DATA MINING

Dimensionality Reduction

Principal Component Analysis

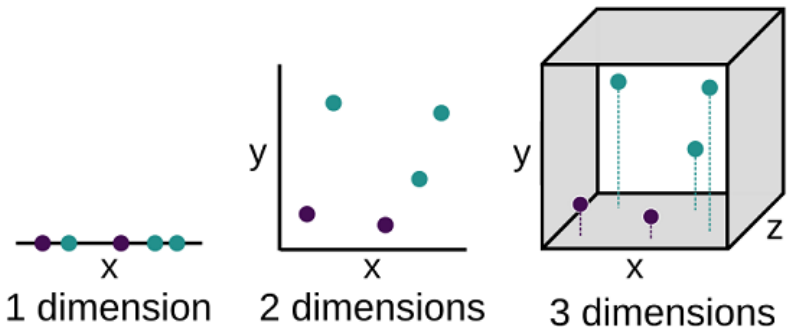
Ashish Pujari

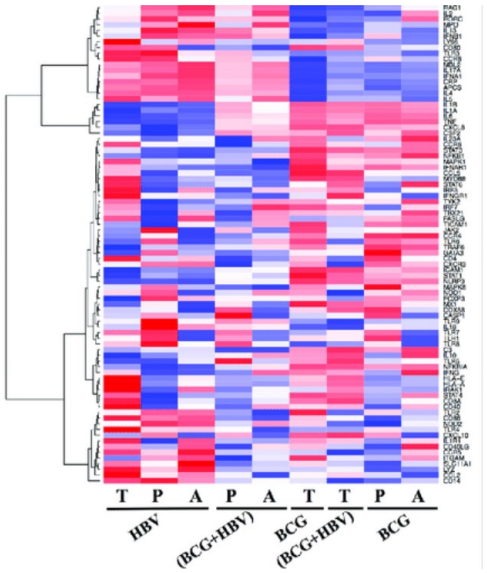
Lecture Outline

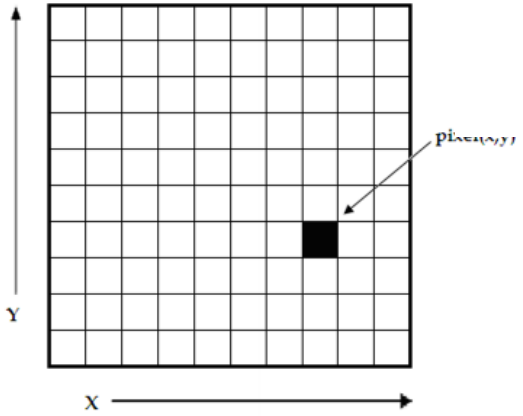
- Dimensionality Reduction
- Linear Algebra Review
- Principal Component Analysis
- Linear Discriminant Analysis

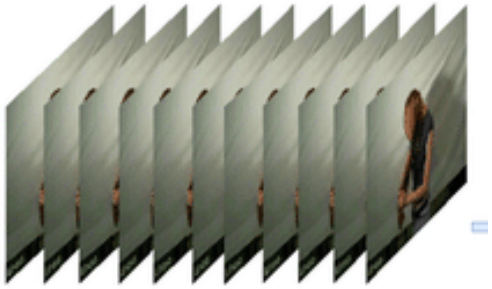
DIMENSIONALITY REDUCTION

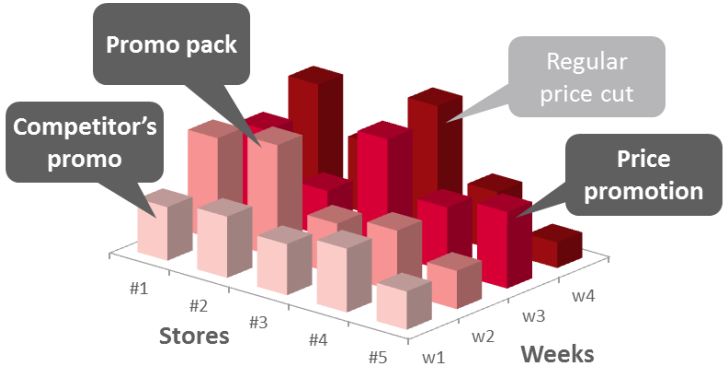
High Dimensional Data





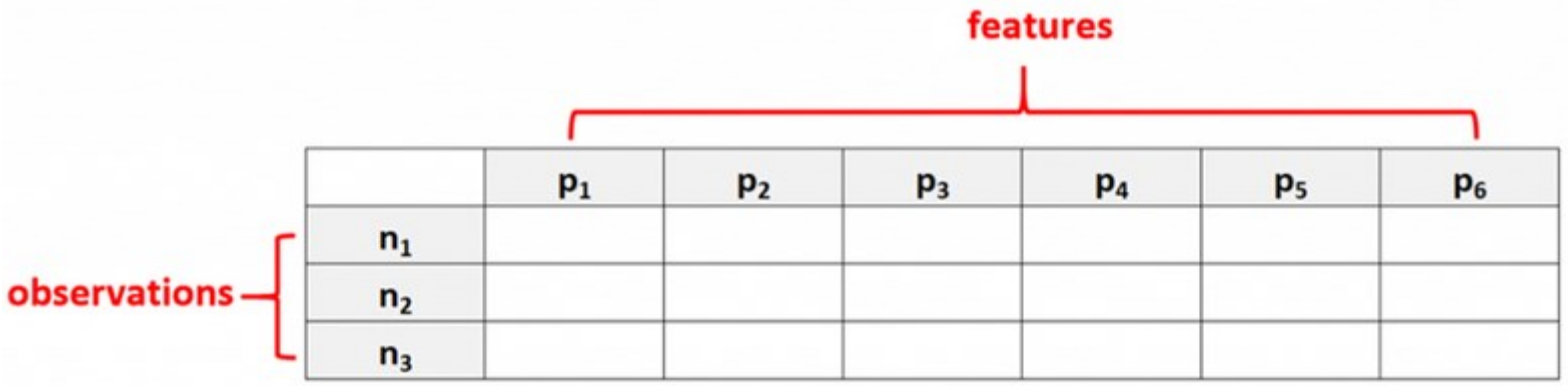






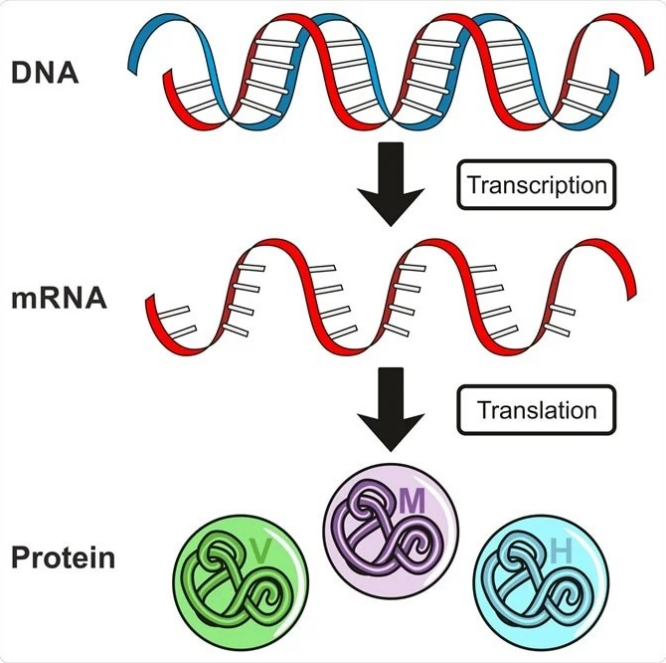
High Dimensional Data

- In complex systems such as neuroscience, photo-science, meteorology and oceanography, etc., the number of variables to measure can be unwieldy and at times deceptive, because the underlying dynamics can often be quite simple.
- Dataset in which the number of features p is larger than the number of observations N, often written as p >> N.

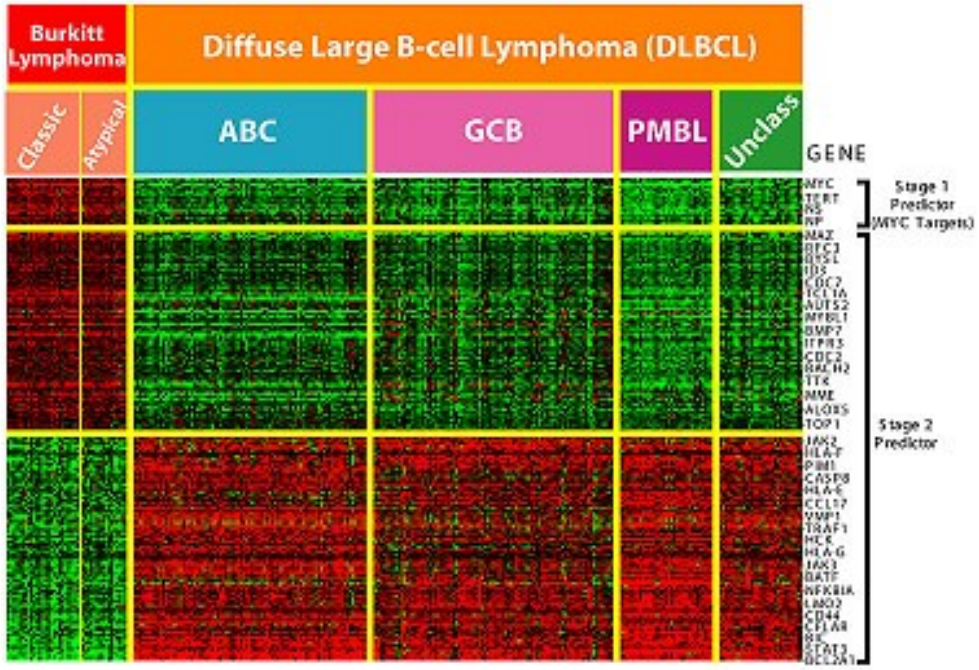


High Dimensional Data - Genomics

- Gene expression controls the amount and type of proteins that are expressed in a cell at any given point in time. It is essential for understanding protein function, biological pathways, and cellular responses to external and internal stimuli



