ED5340 - Data Science: Theory and Practise

L21 - Principal Component Analysis

Ramanathan Muthuganapathy (https://ed.iitm.ac.in/~raman)
Course web page: https://ed.iitm.ac.in/~raman/datascience.html
Moodle page: Available at https://courses.iitm.ac.in/

Feature selection

To reduce the number of features

- Arbitrarily select features to reduce the size
- Easier to solve the problem
- Optimization is made faster

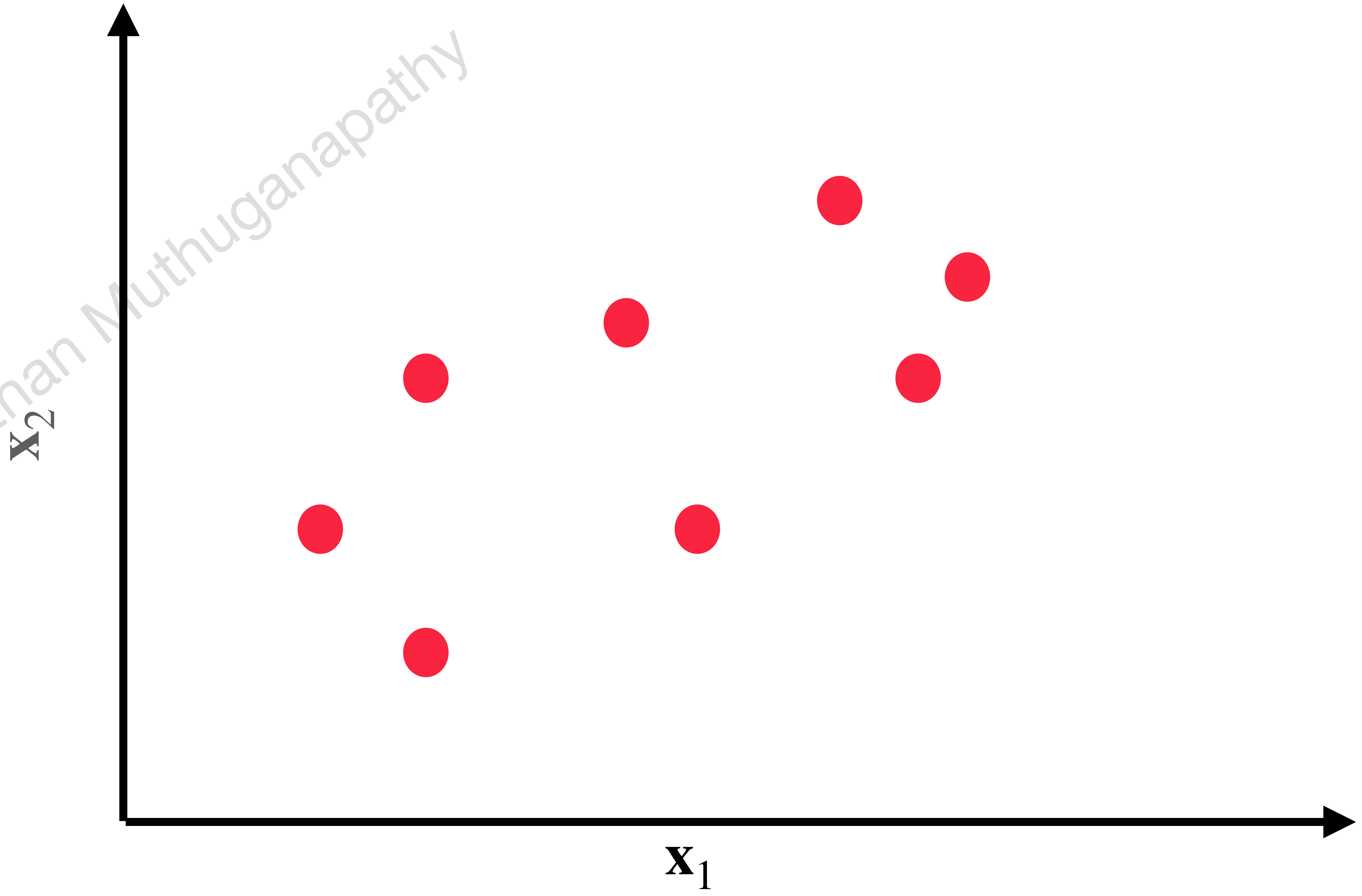
Dimensionality reduction

Typically projection-based

- Principal Component Analysis (PCA)
- Projection-based
- Uses typical vector calculus and linear algebra
- Easier to solve the problem
- computational efficient

