Machine Learning Department
School of Computer Science
Carnegie Mellon University

Deriving Principal Component Analysis (PCA)

Matt Gormley
Lecture 11
Oct. 3, 2018

Reminders

- Quiz 1: Linear Algebra (today)
- Homework 3: Matrix Calculus + Probability
  - Out: Wed, Oct. 3
  - Due: Wed, Oct. 10 at 11:59pm
- Quiz 2: Matrix Calculus + Probability
  - In-class, Wed, Oct. 10

Q&A

DIMENSIONALITY REDUCTION

PCA Outline

- **Dimensionality Reduction**
  - High-dimensional data
  - Learning (low dimensional) representations
- **Principal Component Analysis (PCA)**
  - Examples: 2D and 3D
  - Data for PCA
  - PCA Definition
  - Objective functions for PCA
  - PCA, Eigenvectors, and Eigenvalues
  - Algorithms for finding Eigenvectors / Eigenvalues
- **PCA Examples**
  - Face Recognition
  - Image Compression

High Dimension Data

Examples of high dimensional data:

– High resolution images (millions of pixels)

High Dimension Data

Examples of high dimensional data:

– Multilingual News Stories
(vocabulary of hundreds of thousands of words)

Hilo Tribune.

KILAUEA CRATER CONTINUES IN ERUPTION

FOUR DAYS OF ACTIVITY

ANOTHER OUTBREAK OCCURS

HONOLULU GETS APPROPRIATIONS FOR HARBOR AND LIGHTHOUSE

River and Harbor Bill Passes.

Postmaster Pratt Confirmed.

Latest Sugar Quotations.

Fire At New Orleans.

Dr. Grace and Irwin

Chas. M. LeBlond

Ridgway & Ridgway

叻報

Македония

PRAVA BRAĆA

Истински братя

DEKLARACIJA

Декларация

குடி அரசு

High Dimension Data

Examples of high dimensional data:

– Brain Imaging Data (100s of MBs per scan)

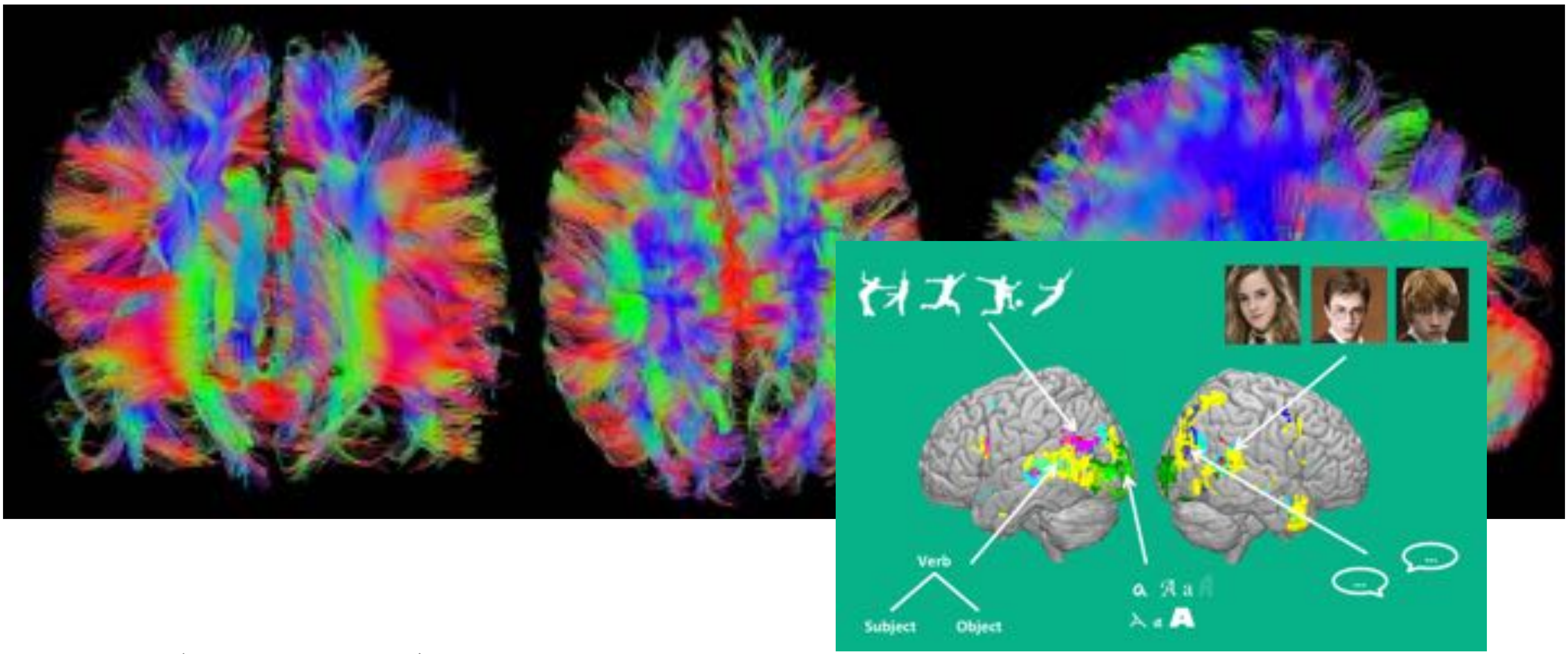


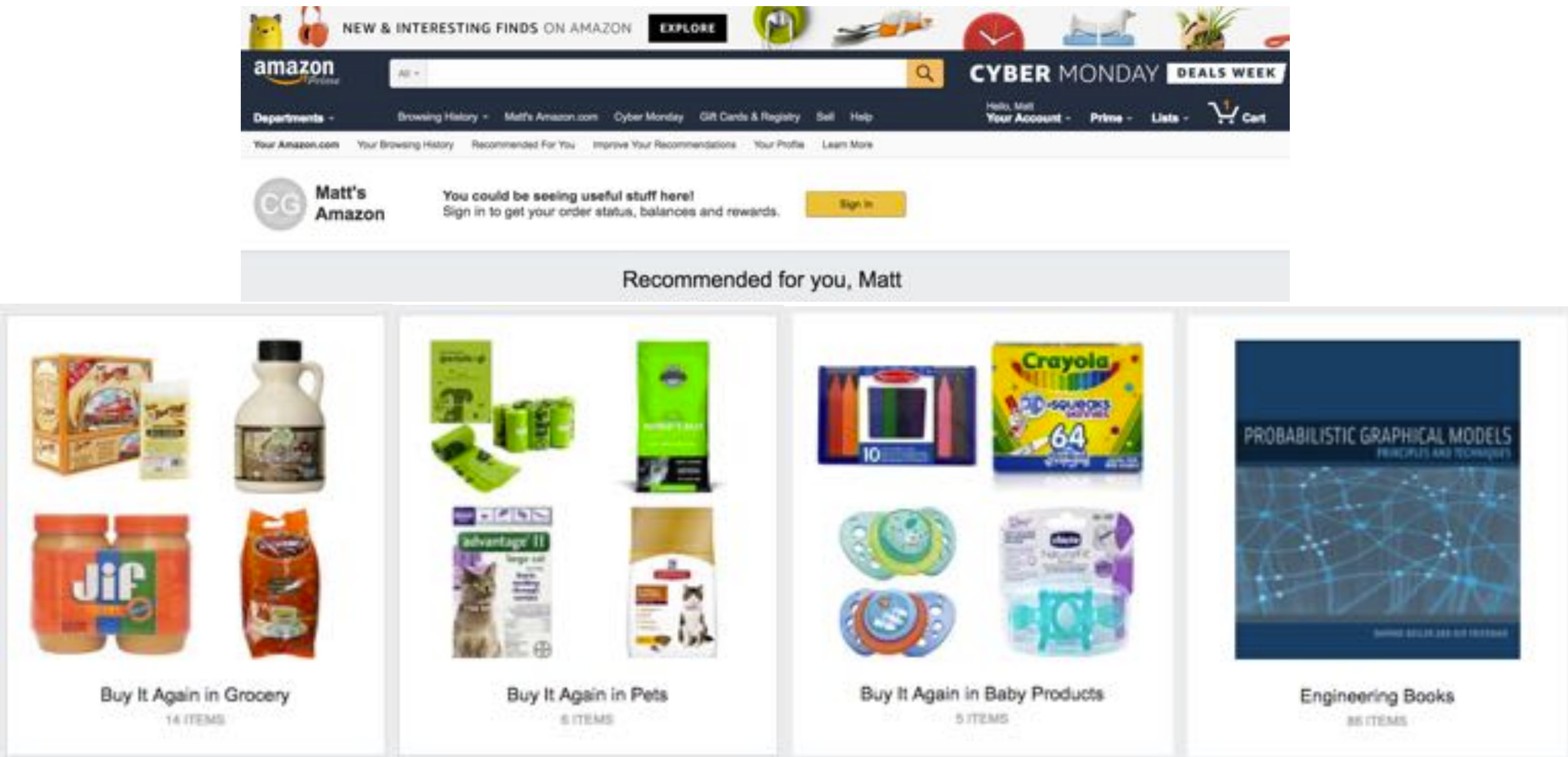
Image from (Wehbe et al., 2014)

Image from https://pixabay.com/en/brain-mrt-magnetic-resonance-imaging-1728449/

High Dimension Data

Examples of high dimensional data:

- Customer Purchase Data



Learning Representations

PCA, Kernel PCA, ICA: Powerful unsupervised learning techniques for extracting hidden (potentially lower dimensional) structure from high dimensional datasets.

