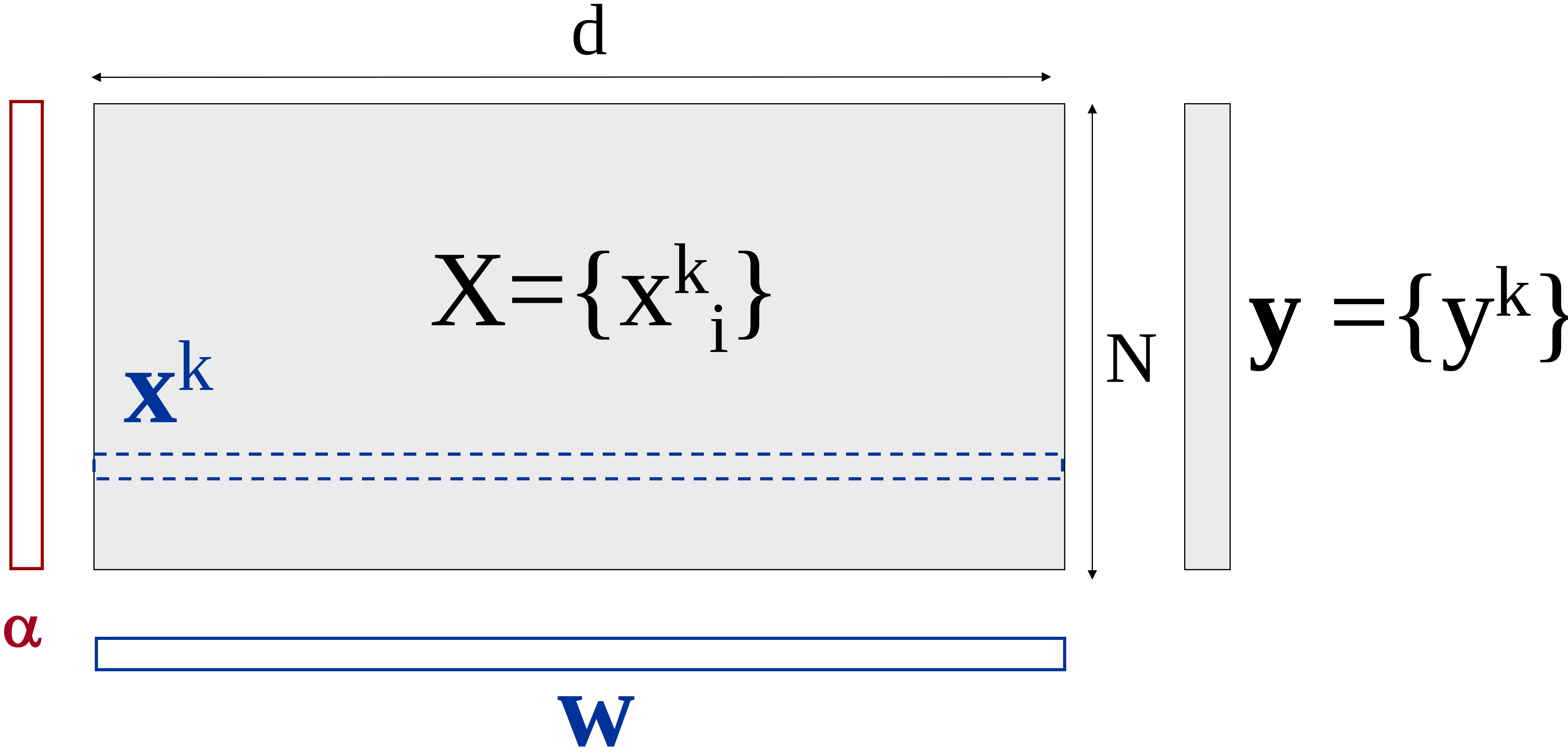
Dimensionality Reduction by PCA

Pattern Matrix



d

$X=\{x^k{}_i\}$

$\mathbf{x}^k$

N

$\mathbf{y}=\{y^k\}$

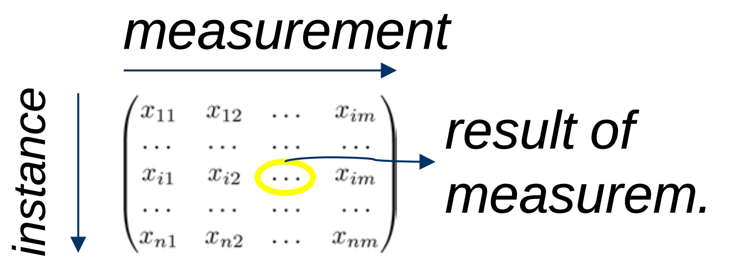
α

$\mathbf{w}$

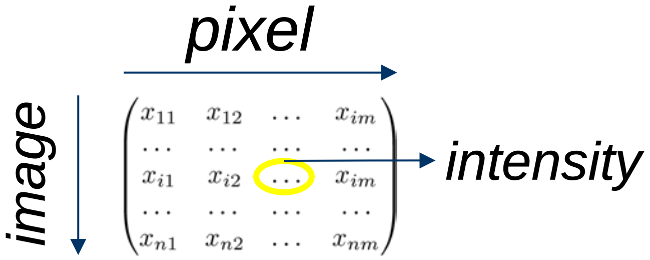
Examples: Pattern Matrices

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

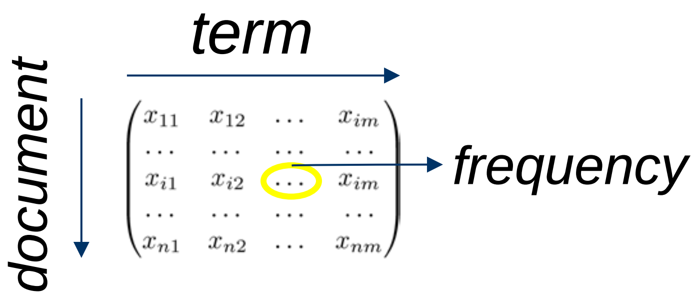
- **Measurement** vectors
  - i: instance number, e.g. a house
  - j: measurement, e.g. the area of a house


measurement
instance
x_{11} x_{12} … x_{1m}
x_{i1} x_{i2} … x_{im}
x_{n1} x_{n2} … x_{nm}
result of measurem.

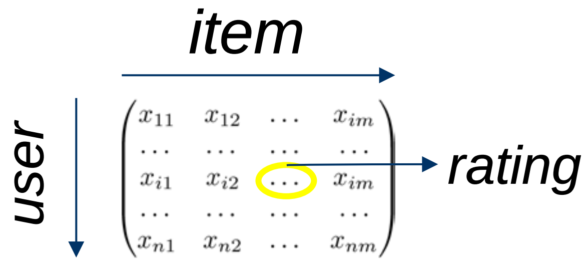
- **Digital images** as gray-scale **vectors**
  - i: image number
  - j: pixel value at location $j=(k,l)$


pixel
image
x_{11} x_{12} … x_{1m}
x_{i1} x_{i2} … x_{im}
x_{n1} x_{n2} … x_{nm}
intensity

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


term
document
x_{11} x_{12} … x_{1m}
x_{i1} x_{i2} … x_{im}
x_{n1} x_{n2} … x_{nm}
frequency

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


item
user
x_{11} x_{12} … x_{1m}
x_{i1} x_{i2} … x_{im}
x_{n1} x_{n2} … x_{nm}
rating

Document-Term Matrix

D = Document collection

W = Lexicon/Vocabulary

intelligence w_j

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

d_i

$$\begin{array}{ccccccccc} & & & \text{artifact} & \text{artificial} & & \text{intelligence} & \text{interest} & \\ d_i & = & \dots & 0 & 1 & \dots & 2 & 0 & \dots \end{array}^{t}$$

X

term weighting

Document-Term Matrix

W

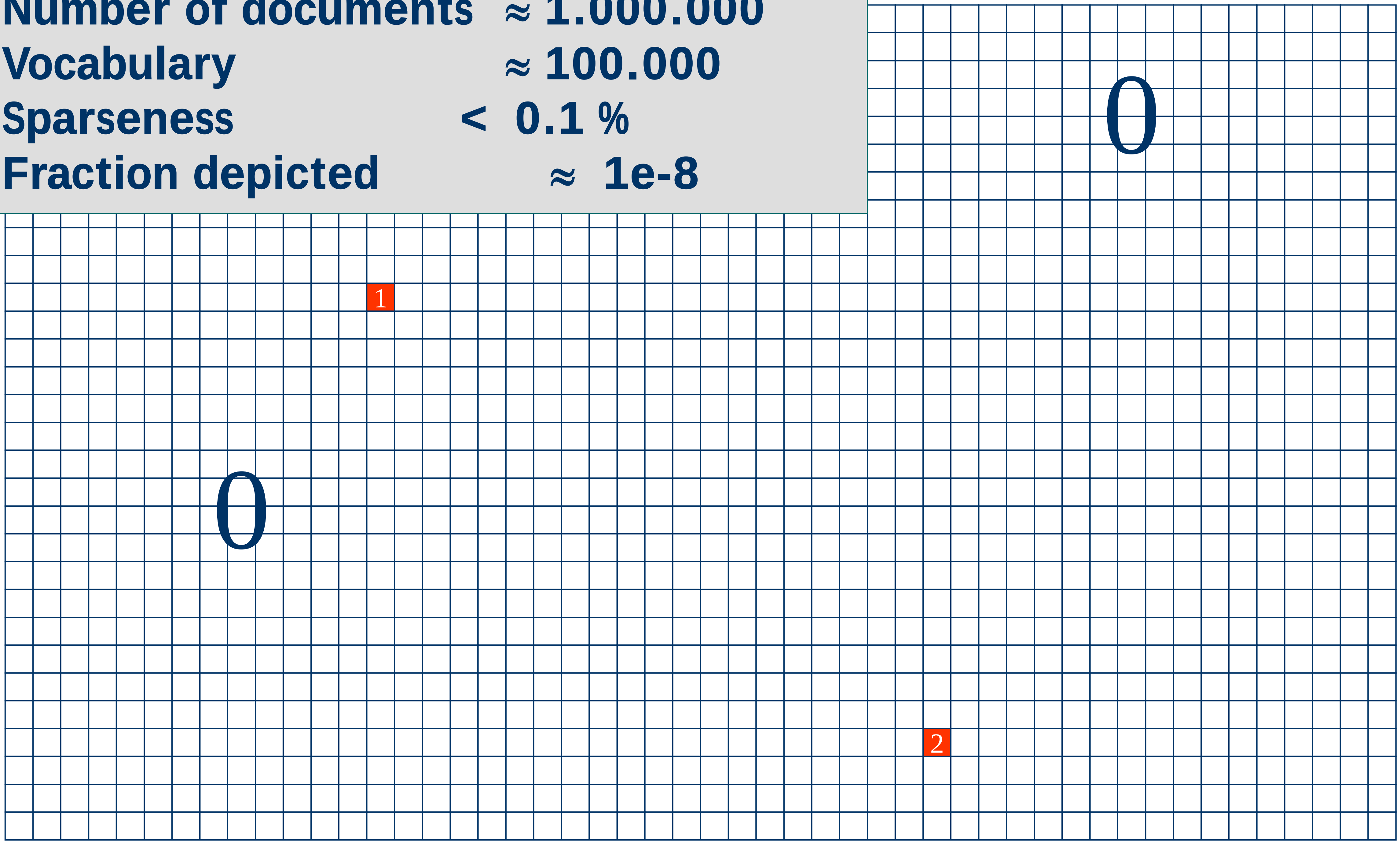
D

$$\begin{array}{c|ccccc} & w_1 & \dots & w_j & \dots & w_J \\ \hline d_1 & & & & & \\ \dots & & & \dots & & \\ d_i & & \dots & c(d_i, w_j) & \dots & \\ \dots & & & \dots & & \\ d_I & & & & & \end{array}$$

