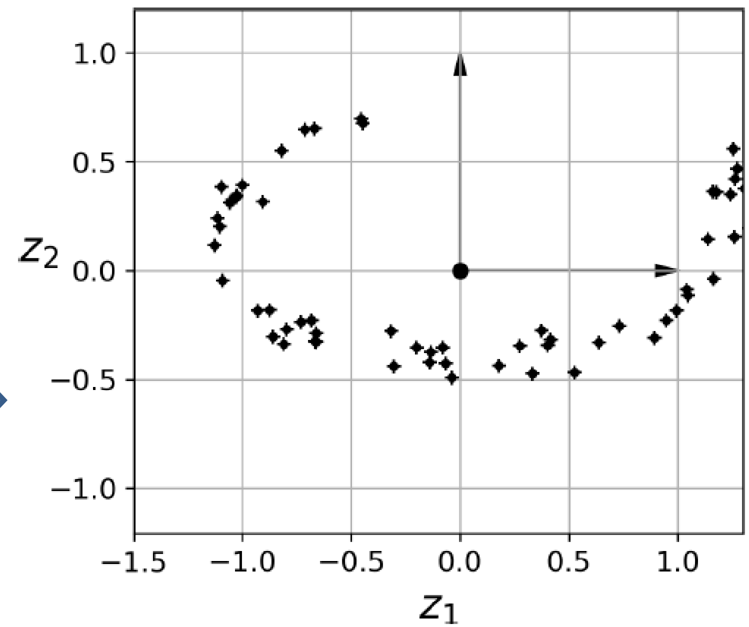
Img. Source: P. Smyth's ICML Presentation

What is Dimensionality Reduction?

- Reducing the number of features/ dimensions of the dataset by preserving as much information as possible while discarding the less important ones.



(A 3D dataset lying close to a 2D subspace)



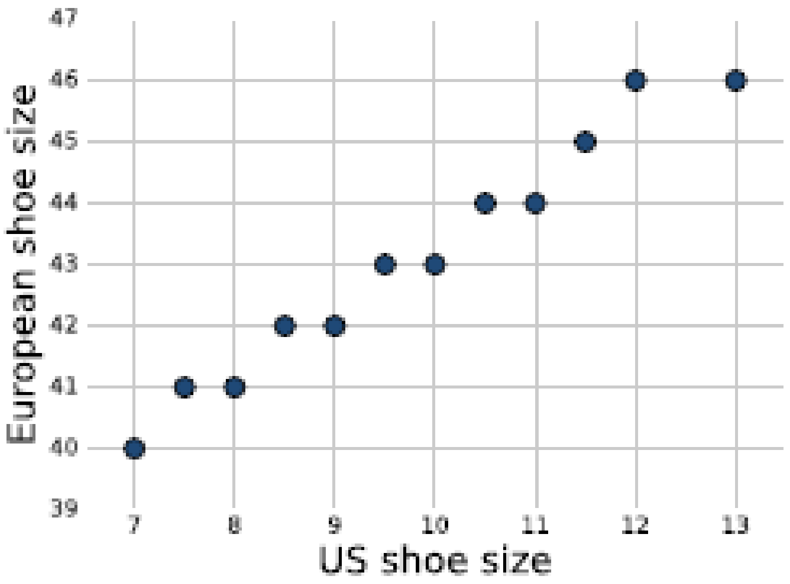
(The new 2D dataset after reduction)