Useful for:

- Visualization
- More efficient use of resources (e.g., time, memory, communication)
- Statistical: fewer dimensions → better generalization
- Noise removal (improving data quality)
- Further processing by machine learning algorithms

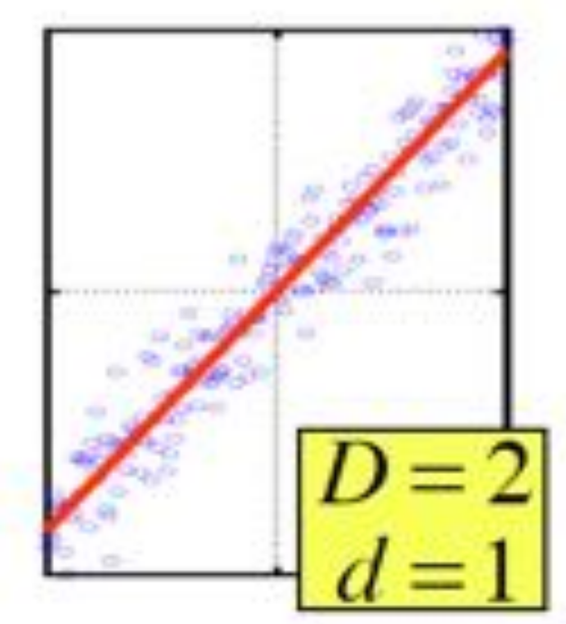
Slide from Nina Balcan

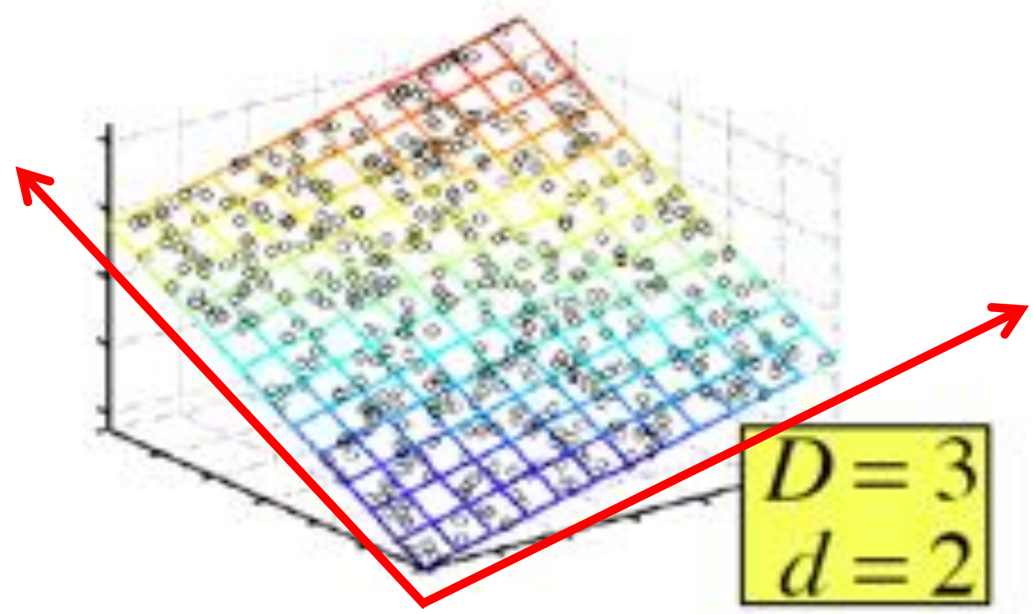
PRINCIPAL COMPONENT ANALYSIS (PCA)

PCA Outline

- **Dimensionality Reduction**
  – High-dimensional data
  – Learning (low dimensional) representations
- **Principal Component Analysis (PCA)**
  – Examples: 2D and 3D
  – Data for PCA
  – PCA Definition
  – Objective functions for PCA
  – PCA, Eigenvectors, and Eigenvalues
  – Algorithms for finding Eigenvectors / Eigenvalues
- **PCA Examples**
  – Face Recognition
  – Image Compression

Principal Component Analysis (PCA)



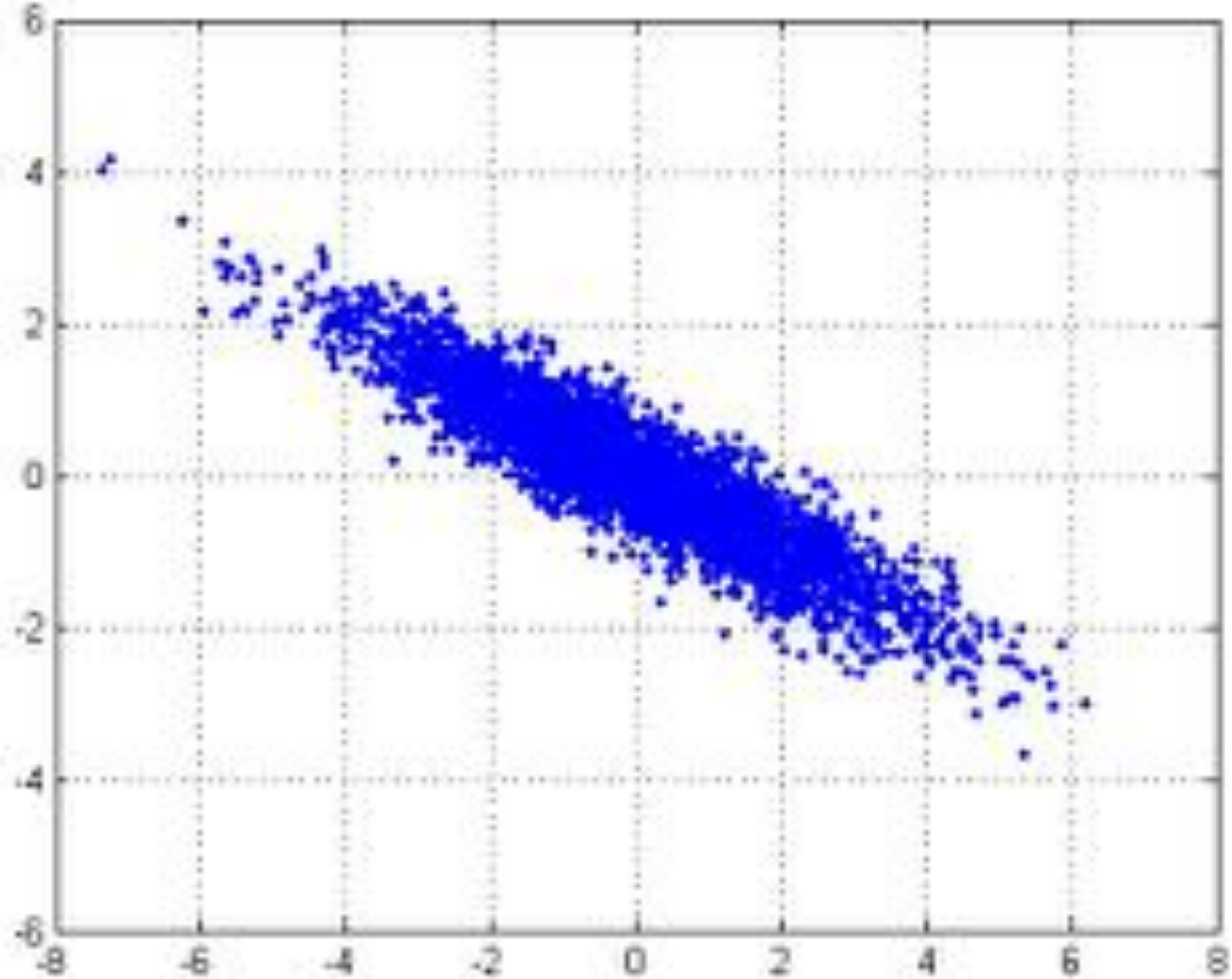


In case where data lies on or near a low d-dimensional linear subspace, axes of this subspace are an effective representation of the data.

