Dimensionality Reduction by PCA

Pattern Matrix

- Statistics and machine learning typically starts from data given in the form of **observations**, **feature vectors** or **patterns**
- Feature vectors (in some m-dimensional Euclidean space)

$$\mathbf{x_i} \in \mathcal{X} \subseteq \mathbb{R}^m, \quad i = 1, \ldots, n$$

- Patterns can be summarizes into the **pattern matrix**

$$\mathbf{X} \in \mathbb{R}^{n \times m}, \quad \mathbf{X} = \begin{pmatrix} \mathbf{x}_1' \\ \cdots \\ \mathbf{x}_i' \\ \cdots \\ \mathbf{x}_n' \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{im} \\ \cdots & \cdots & \cdots & \cdots \\ x_{i1} & x_{i2} & \cdots & x_{im} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix}$$

transposed i-th pattern

Examples: Pattern Matrices

$$\mathbf{X} \in \mathbb{R}^{n \times m}$$

- **Measurement** vectors
  - i: instance number, e.g. a house
  - j: measurement, e.g. the area of a house

$$\begin{array}{cc} & \textit{measurement} \rightarrow \\ \textit{instance} \downarrow & \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{im} \\ \cdots & \cdots & \cdots & \cdots \\ x_{i1} & x_{i2} & \cdots & x_{im} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix} \rightarrow \textit{result of measurem.} \end{array}$$

- **Digital images** as gray-scale **vectors**
  - i: image number
  - j: pixel value at location $j=(k,l)$

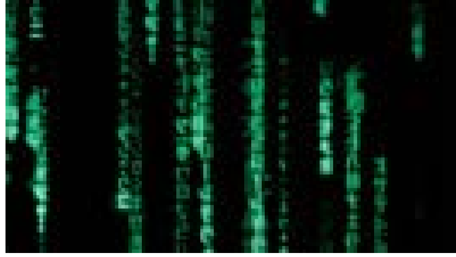
$$\begin{array}{cc} & \textit{pixel} \rightarrow \\ \textit{image} \downarrow & \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{im} \\ \cdots & \cdots & \cdots & \cdots \\ x_{i1} & x_{i2} & \cdots & x_{im} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix} \rightarrow \textit{intensity} \end{array}$$

- **Text documents** in **bag-of-words** representation
  - i: document number
  - j: term (word or phrase) in a vocabulary

$$\begin{array}{cc} & \textit{term} \rightarrow \\ \textit{document} \downarrow & \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{im} \\ \cdots & \cdots & \cdots & \cdots \\ x_{i1} & x_{i2} & \cdots & x_{im} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix} \rightarrow \textit{frequency} \end{array}$$

- **User rating** data
  - i: user number
  - j: item (book, movie)

$$\begin{array}{cc} & \textit{item} \rightarrow \\ \textit{user} \downarrow & \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{im} \\ \cdots & \cdots & \cdots & \cdots \\ x_{i1} & x_{i2} & \cdots & x_{im} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix} \rightarrow \textit{rating} \end{array}$$



Latent Structure

- Given a matrix that “encodes” data …
- Potential problems
  - too large
  - too complicated
  - missing entries
  - noisy entries
  - lack of structure
  - …

$$\mathbf{A} = \begin{pmatrix} a_{11} & \cdots & a_{1j} & \cdots & a_{1m} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{i1} & \cdots & a_{ij} & \cdots & a_{im} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & \cdots & a_{nj} & \cdots & a_{nm} \end{pmatrix}$$

Vocabulary Mismatch & ~~Robustness~~


CNN.com - Analysis: Germany tackles labour shortage - July 5, 2001 - Microsof...

CNN.com

PRINT THIS

Click to Print

SAVE THIS | EMAIL THIS | Close

Analysis: Germany tackles labour shortage

By CNN's Bettina Luscher

BERLIN, Germany -- With Germany's population expected to shrink by one third over the next half century, economic experts see its economy and social welfare system in danger if the country does not encourage more immigrants.

query

match

"labour immigrants Germany"

CNN.com | labor immigrants Ge | FIND

query

?

"German job market for immigrants"

CNN.com | German job market f | FIND

query

?

"foreign workers in Germany"

CNN.com | foreign workers in Ge | FIND

query

?

"German green card"