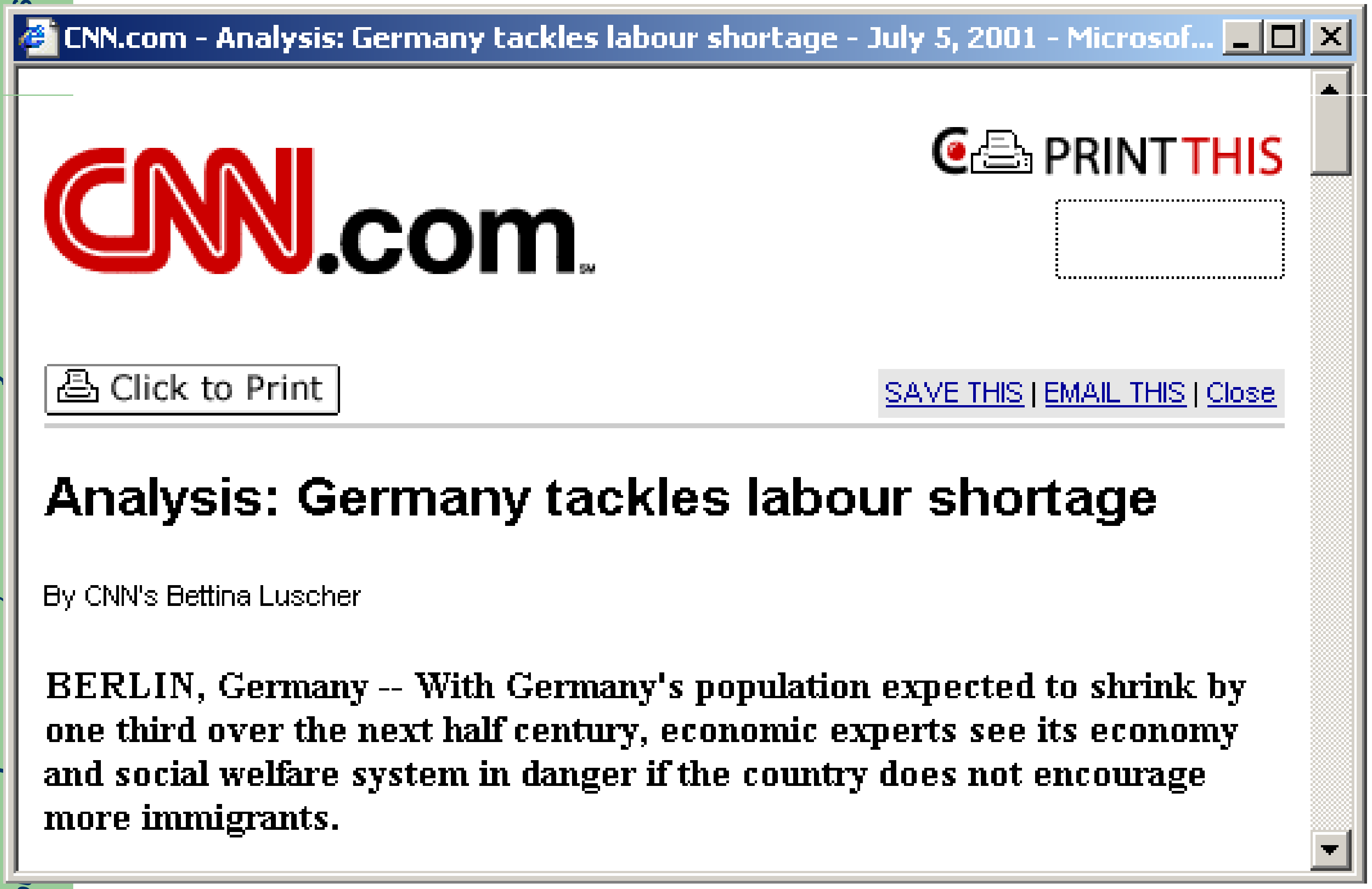
A 100 Millionths of a Typical Document-term Matrix

Typical:

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



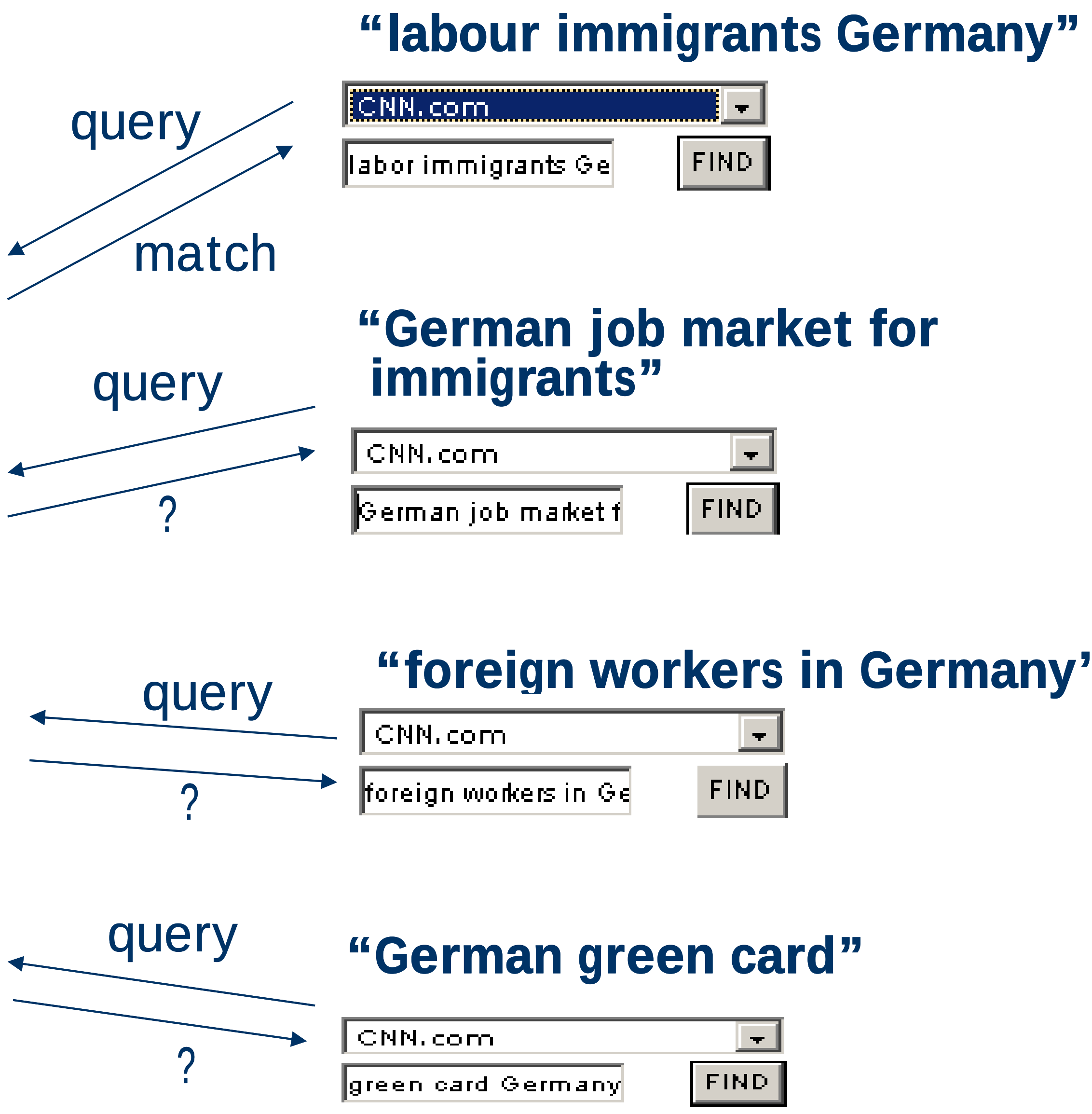
Vocabulary Mismatch & ~~Robustness~~


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.



"labour immigrants Germany"

query

match

CNN.com

labor immigrants Ge

FIND

"German job market for immigrants"

query

?

CNN.com

German job market f

FIND

"foreign workers in Germany"

query

?

CNN.com

foreign workers in Ge

FIND

"German green card"

query

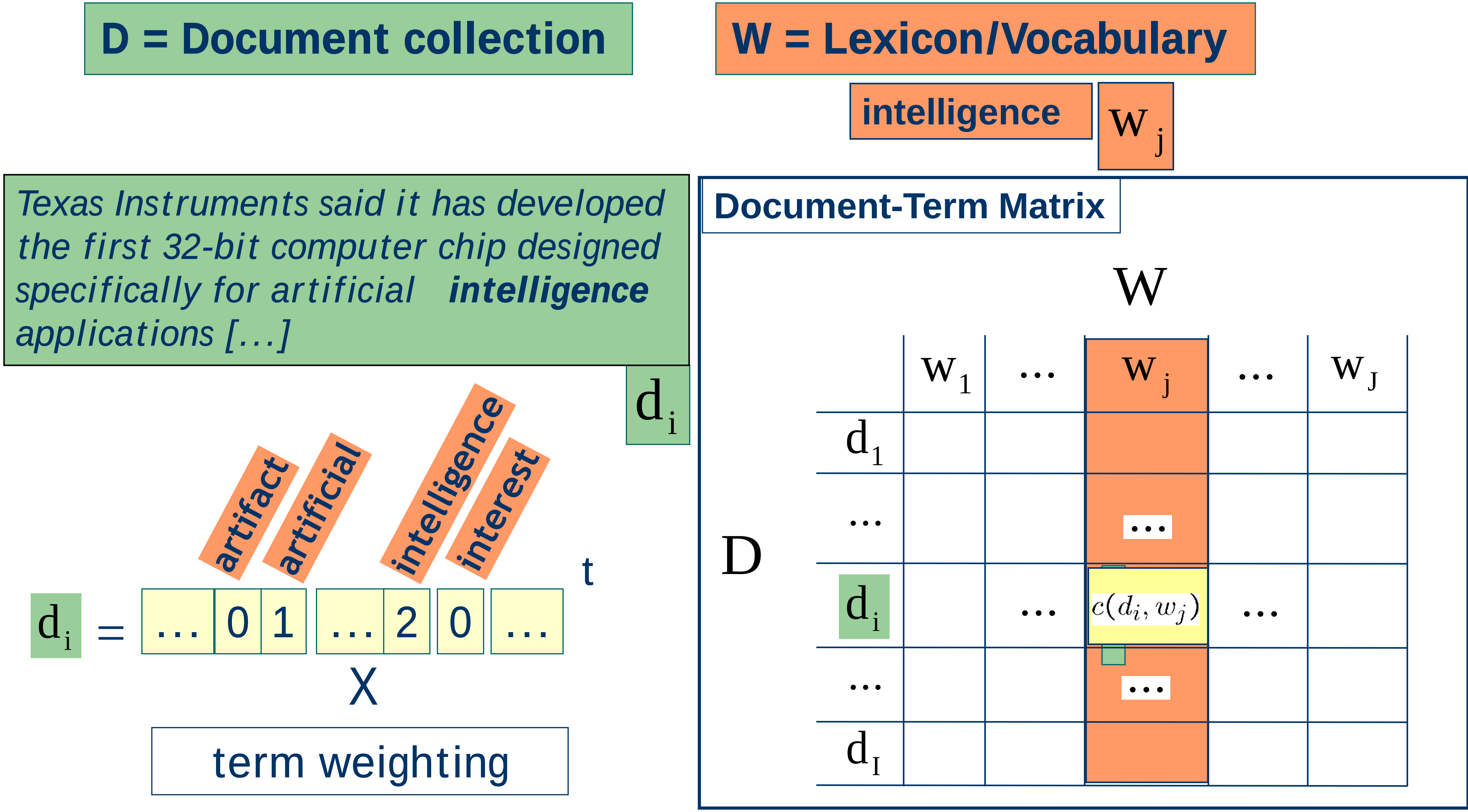
?

CNN.com

green card Germany

FIND

Document-Term Matrix



D = Document collection

W = Lexicon/Vocabulary

intelligence w_j

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

d_i

artifact

artificial

intelligence

interest

$$d_i = (\ldots\ 0\ 1\ \ldots\ 2\ 0\ \ldots)^t$$

X

term weighting

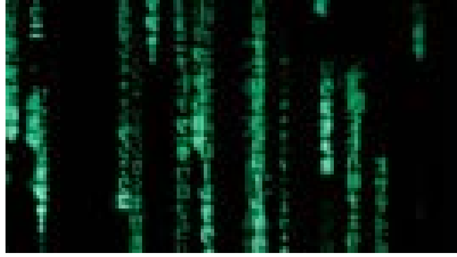
Document-Term Matrix

W

$$D \quad \begin{array}{c|ccccc} & w_1 & \cdots & w_j & \cdots & w_J \\ \hline d_1 & & & & & \\ \cdots & & & \cdots & & \\ d_i & & \cdots & c(d_i, w_j) & \cdots & \\ \cdots & & & \cdots & & \\ d_I & & & & & \end{array}$$

Clustering rows?

Clustering columns?



Latent Structure

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

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

- 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?

