

Principal Component Analysis

Source: *Introduction to Machine Learning*

Computing Science 466 / 551

R. Greiner, B. Póczos, University of Alberta

<https://webdocs.cs.ualberta.ca/~greiner/C-466/>



Contents

- Motivation
- PCA algorithms
- Applications
- PCA theory

Some of these slides are taken from

- Karl Booksh Research group
- Tom Mitchell
- Ron Parr



Data Visualization

Example:

- Given 53 blood and urine measurements (features) from 65 individuals
- How can we visualize the measurements?

Data Visualization

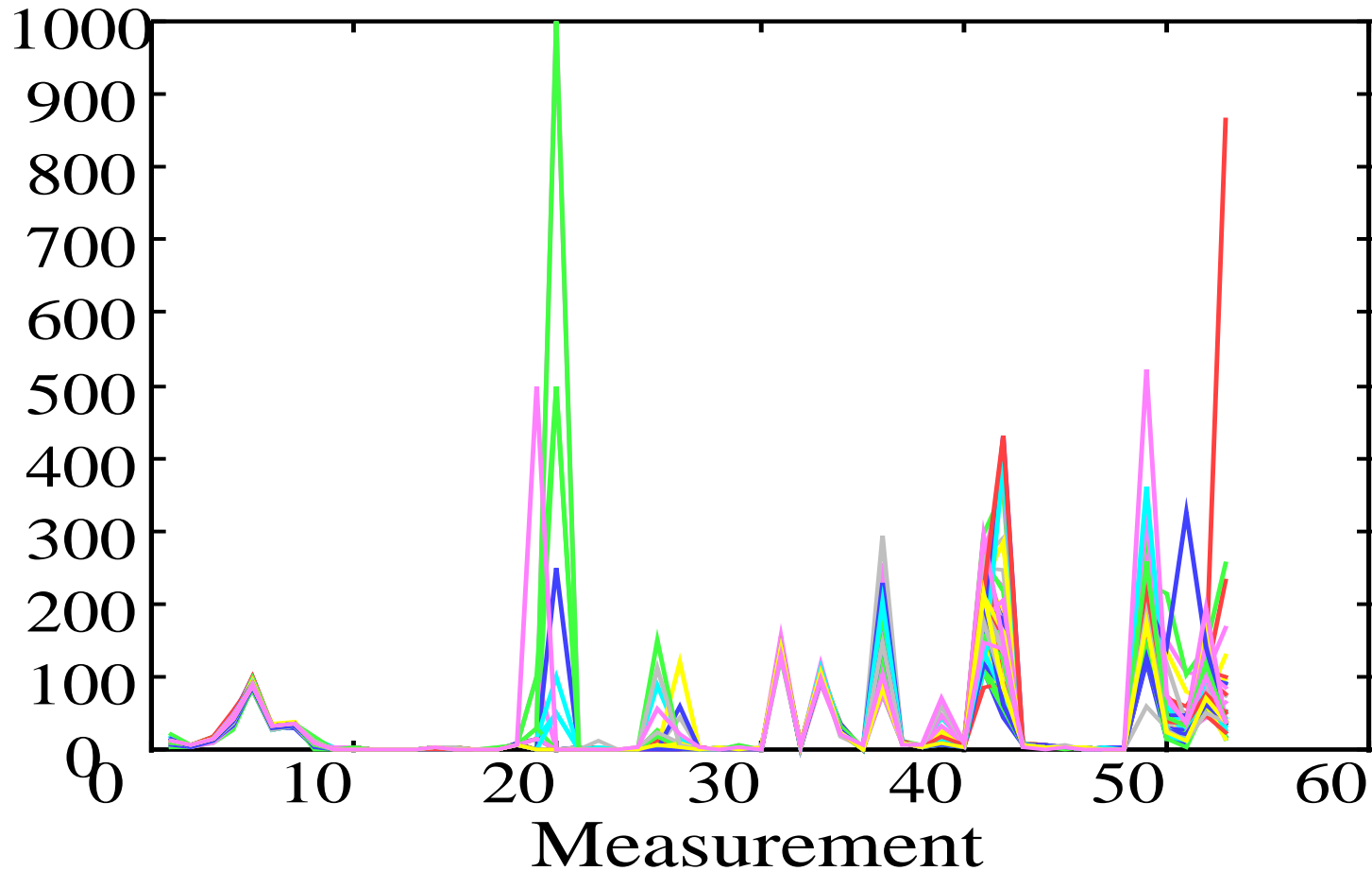
- Matrix format (65x53)

	H-WBC	H-RBC	H-Hgb	H-Hct	H-MCV	H-MCH	H-MCHC
A1	8.0000	4.8200	14.1000	41.0000	85.0000	29.0000	34.0000
A2	7.3000	5.0200	14.7000	43.0000	86.0000	29.0000	34.0000
A3	4.3000	4.4800	14.1000	41.0000	91.0000	32.0000	35.0000
A4	7.5000	4.4700	14.9000	45.0000	101.0000	33.0000	33.0000
A5	7.3000	5.5200	15.4000	46.0000	84.0000	28.0000	33.0000
A6	6.9000	4.8600	16.0000	47.0000	97.0000	33.0000	34.0000
A7	7.8000	4.6800	14.7000	43.0000	92.0000	31.0000	34.0000
A8	8.6000	4.8200	15.8000	42.0000	88.0000	33.0000	37.0000
A9	5.1000	4.7100	14.0000	43.0000	92.0000	30.0000	32.0000

Difficult to see the correlations between the features...

