Introduction to Data Mining Lecture

Week 13: scikit-learn Example 2

Joon Young Kim

Assistant Professor, School of AI Convergence
Sungshin Women's University

Another Regression

- 선형 회귀와 마찬가지로 로지스틱 회귀 경우도 predictor 들에 연계된 특정 모델에 의존하고 있다.

  → 사용자는 포함시킬 predictors 및 형태에 대해서 상세화 시켜야 한다. (e.g., including any interaction terms)

- 로지스틱 회귀 자체가 추론 목적으로 인한 통계 분석에 많이 쓰이는 바 주요 컨셉들에 대한 추가 설명도 진행할 예정이다.

  → coefficient interpretation, goodness-of-fit evaluation, inference, and multiclass models.

Another Regression

- Logistic regression

  → 선형회귀와 기본적 궤는 같이 하나 의존변수가 카테고리 인점이 차이점

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_p X_p + \epsilon$$

0 or 1

- Logistic regression 의 경우

  → predictor 변수를 활용한 분류 (classification).

  → 각 클래스내 각각의 observations들의 유사점 도출 (profiling).

- 설명을 위해서 이중 의존 변수를 중심으로 다룰 예정

Another Regression

- 1단계: 각 클래스에 속할 확률을 추정 (probabilities estimates)

  → Y=0/1인 이중 케이스에서 P(Y = 1)을 추정시 반대 케이스도 자동적으로 구할수 있음.

- 2단계: 각 케이스별 클래스 분류를 위해서 1단계에서 구한 확률 기반하에 cutoff value를 계산

Logistic Regression Model

- Logistic regression model 자체는 다양한 분야에서 활용 가능
  → 카데고리 결과를 내거나 설명해야 되는 어떤 모델이든 가능
  → 경제 선택 행동 (choice behavior in econometrics)

ID	Age	Professional Experience	Income	Family Size	CC Avg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online Banking	Credit Card
1	25	1	49	4	1.60	UG	0	No	Yes	No	No	No
2	45	19	34	3	1.50	UG	0	No	Yes	No	No	No
3	39	15	11	1	1.00	UG	0	No	No	No	No	No
4	35	9	100	1	2.70	Grad	0	No	No	No	No	No
5	35	8	45	4	1.00	Grad	0	No	No	No	No	Yes
6	37	13	29	4	0.40	Grad	155	No	No	No	Yes	No
7	53	27	72	2	1.50	Grad	0	No	No	No	Yes	No
8	50	24	22	1	0.30	Prof	0	No	No	No	No	Yes
9	35	10	81	3	0.60	Grad	104	No	No	No	Yes	No
10	34	9	180	1	8.90	Prof	0	Yes	No	No	No	No
11	65	39	105	4	2.40	Prof	0	No	No	No	No	No
12	29	5	45	3	0.10	Grad	0	No	No	No	Yes	No
13	48	23	114	2	3.80	Prof	0	No	Yes	No	No	No
14	59	32	40	4	2.50	Grad	0	No	No	No	Yes	No
15	67	41	112	1	2.00	UG	0	No	Yes	No	No	No
16	60	30	22	1	1.50	Prof	0	No	No	No	Yes	Yes
17	38	14	130	4	4.70	Prof	134	Yes	No	No	No	No
18	42	18	81	4	2.40	UG	0	No	No	No	No	No
19	46	21	193	2	8.10	Prof	0	Yes	No	No	No	No
20	55	28	21	1	0.50	Grad	0	No	Yes	No	No	Yes

- Acceptance
- Rejection

Logistic Regression Model

■ 로지스틱 주요 요소

→ 결과값 Y를 의존 변수로 두는 대신 함수로 변환시킨다. (logit).

→ Logit 경우 Predictors들의 선형함수로 모델링할 수 있음

→ Logit 가 예측되고 나서는 확률로 다시 변환 가능하다.

$$\text{logit} = \beta_0 + \beta_1 x_1 + \beta_1 x_1 + \cdots + \beta_q x_q$$

Logistic Regression Model

- 먼저 클래스 1에 속할 확률인 p 확인 필요

  → Y와는 달리 p의 경우 [0, 1] 사이에 있어야 한다. 그러나 p를 q개의 예측치 기반의 선형 함수로 구성 시 아래와 같다.