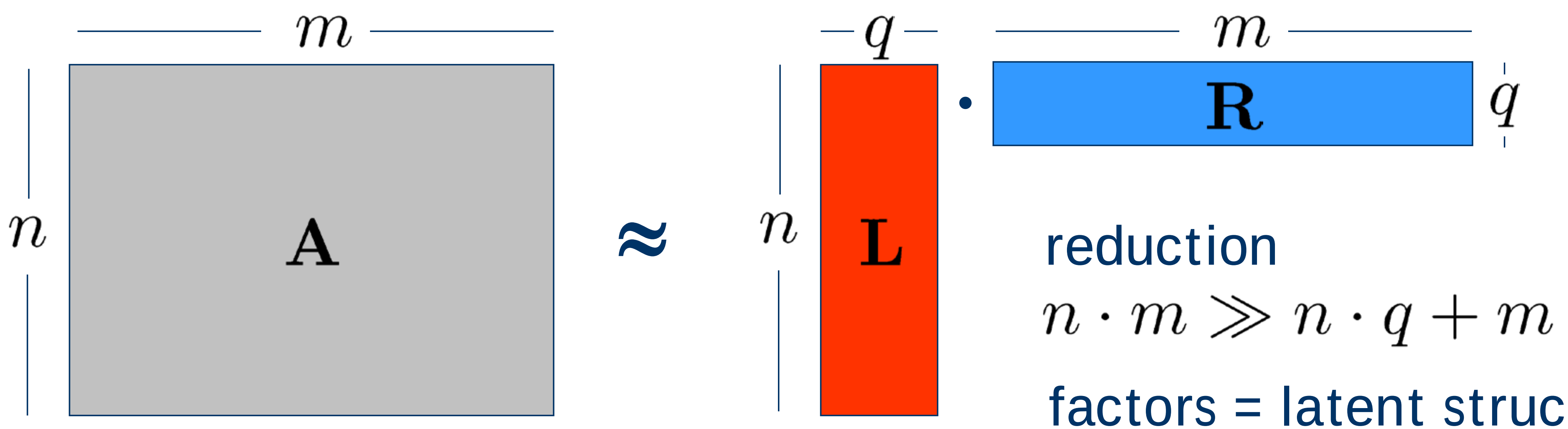
Matrix Decomposition

- 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**"



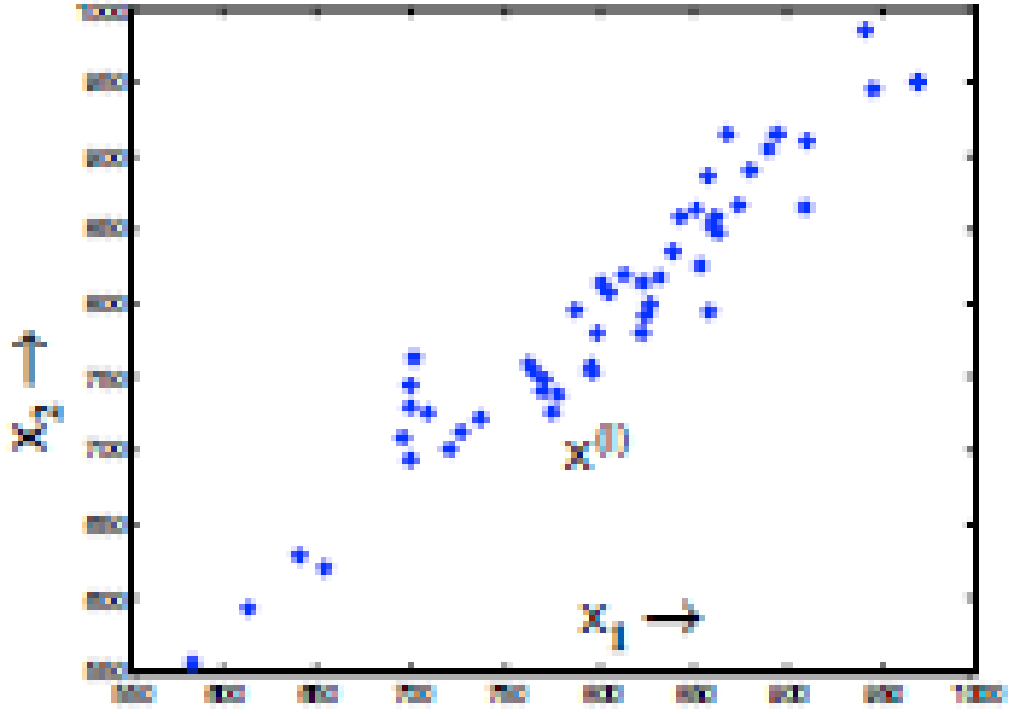
reduction

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

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: $[x_1,x_2] \sim f(z_1)$

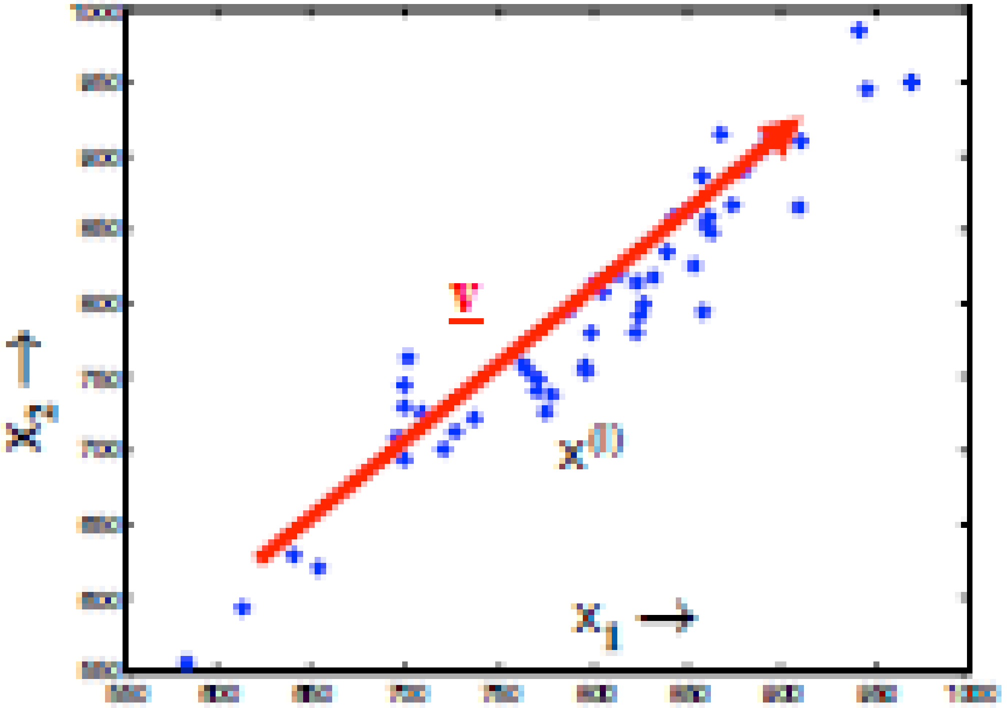


Slides by Alexander Ihler

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: $[x_1,x_2] \sim f(z_1)$

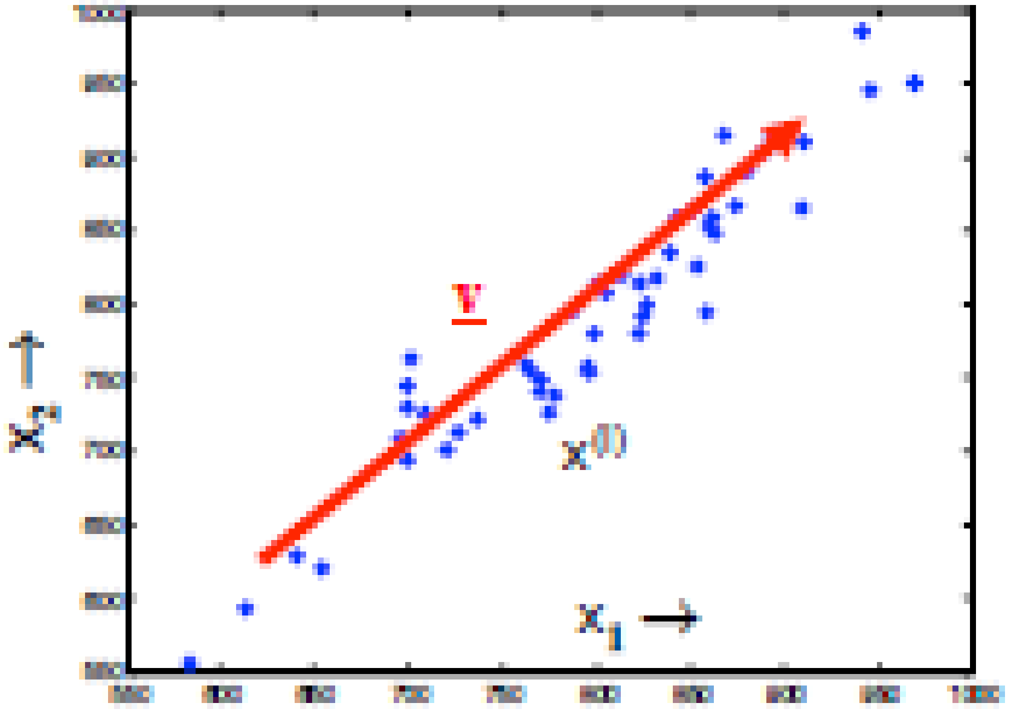
- Ex: linear function f(z): $[x_1,x_2] = z * \underline{v} = z * [v_1,v_2]$
- $\underline{v}$ is the same for all data points (communicate once)
- z tells us the closest point on v to the original point $[x_1,x_2]$

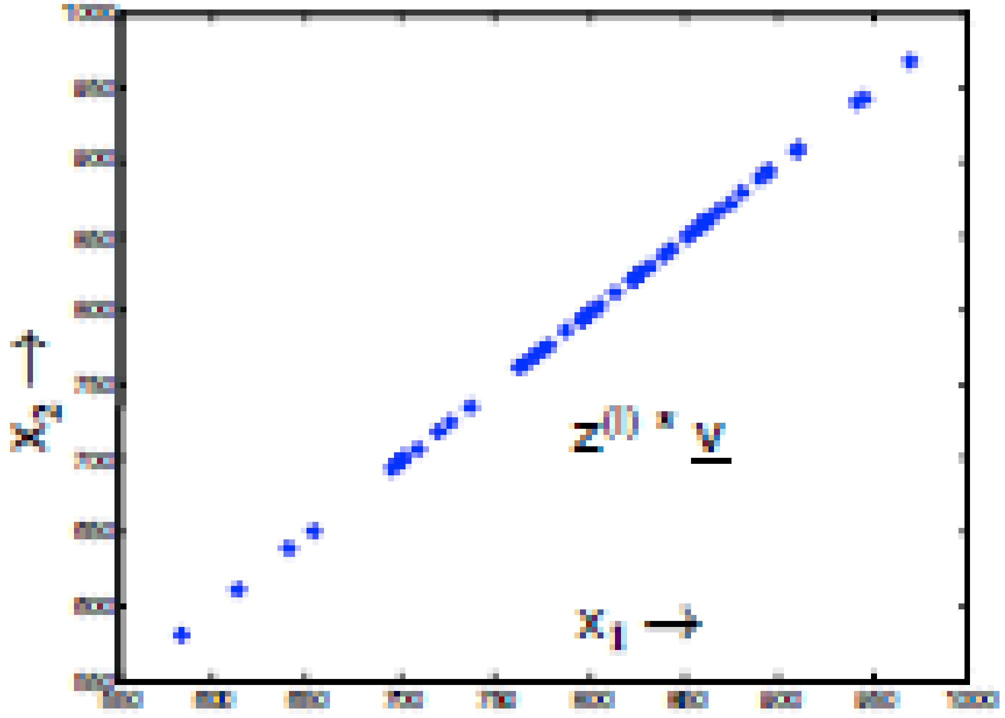


Dimensionality reduction

- 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 $[x_1,x_2]$



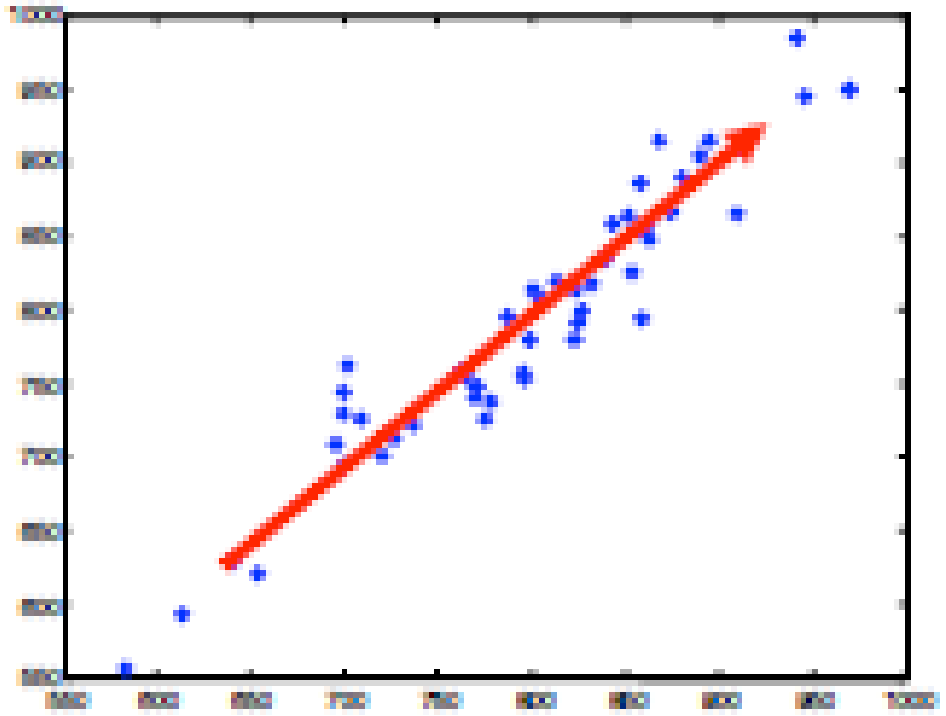


Principal Components Analysis

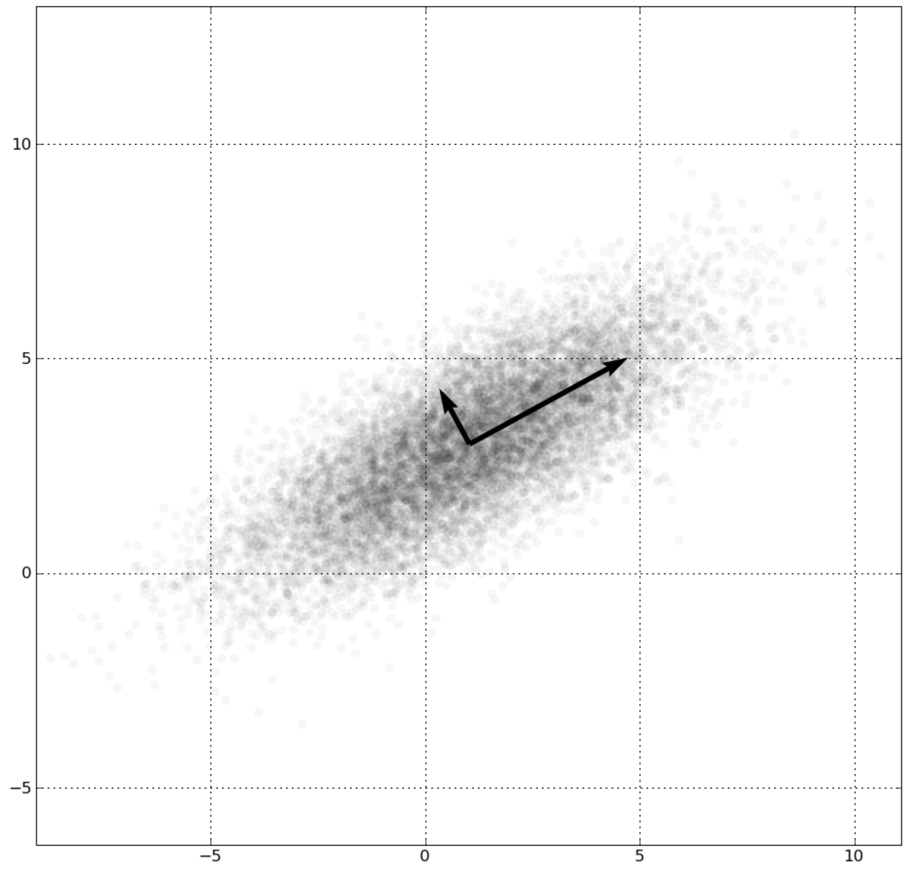
- What is the vector that would most closely reconstruct X?

