Linear Dimensionality Reduction and Linear Discriminant Analysis

Assoc. Prof. Karl Ezra Pilario, Ph.D.

Process Systems Engineering Laboratory

Department of Chemical Engineering

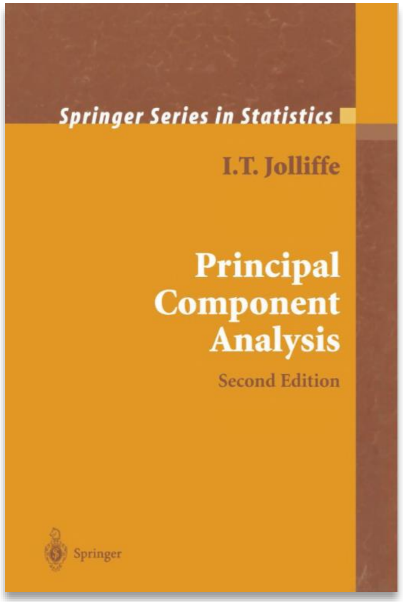
University of the Philippines Diliman

Outline

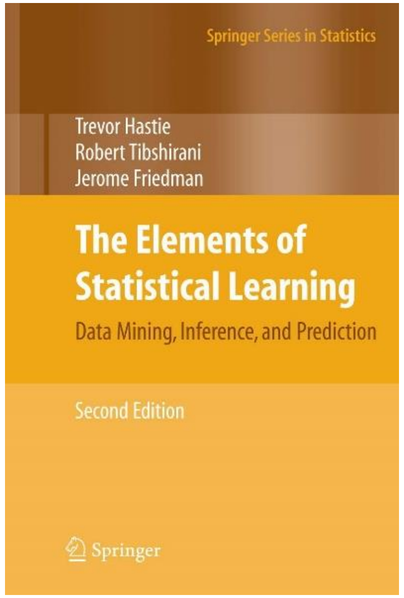
- Dimensionality Reduction
  - Curse of Dimensionality
  - Feature Selection vs. Feature Extraction
  - Principal Components Analysis (PCA)
    - Derivation
  - Non-negative Matrix Factorization (NMF)
  - Independent Components Analysis (ICA)

- Low-dimensional Classification
  - Linear Discriminant Analysis (LDA)

Jolliffe (2002)
Principal Component Analysis.
2nd Ed. Springer.

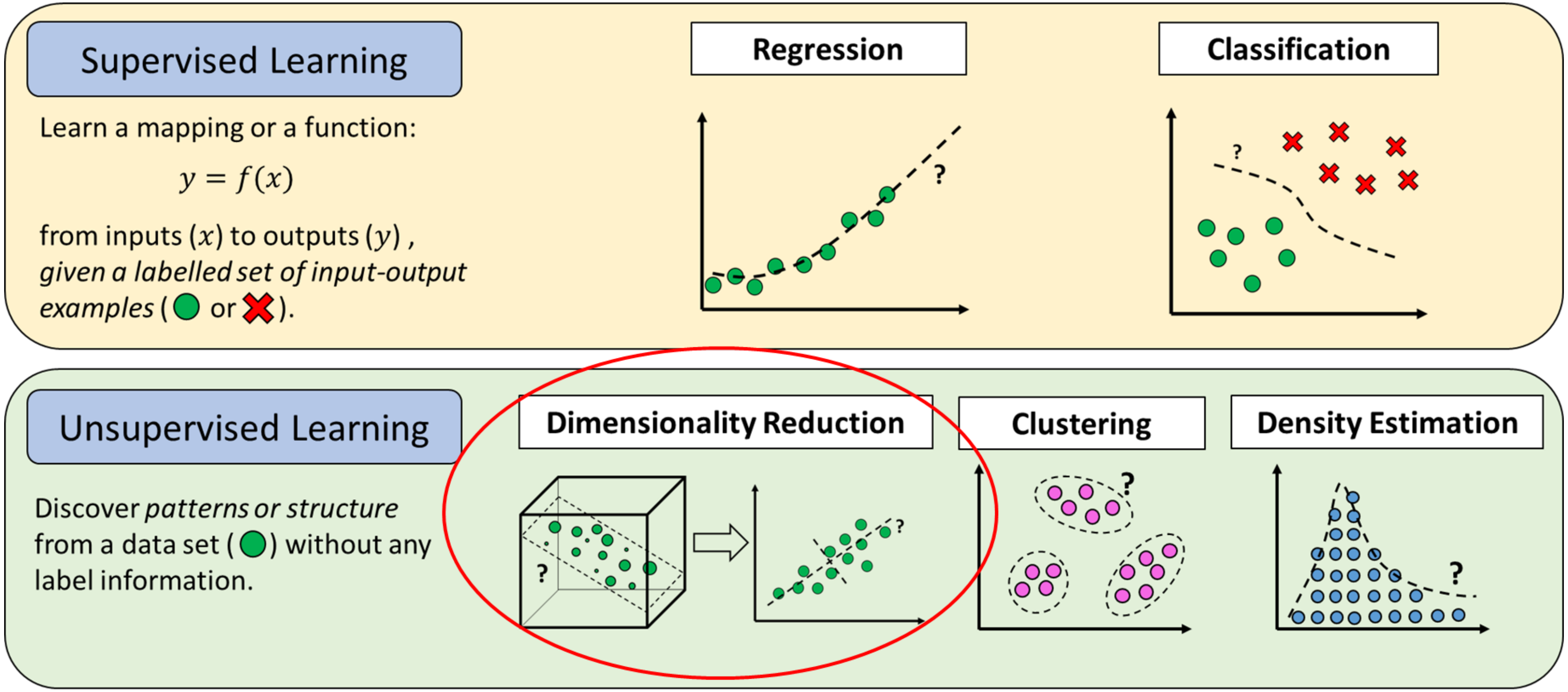


Hastie *et al.* (2008)
The Elements of Statistical Learning.
2nd Ed. Springer.



Overview

In the past few weeks, we mostly discussed supervised learning methods.
Today, we'll move into unsupervised learning.



What is dimensionality?

Recall some data sets we discussed before:

	Fisher Iris Data Set	House Price Data Set	Handwritten Digits Images
Features	Sepal length Sepal width Petal length Petal width	Floor Area No. of Rooms Age	Grayscale values (x 64)
No. of Features	4	3	64

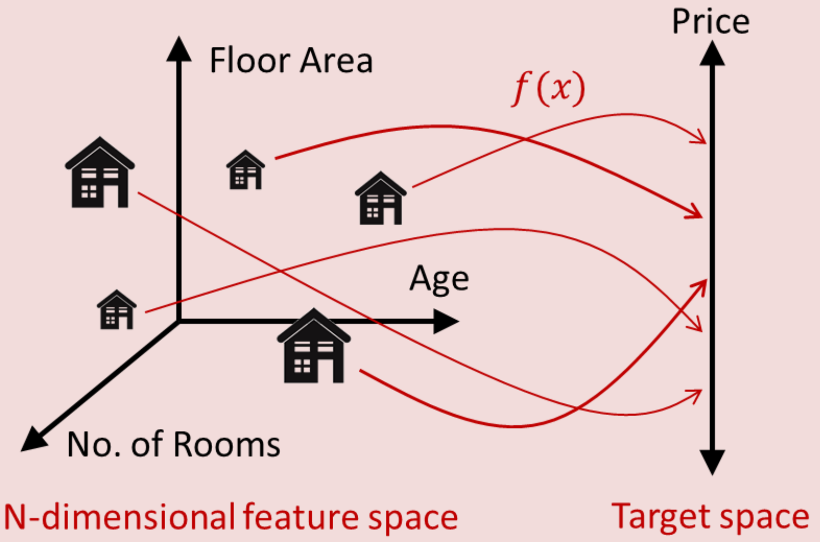
Most data sets have multiple features!

- Patient Data: Age, Height, Weight, BMI, Blood Type, …
- Image Data: RGB values x No. of Pixels
- Weather Data: Temperature, Humidity, Wind speed, …
- Car Data: Mileage, Horsepower, Weight, …
- Etc.

Dimensionality

refers to the number of "attributes" or "features" in the data set.

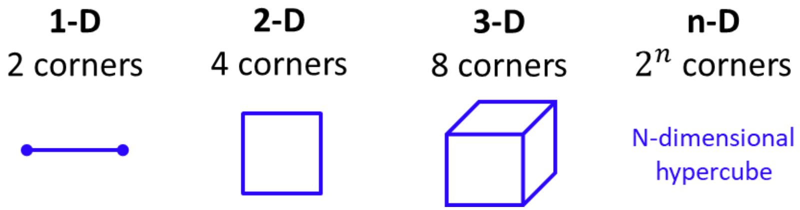
If each sample is a point in the N-dimensional space defined by the features, then our supervised learning model $f(x)$ is nothing more than a *mapping* from feature space to target space.



The Curse of Dimensionality

More often, data sets are *high-dimensional*.
This is seen as a “curse” rather than a blessing.
(Bellman, 1961)

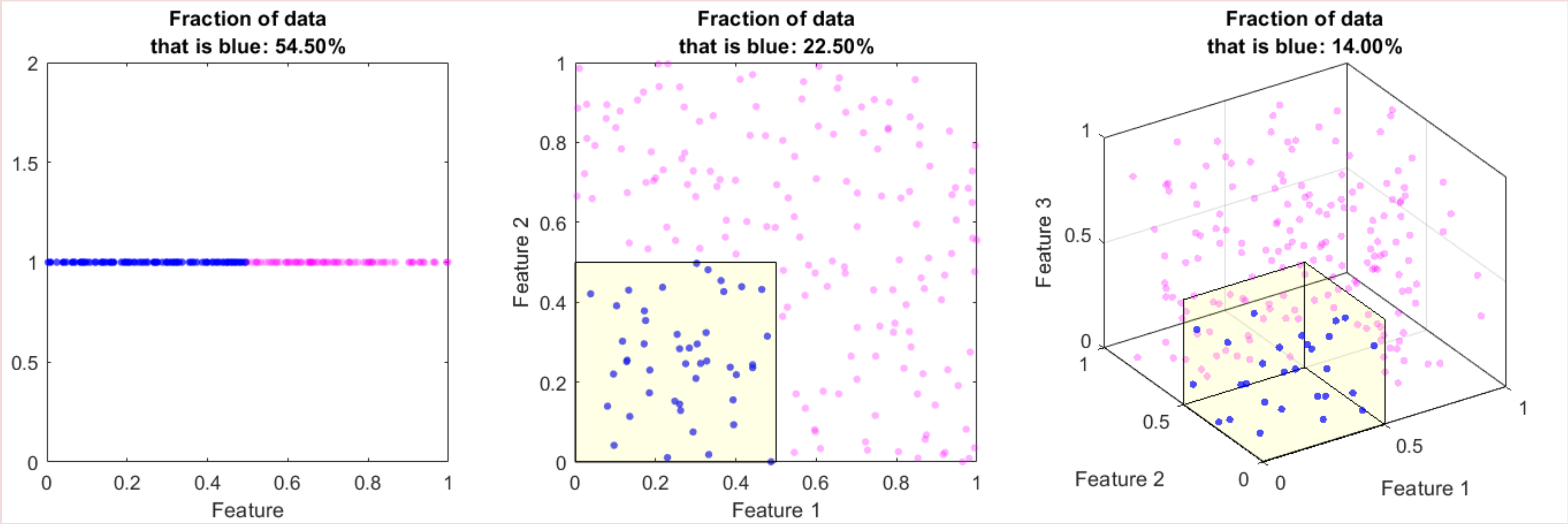
“The number of samples needed to estimate an arbitrary function with a given level of accuracy grows exponentially with respect to the number of input variables (i.e., dimensionality) of the function.”