Image source: Aurelien Geron's text

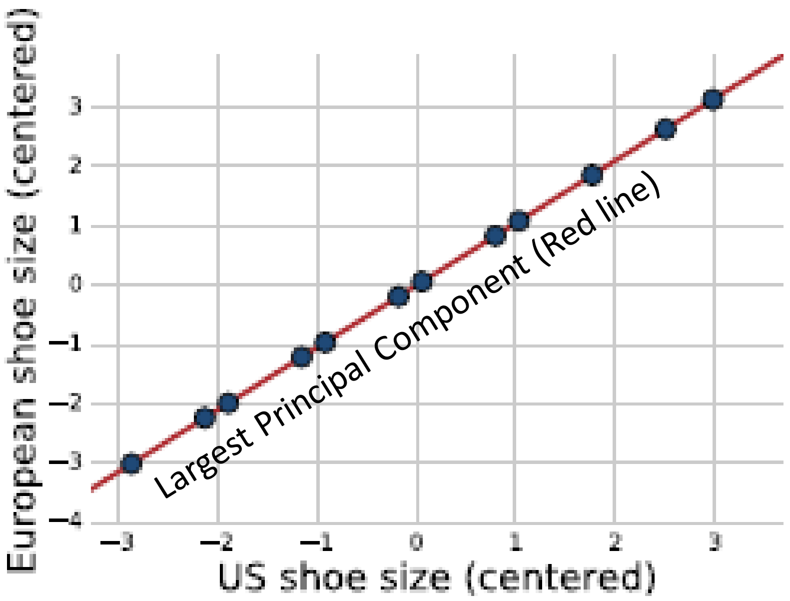
- Ex Tennis: (Service speed, Serve accuracy, Forehand effectiveness, Backhand effectiveness, Net play success) might map to 2 Principal Components.

- Which one might contribute less to both Principal components and hence irrelevant?

Another Example



(2-dimensional data points)



(1-dimensional that captures most variance in the data)

Curse of Dimensionality

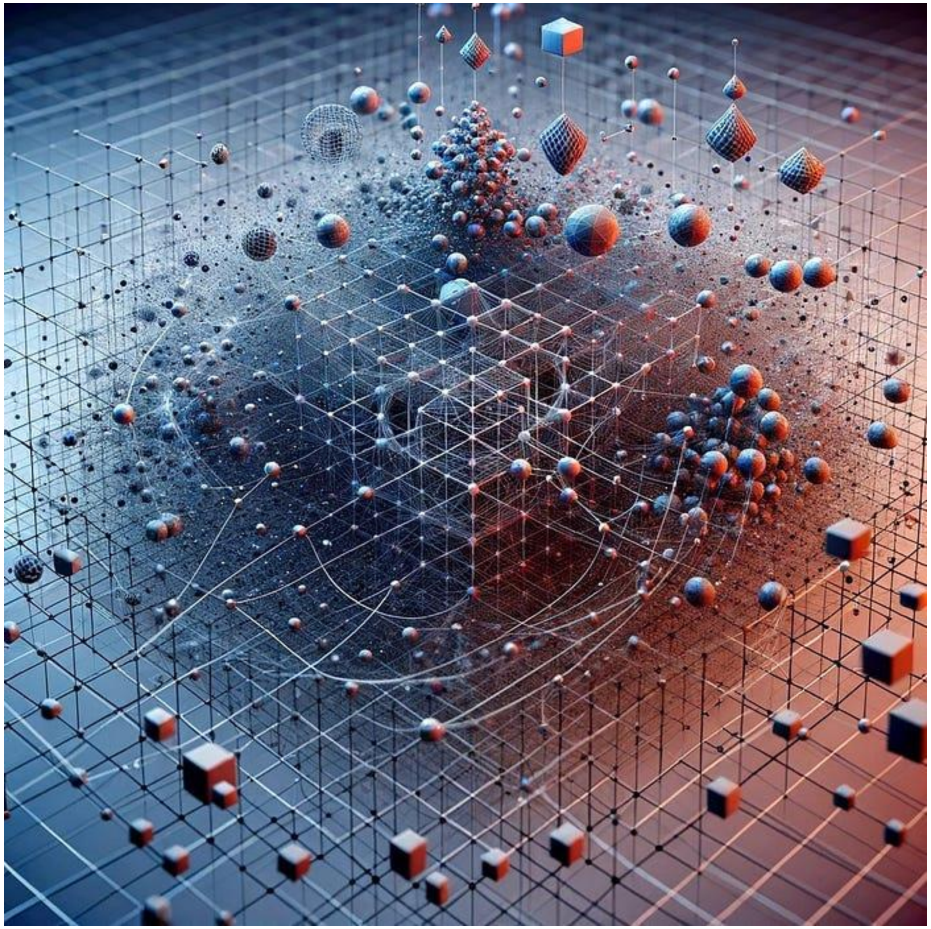
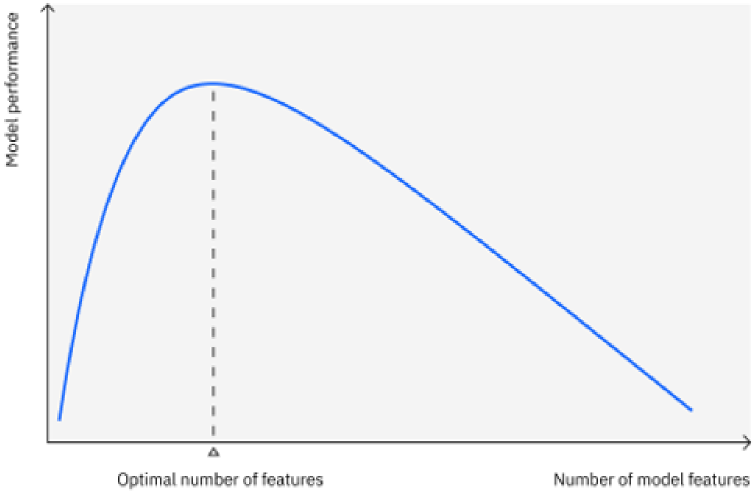