  → p의 범위를 벗어날 가능성이 존재한다.

$$p = \beta_0 + \beta_1 x_1 + \beta_1 x_1 + \cdots + \beta_q x_q$$

Logistic Regression Model

- 오른쪽 수식 부분이 항상 [0, 1] 사이에 있을 가능성이 음

  → 수정 반영을 위해서 predictors 기반의 공식 구성시 비선형 함수로 구성하는 것이 필요하다. (nonlinear function)

  → 로지스틱 반응 함수 (logistic response function)의 경우 predictors $[x_1, x_2, \cdots, x_q]$ 구성시 오른쪽 공식은 항상 $[0, 1]$ 사이에 존재한다.

$$p = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_1 x_1+\cdots+\beta_q x_q)}}$$

Logistic Regression Model

■ 특정 클래스에 속하는 경우에 대한 측정 필요

→ 이를 위해 odds라는 개념을 도입한다.

■ 클래스 1에 속할 odds (Y = 1)

→ 클래스 1에 속할 확률과 클래스 0에 속할 확률의 비율

$$\text{odds} = \frac{p}{1-p}$$

Logistic Regression Model

- Odds와 확률의 차이점

  → 승리 확률이 0.5일때 the odds of winning은 0.5/0.5 = 1.

  → 주어진 이벤트의 Odds를 기반으로 확률 공식에 대한 수정이 가능하다.

$$\text{odds} = \frac{p}{1-p} \quad \Rightarrow \quad p = \frac{\text{odds}}{1+\text{odds}}$$

Logistic Regression Model

- 로지스틱 함수 안에 odds를 교체함으로써 odd와 predictors간의 관계를 아래 공식으로 정리 할 수 있다.

$$\text{odds} = e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_q x_q}$$

- 해당 공식 경우 odd와 predictors의 관계를 곱셈으로써 정리할 수 있다. 해당 관계는 percentage로도 설명 가능하다.

Logistic Regression Model

- 주어진 관계도에 따라 로지스틱 모델의 표준 공식은 다음과 같다.

$$\ln(\text{odds}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_q x_q$$

- ln(odds) 을 logit으로 통칭하며 범위는 [-∞, ∞]이다.

  → Logit은 의존 변수로 치환되며 q개의 predictors들의 선형 함수로 모델링이 가능하다.

Logistic Regression Model

$$p = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_q x_q)}}$$

$$\Downarrow$$

$$p = \frac{\text{odds}}{1+\text{odds}}$$

$$\Downarrow$$

$$\ln(\text{odds}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_q x_q$$

Logistic Regression Model

- **Dataset:**

  → Assumption: 대출 가능 유무 판단을 위해서 5000명의 고객 데이터 사용

  → 이전 기록에 따르면 해당 5000명 데이터중 480명만이 대출 가능 유 판정

ID	Age	Professional Experience	Income	Family Size	CC Avg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online Banking	Credit Card
1	25	1	49	4	1.60	UG	0	No	Yes	No	No	No
2	45	19	34	3	1.50	UG	0	No	Yes	No	No	No
3	39	15	11	1	1.00	UG	0	No	No	No	No	No
4	35	9	100	1	2.70	Grad	0	No	No	No	No	No
5	35	8	45	4	1.00	Grad	0	No	No	No	No	Yes
6	37	13	29	4	0.40	Grad	155	No	No	No	Yes	No
7	53	27	72	2	1.50	Grad	0	No	No	No	Yes	No
8	50	24	22	1	0.30	Prof	0	No	No	No	No	Yes
9	35	10	81	3	0.60	Grad	104	No	No	No	Yes	No
10	34	9	180	1	8.90	Prof	0	Yes	No	No	No	No
11	65	39	105	4	2.40	Prof	0	No	No	No	No	No
12	29	5	45	3	0.10	Grad	0	No	No	No	Yes	No
13	48	23	114	2	3.80	Prof	0	No	Yes	No	No	No
14	59	32	40	4	2.50	Grad	0	No	No	No	Yes	No
15	67	41	112	1	2.00	UG	0	No	Yes	No	No	No
16	60	30	22	1	1.50	Prof	0	No	No	No	Yes	Yes
17	38	14	130	4	4.70	Prof	134	Yes	No	No	No	No
18	42	18	81	4	2.40	UG	0	No	No	No	No	No
19	46	21	193	2	8.10	Prof	0	Yes	No	No	No	No
20	55	28	21	1	0.50	Grad	0	No	Yes	No	No	Yes

