# Lecture 10 – Non-negative Matrix Factorization & Principal Component Analysis

Dr. Qingquan Sun

School of Computer Science and Engineering
Cal State University San Bernardino

# NMF & PCA

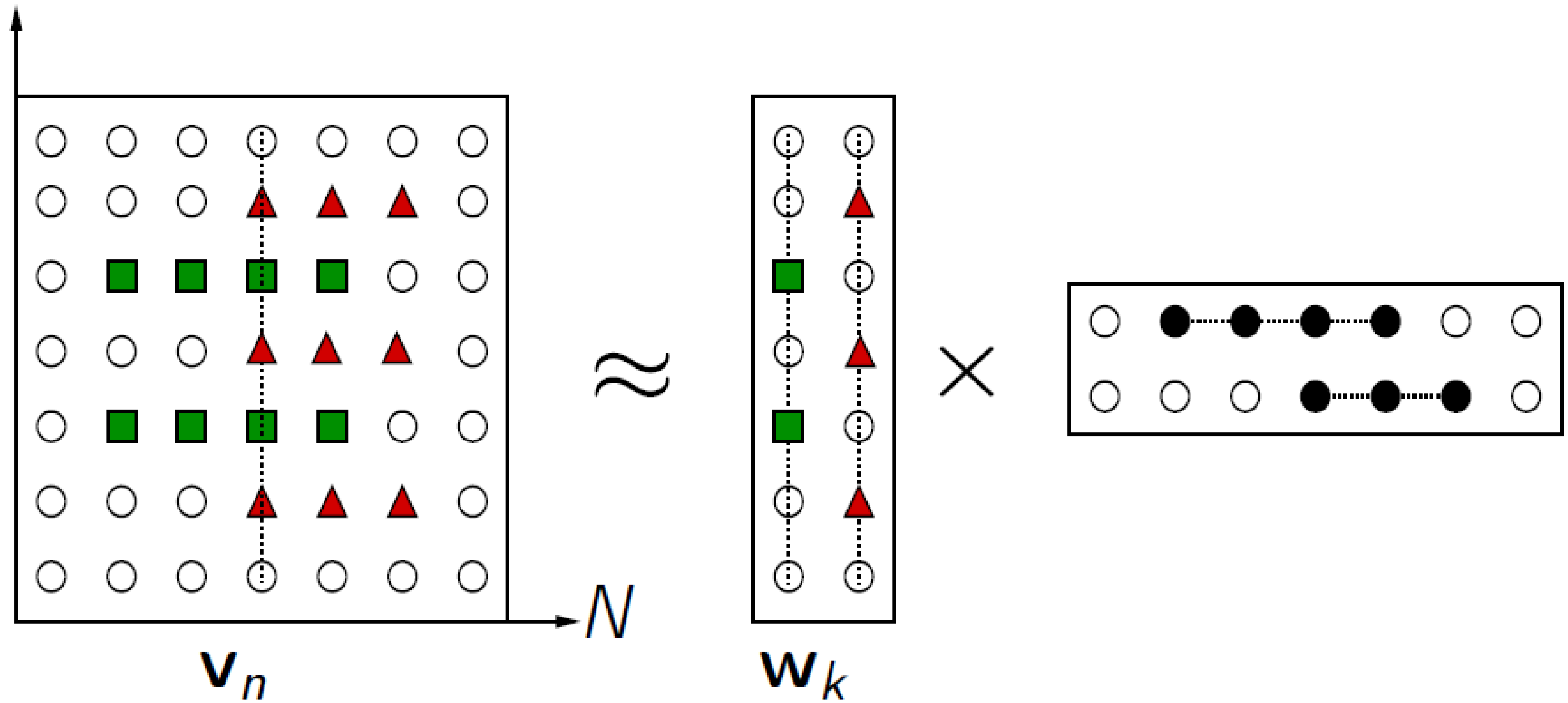
- Least square problems
- Factorization problems
- Eigenvalue and eigenvector
- Numerical computing methods