Principal Component Analysis

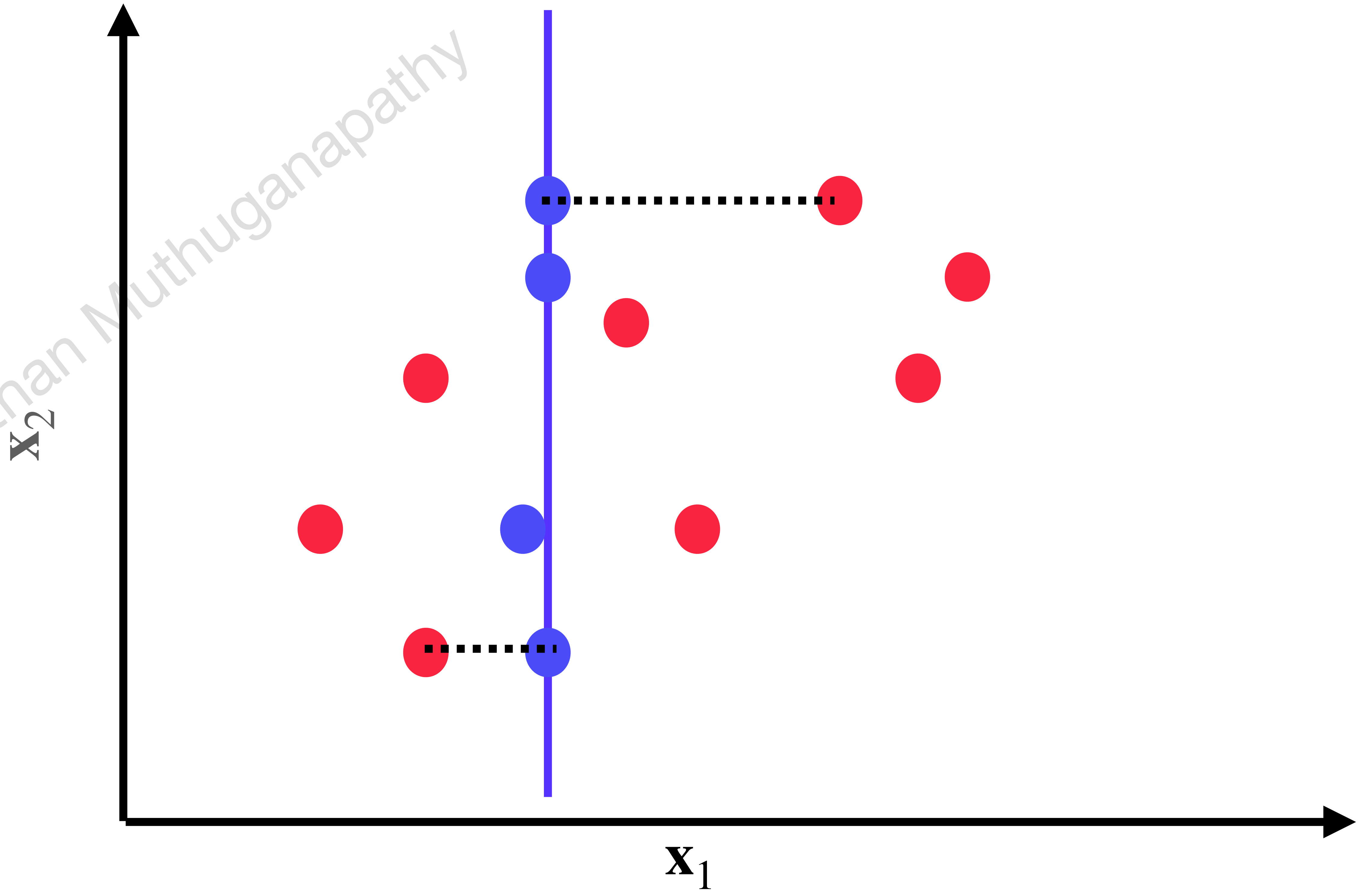
- Data is as shown



Principal Component Analysis

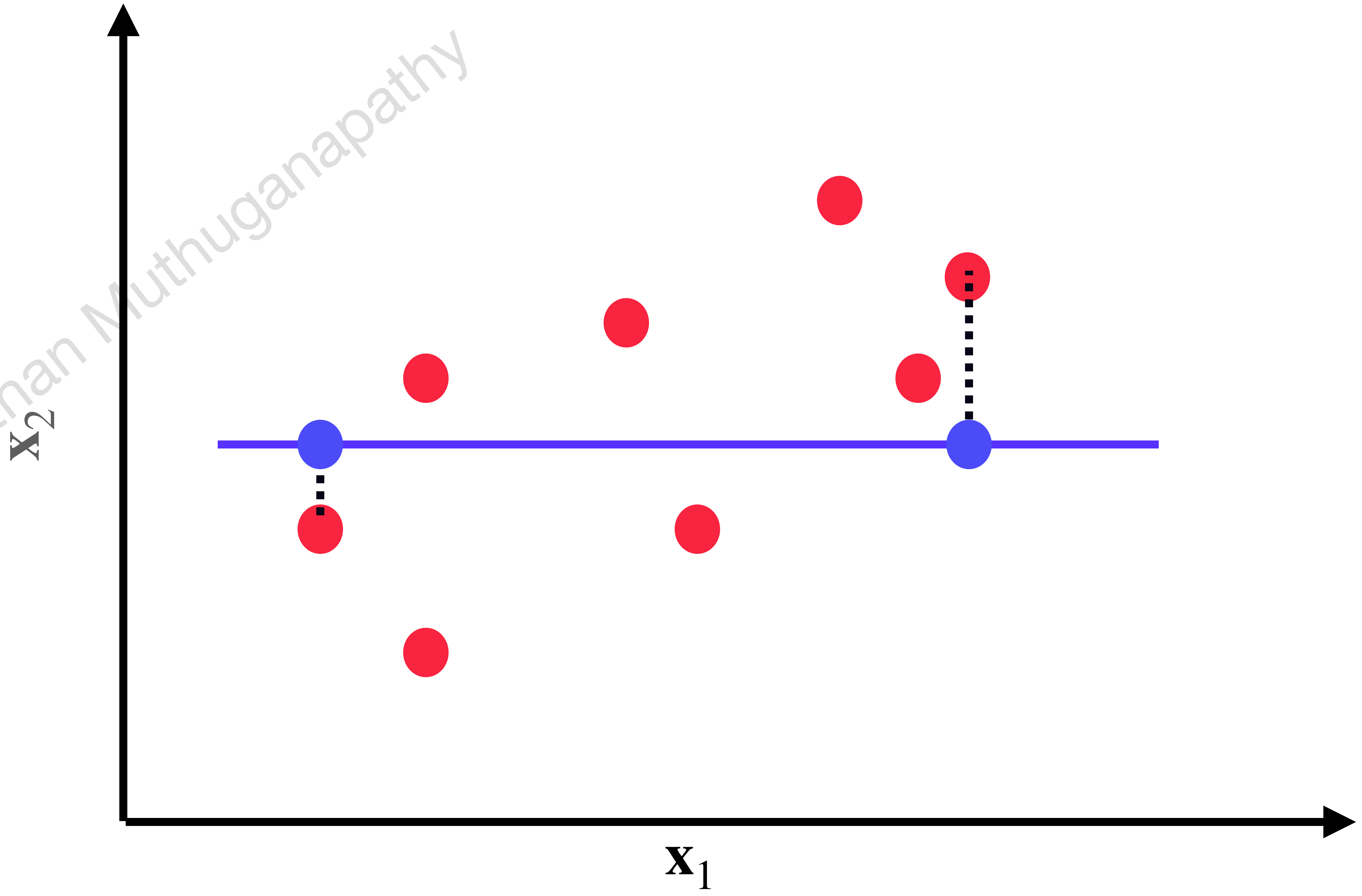
New axis

- Find a new axis
- Project on the new one