Image source: https://towardsdatascience.com/



-Computational efficiency: With fewer dimensions, algorithms can run faster and require less memory.

-Visualization:Dimensionality reduction techniques can help project data into lower-dimensional spaces that can be visualized more easily.

Increasing the number of features could lead to sparse data (if number of data points remain same) leading to dissimilar data points. This might result in overfitting as model learns noise in data. This reduces generalizability.

Principal Component Analysis (PCA)

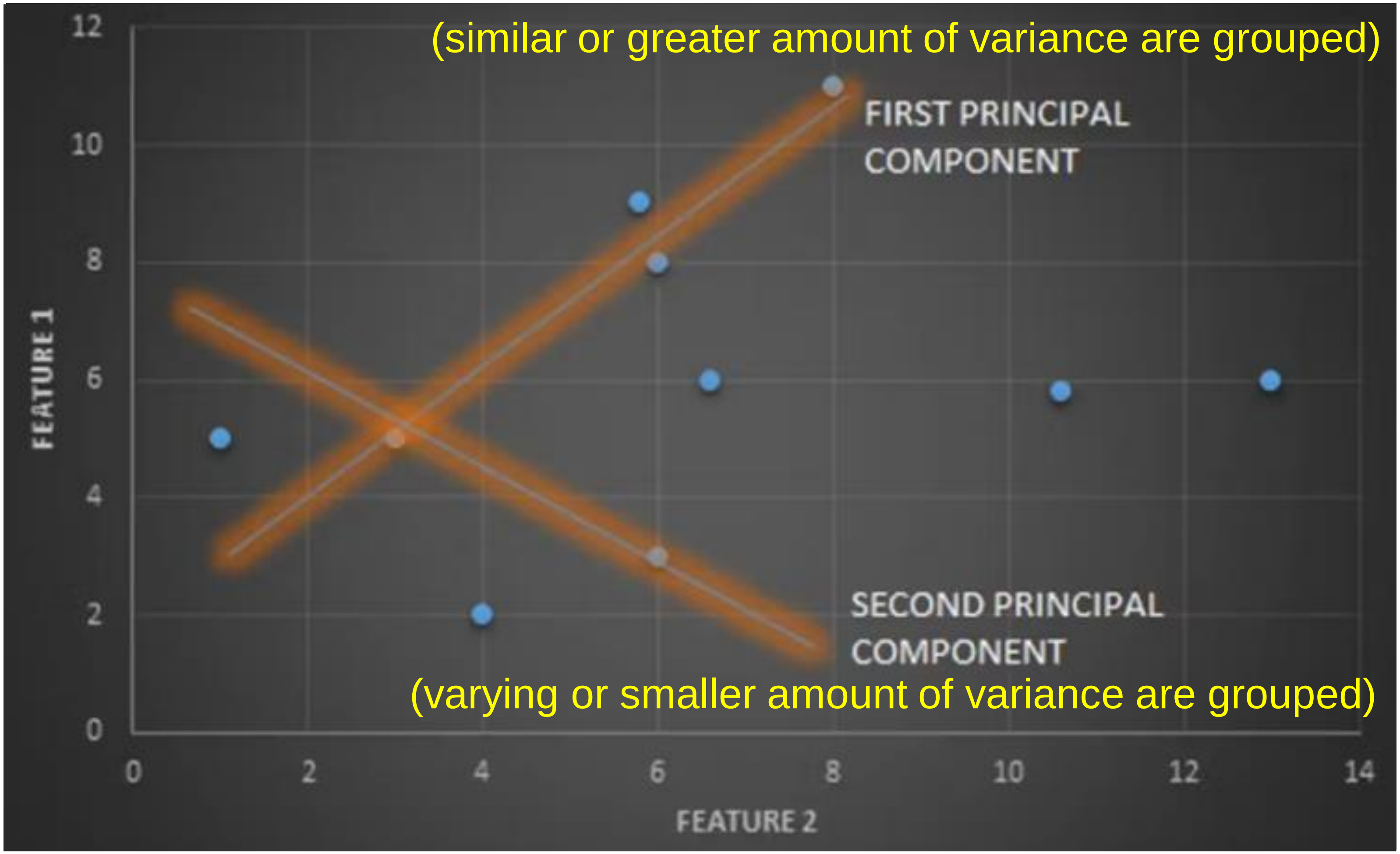
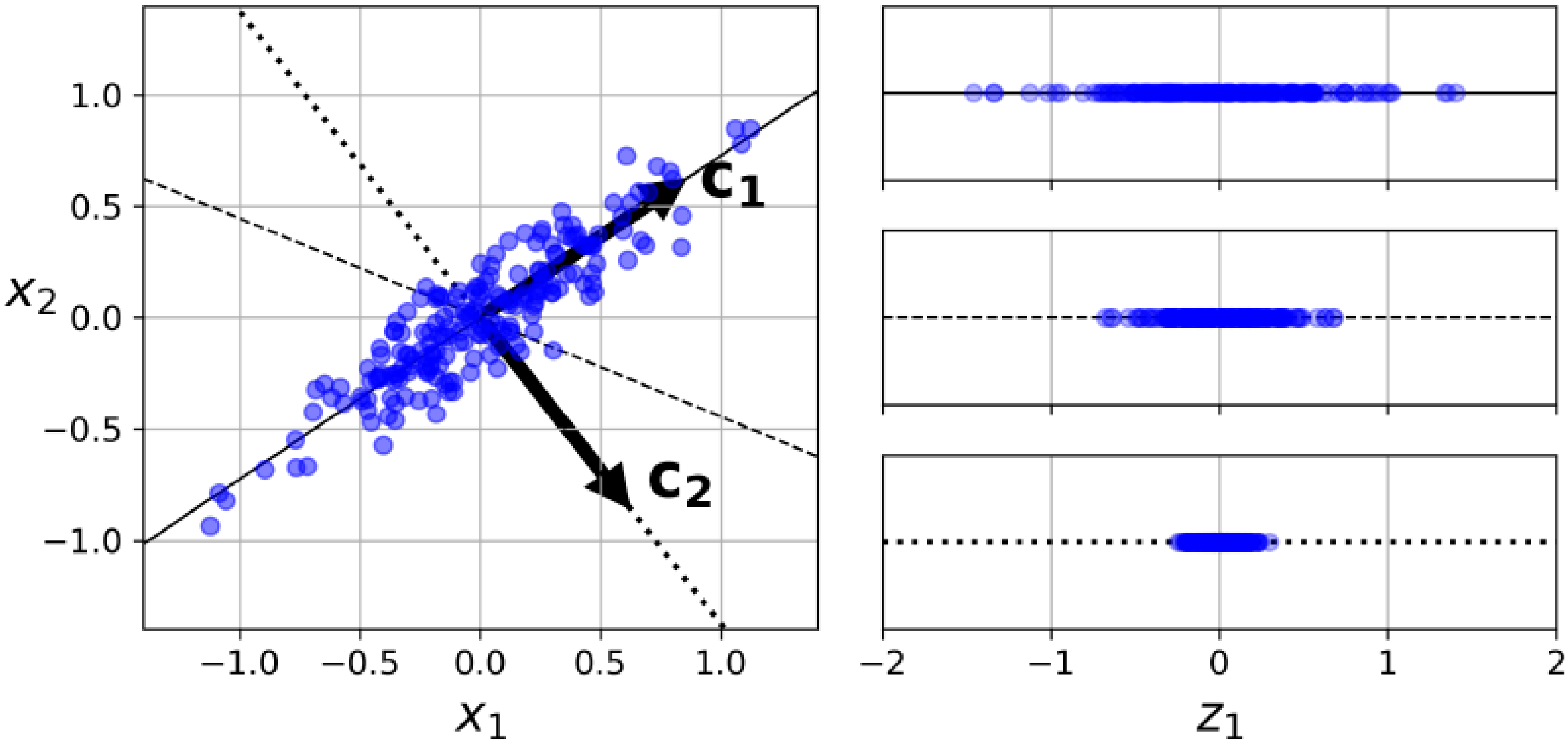


