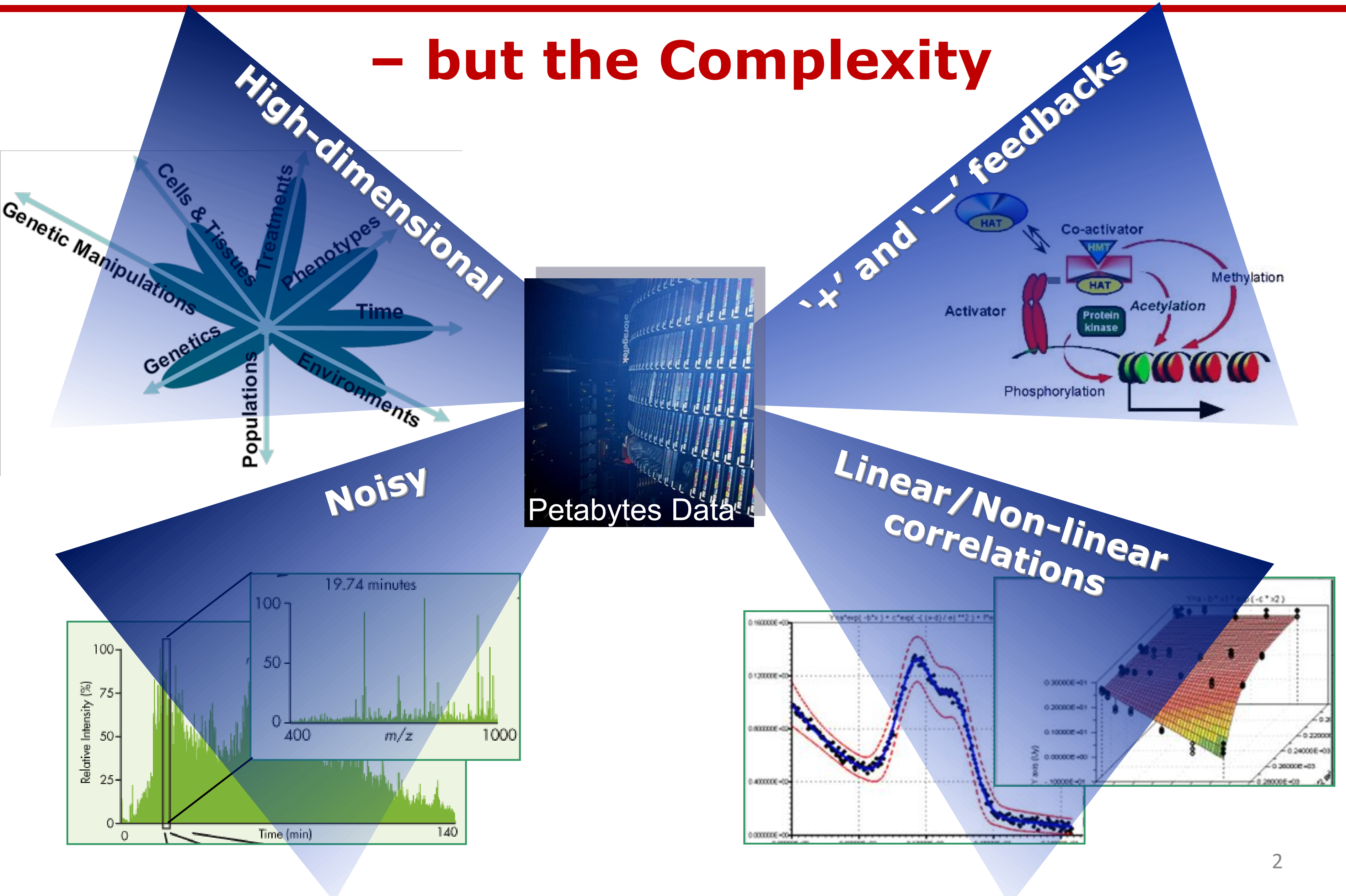
Dimensionality Reduction (DR) (Feature Extraction)

Nagiza F. Samatova, samatova@csc.ncsu.edu
Professor, Department of Computer Science
North Carolina State University

Senior Scientist, Computer Science & Mathematics Division
Oak Ridge National Laboratory

