CSC445: Neural Networks

CHAPTERS 8

UNSUPERVISED LEARNING:

PRINCIPAL-COMPONENTS ANALYSIS (PCA)

Prof. Dr. Mostafa Gadal-Haqq M. Mostafa

Computer Science Department

Faculty of Computer & Information Sciences

AIN SHAMS UNIVERSITY

Credits: Some Slides are taken from presentations on PCA by :

1. **Barnabás Póczos University of Alberta**
2. **Jieping Ye, http://www.public.asu.edu/~jye02**

Outlines

- Introduction
- Tasks of Unsupervised Learning
- What is Data Reduction?
- Why we need to Reduce Data Dimensionality?
- Clustering and Data Reduction
- The PCA Computation
- Computer Experiment

Unsupervised Learning

- In *unsupervised learning*, the requirement is to *discover significant patterns*, or *features*, of the input data through the use of *unlabeled examples*.

- That it, the network operates according to the rule:

"Learn from examples without a teacher"

What is feature reduction?

- **Feature reduction refers to the mapping of the original high-dimensional data onto a lower-dimensional space.**
  - Criterion for feature reduction can be different based on different problem settings.
    - Unsupervised setting: minimize the information loss
    - Supervised setting: maximize the class discrimination

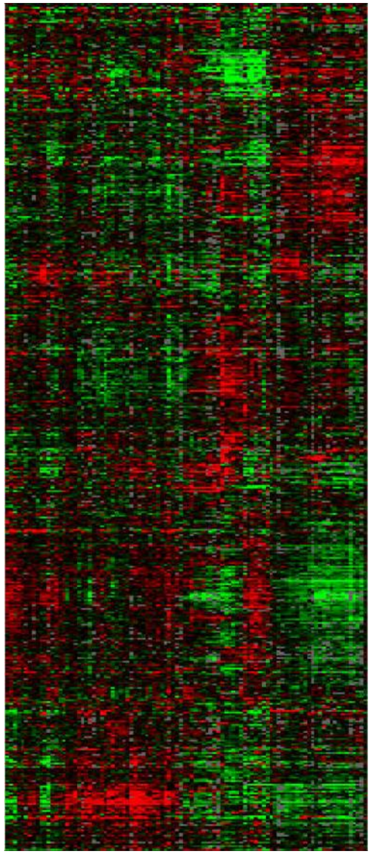
- **Given a set of data points of p variables**

  Compute the linear transformation (projection)

$$\{x_1, x_2, \cdots, x_n\}$$

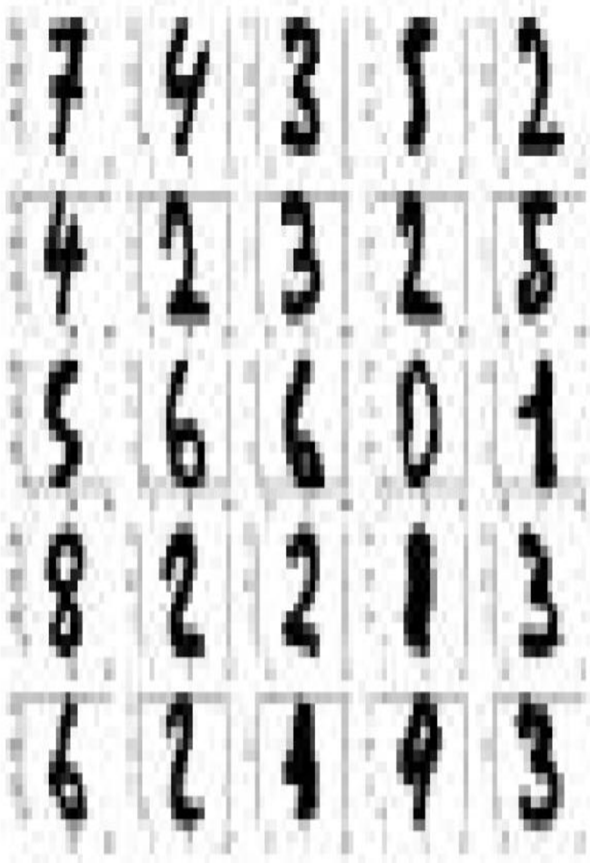
$$G \in \Re^{p \times d} : x \in \Re^p \rightarrow y = G^T x \in \Re^d \ (d << p)$$

High Dimensional Data



Gene expression

Face images



Handwritten digits

Why feature reduction?

- Most machine learning and data mining techniques may not be effective for high-dimensional data
  - Curse of Dimensionality
  - Query accuracy and efficiency degrade rapidly as the dimension increases.
- The intrinsic dimension may be small.
  - For example, the number of genes responsible for a certain type of disease may be small.

Why feature reduction?

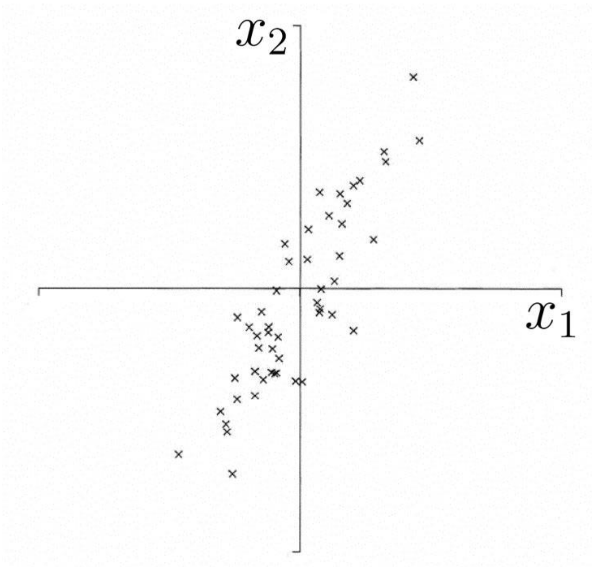
- Visualization: projection of high-dimensional data onto 2D or 3D.
- Data compression: efficient storage and retrieval.
- Noise removal: positive effect on query accuracy.

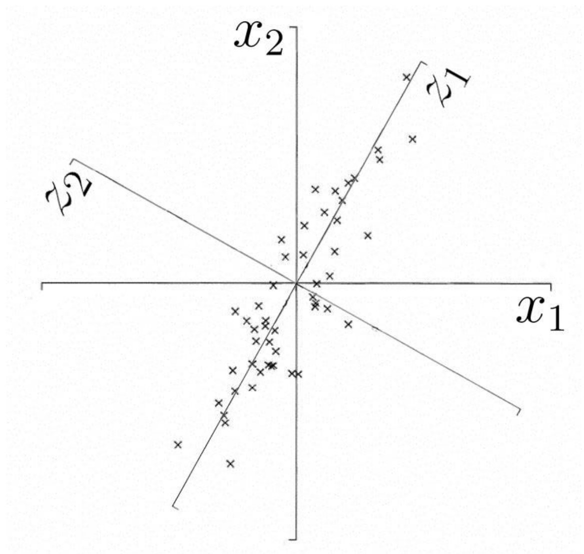
What is Principal Component Analysis?

- Principal component analysis (PCA)
  - Reduce the dimensionality of a data set by finding a new set of variables, smaller than the original set of variables
  - Retains most of the sample's information.
  - Useful for the compression and classification of data.

- By information we mean the variation present in the sample, given by the correlations between the original variables.
  - The new variables, called principal components (PCs), are uncorrelated, and are ordered by the fraction of the total information each retains.

principal components (PCs)





- the 1st PC z_1 is a minimum distance fit to a line in X space
- the 2nd PC z_2 is a minimum distance fit to a line in the plane perpendicular to the 1st PC

PCs are a series of linear least squares fits to a sample, each orthogonal to all the previous.

Algebraic definition of PCs

Given a sample of n observations on a vector of p variables

$$\{x_1, x_2, \cdots, x_n\} \in \Re^p$$

define the first principal component of the sample
by the linear transformation

$$z_1 = a_1^T x_j = \sum_{i=1}^{p} a_{i1} x_{ij}, \quad j = 1, 2, \cdots, n.$$

where the vector

$$a_1 = (a_{11}, a_{21}, \cdots, a_{p1})$$

$$x_j = (x_{1j}, x_{2j}, \cdots, x_{pj})$$

is chosen such that $\mathrm{var}[z_1]$ is maximum.

Algebraic Derivation of the PCA

To find a_1 first note that

$$\operatorname{var}[z_1] = E((z_1 - \overline{z_1})^2) = \frac{1}{n}\sum_{i=1}^{n}\left(a_1^T x_i - a_1^T \overline{x}\right)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n} a_1^T \left(x_i - \overline{x}\right)\left(x_i - \overline{x}\right)^T a_1 = a_1^T \Sigma a_1$$

where

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \overline{x}\right)\left(x_i - \overline{x}\right)^T$$

is the covariance matrix.

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \text{ is the mean.}$$

In the following, we assume the Data is centered.

$$\overline{x} = 0$$

Algebraic derivation of PCs

Assume $\bar{x} = 0$

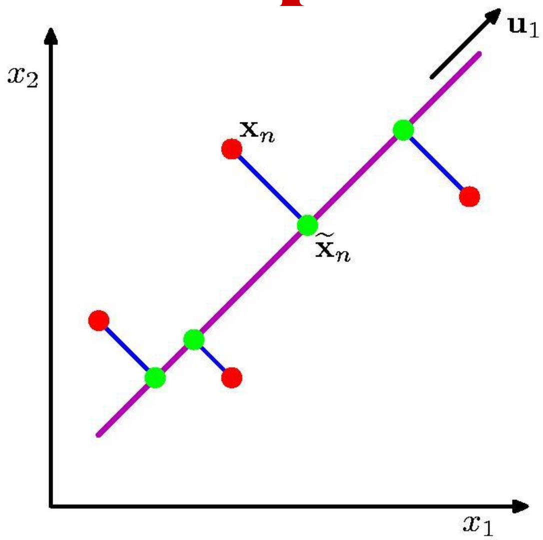
Form the matrix: $X = [x_1, x_2, \cdots, x_n] \in \Re^{p \times n}$

then $S = \frac{1}{n} X X^T$

Obtain eigenvectors of S by computing the SVD of X:

$$X = U \Sigma V^T$$

Principle Component Analysis



PCA:

Orthogonal projection of data onto lower-dimension linear space that...