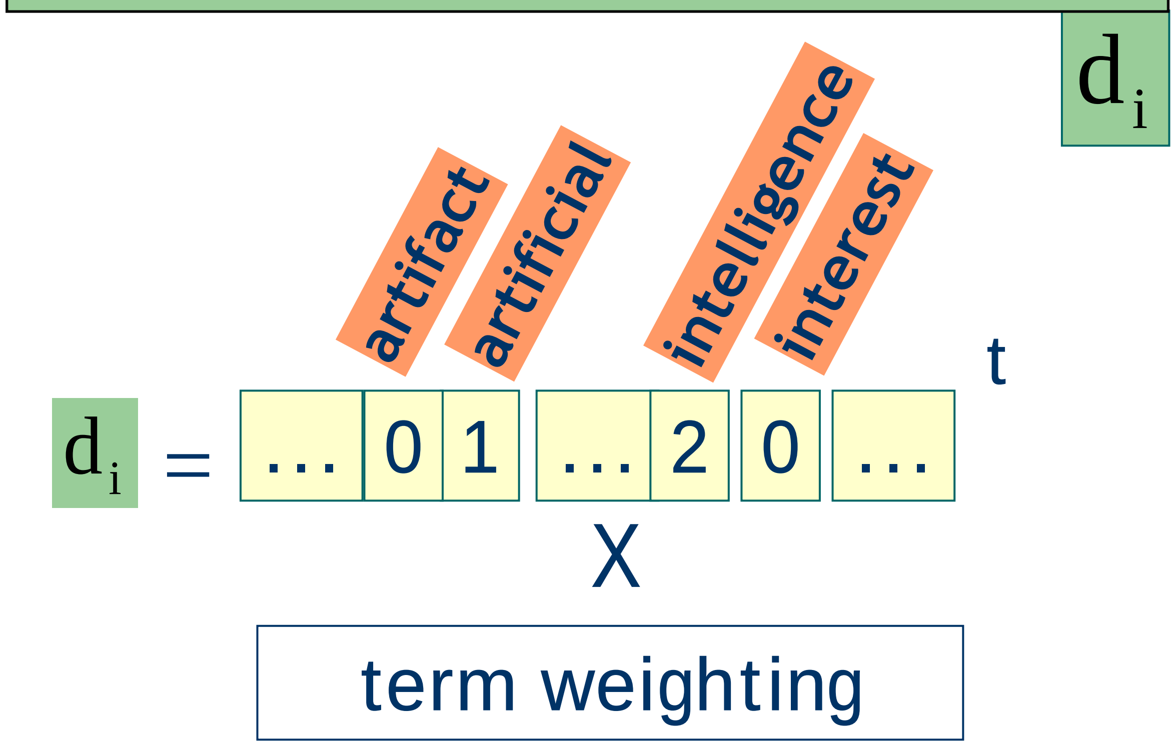
CNN.com | green card Germany | FIND

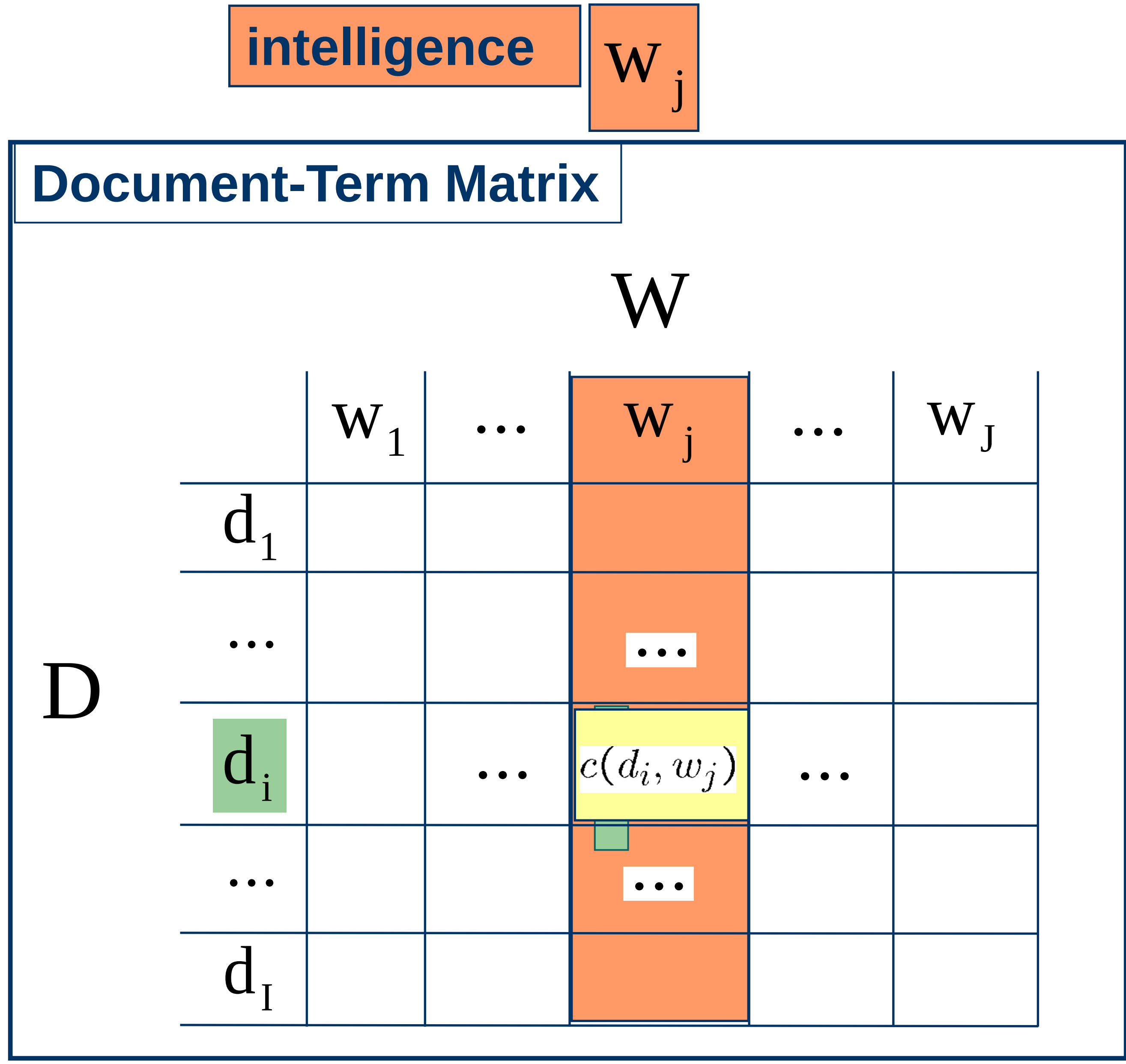
Document-Term Matrix

D = Document collection

W = Lexicon/Vocabulary

Texas Instruments said it has developed the first 32-bit computer chip designed specifically for artificial ***intelligence*** *applications [...]*