# Non-negative Matrix Factorization

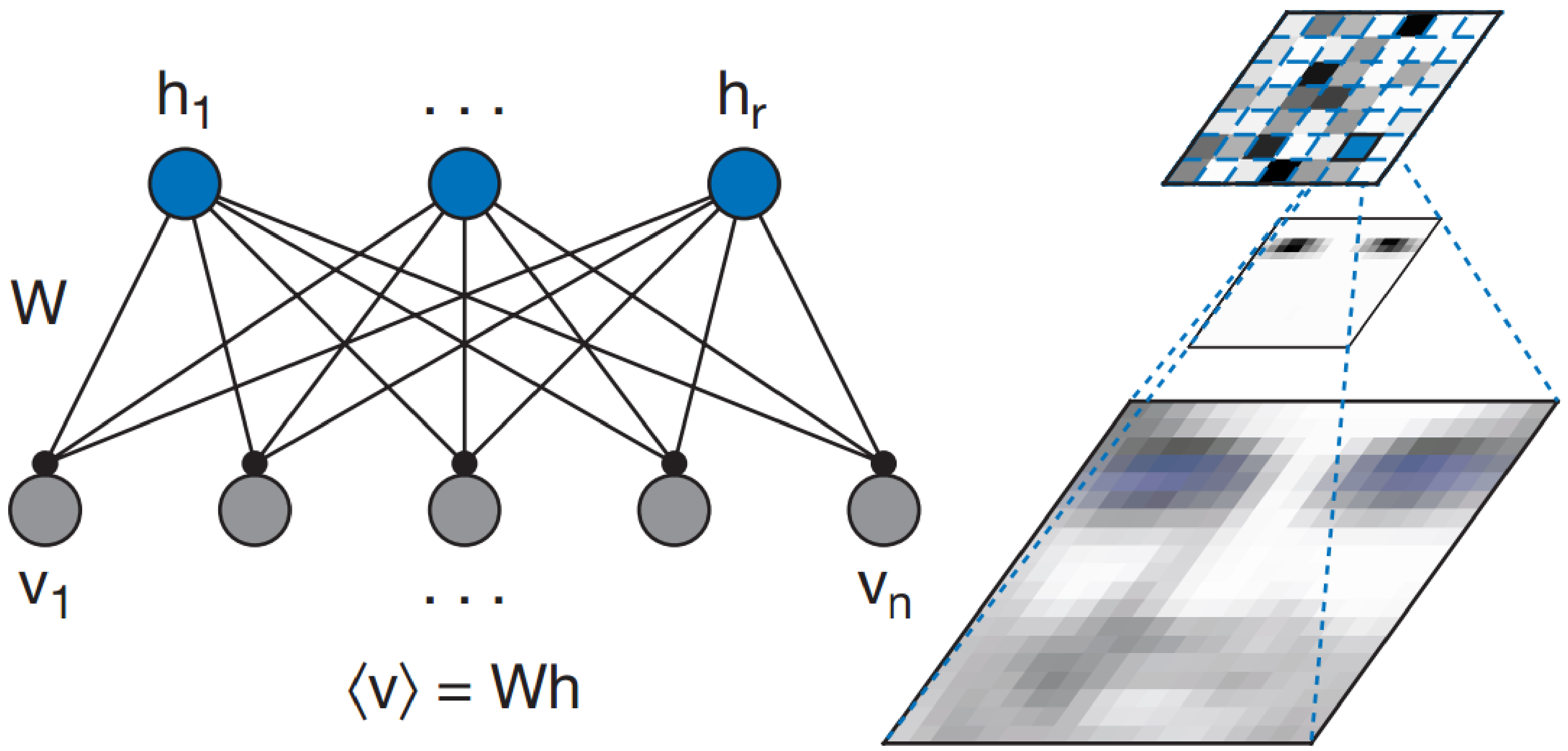
$$\mathbf{V}_{(F \times N)} \approx \mathbf{W}_{(F \times K)} \times \mathbf{H}_{(K \times N)}$$

# Non-negative Matrix Factorization

$$\mathbf{V}_{(F\times N)} \approx \mathbf{W}_{(F\times K)} \times \mathbf{H}_{(K\times N)}$$