- maximizes variance of projected data (purple line)
- minimizes mean squared distance between
  - data point and
  - projections (sum of blue lines)

Principle Components Analysis

Idea:

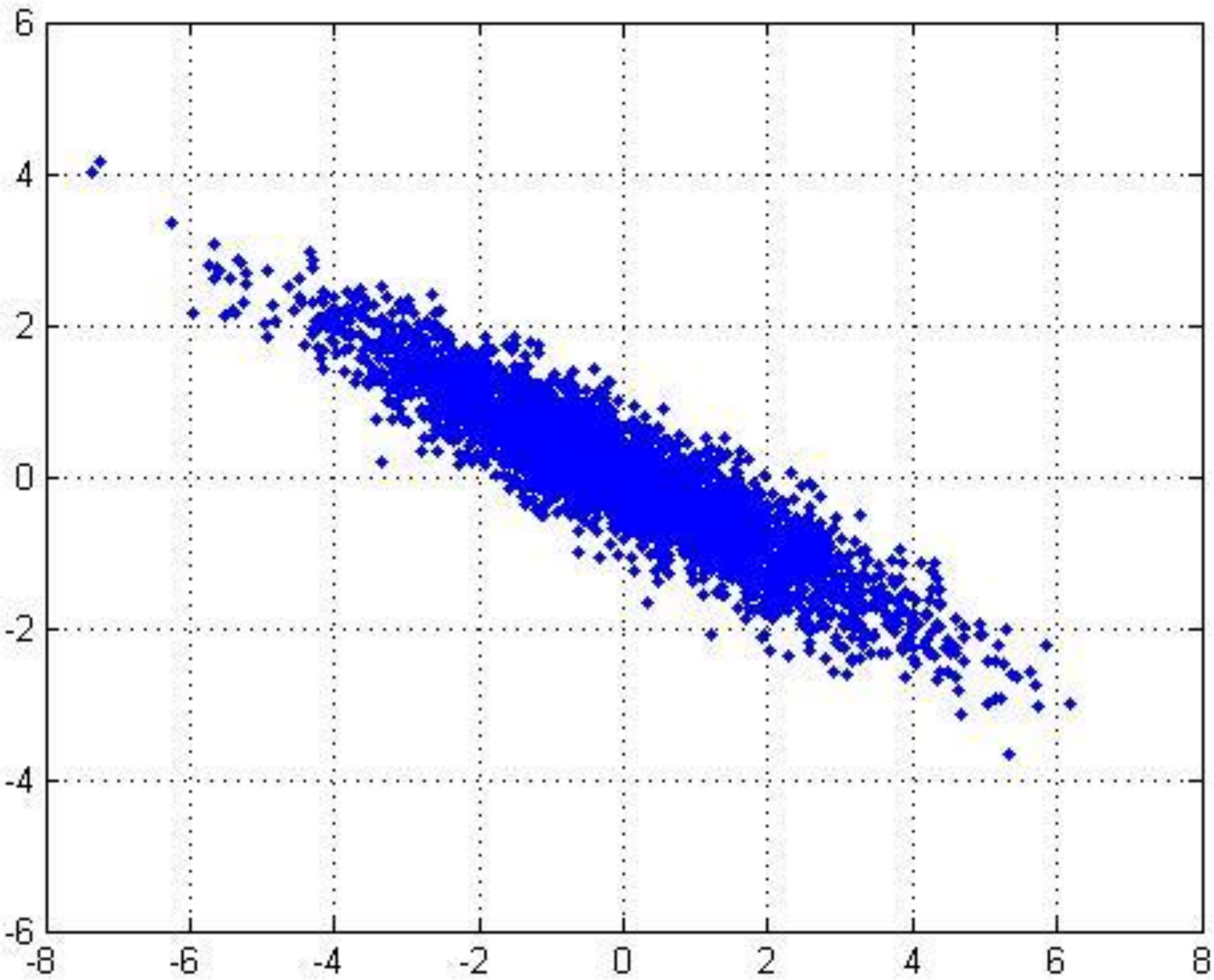
- Given data points in a d-dimensional space, project into lower dimensional space while preserving as much information as possible
  - Eg, find best planar approximation to 3D data
  - Eg, find best 12-D approximation to 10^4-D data

- In particular, choose projection that minimizes *squared error* in reconstructing original data

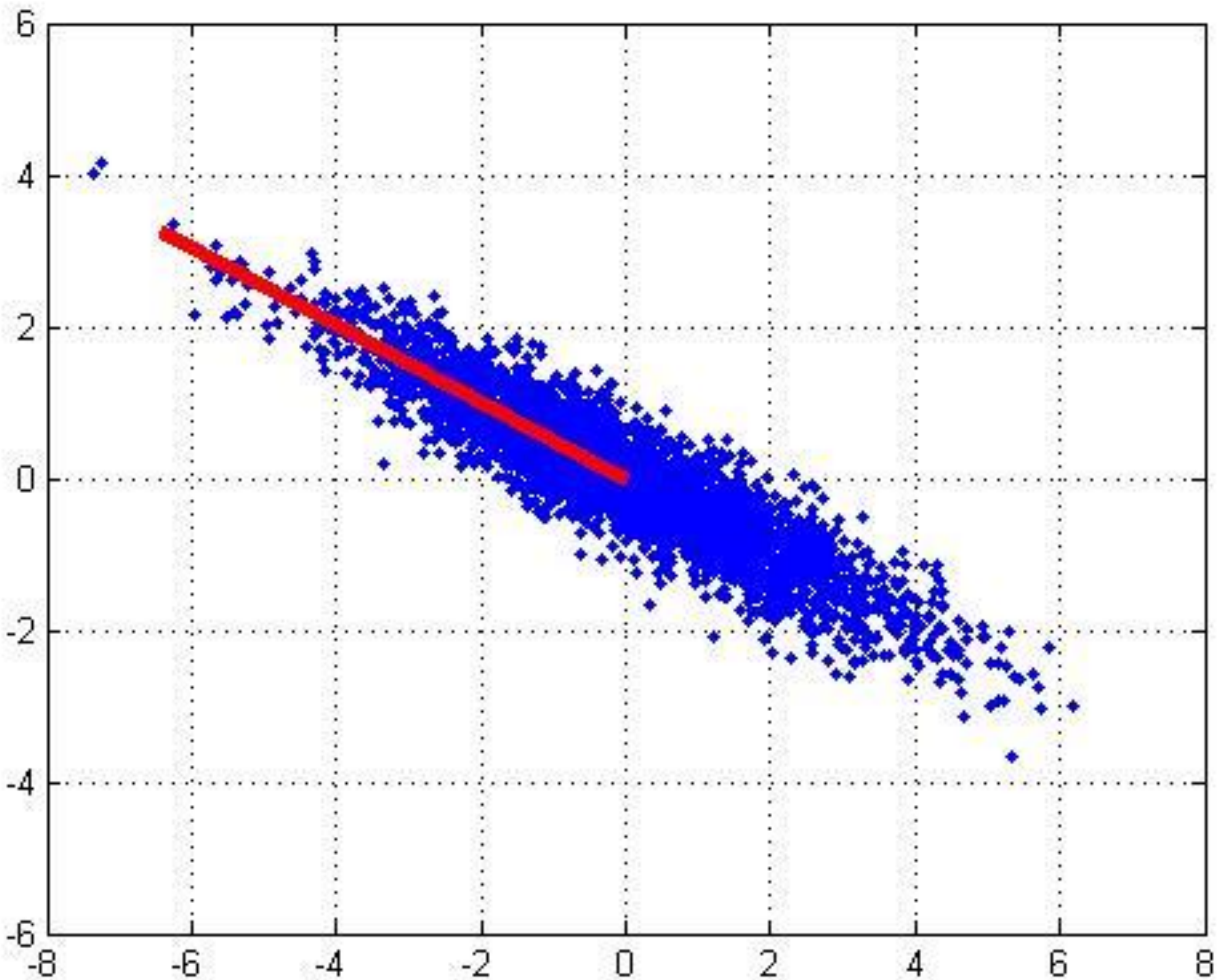
The Principal Components

- **Vectors** originating from the center of mass
- Principal component #1 points
  in the direction of the **largest variance**.
- Each subsequent principal component…
  - is **orthogonal** to the previous ones, and
  - points in the directions of the **largest variance of the residual subspace**

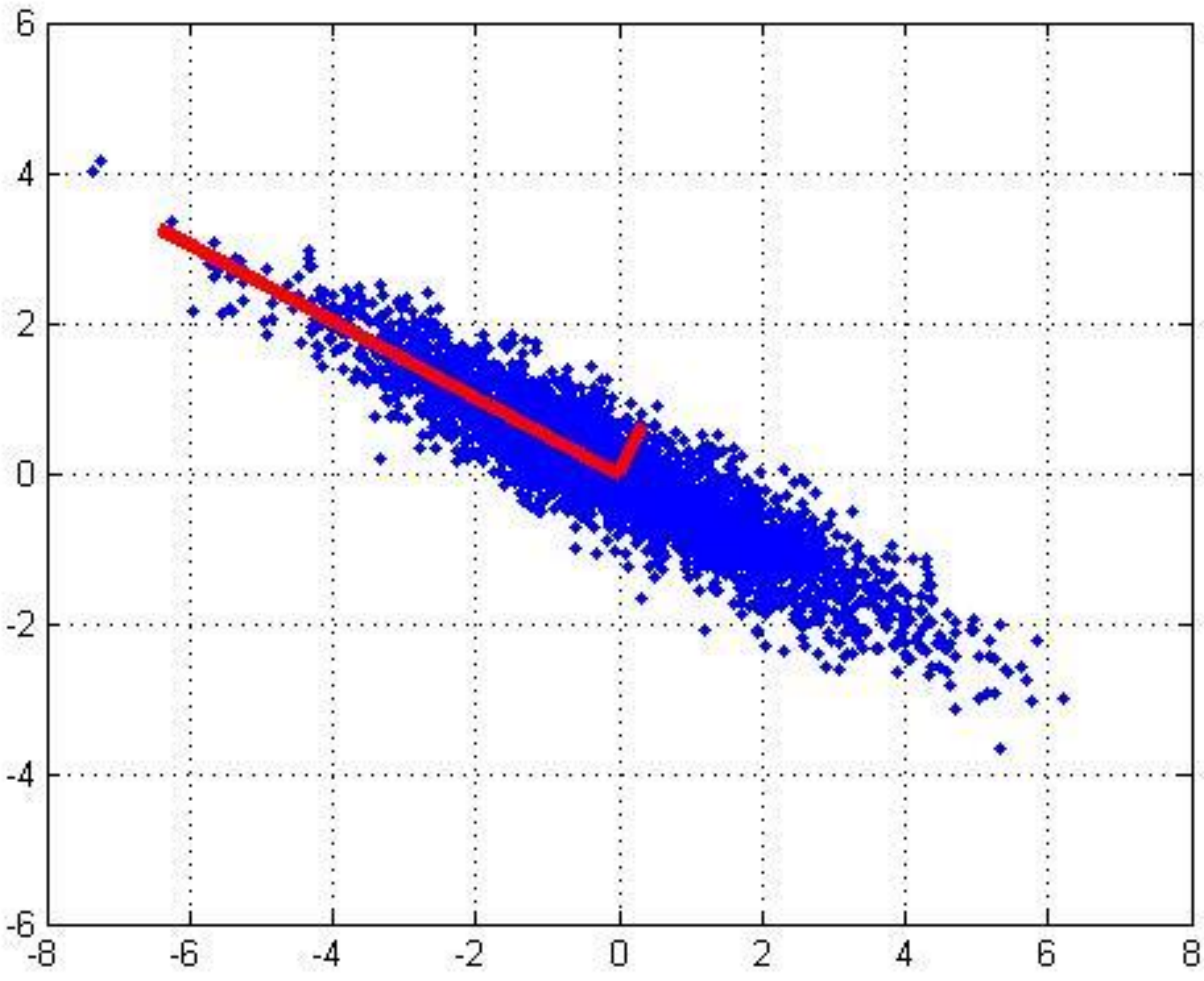
2D Gaussian dataset



1st PCA axis



2nd PCA axis



PCA algorithm I (sequential)