Image source: https://www.turing.com/

(Scatter plot: Data points distributed across the graph. Can you segregate them easily?)

Preserving the Variance: PCA Continued…



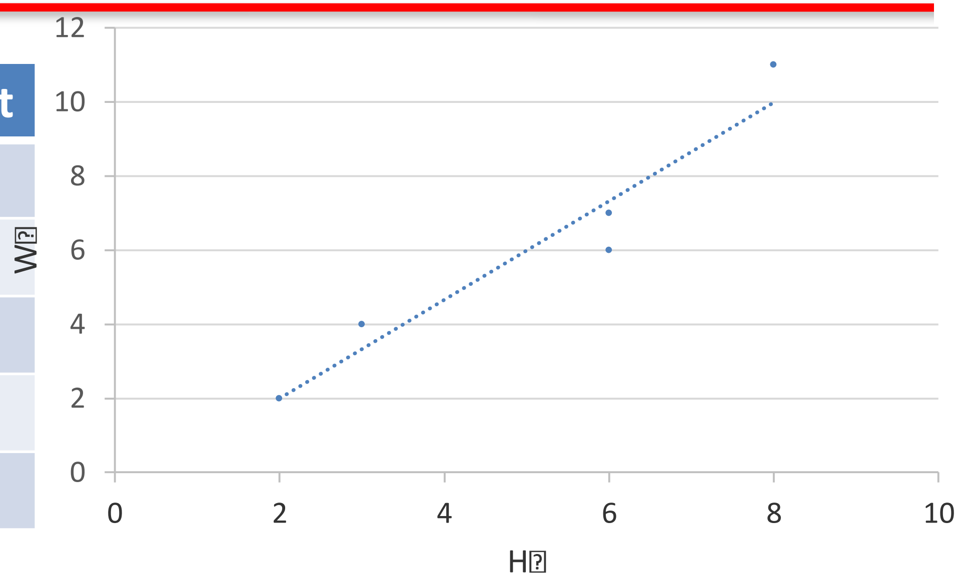
Which one is 1st PC and which one is 2nd PC?