# Non-negative Matrix Factorization



# Non-negative Matrix Factorization

- Properties
  - Part-based
  - Intuitive
  - Interpretable

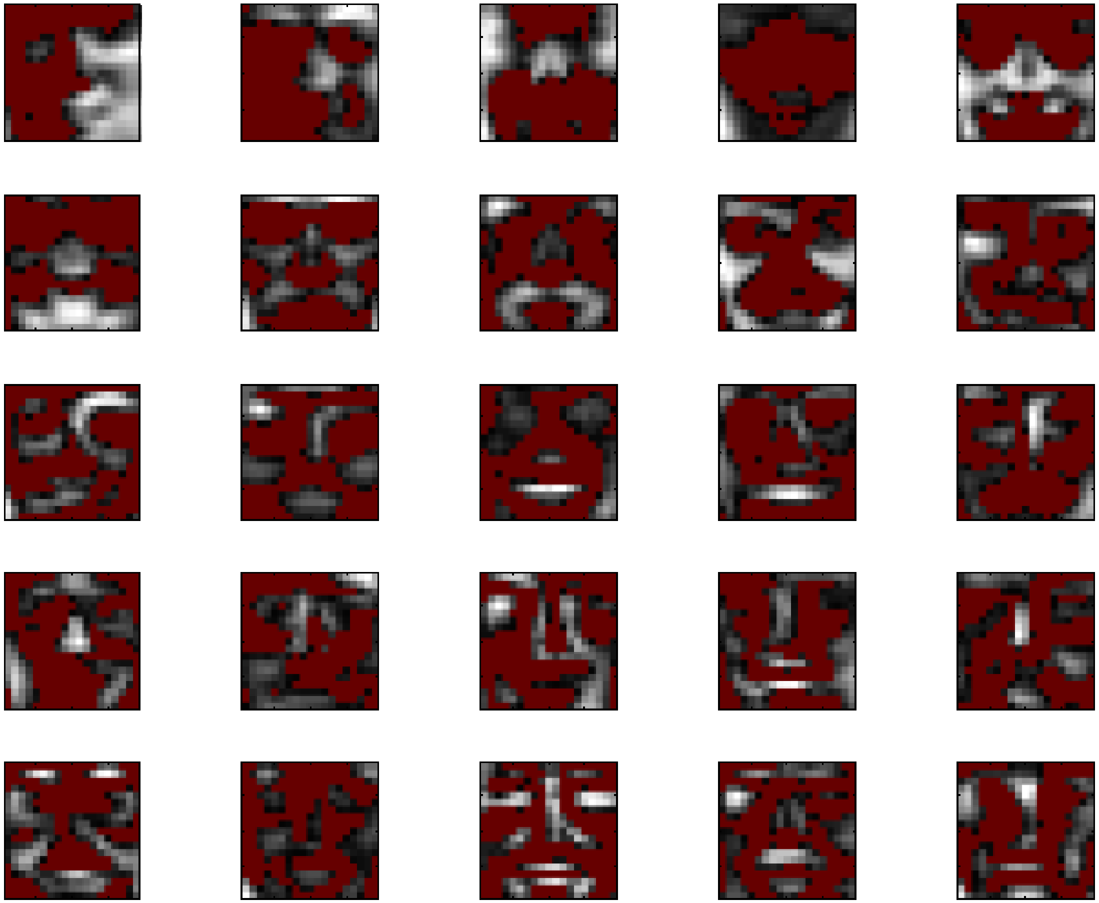
# Non-negative Matrix Factorization

- Properties
  - Part-based
  - Intuitive
  - Interpretable

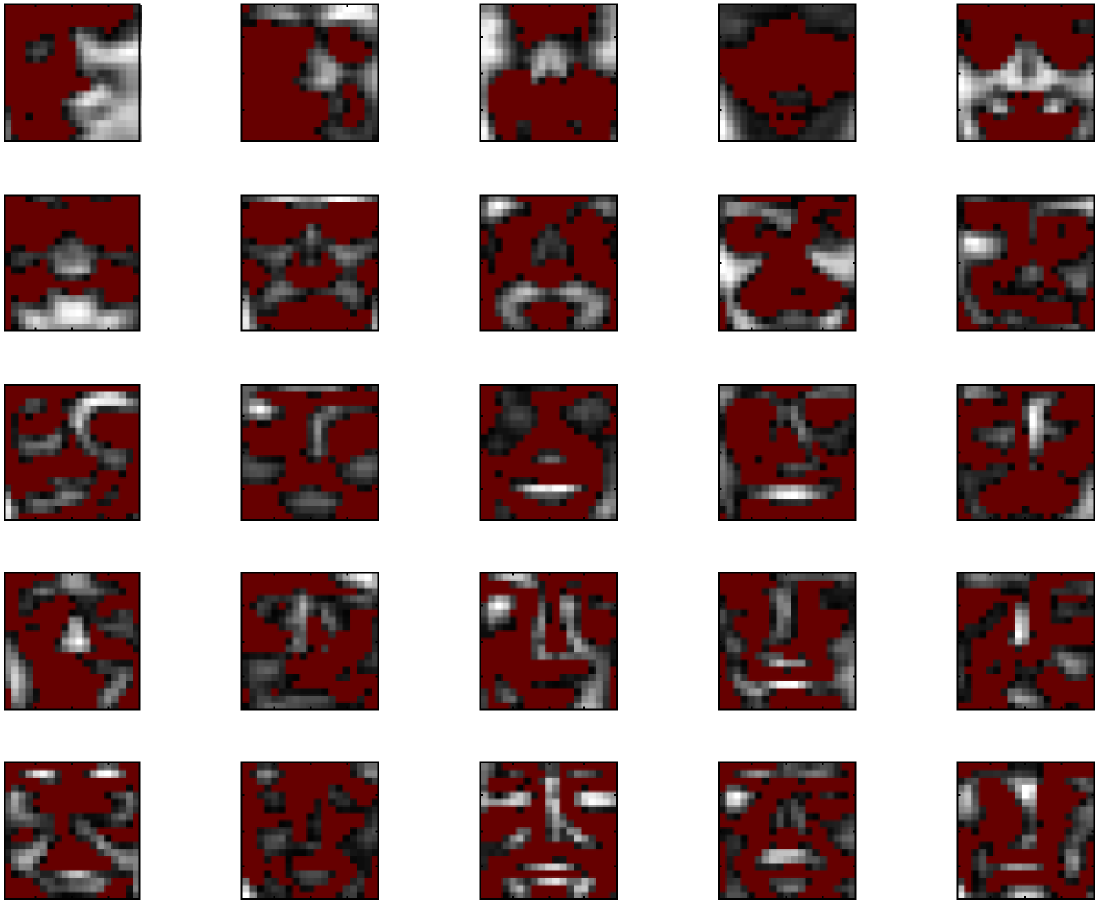
- Disadvantages
  - Non-unique
  - No global solution