It is not just the Data Size

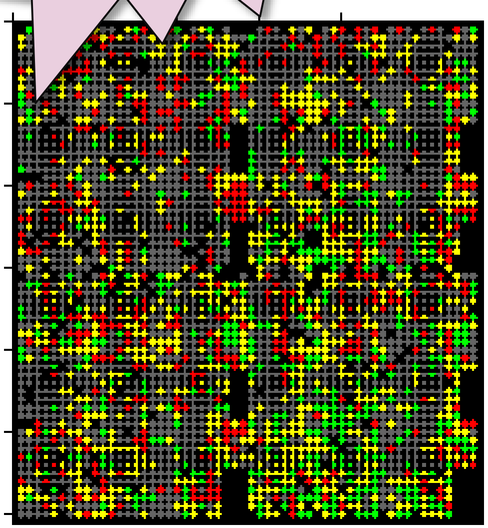
– but the Complexity



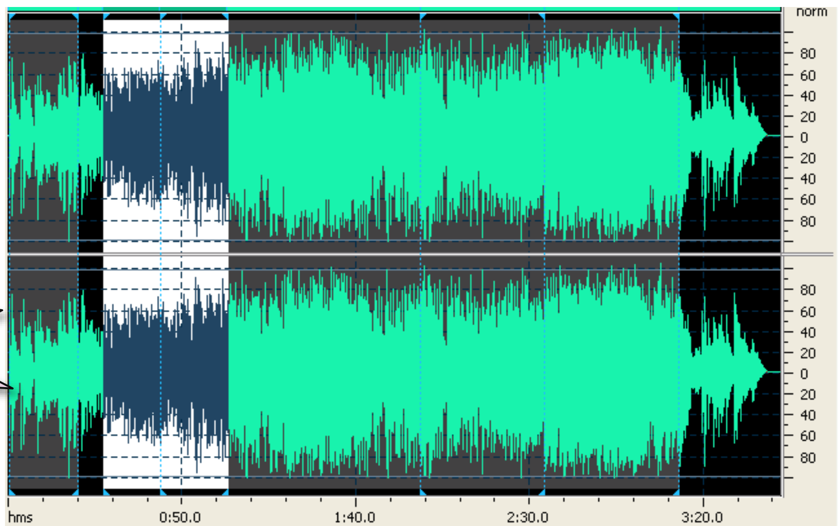
High-dimensional and Noisy Data

Microarray data: one sample point has thousands of genes and one gene point has tens of samples

Images contain a lot of information in the form of pixels



Noisy audio data



Challenges

- Not all the measured variables are important for understanding the underlying "interesting" phenomena – could complicate the process of data analysis
- Difficult to visualize
- High computational cost (time / space)

Dimensionality Reduction as a Solution

- ***Objective:*** **To transform data in high-dimensional space to a corresponding representation in some low-dimensional space, while "best" preserving the information**
- **Given dataset with *n* objects,**

$$\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{m}} \xrightarrow{\text{DR}} \mathbf{Y} \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{p}}$$

m dimensions $\qquad\qquad$ p dimensions

where $\boldsymbol{p} << \boldsymbol{m}$

Classification of DR techniques

DR

Feature Selection

Find subset of original variables
Eg: Information gain, branch and bound

Feature Extraction

Map multidimensional space into space of lower dimension
Eg: Multidimensional scaling, Principal component analysis

Dimension Reduction, DR (informal)

- **<u>Given</u>: a collection of records in *d*-dimensional space**
  - Each record contains a set of *d* attributes

- **<u>Find</u>: a <u>lower-dimensional (k<d)</u> representation of this data that:**
  - Optimizes <u>some objective</u> function and
  - Meets a given set of <u>constraints</u>.

- **Assumption: In a high-dimensional space, the data is often intrinsically low-dimensional. We can exploit this to avoid the curse of high dimensionality.**

Key Questions for Different DR Methods

- **What is the OBJECTIVE FUNCTION?**
- **What are the CONSTRAINTS?**
- **How to solve the OPTIMIZATION problem?**

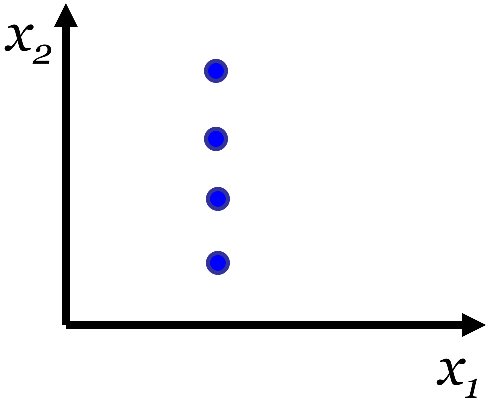
- **Different Dimension Reduction methods answer these questions differently**

Motivation for Dimension Reduction

- **Decrease the computational cost for other data mining tasks:**
  - Proximity measure calculations: $O(d) \rightarrow O(k)$
- **Reduce the noise in the data**
- **Improve the accuracy of predictive models**
- **Reduce collinearity among variables/features**

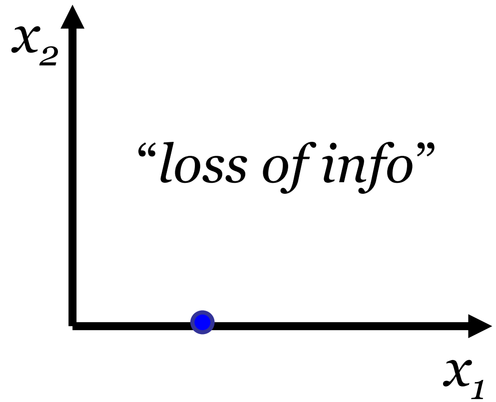
Example 1: d=2→k=1

Original data, d=2



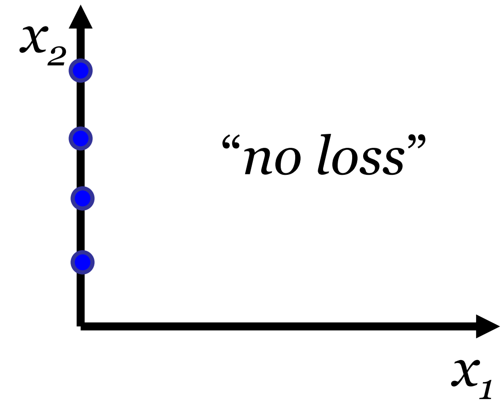
$$X_{4\times 2}$$

x_1 -projection, k=1



$$X'_{4\times 2} = X_{4\times 2} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

x_2 -projection, k=1



$$X'_{4\times 2} = X_{4\times 2} \cdot \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