Given the **centered** data $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, compute the principal vectors:

$$\mathbf{w}_1 = \arg\max_{\|\mathbf{w}\|=1} \frac{1}{m}\sum_{i=1}^{m}\{(\mathbf{w}^T\mathbf{x}_\mathrm{i})^2\}$$

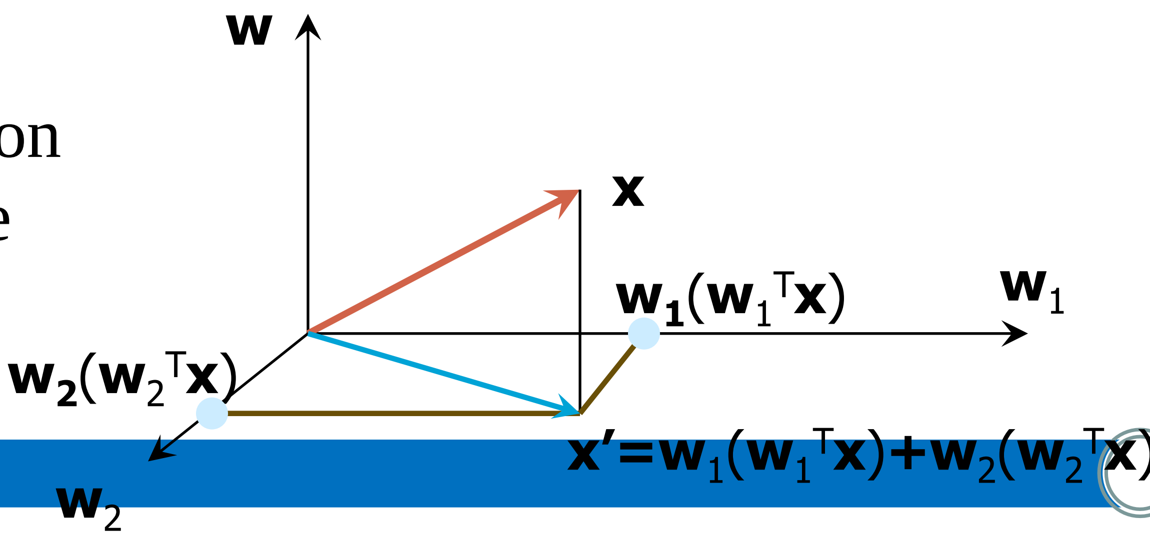
1st PCA vector

We maximize the variance of projection of $\mathbf{x}$

$$\mathbf{w}_k = \arg\max_{\|\mathbf{w}\|=1} \frac{1}{m}\sum_{i=1}^{m}\{[\mathbf{w}^T(\mathbf{x}_i - \underbrace{\sum_{j=1}^{k-1}\mathbf{w}_j\mathbf{w}_j^T\mathbf{x}_i)}_{\mathbf{x}' \text{ PCA reconstruction}}]^2\}$$

kth PCA vector

We maximize the variance of the projection in the residual subspace



PCA algorithm II (sample covariance matrix)

- Given data $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, compute covariance matrix Σ

$$\boxed{\Sigma = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T}$$

where

$$\boxed{\bar{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i}$$

- **PCA** basis vectors = the eigenvectors of Σ
- Larger eigenvalue $\Rightarrow$ more important eigenvectors

PCA algorithm II

PCA algorithm($\mathbf{X}$, k): top k eigenvalues/eigenvectors

% $\underline{\mathbf{X}}$ = N × m data matrix,
% … each data point $\mathbf{x}_i$ = column vector, i=1..m

- $$\underline{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{x}_i$$
- $\mathbf{X}$ ← subtract mean $\underline{\mathbf{x}}$ from each column vector $\mathbf{x}_i$ in $\underline{\mathbf{X}}$
- $\Sigma \leftarrow \mathbf{X}\mathbf{X}^T$ … covariance matrix of $\mathbf{X}$
- $\{ \lambda_i, \mathbf{u}_i \}_{i=1..N}$ = eigenvectors/eigenvalues of Σ
  ... $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$
- Return $\{ \lambda_i, \mathbf{u}_i \}_{i=1..k}$
  % top k principle components

PCA algorithm III (SVD of the data matrix)

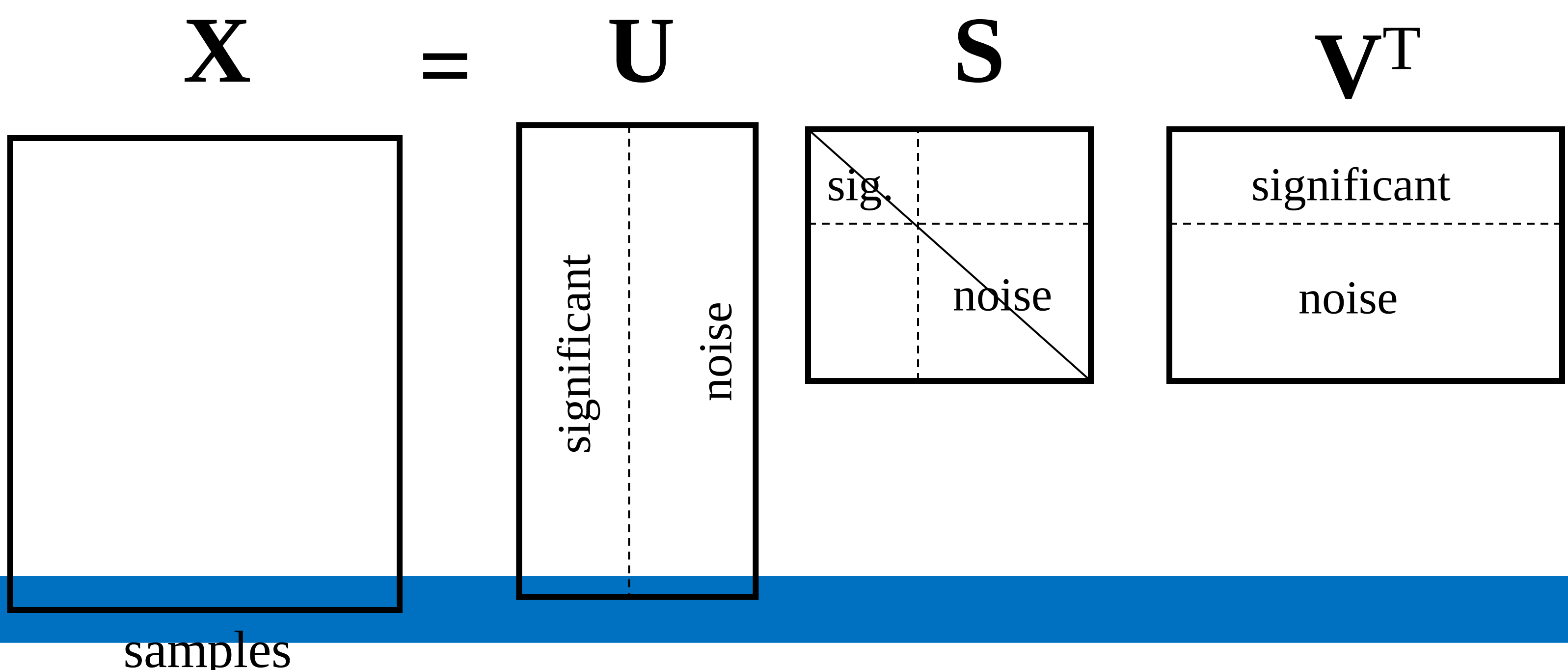
Singular Value Decomposition of the **centered** data matrix **X**.

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m] \in \mathbb{R}^{N \times m},$$

m: number of instances,

N: dimension

$$\mathbf{X}_{\text{features} \times \text{samples}} = \mathbf{USV}^T$$



PCA algorithm III

- **Columns of U**
  - the principal vectors, $\{ \mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)} \}$
  - orthogonal and has unit norm – so $U^T U = I$
  - Can reconstruct the data using linear combinations of $\{ \mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)} \}$

- **Matrix S**
  - Diagonal
  - Shows importance of each eigenvector

- **Columns of V^T**
  - The coefficients for reconstructing the samples

Face recognition

Challenge: Facial Recognition

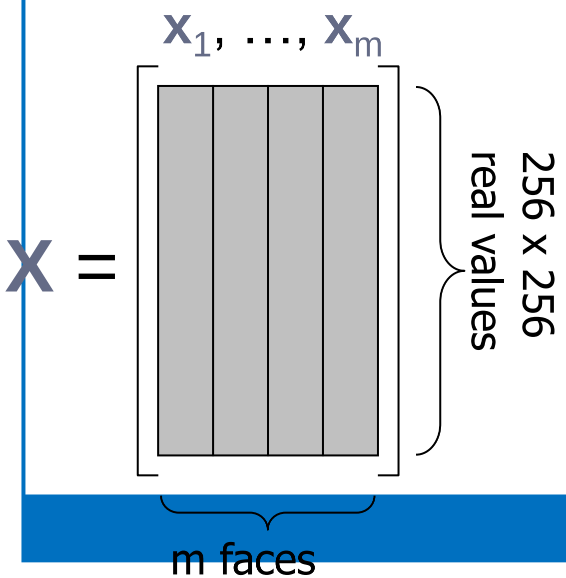
- Want to identify specific person, based on facial image
- Robust to glasses, lighting,…