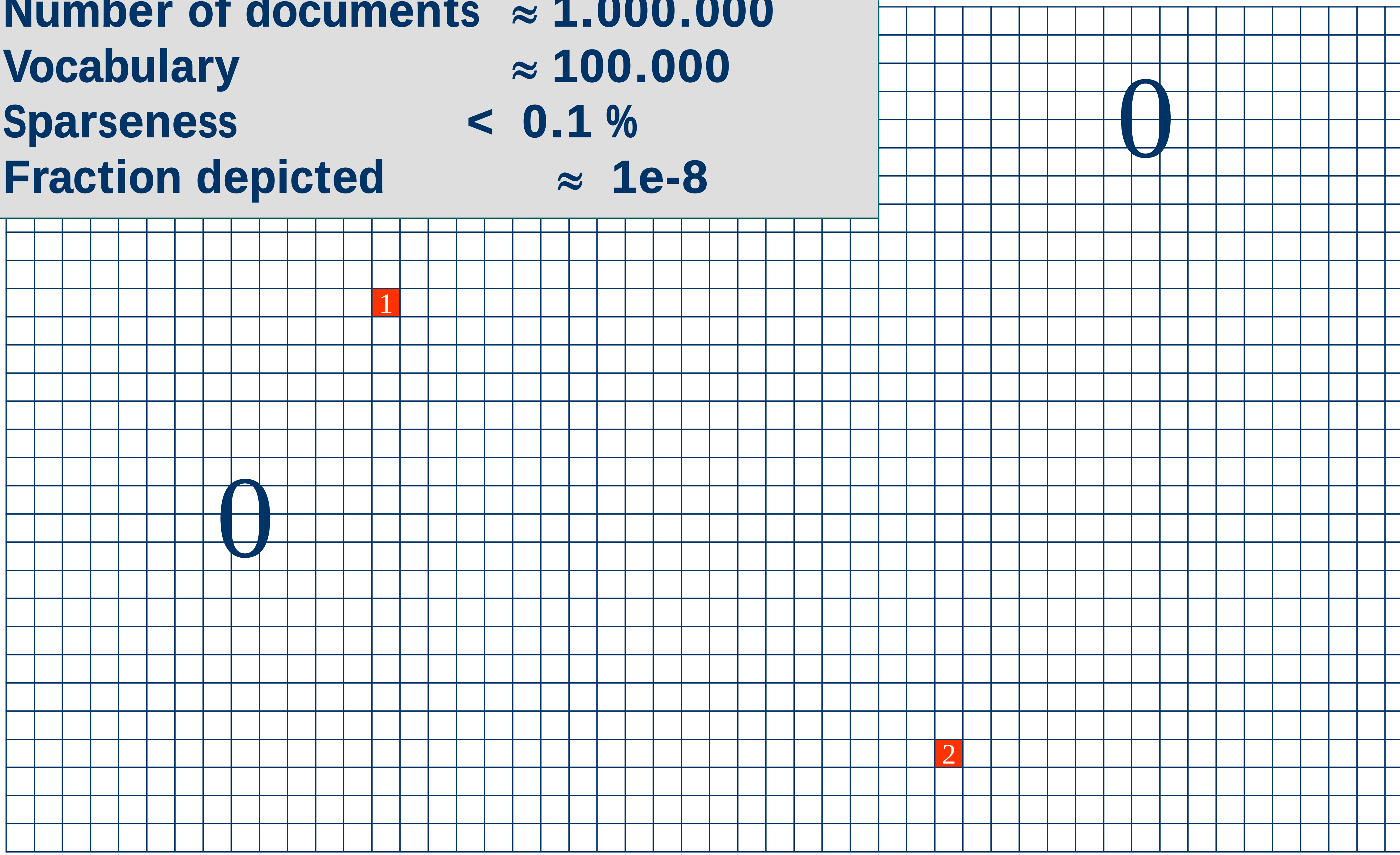
Clustering rows?

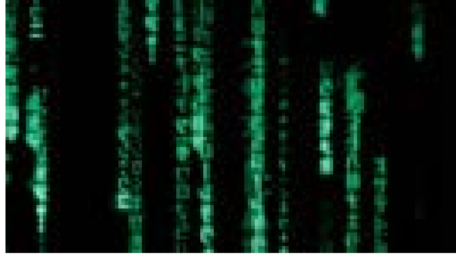
Clustering columns?

A 100 Millionths of a Typical Document-term Matrix

Typical:

- **Number of documents ≈ 1.000.000**
- **Vocabulary ≈ 100.000**
- **Sparseness < 0.1 %**
- **Fraction depicted ≈ 1e-8**





Latent Structure

- Given a matrix that “encodes” data …
- Potential problems
  - too large
  - too complicated
  - missing entries
  - noisy entries
  - lack of structure
  - …

$$\mathbf{A} = \begin{pmatrix} a_{11} & \cdots & a_{1j} & \cdots & a_{1m} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{i1} & \cdots & a_{ij} & \cdots & a_{im} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & \cdots & a_{nj} & \cdots & a_{nm} \end{pmatrix}$$

- Is there a **simpler** way to **explain** entries?
- There might be a **latent structure** underlying the data.
- How can we “find” or “reveal” this structure?