Source: udaix/Shutterstock.com

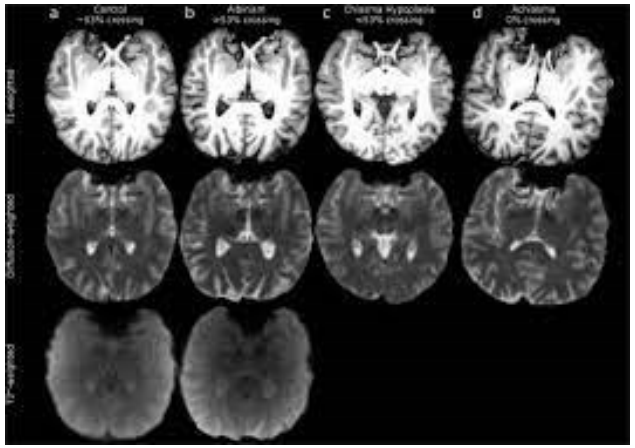


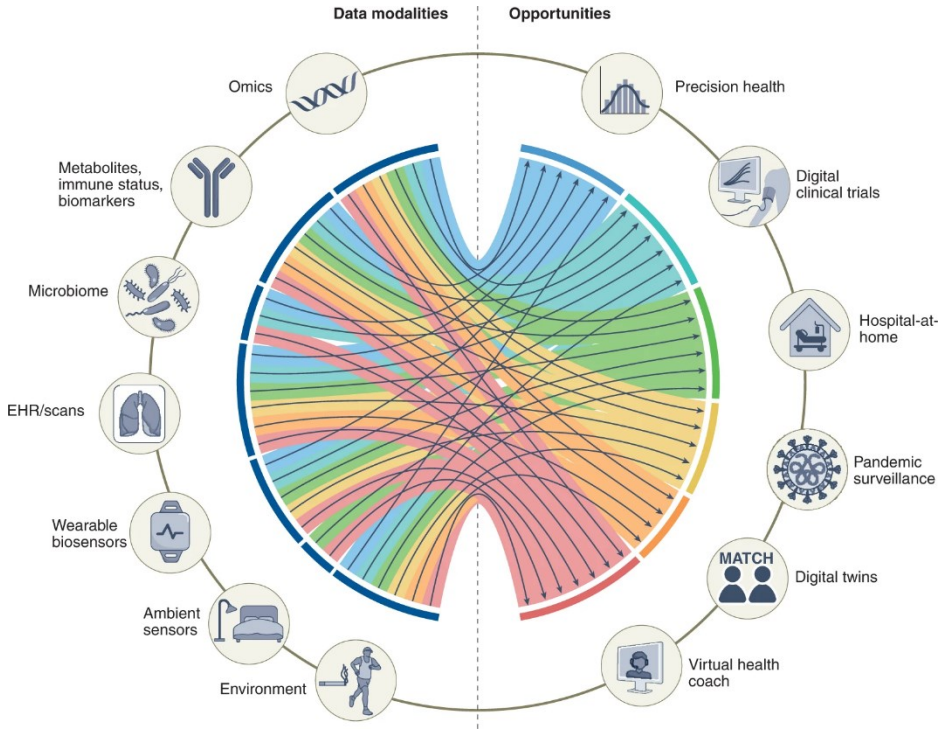
High-dimensional microarray analysis.

http://archive.ics.uci.edu/ml/datasets/gene+expression+cancer+RNA-Seq

High Dimensional Data - Healthcare

- High dimensional data is common in healthcare datasets where the number of features for a given individual can be massive
- E.g., MRI, blood pressure, resting heart rate, immune system status, surgery history, height, weight, existing conditions, etc.





https://www.nature.com/articles/s41591-022-01981-2

High Dimensional Data - Financial

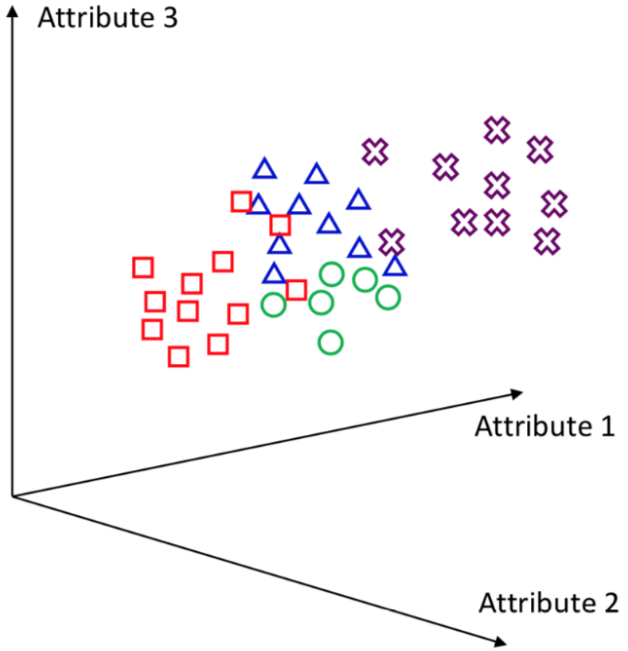
- High dimensional data is also common in financial datasets where the number of features for a given stock can be quite large (i.e., PE Ratio, Market Cap, Trading Volume, Dividend Rate, etc.)

statements

Index	DJIA S&P500	P/E	30.35	EPS (ttm)	9.40	Insider Own	0.06%	Shs Outstand	7.50B	Perf Week	0.01%
Market Cap	2198.40B	Forward P/E	26.50	EPS next Y	10.77	Insider Trans	-17.06%	Shs Float	7.48B	Perf Month	-5.04%
Income	71.19B	PEG	1.89	EPS next Q	2.19	Inst Own	71.90%	Short Float	0.49%	Perf Quarter	-5.73%
Sales	184.90B	P/S	11.89	EPS this Y	39.70%	Inst Trans	-0.02%	Short Ratio	1.01	Perf Half Y	-7.16%
Book/sh	21.32	P/B	13.38	EPS next Y	15.41%	ROA	21.60%	Target Price	369.58	Perf Year	10.47%
Cash/sh	16.27	P/C	17.54	EPS next 5Y	16.08%	ROE	48.40%	52W Range	238.07 - 349.67	Perf YTD	-15.17%
Dividend	2.48	P/FCF	50.65	EPS past 5Y	25.70%	ROI	28.20%	52W High	-18.41%	Beta	0.93
Dividend %	0.87%	Quick Ratio	2.20	Sales past 5Y	13.00%	Gross Margin	68.80%	52W Low	19.84%	ATR	7.60
Employees	181000	Current Ratio	2.20	Sales Q/Q	20.10%	Oper. Margin	42.50%	RSI (14)	42.41	Volatility	2.75% 2.22%
Optionable	Yes	Debt/Eq	0.42	EPS Q/Q	22.30%	Profit Margin	38.50%	Rel Volume	0.61	Prev Close	285.30
Shortable	Yes	LT Debt/Eq	0.38	Earnings	Apr 26 AMC	Payout	24.30%	Avg Volume	36.34M	Price	285.30
Recom	1.60	SMA20	-4.79%	SMA50	-3.50%	SMA200	-6.47%	Volume	0	Change	0.00%

High Dimensional Data

- It is not easy to figure out the trend in 3D
- Is there a better representation of the data?
- How can we find lower dimensional representation that keeps the most information about the original data?



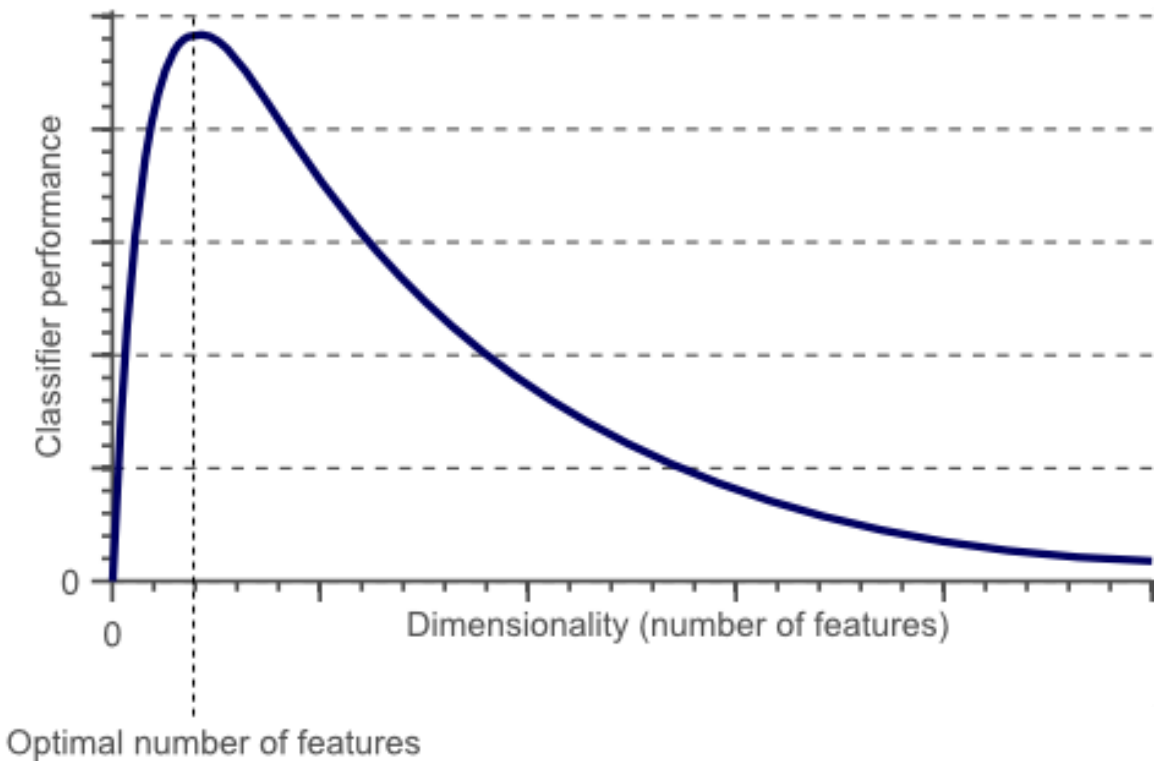
Curse of Dimensionality

As the number of dimensions in the data increases it impacts:

- Sparsity
  - Increases the volume of the feature space, therefore, data becomes more spread out (sparse)
- Sample Size
  - Required sample size n will grow exponentially with data that has d dimensions and might quickly become unmanageable.
- Metrics
  - Distance metrics loose their meaning in higher dimensions
- Performance
  - Machine Learning Model training time increases specially for parametric models.
  - Increase in ‘noisy’ or irrelevant features could cause degradation in model performance
  - Complex models that are harder to interpret than those with a low number of features.