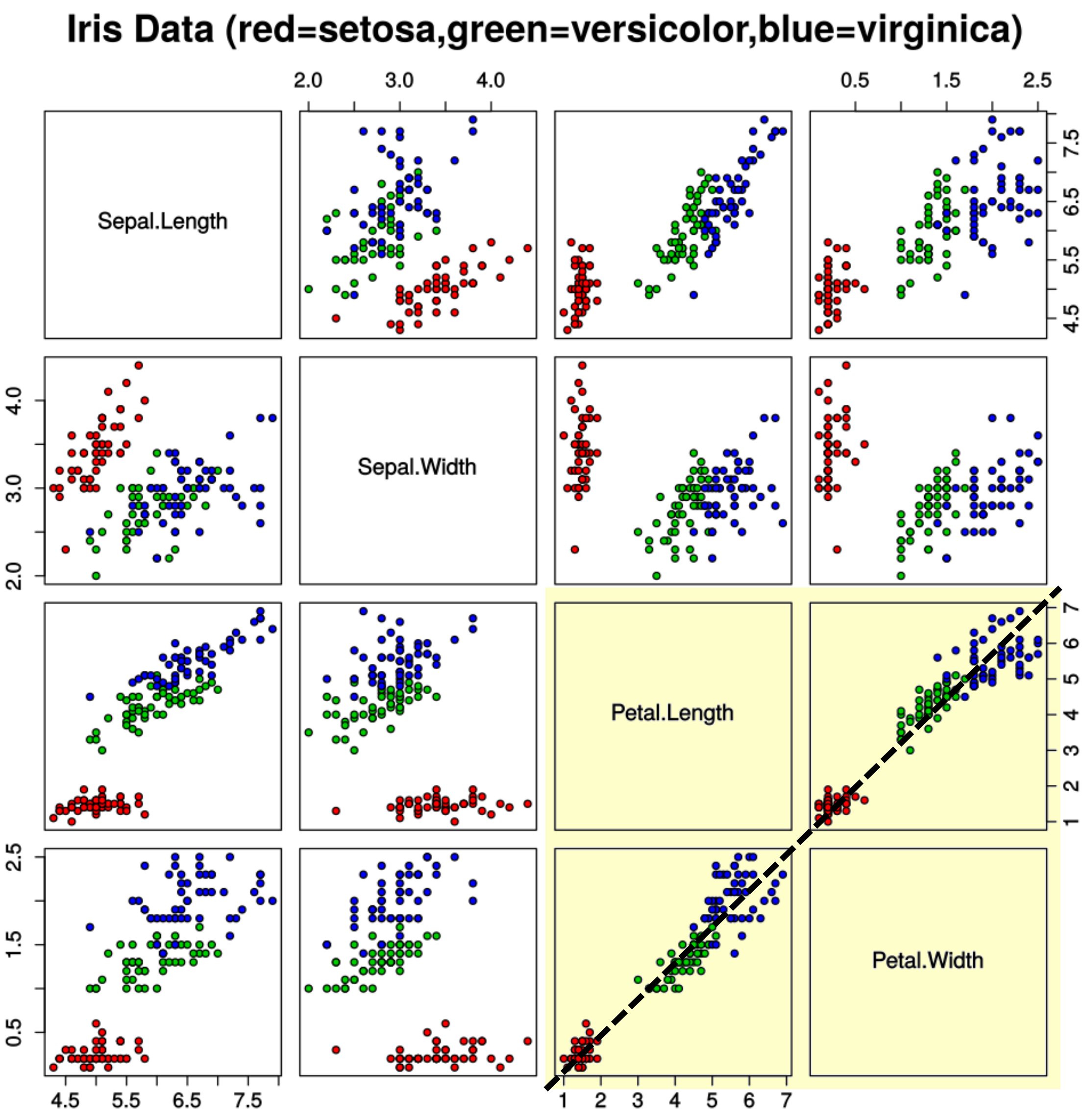
The more dimensions there are in your data, the more samples you need to cover the space.

Say, due to budget constraints, you only sampled *half* of the values in each dimension. As the number of dimensions increase, the fraction of the feature space that you covered becomes *a lot less than half*.

1-D **2-D** **3-D**



The Curse of Dimensionality



High-dimensionality comes with other problems such as:

- multi-collinear or redundant features, and
- unimportant features.

Multi-collinear or Redundant Features

- Features that are highly correlated with each other.
- *Positively correlated:* When one increases, the other also tends to increase. When one decreases, the other also tends to decrease.
- If 2 features are correlated, they could contain the same information content. Hence, one of them is *redundant*.

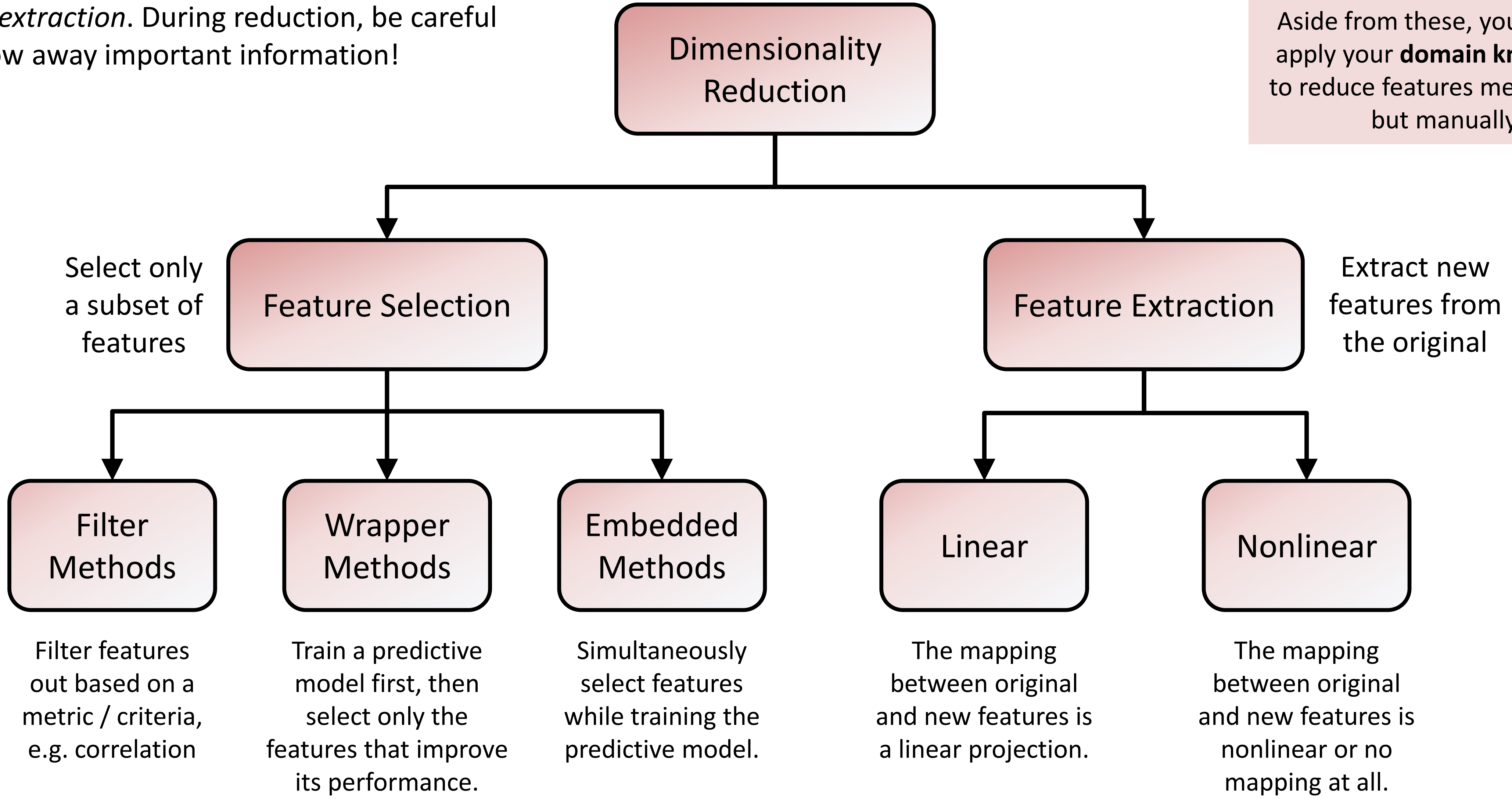
Unimportant Features

- Features that are *irrelevant* to the prediction task.
- These features can fool the learning algorithm, thinking that they contain valuable information, but they don't.

Dimensionality Reduction

We can reduce features either by *feature selection* or *feature extraction*. During reduction, be careful not to throw away important information!

Aside from these, you can also apply your **domain knowledge** to reduce features meaningfully but manually.



Dimensionality Reduction

Here are some of the typical algorithms used in feature selection.

Feature Selection

Filter Methods

- Correlation Feature Selection
- Relief / ReliefF
  - Kira and Rendell (1992)
  - Robnik-Sikonja and Kononenko (1997)
- Minimum Redundancy, Maximum Relevance (mRMR)
  - Peng et al. (2005)

Wrapper Methods

- Sequential Floating Forward Selection (SFFS)
  - Pudil et al. (1994)
- Recursive Feature Elimination (RFE)
  - Guyon (2002)
- Heuristic Search (Genetic Algorithm, etc.)

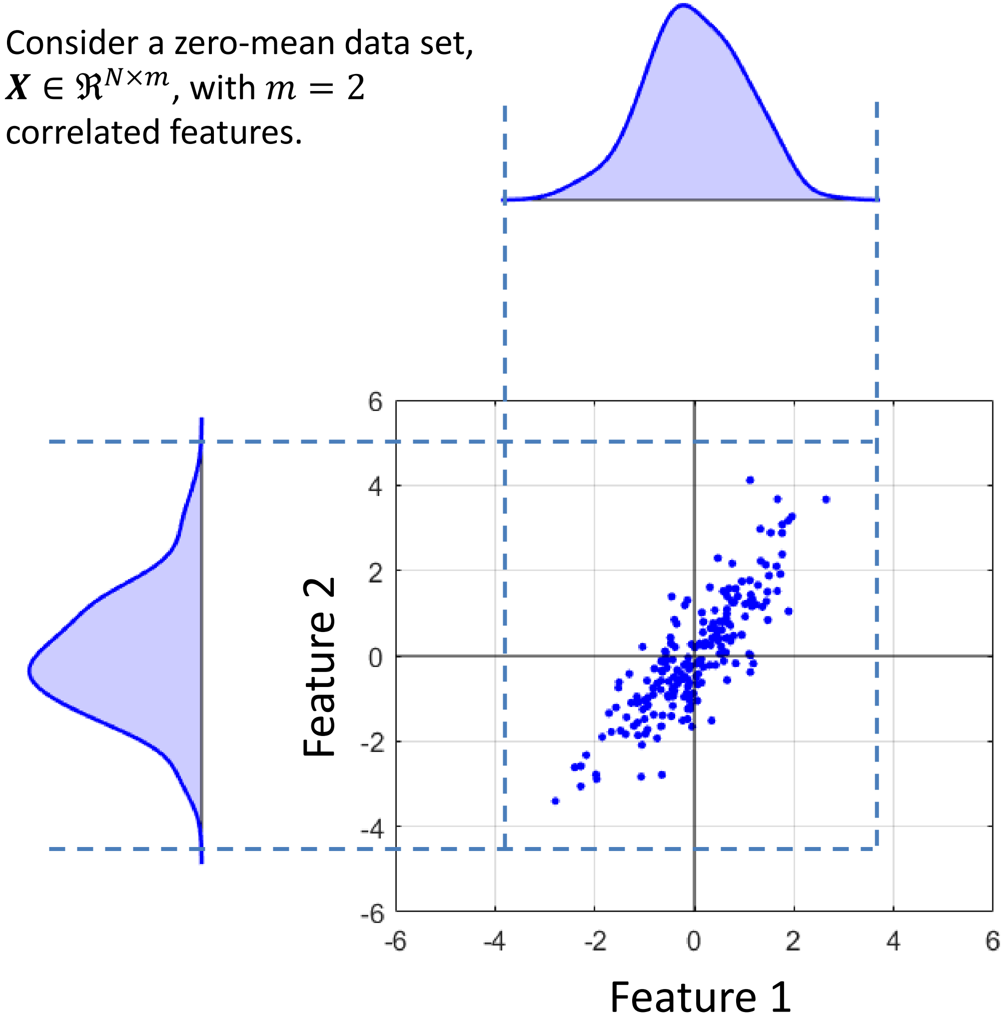
Embedded Methods

- LASSO Penalty (Least absolute shrinkage and selection operator)
  - Tibshirani (1996)
- Other penalties, e.g. Elastic Nets
- Tree-based ML methods:
  - Decision Tree
  - Random Forest
  - XGBoost

Principal Components Analysis

A popular linear feature extraction-based dimensionality reduction method due to Pearson (1901).