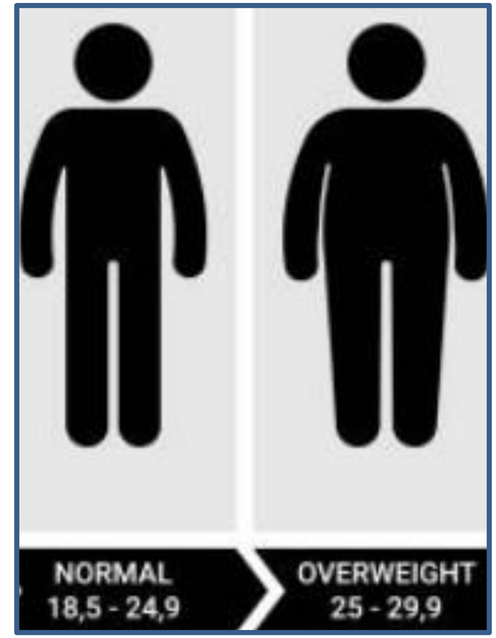
(Projection of dataset into there axes)

Image source: Aurelien Geron's text

Maths behind PCA

Height	Weight
2	2
3	4
6	6
6	7
8	11

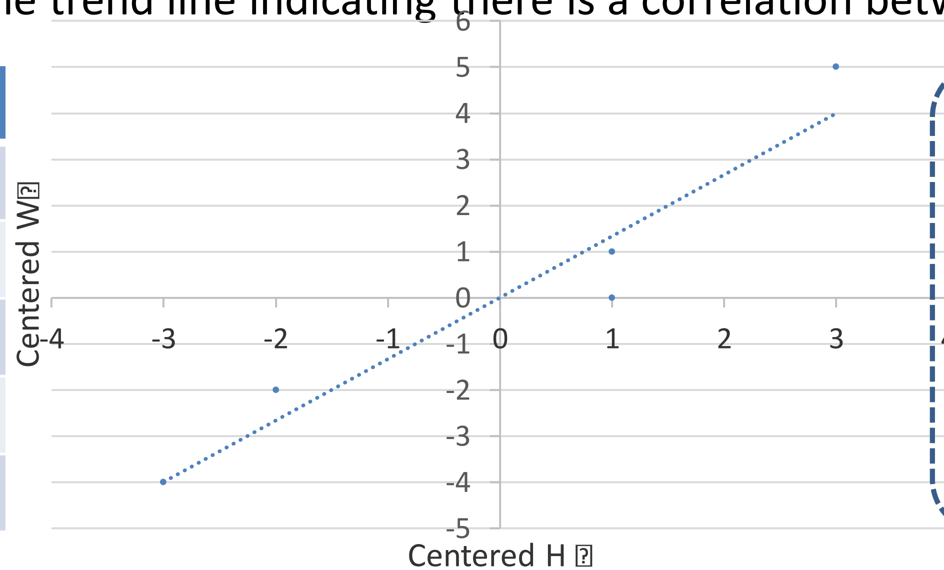




Body Mass Index (BMI)

- Scatter plot showing the trend line indicating there is a correlation between H and W.

Height	Weight
2-5=-3	2-6=-4
3-5=-2	4-6=-2
6-5=1	6-6=0
6-5=1	7-6=1
8-5=3	11-6=5



Centered data/ Standardized data tells us how far any original value is from the mean.

Continued...

- Next, we compute the Covariance matrix based on the centered/ standardized data.

  gives variance of each variable.