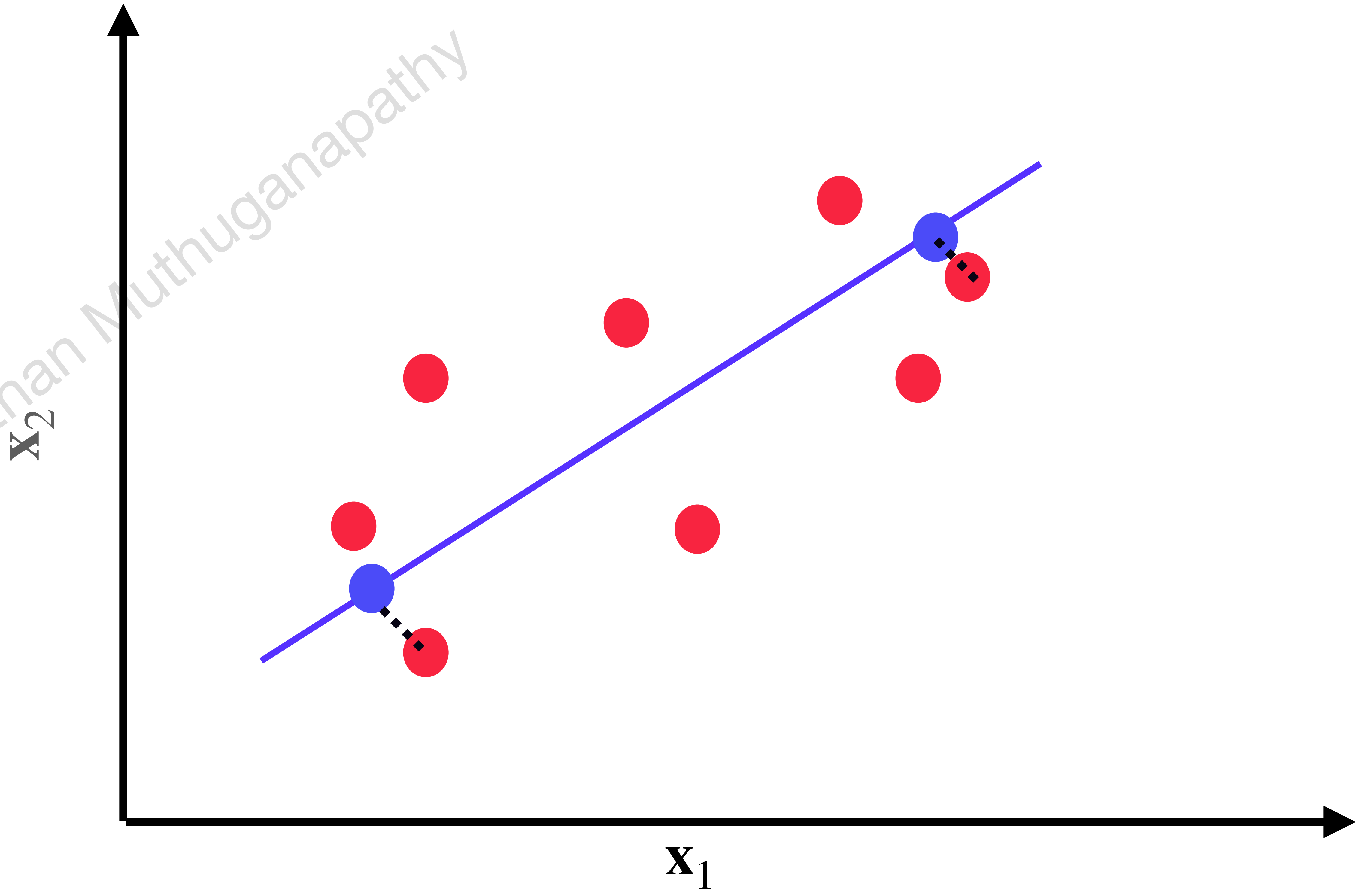
Principal Component Analysis

- Horizontal axis



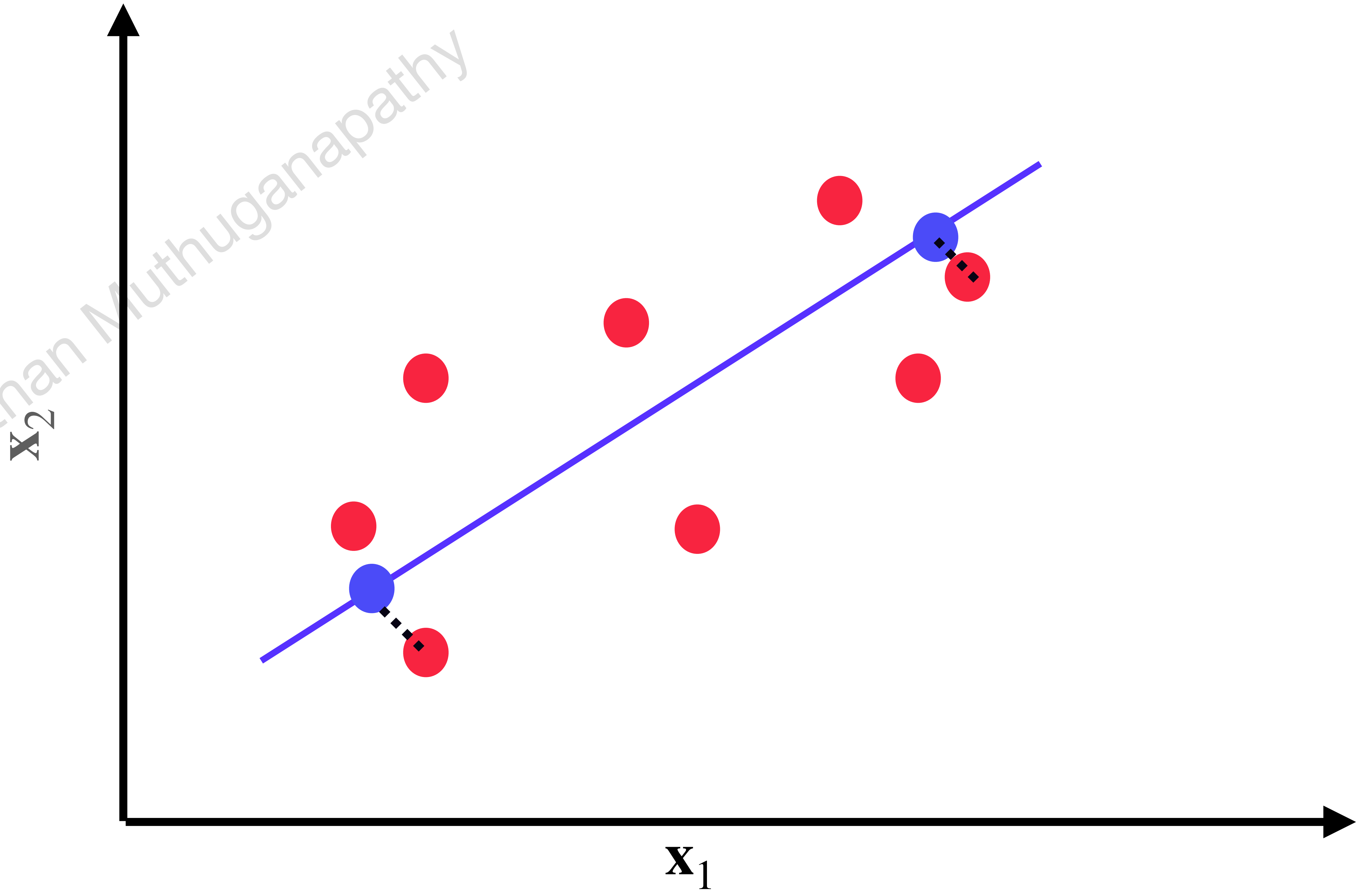
Principal Component Analysis

- Some axis (Principal axis!)
- What are key points?