Consider a zero-mean data set, $\boldsymbol{X} \in \Re^{N \times m}$, with $m = 2$ correlated features.



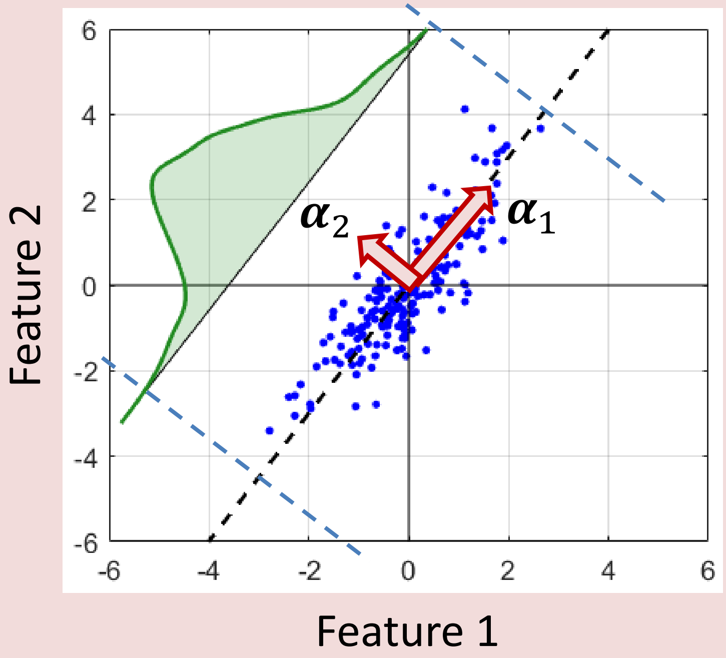
The goal of PCA is to find a projection matrix, $\boldsymbol{P} \in \Re^{m \times m}$, such that $\boldsymbol{P}$ is orthonormal and the *variance* of the projected data, $\boldsymbol{Y} \in \Re^{N \times m}$, is maximized:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{P}$$

where: $\boldsymbol{Y} = [\boldsymbol{y}_1 \quad \boldsymbol{y}_2 \quad \ldots \quad \boldsymbol{y}_m]$ — $\boldsymbol{y}$'s are called scores.

$\boldsymbol{X} = [\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \ldots \quad \boldsymbol{x}_m]$

$\boldsymbol{P} = [\boldsymbol{\alpha}_1 \quad \boldsymbol{\alpha}_2 \quad \ldots \quad \boldsymbol{\alpha}_m]$ — $\boldsymbol{\alpha}$'s are called loadings / coefficients.



$(\boldsymbol{y}_1, \boldsymbol{\alpha}_1)$ is the 1st principal component.
$(\boldsymbol{y}_2, \boldsymbol{\alpha}_2)$ is the 2nd principal component.
...and so on...

Why does PCA accomplish dimensionality reduction?

After PCA, we can take only the *first few* scores as the new extracted features, then discard the rest.

$$\begin{matrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \vdots \\ \boldsymbol{x}_m \end{matrix} \Rightarrow \begin{matrix} \boldsymbol{y}_1 \ \text{(PC1)} \\ \boldsymbol{y}_2 \ \text{(PC2)} \end{matrix}$$

Principal Components Analysis

In the next few slides, we will derive how to obtain the PCA projection matrix, $\boldsymbol{P}$.

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{P}$$

$$\boldsymbol{y}_1 = \boldsymbol{X}\boldsymbol{\alpha}_1$$

$$\boldsymbol{y}_2 = \boldsymbol{X}\boldsymbol{\alpha}_2$$

$$\boldsymbol{y}_3 = \boldsymbol{X}\boldsymbol{\alpha}_3$$

$$\boldsymbol{y}_4 = \boldsymbol{X}\boldsymbol{\alpha}_4$$

…and so on…

Step 1

What is the variance that we are trying to maximize?

The variance of $\boldsymbol{y}_1$ is given as:

$$\begin{aligned} \text{var}\,[\boldsymbol{y}_1] &= (\boldsymbol{X}\boldsymbol{\alpha}_1)^T(\boldsymbol{X}\boldsymbol{\alpha}_1) \\ &= (\boldsymbol{\alpha}_1^T\boldsymbol{X}^T)(\boldsymbol{X}\boldsymbol{\alpha}_1) \\ &= (N-1)\boldsymbol{\alpha}_1^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_1 \end{aligned}$$

where:

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

is the sample covariance of $\boldsymbol{X}$.

Step 2

How is the optimization problem in PCA formulated?

Ans: It is actually a series of optimization problems!

Maximize $\text{var}\,[\boldsymbol{y}_1]$

$$\max_{\boldsymbol{\alpha}_1} \quad \boldsymbol{\alpha}_1^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_1$$

Subject to:

$$\boldsymbol{\alpha}_1^T\boldsymbol{\alpha}_1 = 1$$

($\boldsymbol{\alpha}_1$ should be a normal vector)

Maximize $\text{var}\,[\boldsymbol{y}_2]$

$$\max_{\boldsymbol{\alpha}_2} \quad \boldsymbol{\alpha}_2^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_2$$

Subject to:

$$\boldsymbol{\alpha}_1^T\boldsymbol{\alpha}_2 = 0$$
$$\boldsymbol{\alpha}_2^T\boldsymbol{\alpha}_2 = 1$$

($\boldsymbol{\alpha}_2$ should be a normal vector and orthogonal to $\boldsymbol{\alpha}_1$)

Maximize $\text{var}\,[\boldsymbol{y}_3]$…

$$\max_{\boldsymbol{\alpha}_3} \quad \boldsymbol{\alpha}_3^T\boldsymbol{\Sigma}\boldsymbol{\alpha}_3$$

Subject to:

$$\boldsymbol{\alpha}_1^T\boldsymbol{\alpha}_3 = 0$$
$$\boldsymbol{\alpha}_2^T\boldsymbol{\alpha}_3 = 0$$
$$\boldsymbol{\alpha}_3^T\boldsymbol{\alpha}_3 = 1$$

($\boldsymbol{\alpha}_3$ should be a normal vector and orthogonal to $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2$)

…and so on…

Reference:
[1] Jolliffe and Cadima (2016). Principal component analysis: a review and recent developments. http://rsta.royalsocietypublishing.org/lookup/doi/10.1098/rsta.2015.0202
[2] Jolliffe (2002). Principal Component Analysis. 2nd Ed. Springer Verlag-Berlin. http://onlinelibrary.wiley.com/doi/10.1002/0470013192.bsa501/full

Principal Components Analysis

In the next few slides, we will derive how to obtain the PCA projection matrix, $\boldsymbol{P}$.

Step 3

How can we solve each optimization problem?

The standard approach in constrained optimization is to use Lagrange multipliers, λ:

1. Form the Lagrange equation, $\mathcal{L}$, as the objective function minus all the equality constraints multiplied to λ.
2. Optimize the Lagrange equation by differentiating it with respect to the variables, then equate them to 0.

Maximize $\mathrm{var}\,[\boldsymbol{y}_1]$

$$\max_{\boldsymbol{\alpha_1}} \quad \boldsymbol{\alpha}_1^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_1$$

Subject to:

$$\boldsymbol{\alpha}_1^T \boldsymbol{\alpha}_1 = 1$$

($\boldsymbol{\alpha}_1$ should be a normal vector)

Form the Lagrange equation:

$$\mathcal{L}(\boldsymbol{\alpha}_1, \lambda) = \boldsymbol{\alpha}_1^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_1 - \lambda(\boldsymbol{\alpha}_1^T \boldsymbol{\alpha}_1 - 1)$$

Differentiate and equate to 0:

$$\frac{\partial \mathcal{L}(\boldsymbol{\alpha}_1, \lambda)}{\partial \boldsymbol{\alpha}_1} = 2\boldsymbol{\Sigma}\boldsymbol{\alpha}_1 - 2\lambda\boldsymbol{\alpha}_1 = 0$$

Rearrange the equation:

$$\boldsymbol{\Sigma}\boldsymbol{\alpha}_1 - \lambda\boldsymbol{\alpha}_1 = 0$$

$$(\boldsymbol{\Sigma} - \lambda \boldsymbol{I})\boldsymbol{\alpha}_1 = 0$$

Insight: $\boldsymbol{\alpha}_1$ must be an eigenvector of $\boldsymbol{\Sigma}$.

Substituting λ into the objective:

$$\max_{\boldsymbol{\alpha_1}} \quad \boldsymbol{\alpha}_1^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_1 = \boldsymbol{\alpha}_1^T \lambda \boldsymbol{\alpha}_1 = \lambda \overbrace{\boldsymbol{\alpha}_1^T \boldsymbol{\alpha}_1}^{1} = \lambda$$

Insight: $\boldsymbol{\alpha}_1$ must be an eigenvector of $\boldsymbol{\Sigma}$, corresponding to the largest eigenvalue λ.

Reference:

[1] Jolliffe and Cadima (2016). Principal component analysis: a review and recent developments. http://rsta.royalsocietypublishing.org/lookup/doi/10.1098/rsta.2015.0202

[2] Jolliffe (2002). Principal Component Analysis. 2nd Ed. Springer Verlag-Berlin. http://onlinelibrary.wiley.com/doi/10.1002/0470013192.bsa501/full

Principal Components Analysis

In the next few slides, we will derive how to obtain the PCA projection matrix, $\boldsymbol{P}$.

Step 3
How can we solve each optimization problem?

The standard approach in constrained optimization is to use Lagrange multipliers, λ:

1. Form the Lagrange equation, $\mathcal{L}$, as the objective function minus all the equality constraints multiplied to λ.