Identifying the axes is known as Principal Components Analysis, and can be obtained by using classic matrix computation tools (Eigen or Singular Value Decomposition).

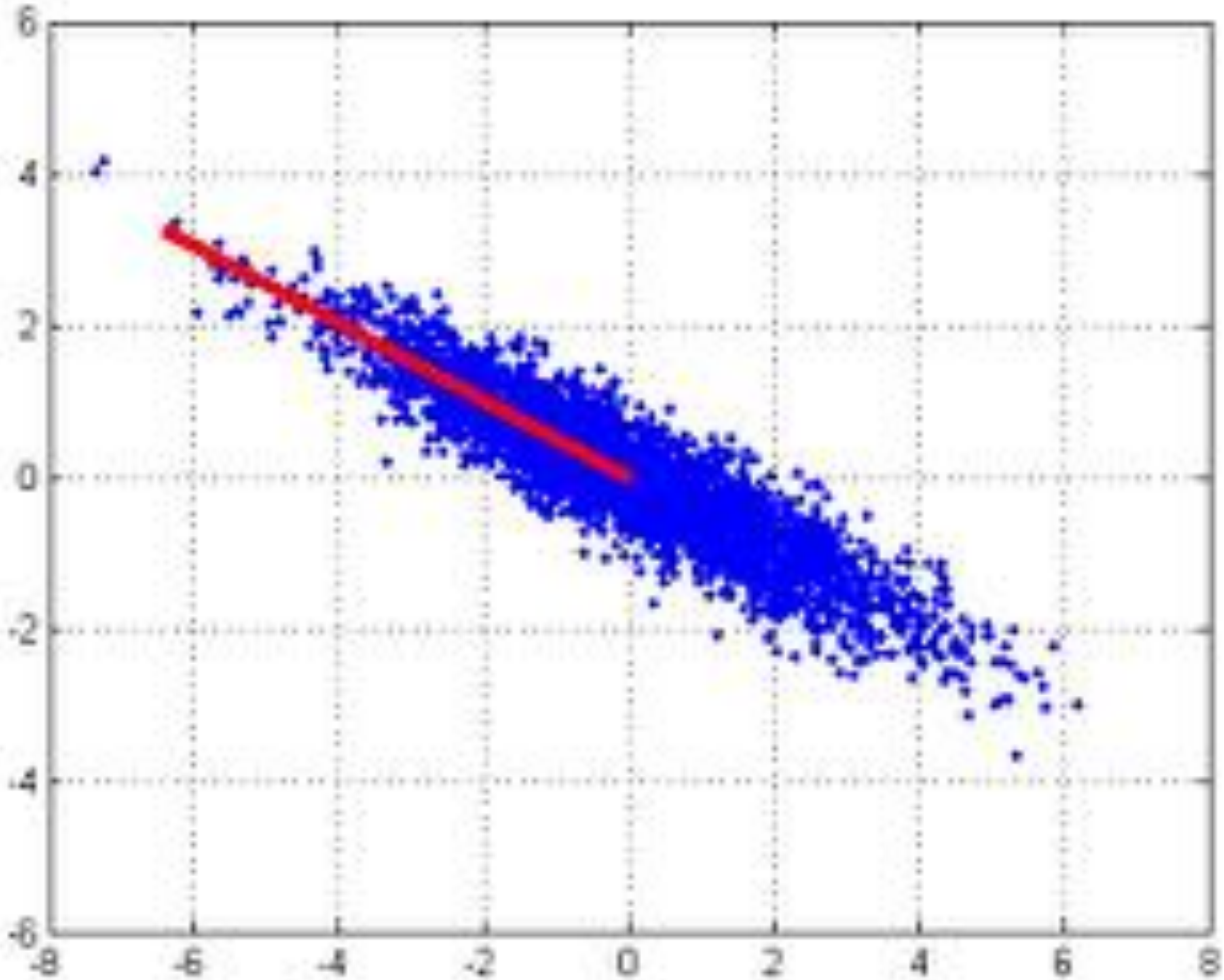
Slide from Nina Balcan

2D Gaussian dataset



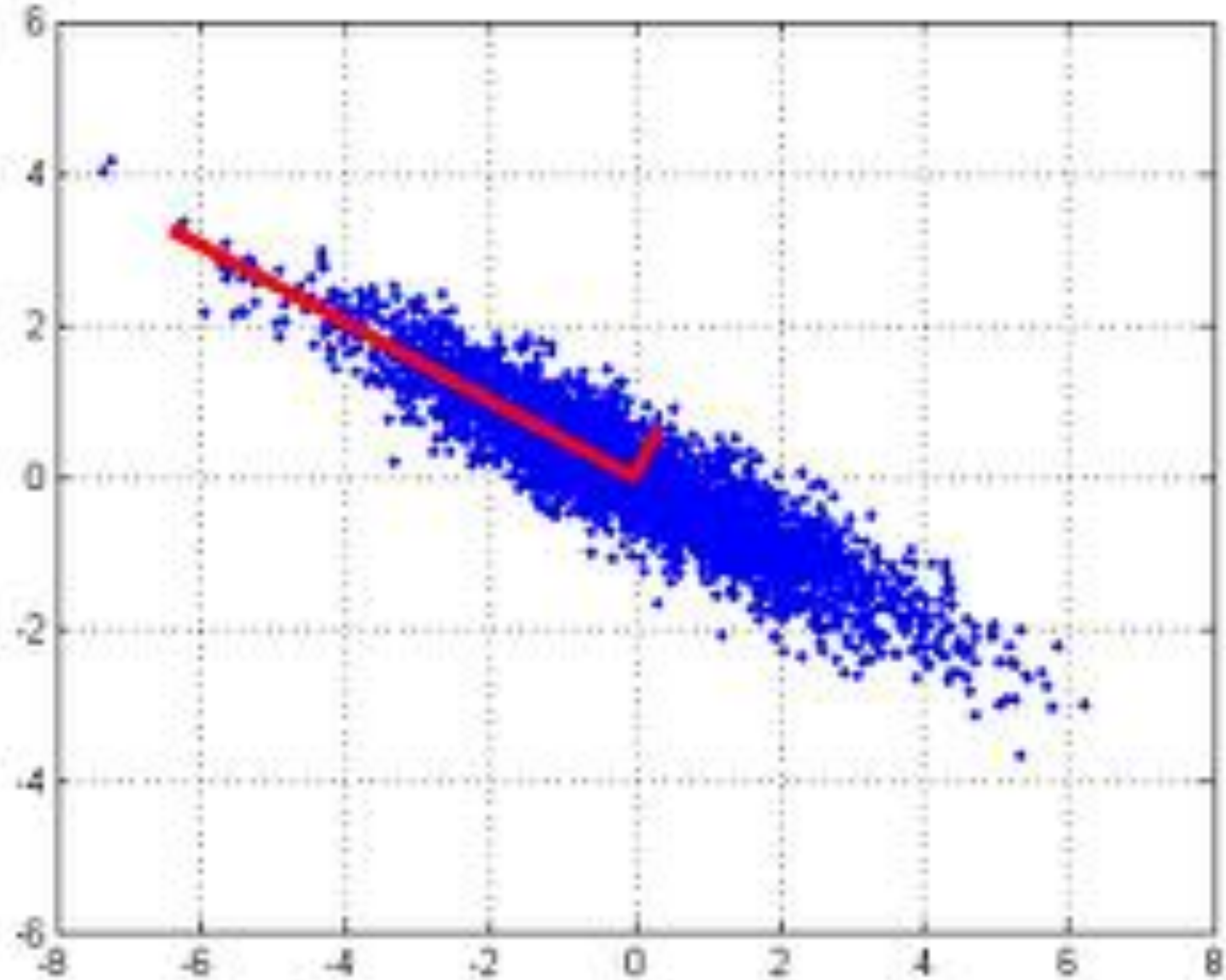
Slide from Barnabas Poczos

1^{st} PCA axis



Slide from Barnabas Poczos

2nd PCA axis



Slide from Barnabas Poczos

Principal Component Analysis (PCA)