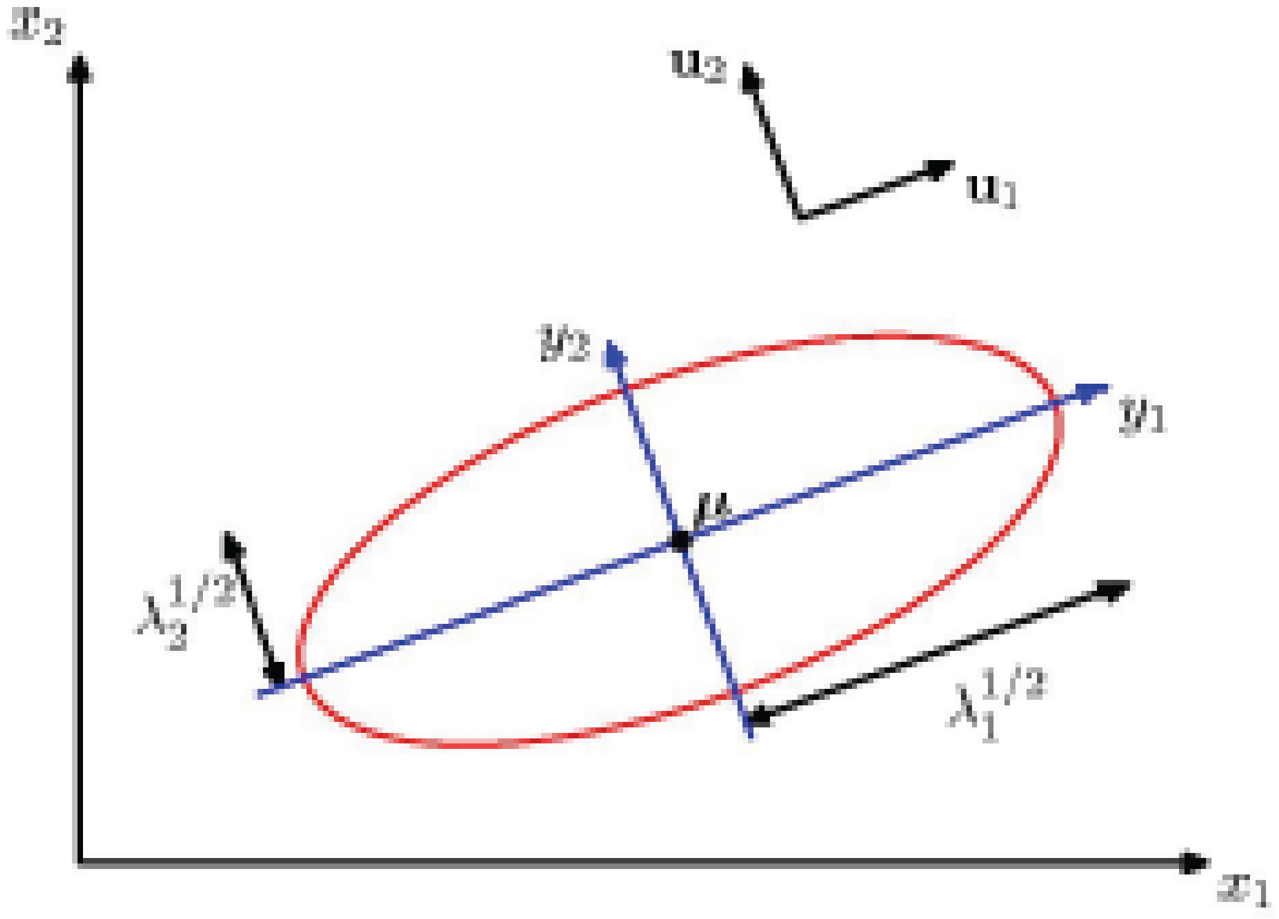
$$\min_{a,v} \sum_i (x^{(i)} - a^{(i)} v)^2$$

  - 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

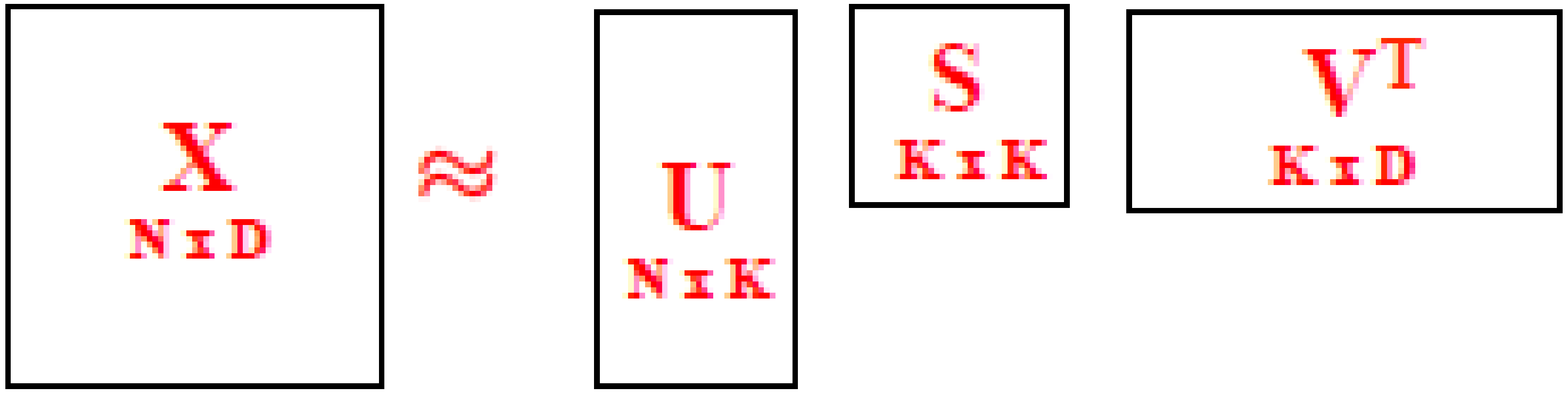
Given pattern matrix X,

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

C = VDV^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

$$X = USV^T$$

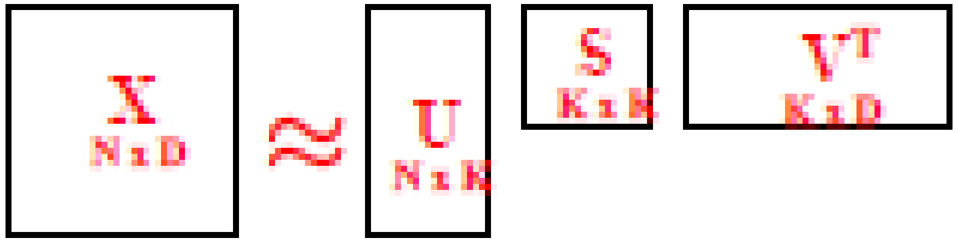
- XX^T and X^TX share the same eigenvalues
- Even better: their eigenvectors are related
  - Xv_i is an eigenvector of XX^T

Collaborative Filtering (Netflix)

From Y. Koren
of BellKor team

users

movies	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	



$$\underset{N\times D}{X} \approx \underset{N\times K}{U}\ \underset{K\times K}{S}\ \underset{K\times D}{V^T}$$

From Y. Koren
of BellKor team

Latent Space Models

Model ratings matrix as "user" and "movie" positions

Infer values from known ratings

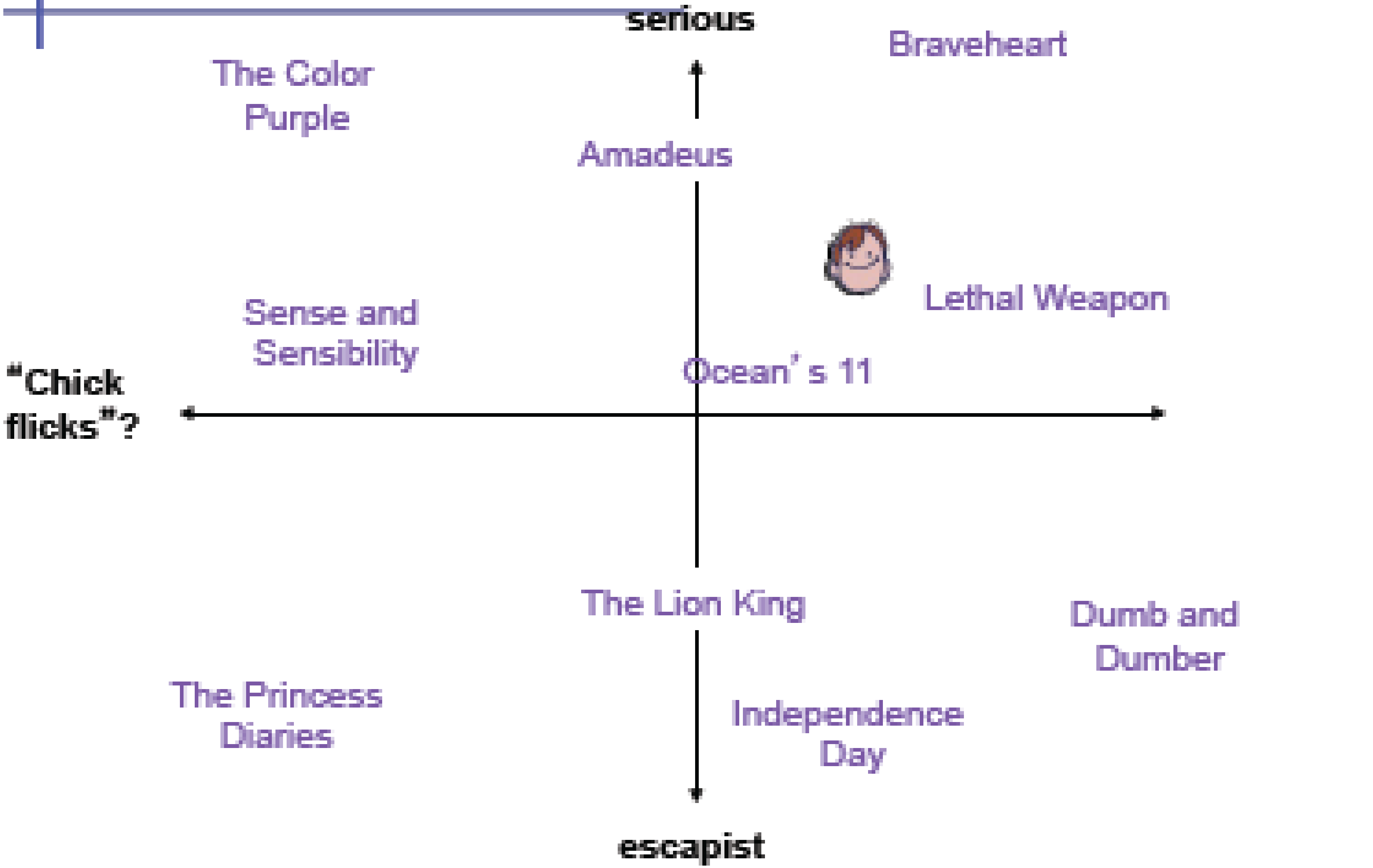
Extrapolate to unranked

$$\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c|}
 & \multicolumn{12}{c}{\text{users}} \\ \hline
\text{items} & 1 & & 3 & & & 5 & & & 5 & & 4 & \\ \hline
 & & & 5 & 4 & & & 4 & & & 2 & 1 & 3 \\ \hline
 & 2 & 4 & & 1 & 2 & & 3 & & 4 & 3 & 5 & \\ \hline
 & & 2 & 4 & & 5 & & & 4 & & & 2 & \\ \hline
 & & & 4 & 3 & 4 & 2 & & & & & 2 & 5 \\ \hline
 & 1 & & 3 & & 3 & & & 2 & & & 4 & \\ \hline
\end{array} \sim$$

$$\sim \ \text{items} \begin{array}{|c|c|c|} \hline
.1 & -.4 & .2 \\ \hline
-.5 & .6 & .5 \\ \hline
-.2 & .3 & .5 \\ \hline
1.1 & 2.1 & .3 \\ \hline
-.7 & 2.1 & -2 \\ \hline
-1 & .7 & .3 \\ \hline
\end{array} \bullet
\begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|}
\multicolumn{12}{c}{\text{users}} \\ \hline
1.1 & -.2 & .3 & .5 & -2 & -.5 & .8 & -.4 & .3 & 1.4 & 2.4 & -.9 \\ \hline
-.8 & .7 & .5 & 1.4 & .3 & -1 & 1.4 & 2.9 & -.7 & 1.2 & -.1 & 1.3 \\ \hline
2.1 & -.4 & .6 & 1.7 & 2.4 & .9 & -.3 & .4 & .8 & .7 & -.6 & .1 \\ \hline
\end{array}$$