2. Optimize the Lagrange equation by differentiating it with respect to the variables, then equate them to 0.

Maximize $\text{var}\,[\boldsymbol{y}_2]$

$$\max_{\boldsymbol{\alpha}_2} \quad \boldsymbol{\alpha}_2^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_2$$

Subject to:

$$\boldsymbol{\alpha}_1^T \boldsymbol{\alpha}_2 = 0$$
$$\boldsymbol{\alpha}_2^T \boldsymbol{\alpha}_2 = 1$$

($\boldsymbol{\alpha}_2$ should be a normal vector and orthogonal to $\boldsymbol{\alpha}_1$)

Form the Lagrange equation:

$$\mathcal{L}(\boldsymbol{\alpha}_2, \lambda) = \boldsymbol{\alpha}_2^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_2 - \lambda_1 \boldsymbol{\alpha}_1^T \boldsymbol{\alpha}_2 - \lambda_2 (\boldsymbol{\alpha}_2^T \boldsymbol{\alpha}_2 - 1)$$

Differentiate and equate to 0:

$$\frac{\partial \mathcal{L}(\boldsymbol{\alpha}_2, \lambda)}{\partial \boldsymbol{\alpha}_2} = 2\boldsymbol{\Sigma}\boldsymbol{\alpha}_2 - 2\lambda_1 \boldsymbol{\alpha}_1 - 2\lambda_2 \boldsymbol{\alpha}_2 = 0$$

$$\boldsymbol{\Sigma}\boldsymbol{\alpha}_2 - \lambda_1 \boldsymbol{\alpha}_1 - \lambda_2 \boldsymbol{\alpha}_2 = 0$$

Pre-multiply $\boldsymbol{\alpha}_1^T$:

$$\underbrace{\boldsymbol{\alpha}_1^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_2}_{0} - \lambda_1 \underbrace{\boldsymbol{\alpha}_1^T \boldsymbol{\alpha}_1}_{1} - \lambda_2 \underbrace{\boldsymbol{\alpha}_1^T \boldsymbol{\alpha}_2}_{0} = 0$$

Hence, $\lambda_1 = 0$.

Now that $\lambda_1 = 0$, we can proceed:

$$\boldsymbol{\Sigma}\boldsymbol{\alpha}_2 - \lambda_2 \boldsymbol{\alpha}_2 = 0$$

$$(\boldsymbol{\Sigma} - \lambda_2 \boldsymbol{I})\boldsymbol{\alpha}_2 = 0$$

Insight: $\boldsymbol{\alpha}_2$ must be an eigenvector of $\boldsymbol{\Sigma}$, corresponding to the 2nd largest eigenvalue, λ_2.

Reference:
[1] Jolliffe and Cadima (2016). Principal component analysis: a review and recent developments. http://rsta.royalsocietypublishing.org/lookup/doi/10.1098/rsta.2015.0202
[2] Jolliffe (2002). Principal Component Analysis. 2nd Ed. Springer Verlag-Berlin. http://onlinelibrary.wiley.com/doi/10.1002/0470013192.bsa501/full

Principal Components Analysis

In the next few slides, we will derive how to obtain the PCA projection matrix, $\boldsymbol{P}$.

Step 4

Finally, we realize that the series of optimization problems in PCA can be solved by simply taking the eigenvalue decomposition of the covariance matrix, $\boldsymbol{\Sigma}$. Each eigenvalue-eigenvector pair corresponds to a principal component!

Maximize var $[y_1]$

$$\max_{\boldsymbol{\alpha}_1} \ \boldsymbol{\alpha}_1^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_1$$

Subject to:

$$\boldsymbol{\alpha}_1^T \boldsymbol{\alpha}_1 = 1$$

($\boldsymbol{\alpha}_1$ should be a normal vector)

Maximize var $[y_2]$

$$\max_{\boldsymbol{\alpha}_2} \ \boldsymbol{\alpha}_2^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_2$$

Subject to:

$$\boldsymbol{\alpha}_1^T \boldsymbol{\alpha}_2 = 0$$
$$\boldsymbol{\alpha}_2^T \boldsymbol{\alpha}_2 = 1$$

($\boldsymbol{\alpha}_2$ should be a normal vector and orthogonal to $\boldsymbol{\alpha}_1$)

Maximize var $[y_3]$...

$$\max_{\boldsymbol{\alpha}_3} \ \boldsymbol{\alpha}_3^T \boldsymbol{\Sigma} \boldsymbol{\alpha}_3$$

Subject to:

$$\boldsymbol{\alpha}_1^T \boldsymbol{\alpha}_3 = 0$$
$$\boldsymbol{\alpha}_2^T \boldsymbol{\alpha}_3 = 0$$
$$\boldsymbol{\alpha}_3^T \boldsymbol{\alpha}_3 = 1$$

($\boldsymbol{\alpha}_3$ should be a normal vector and orthogonal to $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2$)

...and so on...

Solutions:

1. $\boldsymbol{\alpha}_1$ must be an eigenvector of $\boldsymbol{\Sigma}$, corresponding to the largest eigenvalue λ_1.
2. $\boldsymbol{\alpha}_2$ must be an eigenvector of $\boldsymbol{\Sigma}$, corresponding to the 2nd largest eigenvalue λ_2.
3. $\boldsymbol{\alpha}_3$ must be an eigenvector of $\boldsymbol{\Sigma}$, corresponding to the 3rd largest eigenvalue λ_3.
4. And so on…

Recall: Eigenvalue Decomposition

$$\boldsymbol{V} \quad \times \quad \boldsymbol{\Lambda} \quad \times \quad \boldsymbol{V}^T$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} a_1 & b_1 & c_1 & \cdots \\ a_2 & b_2 & c_2 & \cdots \\ a_3 & b_3 & c_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 & \cdots \\ 0 & \lambda_2 & 0 & \cdots \\ 0 & 0 & \lambda_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} a_1 & a_2 & a_3 & \cdots \\ b_1 & b_2 & b_3 & \cdots \\ c_1 & c_2 & c_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

$\boldsymbol{V}$ is our desired projection matrix, $\boldsymbol{V} = \boldsymbol{P}$.

Note: For symmetric matrices such as $\boldsymbol{\Sigma}$, the matrix $\boldsymbol{V}$ satisfies $\boldsymbol{V}^{-1} = \boldsymbol{V}^T$.

Reference:
[1] Jolliffe and Cadima (2016). Principal component analysis: a review and recent developments. http://rsta.royalsocietypublishing.org/lookup/doi/10.1098/rsta.2015.0202
[2] Jolliffe (2002). Principal Component Analysis. 2nd Ed. Springer Verlag-Berlin. http://onlinelibrary.wiley.com/doi/10.1002/0470013192.bsa501/full

Principal Components Analysis

Now, we present the main PCA algorithm.

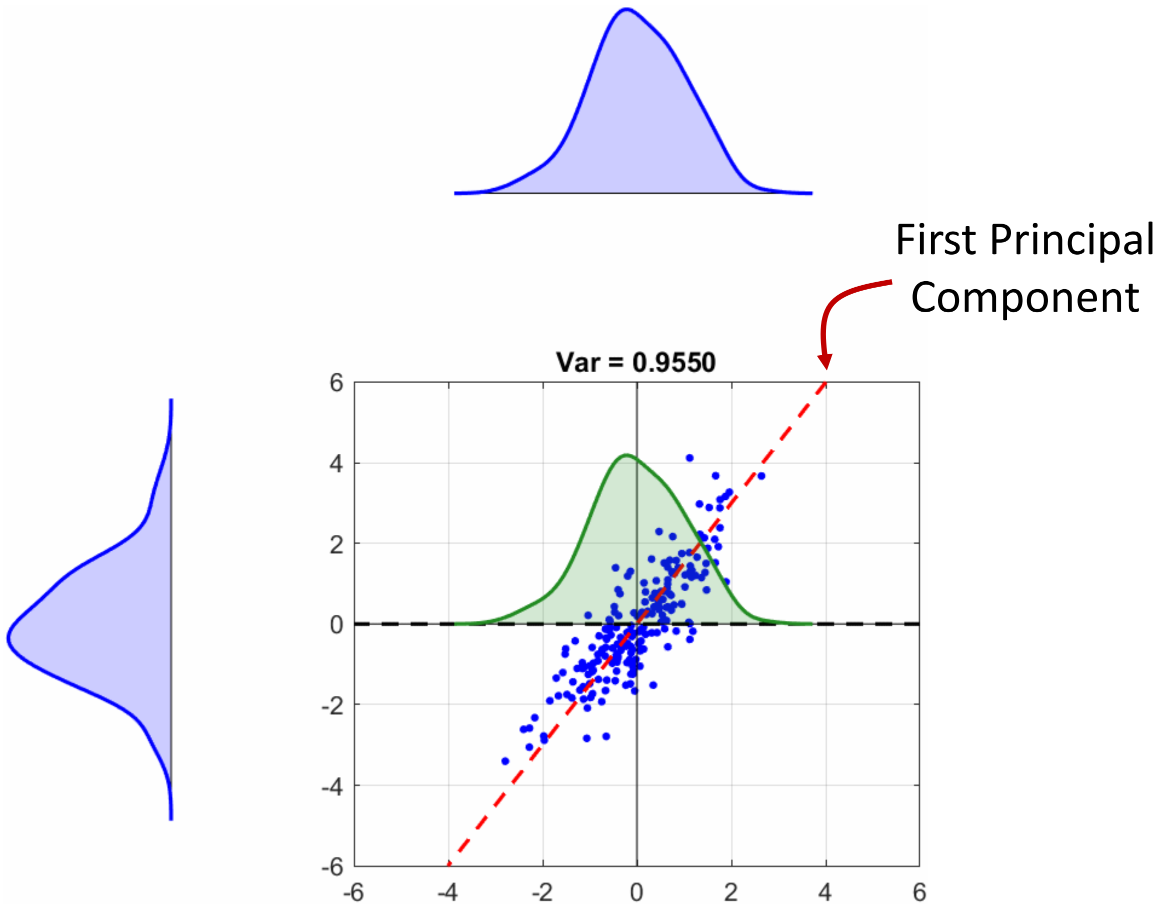
PCA Algorithm

1. Standardize the Data (zero-mean, unit-variance)

2. Compute the covariance of $\boldsymbol{X}$: $$\boldsymbol{\Sigma} = \frac{1}{N-1}\boldsymbol{X}^T\boldsymbol{X}$$

3. Compute the eigenvalue decomposition of $\boldsymbol{\Sigma}$: $$\boldsymbol{\Sigma} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$$

4. Choose only n principal components, then get $\boldsymbol{Y}$: $$\boldsymbol{P} = \boldsymbol{V}_n$$ $$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{P}$$



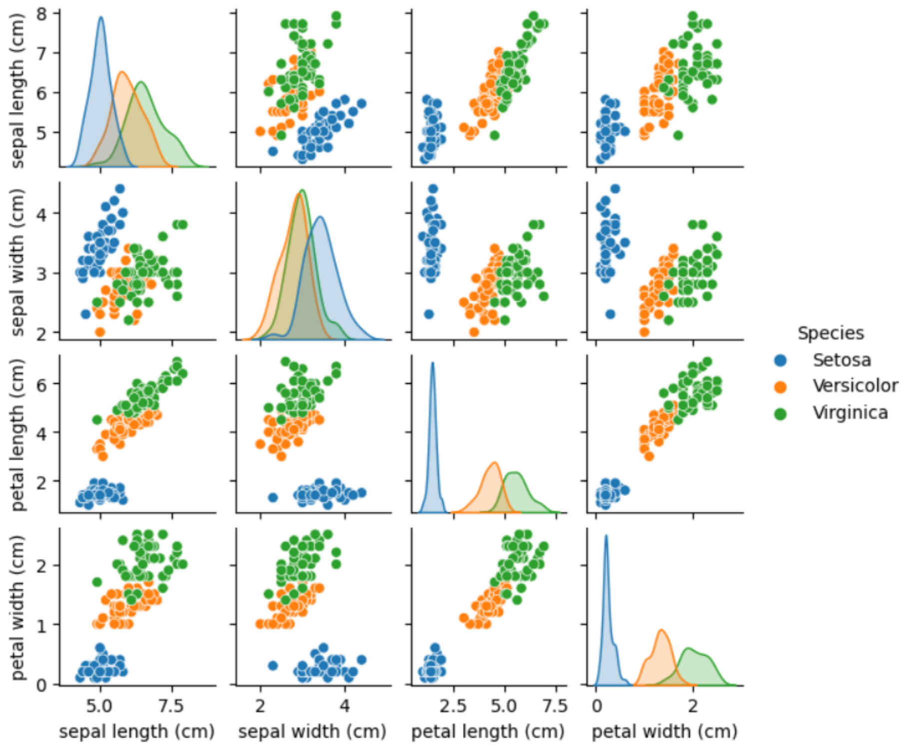
Reference:
[1] Jolliffe and Cadima (2016). Principal component analysis: a review and recent developments. http://rsta.royalsocietypublishing.org/lookup/doi/10.1098/rsta.2015.0202
[2] Jolliffe (2002). Principal Component Analysis. 2nd Ed. Springer Verlag-Berlin. http://onlinelibrary.wiley.com/doi/10.1002/0470013192.bsa501/full