Whiteboard

- Data for PCA
- PCA Definition
- Objective functions for PCA

Data for PCA

$$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$$

$$\mathbf{X} = \begin{bmatrix} (\mathbf{x}^{(1)})^T \\ (\mathbf{x}^{(2)})^T \\ \vdots \\ (\mathbf{x}^{(N)})^T \end{bmatrix}$$

We assume the data is **centered,** and that each axis has **sample variance equal to one.**

$$\mu = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^{(i)} = \mathbf{0}$$

$$\sigma_j^2 = \frac{1}{N} \sum_{i=1}^{N} (x_j^{(i)})^2 = 1$$

Sample Covariance Matrix

The sample covariance matrix is given by:

$$\Sigma_{jk} = \frac{1}{N} \sum_{i=1}^{N} (x_j^{(i)} - \mu_j)(x_k^{(i)} - \mu_k)$$

Since the data matrix is centered, we rewrite as:

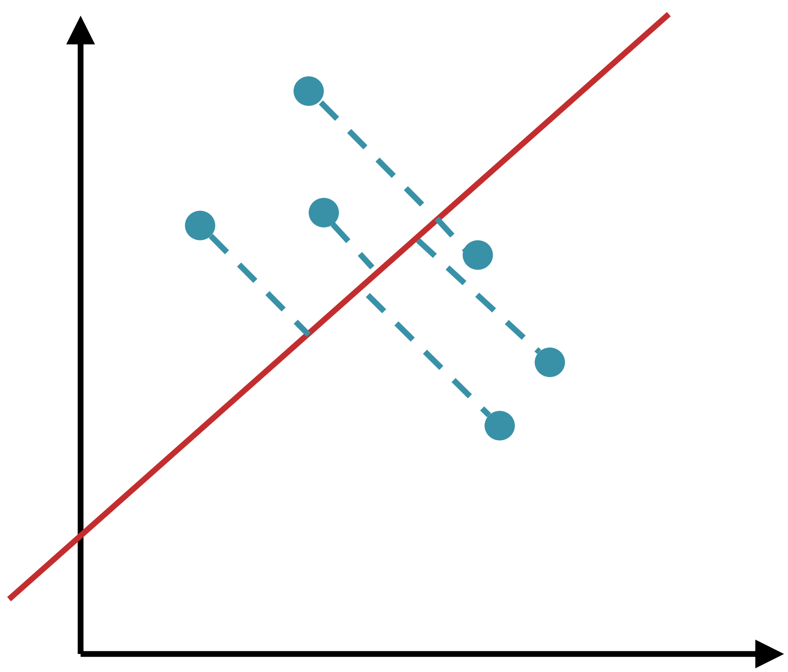
$$\mathbf{\Sigma} = \frac{1}{N} \mathbf{X}^T \mathbf{X}$$

Maximizing the Variance

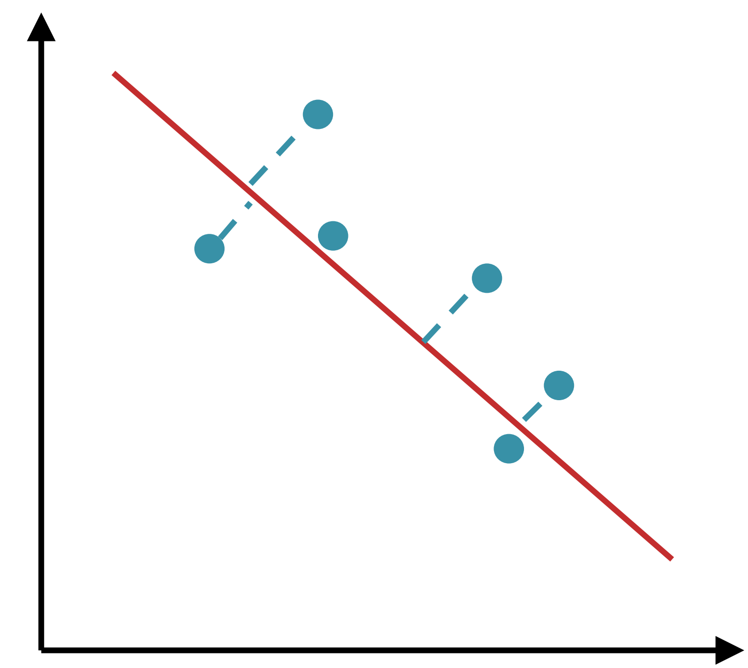
Quiz: Consider the two projections below

1. Which maximizes the variance?
2. Which minimizes the reconstruction error?

Option A



Option B



PCA

Equivalence of Maximizing Variance and Minimizing Reconstruction Error

Claim: Minimizing the reconstruction error is equivalent to maximizing the variance.

Proof: First, note that:

$$||\mathbf{x}^{(i)} - (\mathbf{v}^T\mathbf{x}^{(i)})\mathbf{v}||^2 = ||\mathbf{x}^{(i)}||^2 - (\mathbf{v}^T\mathbf{x}^{(i)})^2 \tag{1}$$

since $\mathbf{v}^T\mathbf{v} = ||\mathbf{v}||^2 = 1$.

Substituting into the minimization problem, and removing the extraneous terms, we obtain the maximization problem.

$$\begin{aligned}
\mathbf{v}^* &= \underset{\mathbf{v}:||\mathbf{v}||^2=1}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{x}^{(i)} - (\mathbf{v}^T\mathbf{x}^{(i)})\mathbf{v}||^2 && (2)\\
&= \underset{\mathbf{v}:||\mathbf{v}||^2=1}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{x}^{(i)}||^2 - (\mathbf{v}^T\mathbf{x}^{(i)})^2 && (3)\\
&= \underset{\mathbf{v}:||\mathbf{v}||^2=1}{\operatorname{argmax}} \frac{1}{N}\sum_{i=1}^{N} (\mathbf{v}^T\mathbf{x}^{(i)})^2 && (4)\\
& && (5)
\end{aligned}$$

Principal Component Analysis (PCA)

Whiteboard

- PCA, Eigenvectors, and Eigenvalues
- Algorithms for finding Eigenvectors / Eigenvalues
- SVD: Relation of Singular Vectors to Eigenvectors

SVD for PCA

For any arbitrary matrix $\mathbf{A}$, SVD gives a decomposition:

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T \qquad (1)$$

where $\mathbf{\Lambda}$ is a diagonal matrix, and $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices.

Suppose we obtain an SVD of our data matrix $\mathbf{X}$, so that:

$$\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T \qquad (1)$$

Now consider what happens when we rewrite $\mathbf{\Sigma} = \frac{1}{N}\mathbf{X}^T\mathbf{X}$ terms of this SVD.

$$\begin{aligned}
\mathbf{\Sigma} &= \frac{1}{N}\mathbf{X}^T\mathbf{X} && (2)\\
&= \frac{1}{N}(\mathbf{U}\mathbf{\Lambda}\mathbf{V}^T)^T(\mathbf{U}\mathbf{\Lambda}\mathbf{V}^T) && (3)\\
&= \frac{1}{N}(\mathbf{V}\mathbf{\Lambda}^T\mathbf{U}^T)(\mathbf{U}\mathbf{\Lambda}\mathbf{V}^T) && (4)\\
&= \frac{1}{N}\mathbf{V}\mathbf{\Lambda}^T\mathbf{\Lambda}\mathbf{V}^T && (5)\\
&= \frac{1}{N}\mathbf{V}(\mathbf{\Lambda})^2\mathbf{V}^T && (6)
\end{aligned}$$

Above we used the fact that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ since $\mathbf{U}$ is orthogonal by definition.

SVD for PCA

For any arbitrary matrix $\mathbf{A}$, SVD gives a decomposition:

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^T \tag{1}$$

where $\boldsymbol{\Lambda}$ is a diag

Suppose we obta

Now consider wh
of this SVD.

We find that $(\boldsymbol{\Lambda})^2$ is a diagonal matrix whose entries are $\Lambda_{ii} = \lambda_i^2$ the squares of the eigenvalues of the SVD of $\mathbf{X}$. Further, both $\mathbf{X}$ and $\mathbf{X}^T\mathbf{X}$ share the same eigenvectors in their SVD.