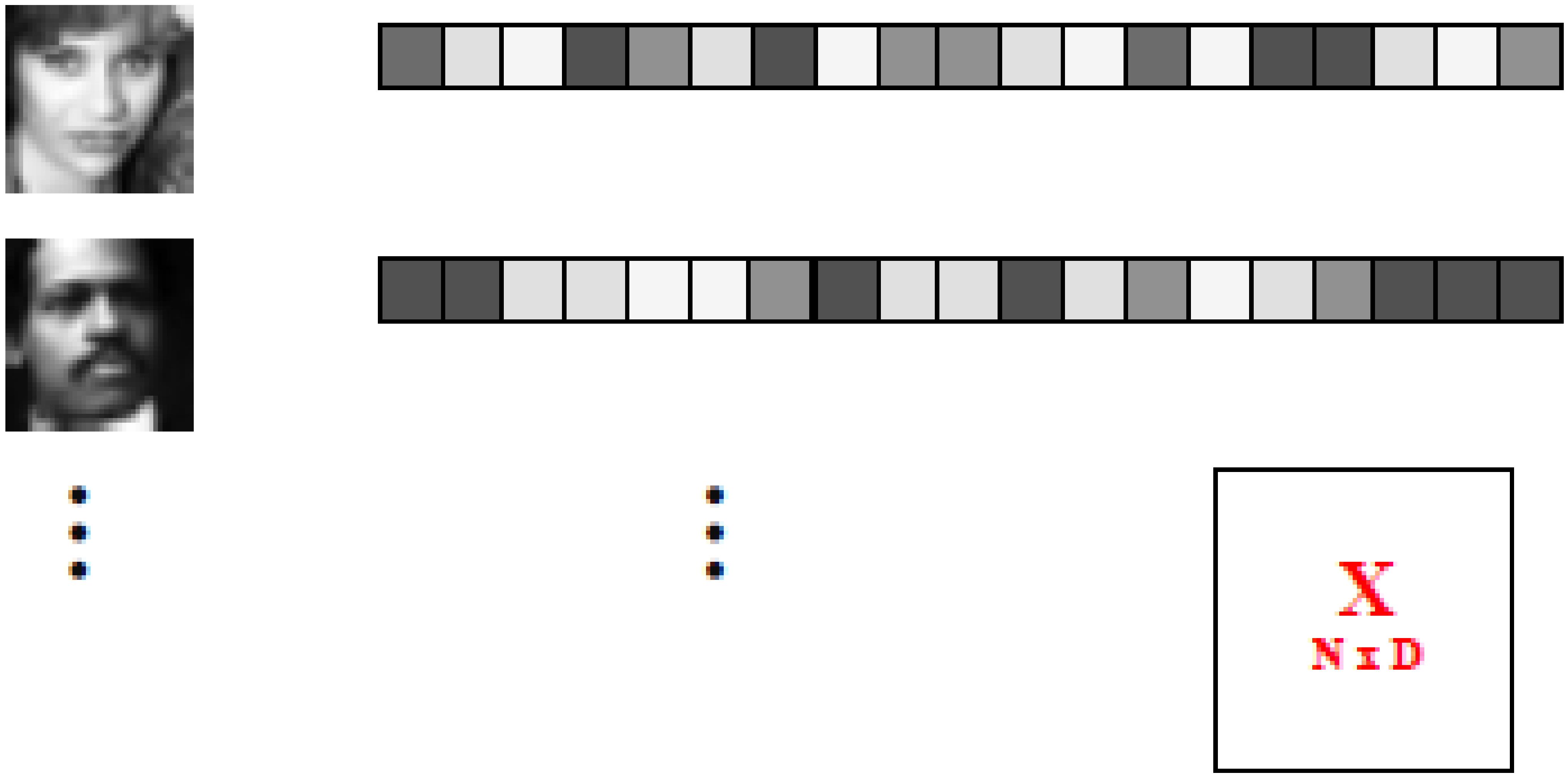
From Y. Koren of BellKor team

Latent Space Models



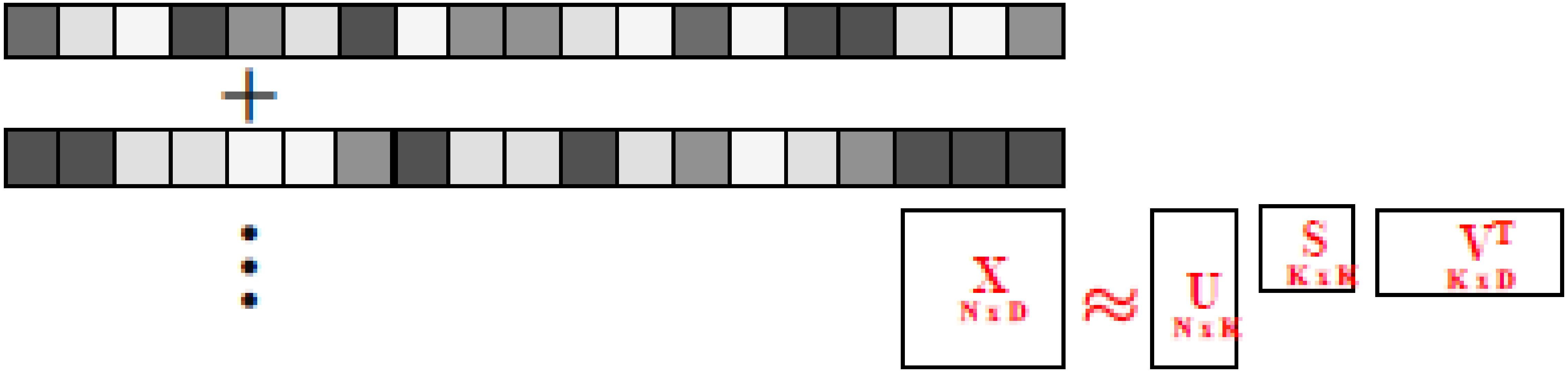
“Eigen-faces”

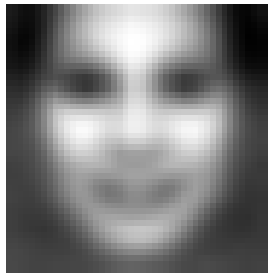
- “Eigen-X” = represent X using PCA
- Ex: Viola Jones data set
  - 24x24 images of faces = 576 dimensional measurements

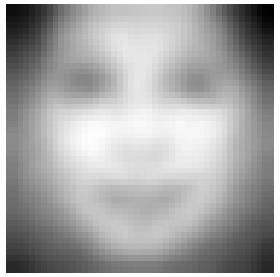


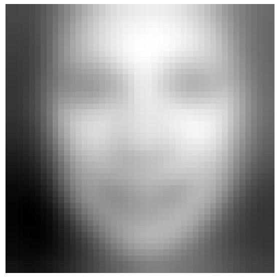
"Eigen-faces"

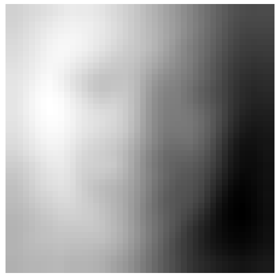
- "Eigen-X" = represent X using PCA
- Ex: Viola Jones data set
  - 24x24 images of faces = 576 dimensional measurements
  - Take first K PCA components

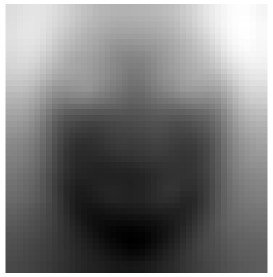



Mean


V(1,:)


V(2,:)

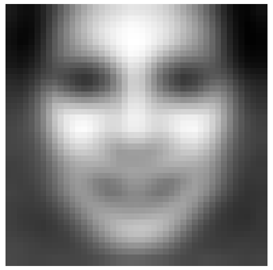

V(3,:)

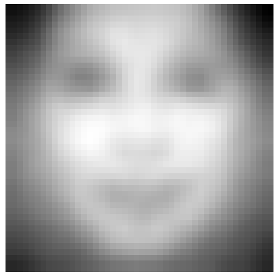

V(4,:)

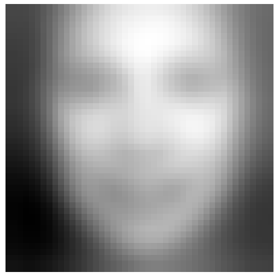
…

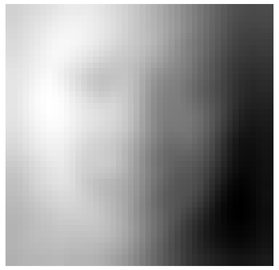
“Eigen-faces”

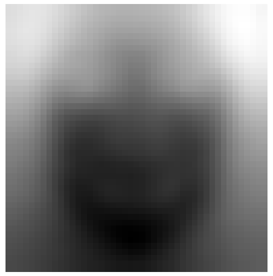
- “Eigen-X” = represent X using PCA
- Ex: Viola Jones data set
  - 24x24 images of faces = 576 dimensional measurements
  - Take first K PCA components