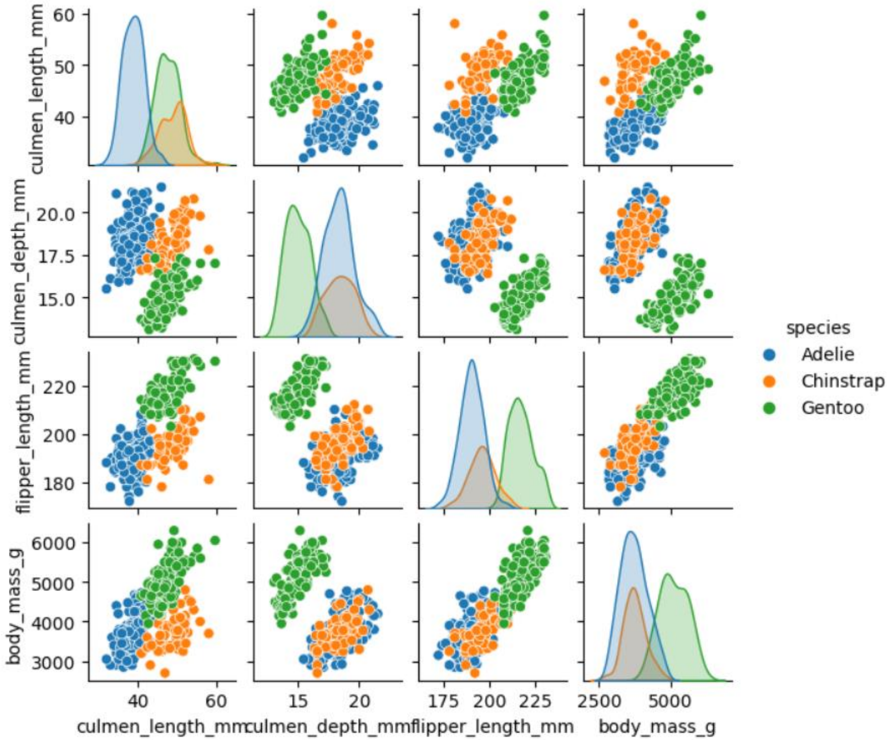
Principal Components Analysis

Example 1: Fisher Iris Data Set and Palmer Penguins Data Set

The following data set has 4 features of **150 Iris flowers**: *sepal length*, *sepal width*, *petal length*, *petal width*. Use PCA to extract 2 principal components then visualize the projected data.

The following data set has 4 features of **345 penguins** from the Palmer Archipelago, Antarctica: *culmen length*, *culmen depth*, *flipper length*, *body mass index*. Use PCA to extract 2 principal components then visualize the projected data.

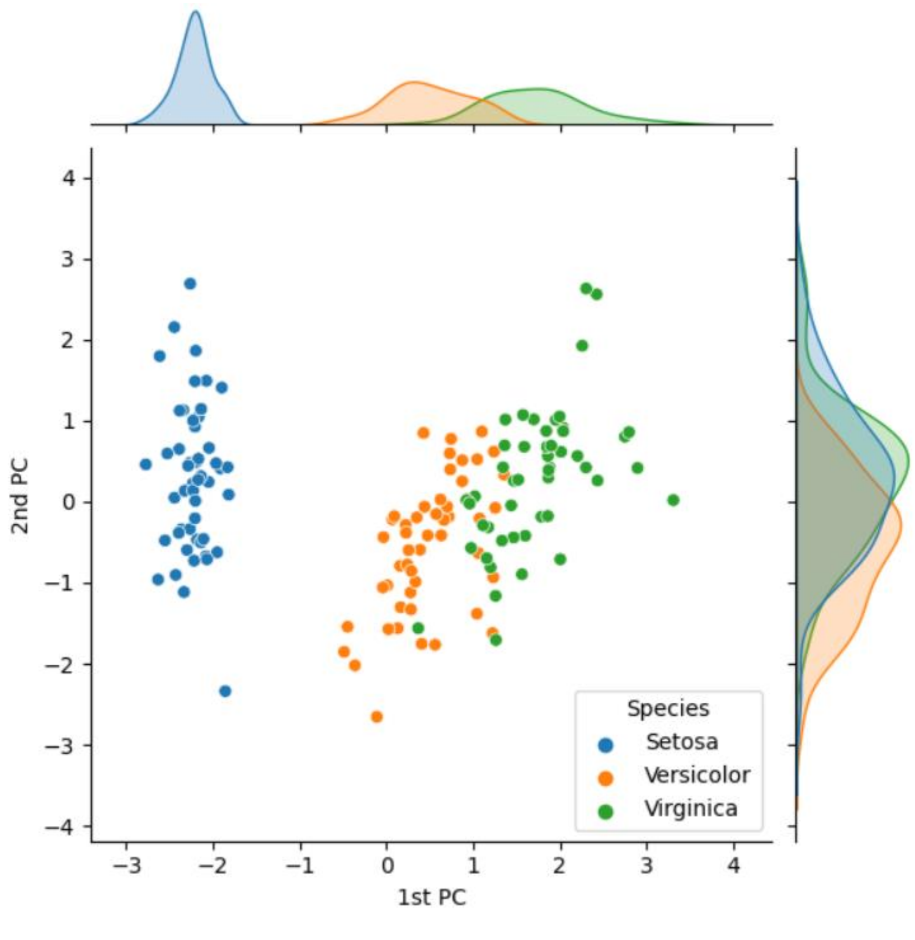




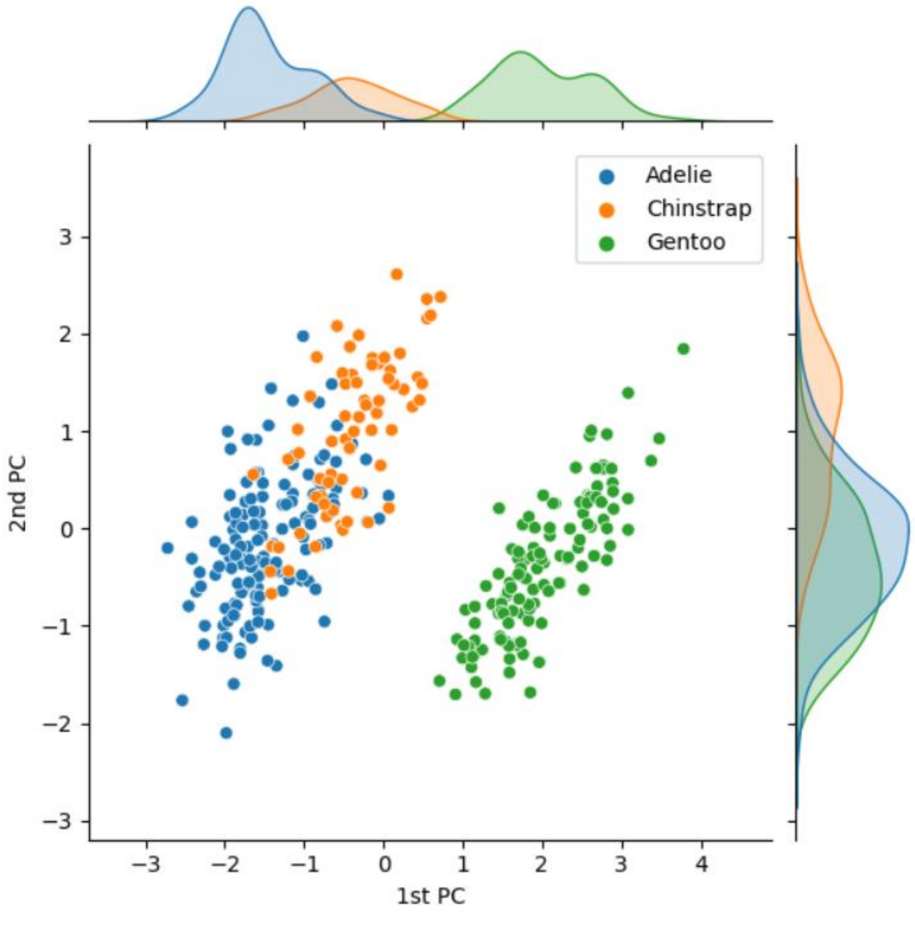
Principal Components Analysis

Example 1: Fisher Iris Data Set and Palmer Penguins Data Set

PCA Result on Fisher Iris Data



PCA Result on Palmer Penguins Data



Principal Components Analysis

How to choose the number of new features to retain (no. of principal components, n)?

It depends on your purpose:

- If you wish to visualize data in 2-D or 3-D, choose $n = 2$ or $n = 3$.
- If you have an idea about the intrinsic dimensionality of the data (from prior knowledge), then use that as the value of n.
- You can use criteria such as the CPV (cumulative percent variance) or the Scree Plot.

Cumulative Percent Variance (CPV) and Scree Plots

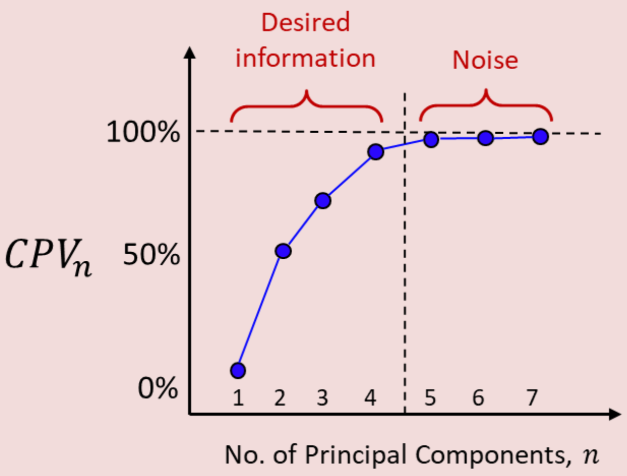
- The sum of all eigenvalues, $\sum \lambda_i$, explains the *total variance* in the data.
- Hence, each cumulative eigenvalue explains the *cumulative variance*.

$$\boldsymbol{\Sigma} = \begin{bmatrix} a_1 & b_1 & c_1 & \cdots \\ a_2 & b_2 & c_2 & \cdots \\ a_3 & b_3 & c_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 & \cdots \\ 0 & \lambda_2 & 0 & \cdots \\ 0 & 0 & \lambda_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} a_1 & a_2 & a_3 & \cdots \\ b_1 & b_2 & b_3 & \cdots \\ c_1 & c_2 & c_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

$$\boxed{CPV_n = \frac{\sum_{i=1}^{n} \lambda_i}{\sum \lambda_i}}$$

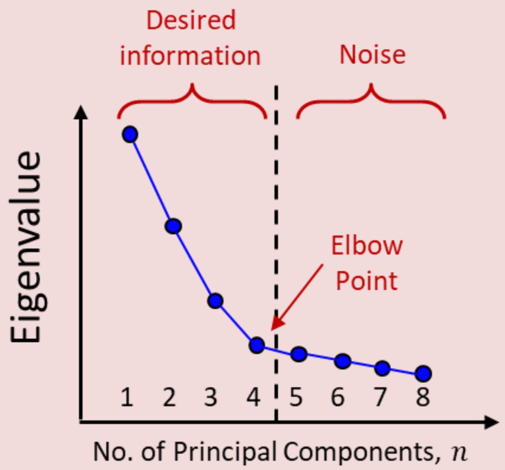
CPV Plot

Choose the first n PC's that cover, say, 95% CPV.



