Statistical Machine Learning

6주차

담당: 15기 염윤석

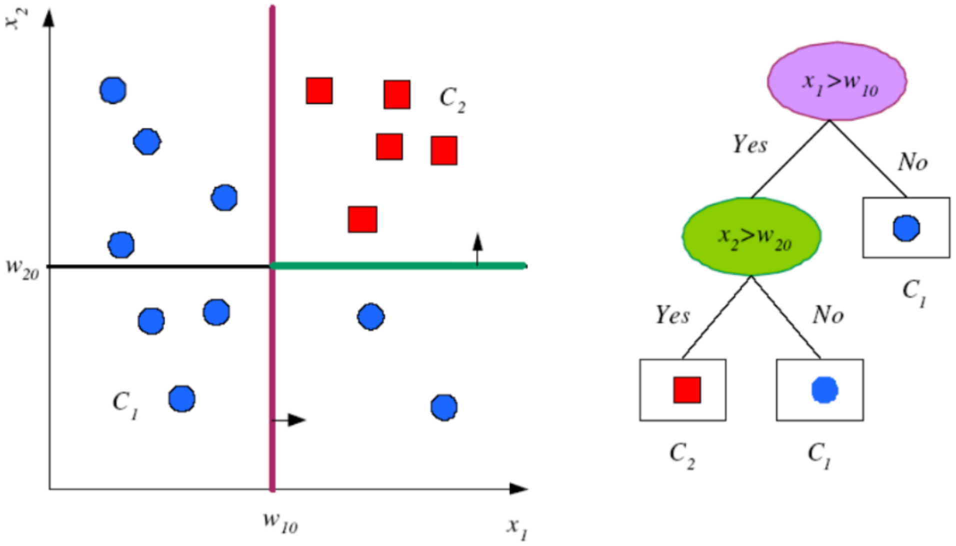
1. Decision Tree

2. Dimension Reduction

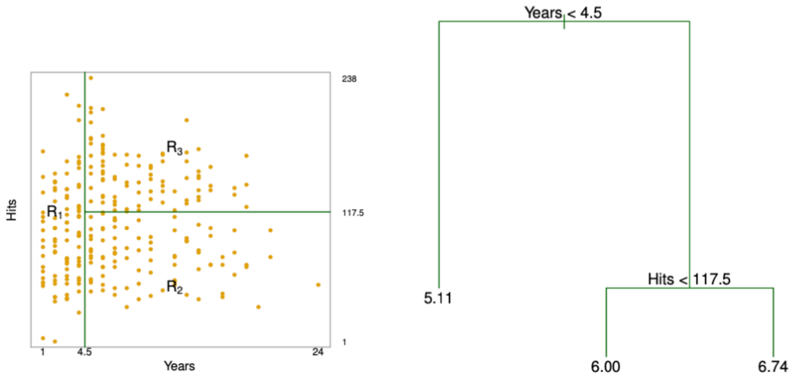
1. Decision Tree

Decision Tree

Classification



Regression



(a) Partitioned Predictor Space (b) Fitted Model

Issues

1. Determine how to split the training records (How to grow tree)

- How to specify the attribute(decision node)? → Depending on attribute types
- How to determine the best split? → Depending on number of ways to split
    - Performance measure

2. Determine when to stop splitting

- A stopping condition
- All records in a node = same class
- All records in a node = same attribute value

How to specify attributes

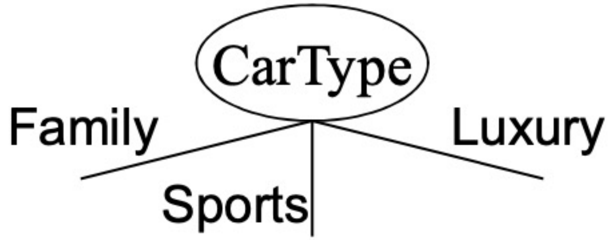
Depending on attribute types

- Categorical : Nominal, Ordinal
- Continuous
  - Discretization : to form an ordinal categorical attribute
  - Binary : finding best cut

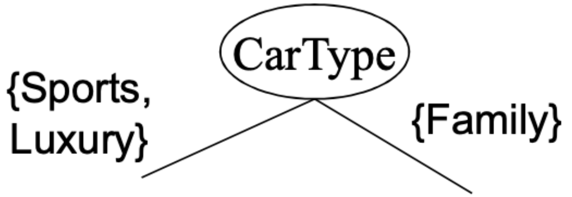
Depending on number of ways to split

- Binary split : $2^{k-1} - 1$
- Multi-way split

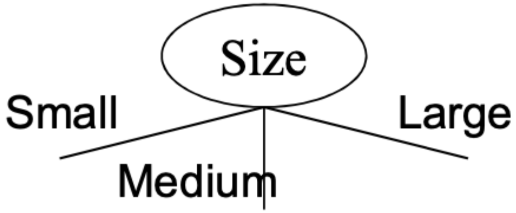
How to specify attributes



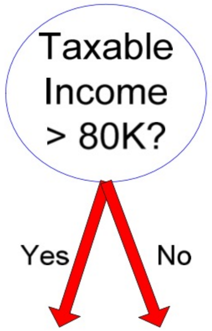
Nominal– Multiway



Nominal - Binary

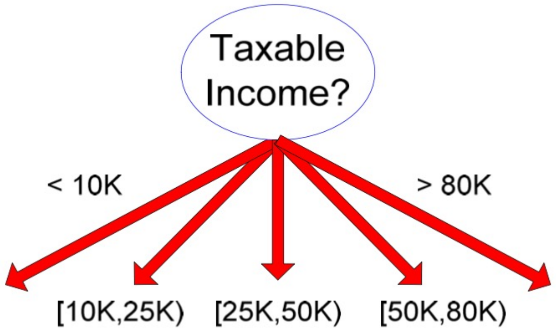


Ordinal - Multiway



(i) Binary split

(Binarization)



(ii) Multi-way split

(Discretization)

Node Impurity

- **Entropy:** $\text{Entropy}(t) = -\sum_j p(j\,|\,t)\cdot\log_2 p(j\,|\,t)$ Min = 0 | Max = logn
- **Gini Index:** $\text{Gini}(t) = 1 - \sum_j [p(j\,|\,t)]^2$ Min = 0 | Max = 1-1/n
- **Misclassification error:** $\text{Classification error}(t) = 1 - \max_j [p(j\,|\,t)]$ Min = 0 | Max = 1-1/n