Thus, we can run SVD on $\mathbf{X}$ without ever instantiating the large $\mathbf{X}^T\mathbf{X}$ to obtain the necessary principal components more efficiently.

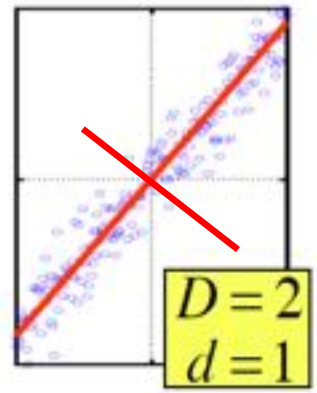
$$\begin{aligned}
\boldsymbol{\Sigma} &= \frac{1}{N}\mathbf{X}^T\mathbf{X} && (2)\\
&= \frac{1}{N}(\mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^T)^T(\mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^T) && (3)\\
&= \frac{1}{N}(\mathbf{V}\boldsymbol{\Lambda}^T\mathbf{U}^T)(\mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^T) && (4)\\
&= \frac{1}{N}\mathbf{V}\boldsymbol{\Lambda}^T\boldsymbol{\Lambda}\mathbf{V}^T && (5)\\
&= \frac{1}{N}\mathbf{V}(\boldsymbol{\Lambda})^2\mathbf{V}^T && (6)
\end{aligned}$$

Above we used the fact that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ since $\mathbf{U}$ is orthogonal by definition.

Principal Component Analysis (PCA)

$(X\,X^T)v = \lambda v$, so v (the first PC) is the eigenvector of sample correlation/covariance matrix $X\,X^T$

Sample variance of projection $v^T X\,X^T v = \lambda v^T v = \lambda$



Thus, the eigenvalue λ denotes the amount of variability captured along that dimension (aka amount of energy along that dimension).

Eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots$

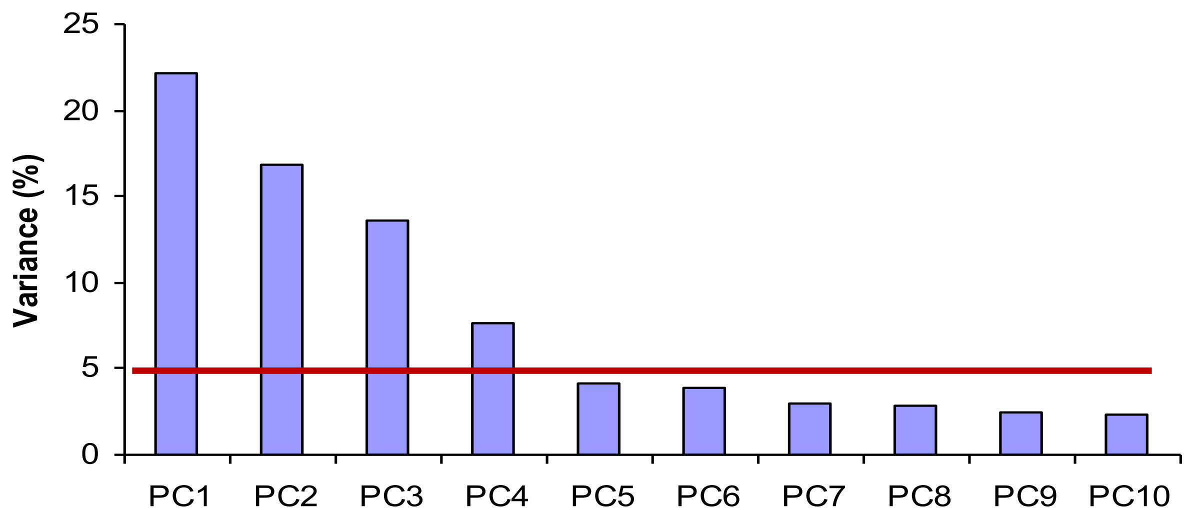
- The 1st PC v_1 is the the eigenvector of the sample covariance matrix $X\,X^T$ associated with the largest eigenvalue
- The 2nd PC v_2 is the the eigenvector of the sample covariance matrix $X\,X^T$ associated with the second largest eigenvalue
- And so on …

Slide from Nina Balcan

How Many PCs?

- For M original dimensions, sample covariance matrix is MxM, and has up to M eigenvectors. So M PCs.
- Where does dimensionality reduction come from?

  Can *ignore* the components of lesser significance.



- You do lose some information, but if the eigenvalues are small, you don't lose much
  - M dimensions in original data
  - calculate M eigenvectors and eigenvalues
  - choose only the first D eigenvectors, based on their eigenvalues
  - final data set has only D dimensions

Slides from Barnabas Poczos

Original sources include:

- Karl Booksh Research group
- Tom Mitchell
- Ron Parr

PCA EXAMPLES

Face recognition

Slide from Barnabas Poczos

Challenge: Facial Recognition

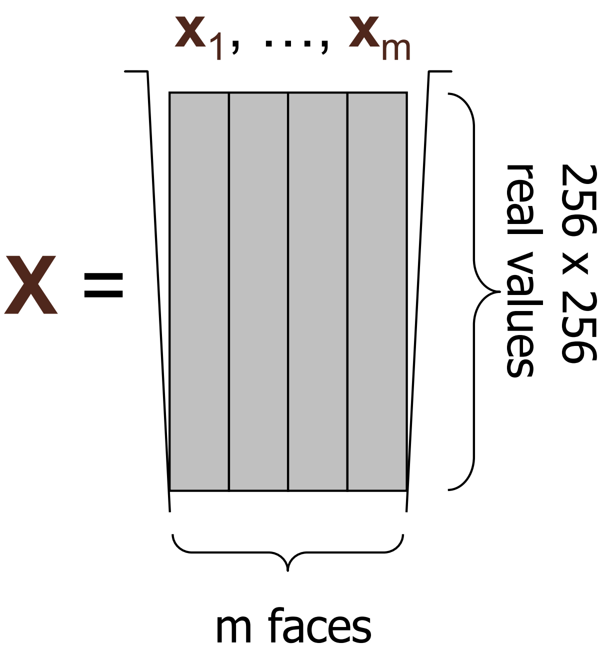
- Want to identify specific person, based on facial image
- Robust to glasses, lighting,…