Scree Plot

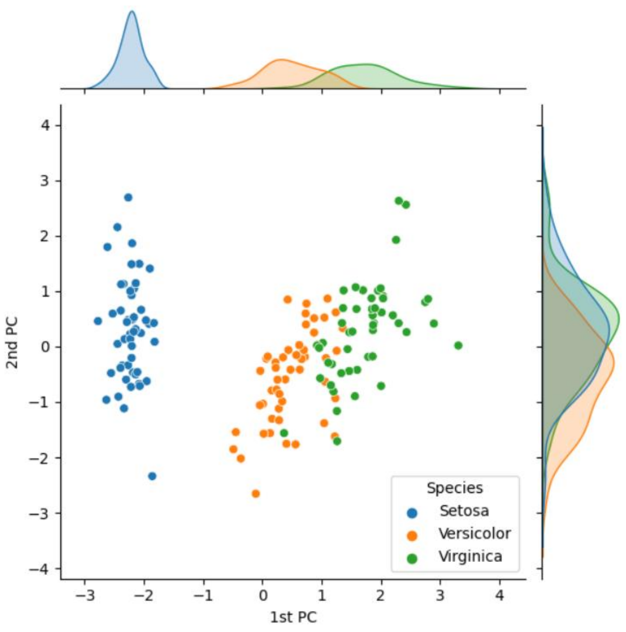
Choose the first n PC's until the elbow point occurs.



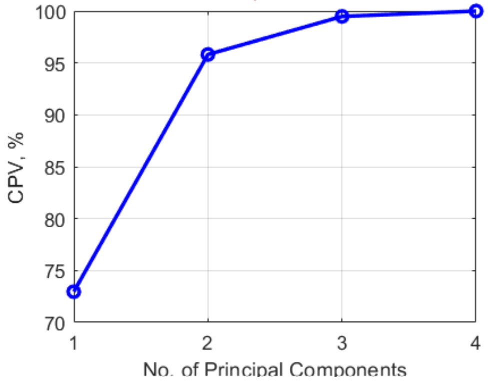
Principal Components Analysis

Example 1: Fisher Iris Data Set and Palmer Penguins Data Set

PCA Result on Fisher Iris Data

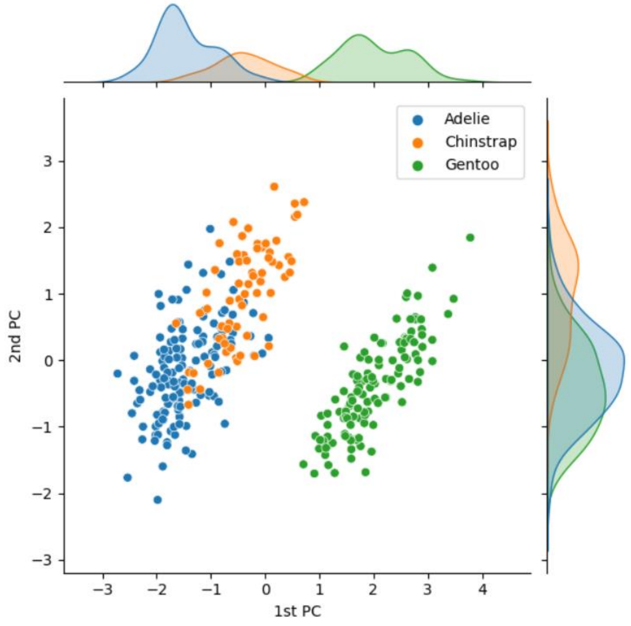


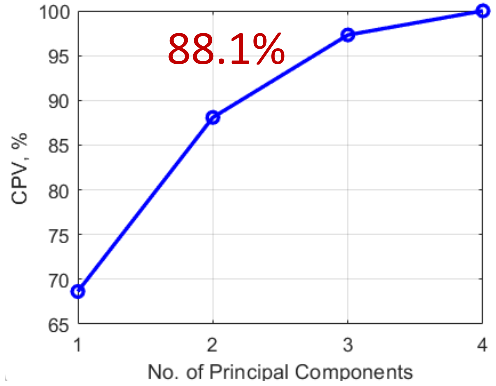
95.8%



CPV Plot

PCA Result on Palmer Penguins Data





CPV Plot

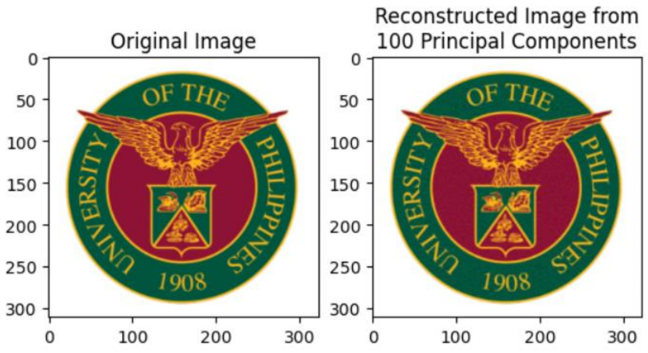
For the same number of principal components, the Fisher Iris data have a higher explained variance than the Palmer Penguins data set. This means that the measurements from the Iris flowers are a lot more correlated. The first 2 principal components already captured 95% of the variation in all 4 features, compared to only 88% in the first 2 principal components of the Palmer Penguins data.

Principal Components Analysis

Other applications of PCA:

Image Reconstruction using Less Information

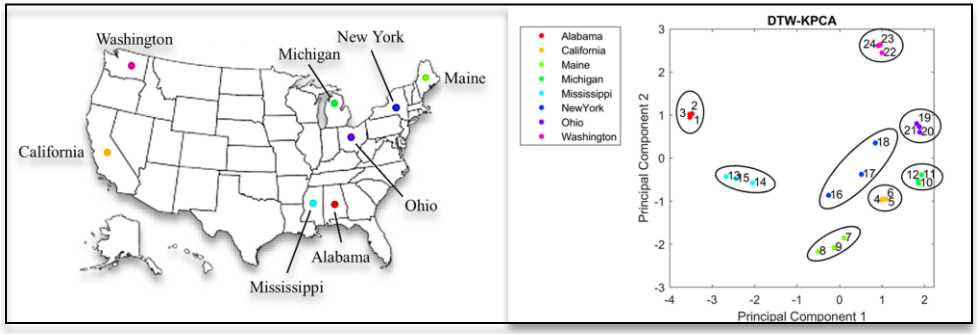
Image compression and reconstruction can be done using eigenfaces.



Discrimination of Substances via PCA on Chemometric Data

Bee substance (propolis) collected from different origins can be traced by applying PCA on their chromatographic data. (Pilario et al., 2022)

https://www.sciencedirect.com/science/article/pii/S0957417421012926



Finding Personality Traits using Factor Analysis

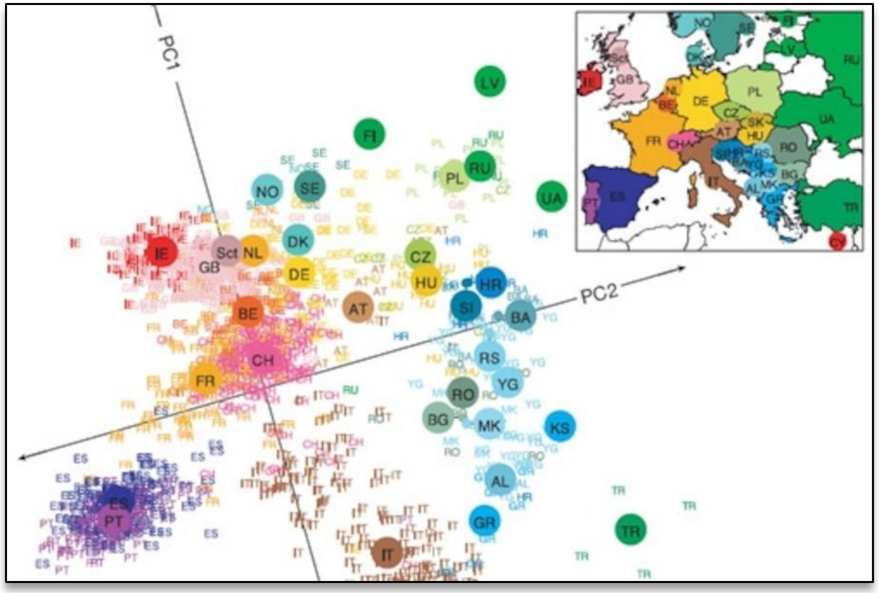
Factor Analysis is a variant of PCA used in Psychology to study personalities.

Trait Dimension	Endpoints of the Dimension	
Conscientiousness	Organized ⟷	Disorganized
	Careful ⟷	Careless
	Disciplined ⟷	Impulsive
Agreeableness	Soft-hearted ⟷	Ruthless
	Trusting ⟷	Suspicious
	Helpful ⟷	Uncooperative
Neuroticism (emotional stability vs. instability)	Calm ⟷	Anxious
	Secure ⟷	Insecure
	Self-satisfied ⟷	Self-pitying
Openness	Imaginative ⟷	Practical
	Preference for variety ⟷	Preference for routine
	Independent ⟷	Conforming
Extraversion	Sociable ⟷	Retiring
	Fun-loving ⟷	Sober
	Affectionate ⟷	Reserved

PCA on Single Nucleotide Polymorphisms (SNPs)

European genes mirror European geography (2008)

https://www.nationalgeographic.com/science/article/european-genes-mirror-european-geography



PCA, NMF, ICA

These algorithms are related but they differ in only a few aspects. They each have their own applications.

	Principal Components Analysis (PCA)	Non-negative Matrix Factorization (NMF)	Independent Components Analysis (ICA)
Transformation of Data, $\boldsymbol{X}$	$\boldsymbol{Y} = \boldsymbol{XP}$ or $\boldsymbol{X} = \boldsymbol{YP}^T$	$\boldsymbol{X} = \boldsymbol{WH}$	$\boldsymbol{S} = \boldsymbol{WX}$
Requirements	• Variances of $\boldsymbol{Y}$ are *maximized*. • $\boldsymbol{P}$ is *orthonormal*.	• $\boldsymbol{W}$ and $\boldsymbol{H}$ elements must all be *non-negative*: $\boldsymbol{W} \geq 0, \boldsymbol{H} \geq 0$	• $\boldsymbol{S}$ contains *maximally independent* components.