Curse of Dimensionality

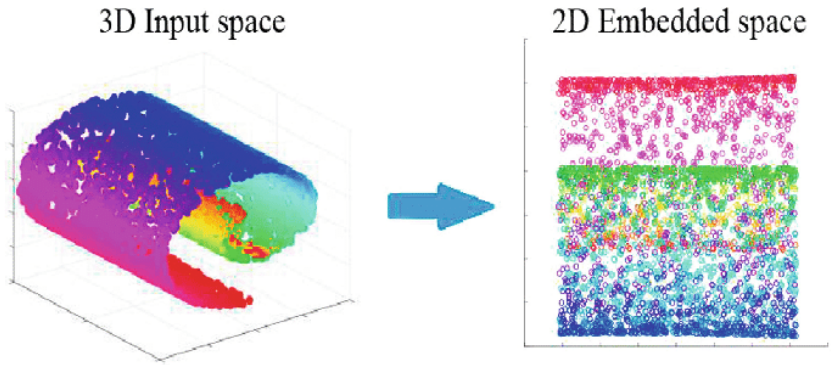
- As the number of features (dimensions) grows the amount of data we need to generalize grows exponentially



Dimensionality Reduction

- Goal
  - Map high dimensional data onto lower-dimensions in a manner that preserves distances / similarities.
  - Simplify understanding of data, either numerically or visually without loss of data integrity.
- Fundamental Idea
  - Exploit redundancy in the data to find a lower dimensional representation.

$$X = \{x_{1,}x_{2,}, \dots, x_{n,} \in \mathbb{R}^{\mathrm{D}}\} \rightarrow Y = \{y_{1,}y_{2,}, \dots, y_{n,} \in \mathbb{R}^{\mathrm{M}}\}$$



Data Compression

- Consider vectors

$$v_1 = (1,2,3), v_2 = (2,4,6), v_3 = (3,6,9), v_4 = (4,8,12)$$

  - Need to store 12 integers

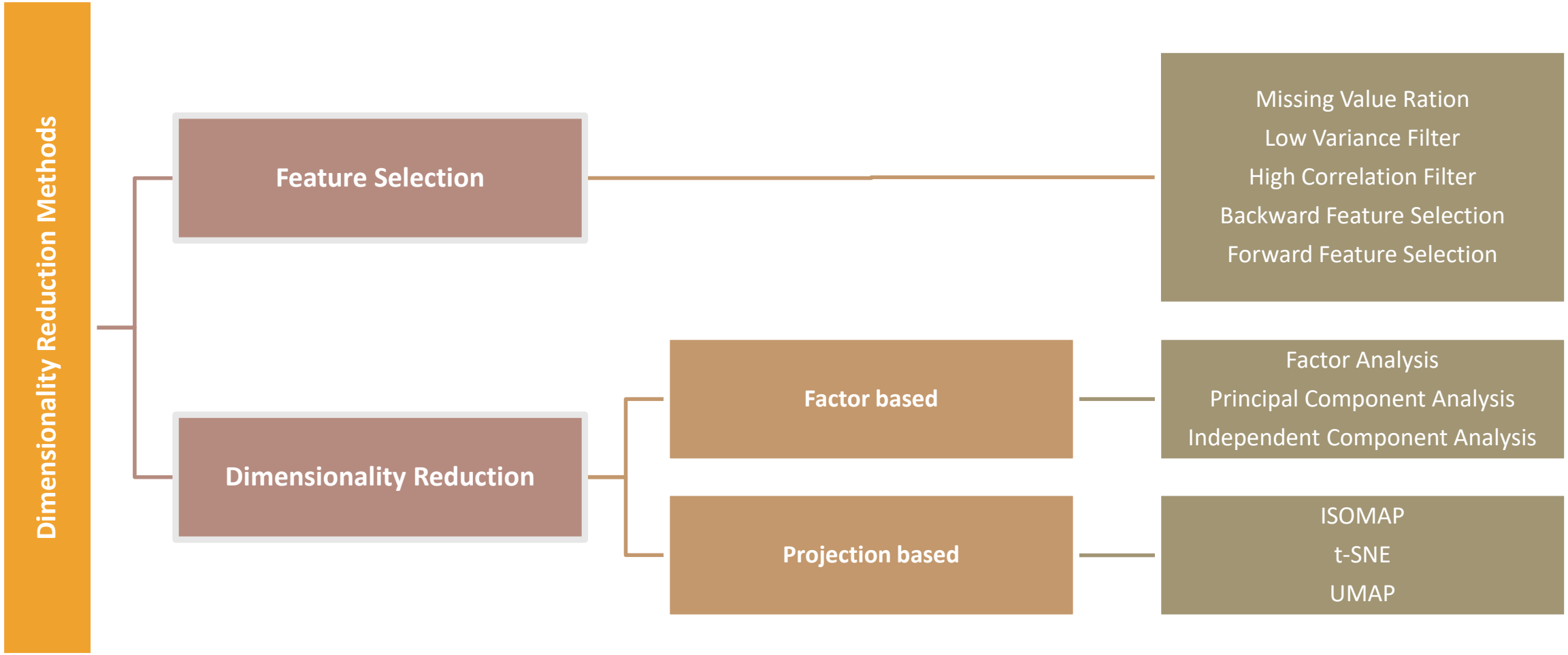
- However, they are all related

$$v_1 = 1 \cdot (1,2,3), v_2 = 2 \cdot (1,2,3), v_3 = 3 \cdot (1,2,3), v_4 = 4 \cdot (1,2,3)$$

- We can save $v_0 = (1,2,3)$ and multipliers 1,2,3,4

  - Only 7 integers need to be stored

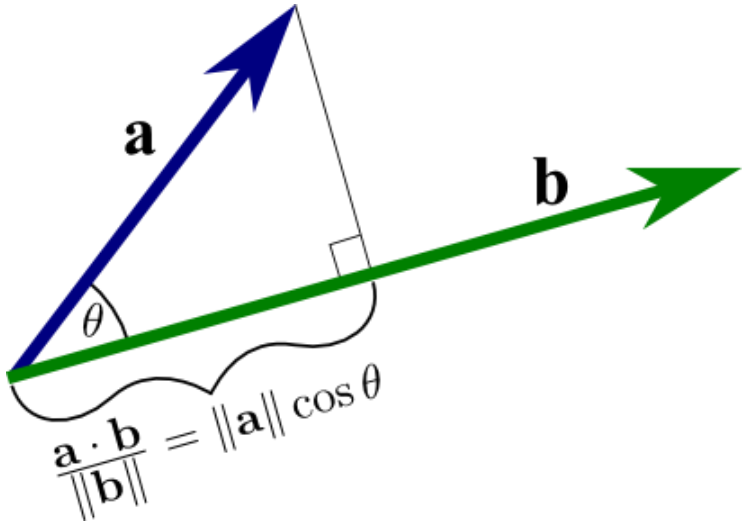
Dimensionality Reduction Methods



LINEAR ALGEBRA REVIEW

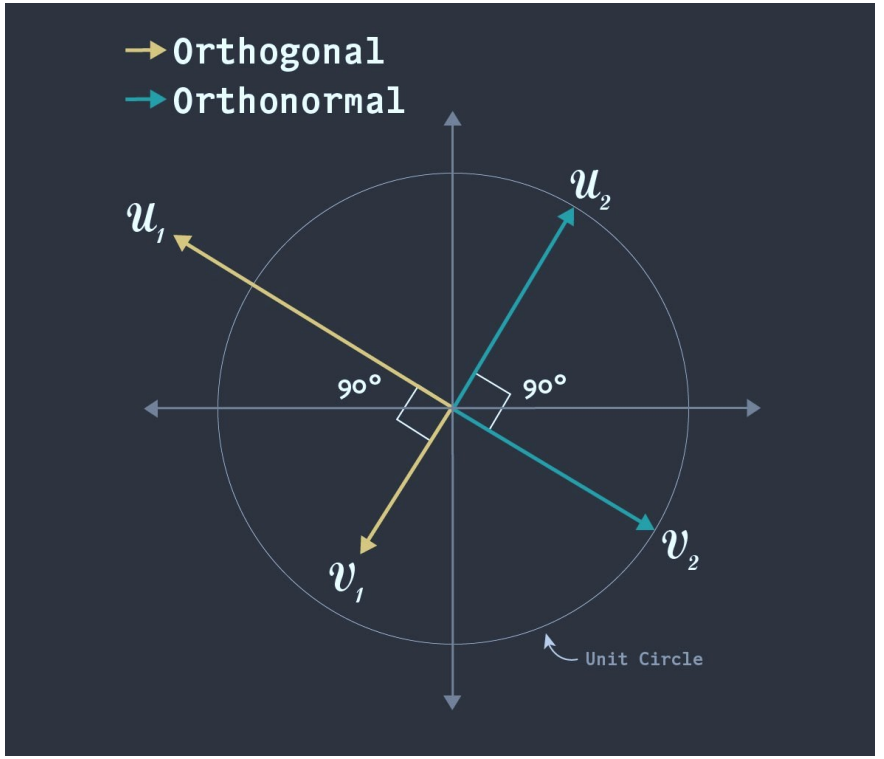
Dot Products

- Dot product between two vectors is based on the projection of one vector onto another
- Output of vector dot products :
  - The angle between the vectors is obtuse if the dot product is < 0
  - The angle between the vectors is acute if the dot product is > 0
  - The vectors are orthogonal (at right angles) if the dot product = 0



Orthogonality

- Two vectors u and v are considered to be **orthogonal** $u \perp v$ when the angle between them is 90° i.e. orthogonal vectors are perpendicular to each other.
- The dot product $u \cdot v = 0$
- They are **orthonormal** if they are orthogonal, and each vector has unit length.



Orthogonality (and orthonormality) is necessary to project vectors onto subspaces, find better estimates of nonlinear objects, and measure many properties of vector spaces.

Covariance Matrix

- Variance measures the variation of a single random variable

$$\sigma_x^2 = \frac{1}{n-1}\sum_{i=1}^{N}(x_i - \bar{x})^2$$

- Covariance is a measure of how much two random variables vary together

$$\sigma(x, y) = \frac{1}{n-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})$$

- Using covariance we can calculate entries of the covariance matrix, which is a square matrix given by $C_{i,j} = \sigma(x_i, x_j)$ where our data set is expressed by the matrix $X \in \mathbb{R}^{n \times d}$

$$C = \frac{1}{n-1}\sum_{i=1}^{N}(X_i - \bar{X})(X_i - \bar{X})^T$$

Eigenvectors and Eigenvalues

- A vector $\boldsymbol{x}$ of dimension N is an eigenvector of a square $N \times N$ matrix $\boldsymbol{A}$ and λ is the corresponding eigenvalue if

$$\boldsymbol{Ax} = \lambda \boldsymbol{x}$$

Almost all vectors change direction, when they are multiplied by $\boldsymbol{A}$. Eigenvectors are special unit vectors $\boldsymbol{x}$ that are in the same direction as $\boldsymbol{Ax}$.