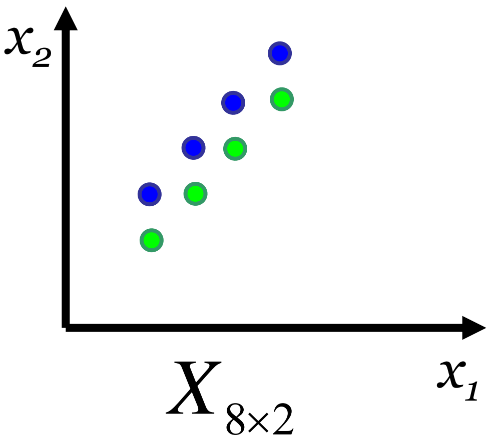
$$X'_{m\times d} = X_{m\times d} \cdot P_{d\times d}$$

Which projection is "better"?

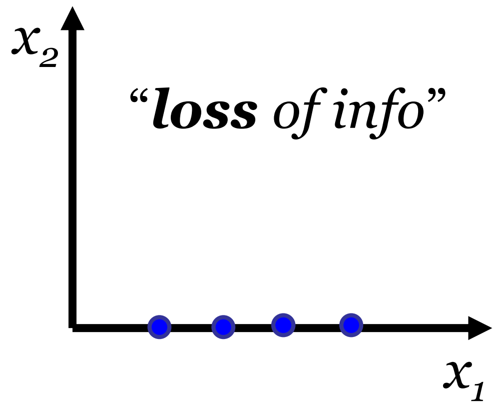
$$\boldsymbol{P}_{d\times d} = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & 1 \end{bmatrix} - \text{ projection matrix; some diagonal elements are 0}$$

Example 2: Linear, Orthogonal Projection

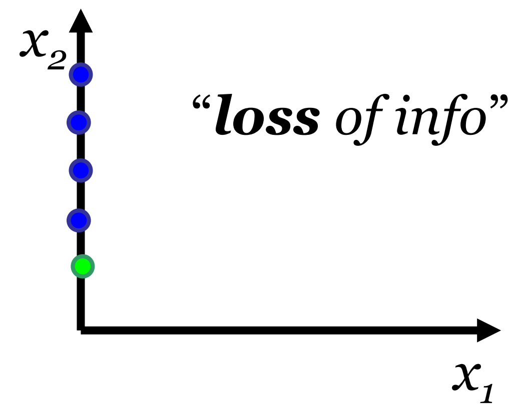
Original data, d=2



$\boldsymbol{x}_1$ -projection, k=1



$\boldsymbol{x}_2$ -projection, k=1



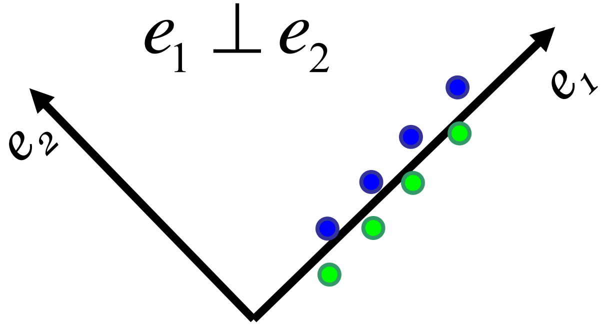
Is there a “better” projection?

Projection:
- **Linear**, $\boldsymbol{e}_1$ – line
- **Orthogonal**

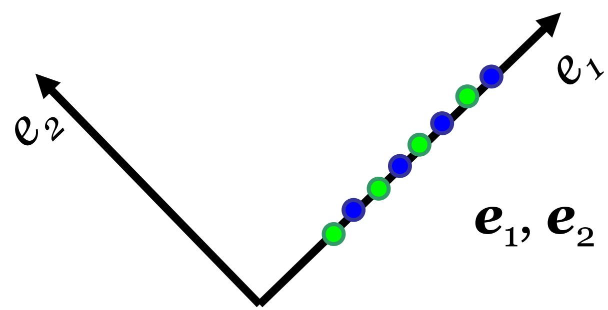
$$e_1 \perp e_2$$

Another **basis**, d=2
(rotate coord. system)