Data Visualization

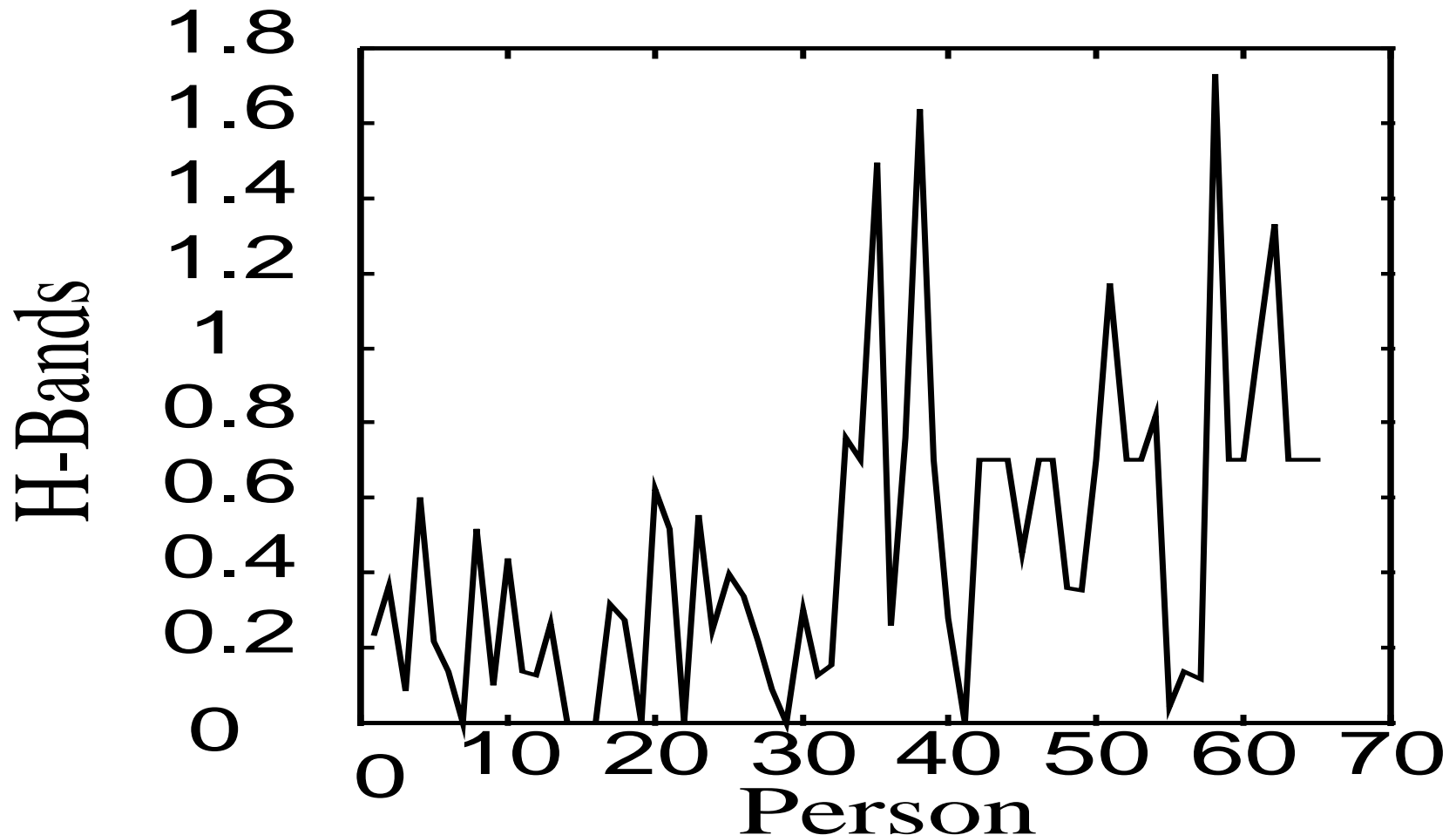
- Spectral format (65 pictures, one for each person)



Difficult to compare the different patients...

Data Visualization

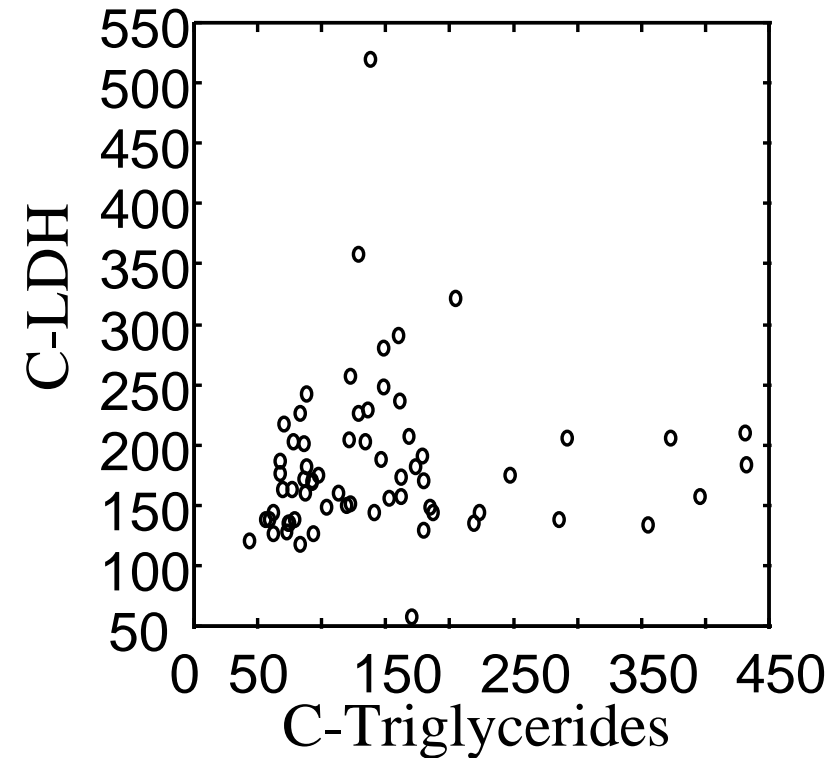
- Spectral format (53 pictures, one for each feature)