Mean

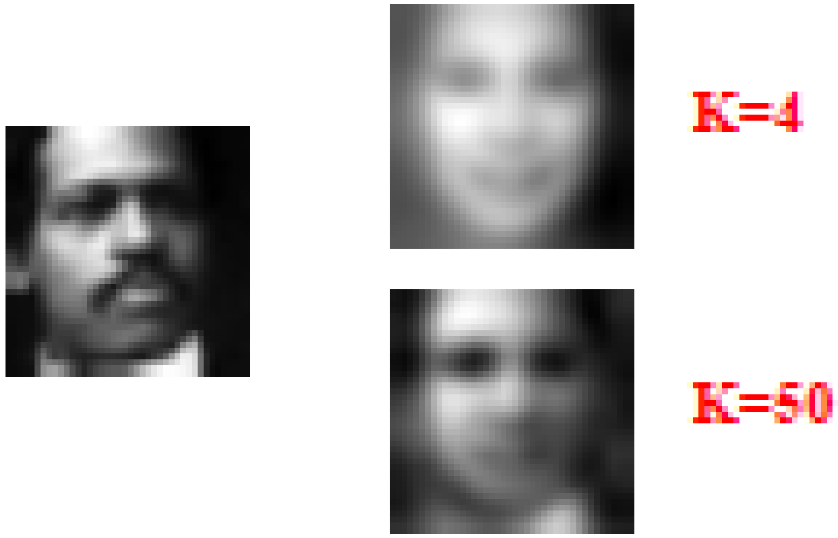

Dir 1

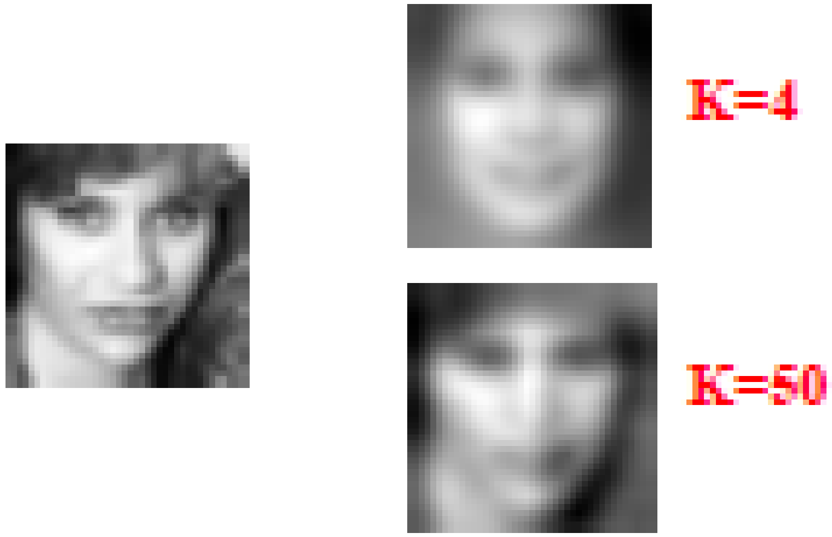

Dir 2


Dir 3

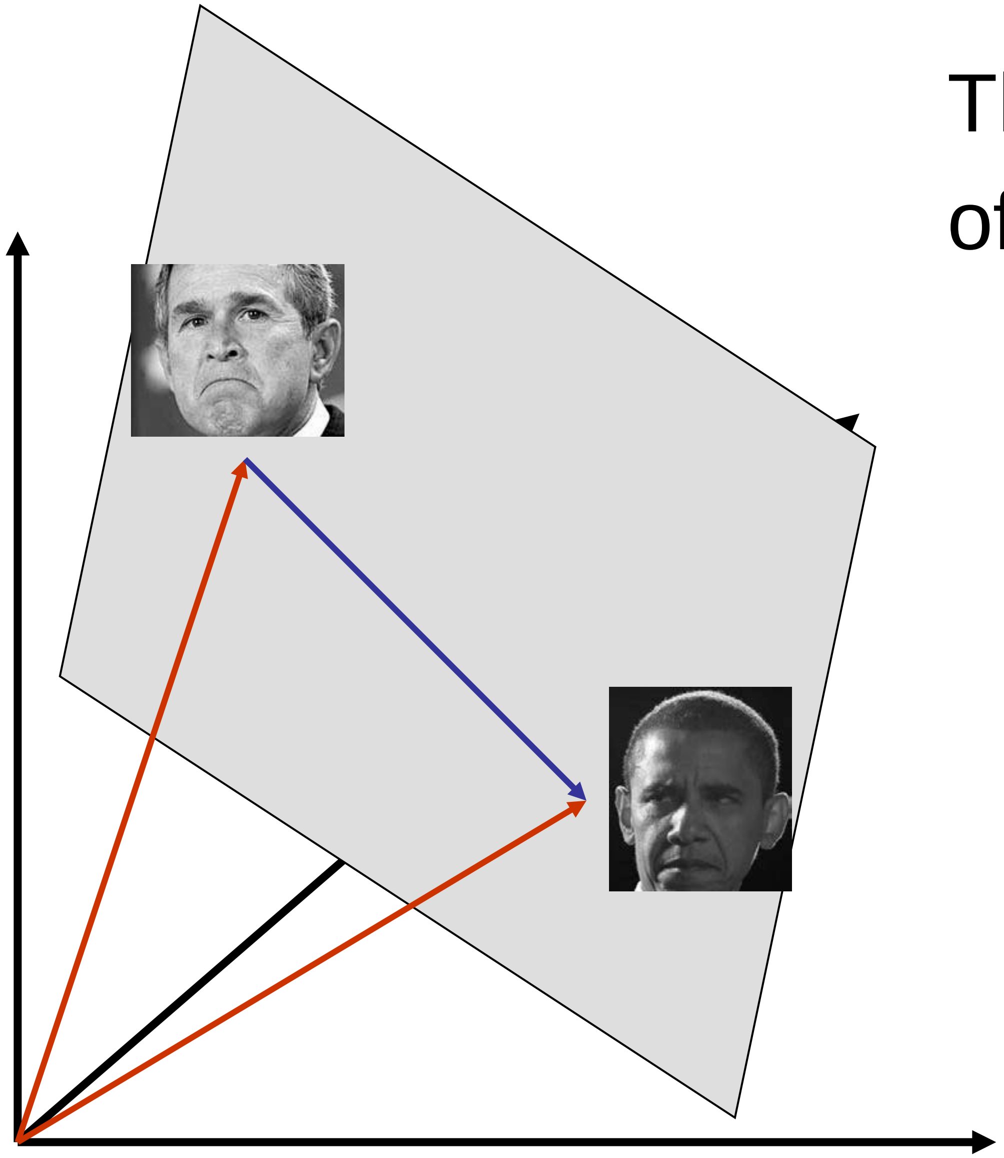

Dir 4

…





The Face Subspace



The set of faces is a “subspace” of the set of images

- Suppose it is K dimensional
- We can find the best subspace using PCA
- This is like fitting a “hyper-plane” to the set of faces
  - spanned by vectors $\mathbf{v_1}$, $\mathbf{v_2}$, ..., $\mathbf{v_K}$

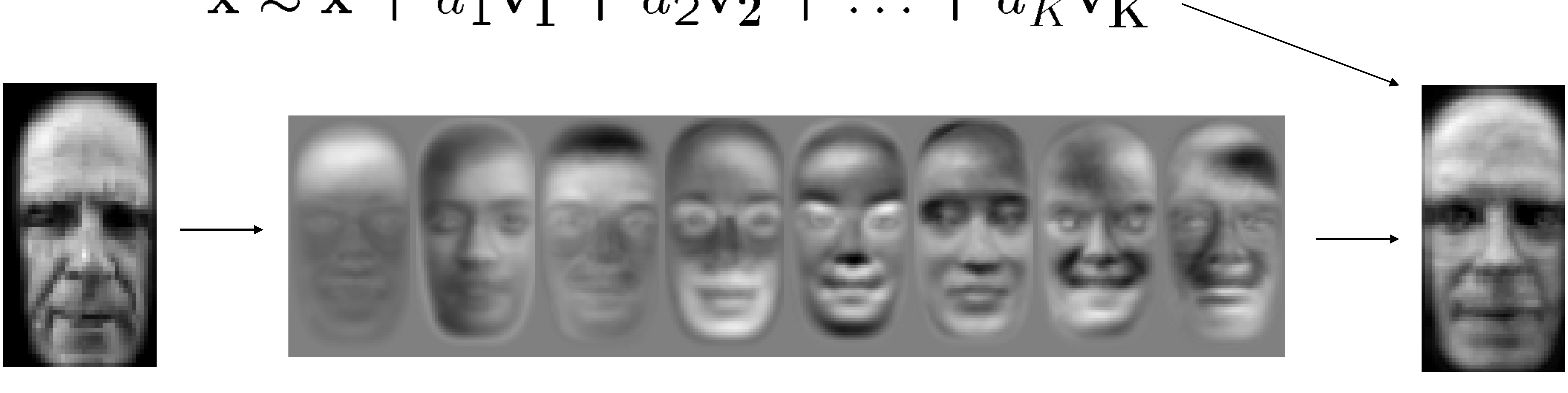
Any face: $\mathbf{x} \approx \overline{\mathbf{x}} + a_1\mathbf{v_1} + a_2\mathbf{v_2} + \ldots + a_k\mathbf{v_k}$

Projecting onto the Eigenface Subspace

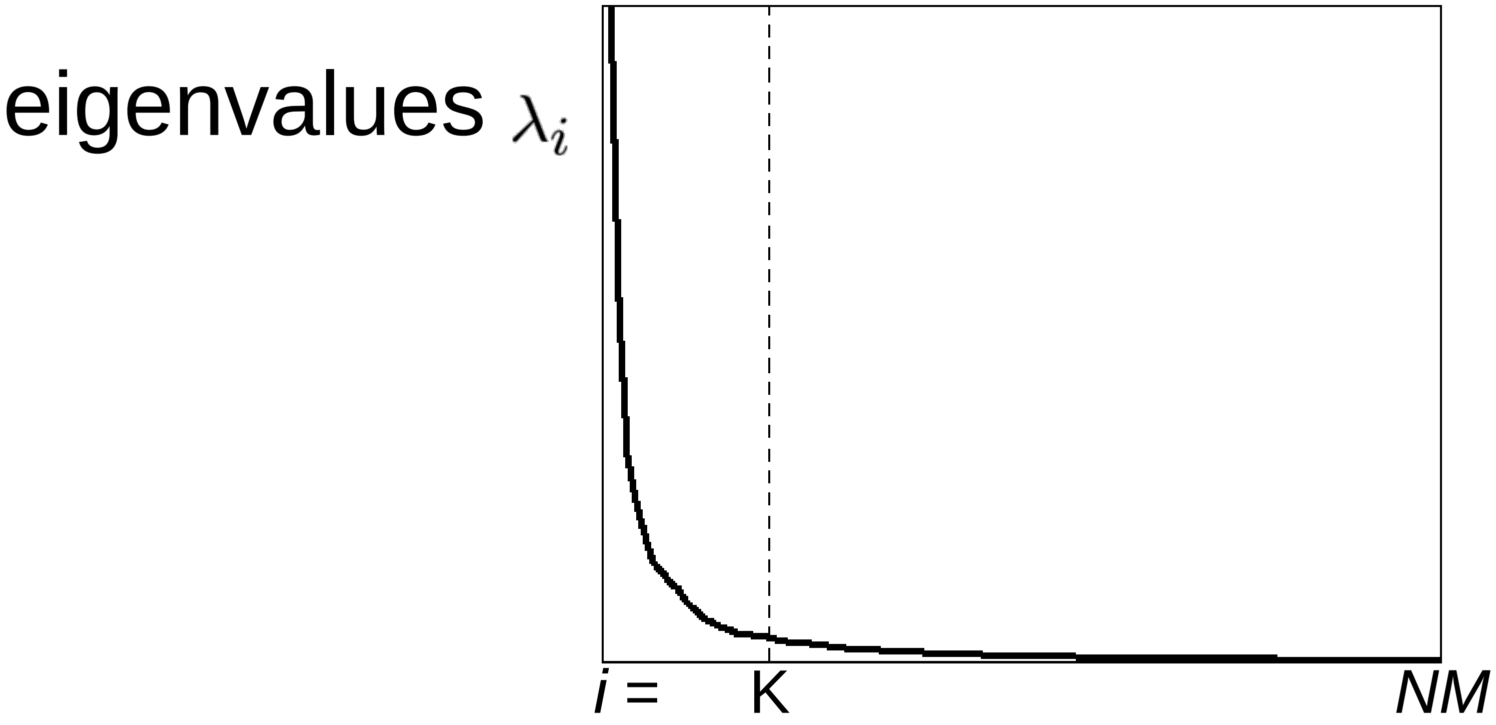
- The eigenfaces $\mathbf{v}_1$, ..., $\mathbf{v}_K$ span the space of faces
  - A face is converted to eigenface coordinates by

$$\mathbf{x} \rightarrow (\underbrace{(\mathbf{x}-\overline{\mathbf{x}})\cdot\mathbf{v_1}}_{a_1},\ \underbrace{(\mathbf{x}-\overline{\mathbf{x}})\cdot\mathbf{v_2}}_{a_2}, \ldots,\ \underbrace{(\mathbf{x}-\overline{\mathbf{x}})\cdot\mathbf{v_K}}_{a_K})$$

$$\mathbf{x} \approx \overline{\mathbf{x}} + a_1\mathbf{v_1} + a_2\mathbf{v_2} + \ldots + a_K\mathbf{v_K}$$



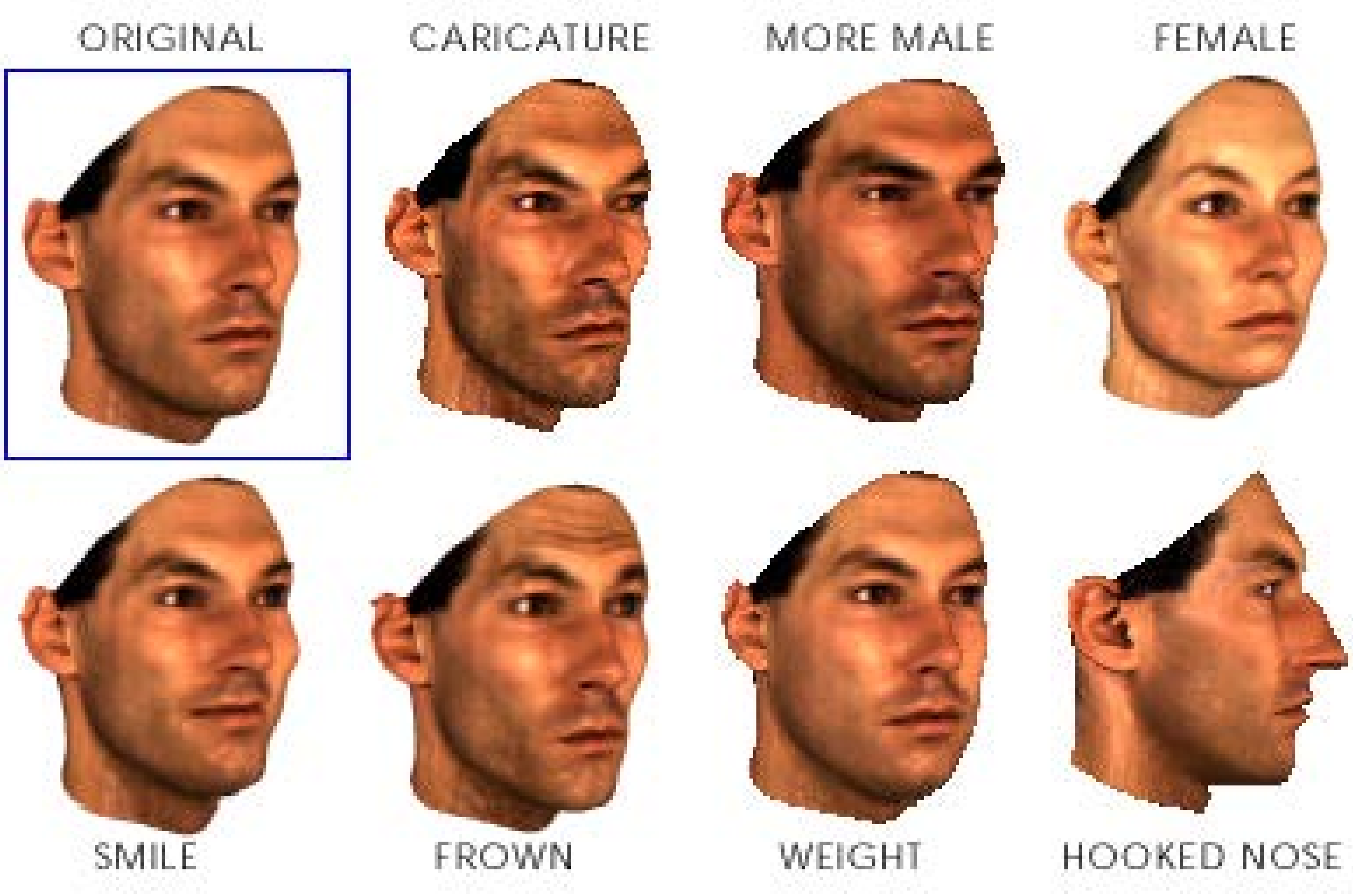
Choosing the Dimension K



- How many eigenfaces to use?