Logistic Regression Model

- **Data Preprocessing:**

  → 1. training and validation 데이터셋 파티셔닝을 60%대 40% 비율로 진행

  → 2. Dummy(Categorical) variables 셋업

  예) 대출 신청 중요 요소: 주담대유무, 대학 졸업, 증권계좌, 온라인 뱅킹, 신용카드

$$\text{Mortgage} = \begin{cases} 1 & \text{If Mortgage,} \\ 0 & \text{Otherwise.} \end{cases} \qquad \text{Securities Account} = \begin{cases} 1 & \text{If Securities Account,} \\ 0 & \text{Otherwise.} \end{cases}$$

$$\vdots$$

$$\text{Online} = \begin{cases} 1 & \text{If Online Banking,} \\ 0 & \text{Otherwise.} \end{cases} \qquad \text{CreditCard} = \begin{cases} 1 & \text{If Credit Card,} \\ 0 & \text{Otherwise.} \end{cases}$$

Categorical Predictors

Logistic Regression Model

- **Data Preprocessing:**

  → Single Predictor Case (Income)

$$P(\text{Loan} = \text{Yes}|\text{Income} = \text{x}) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

$$\text{odds}(\text{Loan} = \text{Yes}) = e^{(\beta_0+\beta_1 x)}$$

$$\beta_0 = -6.3523, \beta_1 = 0.0392$$ ← 트레이닝 통해 도출된 값

$$P(\text{Loan} = \text{Yes}|\text{Income} = \text{x}) = \frac{1}{1+e^{6.3523-0.0392x}}$$

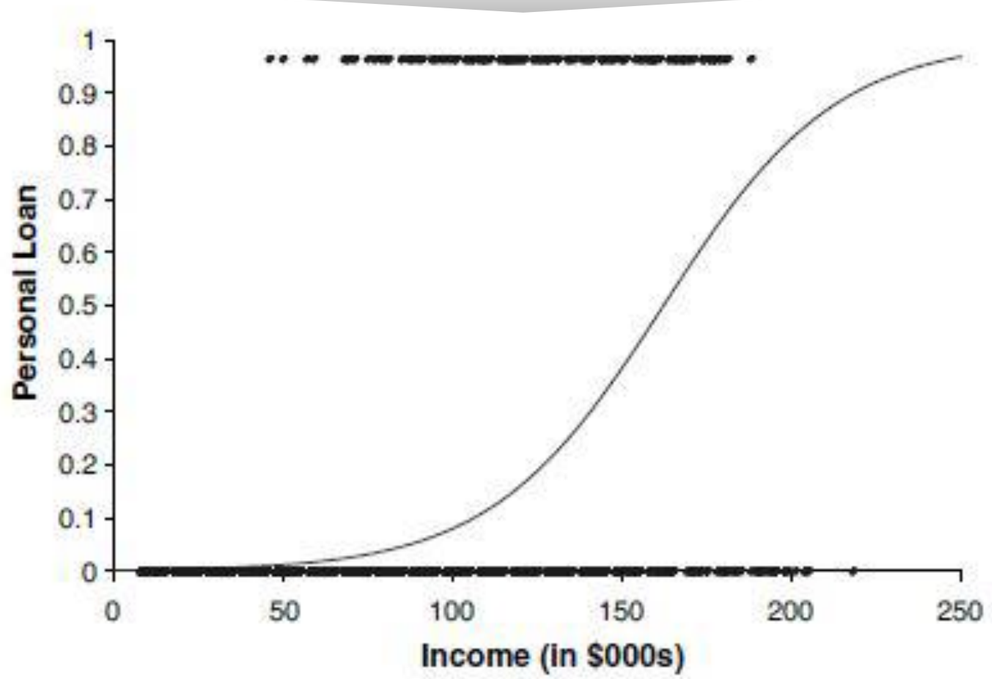
Logistic Regression Model

- **Data Preprocessing:**

  → Single Predictor Case (Income)

  → 데이터가 정확하게 0과 1로 구분되어지지 않음.

$$P(\text{Loan} = \text{Yes}|\text{Income} = \text{x}) = \frac{1}{1+e^{6.3523-0.0392x}}$$