$$e_1 \perp e_2$$

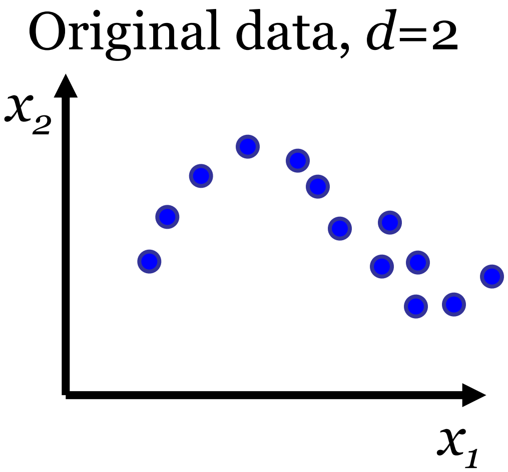


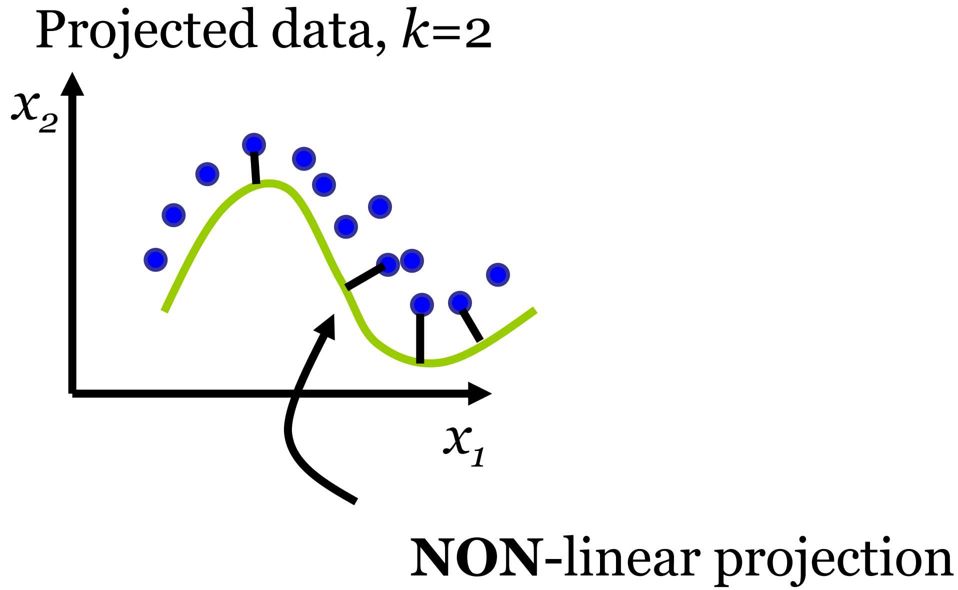
$\boldsymbol{e}_1$-projection,
k=1



$\boldsymbol{e}_1$, $\boldsymbol{e}_2$ – eigenvectors of X

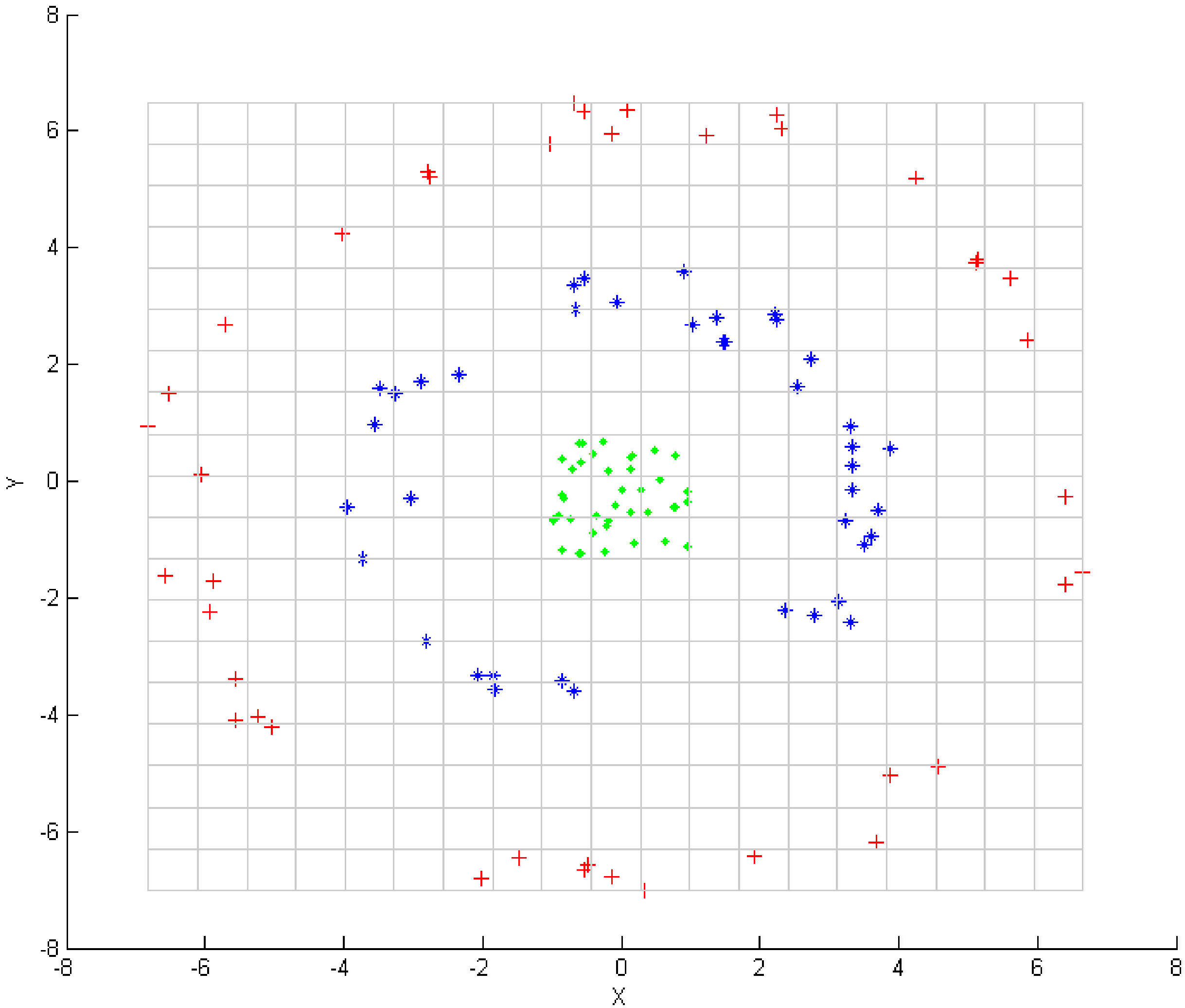
Example 3: Non-linear projection





Example 4: DR for Labeled Data

What is a "better" projection?



Types of Projection

- **Linear vs. NON-linear**
- **Orthogonal vs. non-orthogonal**
- **Unsupervised (unlabeled data) vs. supervised (labeled data)**

What is “better” projection: $d \rightarrow k$ ($k<d$)?