$\Rightarrow$ Can't just use the given 256 x 256 pixels

Applying PCA: Eigenfaces

Method A: Build a PCA subspace for each person and check which subspace can reconstruct the test image the best

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



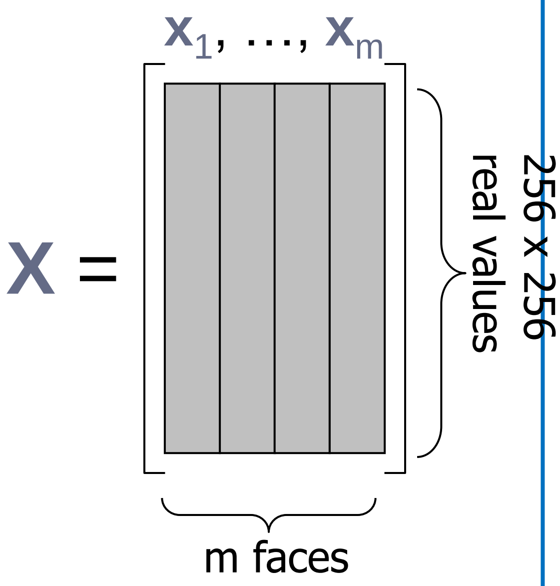
- Example data set: Images of faces
  - Famous Eigenface approach [Turk & Pentland], [Sirovich & Kirby]
- Each face $\mathbf{x}$ is …
- 256×256 values (luminance at location)
  - $\mathbf{x}$ in $\Re^{256\times256}$ (view as 64K dim vector)
- Form $\mathbf{X} = [\, \mathbf{x}_1, \ldots, \mathbf{x}_m \,]$ **centered** data mtx
- Compute $\Sigma = \mathbf{X}\mathbf{X}^T$
- Problem: Σ is $64K \times 64K$ … HUGE!!!

Computational Complexity

- Suppose m instances, each of size N
  - Eigenfaces: m=500 faces, each of size N=64K
- Given N×N covariance matrix Σ, can compute
  - all N eigenvectors/eigenvalues in $O(N^3)$
  - first k eigenvectors/eigenvalues in $O(k\ N^2)$
- But if N=64K, EXPENSIVE!

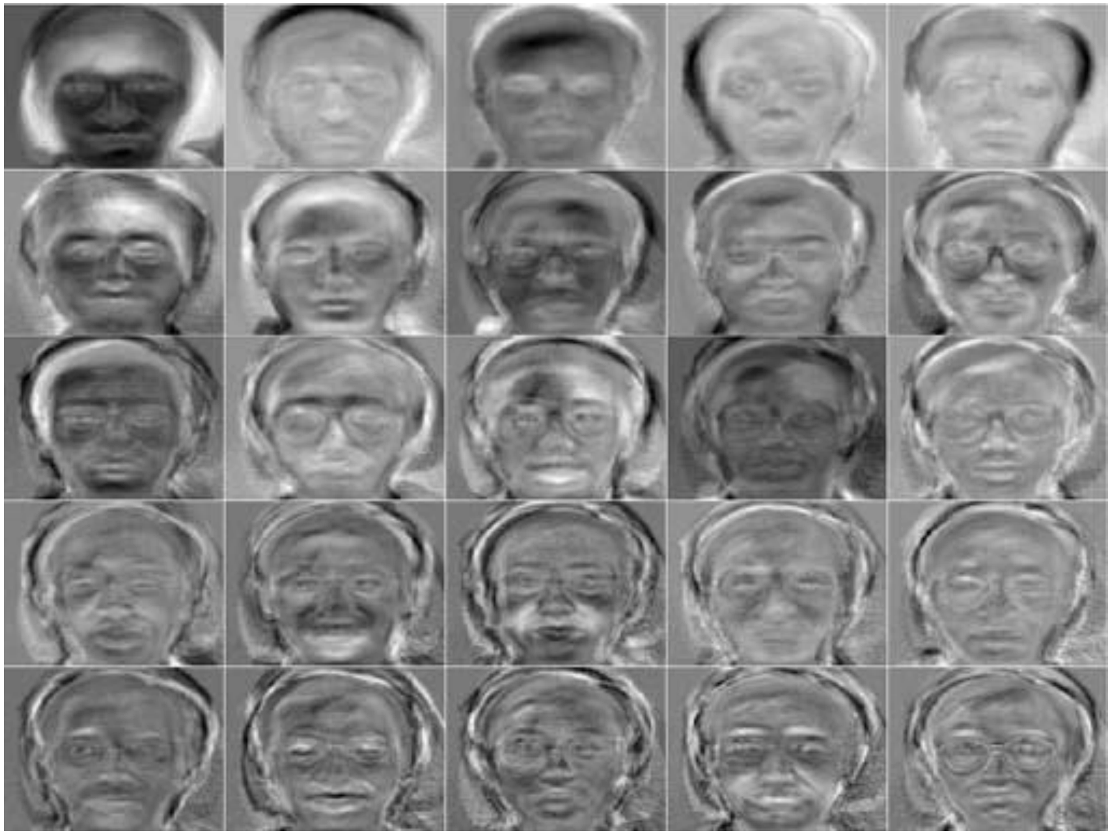
A Clever Workaround

- Note that $m << 64K$
- Use $\mathbf{L}=\mathbf{X}^T\mathbf{X}$ instead of $\Sigma=\mathbf{X}\mathbf{X}^T$
- If $\mathbf{v}$ is eigenvector of $\mathbf{L}$
  then $\mathbf{X}\mathbf{v}$ is eigenvector of Σ



Proof:
$$\mathbf{L}\ \mathbf{v} = \gamma\ \mathbf{v}$$
$$\mathbf{X}^T\mathbf{X}\ \mathbf{v} = \gamma\ \mathbf{v}$$
$$\mathbf{X}\ (\mathbf{X}^T\mathbf{X}\ \mathbf{v}) = \mathbf{X}(\gamma\ \mathbf{v}) = \gamma\ \mathbf{X}\mathbf{v}$$
$$(\mathbf{X}\mathbf{X}^T)\mathbf{X}\ \mathbf{v} = \gamma\ (\mathbf{X}\mathbf{v})$$
$$\Sigma\ (\mathbf{X}\mathbf{v}) = \gamma\ (\mathbf{X}\mathbf{v})$$

Principle Components (Method B)



Reconstructing… (Method B)

- … 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

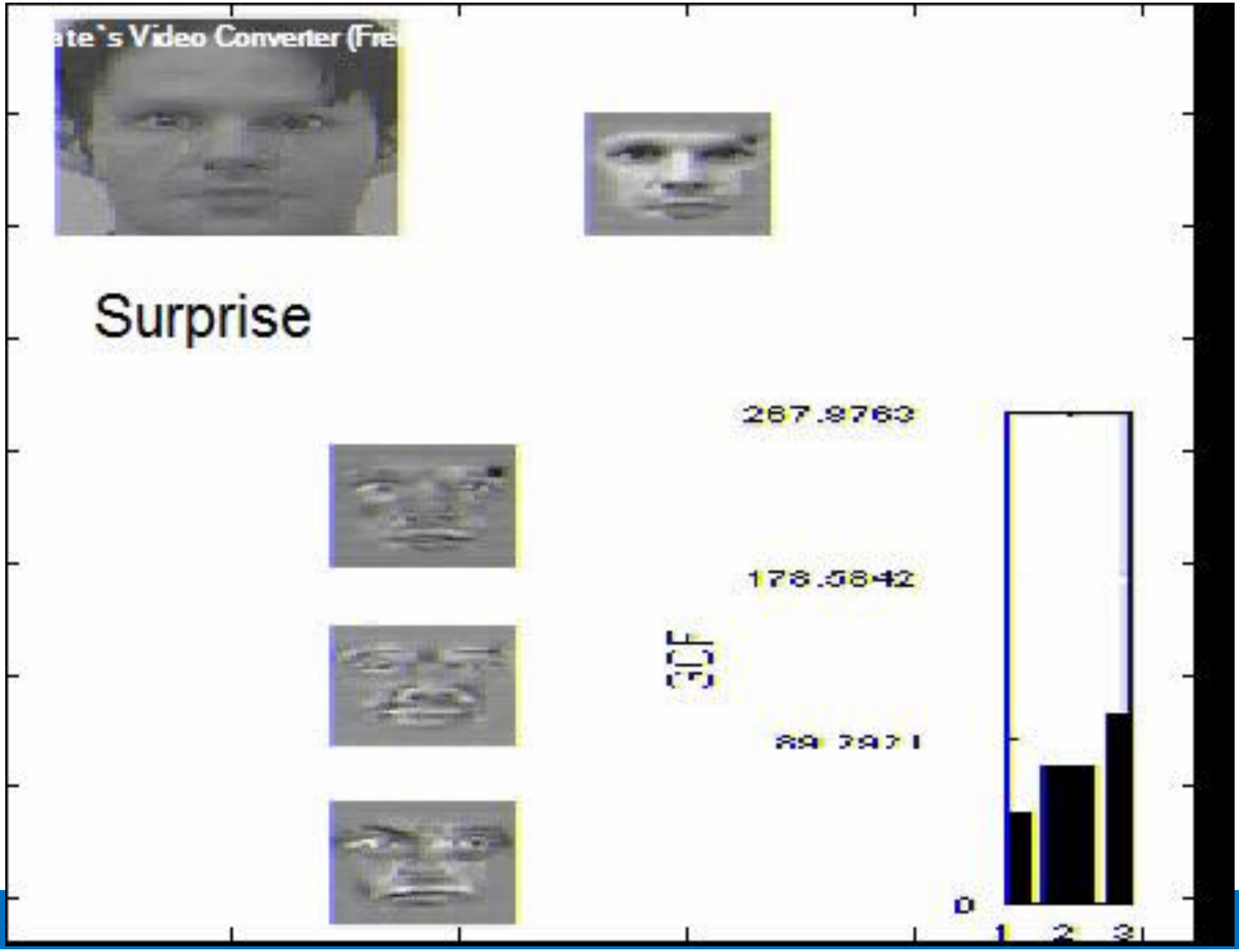
- Method is completely knowledge free
  - (sometimes this is good!)
  - Doesn’t know that faces are wrapped around 3D objects (heads)
  - Makes no effort to preserve class distinctions

Facial expression recognition

Happiness subspace (method A)

Disgust subspace (method A)

Facial Expression Recognition Movies (method A)



Facial Expression Recognition Movies (method A)

Facial Expression Recognition Movies (method A)