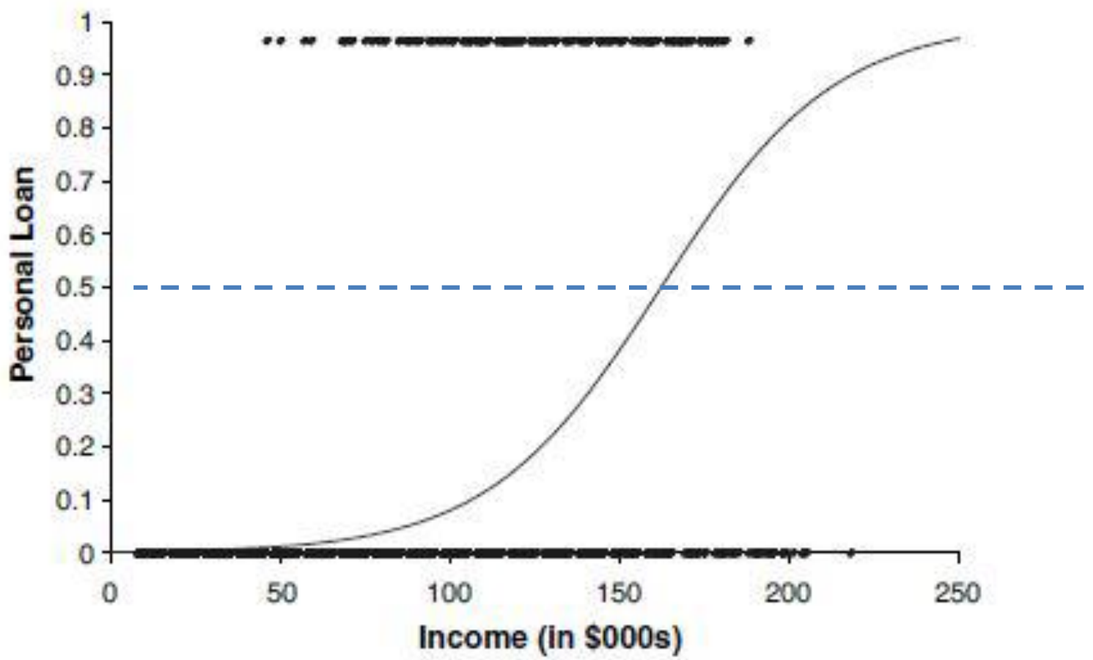


Logistic Regression Model

- **Cutoff Value**

  → 각 샘플별로 두개 클래스중 하나로 분류되기 위해서 확률 대상으로 cutoff이 세팅되어야 한다. 해당 값을 c로 정의한다.

  → 샘플별로 확률이 c보다 높을 경우 클래스 1로 구분한다. 반대로 아래일 경우 클래스 0으로 분류한다.



Logistic Regression Model

- “최적“ cutoff 확률(값)을 결정하기 위한 다양한 접근법이 존재한다.
  - → Two-class 상에서 가장 단순하고 알려져 있는 cutoff value은 0.5이다.
  - → Training set상에서 테스트를 통해서 가장 정확률이 높은 cutoff value을 선정한다. 다만 해당 값의 경우 overfitting의 문제가 존재한다.
  - → Sensitivity를 최대화하고 specificity를 최소화하는 값 (false positive를 최소화)
  - → 비용 기반 접근의 경우 오분류 비용을 최소화하는 cutoff 값을 찾는 경우다. 이때 오분류 비용에 대해서 상세화가 이루어져야 되며 각 클래스별 확률값이 존재해야 한다.

- 로지스틱 회귀의 경우 Y 와 β 값들의 관계는 비선형이다.
  - → 따라서 해당 β parameters의 경우 least squares로 구하는게 불가능하다
  - → Maximum Likelihood Estimate(MLE)을 활용하여서 데이터의 분포에 기반한 개선을 기대할수 있다. 해당 estimate의 경우 컴퓨터로 계산이 필요하다.