- Look at the decay of the eigenvalues
  - the eigenvalue tells you the amount of variance “in the direction” of that eigenface
  - ignore eigenfaces with low variance

PCA with depth data: Blinz & Vetter, 1999

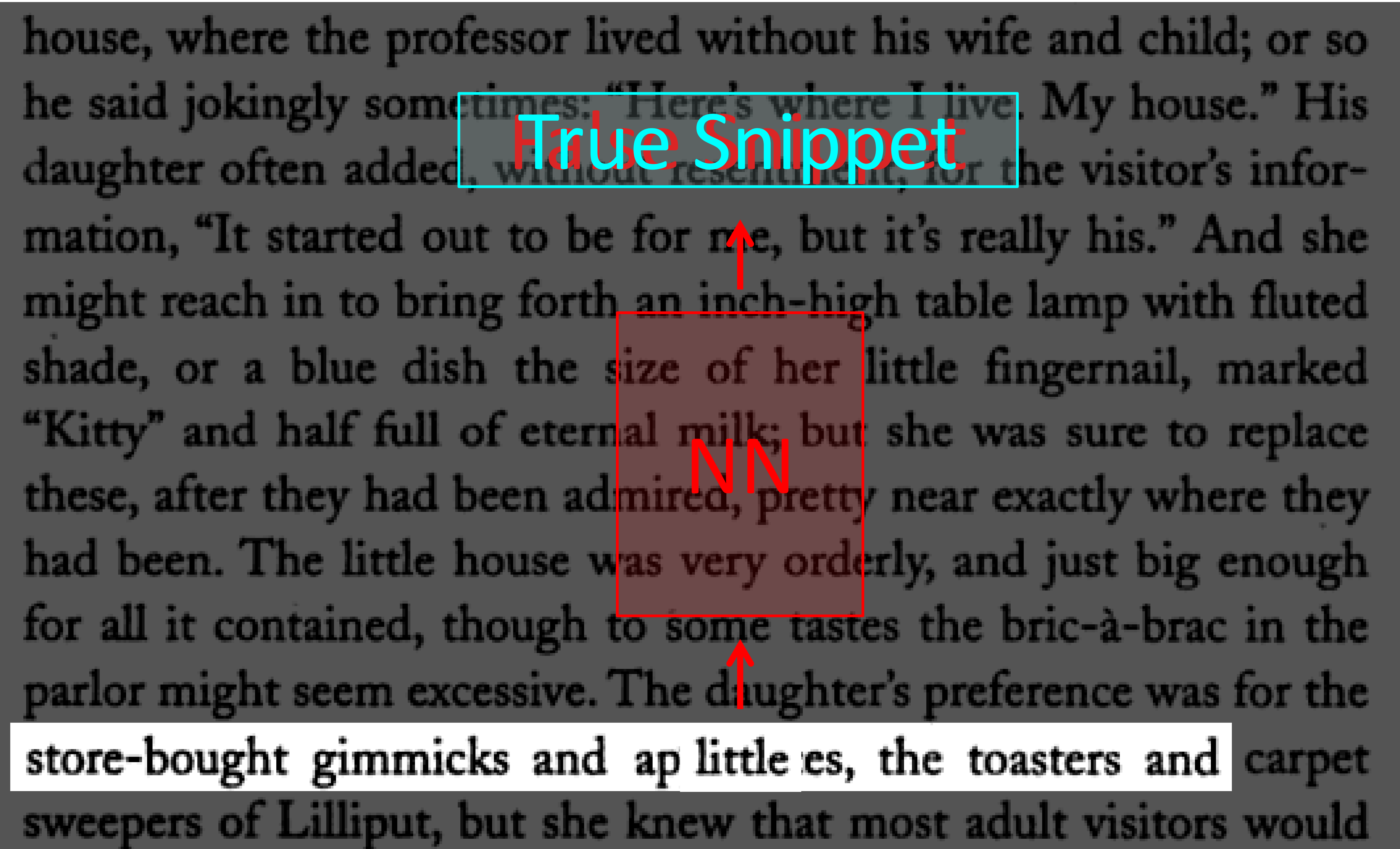


http://www.youtube.com/watch?v=jrutZaYoQJo

Non-linear Dimensionality Reduction

Learn an embedding ("self-supervision")

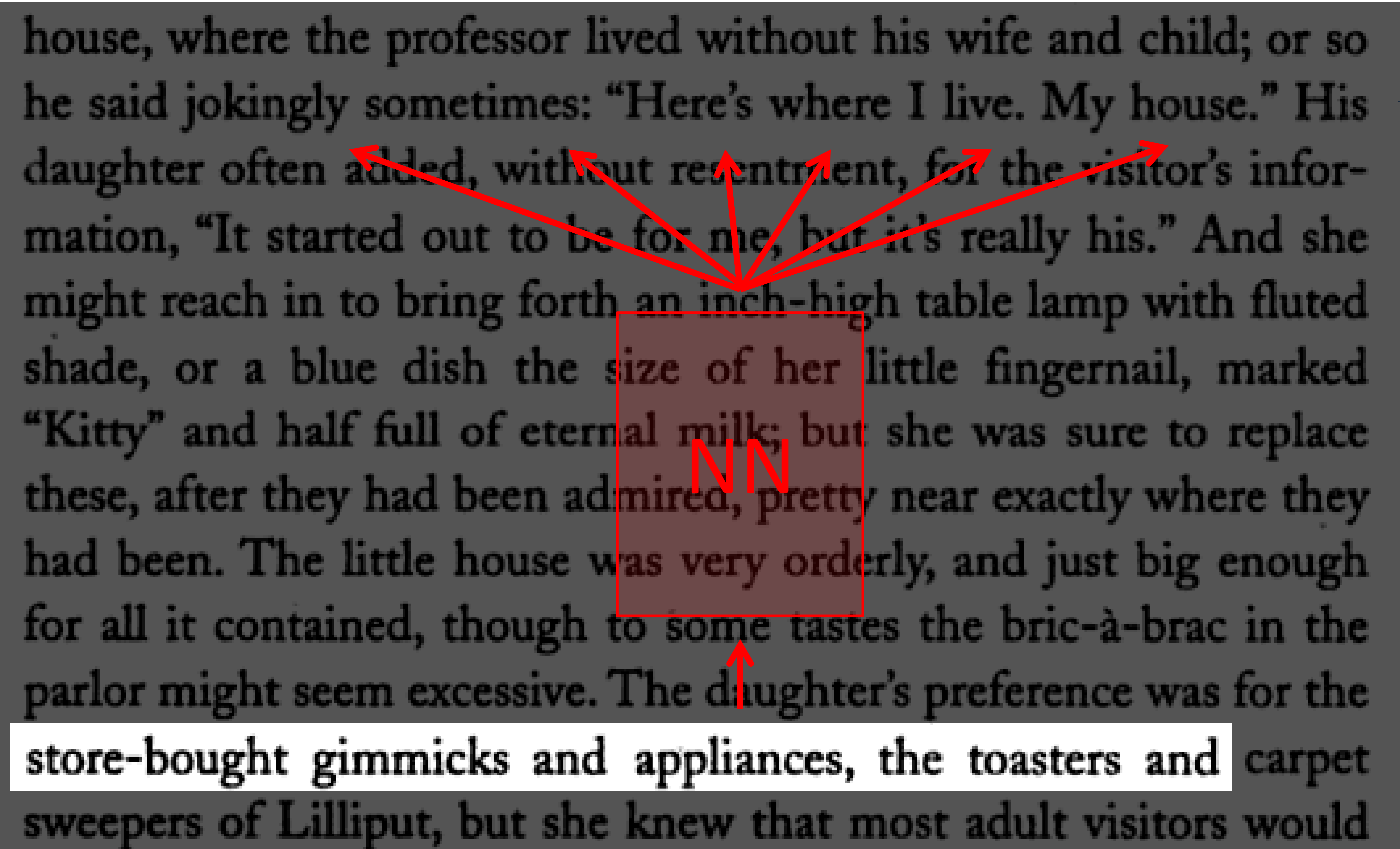
[Collobert and Weston 2008]



house, where the professor lived without his wife and child; or so he said jokingly sometimes: "Here's where I live. My house." His daughter often added, without resentment, for the visitor's information, "It started out to be for me, but it's really his." And she might reach in to bring forth an inch-high table lamp with fluted shade, or a blue dish the size of her little fingernail, marked "Kitty" and half full of eternal milk; but she was sure to replace these, after they had been admired, pretty near exactly where they had been. The little house was very orderly, and just big enough for all it contained, though to some tastes the bric-à-brac in the parlor might seem excessive. The daughter's preference was for the store-bought gimmicks and ap little es, the toasters and carpet sweepers of Lilliput, but she knew that most adult visitors would

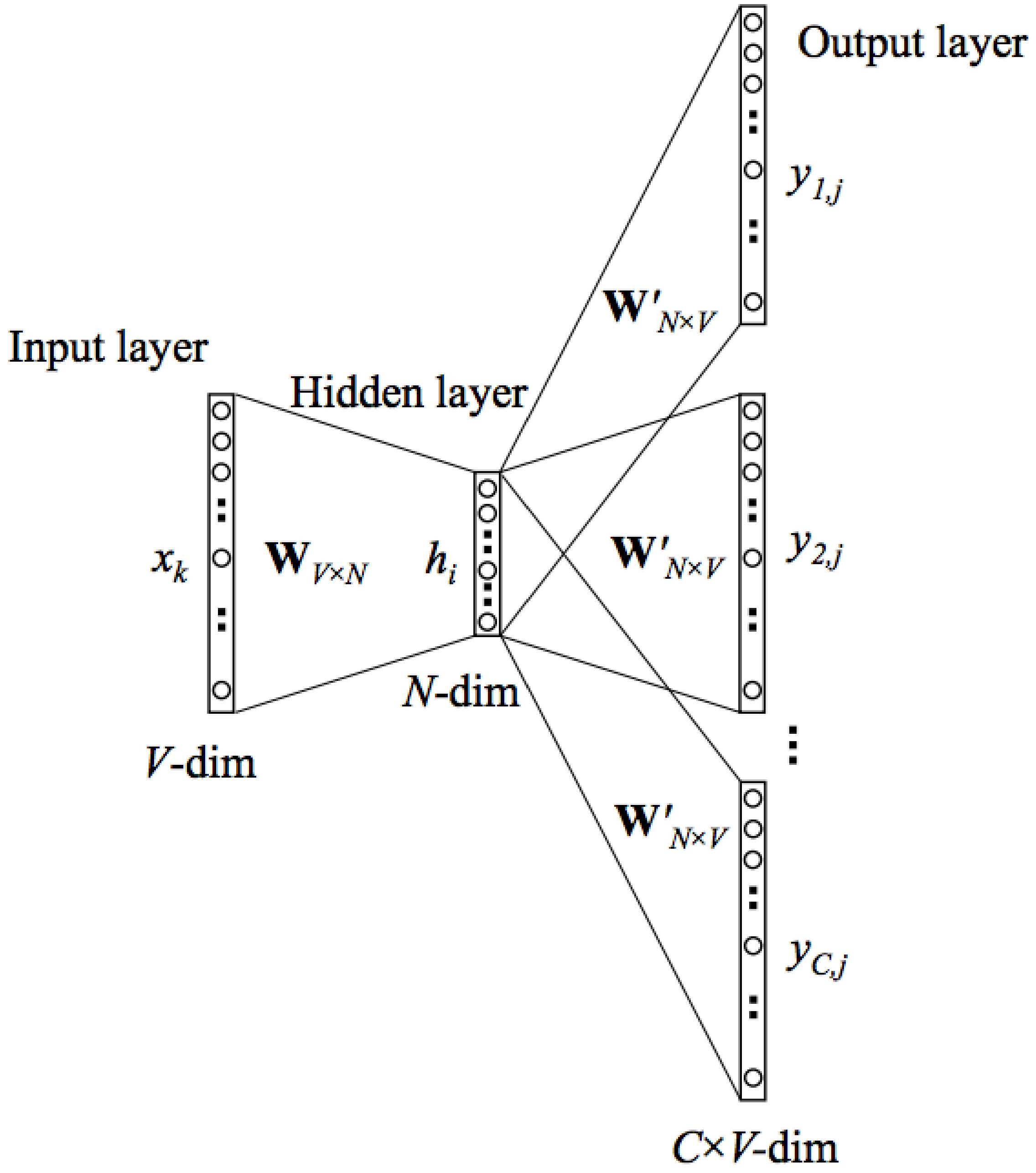
Skip Gram (word2vec)