- **Many definitions are possible**
- **Definition 1:**
  - Projection that **maximizes the VARIANCE** of the data in its target k-dimensional projection

R Example

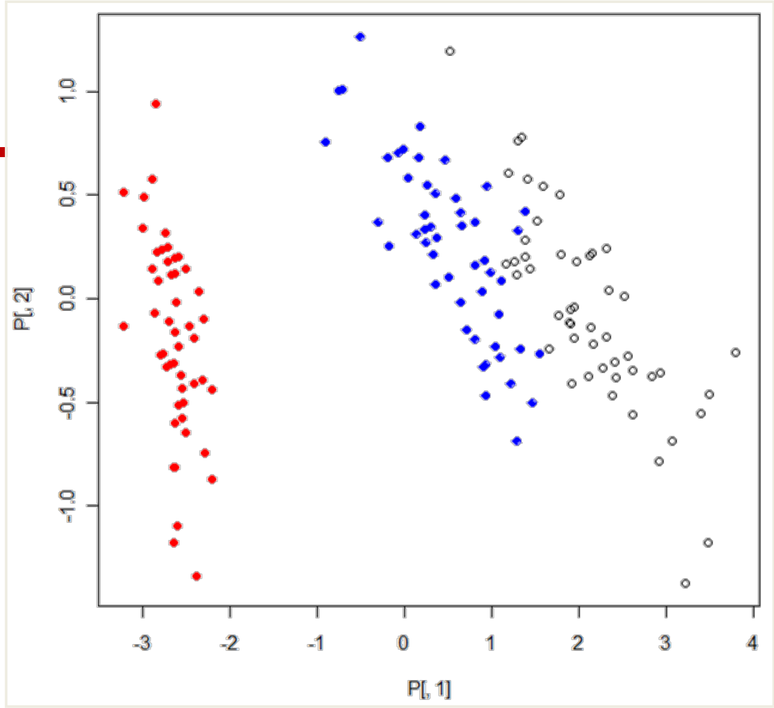
data (iris)
iris
X = iris [, 1:4]

pca = princomp (X, center=TRUE)
pca
plot (pca) # screeplot

loadings(pca) # matrix of eigenvectors
summary (pca) # check proportion of variance
P=pca$scores # projection of X onto eigenvectors

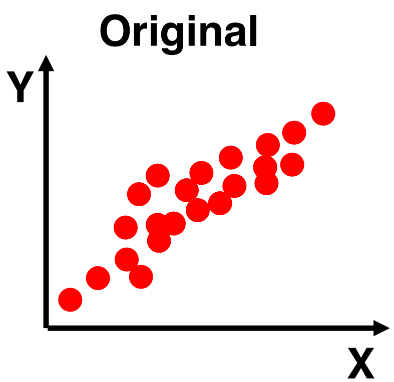
plot (P[,1], P[,2])
points (P [1:50, 1], P[1:50,2], col="red")
points (P [51:100, 1], P[51:100,2], col="blue")