A

	H	W
H	6	8
W	8	11.5

Var(H) =(1/5-1) $\sum_{i=1}^{n}(x_i - \bar{x})^2$=($-3^2+(-2)^2+1^2+1^2+3^2$)/4=24/4=6

Var (W) = $((-4)^2 + (-2)^2 + (0)^2 + (1)^2 + (5)^2)/4$ = 46/4 = 11.5

- Now compute how much the two variables spread from each other i.e Cov(H, W)? Remember mean is already 0 for centered data.

  Cov (H, W) = (-3 X -4 + -2 X -2 + 1 X 0 + 1 X 1 + 3 X 5) / (5 - 1) = 32/4 = 8

- Next Compute the Eigen values: det | A – λI | = 0

$$\begin{pmatrix} 6 & 8 \\ 8 & 11.5 \end{pmatrix} - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} = 0$$

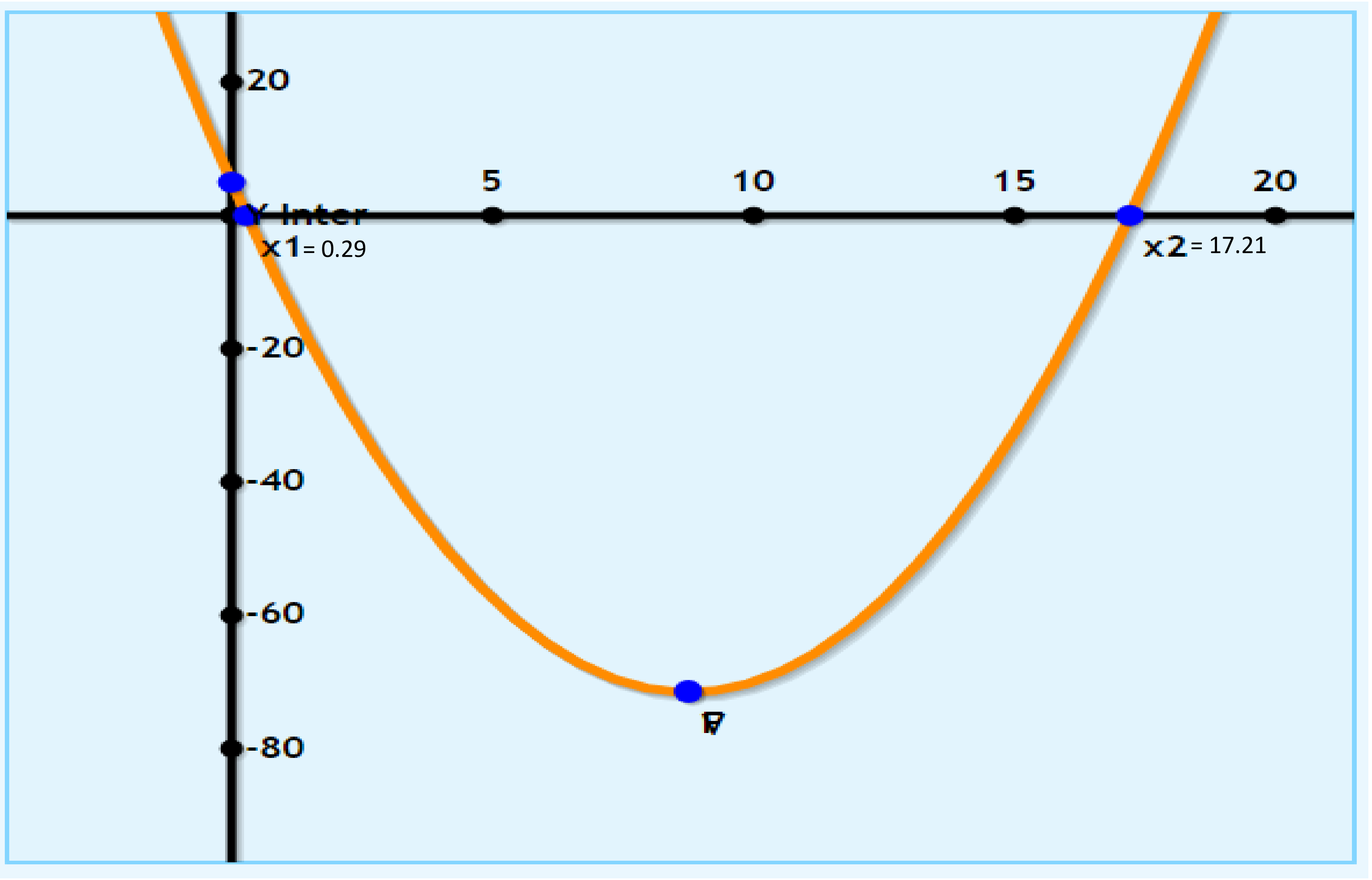
$(6 - \lambda) \times (11.5 - \lambda) - 8 \times 8 = 0$