# Samples

# Explanation of face image by PCA



# Explanation of face image by PCA



*Red pixels indicate* ***negative values****! How to interpret this?*

# Explanation of face image by NMF

# Cost Function & Updating Rules

- Cost function

$$\underset{W,H}{\operatorname{argmin}} \left\| V - WH \right\|^2, \quad s.t.\, W \geq 0, H \geq 0$$

$$\underset{W,H}{\operatorname{argmin}} \sum_{i,j} \left( V_{ij} \log \frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij} \right), \quad s.t.\, W \geq 0, H \geq 0$$

# Cost Function & Updating Rules

- Cost function

$$\operatorname*{argmin}_{W,H} \left\| V - WH \right\|^2, \quad s.t.\, W \geq 0, H \geq 0$$

$$\operatorname*{argmin}_{W,H} \sum_{i,j} \left( V_{ij} \log \frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij} \right), \quad s.t.\, W \geq 0, H \geq 0$$

- Updating rules
  - Multiplicative updating rule
  - Gradient descent updating rule
  - Alternative least square updating rule

# Updating Rules

- Cost function

$$f\left(W,H\right)=\frac{1}{2}\left\|V-WH\right\|^{2},\quad s.t.\, W\geq 0, H\geq 0$$

# Updating Rules

- Cost function

$$f(W,H)=\frac{1}{2}\left\|V-WH\right\|^2, \quad s.t.\, W\geq 0, H\geq 0$$

- Gradient

$$\begin{cases} \nabla f_W = (V-WH)*(-H^T) \\ \nabla f_H = -W^T*(V-WH) \end{cases}$$

# Updating Rules

- Cost function

$$f(W,H)=\frac{1}{2}\|V-WH\|^2, \quad s.t.\, W\geq 0, H\geq 0$$

- Gradient

$$\begin{cases} \nabla f_W = (V-WH)*(-H^T) \\ \nabla f_H = -W^T*(V-WH) \end{cases}$$

- Solution

$$W_{ia}^* \leftarrow W_{ia}\cdot\frac{(VH^T)_{ia}}{(WHH^T)_{ia}}$$

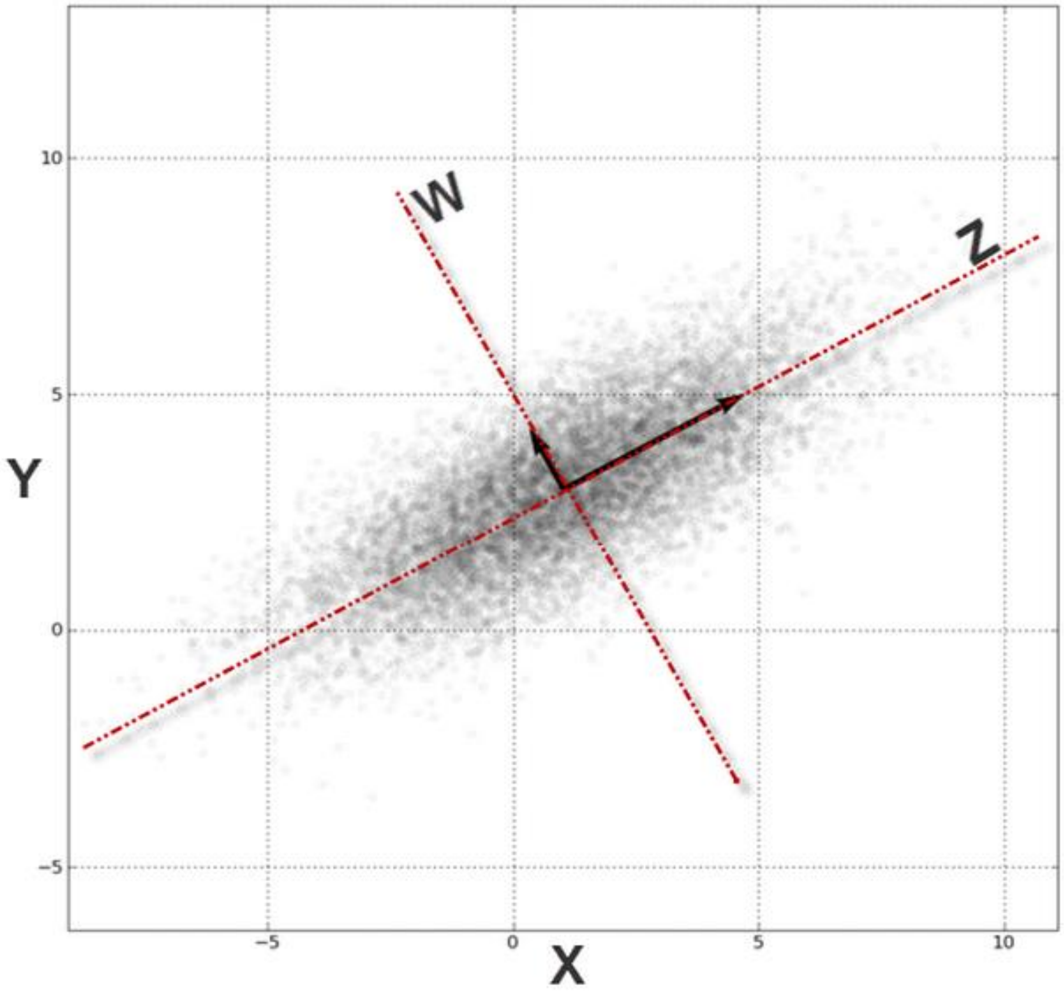
$$H_{aj}^* \leftarrow H_{aj}\cdot\frac{(W^TV)_{aj}}{(W^TWH)_{aj}}$$