⇒ Can't just use the given 256 x 256 pixels

Slide from Barnabas Poczos

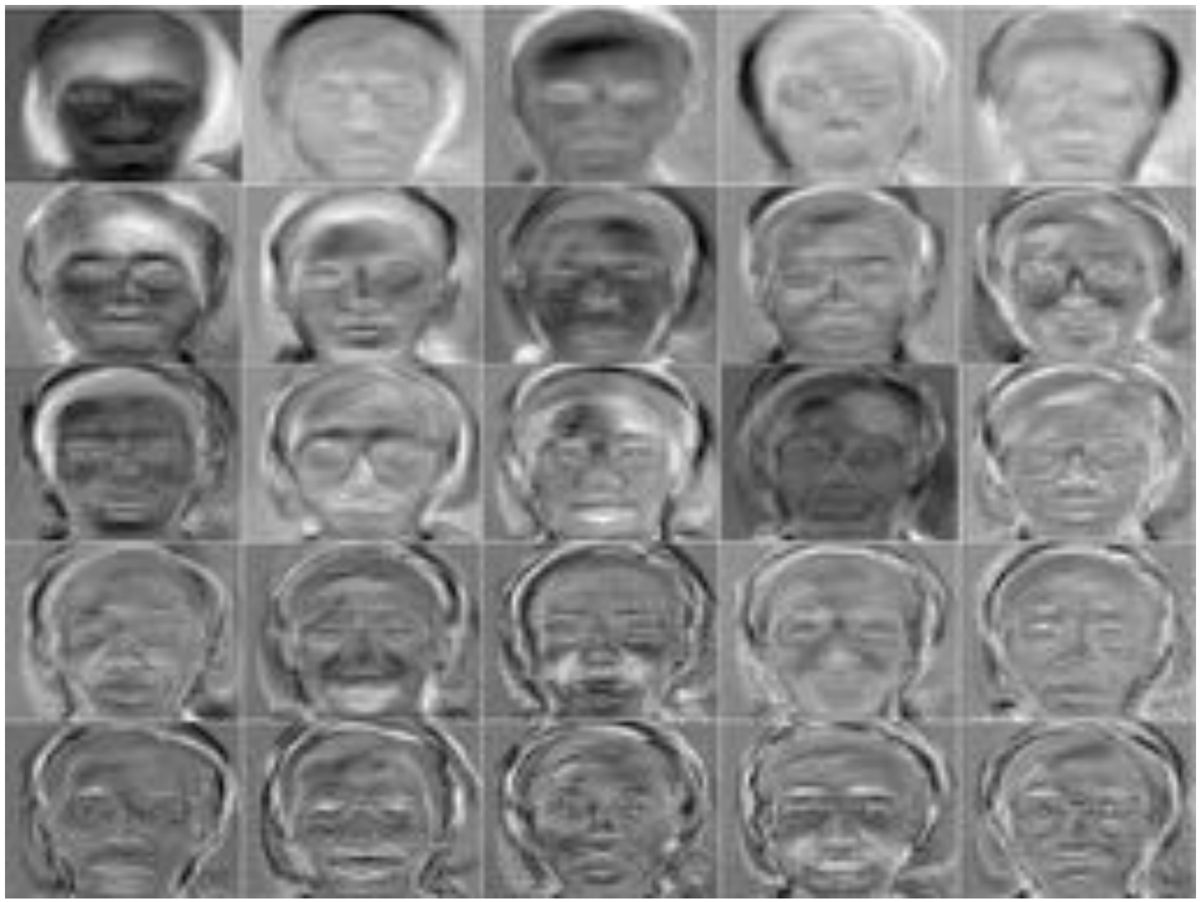
Applying PCA: Eigenfaces

Method: Build one PCA database for the whole dataset and then classify based on the weights.



- Example data set: Images of faces
  - Famous Eigenface approach
    [Turk & Pentland], [Sirovich & Kirby]
- Each face $\mathbf{x}$ is …
  - 256×256 values (luminance at location)
  - $\mathbf{x}$ in $\Re^{256 \times 256}$ (view as 64K dim vector)

Principle Components



Slide from Barnabas Poczos

Reconstructing…

- … faster if train with…
  - only people w/out glasses
  - same lighting conditions

Shortcomings

- Requires carefully controlled data:
  - All faces centered in frame
  - Same size
  - Some sensitivity to angle
- Alternative:
  - "Learn" one set of PCA vectors for each angle
  - Use the one with lowest error

- Method is completely knowledge free
  - (sometimes this is good!)
  - Doesn't know that faces are wrapped around 3D objects (heads)
  - Makes no effort to preserve class distinctions

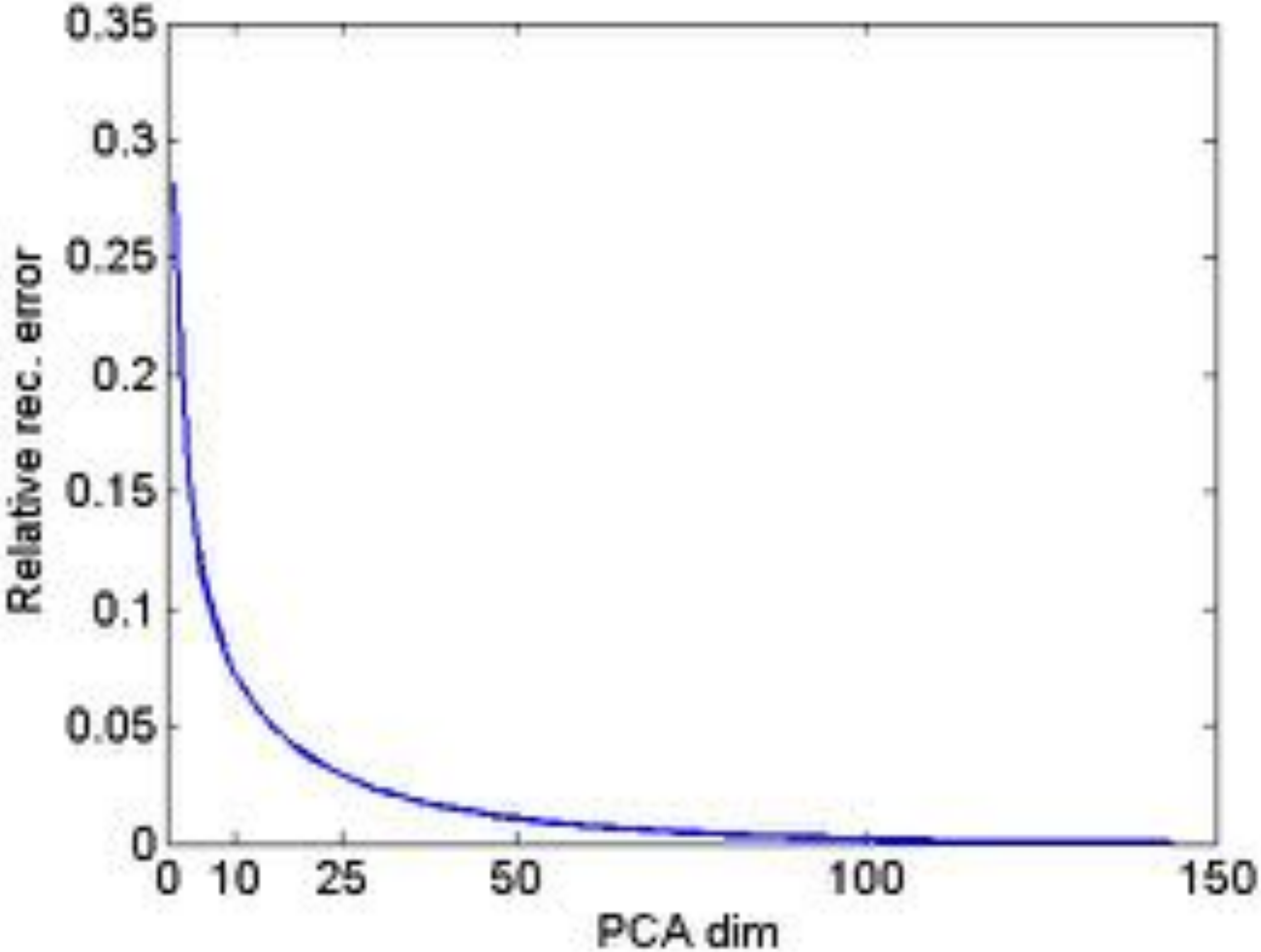
Image Compression

Slide from Barnabas Poczos

Original Image

- Divide the original 372x492 image into patches:
  - Each patch is an instance that contains 12x12 pixels on a grid
- View each as a 144-D vector