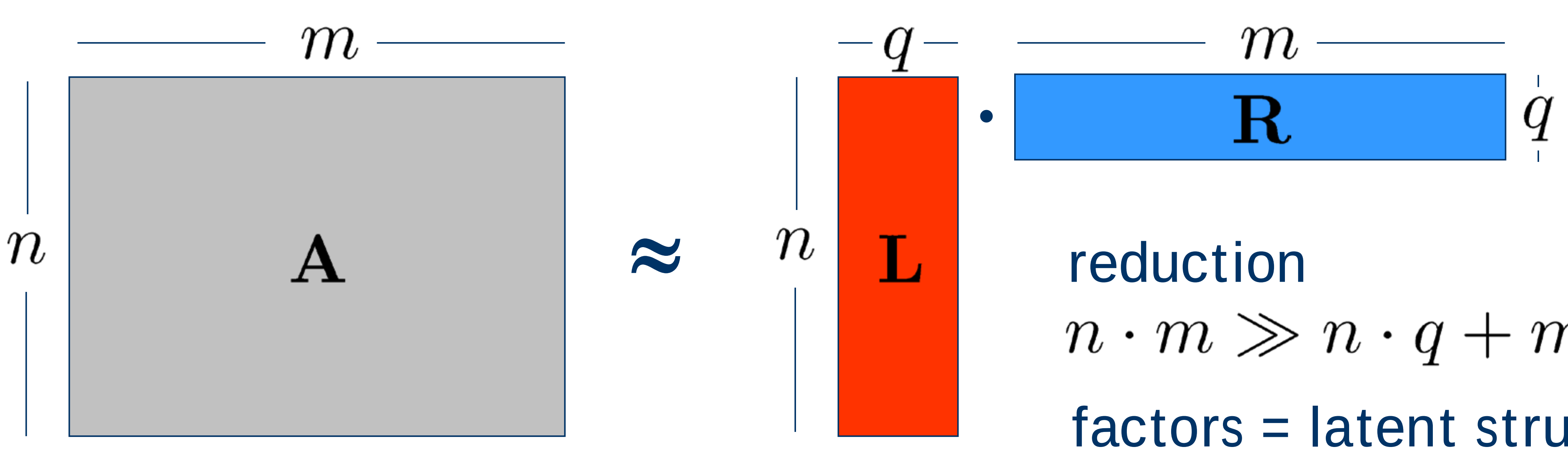
Matrix Decomposition

- Common approach: approximately **factorize** matrix

$$\mathbf{A} \approx \hat{\mathbf{A}} = \mathbf{L} \cdot \mathbf{R}$$

approximation left factor right factor

- Factors are typically constrained to be "**thin**"



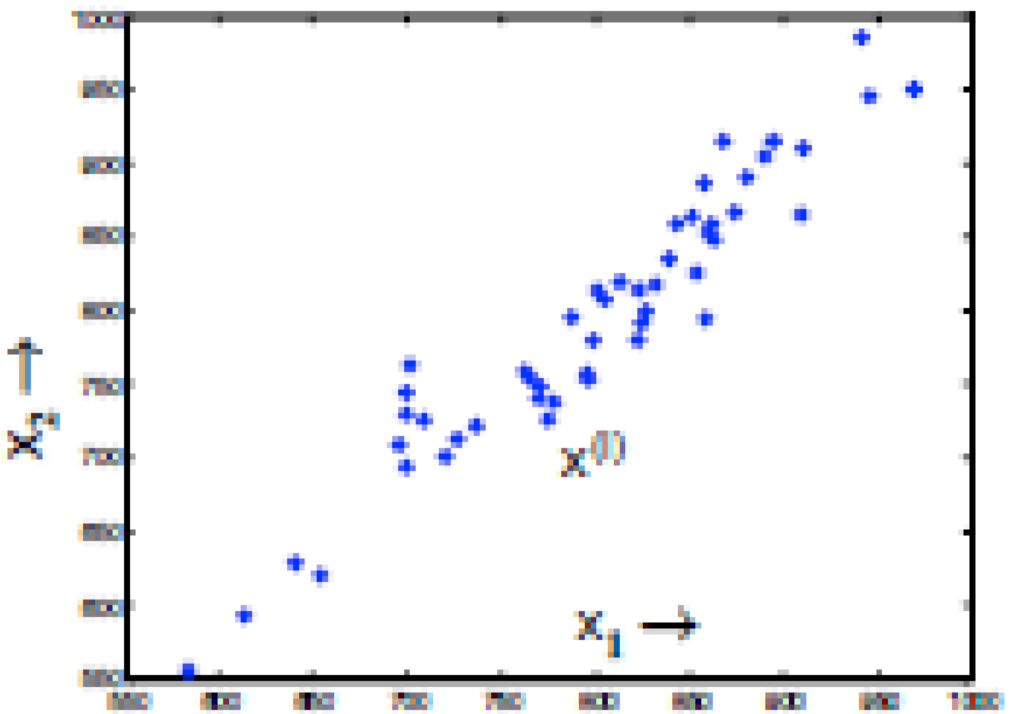
reduction

$$n \cdot m \gg n \cdot q + m \cdot q$$

factors = latent structure (?)

Dimensionality reduction

- Ex: data with two real values $[x_1, x_2]$
- We'd like to describe each point using only one value $[z_1]$
- We'll communicate a "model" to convert: $[x_1, x_2] \sim f(z_1)$