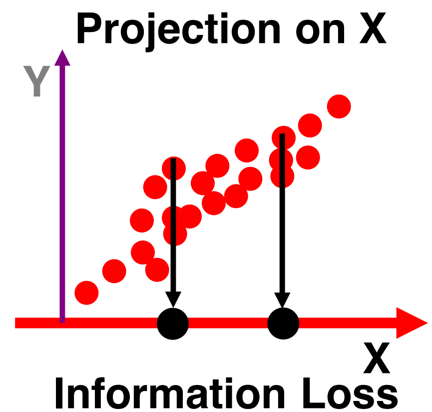
 # plot the same for X

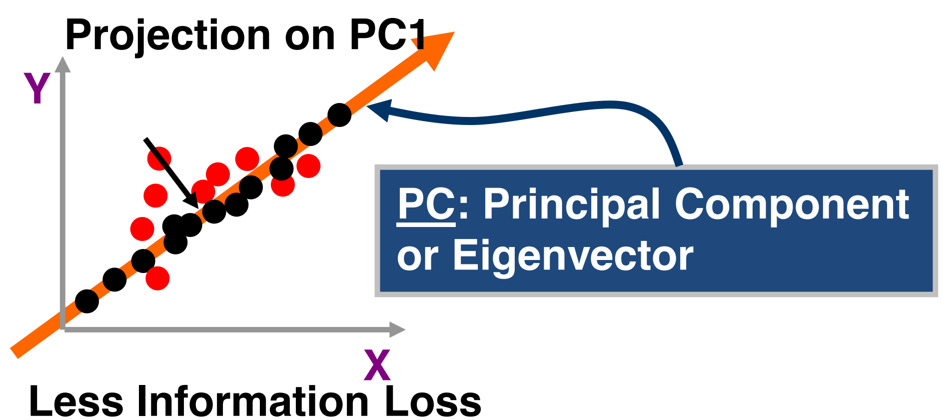


PCA: Linear Orthogonal DR

Principal Component Analysis (PCA) finds intrinsic dimensionality and allows for low-dimensional representation.







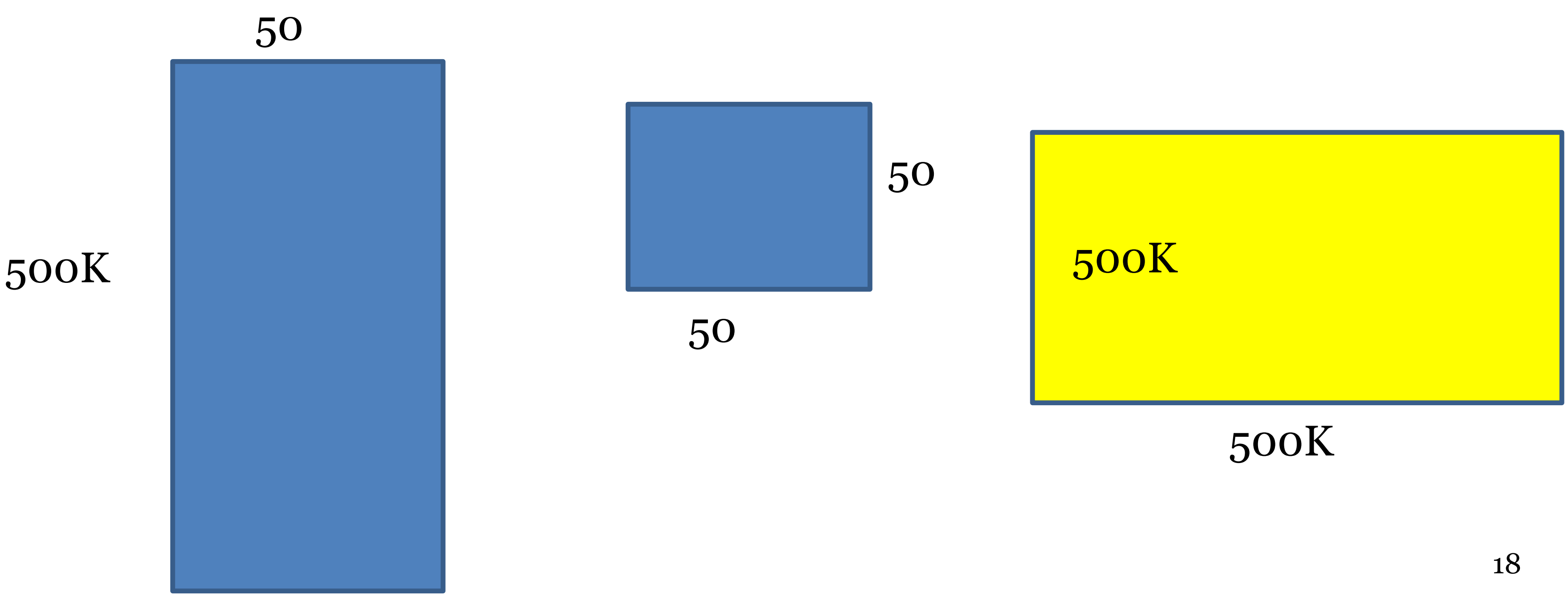
Covariance Matrix

$$\Sigma_{d\times d} = \mathrm{cov}(\boldsymbol{X}_{m\times d}) = \frac{1}{m-1}(\boldsymbol{X}^{T}{}_{m\times d}\boldsymbol{X}_{m\times d} - m\cdot\bar{\boldsymbol{x}}\cdot\bar{\boldsymbol{x}}^{T})$$