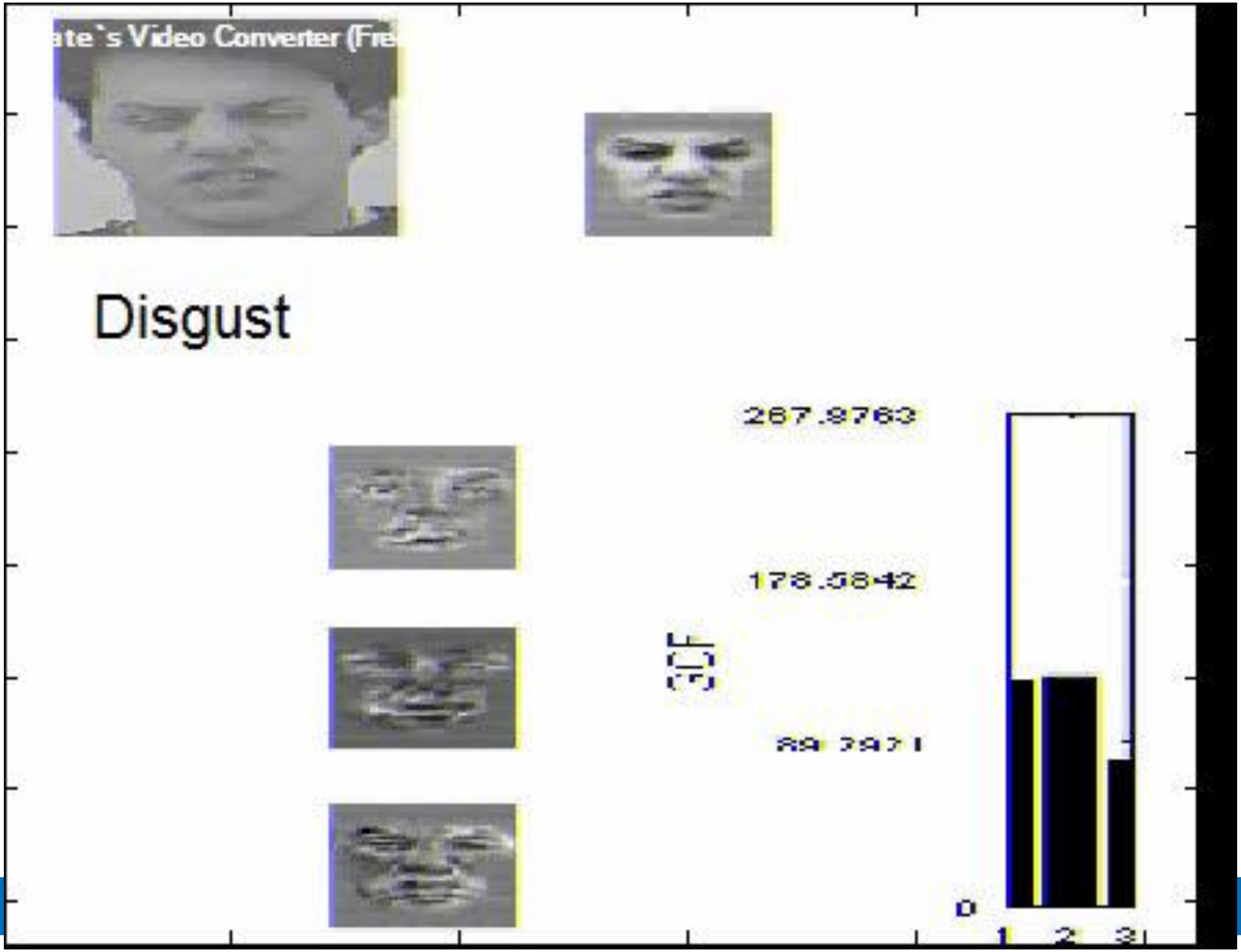
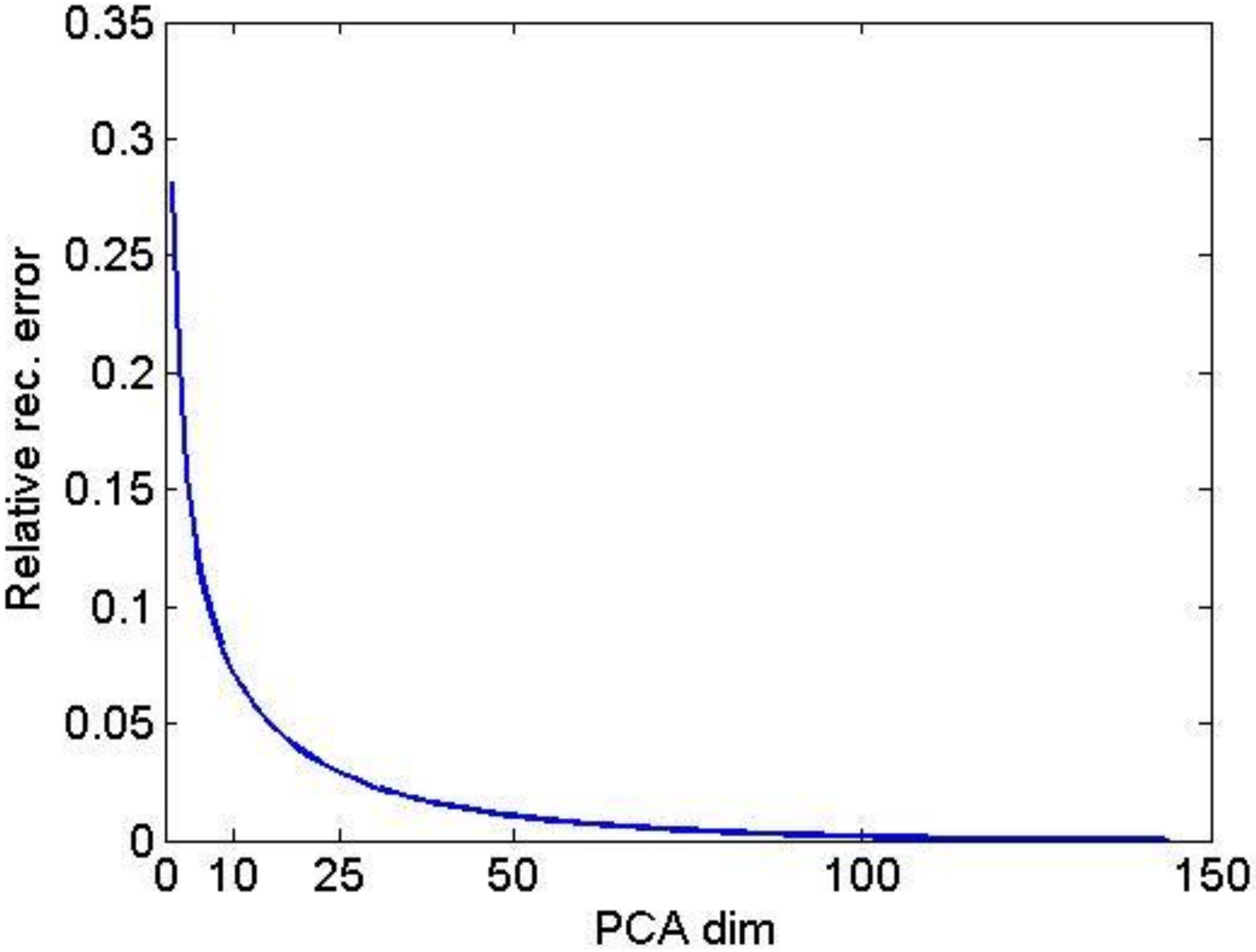