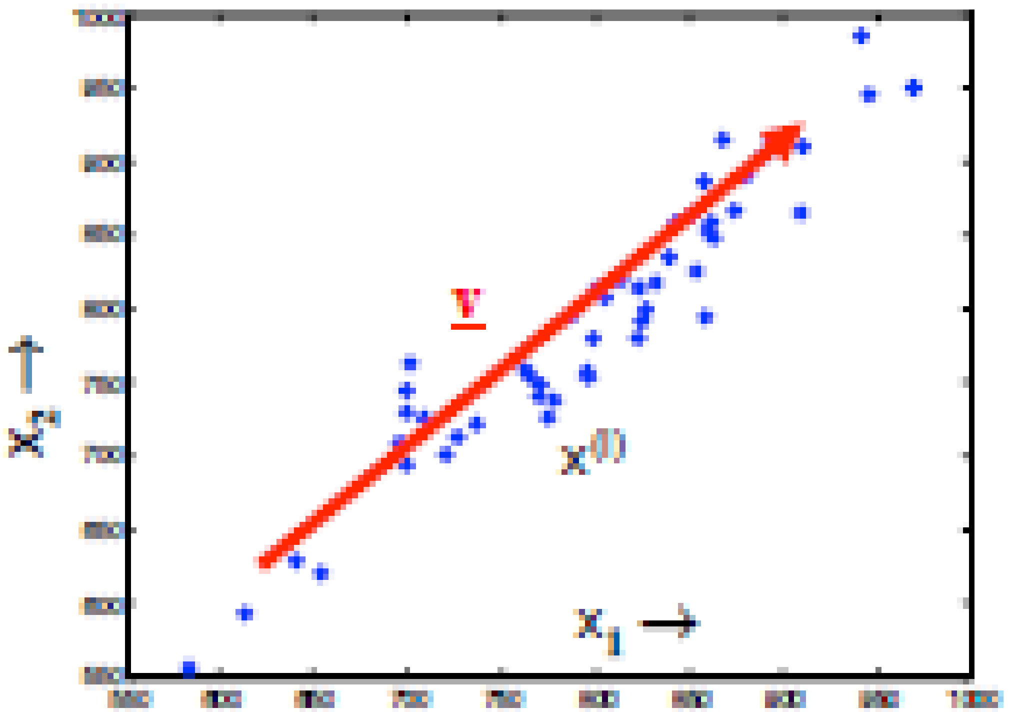


Slides by Alexander Ihler

Dimensionality reduction

- Ex: data with two real values $[x_1,x_2]$
- We'd like to describe each point using only one value $[z_1]$
- We'll communicate a "model" to convert: $[x_1,x_2] \sim f(z_1)$

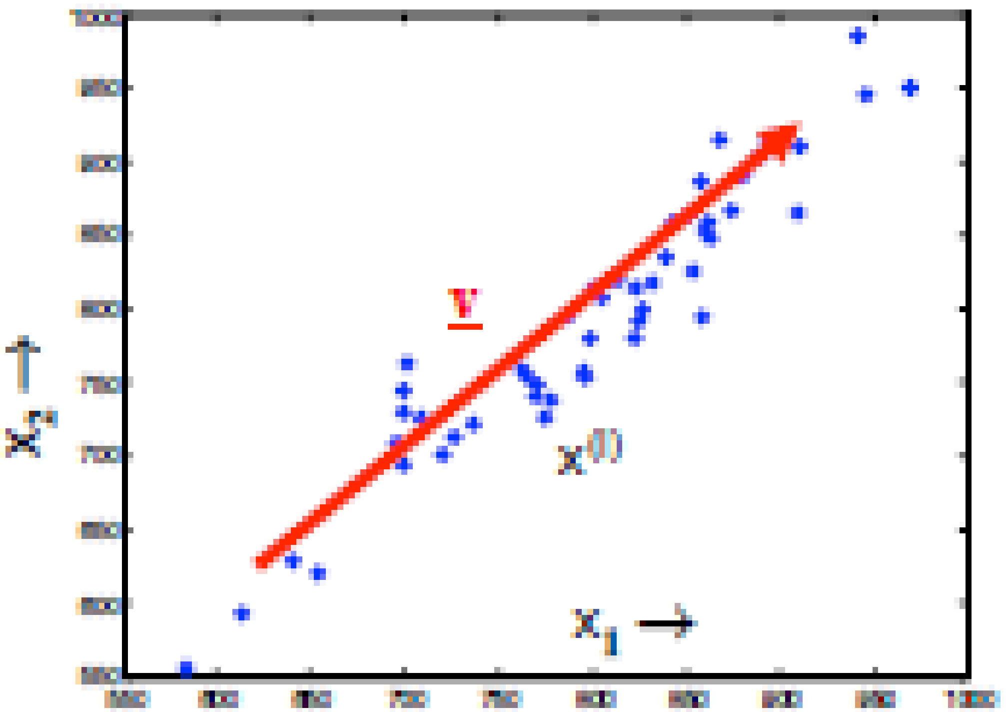
- Ex: linear function f(z): $[x_1,x_2] = z * \underline{v} = z * [v_1,v_2]$
- $\underline{v}$ is the same for all data points (communicate once)
- z tells us the closest point on v to the original point $[x_1,x_2]$

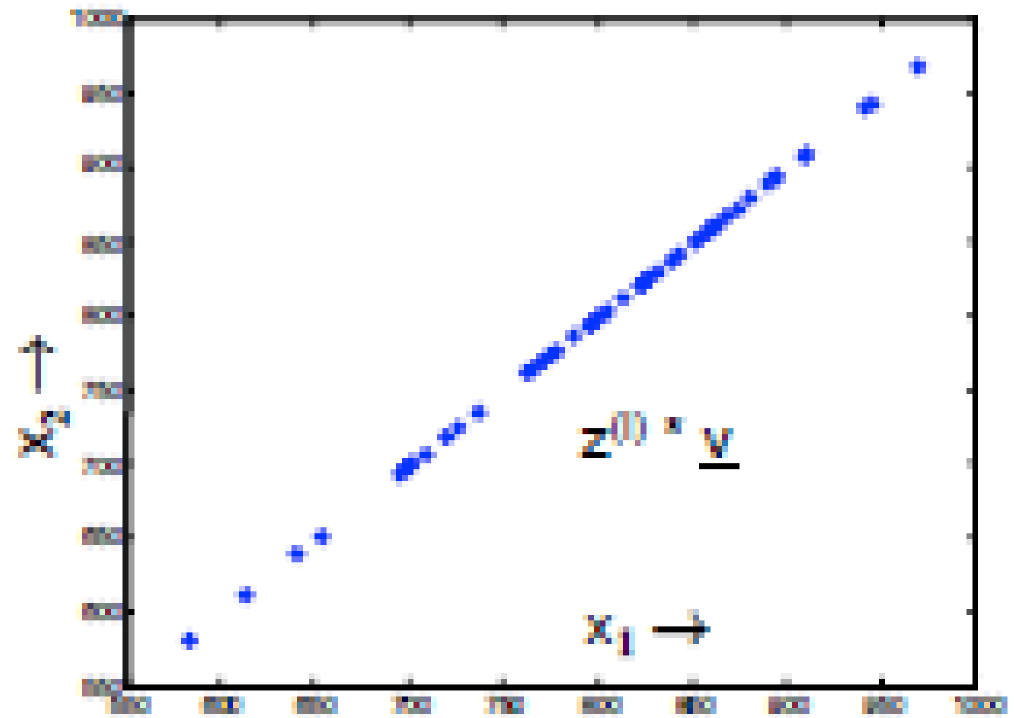


Dimensionality reduction

- Ex: data with two real values $[x_1,x_2]$
- We'd like to describe each point using only one value $[z_1]$
- We'll communicate a "model" to convert: $[x_1,x_2] \sim f(z_1)$