# Principal Component Analysis

- PCA

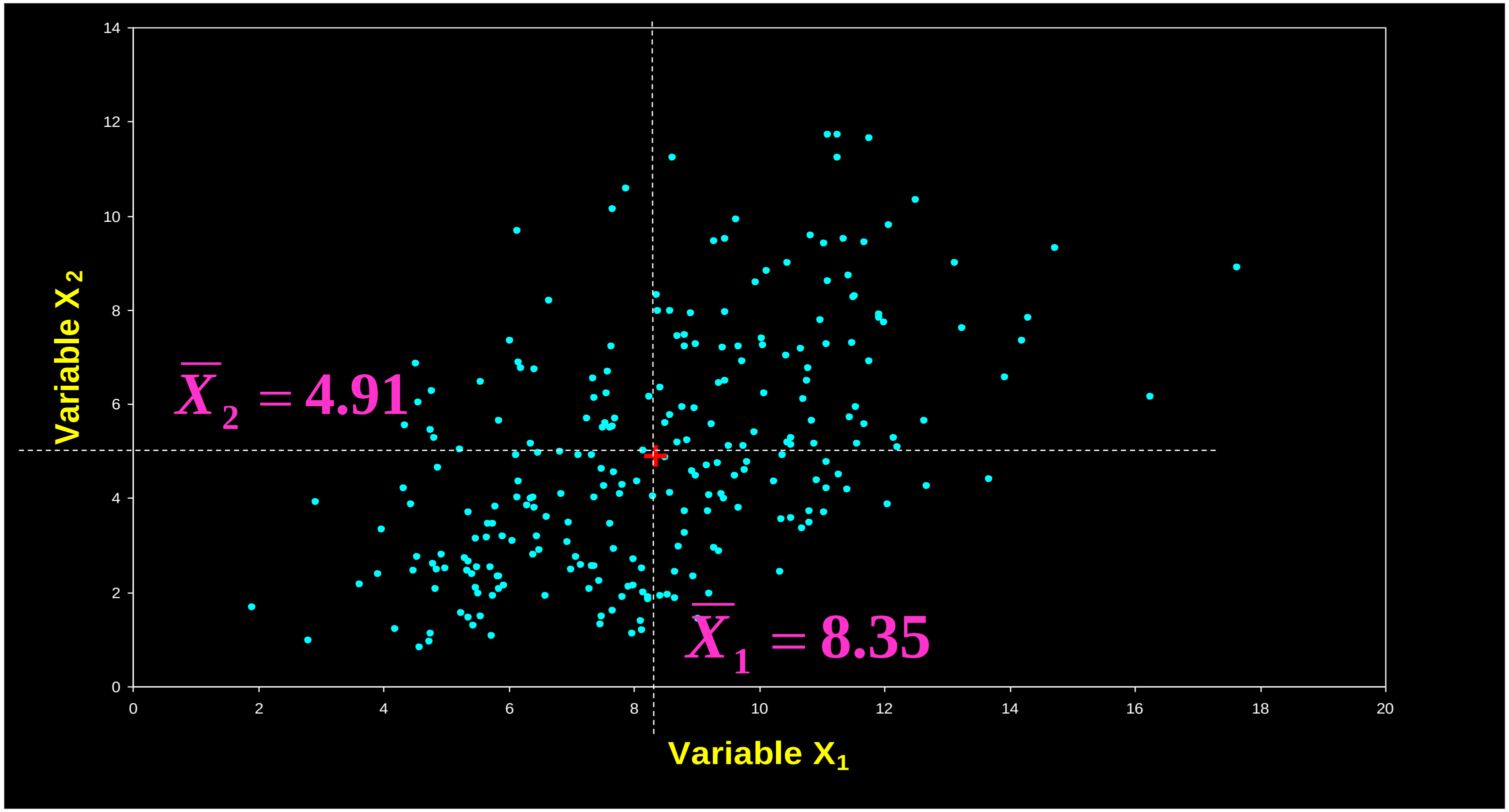
$$Y = PX$$

- De-correlation



# Principal Component