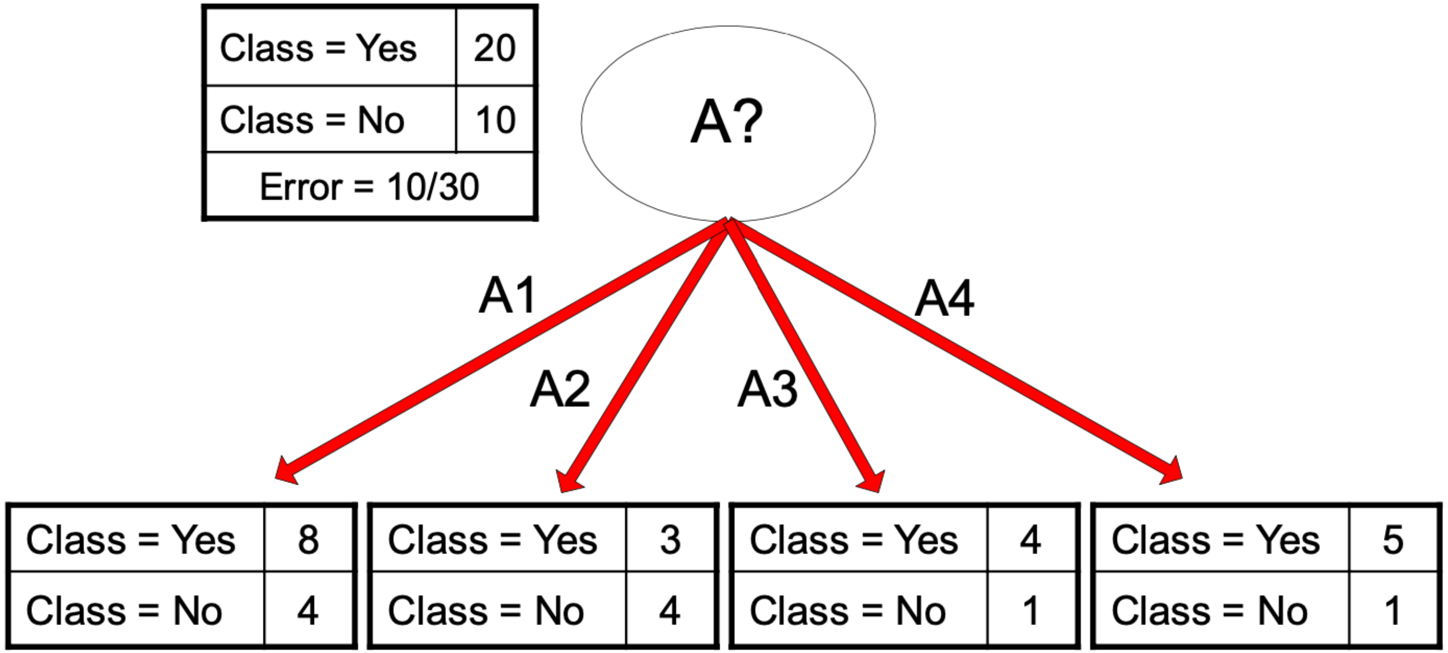
$$\boxed{Gain_{split} = Impurity - Impurity_{split}}$$

Weighted sum (average) of every node after split

$$\boxed{Gain\ ratio_{split} = \frac{Gain_{split}}{Split\ INFO} \qquad Split\ INFO = -\sum_i^k \frac{n_i}{n} log \frac{n_i}{n}}$$

Like impurity of partition
If partition size gets smaller, then Split INFO gets higher

Node impurity



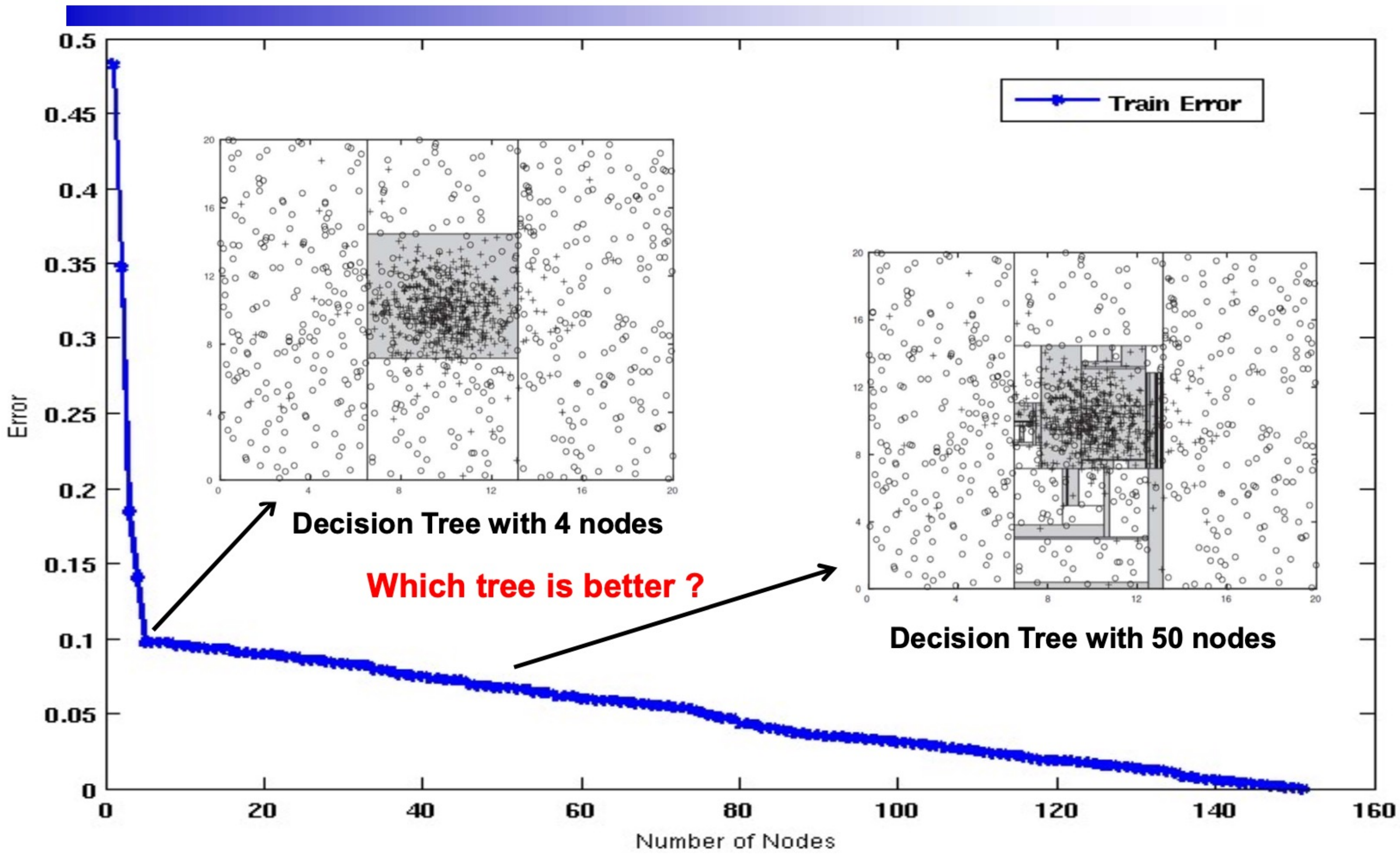
Entropy Gain(Information Gain) =

Gini Gain =

Misclassification error Gain

Gain Ratio =

Pruning



Train Error
Error
Number of Nodes
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
20
40
60
80
100
120
140
160
Decision Tree with 4 nodes
Which tree is better ?
Decision Tree with 50 nodes