Cost Function / Objective Func.	• $\max_{\boldsymbol{P}} \mathrm{var}(\boldsymbol{Y})$	• $\min_{\boldsymbol{H},\boldsymbol{W}} \|\boldsymbol{X} - \boldsymbol{WH}\|^2$ (Frobenius norm)
		• $\min_{\boldsymbol{H},\boldsymbol{W}} \boldsymbol{D}(\boldsymbol{X}\|\|\boldsymbol{WH})$ (Kullback-Leibler divergence)
			• *Maximum Independence* can mean:
			  • Min: Mutual Information
			  • Max: Non-Gaussianity
			    • Max: Negentropy
			    • Max: Kurtosis

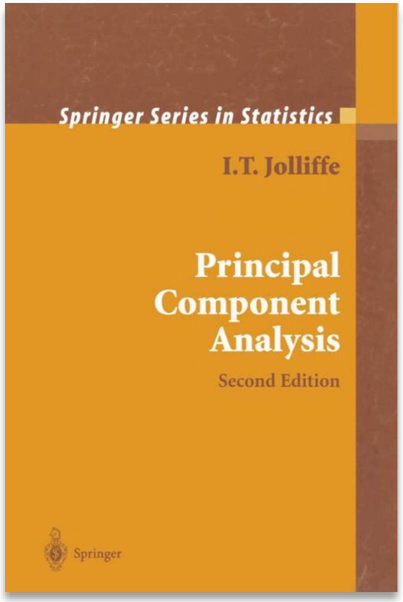
Solver/s	• Eigenvalue Decomposition (EVD), or, • Singular Value Decomposition (SVD)	• Fixed-point iteration (Lee and Seung, 2000) • *Multiplicative Update Rules* • Any optimization solver	• EVD or SVD • Fixed-point iteration (Hyvarinen, 2000) • Any optimization solver
Other Related Algorithms / Applications	• Factor Analysis • Correspondence Analysis • Kernel PCA, Sparse PCA, Robust PCA • Directional Component Analysis • Canonical Correlation Analysis • PARAFAC	• K-means clustering ($\boldsymbol{HH}^T = I$) • Probabilistic Latent Semantic Analysis (PLSA) • Collaborative Filtering (Recommendation systems)	• Blind Source Separation • Projection Pursuit • Infomax • FastICA

Outline

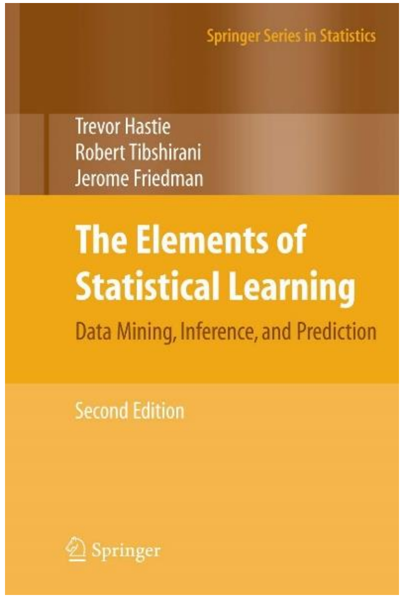
- Dimensionality Reduction
    - Curse of Dimensionality
    - Feature Selection vs. Feature Extraction
    - Principal Components Analysis (PCA)
        - Derivation
    - Non-negative Matrix Factorization (NMF)
    - Independent Components Analysis (ICA)

- **Low-dimensional Classification**
    - **Linear Discriminant Analysis (LDA)**

Jolliffe (2002)
Principal Component Analysis.
2nd Ed. Springer.



Hastie *et al.* (2008)
The Elements of Statistical Learning.
2nd Ed. Springer.

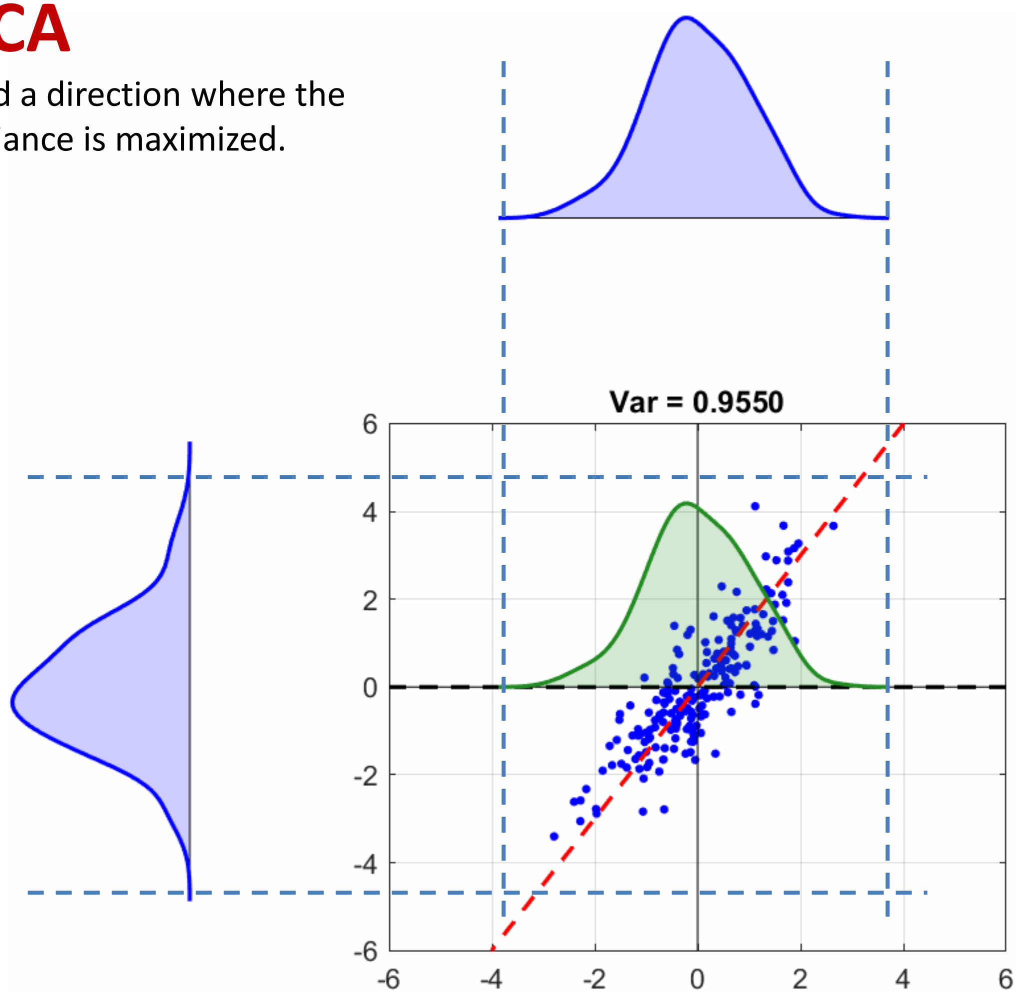


Linear Discriminant Analysis

LDA performs dimensionality reduction, but its main purpose is classification.
Since it requires knowledge of class labels, it is a supervised learning method.

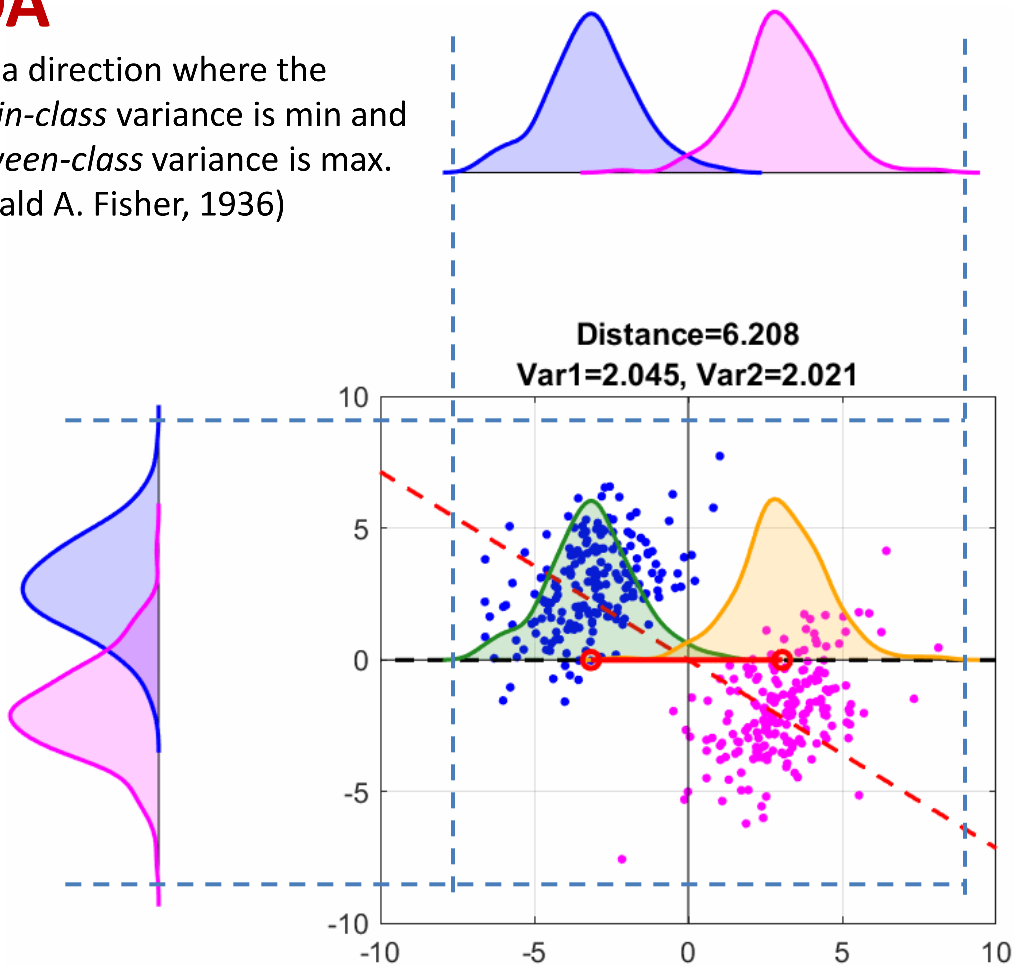
PCA

Find a direction where the variance is maximized.



LDA

Find a direction where the *within-class* variance is min and *between-class* variance is max.
(Ronald A. Fisher, 1936)



Linear Discriminant Analysis

Similar to PCA, the solution to LDA also involves an eigenvalue decomposition.