Analysis

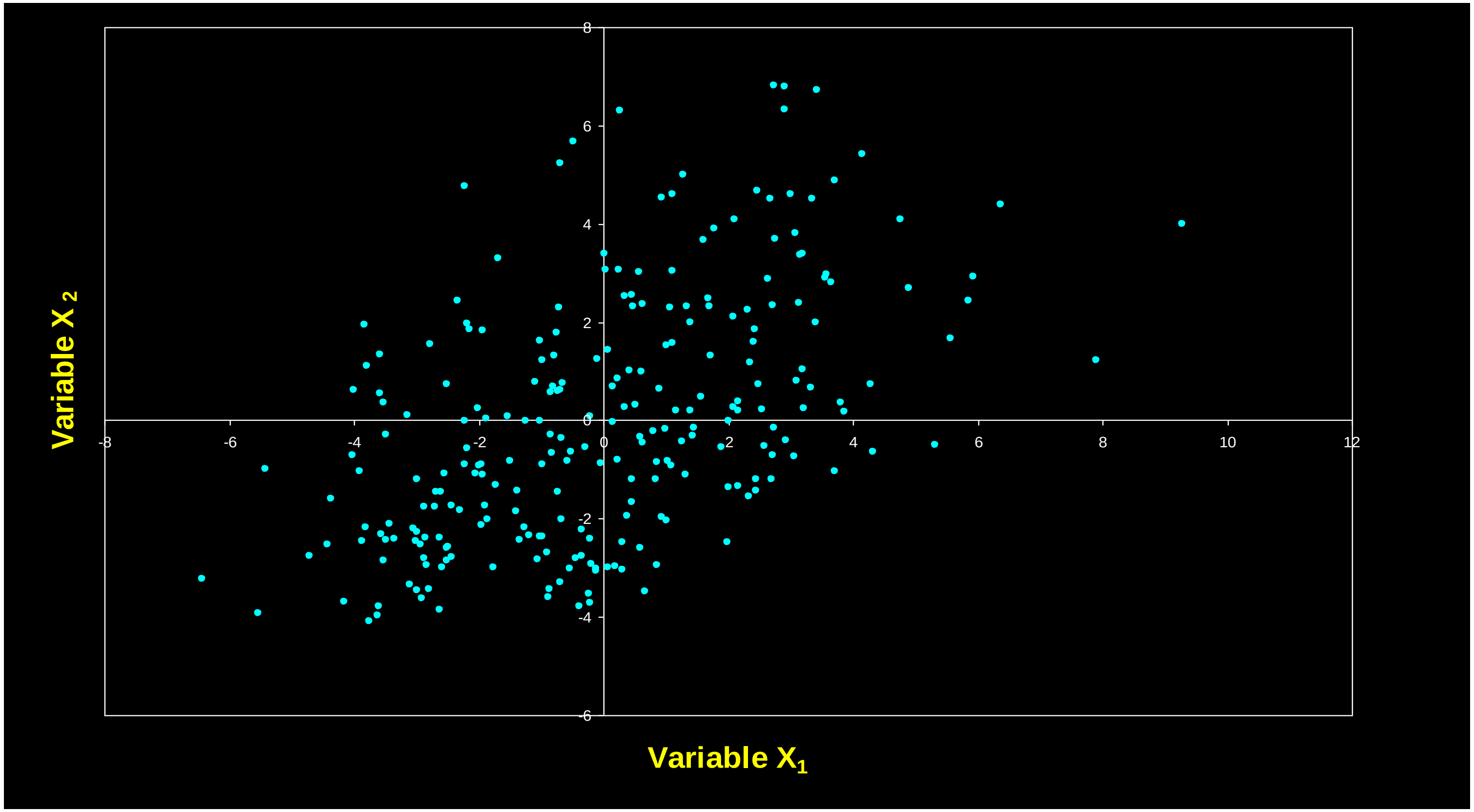
- variables $X_1$ and $X_2$ have positive covariance & each has a similar variance.