Pruning

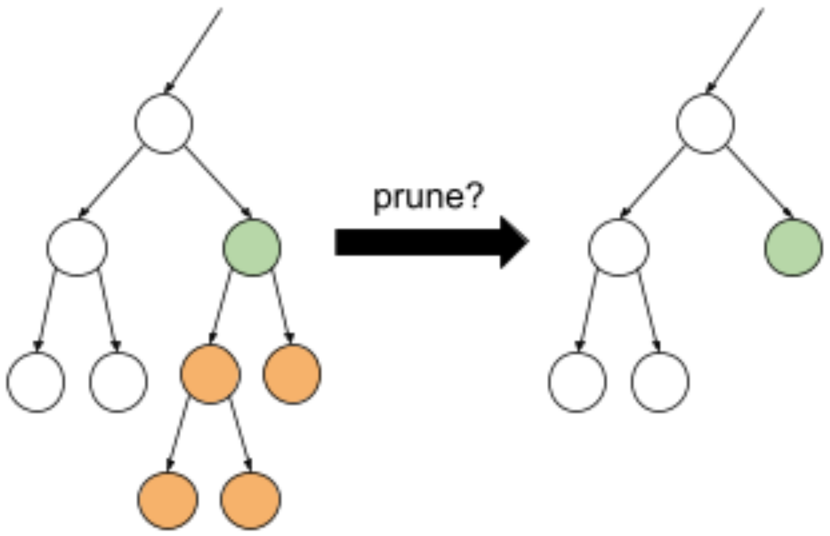
Size of tree is a tuning parameter

- max_depth : tree 최대 깊이
- max_nodes : 노드 최대 개수
- min_samples_split : 노드들이 가지고 있는 샘플 최소 수
- min_sample_leaf : leaf node에서 가지고 있는 샘플 최소 수

→ Too large Tree : overfitting (High Variance / Low Bias)
→ Too small Tree : Underfitting (Low Variance / High Bias)

Pruning methods

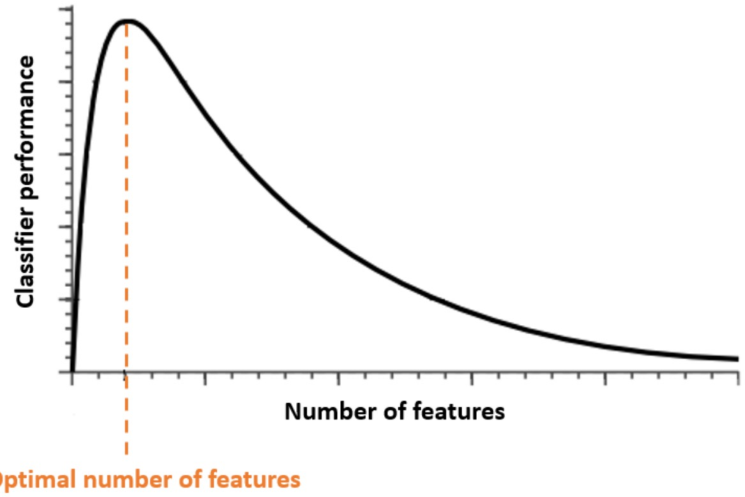
- Pre- pruning : Early Stopping
- Post- pruning : Grow the whole tree then prune subtress



2. Dimension Reduction

Why Reduce Dimensionality?

- Reduces time complexity : Less computation
- Reduces space complexity : Fewer parameters
- Saves the cost of observing the feature
- Simpler models are more robust on small datasets : more General model
- More interpretable : simpler explanation
- Data visualization (structure, groups, outliers, etc) if plotted in 2 or 3 dimensions



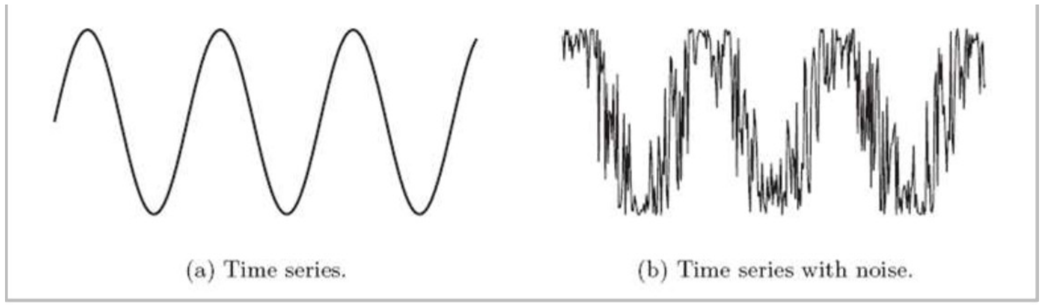


Figure 2.5. Noise in a time series context.

Curse of Dimensionality

When dimensionality increases

- data becomes increasingly sparse in the data space
- most training instances are likely to be far away from each other
- New instance will be likely be far away from training instance → overfitting