Image Compression

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

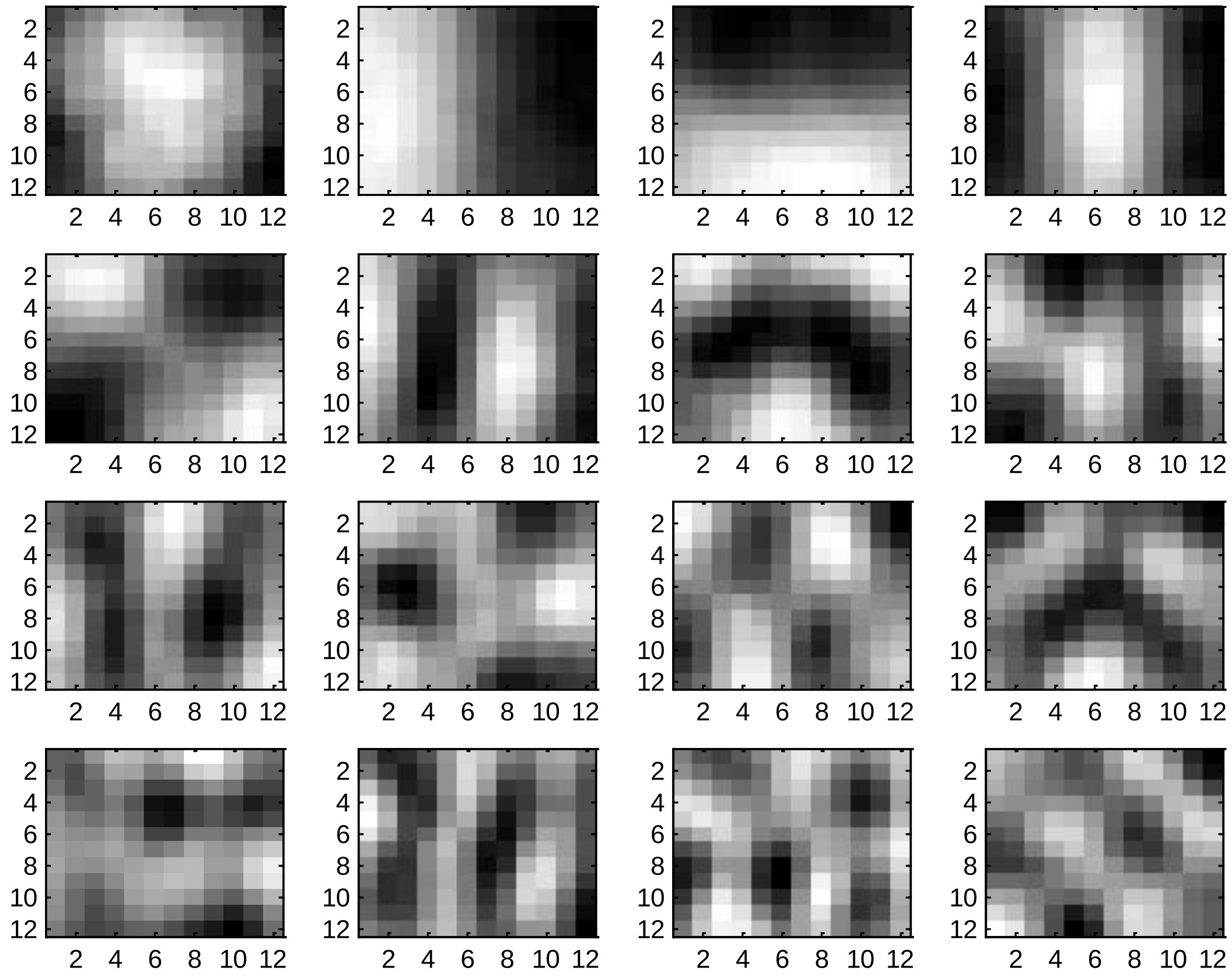
L_2 error and PCA dim



PCA compression: 144D) 60D

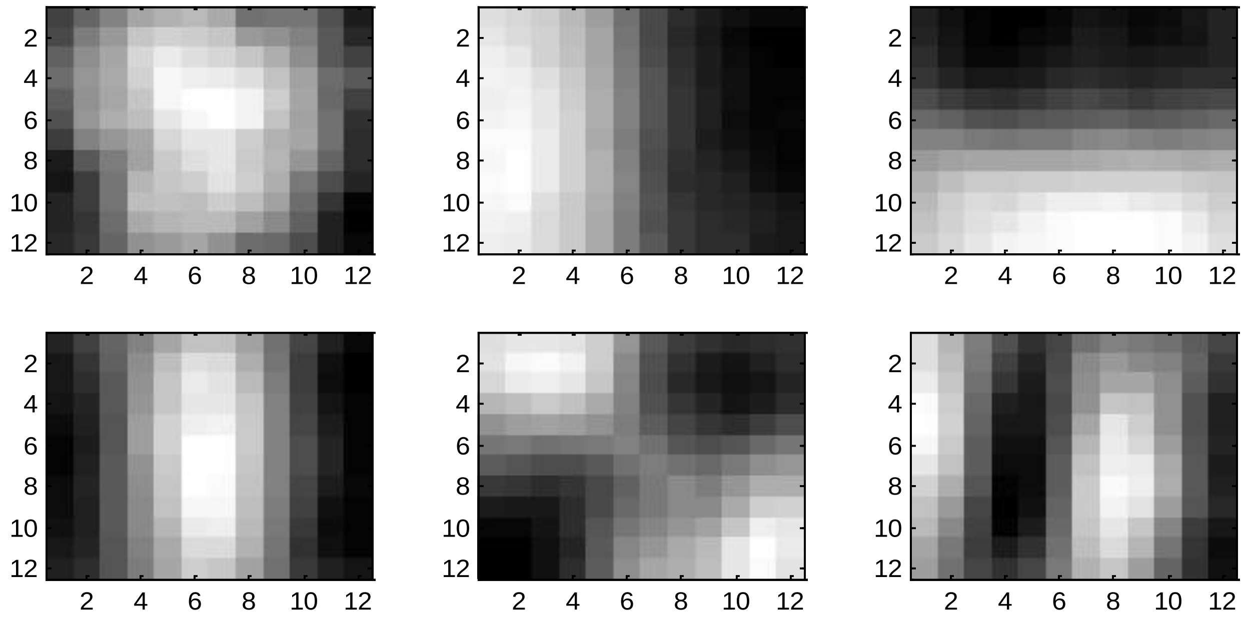
PCA compression: 144D) 16D

16 most important eigenvectors

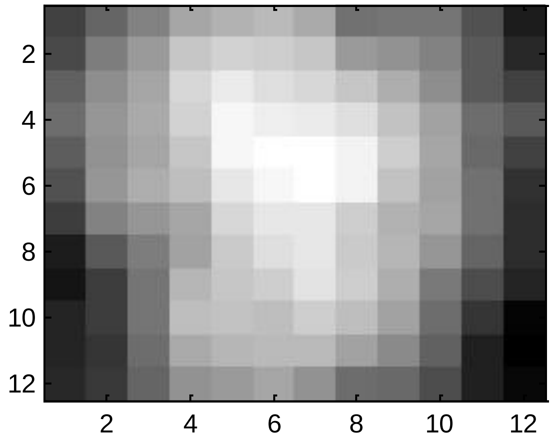


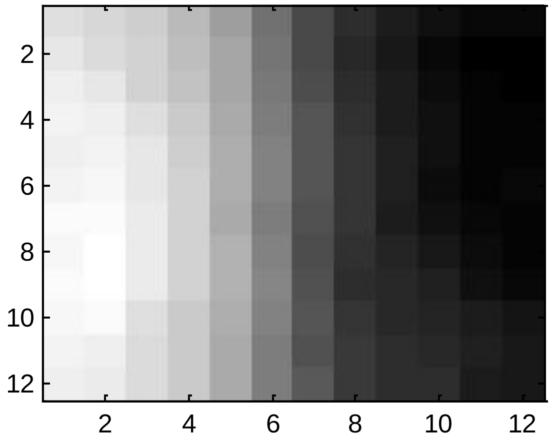
PCA compression: 144D) 6D

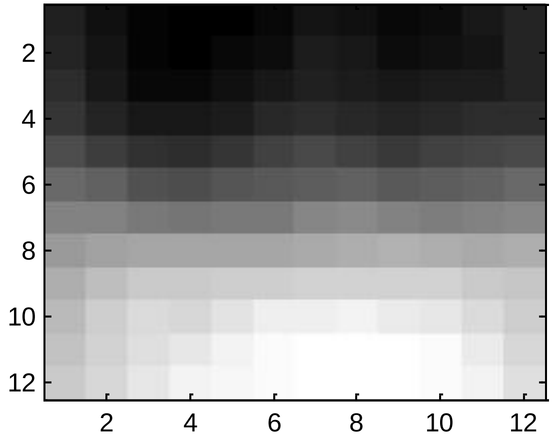
6 most important eigenvectors



3 most important eigenvectors

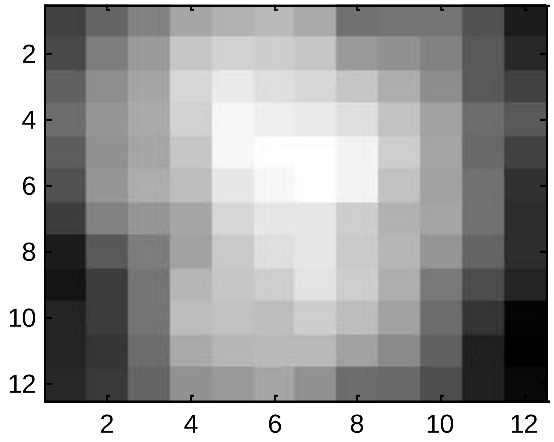


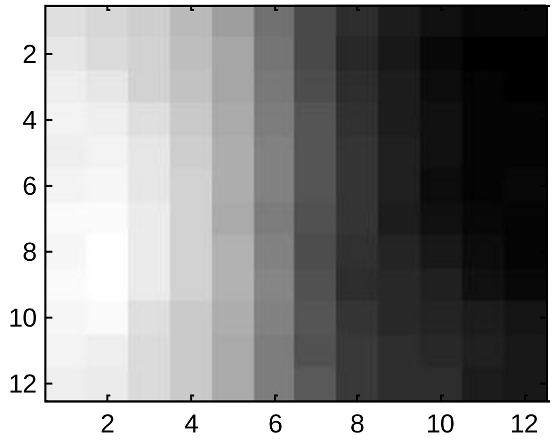


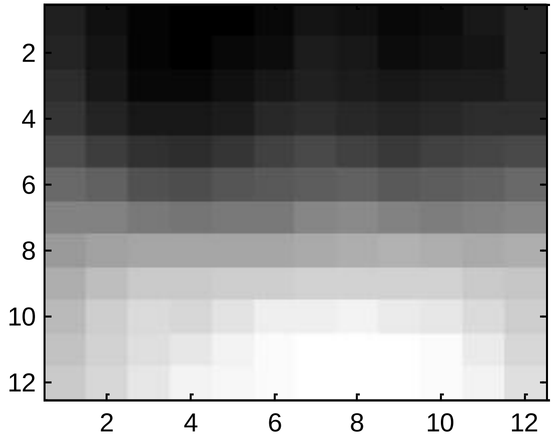


PCA compression: 144D) 3D

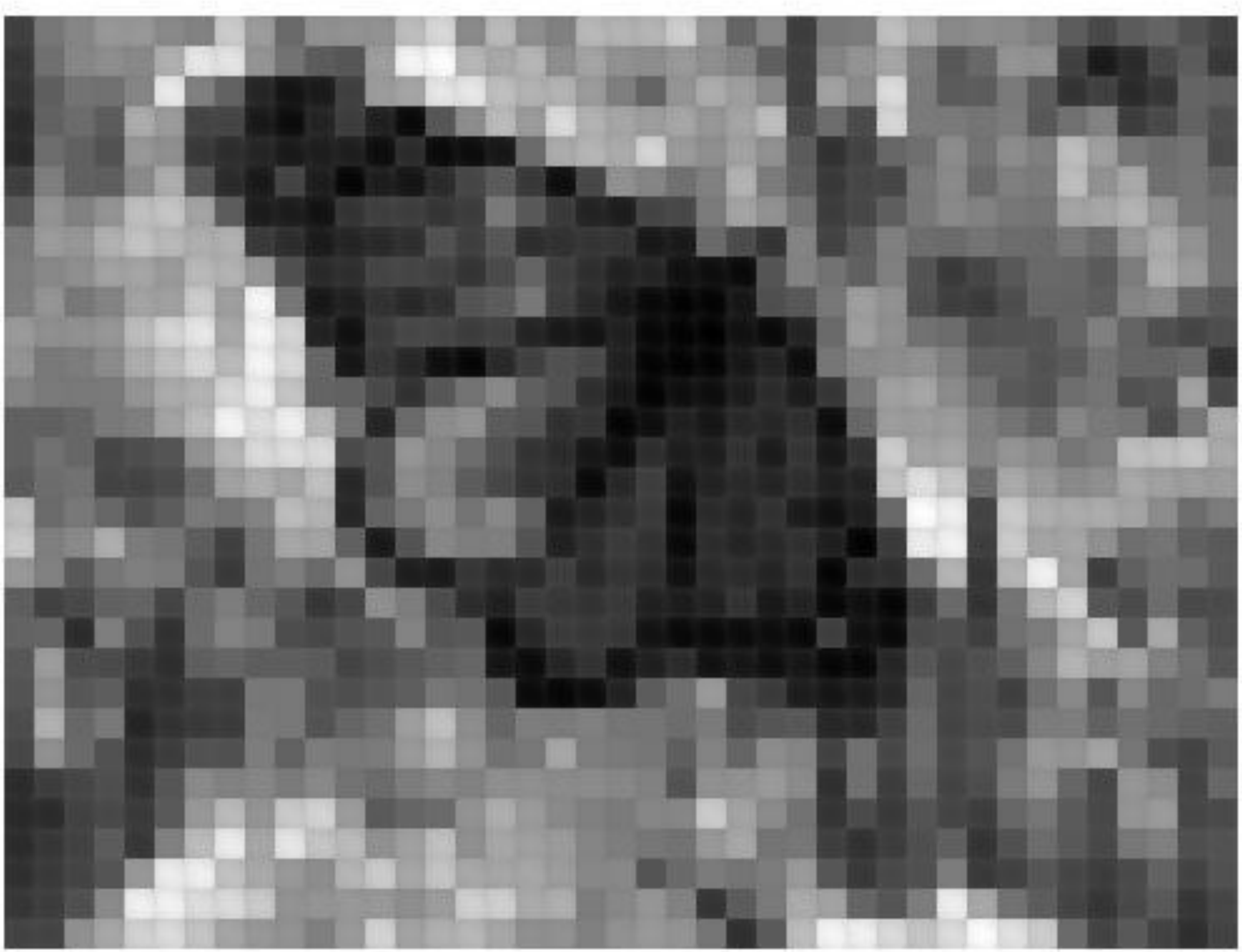
3 most important eigenvectors



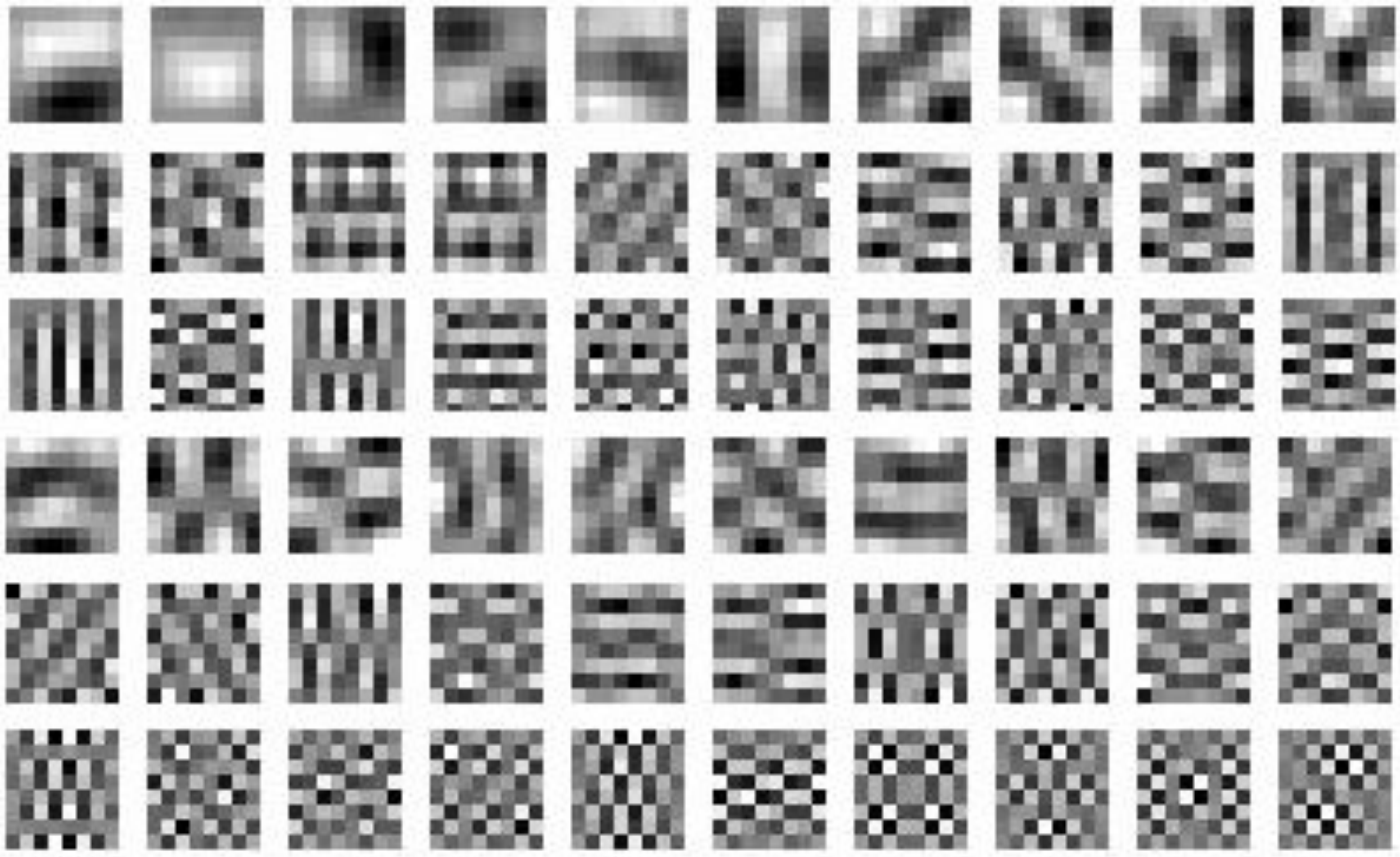




PCA compression: 144D) 1D