Given $\boldsymbol{X} \in \Re^{N\times m}$ and class labels $\boldsymbol{y} \in \Re^{N}$, let:

$$\chi_j = \text{set of samples that belong to class } j$$

$$p = \text{no. of classes}$$
$$m = \text{no. of features}$$
$$N = \text{total no. of samples}$$

$$\boxed{\boldsymbol{S}_w = \sum_{j=1}^{p} \boldsymbol{S}_j}$$

(Within-class Scatter Matrix)

where:

$$\boldsymbol{S}_j = \sum_{\boldsymbol{x}_i \in \chi_j} \left(\boldsymbol{x}_i - \text{mean}(\boldsymbol{x}_j)\right)\left(\boldsymbol{x}_i - \text{mean}(\boldsymbol{x}_j)\right)^T$$

(Within-scatter Matrix for class *j*)

$$\boxed{\boldsymbol{S}_b = \sum_{i=1}^{p} n_i\left(\text{mean}(\boldsymbol{x}_i) - \text{mean}(\boldsymbol{X})\right)\left(\text{mean}(\boldsymbol{x}_i) - \text{mean}(\boldsymbol{X})\right)^T}$$

(Between-class Scatter Matrix)

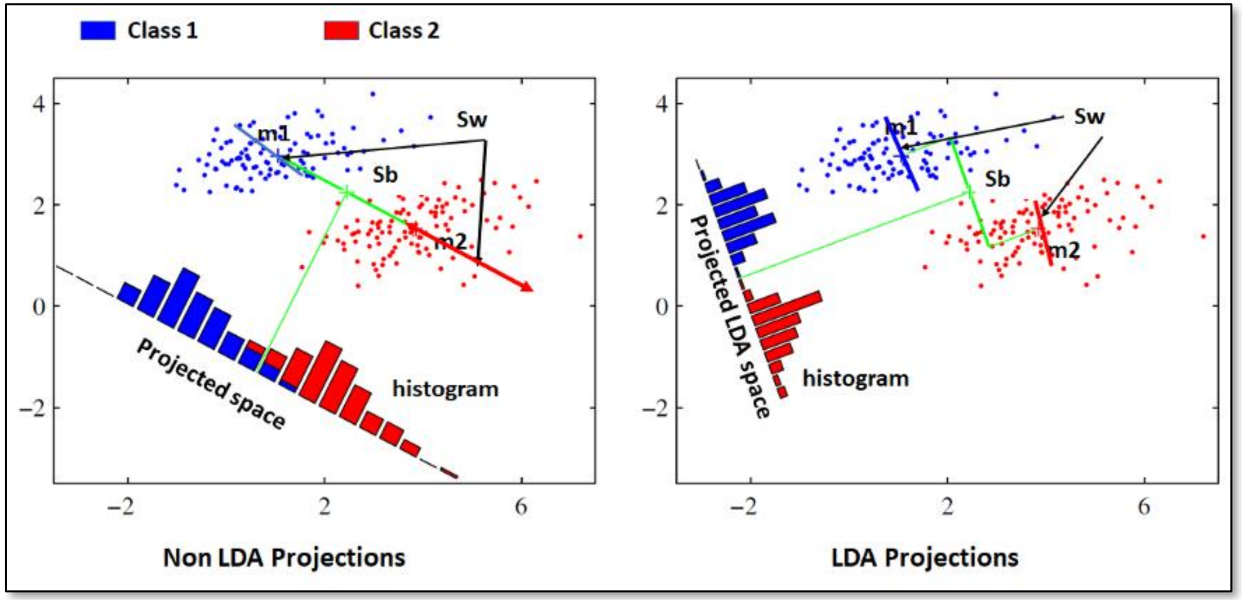
where: n_i = no. of samples in class i

$$\boxed{\boldsymbol{S}_t = \sum_{i=1}^{N}\left(\boldsymbol{x}_i - \text{mean}(\boldsymbol{X})\right)\left(\boldsymbol{x}_i - \text{mean}(\boldsymbol{X})\right)^T}$$

Note: $\boldsymbol{S}_t = \boldsymbol{S}_b + \boldsymbol{S}_w$

(Total Scatter Matrix)

a.k.a. covariance matrix



Source: https://blog.devgenius.io/part-3-linear-discriminant-analysis-b311fbef7369

The goal of LDA is to find a projection matrix $\boldsymbol{W}_r = [\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_r] \in \Re^{m\times r}$, such that the following Fisher criterion is maximized:

$$\max_{\boldsymbol{w}\neq 0} \frac{\boldsymbol{w}^T \boldsymbol{S}_b \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_w \boldsymbol{w}}$$

which is equivalent to solving the following EVD: $\boldsymbol{S}_b \boldsymbol{w}_i = \lambda_i \boldsymbol{S}_w \boldsymbol{w}_k$

The new features extracted from LDA are computed as: $\boldsymbol{z}_i = \boldsymbol{W}_r^T \boldsymbol{x}_i$

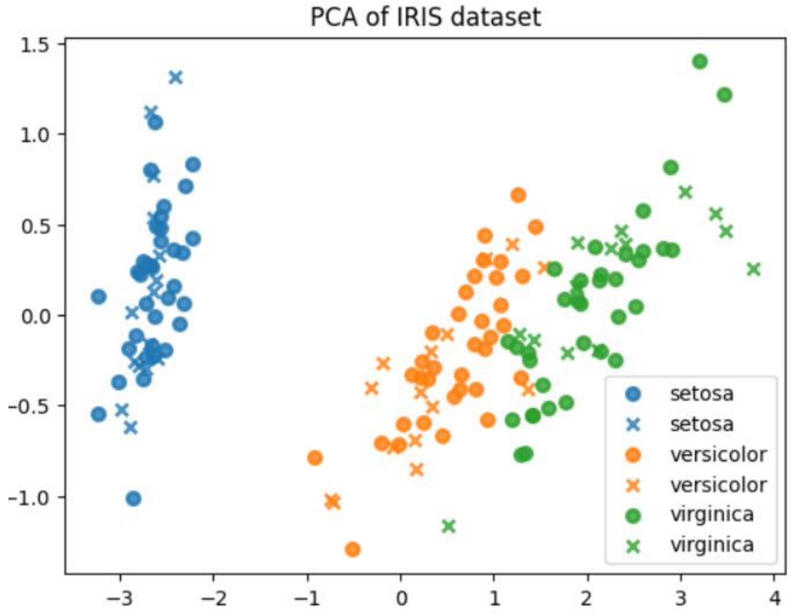
Dimensionality reduction occurs since the data is now projected onto r-dimensional space. However, since the rank of $\boldsymbol{S}_b$ is less than p (no. of classes), we can only choose r to be at most $p - 1$.

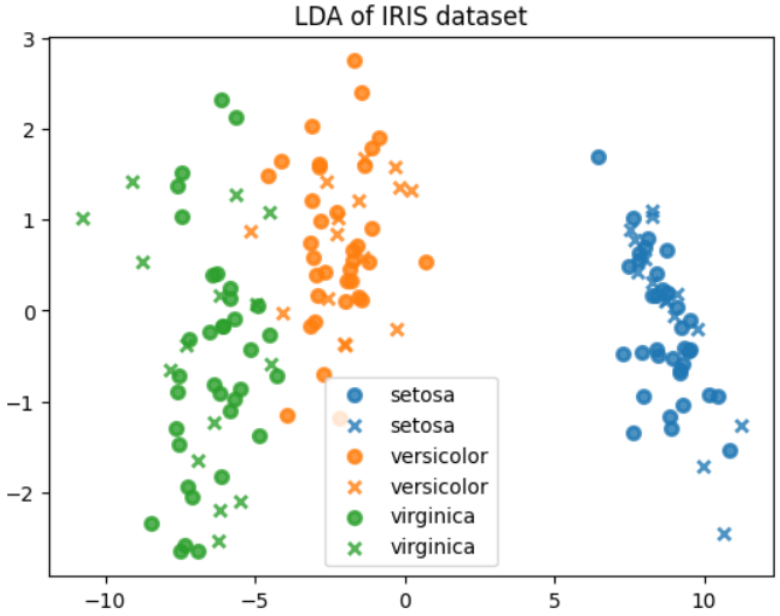
Linear Discriminant Analysis

Example: PCA+SVM vs. LDA on Fisher Iris Data Set

In the Iris Data Set, split the data into 70%-30% train-test with stratification. Project the training data using PCA and LDA, then transform the test data using those projections. Compare the PCA vs. LDA results in 2D.

In addition, compare the result of PCA + SVM vs. LDA in terms of training and testing accuracy.





PCA + SVM Classification

Training Accuracy: 98.10%
Testing Accuracy: 93.33%

LDA Classification

Training Accuracy: 99.05%
Testing Accuracy: 95.56%

Note: PCA and LDA may fail if
- Noise in data are multi-modal or non-Gaussian.
- The underlying manifold is highly nonlinear.

Reference: https://scikit-learn.org/stable/auto_examples/decomposition/plot_pca_vs_lda.html

Outline

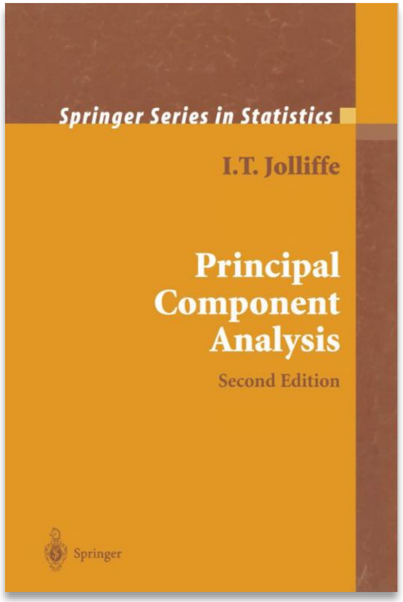
- Dimensionality Reduction
  - Curse of Dimensionality
  - Feature Selection vs. Feature Extraction
  - Principal Components Analysis (PCA)
    - Derivation
  - Non-negative Matrix Factorization (NMF)
  - Independent Components Analysis (ICA)

- Low-dimensional Classification
  - Linear Discriminant Analysis (LDA)

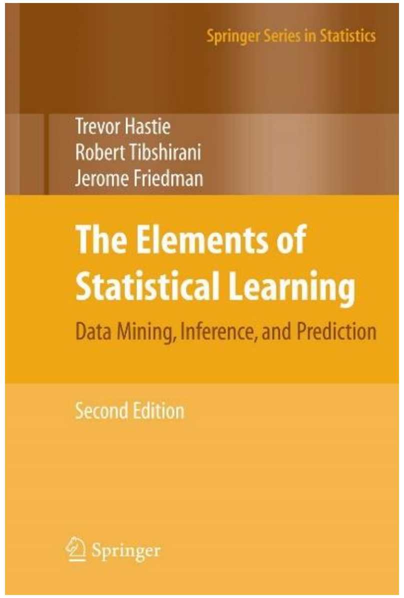
Related methods:

- *PLS (Partial Least Squares)*
  - Low-dimensional Regression
- *NCA (Neighborhood Components Analysis)*
  - Another low-dimensional classifier like LDA.
- *CCA (Canonical Correlation Analysis)*
  - Maximize correlation instead of covariance as in PCA.

Jolliffe (2002)
Principal Component Analysis.
2nd Ed. Springer.



Hastie *et al.* (2008)
The Elements of Statistical Learning.
2nd Ed. Springer.



Further Reading

- “Curse of Dimensionality.” Bellman R.E. Adaptive Control Processes. Princeton University Press, Princeton, NJ, 1961.
- Chandrashekar and Sahin (2014). A Survey on Feature Selection Methods. *Computers and Electrical Engineering*. Vol. 40, No. 1, 16-28. https://www.sciencedirect.com/science/article/pii/S0045790613003066
- https://www.izen.ai/blog-posts/feature-selection-filter-method-wrapper-method-and-embedded-method/
- Ron Kohavi and George H. John (1997). "Wrappers for feature subset selection", Artificial Intelligence. Vol. 97, 273-324. doi: 10.1016/S0004-3702(97)00043-X
- Jolliffe and Cadima (2016). Principal component analysis: a review and recent developments. http://rsta.royalsocietypublishing.org/lookup/doi/10.1098/rsta.2015.0202
- Lee and Seung (2000). Algorithms for Non-negative Matrix Factorization. Advances in Neural Information Processing Systems, 13. https://papers.nips.cc/paper/2000/file/f9d1152547c0bde01830b7e8bd60024c-Paper.pdf
- Hyvarinen and Oja (2000). ICA: Algorithms and Applications. Neural Networks, 13(4-5): 411-430. https://www.cs.helsinki.fi/u/ahyvarin/papers/NN00new.pdf
- https://www.kaggle.com/code/tirendazacademy/penguin-dataset-data-visualization-with-seaborn
- https://scikit-learn.org/stable/auto_examples/decomposition/plot_pca_vs_lda.html
- https://scikit-learn.org/stable/auto_examples/decomposition/plot_pca_iris.html
- https://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html
- Van der Maaten et al. (2009). Dimensionality Reduction: A Comparative Review, *Journal of Machine Learning Research*.