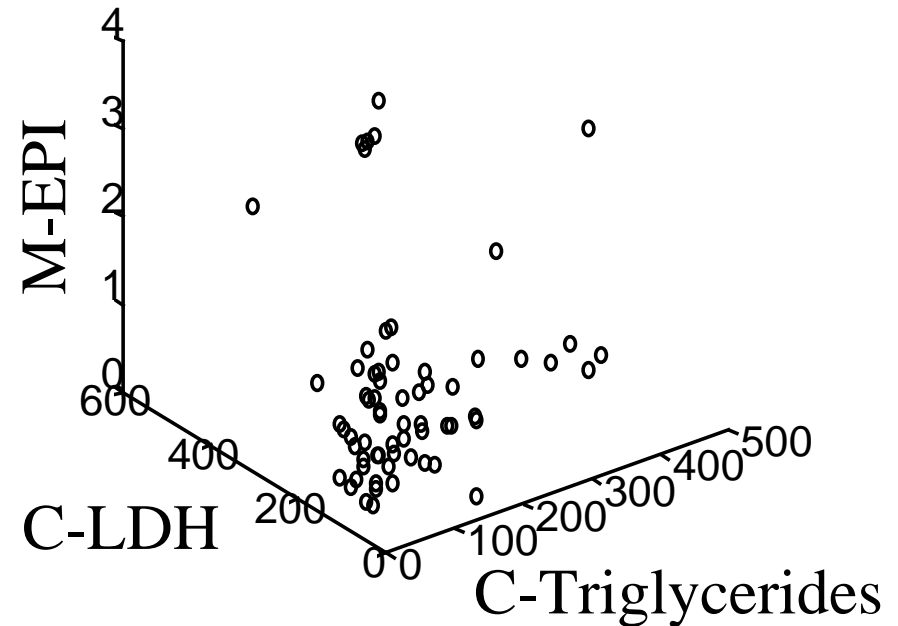
Difficult to see the correlations between the features...

Data Visualization

Bi-variate



Tri-variate



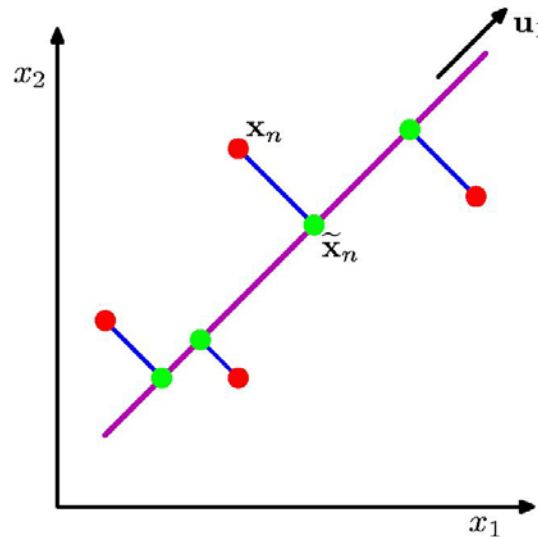
How can we visualize the other variables???

... difficult to see in 4 or higher dimensional spaces...

Data Visualization

- Is there a better representation than the coordinate axes?
- Is it really necessary to show all the 53 dimensions?
 - ... what if there are strong correlations between some of the features?
- How could we find the *smallest* subspace of the 53-D space that keeps the *most information* about the original data?
- A solution: **Principal Component Analysis**

Principal 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 points and their projections (the blue segments)