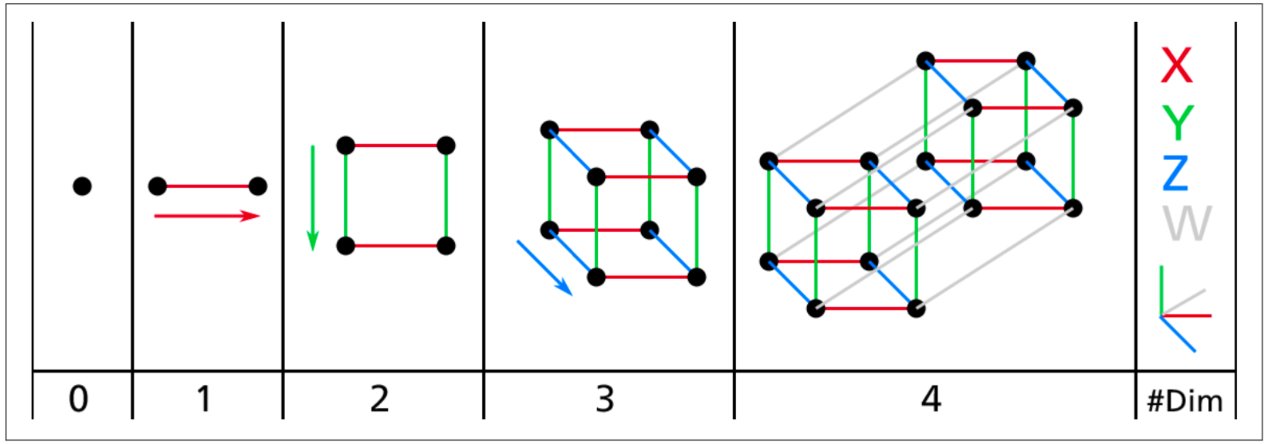
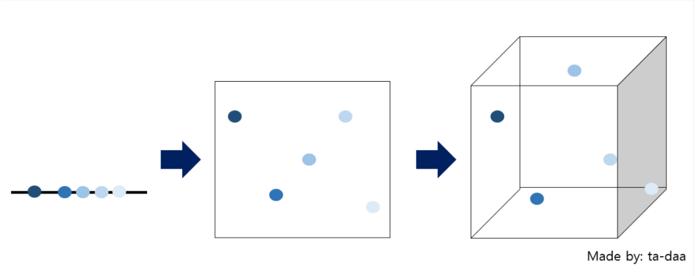


Figure 8-1. Point, segment, square, cube, and tesseract (0D to 4D hypercubes)[2]



Curse of Dimensionality

Solution

Increase Size N

- the number of training instances required to reach a given density grows exponentially with the number of dimensions

Feature Selection

- Choosing k<d important features, ignoring the remaining d – k
- Subset selection algorithms

Feature Extraction

- Project the original x_i , i =1,...,d dimensions to new $k<d$ dimensions, z_j , j =1,...,k
- Ex) PCA

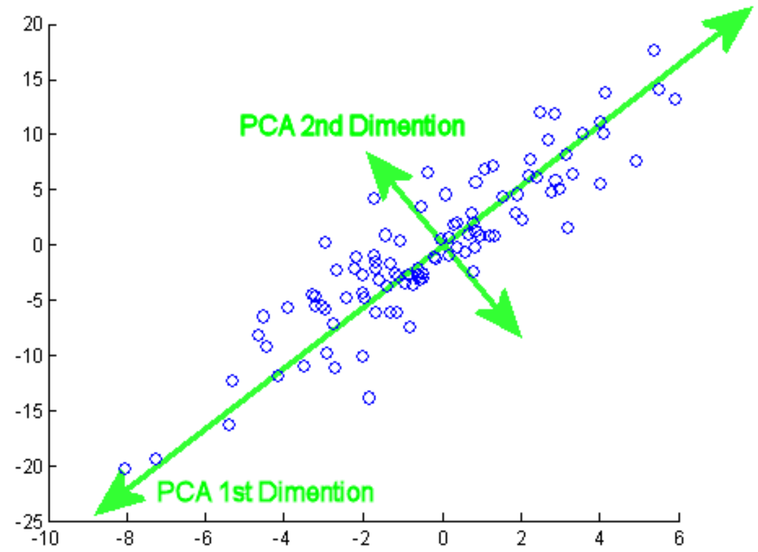
Algorithm 1. PCA

Feature Extraction

- Project the original x_i , i =1,...,d dimensions to new $k<d$ dimensions, z_j , j =1,...,k

PCA : Principal Component Analysis

- Find a low-dimensional space such that when x is projected there
- The projection of x on the direction of w is : z
- Find W such that Var(z) is maximized

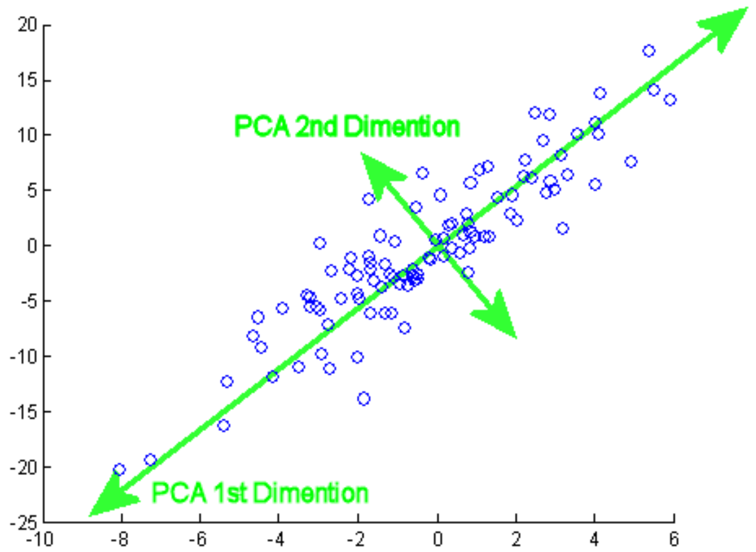


Var(z) =

PCA

Maximize Var(Z), subject to |W| = 1 → Lagrangian Multiplier method)

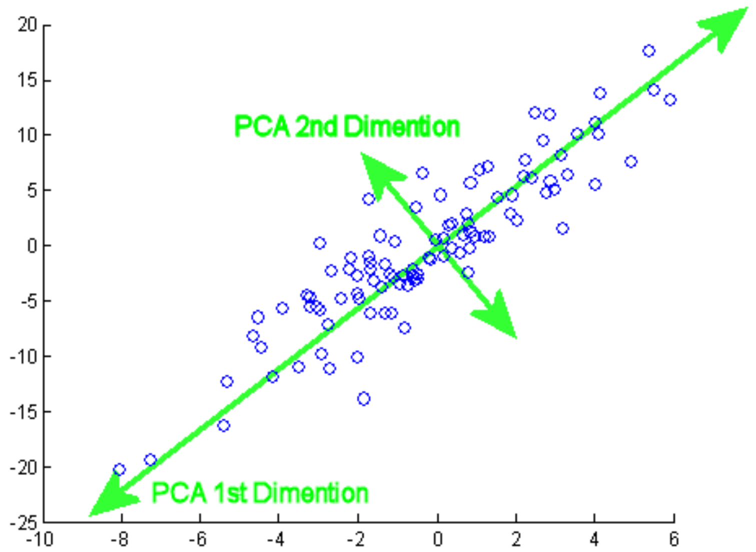
$$\max_{w} W^T \Sigma W \ - \ \alpha(W^T W \ - 1)$$