- Ex: linear function f(z): $[x_1,x_2] = z * \underline{v} = z * [v_1,v_2]$
- $\underline{v}$ is the same for all data points (communicate once)
- z tells us the closest point on v to the original point $[x_1,x_2]$



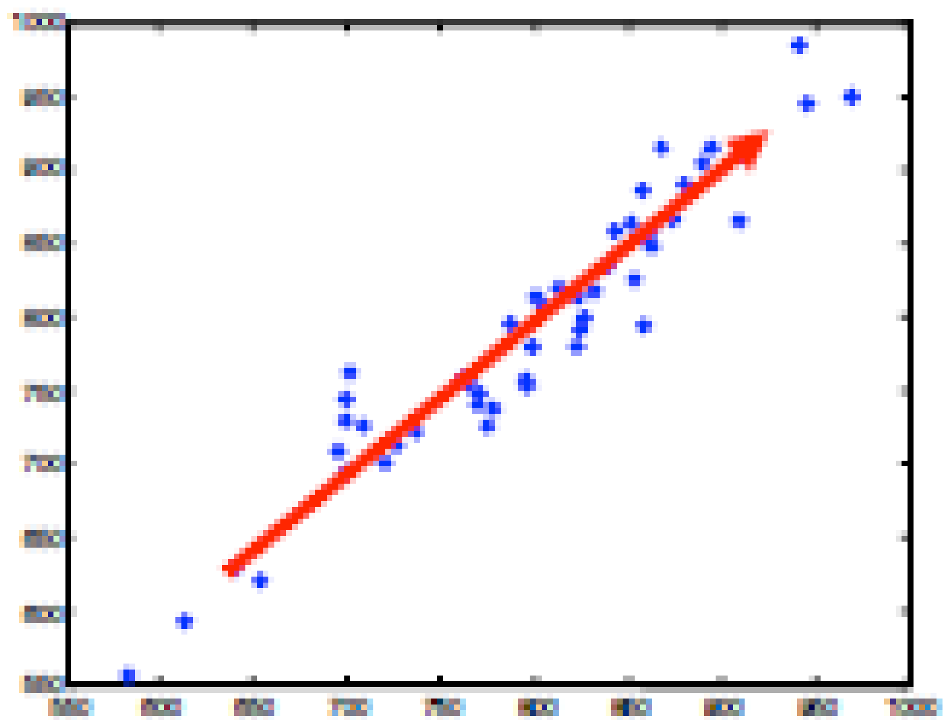


Principal Components Analysis

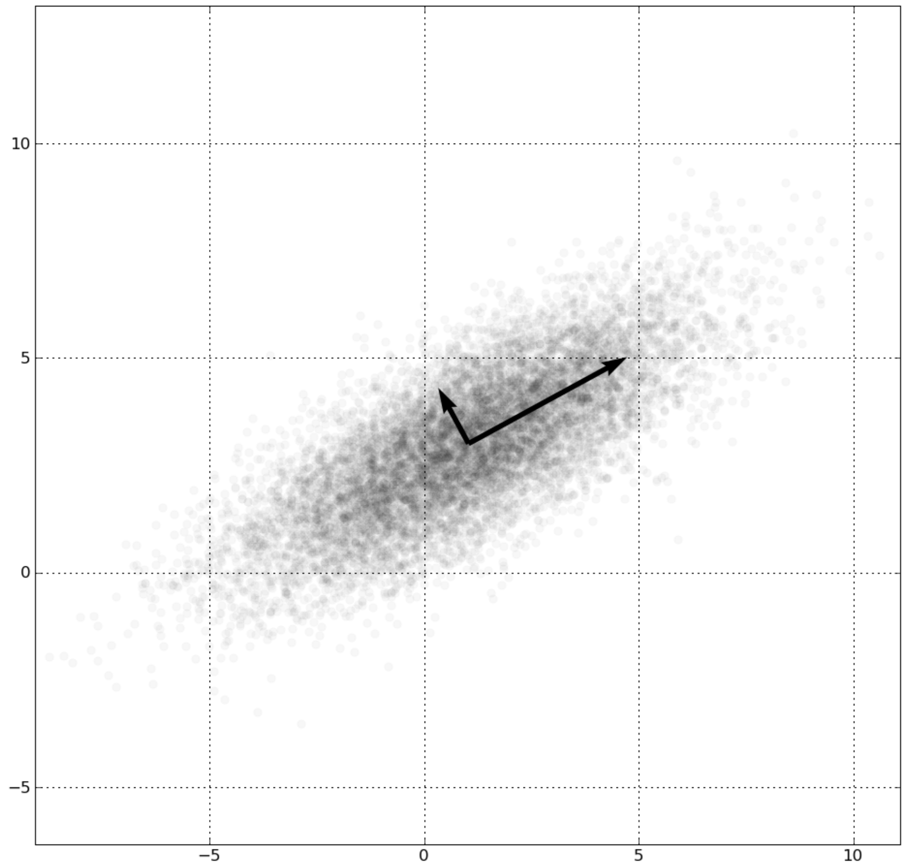
- What is the vector that would most closely reconstruct X?