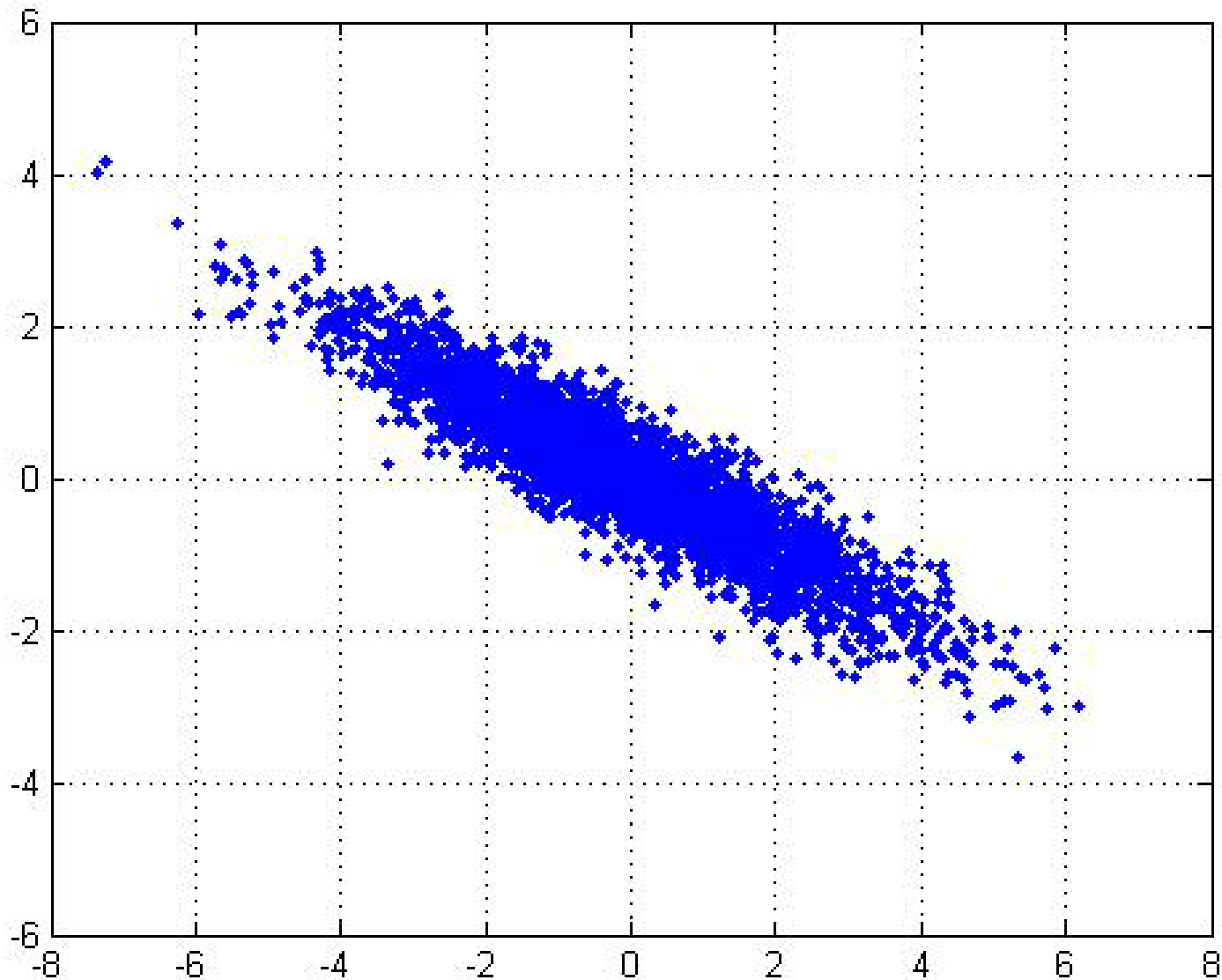
PCA: the idea

- 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

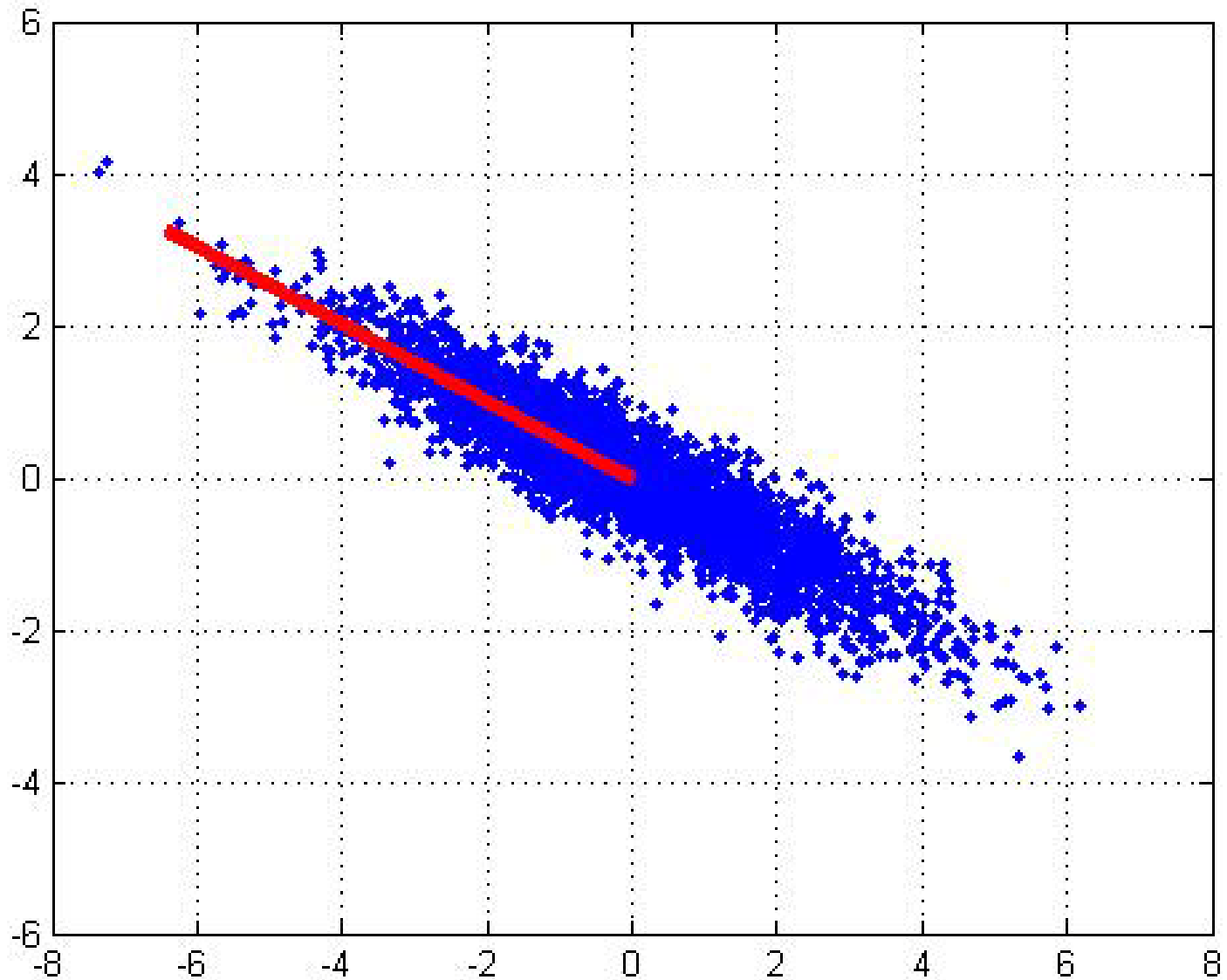
The Principal Components

- **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**

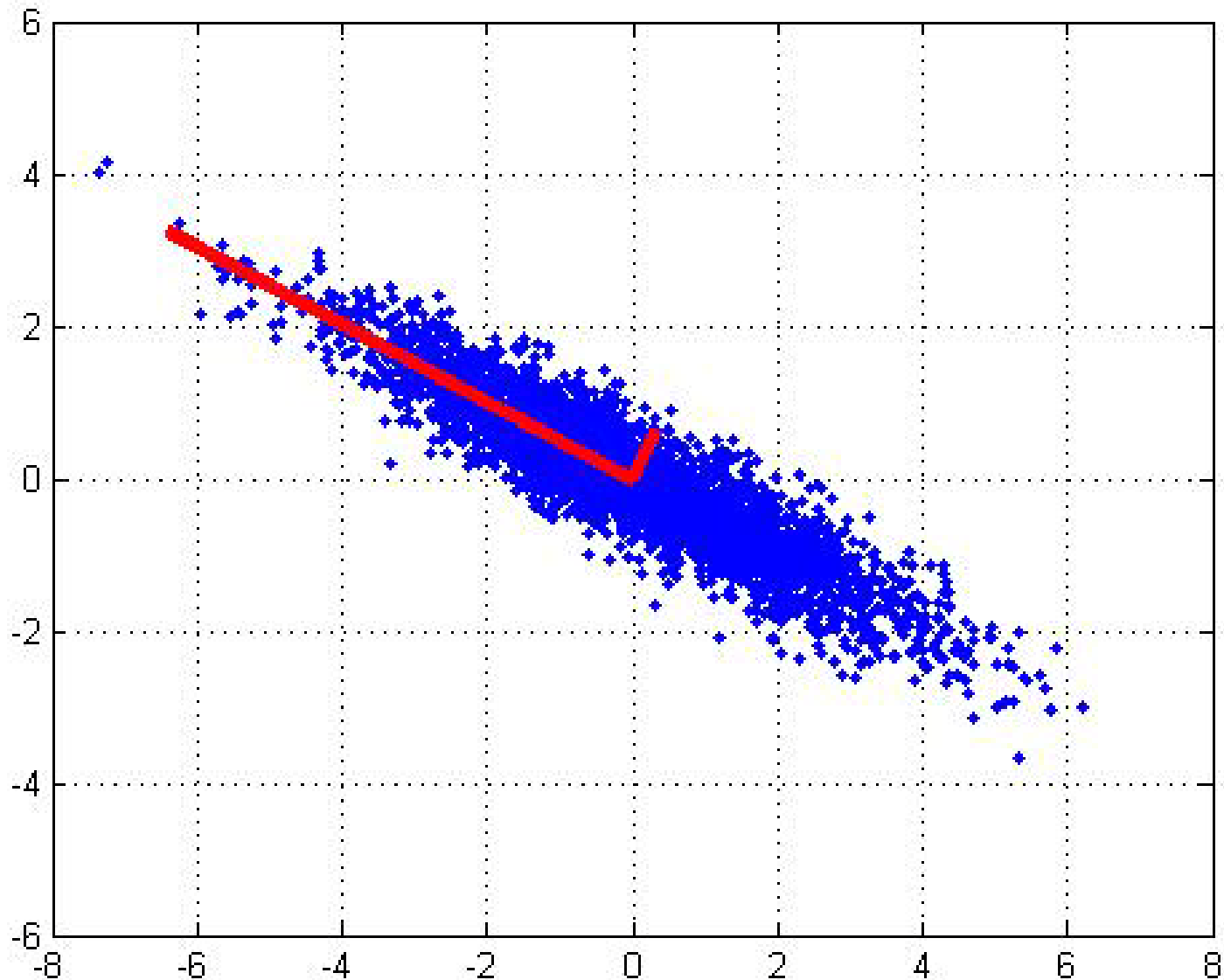
2D Gaussian dataset



1st PCA axis



2nd PCA axis



PCA: a sequential algorithm

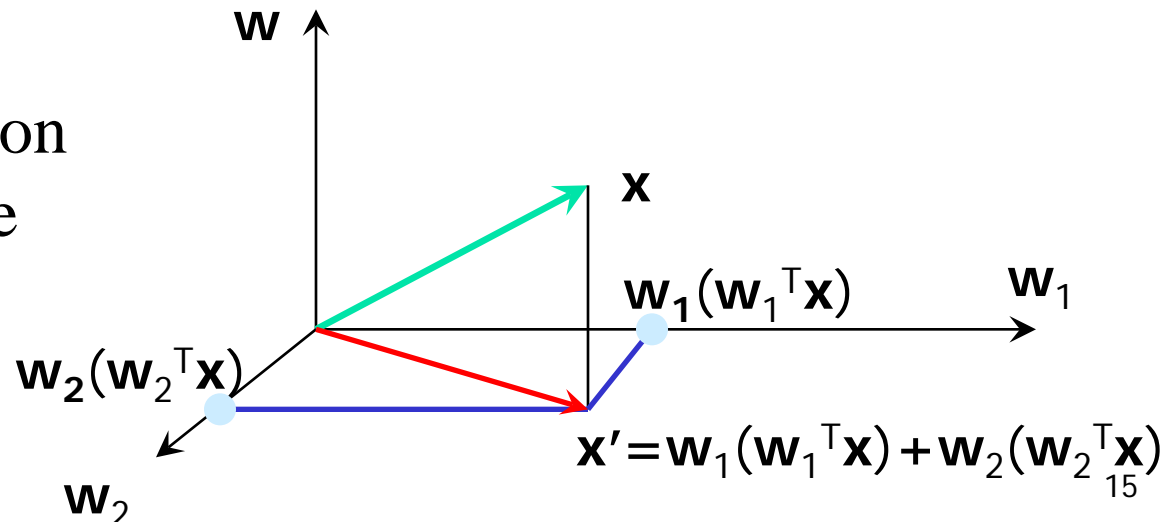
Given the **centered** data $\{\mathbf{x}_1, \dots, \mathbf{x}_m\}$, compute the principal vectors:

$$\mathbf{w}_1 = \arg \max_{\|\mathbf{w}\|=1} \frac{1}{m} \sum_{i=1}^m \{(\mathbf{w}^T \mathbf{x}_i)^2\} \quad \text{1st PCA vector}$$

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

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

We maximize the variance of the projection in the residual subspace



PCA algorithm

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

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

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

PCA algorithm

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

% \mathbf{X} = $N \times m$ data matrix,

% ... each data point \mathbf{x}_i = column vector, $i=1..m$

- $\underline{\mathbf{x}} = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i$
- $\mathbf{X} \leftarrow$ subtract mean $\underline{\mathbf{x}}$ from each column vector \mathbf{x}_i in \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
... $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$
- Return $\{ \lambda_i, \mathbf{u}_i \}_{i=1..k}$
% top k principal components