PCA

Maximize Var(Z), subject to |W| = 1 → Lagrangian Multiplier method)

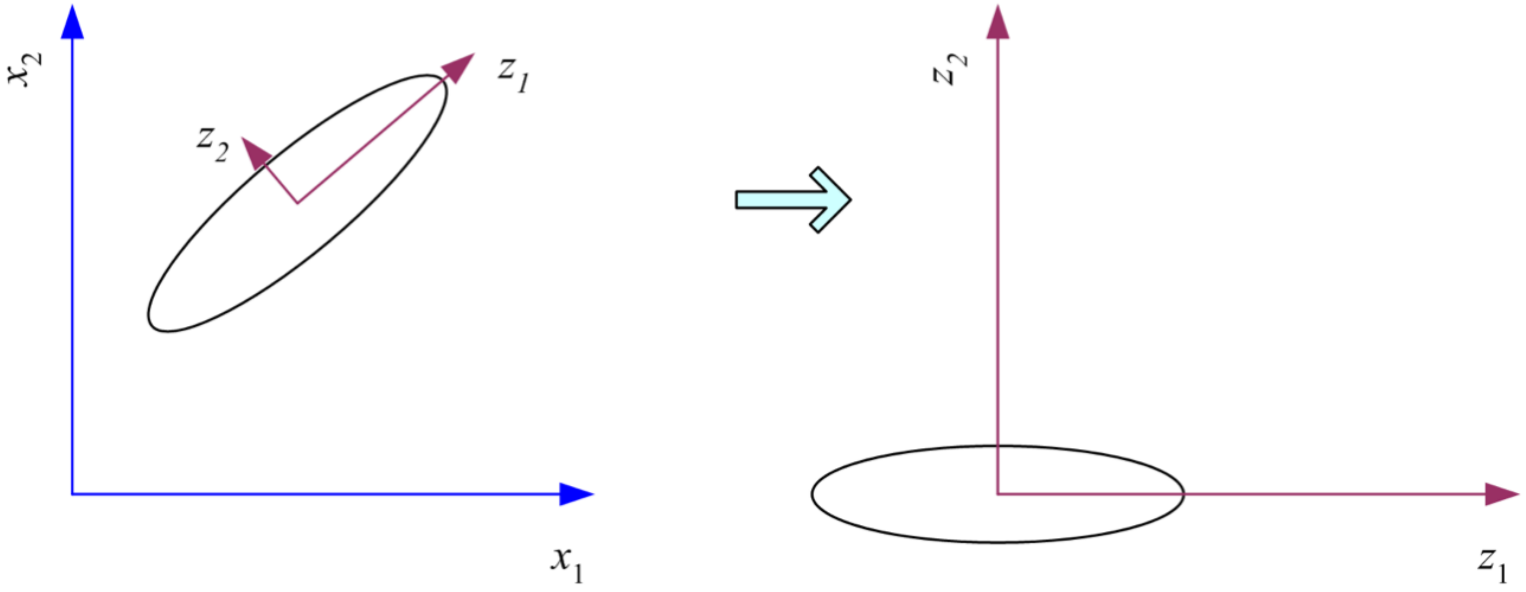
$$\max_{w} W^T \Sigma W - \alpha(W^T W - 1)$$



What PCA does

By centering, X-m → X'

Z = X'W
Where the columns of W are the eigenvector of Covariance matrix.
Centers the data at the origin and rotates the axes



How Many dimension K?

W : eigenvector of Σ (covariance matrix of X)
α : eigenvalue of Σ (covariance matrix of X)

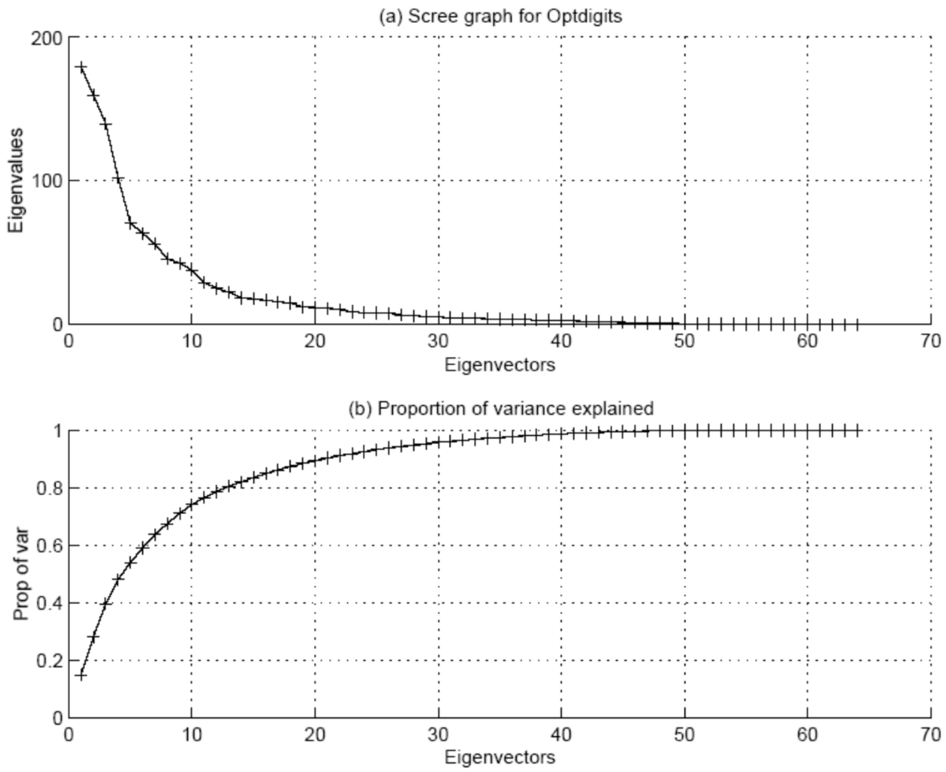
→ Σ : dxd matrix → we can find w & α for all d vectors

- Proportion of Variance (PoV)