$$V_1 = 6.67 \qquad V_2 = 6.24 \qquad C_{1,2} = 3.42$$

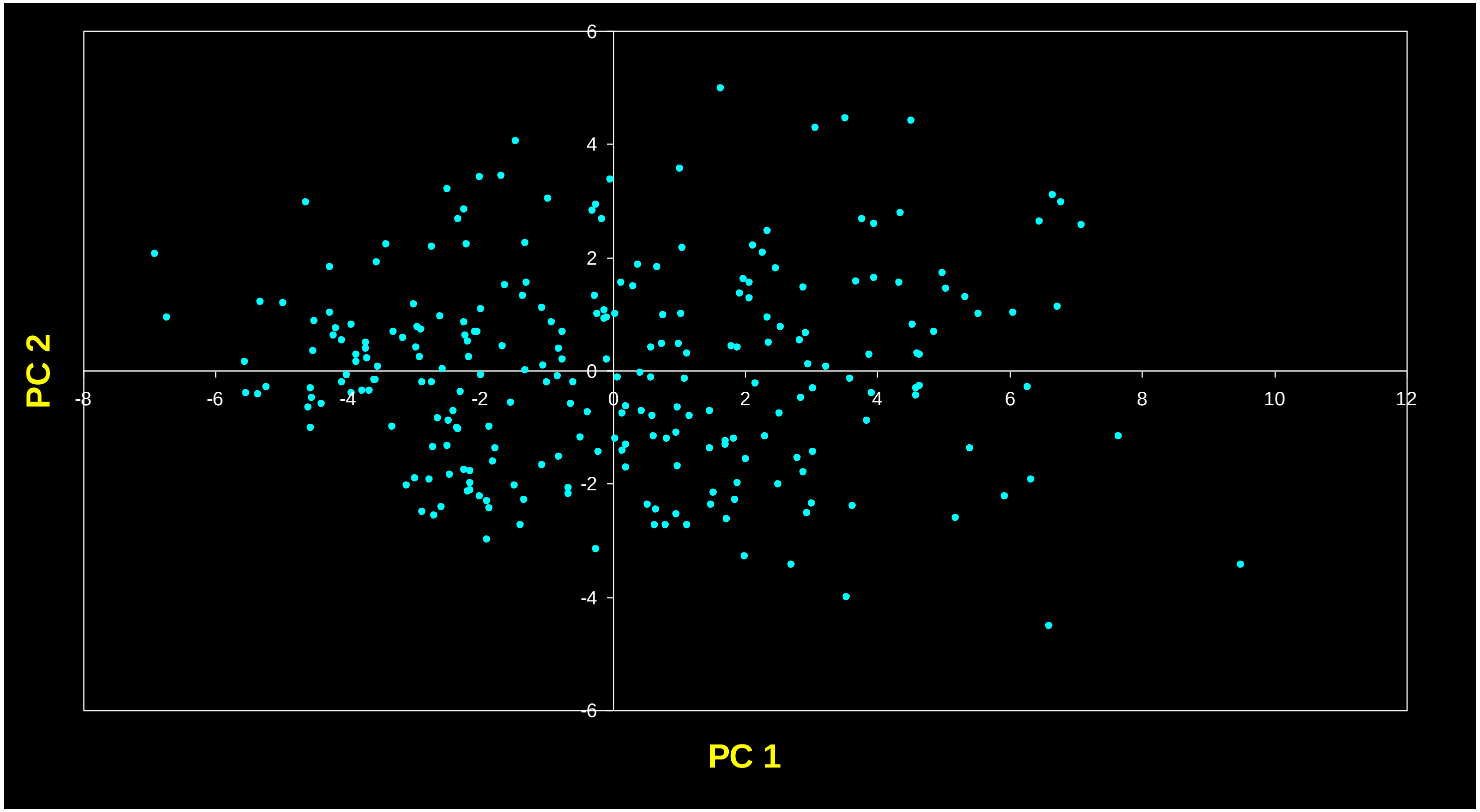
# Principal Component Analysis

- each variable is adjusted to a mean of zero (by subtracting the mean from each value)



# Principal Component Analysis

- PC 1 has the highest possible variance (9.88)
PC 2 has a variance of 3.03
PC 1 and PC 2 have zero covariance.