Eigenvalue λ tells whether the special vector $\boldsymbol{x}$ is stretched or shrunk or reversed or left unchanged when it is multiplied by $\boldsymbol{A}$

$$\boldsymbol{Ax} = \lambda \boldsymbol{Ix}$$

$$\boldsymbol{Ax} - \lambda \boldsymbol{Ix} = 0$$

$$(\boldsymbol{A} - \lambda \boldsymbol{I})\, \boldsymbol{x} = 0$$

$$det(\boldsymbol{A} - \lambda \boldsymbol{I}) = 0$$

Eigen-decomposition

- Eigen-decomposition is the factorization of a matrix into a canonical form, whereby the matrix is represented in terms of its eigenvalues and eigenvectors
- If a square matrix A is diagonalizable, then there is a matrix P such that

$$A = P\,D\,P^{-1}$$

Original Matrix		Eigenvectors Matrix		Eigenvalues Matrix		Inverse of Eigenvectors Matrix	
$\begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}$	$=$	$\begin{bmatrix} -1 & -1 \\ 2 & 1 \end{bmatrix}$		$\begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix}$		$\begin{bmatrix} 1 & 1 \\ -2 & -1 \end{bmatrix}$	

- The nondiagonal matrices P and P^{-1} are inverses of each other
- Provides valuable insights into its properties and makes matrix calculations (power, etc.) much easier

Eigen-decomposition

- Eigen-decomposition only exists for square matrices, and even among square matrices sometimes it doesn't exist
- When $\boldsymbol{A}$ is squared, the eigenvectors stay the same; the eigenvalues are squared

Matrix Type	Properties
Square Symmetric	Eigenvalues: always real, non-negative Eigenvectors: always orthogonal
Square Asymmetric	Eigenvalues: can be complex Eigenvectors: don’t necessarily exist
Non-square	Eigen decomposition not possible

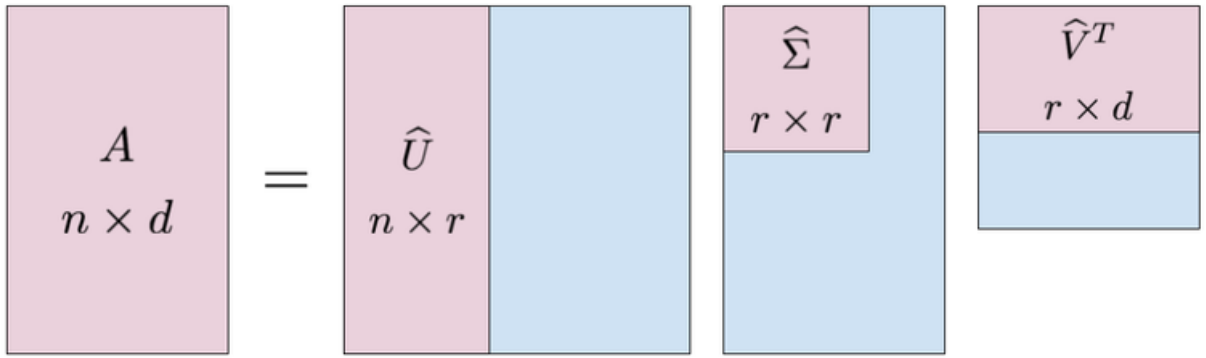
Singular Value Decomposition (SVD)

- The SVD of a matrix A is a factorization of that matrix into three matrices given by the formula :

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

- Vectors in the matrices U and V in the SVD are orthonormal and not necessarily the inverse of one another
- SVD can be used to compute optimal low-rank approximations of arbitrary matrices.
- SVD always exists for any rectangular or square matrix

Singular Value Decomposition (SVD)

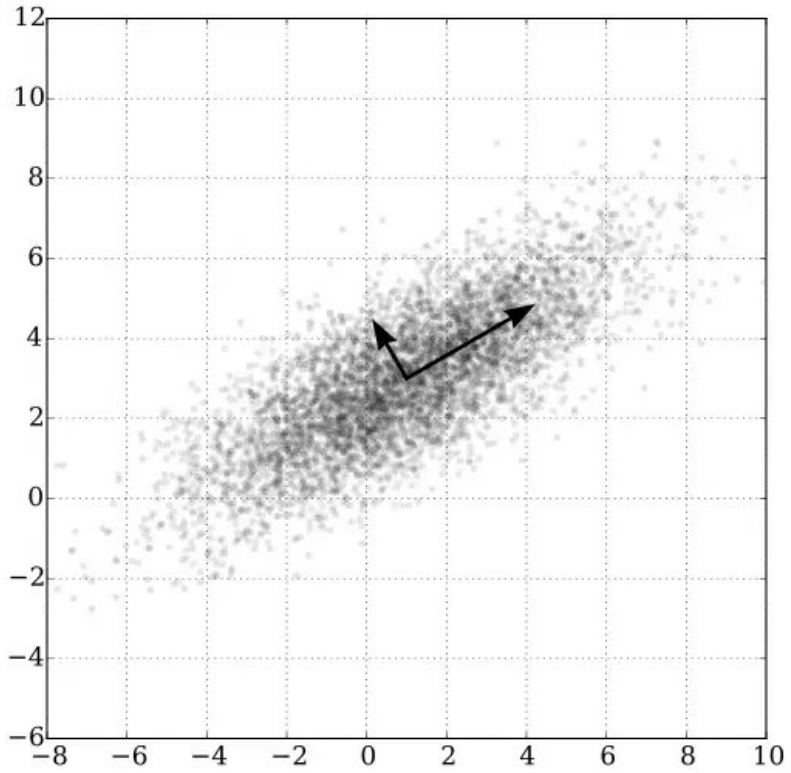


U: $n \times n$ matrix of the orthonormal eigenvectors of AA^T.
Σ: $n \times d$ diagonal matrix of the singular values of A which are the square roots of the eigenvalues of A^TA. The number of non-zero singular values is the rank of A
V^T: transpose of $d \times d$ matrix containing the orthonormal eigenvectors of A^TA

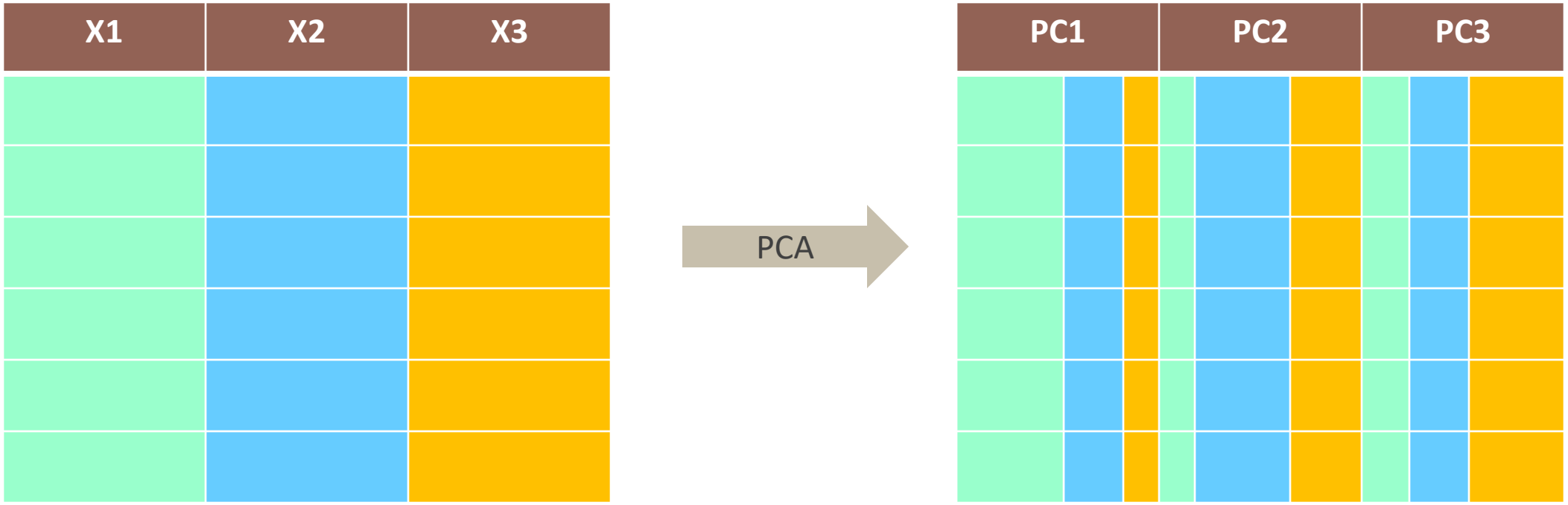
PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal Component Analysis (PCA)

- What is PCA
  - PCA finds a lower-dimensional representation of data by constructing new features (Principal Components) which are linear combinations of the original features
- Objectives
  - To reduce the number of variables
  - Examine the relationship between variables
  - Address the problem of multicollinearity
- Assumptions
  - Original variables should be normalized
  - Factors are independent of each other
  - There exist some underlying factors that can describe the original variables



Principal Component Analysis (PCA)