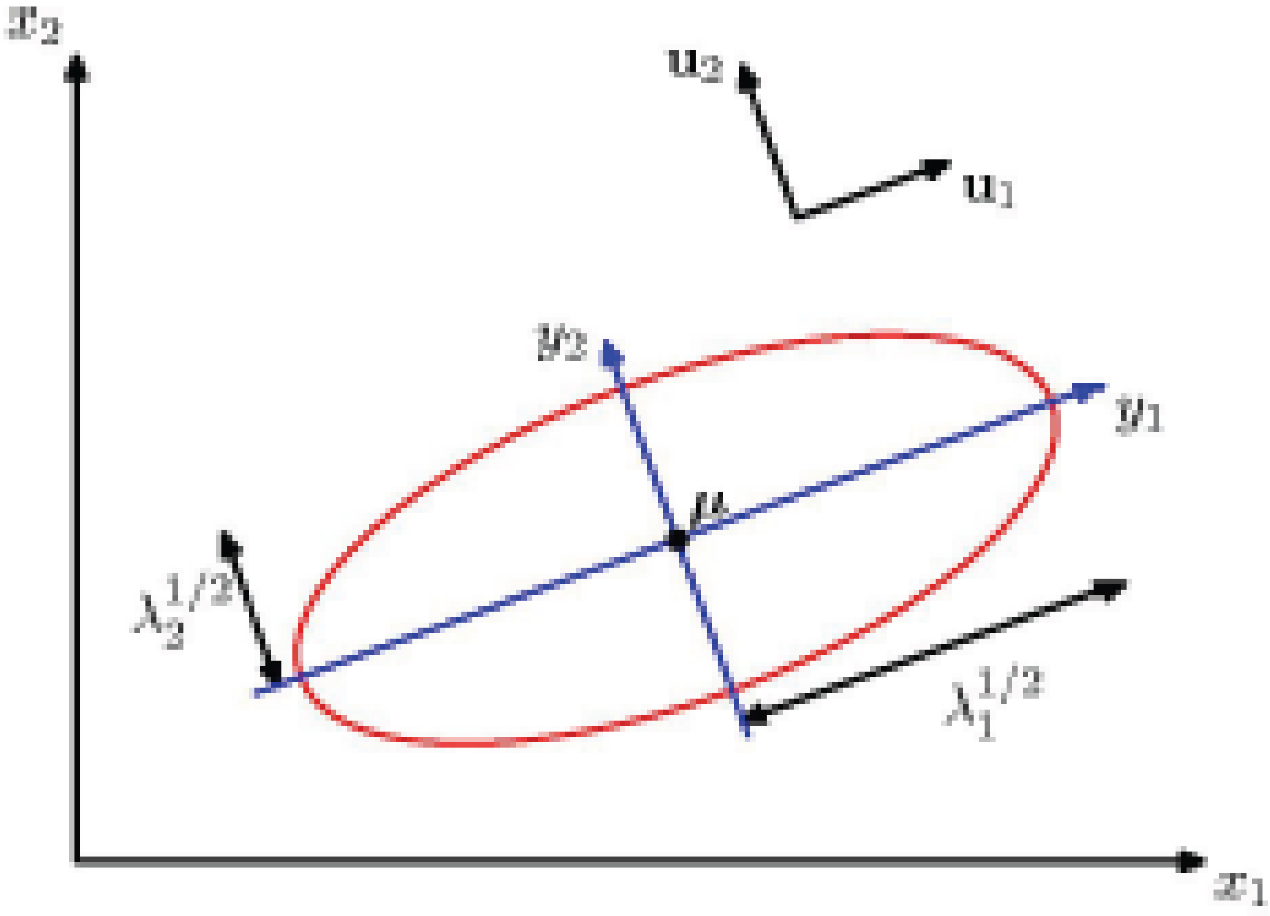
$$\min_{a,v} \sum_i (x^{(i)} - a^{(i)} v)^2$$

  - Given v: $a^{(i)}$ is the projection of each point $x^{(i)}$ onto v
  - v chosen to minimize the residual variance
  - Equivalently, v is the direction of maximum variance
  - Extensions: best two dimensions: xi= ai*v + bi*w + m



Eigenvectors





PCA

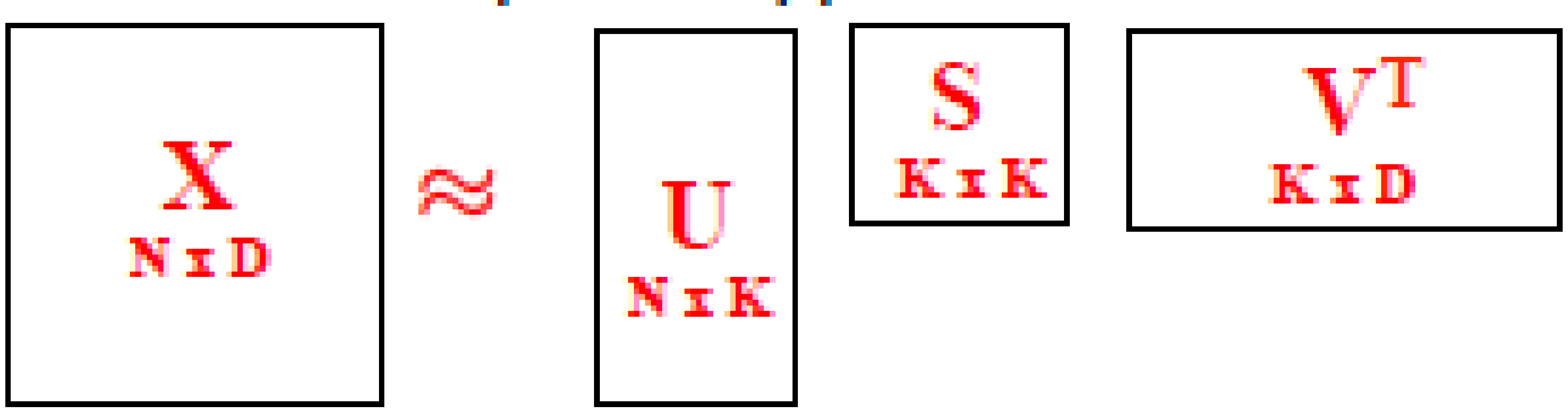
Given pattern matrix X,

1. Subtract mean from each point
2. (sometimes) scale each dimension by its variance
3. Compute covariance matrix $S=X^T X$
4. Compute k largest eigenvectors of S