Centroid: Column means

Covariance matrix

<u>Exercise</u>: Go over the script covariance.R



PCA is the SVD of Covariance Matrix

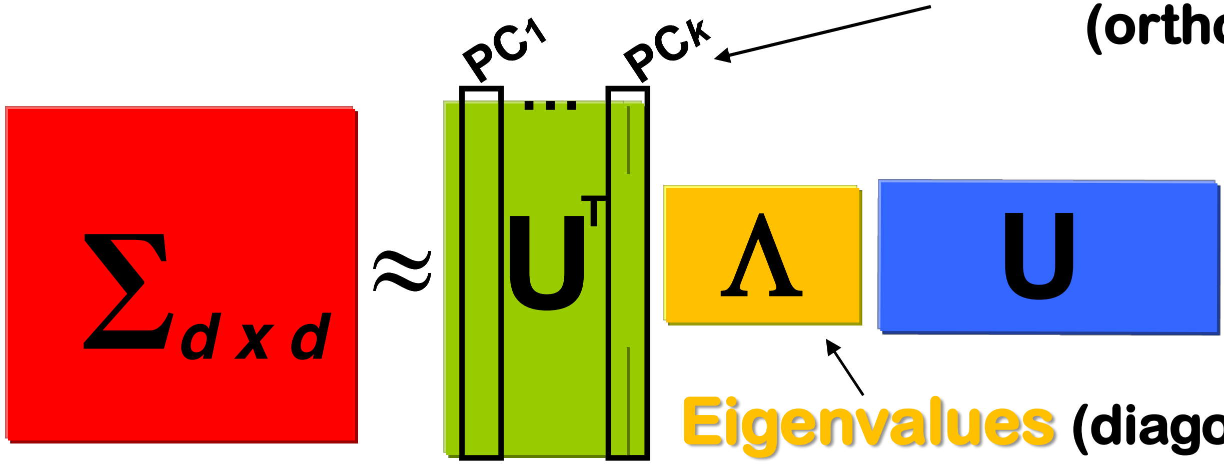
Singular Value Decomposition (SVD)

The technique underlying PCA analysis

PCA = SVD (Covariance Matrix)

PCA is the SVD applied to a covariance matrix

Eigenvectors/
Principal Components
(orthonormal)



Eigenvalues (diagonal matrix)

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k \geq \ldots \geq \lambda_d \geq 0$$

Details

`princomp` is a generic function with `"formula"` and `"default"` methods.

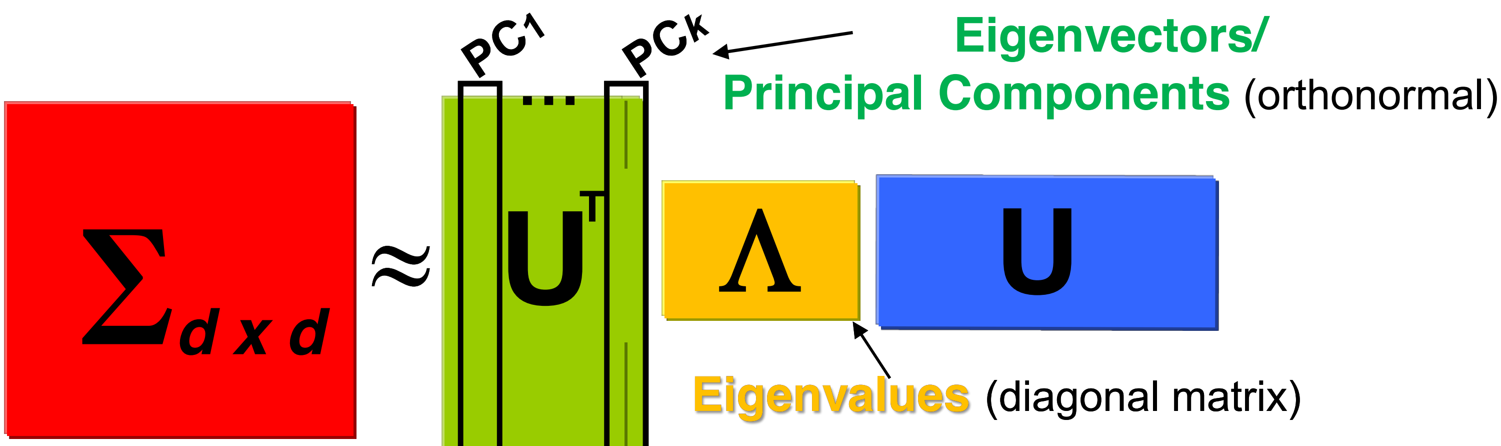
The calculation is done using `eigen` on the correlation or covariance matrix, as determined by `cor`.

PC is a weighted linear sum of original features

Extracted Feature → PCA-based Feature Extraction:

$$\boxed{\boldsymbol{PC = w_1 * f_1 + w_2 * f_2 + \cdots + w_d * f_d}}$$

The magnitude of each weight indicates how important the corresponding feature is
→ it could be used as a ***feature selection*** technique!