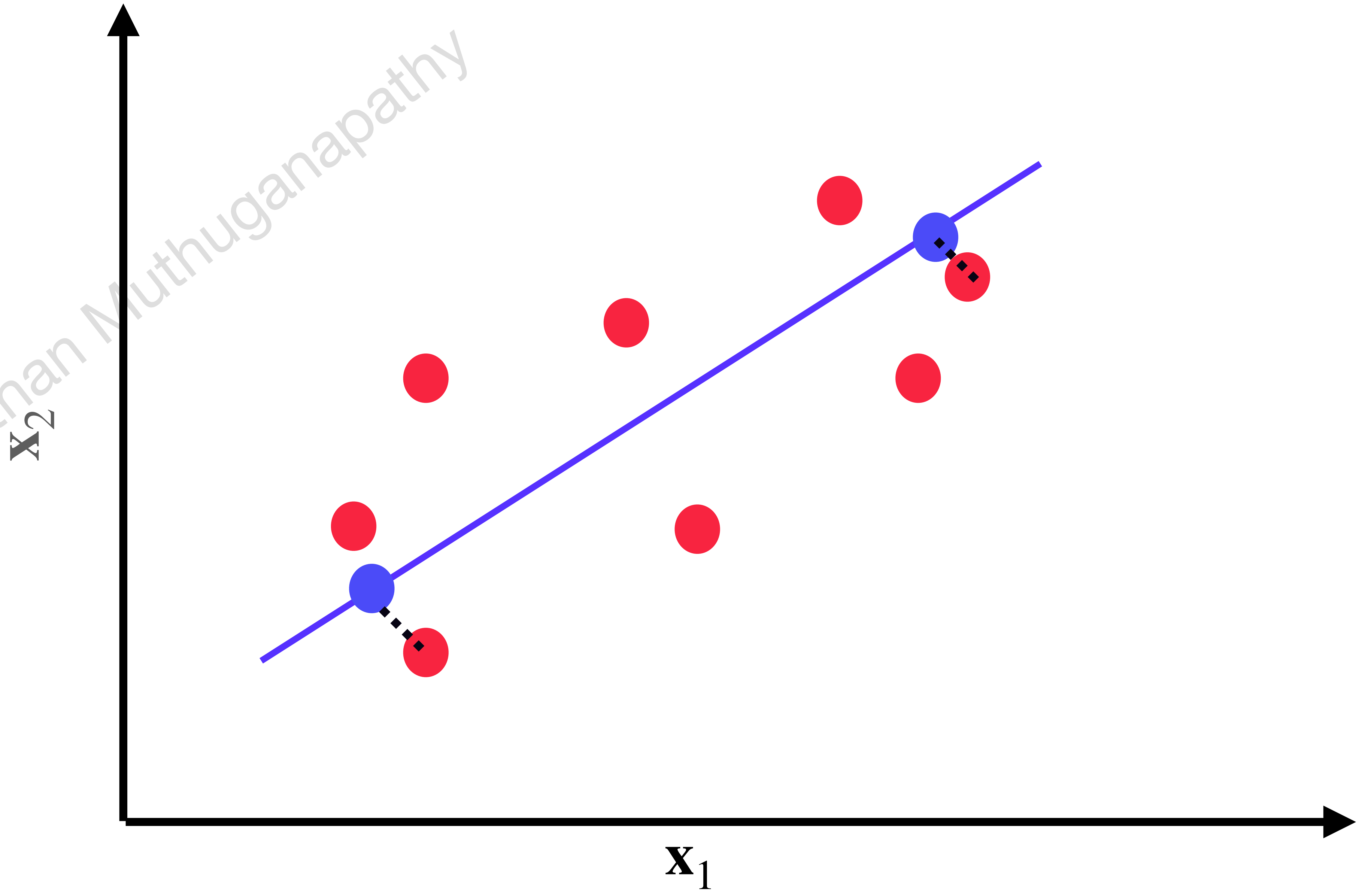
Principal Component Analysis

- Maximize the variance (retain the most information)
- Projection is perpendicular to the data (compare with linear regression!)
- Extracts the so-called new features



Principal Component Analysis

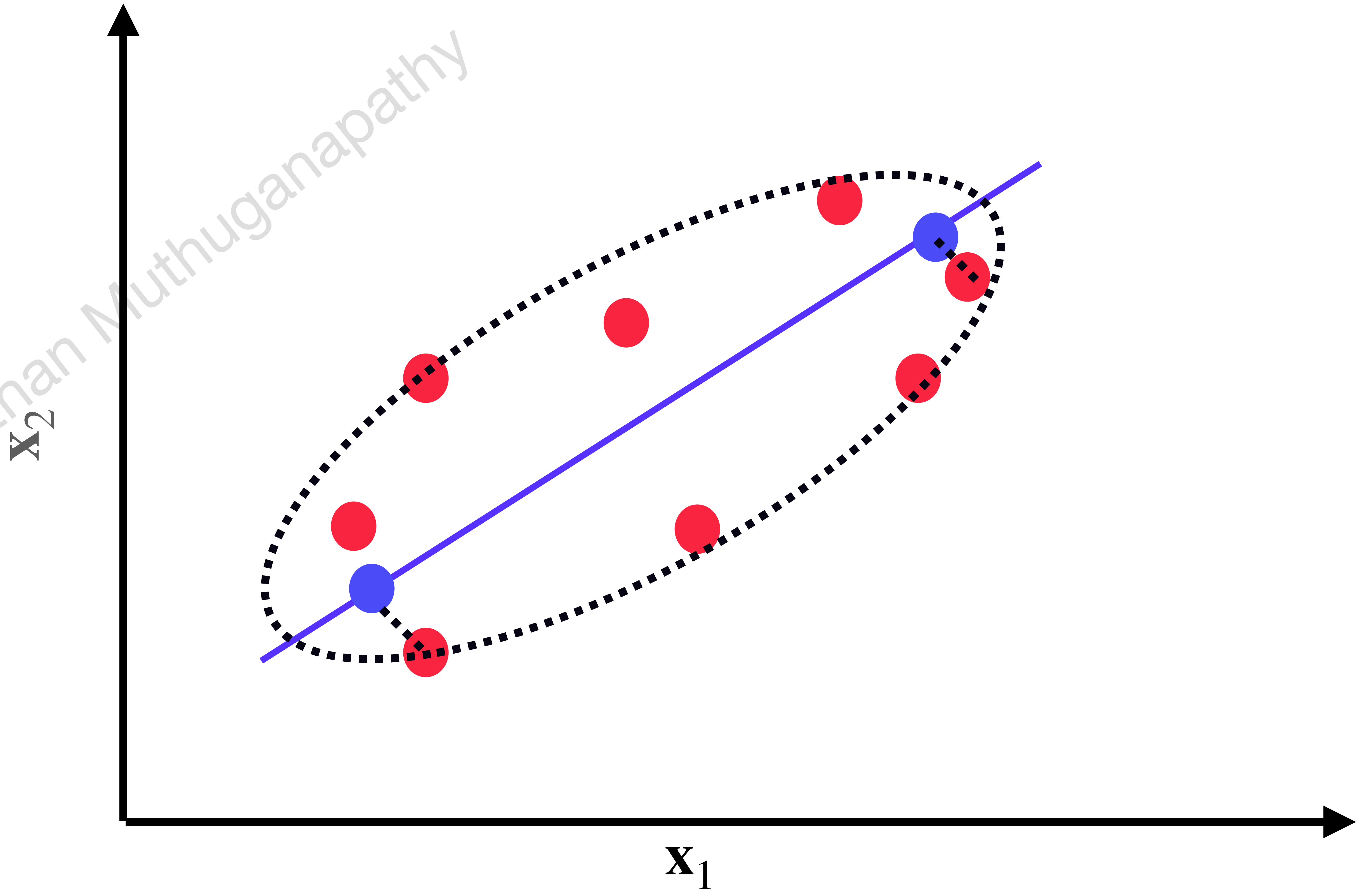
- How do we find this axis (axes)?
- Metric to use (we talked about variance)