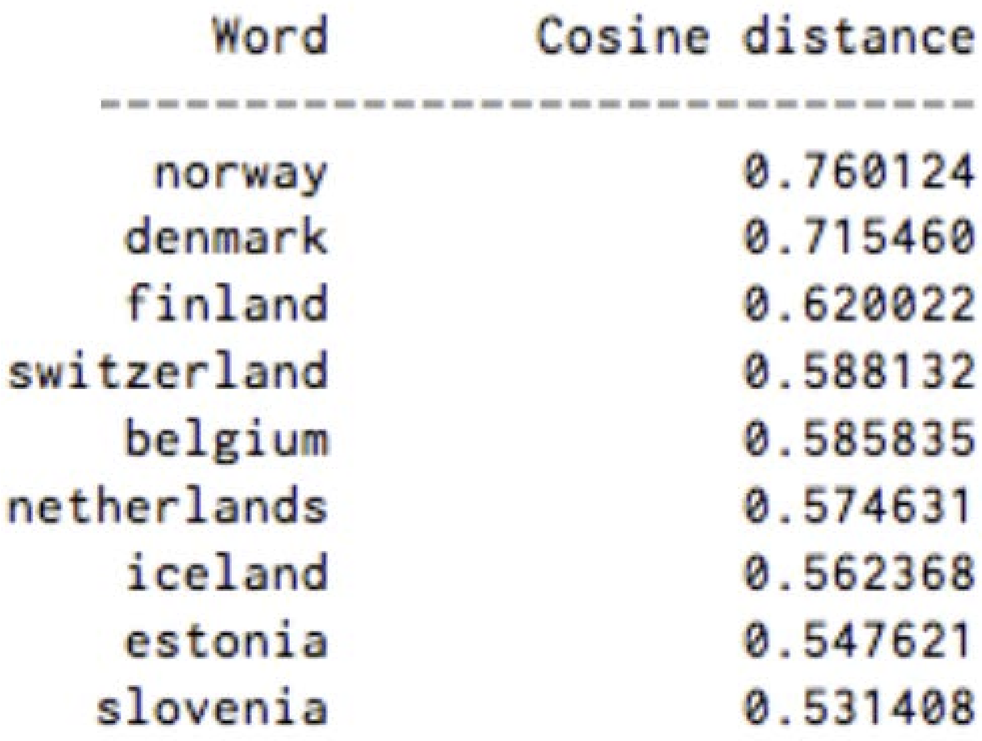
[Mikolov et al. 2013]



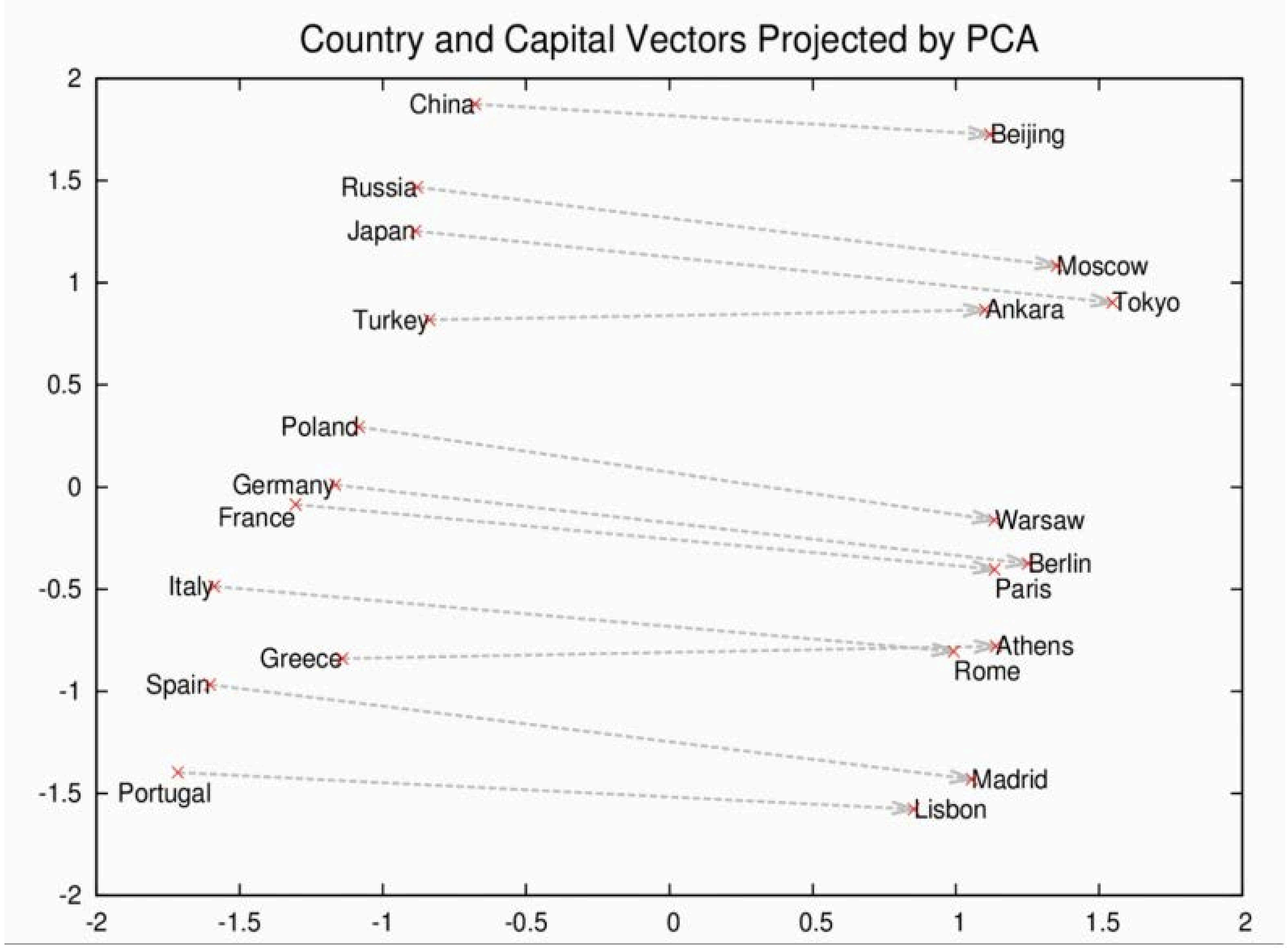
house, where the professor lived without his wife and child; or so he said jokingly sometimes: "Here's where I live. My house." His daughter often added, without resentment, for the visitor's information, "It started out to be for me, but it's really his." And she might reach in to bring forth an inch-high table lamp with fluted shade, or a blue dish the size of her little fingernail, marked "Kitty" and half full of eternal milk; but she was sure to replace these, after they had been admired, pretty near exactly where they had been. The little house was very orderly, and just big enough for all it contained, though to some tastes the bric-à-brac in the parlor might seem excessive. The daughter's preference was for the store-bought gimmicks and appliances, the toasters and carpet sweepers of Lilliput, but she knew that most adult visitors would

Learning word2vec embeddings



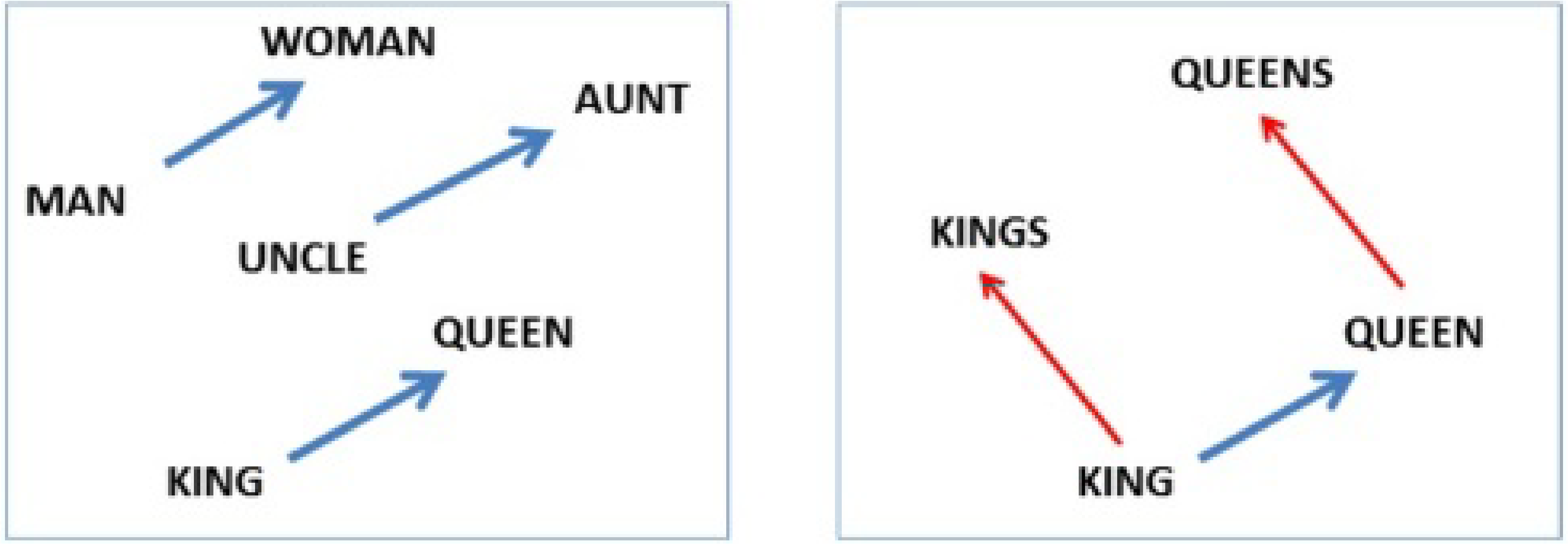


```
         Word       Cosine distance
    ------------------------------------
       norway              0.760124
      denmark              0.715460
      finland              0.620022
  switzerland              0.588132
      belgium              0.585835
  netherlands              0.574631
      iceland              0.562368
      estonia              0.547621
     slovenia              0.531408
```



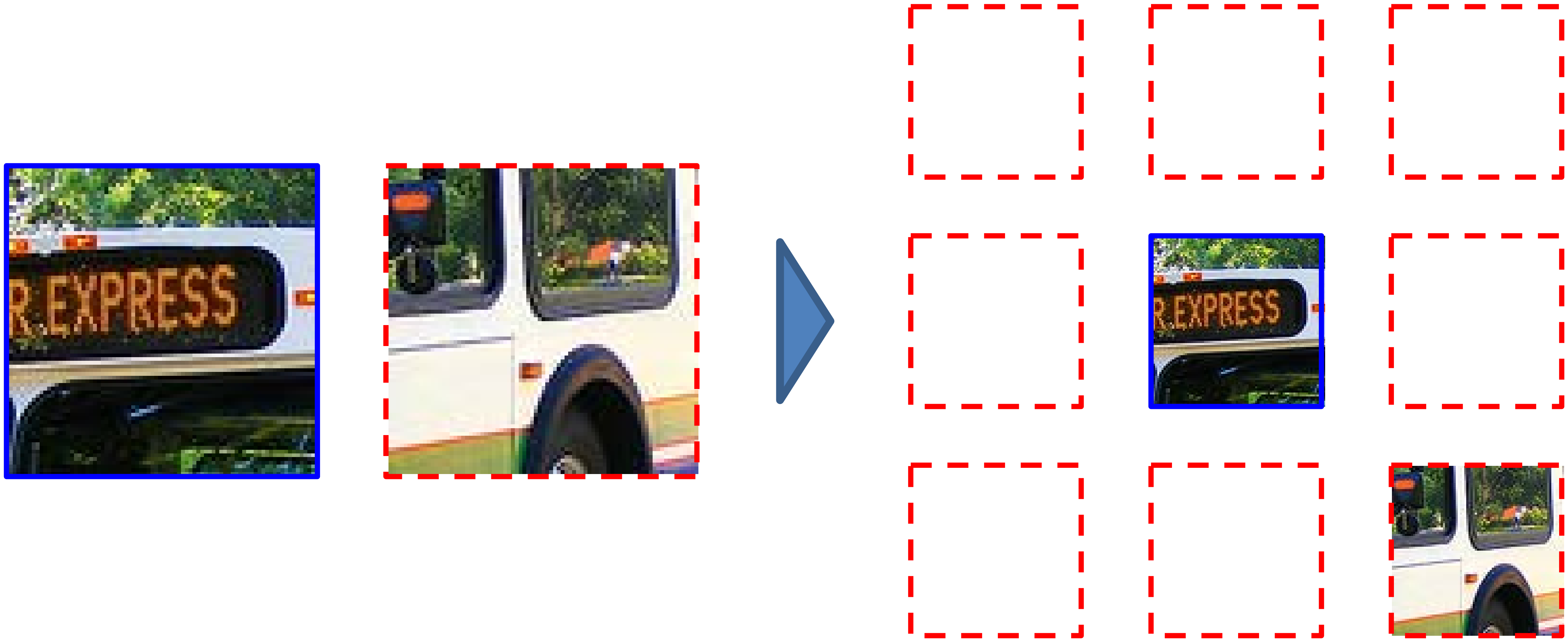
Example

vec(“man”) – vec(“king”) + vec(“woman”) = vec(“queen”)



http://deeplearning4j.org/word2vec.html

Visual Context Task



Carl Doersch, Abhinav Gupta, and Alexei A. Efros. **Unsupervised Visual Representation Learning by Context Prediction.** In *ICCV 2015*