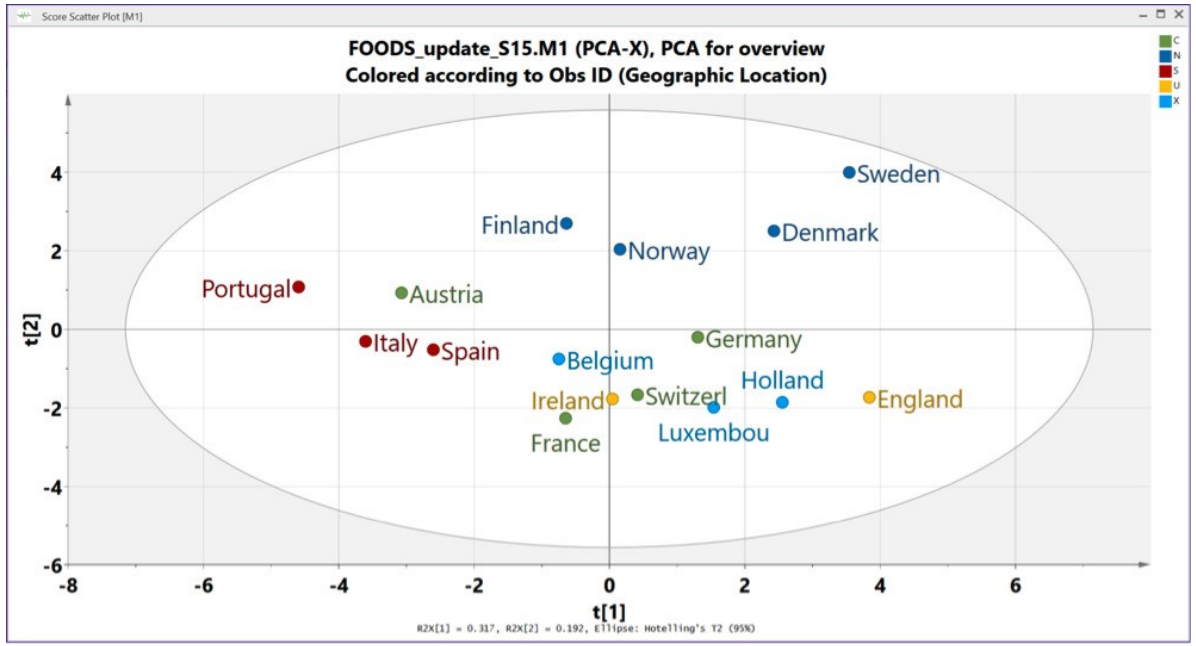
$$PC1 = a_1x_1 + a_2x_2 + a_3\,x_3$$

$$PC2 = b_1x_1 + b_2x_2 + b_3\,x_3$$

$$PC3 = c_1x_1 + c_2x_2 + c_3\,x_3$$

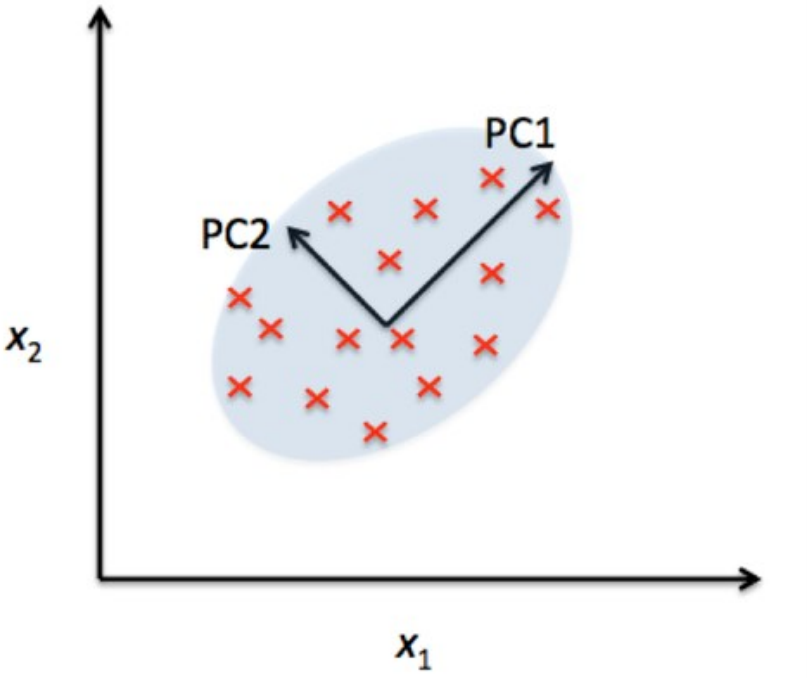
PCA Applications

- Data visualization
- Data compression (Lossy)
- Noise reduction
- Factor analysis
- Feature extraction
  - High dimensionality of the input features
  - Applied to data having multi-collinearity between the features/variables



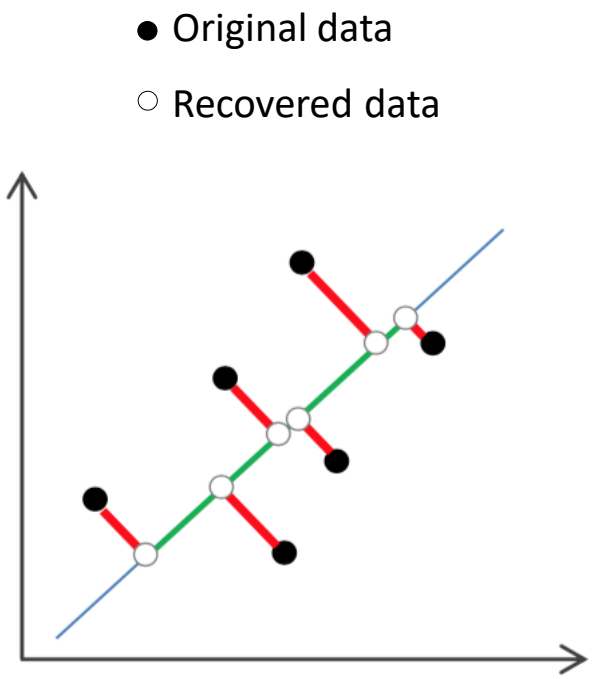
PCA Solution Approach

- PCA seeks a space of lower dimensionality known as principal subspace, such that the orthogonal projection of data points into this subspace maximizes the variance of the projected points
- This is done by projecting (dot product) the original data into the reduced PCA space using the eigenvectors of the covariance matrix (i.e Principal Components)
- The resulting projected data are linear combinations of the original data capturing most of the variance in the data
- The first component explains the most variance in the data with each subsequent component explaining less



PCA: Solution Approach

- Let's say we have an i.i.d dataset $X = \{x_{1,}x_{2,}, \dots, x_{n,} \in \mathbb{R}^{D}\}$ with D dimensions and a mean value of 0
- Goal is to find projections that are as similar to the original data as possible but have lower dimensionality $(M < D)$.
- There are two approaches:
  1. Maximum variance
  2. Minimum error



PCA: Solution Approach

$$\underbrace{Variance\ of\ data}_{\text{fixed}} = \underbrace{captured\ variance}_{\text{maximize}} + \underbrace{reconstruction\ error}_{\text{minimize}}$$

- Maximum variance formulation
  - Find a low-dimensional representation which maximizes the variance of the projected data.

$$\max_{X \in \mathbb{R}^{m \times p}} \|AX\|^2 \text{ subject to } X^T X = I$$

- Minimum error formulation
  - Find a low-dimensional representation which minimizes the average reconstruction error between the original data and the reconstructed data.

$$\min_{X \in \mathbb{R}^{m \times p}} \|A - AXX^T\|^2 \text{ subject to } X^T X = I$$

PCA: Maximum Variance

- Consider the simplest case $\boldsymbol{M} = 1$. We define a vector $\boldsymbol{w_1} \in \boldsymbol{R^D}$ as the direction of the lower dimensional space.
- Since we are only interested in the direction, we set $\boldsymbol{w_1}$ to be of unit length. i.e.

$$\mathbf{w}_1^{\mathbf{T}} \boldsymbol{w_1} = 1$$

- Then the data observations $\mathbf{x_n}$ can be projected onto this new space as

$$\hat{\mathbf{x}}_\mathbf{n} = \mathbf{w}_1^{\mathbf{T}} \mathbf{x_n}$$

- If $\bar{x}$ is the mean of the data observations in the original space, then the mean of the samples in the projected space is given by

$$\hat{\bar{\mathbf{x}}} = \mathbf{w}_1^{\mathbf{T}} \bar{\mathbf{x}}$$

PCA: Maximum Variance

- Variance of the projected data can be derived as shown on the right:

$$
\begin{aligned}
\sigma^2(\hat{\mathbf{x}}) &= \frac{1}{N}\sum_{n=1}^{N}(\hat{\mathbf{x}}_n - \hat{\bar{\mathbf{x}}})^2 \\
&= \frac{1}{N}\sum_{n=1}^{N}(\mathbf{w}_1^T\mathbf{x}_n - \mathbf{w}_1^T\bar{\mathbf{x}})^2 \\
&= \frac{1}{N}\sum_{n=1}^{N}(\mathbf{w}_1^T\mathbf{x}_n - \mathbf{w}_1^T\bar{\mathbf{x}})(\mathbf{w}_1^T\mathbf{x}_n - \mathbf{w}_1^T\bar{\mathbf{x}})^T \\
&= \frac{1}{N}\sum_{n=1}^{N}(\mathbf{w}_1^T\mathbf{x}_n - \mathbf{w}_1^T\bar{\mathbf{x}})(\mathbf{x}_n^T\mathbf{w}_1 - \bar{\mathbf{x}}^T\mathbf{w}_1) \\
&= \frac{1}{N}\sum_{n=1}^{N}\mathbf{w}_1^T(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n^T - \bar{\mathbf{x}}^T)\mathbf{w}_1 \\
&= \frac{1}{N}\sum_{n=1}^{N}\mathbf{w}_1^T(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T\mathbf{w}_1 \\
&= \mathbf{w}_1^T\mathbf{S}\mathbf{w}_1
\end{aligned}
$$

- Where **S** is the covariance matrix of the observed data in the original high dimensional space.

$$\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(x_n - \bar{x})^T$$

PCA: Maximum Variance

- We need to maximize the variance of $\hat{x}$: $\sigma^2(\hat{x}) = \mathbf{w_1}^{\mathbf{T}}\mathbf{S}\,\mathbf{w_1}$
- To prevent the trivial solution where $\|w_1\| \rightarrow \infty$, we make use of the unit norm constraint we set earlier $\boldsymbol{w_1^T w} = 1$
- We introduce a Lagrange multiplier λ_1 and formulate our optimization objective as :

$$J(\boldsymbol{w_1}) = \boldsymbol{w_1^T S w_1} + \lambda_1\left(1 - \boldsymbol{w_1^T w_1}\right)$$

$$\frac{\partial J(\boldsymbol{w_1})}{\partial \boldsymbol{w_1}} = 2\mathbf{S}\boldsymbol{w_1} - 2\lambda_1\boldsymbol{w_1}$$