L_2 error and PCA dim



Slide from Barnabas Poczos

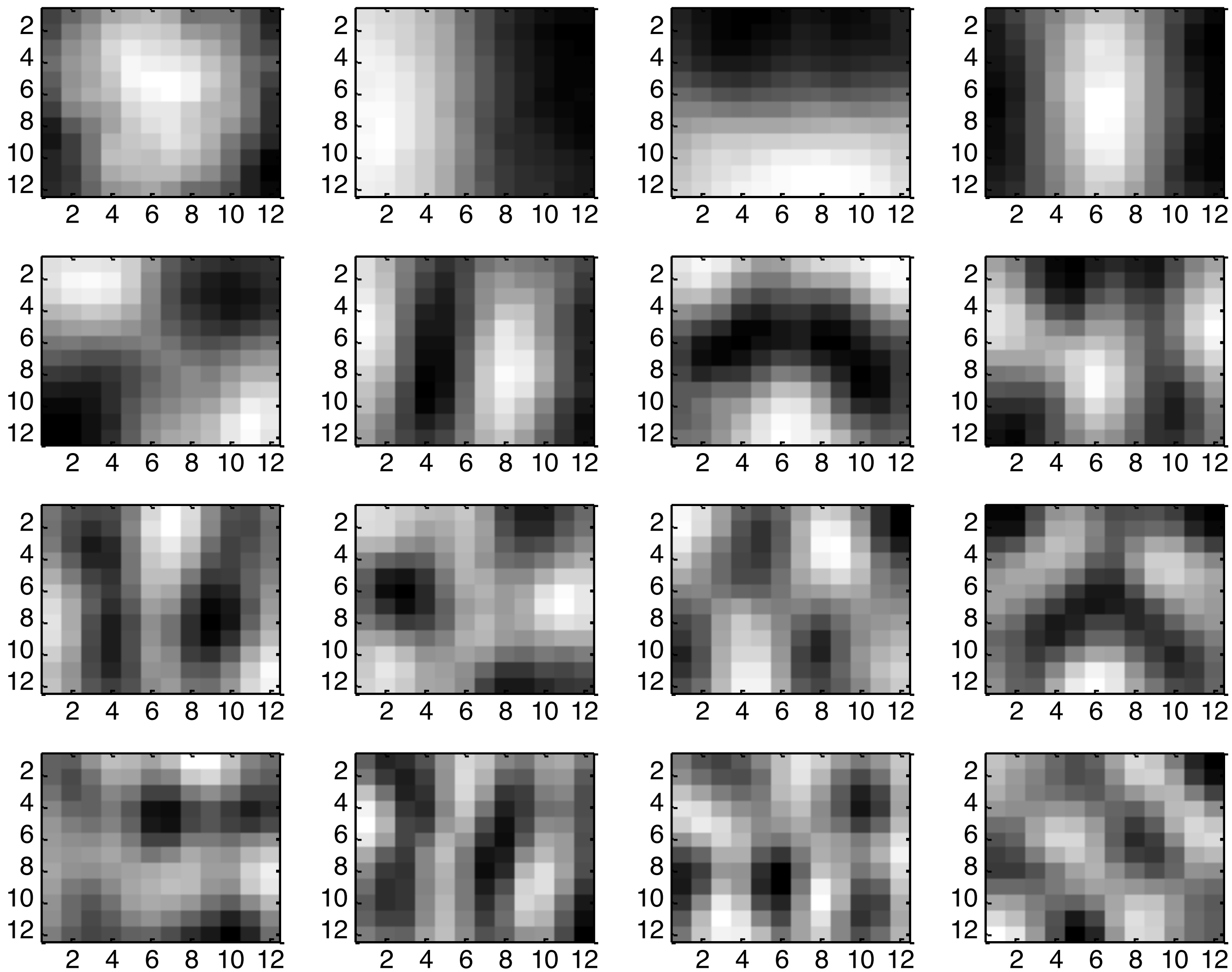
PCA compression: 144D → 60D

Slide from Barnabas Poczos

PCA compression: 144D → 16D

Slide from Barnabas Poczos

16 most important eigenvectors

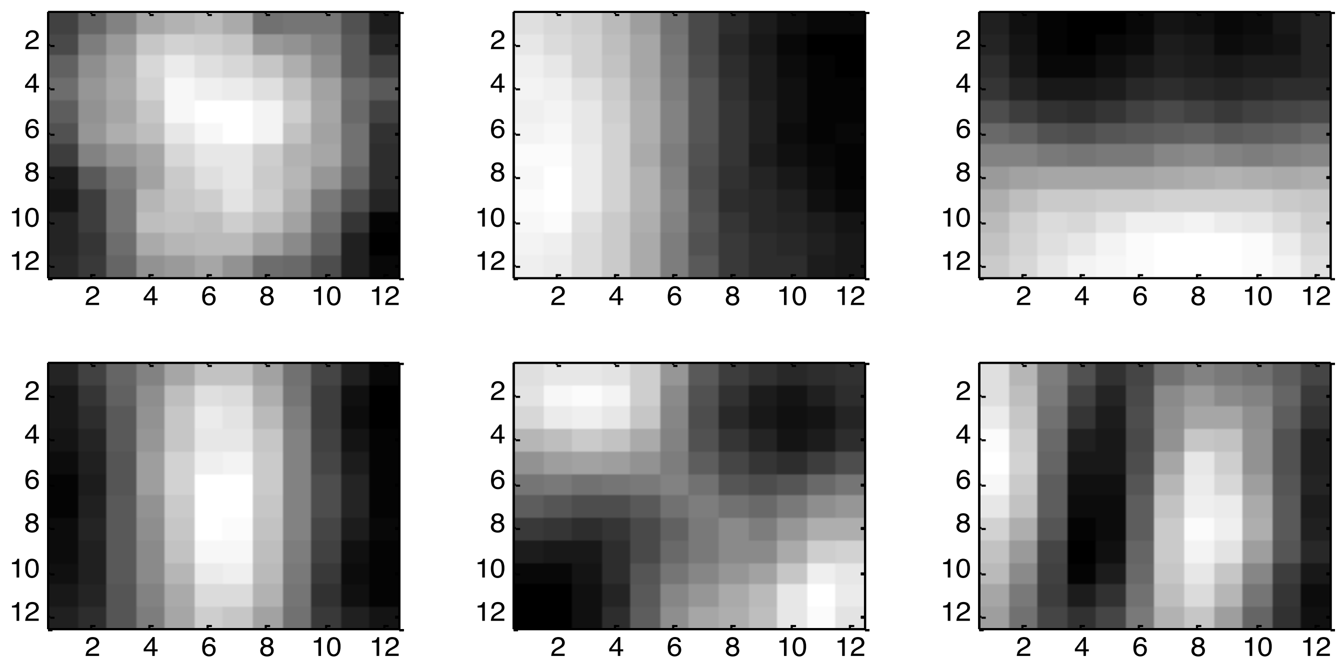


Slide from Barnabas Poczos

PCA compression: 144D → 6D

Slide from Barnabas Poczos

6 most important eigenvectors

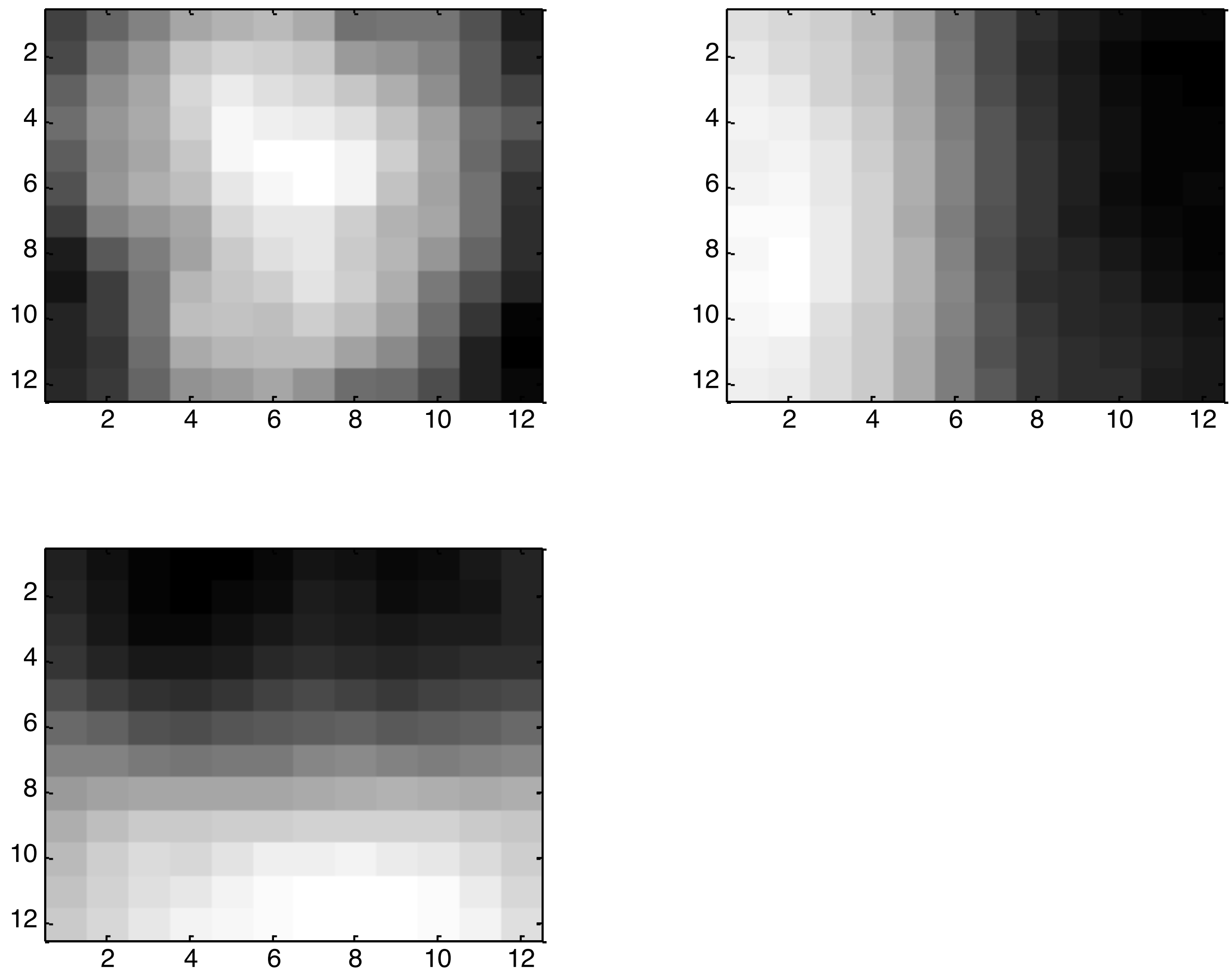


Slide from Barnabas Poczos