Python Example

■ Python Case

→ Logistic Regresion 함수 활용한 예측

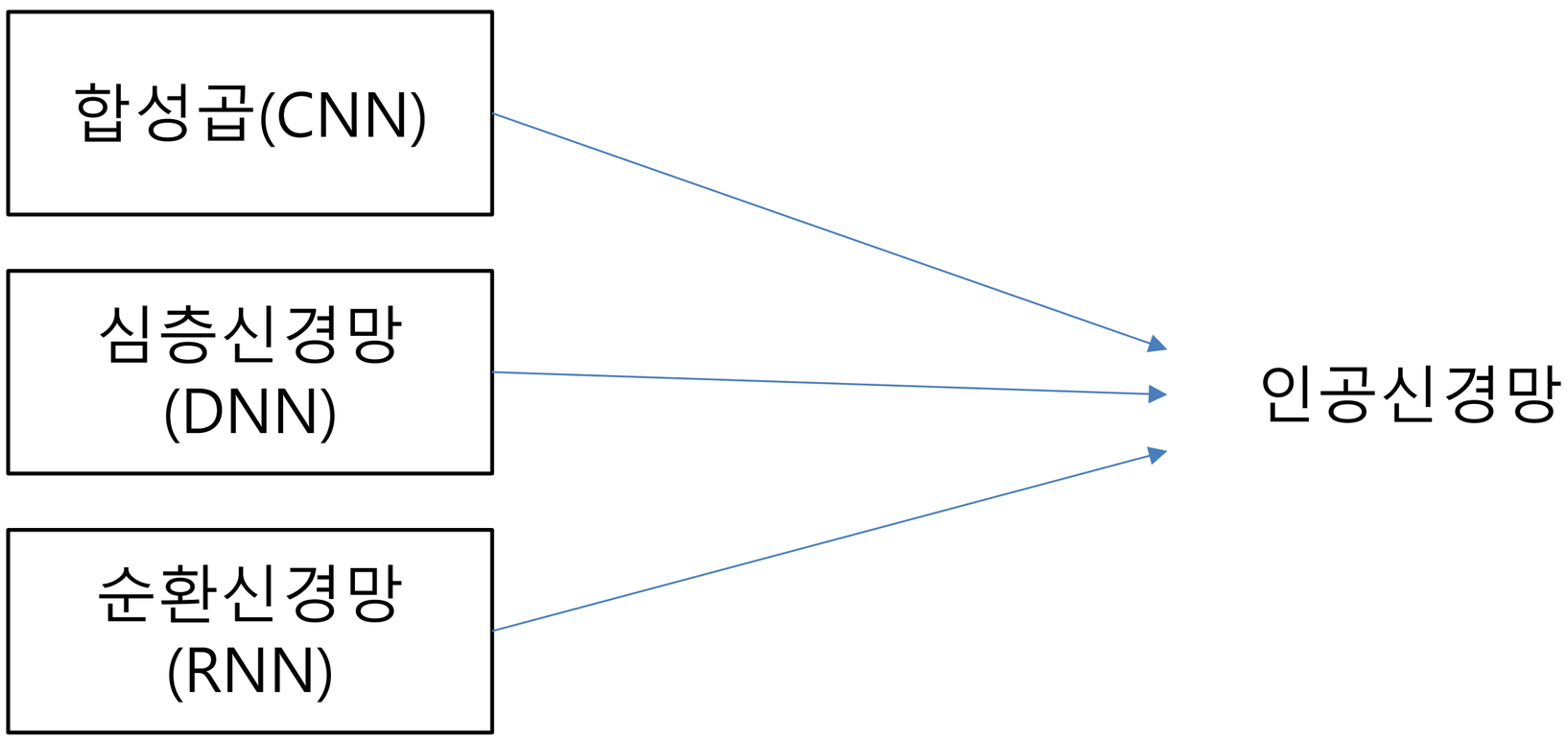
```
from sklearn.datasets import load_iris
from sklearn.linear_model import LogisticRegression
X, y = load_iris(return_X_y=True)
clf = LogisticRegression(random_state=0).fit(X, y)
clf.predict(X[:2, :])
clf.predict_proba(X[:2, :])
clf.score(X, y)
```

Neural Networks. Finally

- 딥러닝, CNN등 다양한 기법들 최근 적용중

  → 이 모든 것들의 조상격인 기법이 인공 뉴럴 네트워크(신경망)