Principal Component Analysis

Geometric intuition

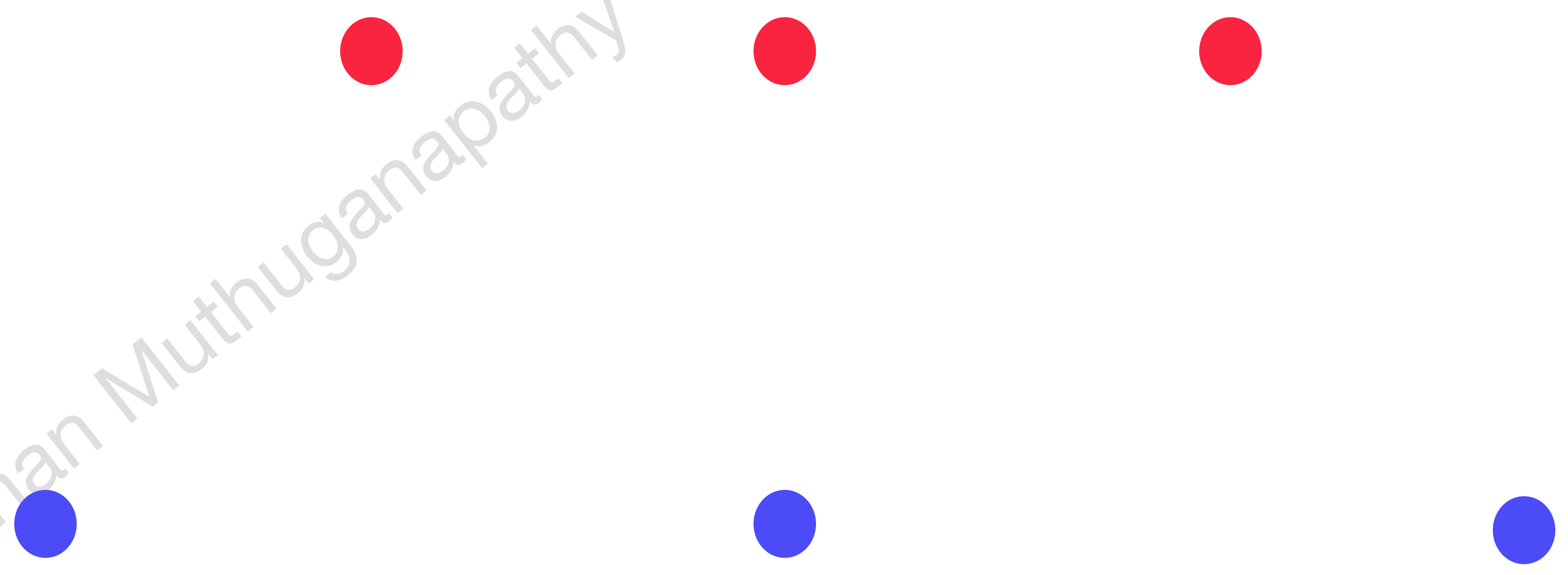
- How do we find this axis (axes)?
- Metric to use (we talked about variance)



Principal Component Analysis

Projection along variance

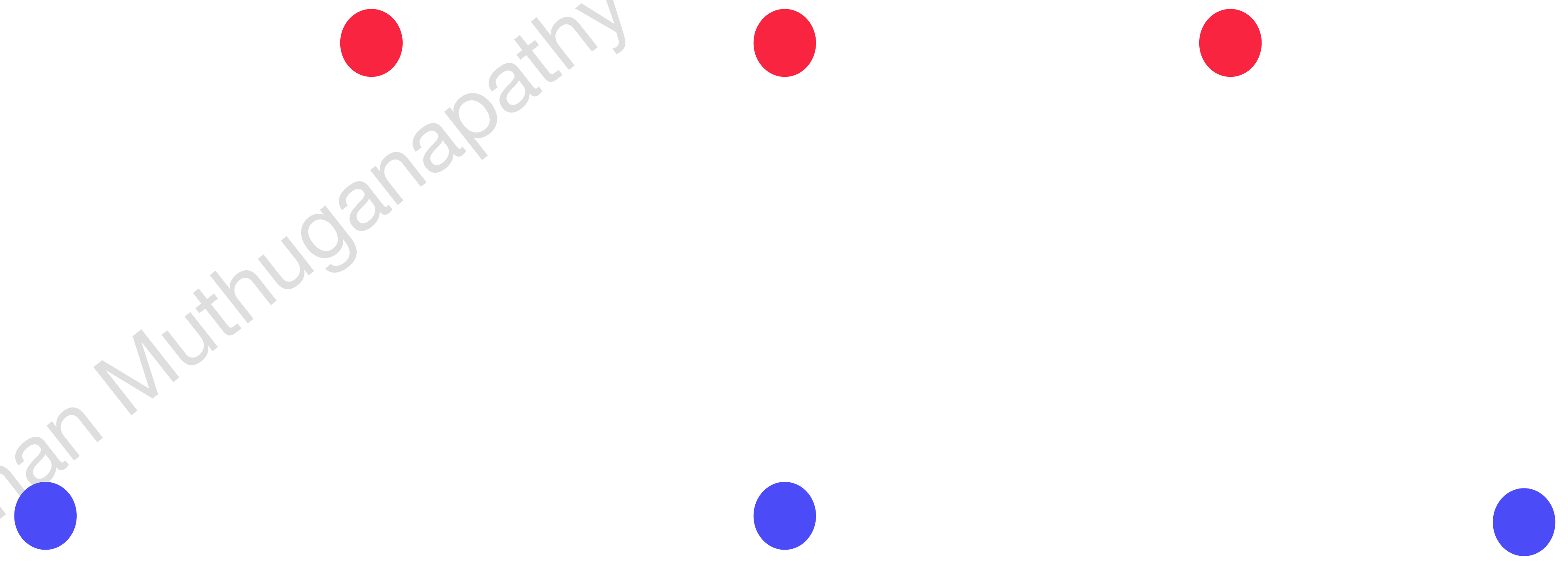
- Mean may not distinguish well enough (why)
- $v_1 = (1^2 + 0^2 + 1^2)/3 = 2/3$
- $v_1 = (2^2 + 0^2 + 2^2)/3 = 8/3$