- This shows that at the stationary point, $\boldsymbol{w_1}$ must be an eigenvector of **S** and λ_1 the eigenvalue. corresponding to the eigenvector $\boldsymbol{w_1}$

$$\mathbf{S}\boldsymbol{w_1} = \lambda_1\boldsymbol{w_1}$$

PCA: Maximum Variance

- Left-multiplying with $\boldsymbol{w_1^T}$

$$\boldsymbol{w_1^T S w_1} = \boldsymbol{w_1^T} \lambda_1 \boldsymbol{w_1}$$

$$\boldsymbol{w_1^T S w_1} = \lambda_1 \boldsymbol{w_1^T w_1}$$

$$\boldsymbol{w_1^T S w_1} = \lambda_1$$

- The maximum variance in the lower dimensional space is equal to the eigenvalue corresponding to eigenvector $\boldsymbol{w_1}$

PCA: Maximum Variance

- Identifying additional components
  - We can identify additional principal components by choosing directions that maximize variance while being orthogonal to the existing ones.

$$\boldsymbol{w}_2 \perp \boldsymbol{w_1} \ , \ \mathbf{w_2}^{\mathbf{T}}\boldsymbol{w}_2 = 1$$

- General case of a lower dimensional space
  - with $\boldsymbol{M}$ dimensions with $\boldsymbol{M} < \boldsymbol{D}$, the principal components are the eigenvectors $\boldsymbol{w_1}$, $\boldsymbol{w}_2$, … $\boldsymbol{w}_{\mathrm{m}}$ corresponding to the M largest eigenvalues: $\lambda_1, \lambda_2, \ldots \lambda_m$

Relation between PCA and SVD

- Applying SVD to the data matrix $\mathbf{X}$

$$\mathrm{X} = \mathbf{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\boldsymbol{T}}$$

$$\frac{1}{n-1}\mathbf{X}\boldsymbol{X}^{\boldsymbol{T}} = \frac{1}{n-1}(\mathbf{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\boldsymbol{T}})(\mathbf{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\boldsymbol{T}})^{\boldsymbol{T}}$$

$$= \frac{1}{n-1}\left(\mathbf{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\boldsymbol{T}}\right)\left(\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^{\boldsymbol{T}}\right)$$

- As V is an orthogonal matrix $\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{V} = \boldsymbol{I}$

$$\frac{1}{n-1}\mathbf{X}\boldsymbol{X}^{\boldsymbol{T}} = \frac{1}{n-1}\left(\boldsymbol{U}\boldsymbol{\Sigma}^{\boldsymbol{2}}\boldsymbol{U}^{\boldsymbol{T}}\right)$$

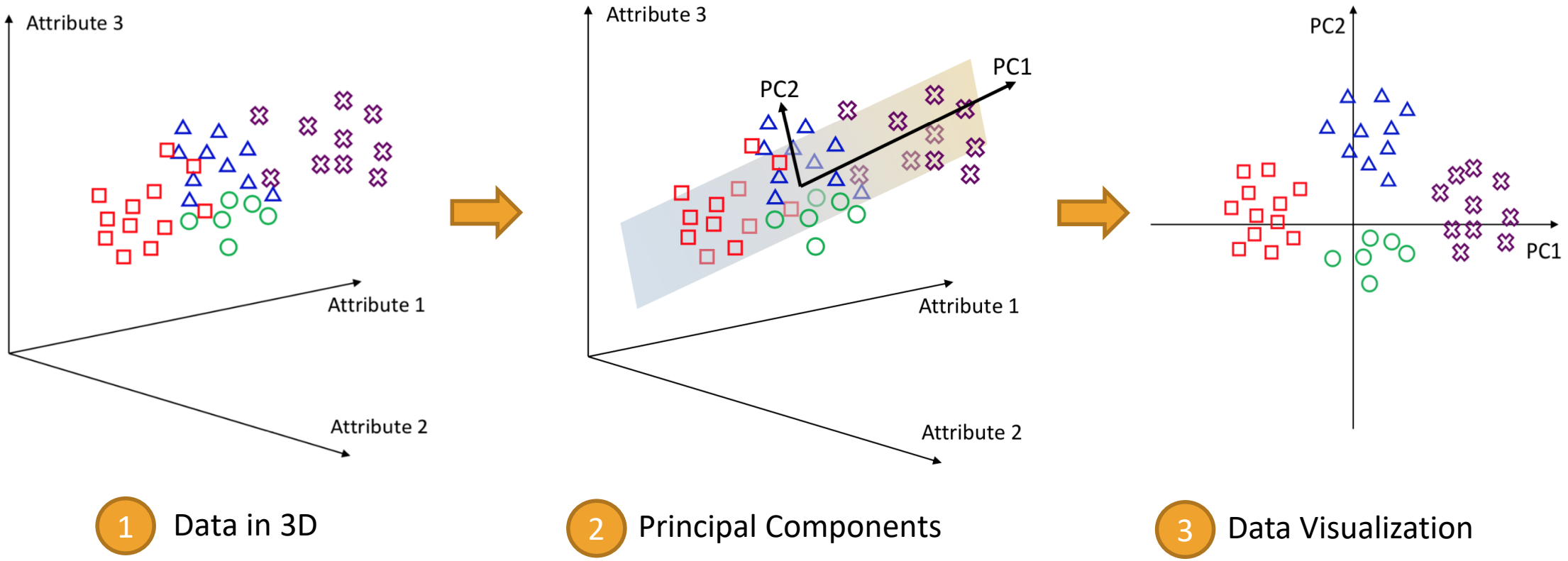
- The square roots of the eigenvalues of $\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}}$ are the singular values of $\boldsymbol{X}$.
- SVD can therefore be used to perform PCA without forming the covariance matrix as $\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}}$ can cause loss of numerical precision.

PCA: Steps

Let X be a matrix containing the original data with shape *[n_samples, n_features]*

1. Input variables of X are z-scored such each original variable has zero mean and unit standard deviation.
2. Construction and eigendecomposition of the covariance matrix. Covariance is equal to the correlation matrix for z-scored data.
3. Eigenvalues are then sorted in a decreasing order representing decreasing variance in the data (the eigenvalues are equal to the variance).
4. Finally, the projection of the original normalized data onto the reduced PCA space is obtained by multiplying (dot product) the originally normalized data by the leading eigenvectors of the covariance matrix i.e. the PCs.
5. To visualize the projected data as well as the contribution of the original variables, in a joint plot, we can use the biplot.

PCA: Data Visualization



1 Data in 3D

2 Principal Components

3 Data Visualization

Number of Principal Components

- For N original dimensions, sample covariance matrix is $N \times N$, and has up to N eigenvectors. So, N PCs.
- We can ignore the components of lesser significance
- Some information is lost, but if the eigenvalues are small, you don't lose much
  - N dimensions in original data
  - Calculate N eigenvectors and eigenvalues
  - Choose only the first D eigenvectors, based on their eigenvalues
  - Final data set has only D dimensions

Example 1

- Covariance matrix

$$\Sigma = \begin{bmatrix} 1 & -2 & 0 \\ -2 & 5 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

- Eigenvalue-eigenvector pairs are

$$\lambda_1 = 5.83, \quad x_1 = (0.383, -0.924, 0)$$
$$\lambda_2 = 2, \quad x_2 = (0, 0, 1)$$
$$\lambda_3 = 0.17, \quad x_3 = (0.924, 0.383, 0)$$

  - $\lambda_1 > \lambda_2 > \lambda_3 \Rightarrow$ order of importance: x_1, x_2, x_3
- Dimensionality reduction
  - Pick eigenvectors (PC) with largest p eigenvalues
  - If $p = 1$, pick x_1
  - If $p = 2$, pick x_1 and x_2

Example 1 - How Many PCs?

- Comparison of recovered matrices with $p = 1,2$ and original matrix

Original data

0.8	4.4	-0.9
5.8	12.4	6.1
-3.2	-14.6	-5.9
-6.2	-15.6	-1.9
-2.2	-8.6	2.1
1.8	8.4	0.1
4.8	8.4	5.1
1.8	13.4	6.1
-3.2	-12.6	-6.9
-4.2	-10.6	-7.9
2.8	19.4	6.1
-0.2	1.4	2.1
1.8	-5.6	-3.9

Recovered data with $p = 2$

1.2	4.3	-0.9
3.4	13.0	6.0
-3.8	-14.4	-5.9
-4.4	-16.1	-1.9
-2.4	-8.6	2.1
2.3	8.2	0.1
2.4	9.1	5.0
3.4	12.9	6.1
-3.3	-12.6	-6.9
-2.8	-11.0	-7.9
5.0	18.8	6.1
0.3	1.2	2.1
-1.2	-4.8	-4.0

Recovered data with $p = 1$

0.9	3.5	1.3
3.6	13.4	5.0
-3.9	-14.6	-5.5
-4.0	-14.9	-5.6
-1.9	-7.0	-2.6
2.0	7.3	2.7
2.5	9.5	3.6
3.6	13.3	5.0
-3.5	-13.3	-5.0
-3.2	-12.1	-4.5
4.9	18.5	6.9
0.5	1.7	0.6
-1.5	-5.5	-2.0

Example 1 - How Many PCs?

- Comparison of recovered matrices with $p = 1,2$ and original matrix

Original data

0.8	4.4	-0.9
5.8	12.4	6.1
-3.2	-14.6	-5.9
-6.2	-15.6	-1.9
-2.2	-8.6	2.1
1.8	8.4	0.1
4.8	8.4	5.1
1.8	13.4	6.1
-3.2	-12.6	-6.9
-4.2	-10.6	-7.9
2.8	19.4	6.1
-0.2	1.4	2.1
1.8	-5.6	-3.9

Error matrix with $p = 2$

-0.4	0.1	0.0
2.3	-0.6	0.1
0.6	-0.2	0.0
-1.8	0.5	-0.1
0.2	-0.1	0.0
-0.5	0.1	0.0
2.4	-0.7	0.1
-1.6	0.5	0.0
0.1	0.0	0.0
-1.4	0.4	0.0
-2.3	0.6	-0.1
-0.5	0.1	0.0
3.0	-0.8	0.1