$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k \geq \ldots \geq \lambda_d \geq 0$$

Preserved Variability for top-k PCs

Percentage of variability preserved if the first **k** PCs are used for **projection**:

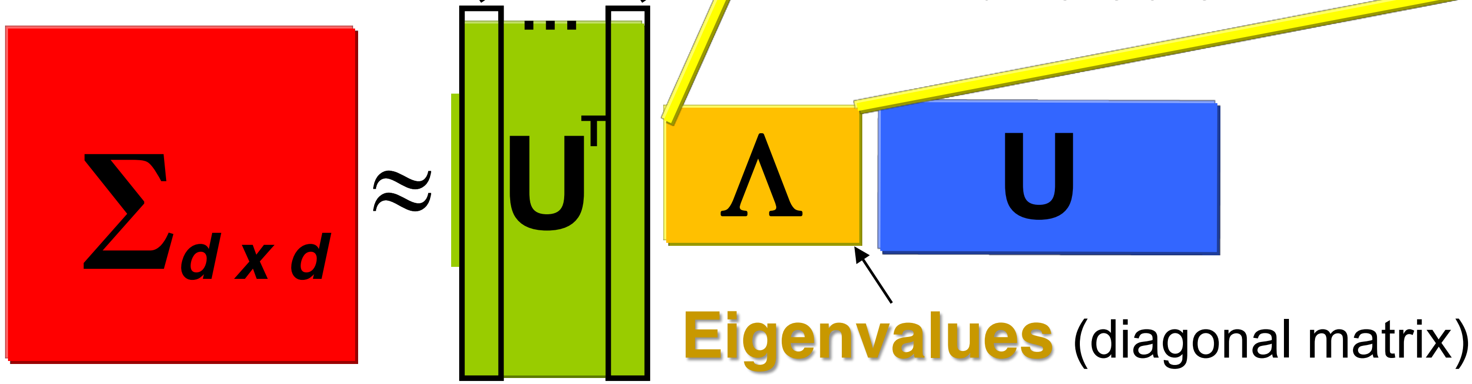
$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i} = \frac{\sum_{i=1}^{k} \lambda_i}{\text{trace}\,(\Sigma)}$$

SORTED Eigenvalues

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k \geq \ldots \geq \lambda_d \geq 0$$

Preserved variability = $\dfrac{\text{Sum of } \textbf{\textit{k}} \text{ eigenvalues}}{\text{Sum of } \textbf{\textit{all}} \text{ eigenvalues}}$

with ***k*** dimensions

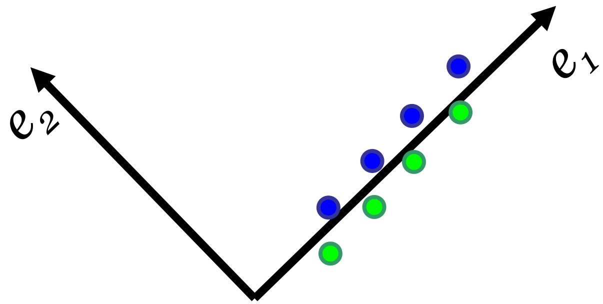


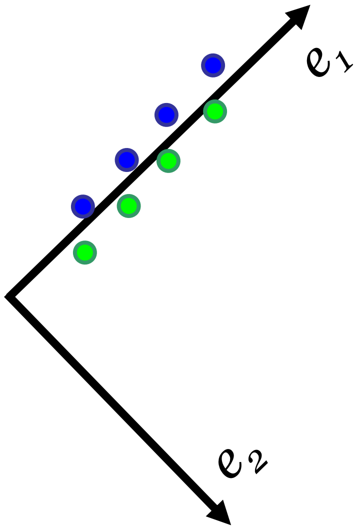
Key Points about PCA

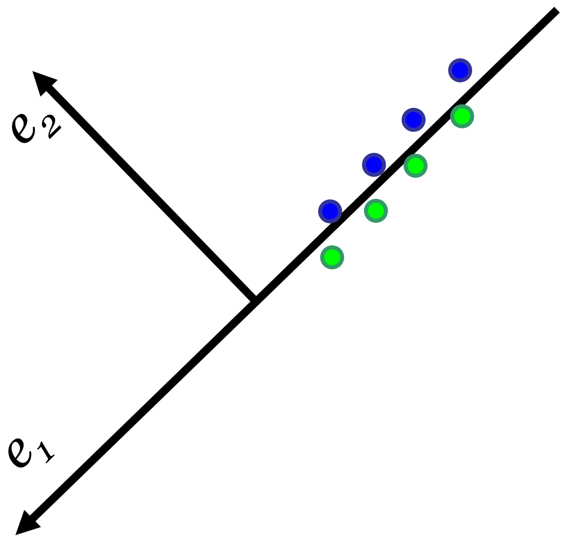
- **PCA = SVD (Covariance Matrix, ∑)**
- **Linear, orthogonal projection:**
  - "Best" linear orthogonal k-dimensional ($k < d$) view of data
- **How "good" the *k*-dimensional view is:**

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i} = \frac{\sum_{i=1}^{k} \lambda_i}{\mathrm{trace}\,(\Sigma)}$$

Eigenvectors are NOT unique







Eigenvalues: importance of eigenvectors

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k \geq \ldots \geq \lambda_d \geq 0$$

Importance of Principal Components (PC)/Eigenvectors/Loadings

PC1 preserves more variance than PC2
PC2 preserves more variance than PC3
....

Proportion of Variance Preserved if only *k* PCs are used:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i} = \frac{\sum_{i=1}^{k} \lambda_i}{\text{trace}\,(\Sigma)}$$