Principal Component Analysis

Projection along mean

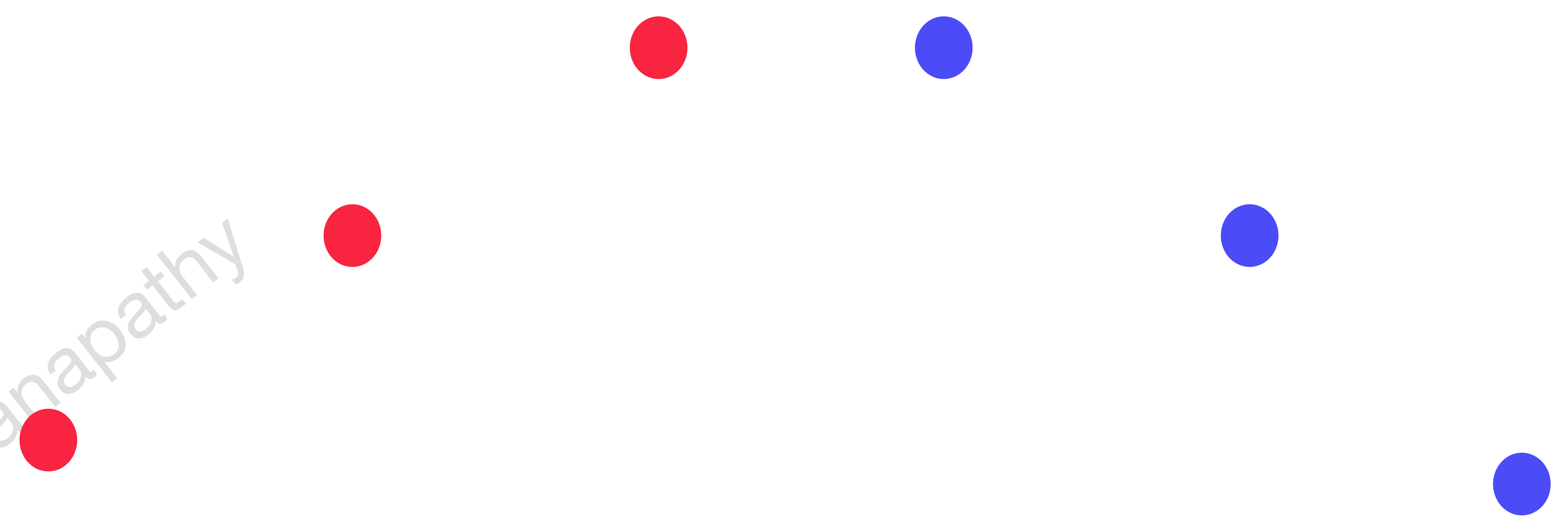
- Mean may not distinguish well enough



Principal Component Analysis

Projection along variance

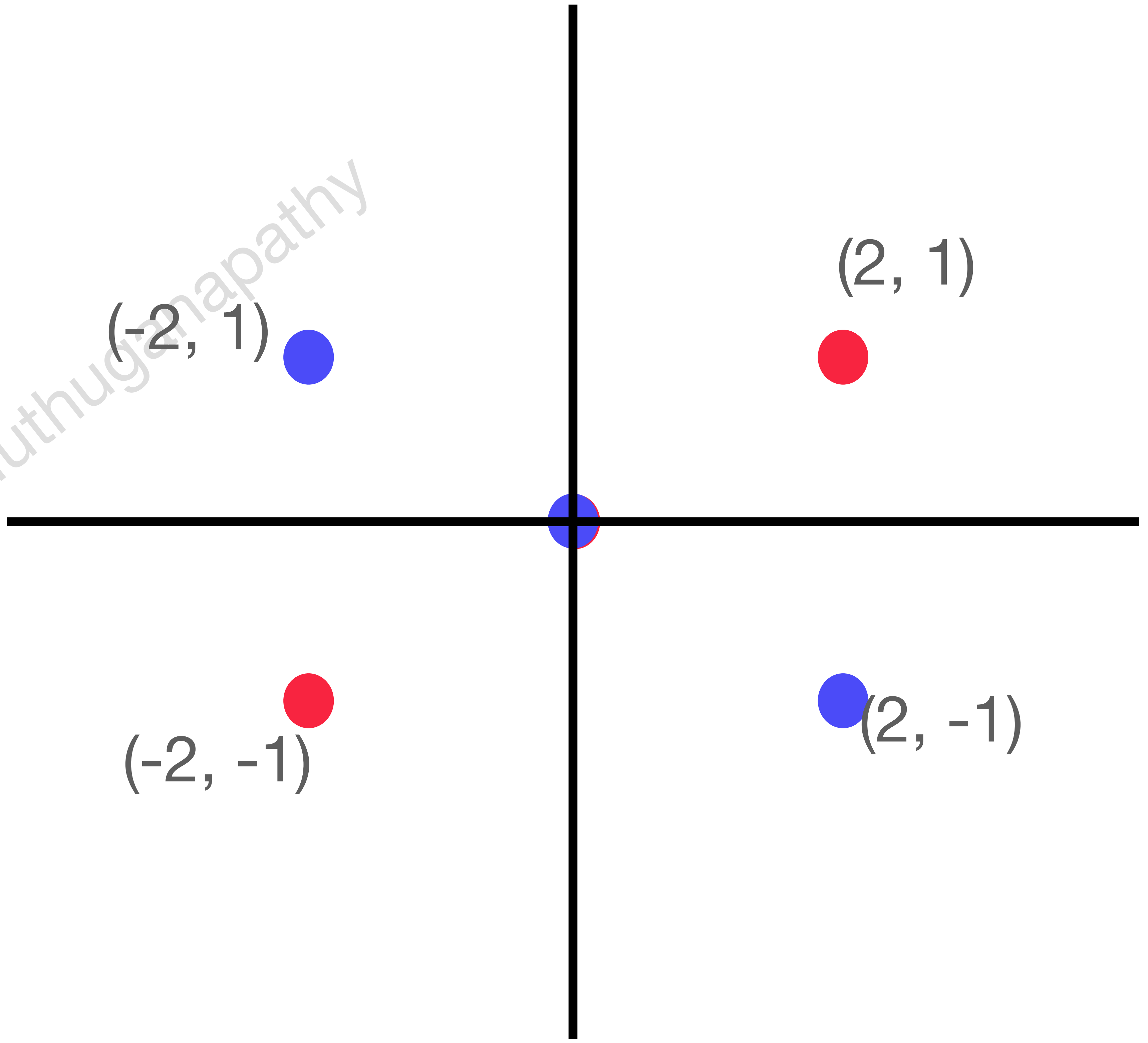
- x varies by 2 and y varies by 1
- $x_{v1} = (2^2 + 0^2 + 2^2)/3 = 8/3$
- $y_{v1} = (1^2 + 0^2 + 1^2)/3 = 2/3$
- Compute the x and y variance of the other data
- What do you say?