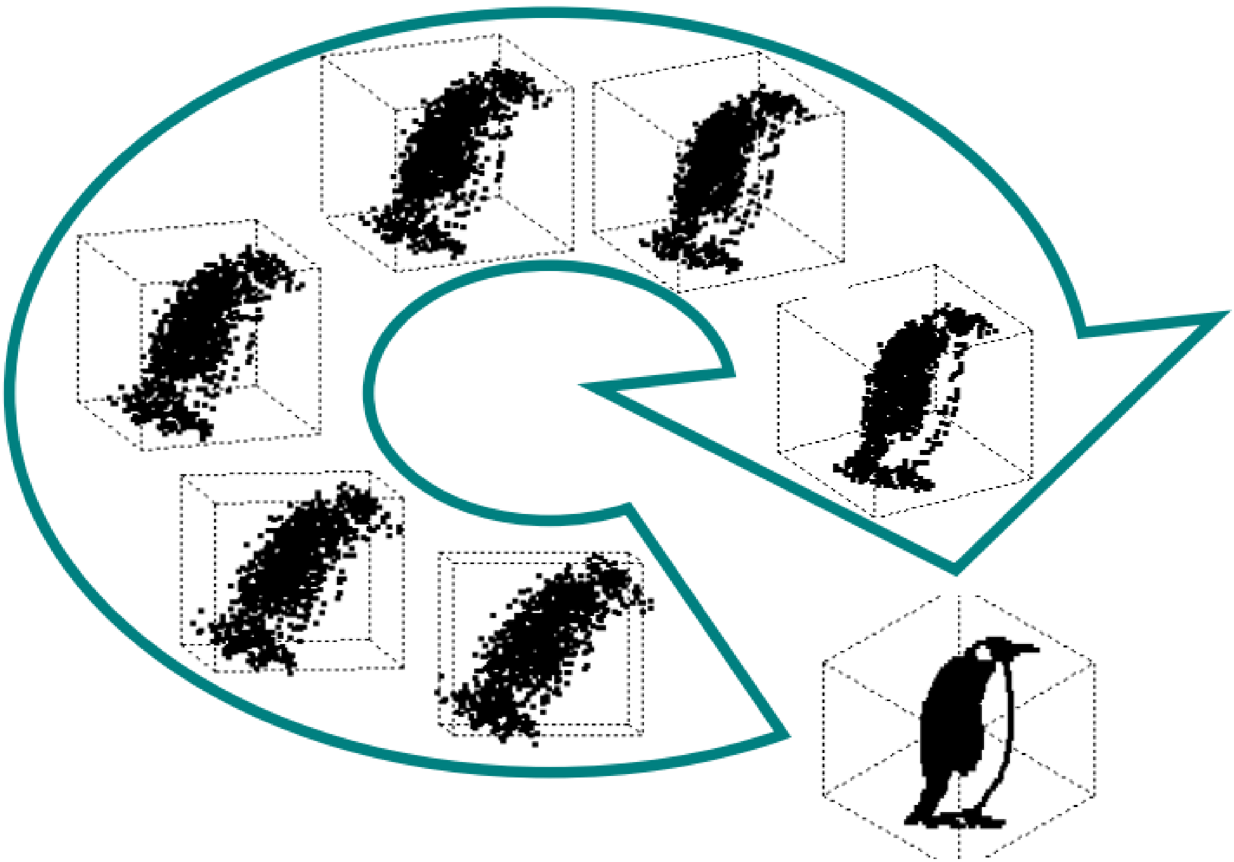
# Principal Component Analysis

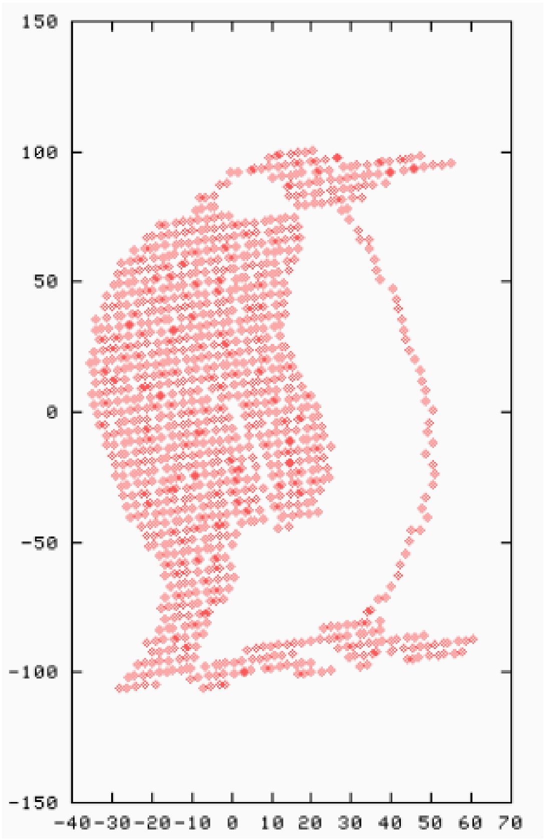
- 1. principal component (PC1)
  The eigenvalue with the largest absolute value will indicate that the data have the largest variance along its eigenvector, the direction along which there is greatest variation

- 2. principal component (PC2)
  the direction with maximum variation left in data, orthogonal to the PC1

- 3. eigenvalue of the covariance matrix
  In general, only few directions manage to capture most of the variability in the data.

# Principal Component Analysis





# Singular Value Decomposition

- Singular value decomposition (SVD) of $m \times n$ matrix $\boldsymbol{A}$ has form

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$$

where $\boldsymbol{U}$ is $m \times m$ orthogonal matrix, $\boldsymbol{V}$ is $n \times n$ orthogonal matrix, and $\boldsymbol{\Sigma}$ is $m \times n$ diagonal matrix, with

$$\sigma_{ij} = \begin{cases} 0 & \text{for } i \neq j \\ \sigma_i \geq 0 & \text{for } i = j \end{cases}$$

# Singular Value Decomposition

- Singular value decomposition (SVD) of $m \times n$ matrix $\boldsymbol{A}$ has form

$$\boxed{\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T}$$

where $\boldsymbol{U}$ is $m \times m$ orthogonal matrix, $\boldsymbol{V}$ is $n \times n$ orthogonal matrix, and $\boldsymbol{\Sigma}$ is $m \times n$ diagonal matrix, with

$$\sigma_{ij} = \begin{cases} 0 & \text{for } i \neq j \\ \sigma_i \geq 0 & \text{for } i = j \end{cases}$$

- Diagonal entries $\sigma_i$, called *singular values* of $\boldsymbol{A}$, are usually ordered so that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$

- Columns $\boldsymbol{u}_i$ of $\boldsymbol{U}$ and $\boldsymbol{v}_i$ of $\boldsymbol{V}$ are called left and right *singular vectors*

# Singular Value Decomposition

- SVD of $\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}$ is given by $\boldsymbol{U\Sigma V}^T =$

$$\begin{bmatrix} .141 & .825 & -.420 & -.351 \\ .344 & .426 & .298 & .782 \\ .547 & .0278 & .664 & -.509 \\ .750 & -.371 & -.542 & .0790 \end{bmatrix} \begin{bmatrix} 25.5 & 0 & 0 \\ 0 & 1.29 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} .504 & .574 & .644 \\ -.761 & -.057 & .646 \\ .408 & -.816 & .408 \end{bmatrix}$$