Proof

Let $\mathbf{x} \in \mathbb{R}^N$

Let $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m] \in \mathbb{R}^{N \times m}$,
 m : number of instances, N : dimension

Let $\mathbf{U} = \begin{pmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_N^T \end{pmatrix} \in \mathbb{R}^{N \times N}$ orthogonal matrix, $\mathbf{U}\mathbf{U}^T = \mathbf{I}_N$

$$\mathbf{y} \doteq \mathbf{U}\mathbf{x}, \quad \mathbf{x} = \mathbf{U}^T\mathbf{y} = \sum_{i=1}^N \mathbf{u}_i y_i$$

$\hat{\mathbf{x}} \doteq \sum_{i=1}^M \mathbf{u}_i y_i, \quad (M \leq N)$ approximation of \mathbf{x}
using M basis vectors only.


$$\varepsilon^2 \doteq \mathbb{E}\{\|\mathbf{x} - \hat{\mathbf{x}}\|^2\} = \frac{1}{m} \sum_{j=1}^m \|\mathbf{x}_j - \hat{\mathbf{x}}_j\|^2, \text{ average error}$$

GOAL:

$$\arg \min_{\mathbf{U}} \varepsilon^2, \text{ s.t. } \mathbf{U}^T \mathbf{U} = \mathbf{I}_N$$

$$\begin{aligned}
\varepsilon^2 &= \mathbb{E}\{\|\mathbf{x} - \hat{\mathbf{x}}\|^2\} = \mathbb{E}\{\|\sum_{i=1}^N \mathbf{u}_i y_i - \sum_{i=1}^M \mathbf{u}_i y_i\|^2\} \\
&= \mathbb{E}\{\sum_{i=M+1}^N y_i \mathbf{u}_i^T \mathbf{u}_i y_i\} = \sum_{i=M+1}^N \mathbb{E}\{y_i^2\} \\
&= \sum_{i=M+1}^N \mathbb{E}\{(\mathbf{u}_i^T \mathbf{x})(\mathbf{x}^T \mathbf{u}_i)\} \\
&= \sum_{i=M+1}^N \mathbf{u}_i^T \mathbb{E}\{\mathbf{x} \mathbf{x}^T\} \mathbf{u}_i \\
&= \sum_{i=M+1}^N \mathbf{u}_i^T \Sigma \mathbf{u}_i
\end{aligned}$$

x is centered!



Justification of Algorithm II

GOAL: $\arg \min_{\mathbf{u}_{M+1}, \dots, \mathbf{u}_N} \varepsilon^2$

Use Lagrange-multipliers for the constraints.

$$\begin{aligned} L &= \varepsilon^2 - \sum_{i=M+1}^N \lambda_i (\mathbf{u}_i^T \mathbf{u}_i - 1) \\ &= \sum_{i=M+1}^N \mathbf{u}_i^T \Sigma \mathbf{u}_i - \sum_{i=M+1}^N \lambda_i (\mathbf{u}_i^T \mathbf{u}_i - 1) \\ \frac{\partial L}{\partial \mathbf{u}_i} &= [2\Sigma \mathbf{u}_i - 2\lambda_i \mathbf{u}_i] = 0 \end{aligned}$$

$$\frac{\partial L}{\partial \mathbf{u}_i} = [2\Sigma \mathbf{u}_i - 2\lambda_i \mathbf{u}_i] = 0 \Rightarrow \Sigma \mathbf{u}_i = \lambda_i \mathbf{u}_i$$

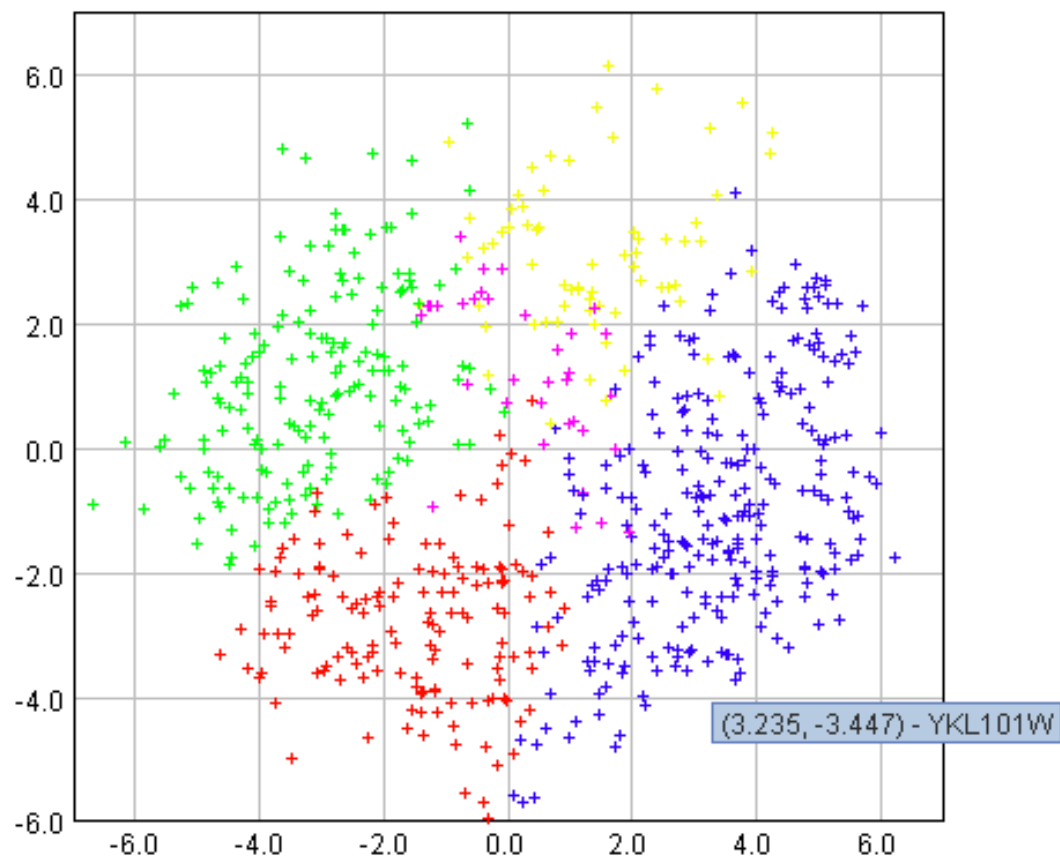
$\Rightarrow [\mathbf{u}_i, \lambda_i] = \text{eigenvector/eigenvalue of } \Sigma.$