Principal Component Analysis

Superimposing both data

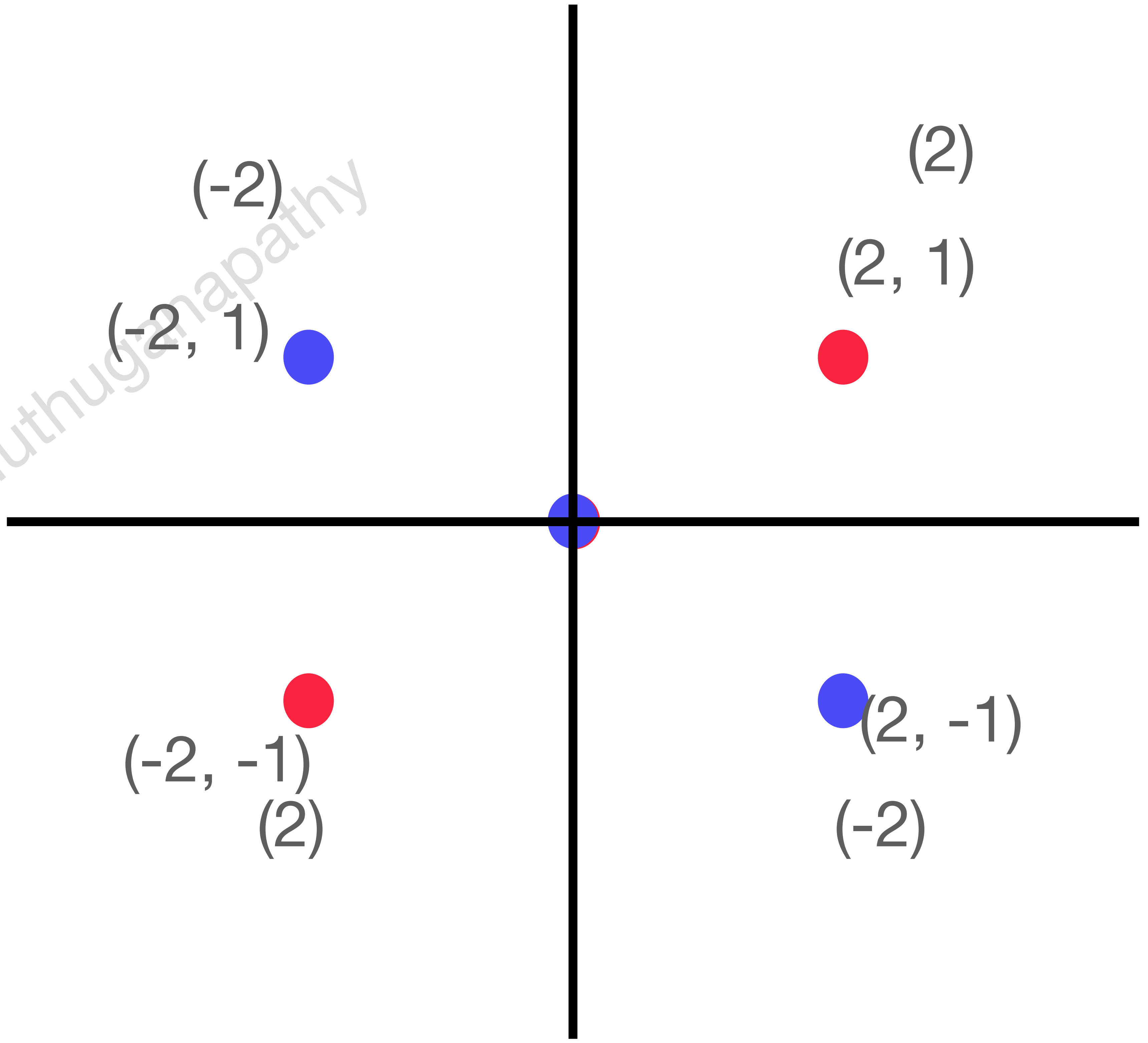
- x varies by 2 and y varies by 1
- Compute the product of the coordinates
- covariance



Principal Component Analysis

Superimposing both data

- x varies by 2 and y varies by 1
- Compute the product of the coordinates
- covariance (sum of the products / num)
- 4/3, -4/3



Formulating covariance matrix

$$\begin{bmatrix} var(x) & cov(x,y) \\ cov(y,x) & var(y) \end{bmatrix}$$

Formulating covariance matrix

For data 1

$$\begin{bmatrix} 8/3 & 4/3 \\ 4/3 & 8/3 \end{bmatrix}$$

Formulating covariance matrix

for data 2

$$\begin{bmatrix} 8/3 & -4/3 \\ -4/3 & 8/3 \end{bmatrix}$$

Formula - Covariance matrix

for data 2 - m samples and n features

- $$Cov(j,k) = \frac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)} - \mu^{(i)}\right)\left(x_k^{(i)} - \mu^{(i)}\right)$$

Formula - Covariance matrix

$$Cov(j,k) = \frac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)} - \mu^{(i)}\right)\left(x_k^{(i)} - \mu^{(i)}\right)$$

5 samples and 2 features

	sample number	Size $(x_1^{(i)})$	Type $(x_2^{(i)})$	Maintenance $(x_3^{(i)})$
$(x_1^{(1)}, x_2^{(1)}, x_3^{(1)})$	**1**	2	1	2
$(x_1^{(2)}, x_2^{(2)}, x_3^{(2)})$	**2**	4	2	2.5
$(x_1^{(3)}, x_2^{(3)}, x_3^{(3)})$	**3**	6	3	3
$(x_1^{(4)}, x_2^{(4)}, x_3^{(4)})$	**4**	8	4	3.5
$(x_1^{(5)}, x_2^{(5)}, x_3^{(5)})$	**5**	10	5	4
		$\mu^{(1)}$	$\mu^{(2)}$	$\mu^{(3)}$

Formula - Covariance matrix

$$Cov(j,k) = \frac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)} - \mu^{(i)}\right)\left(x_k^{(i)} - \mu^{(i)}\right)$$

5 samples and 2 features

	sample number	Size $(x_1^{(i)})$	Type $(x_2^{(i)})$
$(x_1^{(1)}, x_2^{(1)})$	**1**	10	1
$(x_1^{(2)}, x_2^{(2)})$	**2**	20	2
$(x_1^{(3)}, x_2^{(3)})$	**3**	30	3
$(x_1^{(4)}, x_2^{(4)})$	**4**	40	4
$(x_1^{(5)}, x_2^{(5)})$	**5**	50	5
		$\mu^{(1)}$	$\mu^{(2)}$

Formula - Covariance matrix

$$Cov(j,k) = \frac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)} - \mu^{(i)}\right)\left(x_k^{(i)} - \mu^{(i)}\right)$$

5 samples and 2 features

$\mu^{(1)} = 30$

$\mu^{(2)} = 3$

	sample number	Size $(x_1^{(i)})$	Type $(x_2^{(i)})$
	1	10	1
	2	20	2
	3	30	3
	4	40	4
	5	50	5
		$\mu^{(1)}$	$\mu^{(2)}$

Formula - Covariance matrix

$$Cov(j,k) = \frac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)} - \mu^{(i)}\right)\left(x_k^{(i)} - \mu^{(i)}\right)$$

5 samples and 2 features

$\mu^{(1)} = 30$

$\mu^{(2)} = 3$

	sample number	Size $(x_1^{(i)})$	Type $(x_2^{(i)})$
	1	-20	-2
	2	-10	-1
	3	0	0
	4	10	1
	5	20	2
		$\mu^{(1)}$	$\mu^{(2)}$