Error with $p = 1$

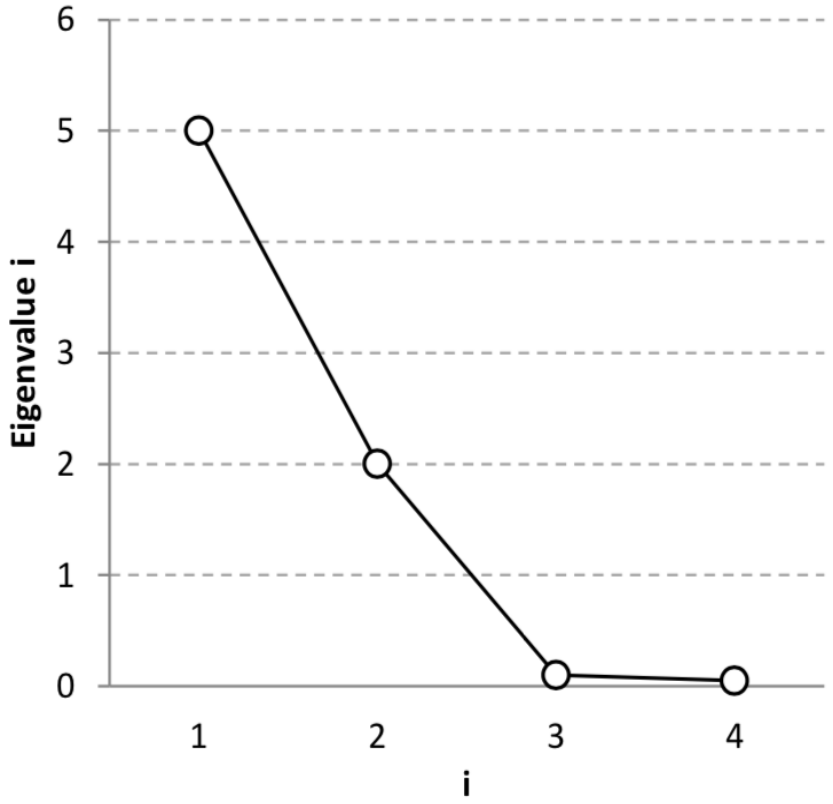
0.2	-0.9	2.2
-2.2	1.0	-1.1
-0.7	0.0	0.5
2.3	0.8	-3.6
0.4	1.7	-4.7
0.2	-1.0	2.7
-2.2	1.2	-1.5
1.8	-0.1	-1.1
-0.3	-0.6	2.0
1.0	-1.5	3.4
2.2	-0.9	0.8
0.7	0.3	-1.4
-3.2	0.2	1.9

- Errors greater than 1 are in red
- Sum of squared errors are 120.2 and 36.9 with $p = 1$ and 2, respectively

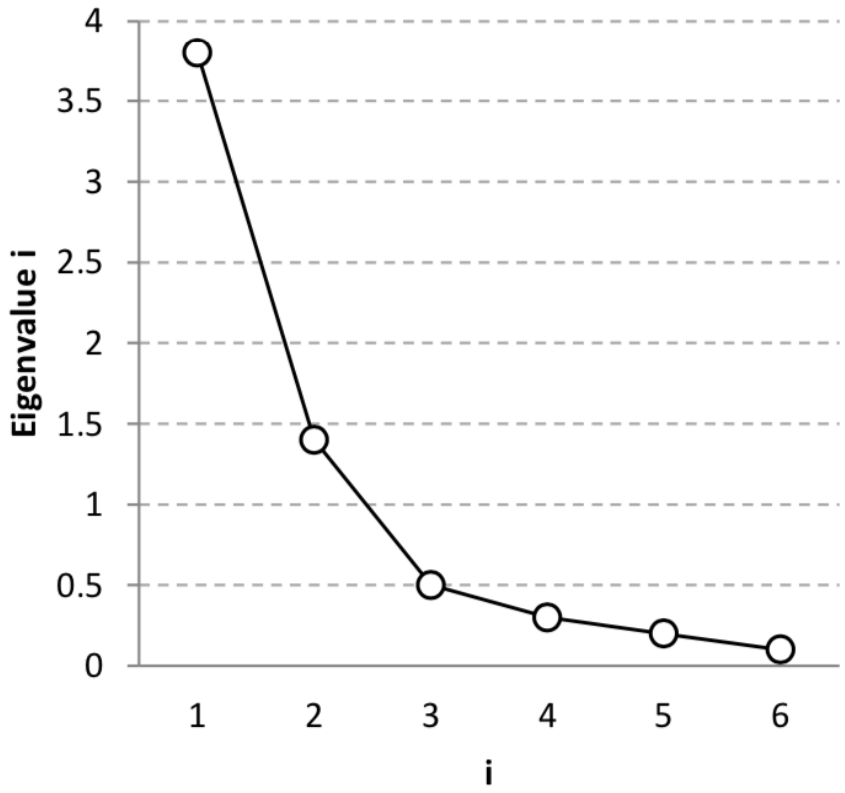
Example 1 - Number of PCs

- How many principal components?
  - There is no definitive answer
- Scree plot: a popular visual aid
  - Larger eigenvalues ⇒ more important eigenvectors
  - Small eigenvalues may be ignored without loss of important information
- Scree plot
  - Plot of λ_i versus i (sorted)
- To determine appropriate number of components (p), look for an elbow

Scree Plot



$p = 2$ is appropriate



$p = 2$ may be appropriate

Eigenvalues are a measure of the amount of variance accounted for by a factor

Example 2 - Iris Dataset

- Number of observations: 150
- Number of attributes: 4 numeric, predictive attributes and the class
- Attribute Information:
  - Sepal length in cm
  - Sepal width in cm
  - Petal length in cm
  - Petal width in cm
- Classes: Iris Setosa, Iris Versicolour, Iris Virginica

Iris Setosa

Iris Versicolour

Iris Virginica

Step 1: Calculate Covariance Matrix

- $A \in \mathbb{R}^{150 \times 4}$ (excluding class attribute)

- $\Sigma = \frac{1}{n} A^T A$ (assuming columns of A have zero mean)

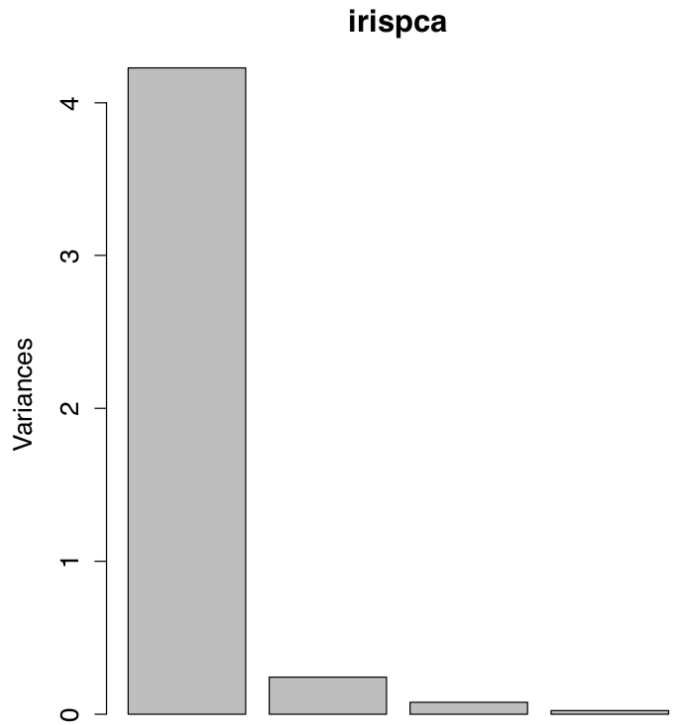
- $$\Sigma = \begin{matrix} \\ Sepal.Length \\ Sepal.Width \\ Petal.Length \\ Petal.Width \end{matrix} \begin{matrix} \begin{matrix} Sepal.Length & Sepal.Width & Petal.Length & Petal.Width \end{matrix} \\ \begin{pmatrix} 0.6857 & -0.0424 & 1.2743 & 0.5163 \\ -0.0424 & 0.1899 & -0.3297 & -0.1216 \\ 1.2743 & -0.3297 & 3.1163 & 1.2956 \\ 0.5163 & -0.1216 & 1.2956 & 0.5810 \end{pmatrix} \end{matrix}$$

Step 2: Calculate Eigenvalues and Eigenvectors

- $\text{Eigenvalues} = (4.2282, 0.2427, 0.0782, 0.0238)$
- $$\Sigma = \begin{array}{c} \\ Sepal.Length \\ Sepal.Width \\ Petal.Length \\ Petal.Width \end{array} \begin{array}{c} \begin{array}{cccc} PC1 & PC2 & PC3 & PC4 \end{array} \\ \begin{pmatrix} 0.3614 & -0.6566 & -0.5820 & 0.3155 \\ -0.0845 & -0.7302 & 0.5979 & -0.3197 \\ 0.85671 & 0.1734 & 0.0762 & -0.4798 \\ 0.3583 & 0.0755 & 0.5458 & 0.7537 \end{pmatrix} \end{array}$$
- The first principal component is the most important (largest eigenvalue), while others are not very significant.

Step 3: Scree plot

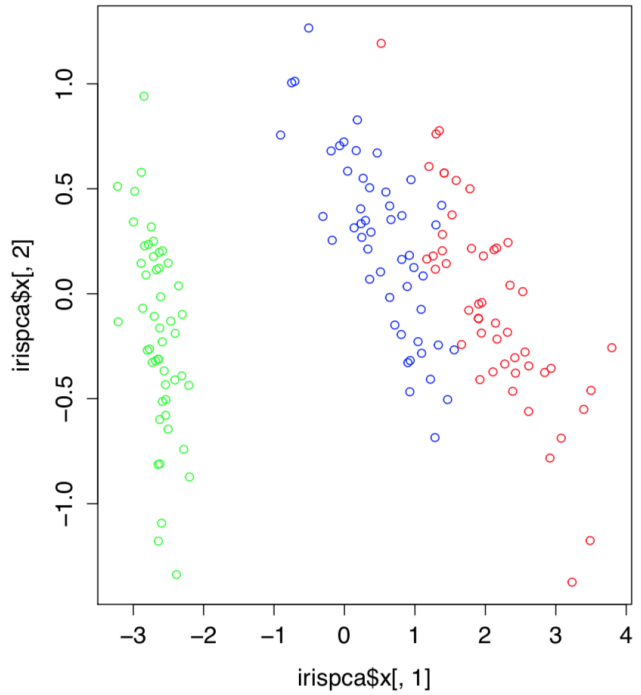
- Eigenvalues = (4.2282,0.2427,0.0782,0.0238)



- Confirm that using only one or two components is enough!