$$\varepsilon^2 = \sum_{i=M+1}^N \mathbf{u}_i^T \Sigma \mathbf{u}_i = \sum_{i=M+1}^N \mathbf{u}_i^T \lambda_i \mathbf{u}_i = \sum_{i=M+1}^N \lambda_i$$

The error ε^2 is minimal
if $\lambda_{M+1}, \dots, \lambda_N$ are the smallest eigenvalues of Σ ,
and $\mathbf{u}_{M+1}, \dots, \mathbf{u}_N$ are the corresponding eigenvectors.

PCA Applications

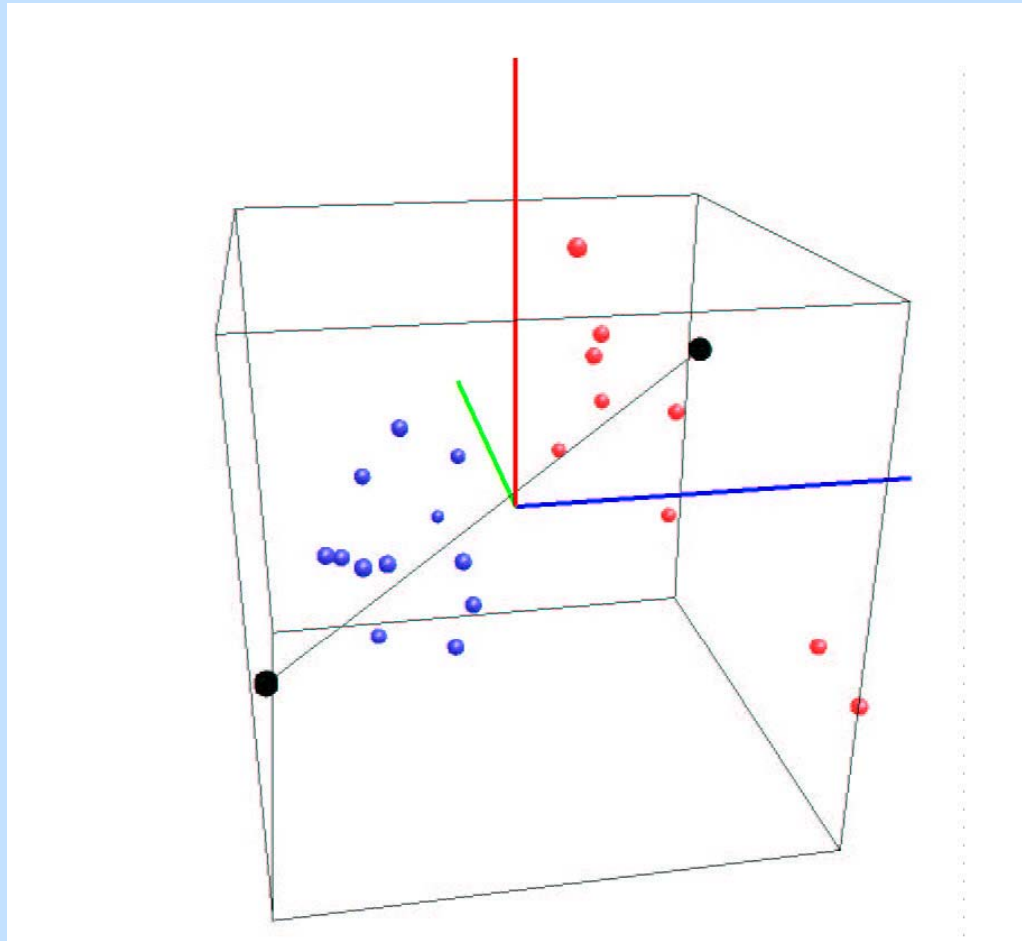
- Data Visualization
- Data Compression
- Noise Reduction
- Data Classification
- ...
- In genomics (and in general): a first step in data exploration: does my data have inner structure? Is it clusterable?



Coloring scheme:

- ☐ None
- ☒ CLICK 1.1

- Cluster 1
- Cluster 2
- Cluster 3
- Cluster 4
- Cluster 5

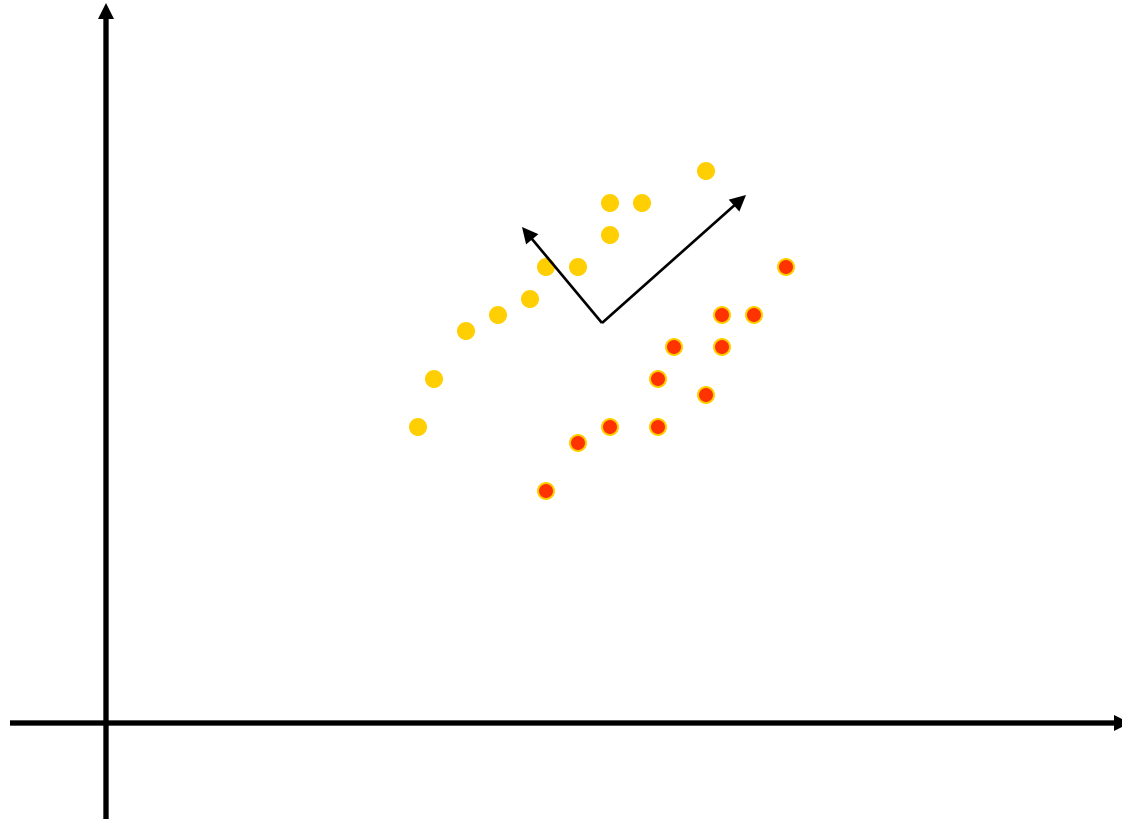


A PCA result of ALL 21 samples using 7,913 genes.
Red: good prognosis (upper right),
Blue: bad prognosis (lower left).

PROMO demo

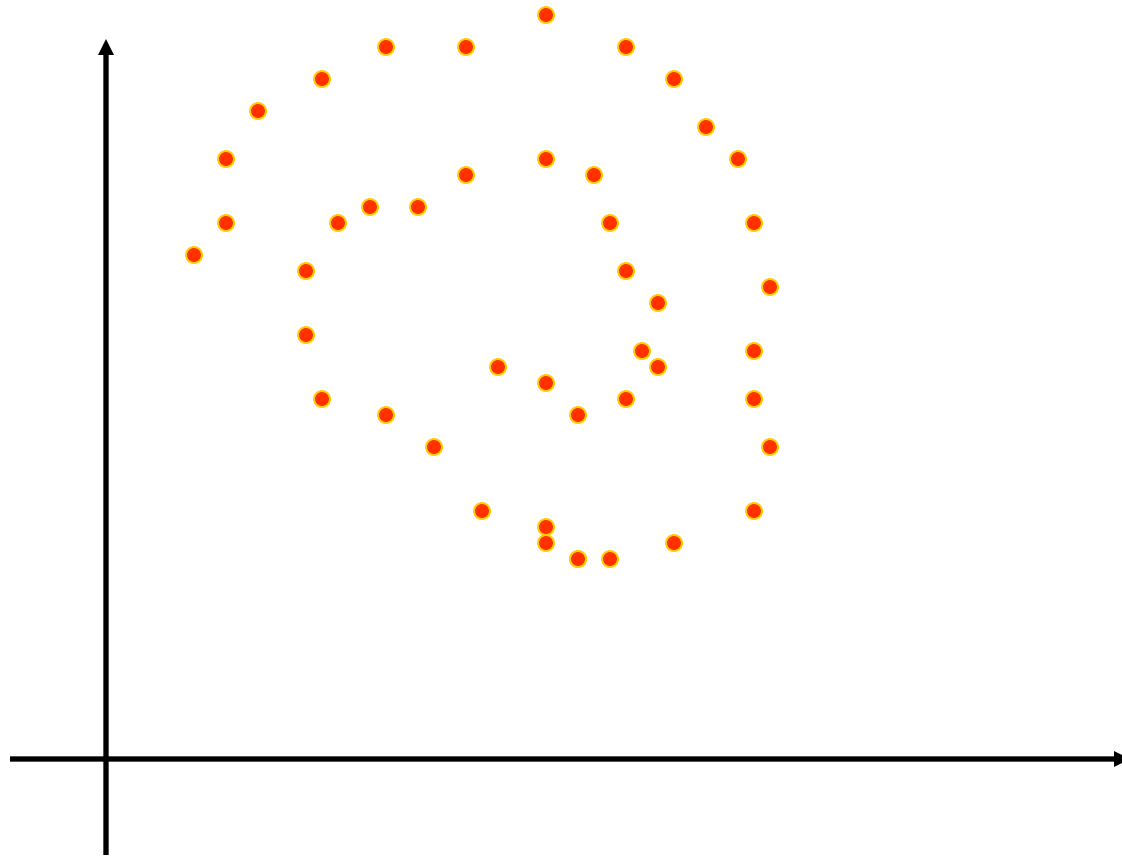


PCA shortcomings



PCA doesn't know labels

PCA shortcoming (3)



PCA cannot capture NON-LINEAR structure

Summary: PCA

- Finds orthonormal basis for data
- Sorts dimensions in order of "importance" = variance
- Discards low importance dimensions
- Uses:
 - Get compact description
 - View and assess the data
 - Ignore noise
 - Improve clustering (hopefully)
- Not magic:
 - Doesn't know class labels
 - Can only capture linear variations
- One of many tricks to reduce dimensionality!



Karl Pearson, father of mathematical statistics (1857-1936)



Invented PCA in 1901.
Rediscovered multiple times in many fields.