$S = V D V^T$

Singular Value Decomposition

- Alternative method to calculate (still subtract mean 1st)
- Decompose $X = U\ S\ V^T$
  - Orthogonal: $X^T X = V\ S\ S\ V^T = V\ D\ V^T$
  - $X\ X^T = U\ S\ S\ U^T = U\ D\ U^T$
- U*S matrix provides coefficients
  - Example $x_i = U_{i,1}\ S_{11}\ v_1 + U_{i,2}\ S_{22}\ v_2 + \ldots$
- Gives the least-squares approximation to X of this form



Glorious SVD

$$X = USV^T$$

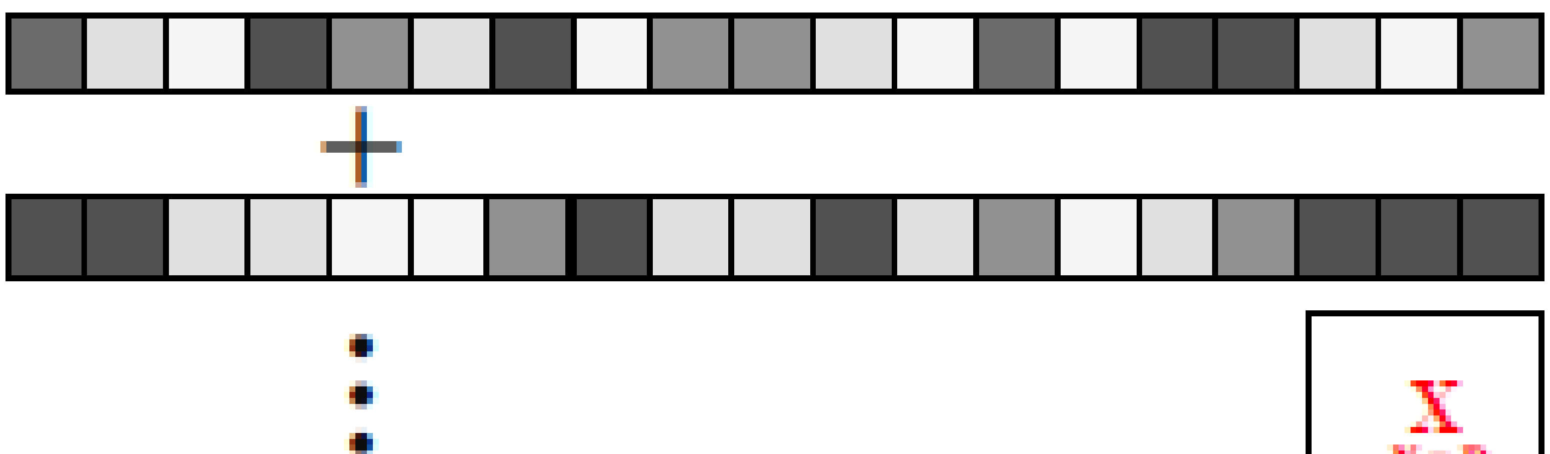
- XX^T and X^TX share the same eigenvalues
- Even better: their eigenvectors are related
  - Xv_i is an eigenvector of XX^T

"Eigen-faces"

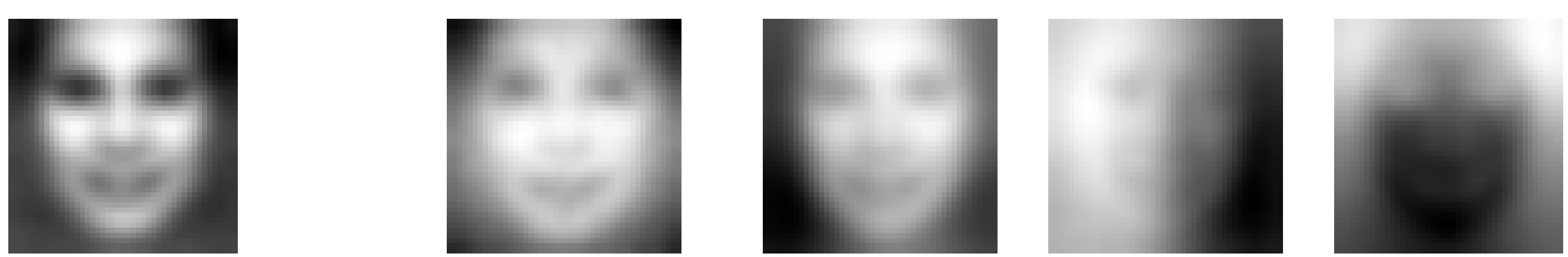
- "Eigen-X" = represent X using PCA
- Ex: Viola Jones data set
  - 24x24 images of faces = 576 dimensional measurements

"Eigen-faces"

- "Eigen-X" = represent X using PCA
- Ex: Viola Jones data set
  - 24x24 images of faces = 576 dimensional measurements
  - Take first K PCA components



$$X_{N \times D} \approx U_{N \times K} \; S_{K \times K} \; V^T_{K \times D}$$



Mean V(1,:) V(2,:) V(3,:) V(4,:) …

“Eigen-faces”

- “Eigen-X” = represent X using PCA
- Ex: Viola Jones data set
  - 24x24 images of faces = 576 dimensional measurements
  - Take first K PCA components