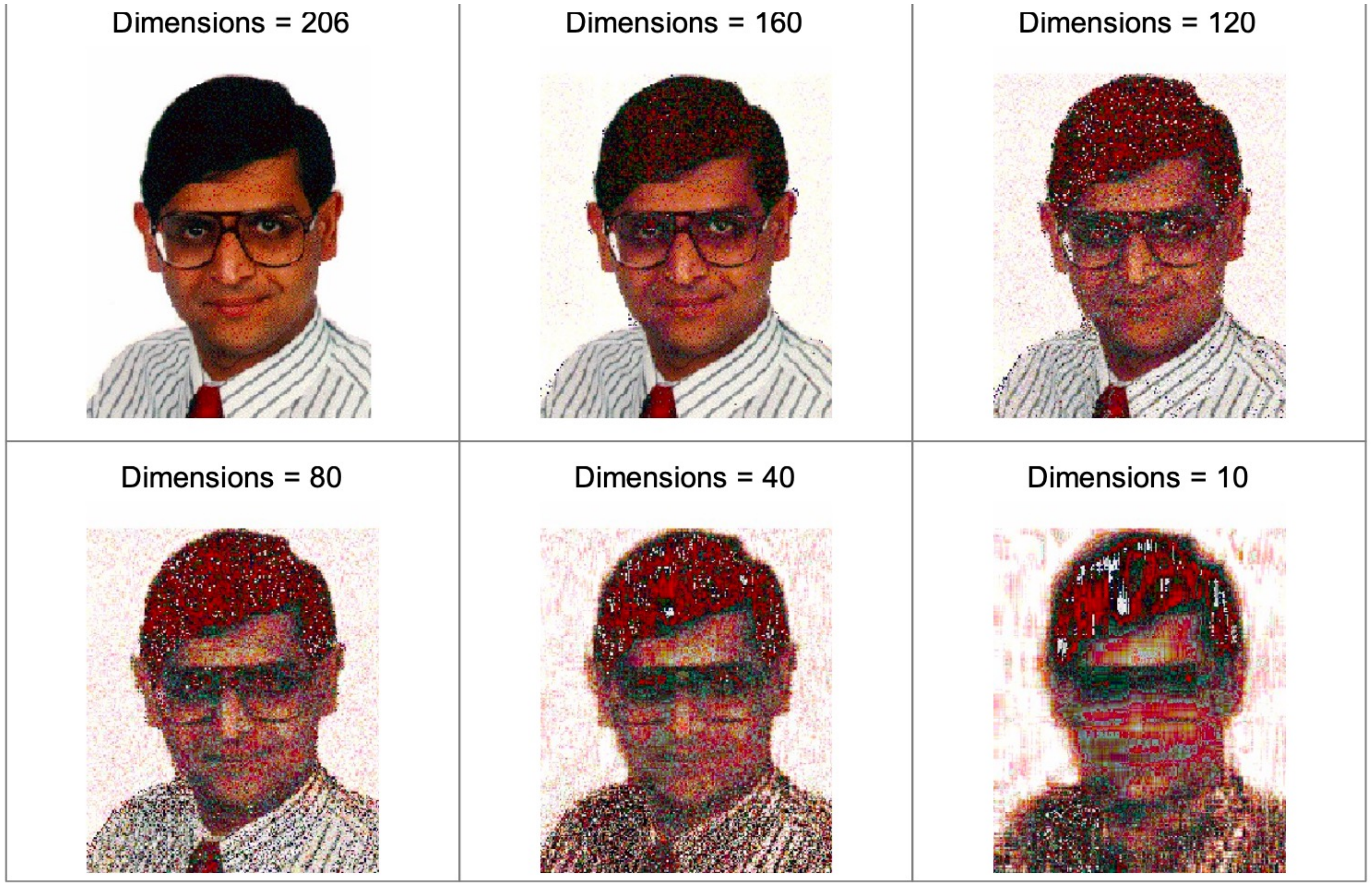
$$\frac{\alpha_1 + \alpha_2 + \cdots + \alpha_k}{\alpha_1 + \alpha_2 + \cdots + \alpha_k + \cdots + \alpha_d}$$

$$\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_k \cdots \geq \alpha_d$$

- Typically, stop at PoV >0.9
- Screen graph plots of PoV vs k, stop at ellbow



PCA


Dimensions = 206
Dimensions = 160
Dimensions = 120
Dimensions = 80
Dimensions = 40
Dimensions = 10

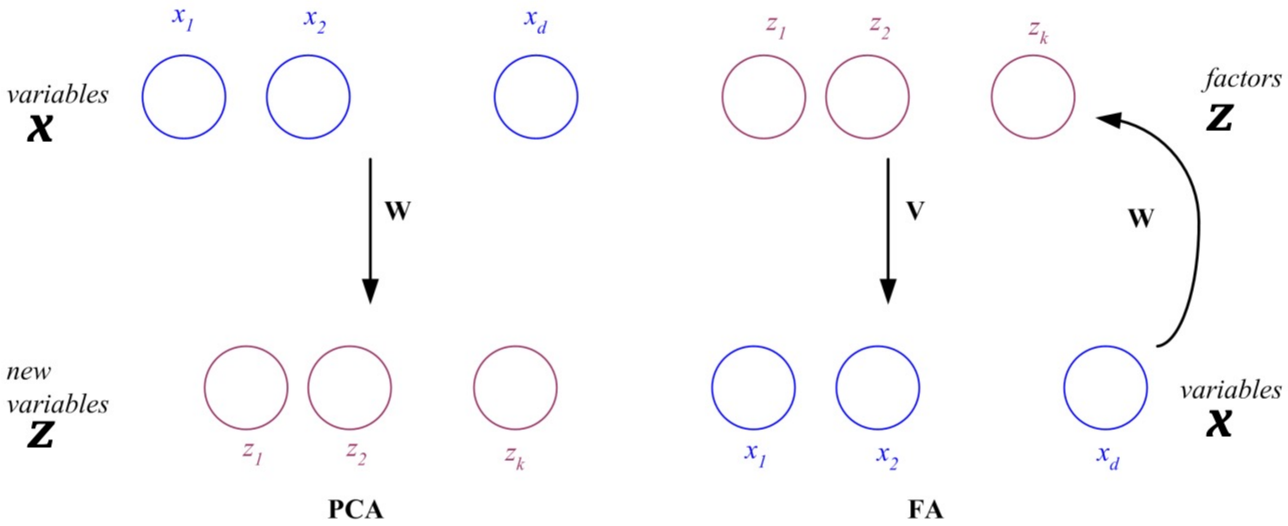
Algorithm 2. FA (Factor Analysis)

- Find a small number of factors z, which when combined generate x:

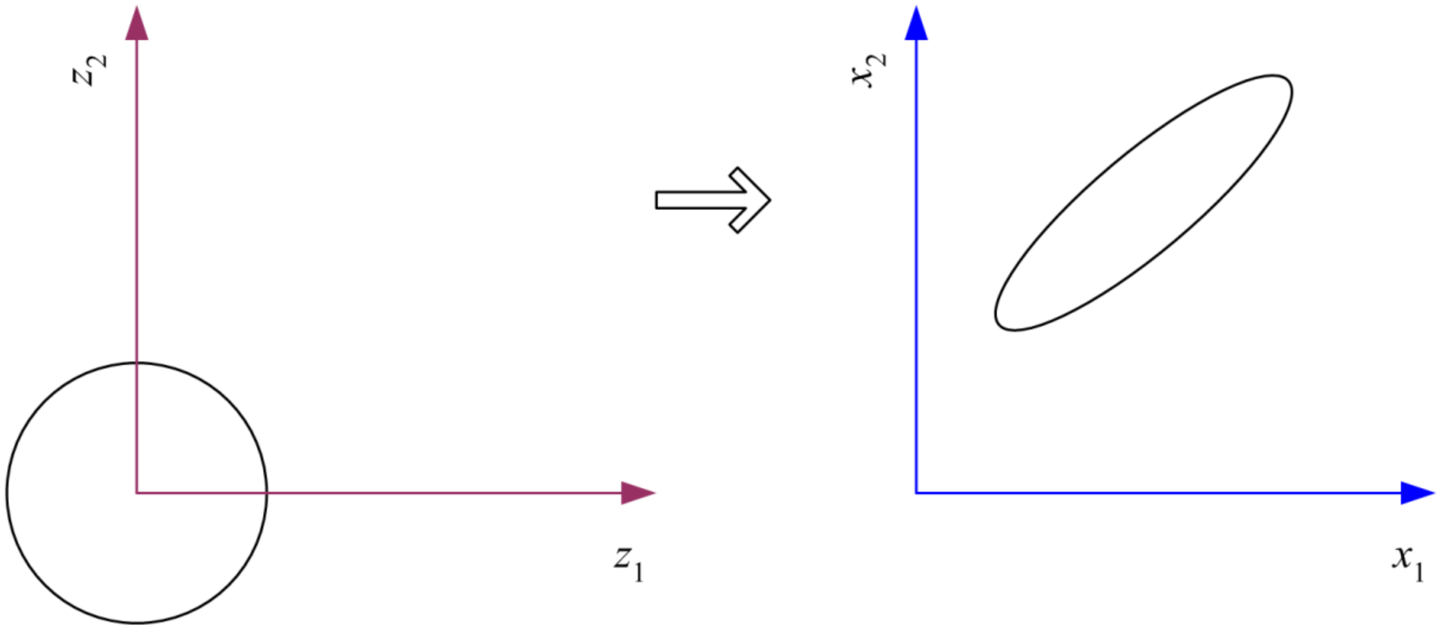
$$x \; - m = Vz + \varepsilon$$

- z : $latent\ vector$, ε: $noise\ sources$, V: $factor\ loadings$

$factor$ analysis $\sim$ $\boldsymbol{Matrix\ Factorization}$

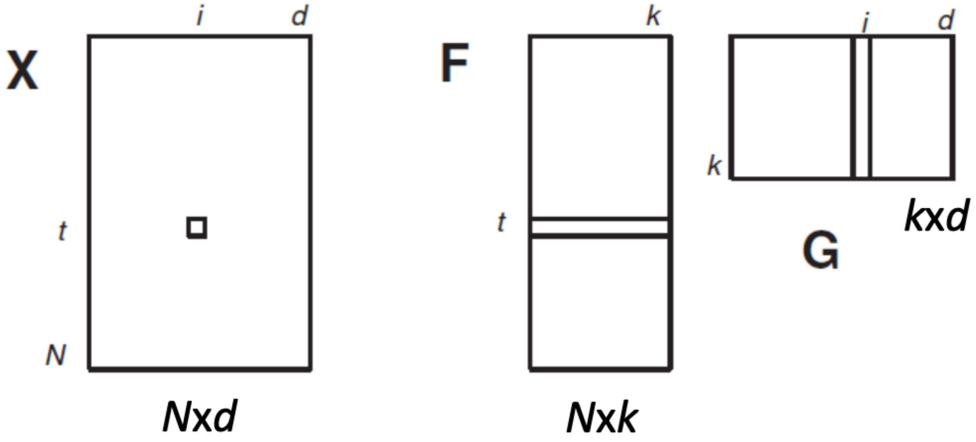


FA



Matrix Factorization

- Matrix factorization: ***X=FG***



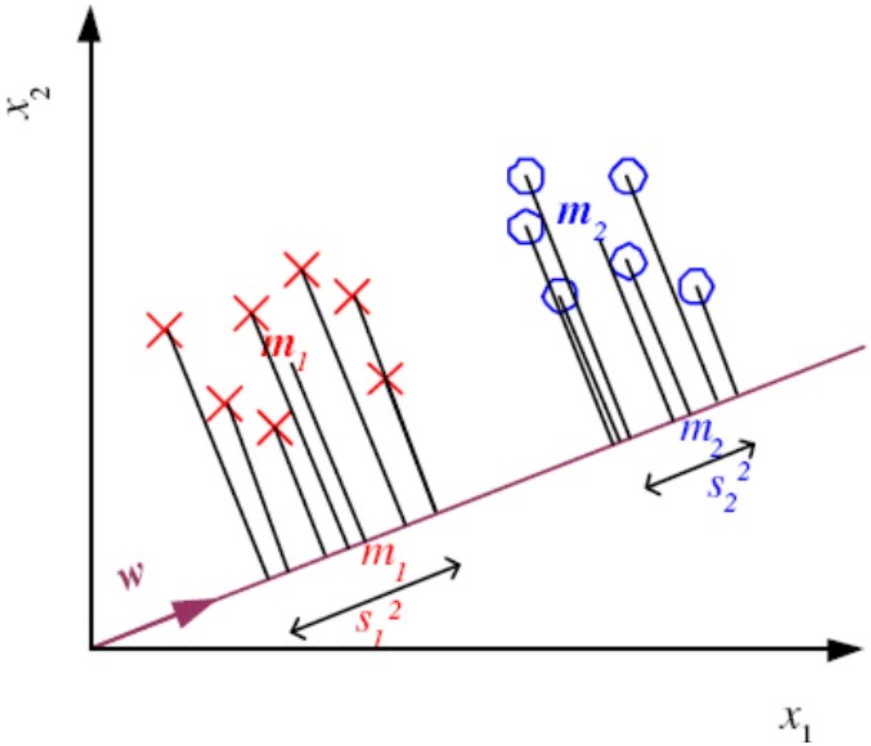
$$\mathbf{X}_{ti} = \mathbf{F}_t^T \mathbf{G}_i = \sum_{j=1}^{k} \mathbf{F}_{tj} \mathbf{G}_{ji}$$