$(\lambda^2) - 17.5\lambda + 5 = 0$

Eigen values

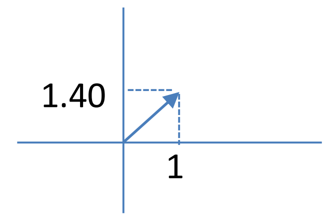
$\boxed{\lambda_1 = 17.21 \quad \lambda_2 = 0.29}$

Continued…



Continued…

- Next, find out the Eigen vectors to these two values.



- $A.v = \lambda.v$ $\quad \begin{pmatrix} 6 & 8 \\ 8 & 11.5 \end{pmatrix} . \begin{pmatrix} x \\ y \end{pmatrix} = 17.21 \begin{pmatrix} x \\ y \end{pmatrix}$

$$\begin{matrix} 6x + 8y = 17.21\,x \\ 8x + 11.5y = 17.21\,y \end{matrix} \;\rangle\; \left\{ \begin{matrix} 8y = 11.21\,x \\ 8x = 5.71\,y \end{matrix} \right. \;\rangle\; y = 1.40\,x \;\rangle\; \underset{v_1}{\begin{pmatrix} 1 \\ 1.40 \end{pmatrix}}$$

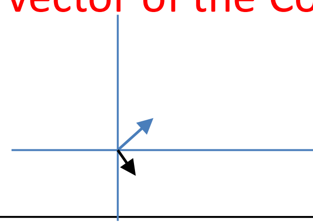
Eigen vector of Covariance matrix

- Now, normalize to unit length:

Length of vector = Sqrt $(1^2 + 1.40^2) = 1.72$ $\;\rangle\;$ $v_1 = \begin{pmatrix} 1/1.72 \\ 1.40/1.72 \end{pmatrix} = \begin{pmatrix} .5814 \\ .8139 \end{pmatrix}$

Similarly get the Eigen vector of the Covariance matrix for Eigen value 2:

$v_2 = \begin{pmatrix} .8139 \\ -.5811 \end{pmatrix}$



$\rangle$ $\begin{pmatrix} .5814 & .8139 \\ .8139 & -.5811 \end{pmatrix}$ Order the Eigen vectors

Continued…

- Now, calculate the Principal components:

Principal Components

$$\underbrace{\begin{pmatrix} -3 & -4 \\ -2 & -2 \\ 1 & 0 \\ 1 & 1 \\ 3 & 5 \end{pmatrix}}_{D} \cdot \underbrace{\begin{pmatrix} .5814 & .8139 \\ .8139 & -.5811 \end{pmatrix}}_{V} = \begin{pmatrix} -4.9998 & -.1173 \\ -2.7906 & -.4656 \\ .5814 & .8139 \\ 1.3953 & .2328 \\ 5.8137 & -.4638 \end{pmatrix}$$

Stores information about all variables

Does not store much.

- Why PC_2 does not store much info?

- How much % of total variance is contributed by PC_1?

17.21/17.21+.29
= 98.34%

Thank You!