Step 4: Projection to Smaller Dimension

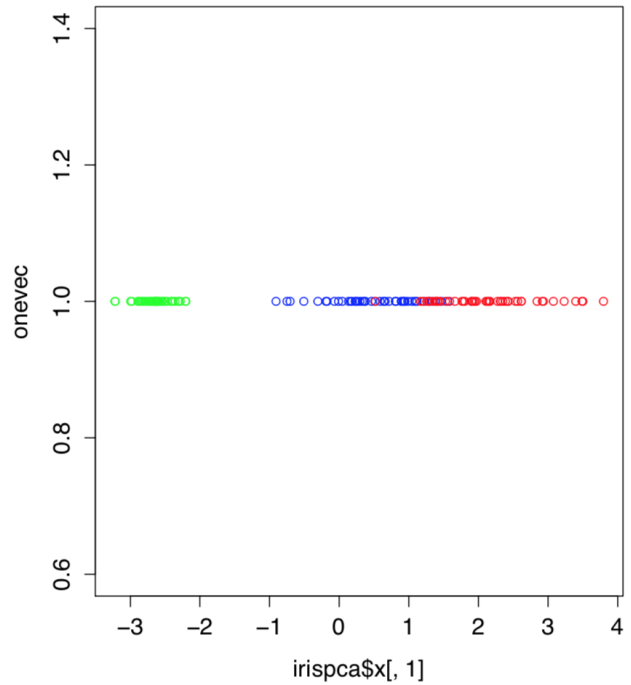
- With $p = 2$, projected data is obtained by $Y = AX \in \mathbb{R}^{150\times 2}$
- Visualize Y with class attribute (different class in different colors)



Green = setosa, blue = versicolour, red = virginica

Step 4: Projection to Smaller Dimension

- With $p = 1$, projected data is obtained by $Y = AX \in \mathbb{R}^{150\times 1}$
- Visualize Y with class attribute (different class in different colors)



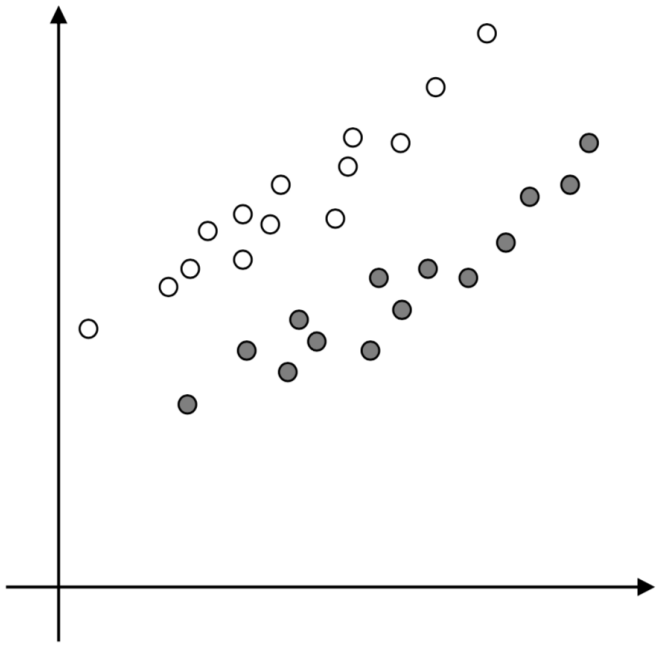
Green = setosa, blue = versicolour, red = virginica

PCA: Limitations

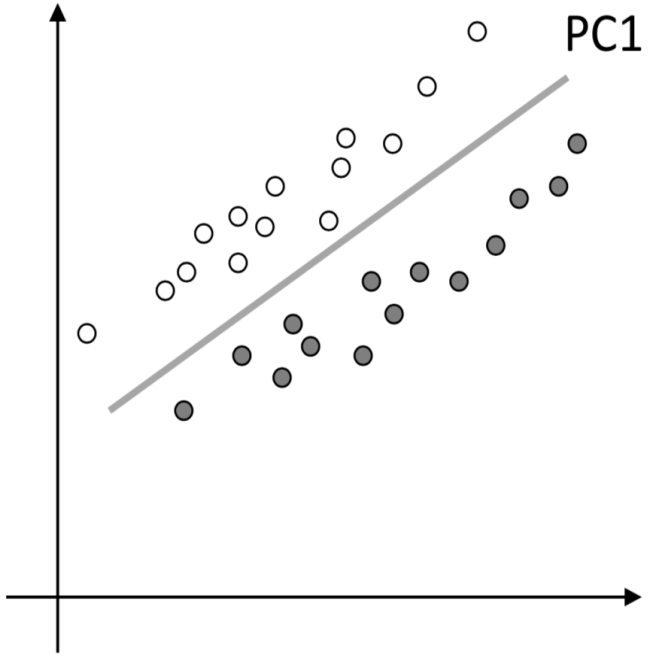
- PCA assumes a linear relationship between features i.e. it cannot capture non-linear structure in the data (as in many real-world applications).
- PCA assumes a correlation between features.
- PCA is sensitive to the scale of the features
- PCA is not robust against outliers
- Low interpretability of principal components.
- There is a trade-off between information loss and dimensionality reduction
- Technical implementations often assume no missing values

Issues: Data with Labels

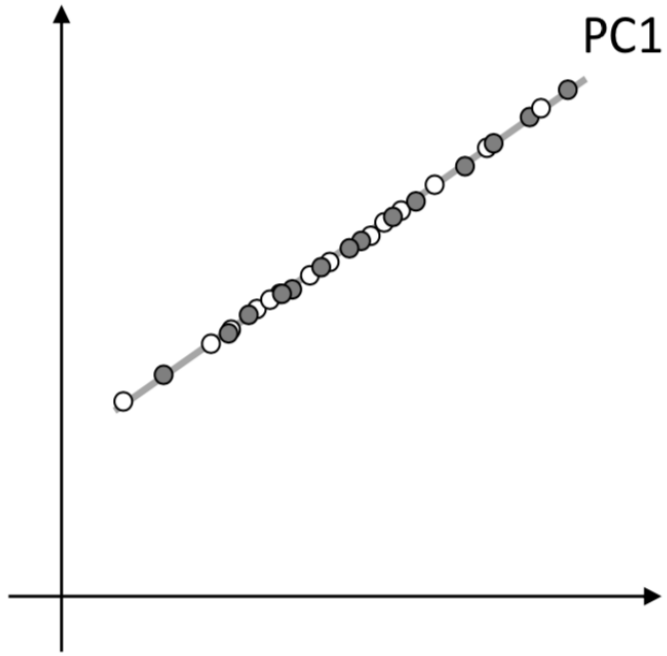
- A problematic example



Data with labels (white and gray)



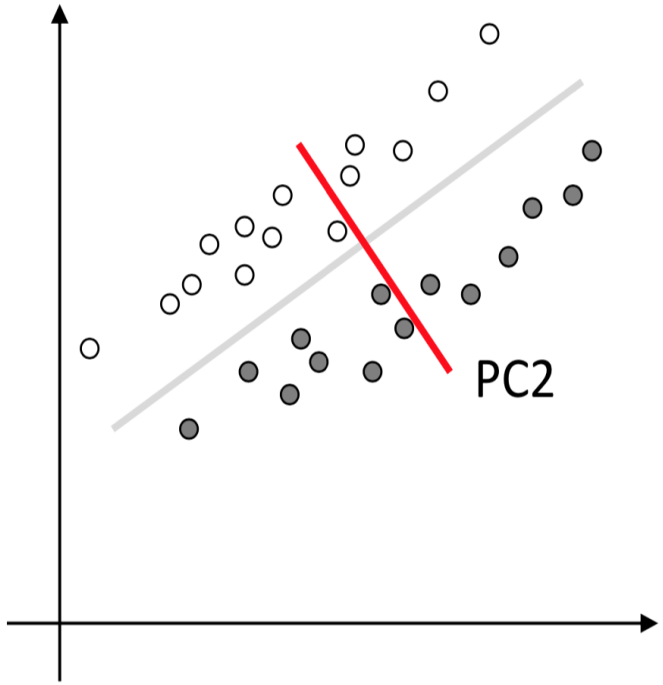
The first PC that explains most of the variance



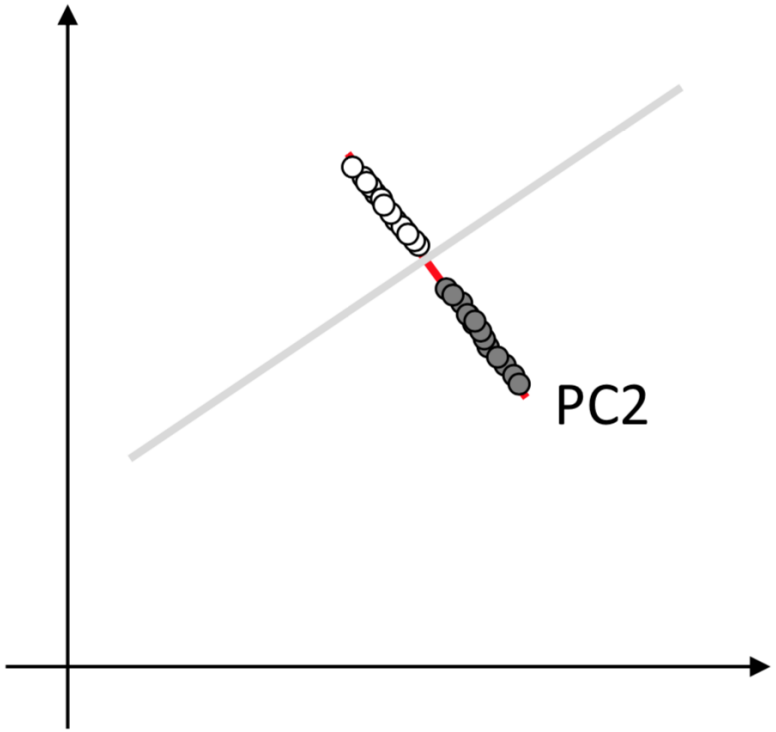
After projection, the result is useless

Issues: Data with Labels

- A problematic example



In fact, if we use the second PC, we obtain a better result



In fact, if we use the second PC, we obtain a better result

LINEAR DISCRIMINANT ANALYSIS (LDA)

Linear Discriminant Analysis (LDA)

- LDA is a supervised dimensionality reduction technique that also achieves classification of the data simultaneously.
- PCA focuses on capturing the direction of maximum variation in the data set.
- LDA focuses on finding a feature subspace that maximizes the separability between the groups.

LDA Solution Approach

- LDA projects the data points onto new axes such that these new components maximize the separability among categories while keeping the variation within each of the categories at a minimum value.
- As with PCA, LDA assumes that your data is centered around the origin and that your features are uncorrelated with one another
- The LD1 the first new axes created by LDA will account for capturing most variation between the groups or categories and then comes LD2 and so on.