Formula - Covariance matrix

$$Cov(j,k) = \frac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)}\right)\left(x_k^{(i)}\right)$$

5 samples and 2 features

	sample number	Size	Type
	1	-20	-2
	2	-10	-1
	3	0	0
	4	10	1
	5	20	2
		$\mu^{(1)}$	$\mu^{(2)}$

X matrix

$$X = \begin{bmatrix} 10 & 1 \\ 20 & 2 \\ 30 & 3 \\ 40 & 4 \\ 50 & 5 \end{bmatrix}_{mXn}$$

X - mean

$$\mathbf{X} = \begin{bmatrix} -20 & -2 \\ -10 & -1 \\ 0 & 0 \\ 10 & 1 \\ 20 & 2 \end{bmatrix}_{mXn}$$

Transpose of X

$$\mathbf{X}^T = \begin{bmatrix} -20 & -10 & 0 & 10 & 20 \\ -2 & -1 & 0 & 1 & 2 \end{bmatrix}_{nXm}$$

Covariance matrix computation

$$Cov(j,k) = \frac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)}\right)\left(x_k^{(i)}\right)$$

$$\frac{1}{m}\mathbf{X}^T X = \frac{1}{5}\begin{bmatrix} -20 & -10 & 0 & 10 & 20 \\ -2 & -1 & 0 & 1 & 2 \end{bmatrix}_{nXm} \begin{bmatrix} -20 & -2 \\ -10 & -1 \\ 0 & 0 \\ 10 & 1 \\ 20 & 2 \end{bmatrix}_{mXn}$$

Covariance matrix computation

$$Cov(j,k) = \frac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)}\right)\left(x_k^{(i)}\right)$$

$$\frac{1}{m}\mathbf{X}^T X = \quad \frac{1}{5}\begin{bmatrix} 1000 & 100 \\ 100 & 10 \end{bmatrix}$$

Covariance matrix

For the given data

$$\frac{1}{m}\mathbf{X}^T X = \begin{bmatrix} 200 & 20 \\ 20 & 2 \end{bmatrix}$$

Properties

- Real symmetric matrix
  - Eigenvalues are ………
- Eigen decomposition or Singular value decomposition (SVD)

Eigen Decomposition

- Eigen decomposition $A = U D V^{-1}$
  - Eigen decomposition $A = U D U^{-1} = U D U^T$
    - When $U^{-1} = U^T$, the matrix is called ………… (example?)
  - U is the matrix of Eigen vector
  - D is a diagonal matrix of Eigen values

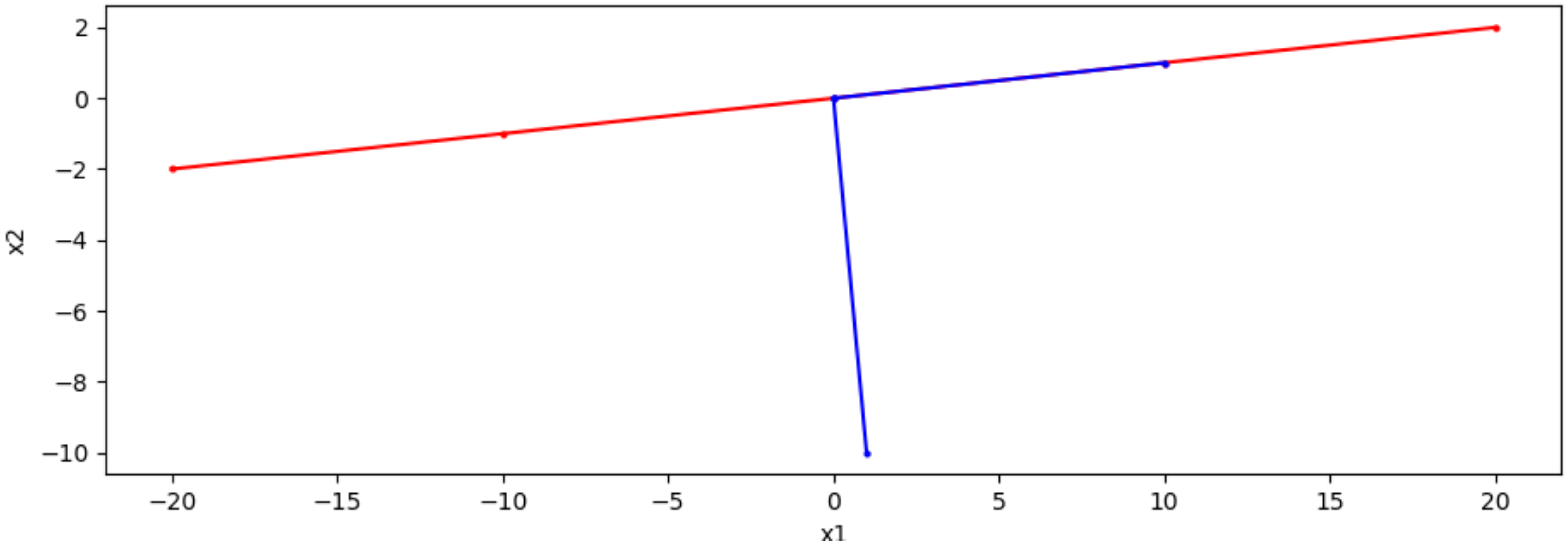
Find out details of SVD

Eigen values and vectors

$$\begin{vmatrix} 200 - \lambda & 20 \\ 20 & 2 - \lambda \end{vmatrix} = 0$$

PCA_plot.py

- Eigen values are (202, 0)
- Eigen vectors are [10, 1] and [1, -10]



$$\begin{bmatrix} 200 & 20 \\ 20 & 2 \end{bmatrix} \begin{bmatrix} 10 \\ 1 \end{bmatrix} = \begin{bmatrix} 2020 \\ 202 \end{bmatrix} = 202 \begin{bmatrix} 10 \\ 1 \end{bmatrix}$$

Projecting the data

PCA_plot.py

$$\begin{bmatrix} -20 & -2 \\ -10 & -1 \\ 0 & 0 \\ 10 & 1 \\ 20 & 2 \end{bmatrix}_{mXn} \begin{bmatrix} 10 \\ 1 \end{bmatrix}_{nX1} = \begin{bmatrix} -202 \\ -101 \\ 0 \\ 101 \\ 202 \end{bmatrix}_{mXn}$$

Overall procedure

PCA - m samples, n features - pca_in_depth.py

- Arrange each feature data as columns (or each sample as rows) - $\mathbf{X}_{mXn}$ matrix
- Subtract from the mean of each feature (columns). $\mathbf{X} = \mathbf{X} - \mu$
- Compute $\mathbf{P}_{nXn} = \dfrac{1}{m}\mathbf{X}^T\mathbf{X}$
- Perform Eigen decomposition or SVD of $\mathbf{P}_{nXn}$ (or compute Eigen values and Vectors)
- E. D. $\mathbf{P}_{nXn} = UDU^T$, U is a matrix of Eigen vectors (Column-wise)
- Take k Eigen vectors, i.e. U_k
- Compute the projection $\mathbf{X}U_k$