PCA compression: 144D → 3D

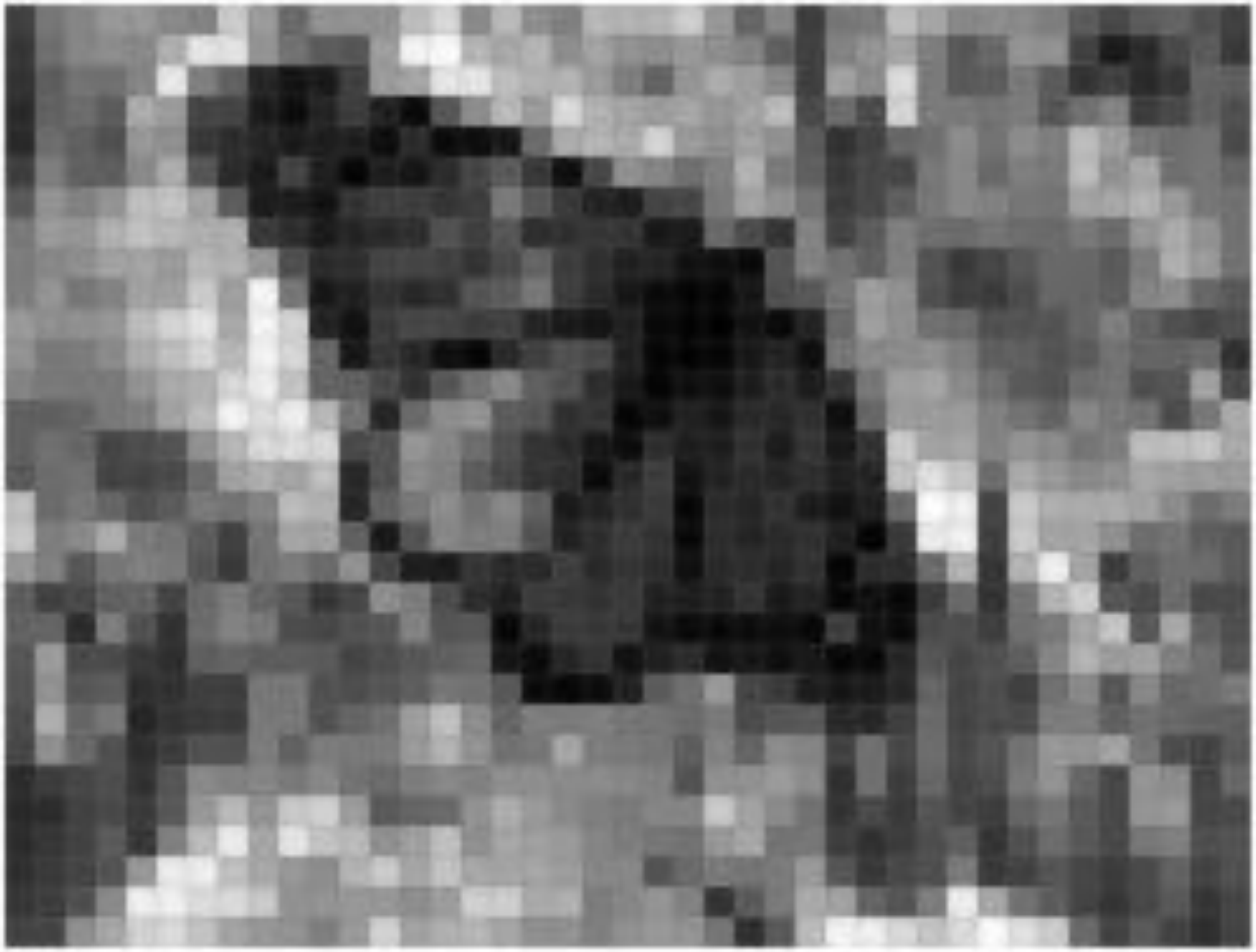
Slide from Barnabas Poczos

3 most important eigenvectors



Slide from Barnabas Poczos

PCA compression: 144D → 1D



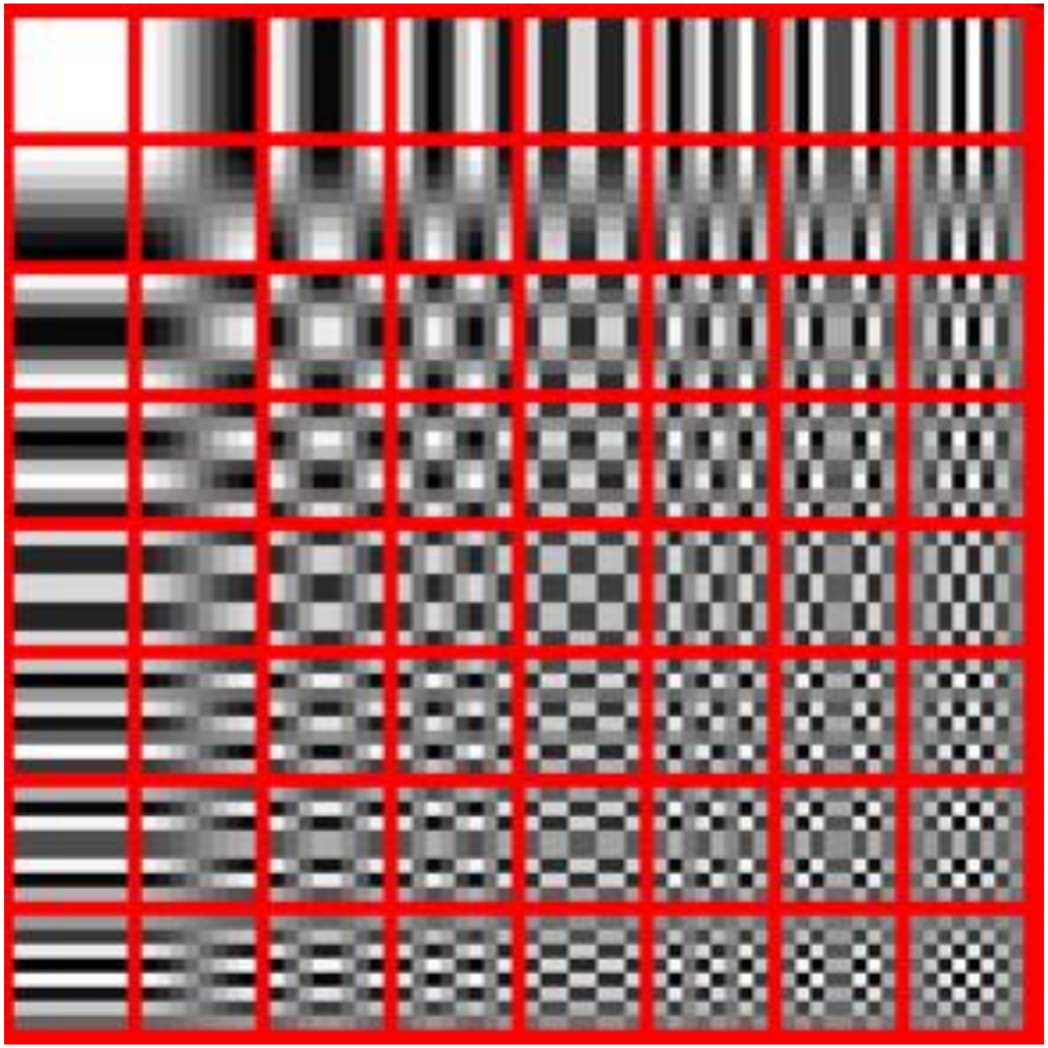
Slide from Barnabas Poczos

60 most important eigenvectors

Looks like the discrete cosine bases of JPG!...

Slide from Barnabas Poczos

2D Discrete Cosine Basis



http://en.wikipedia.org/wiki/Discrete_cosine_transform