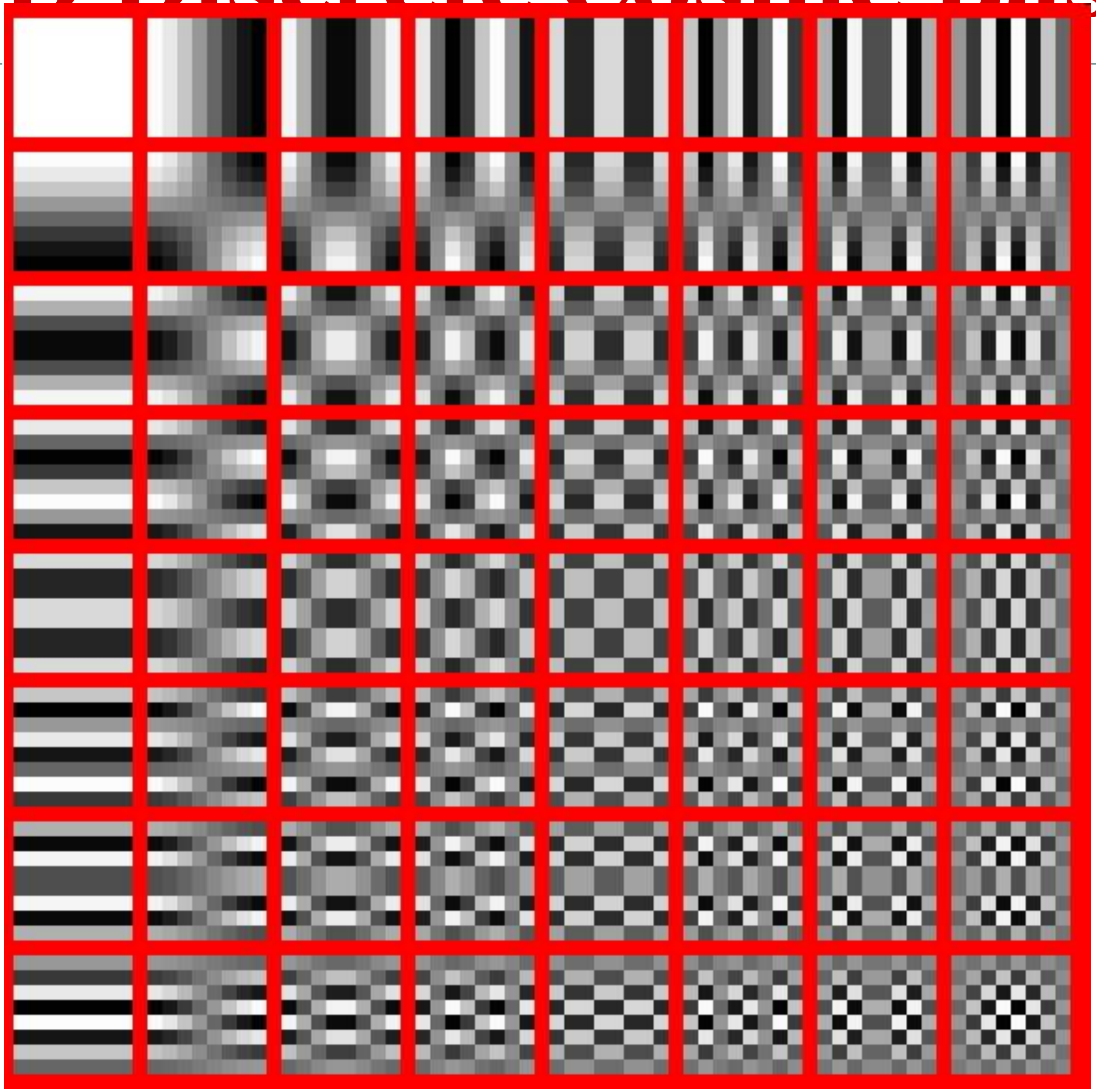


60 most important eigenvectors



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

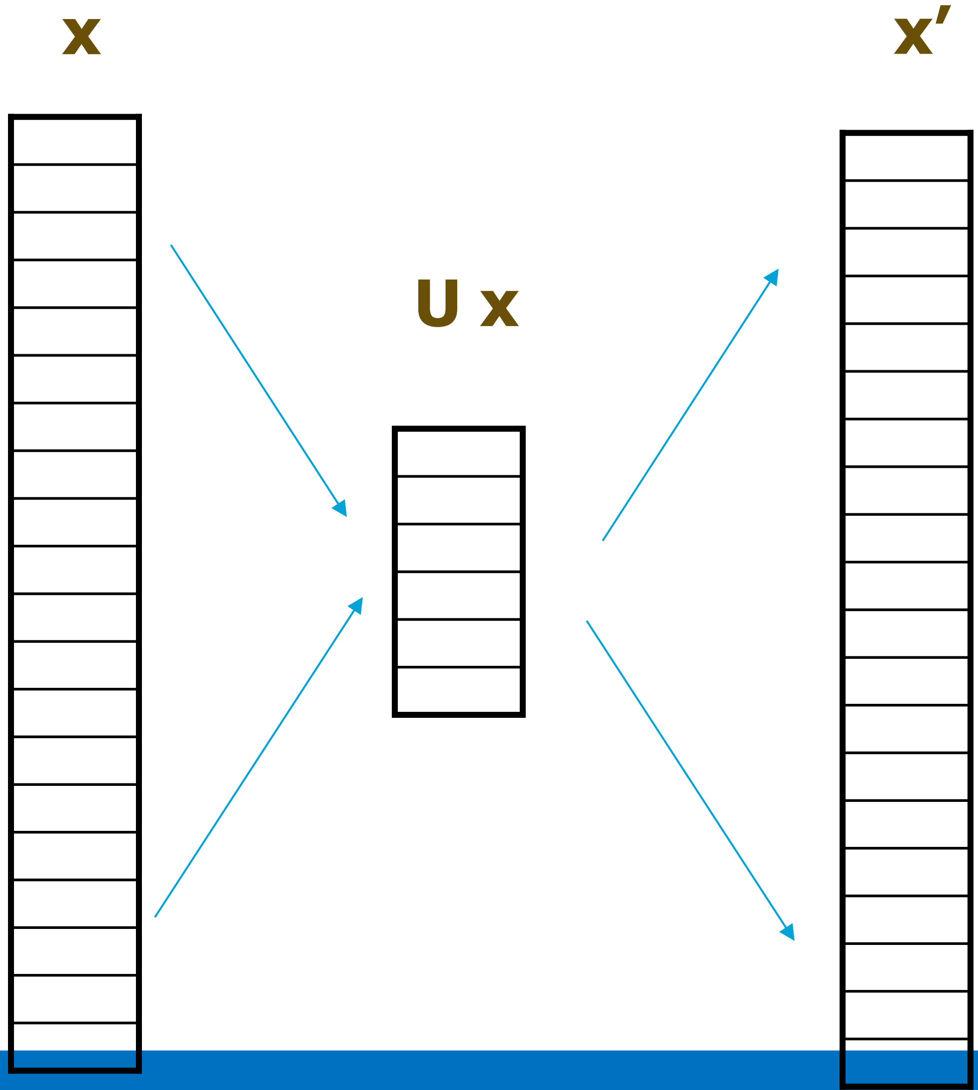
2D Discrete Cosine Basis



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

Noise Filtering

Noise Filtering, Auto-Encoder…

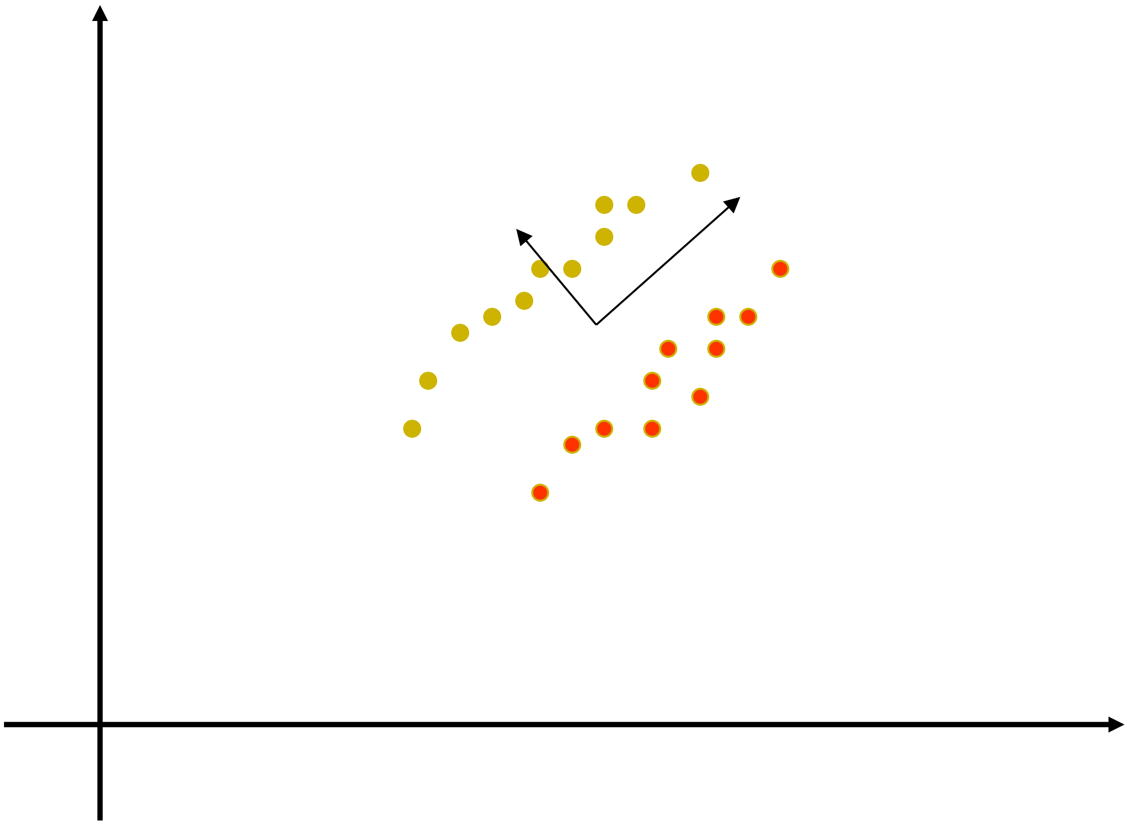


Noisv image

Denoised image using 15 PCA components

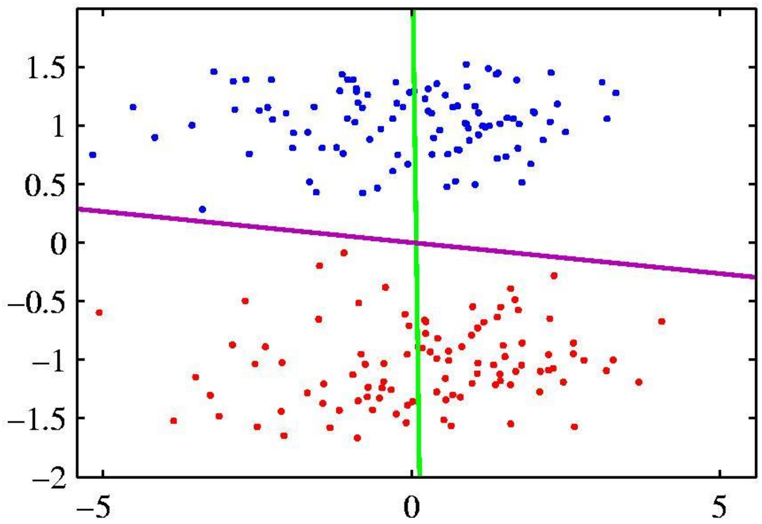
PCA Shortcomings

PCA, a Problematic Data Set



PCA doesn't know labels!

PCA vs Fisher Linear Discriminant



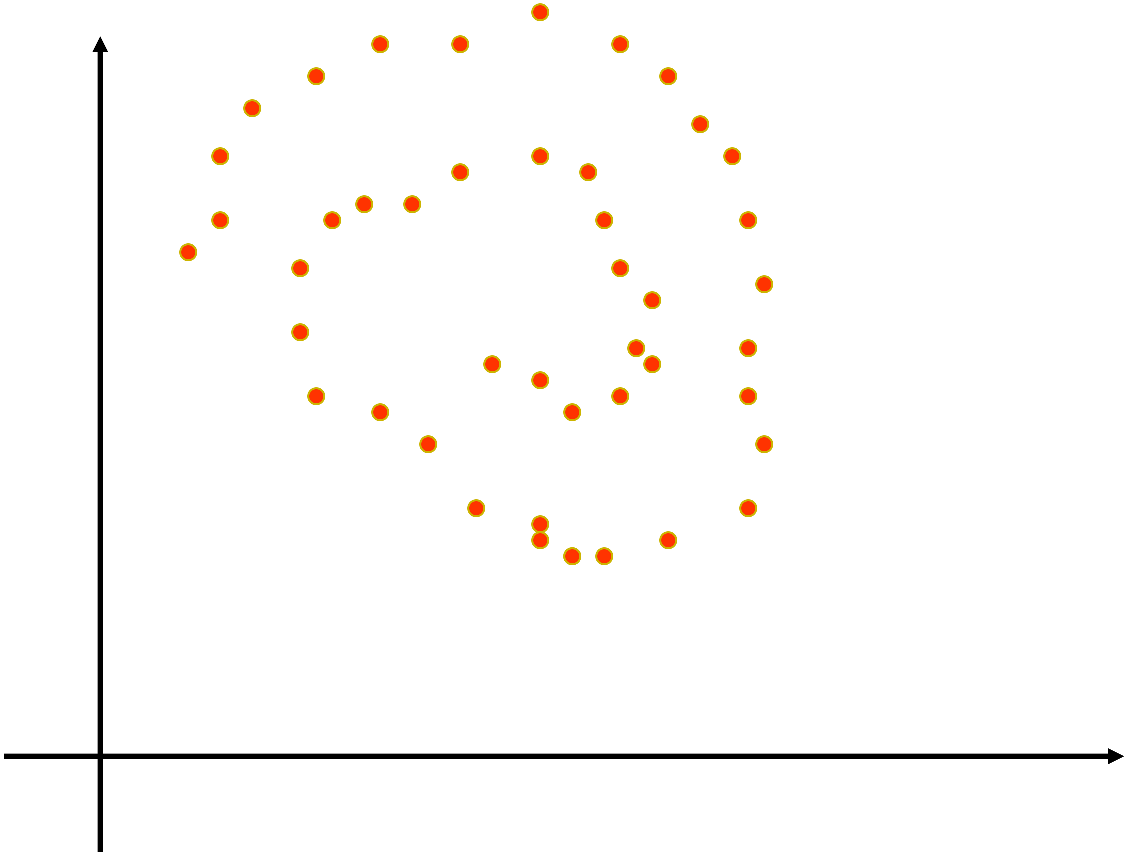
- PCA maximizes variance, *independent of class*

  ⇒ magenta

- FLD attempts to separate classes

  ⇒ green line

PCA, a Problematic Data Set



PCA cannot capture NON-LINEAR structure!

PCA Conclusions

- PCA
  - finds orthonormal basis for data
  - Sorts dimensions in order of “importance”
  - Discard low significance dimensions
- Uses:
  - Get compact description
  - Ignore noise
  - Improve classification (hopefully)
- Not magic:
  - Doesn’t know class labels
  - Can only capture linear variations
- One of many tricks to reduce dimensionality!

Applications of PCA

- *Eigenfaces for recognition*. Turk and Pentland. 1991.
- *Principal Component Analysis for clustering gene expression data*. Yeung and Ruzzo. 2001.
- *Probabilistic Disease Classification of Expression-Dependent Proteomic Data from Mass Spectrometry of Human Serum*. Lilien. 2003.

PCA for image compression


d=1


d=2


d=4


d=8

d=16

d=32

d=64

d=100


Original Image