Latent semantic indexing

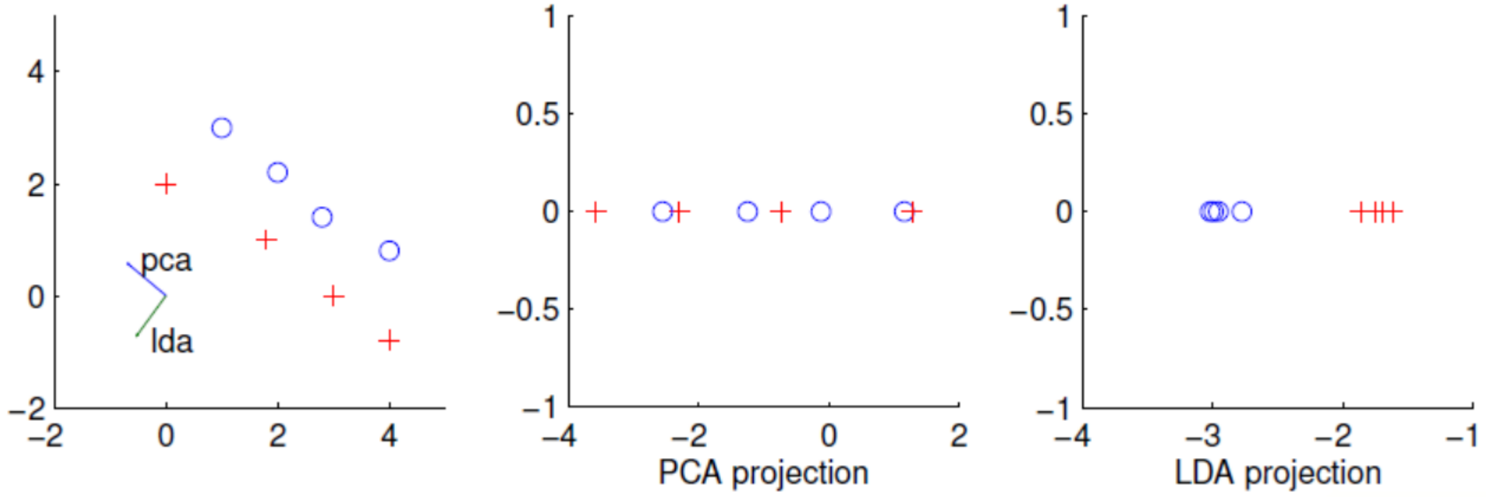
Algorithm 3. LDA (Linear Discriminant Analysis)

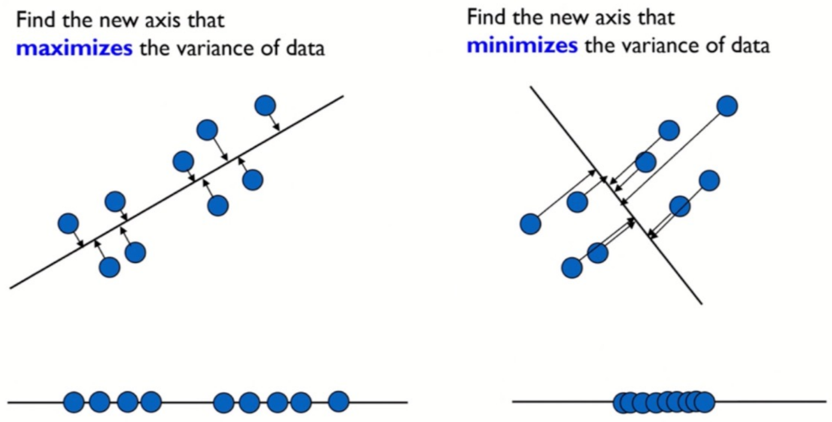
- Find a low-dimensional space such that when x is projected, classes are well-separated
- Find W that maximizes

$$J(w) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}$$



PCA vs LDA


4
2
0
−2
−2
0
2
4
pca
lda
1
0.5
0
−0.5
−1
−4
−2
0
2
PCA projection
1
0.5
0
−0.5
−1
−4
−3
−2
−1
LDA projection


Find the new axis that
maximizes the variance of data
Find the new axis that
minimizes the variance of data

Algorithm 4. t-SNE

Probability that x^r picks x^s as its neighbor in the original space : Gaussian kernel

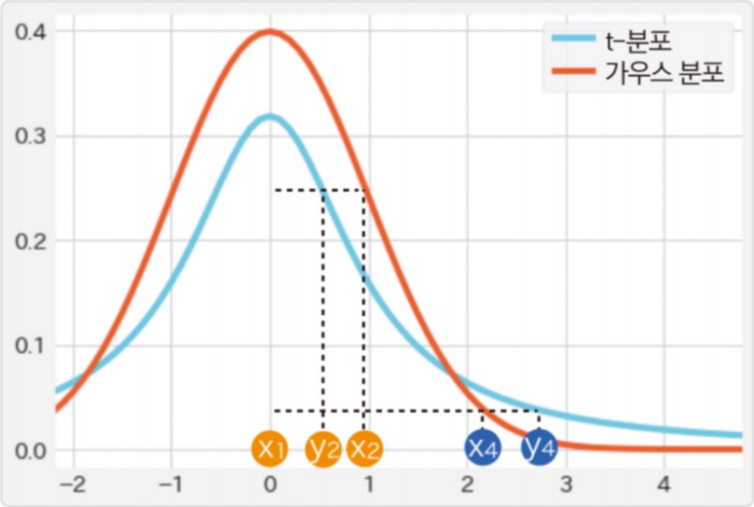
$$p_{s|r} = \frac{\exp[-\|\mathbf{x}^r - \mathbf{x}^s\|^2/2\sigma_r^2]}{\sum_{l\neq r}\exp[-\|\mathbf{x}^r - \mathbf{x}^l\|^2/2\sigma_r^2]}$$

Probability that z^r picks z^s as its neighbor in the new space: $t-$distribution

$$q_{rs} = \frac{(1+\|\mathbf{z}^r - \mathbf{z}^s\|^2)^{-1}}{\sum_l \sum_{m\neq l}(1+\|\mathbf{z}^l - \mathbf{z}^m\|^2)^{-1}}$$

Find z such that these are as similar as possible in terms of KL-distance

$$KL(P||Q) = \sum_r \sum_s p_{rs} \log \frac{p_{rs}}{q_{rs}}$$



Other Algorithms

- Canonical Correlation Analysis (CCA)
- Isometric feature mapping(Isomap)
- multi-dimensional scaling (MDS)
- Locally linear Embedding(LLE)

수고하셨습니다!

해당 세션자료는 KUBIG Github에서 보실 수 있습니다!

이제부터 머신러닝 분반 프로젝트 발표를 진행하겠습니다.