  → 멀티퍼셉트론 이후로 정체되었다가 2014년도 이후부터 다시 확장

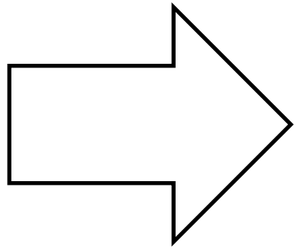


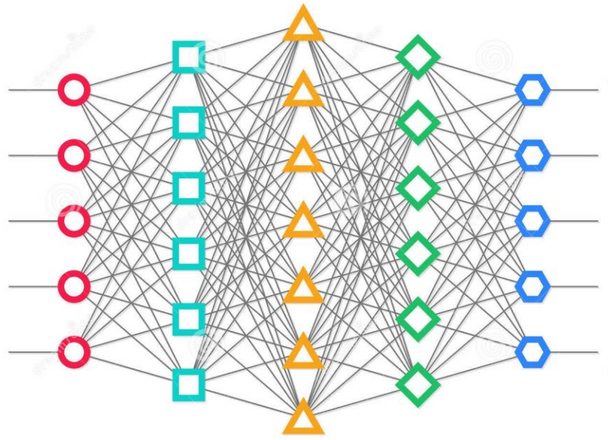
Concept and structure of a neural network

- 인공 신경망

  → 뇌신경 세포 (뉴럴세포구성)에서 부터 나옴

  → 뇌신경 구조의 복잡성을 통한 최적화 기법

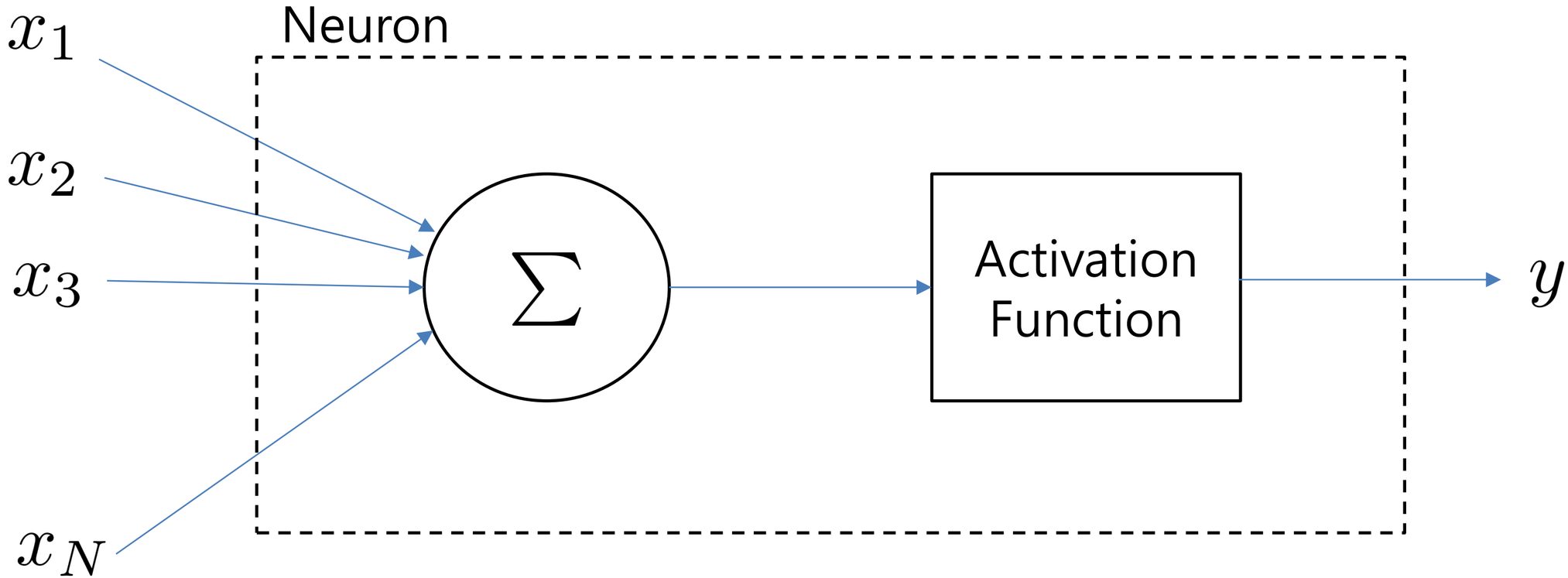




Basics of a neural network

■ 인공 신경망

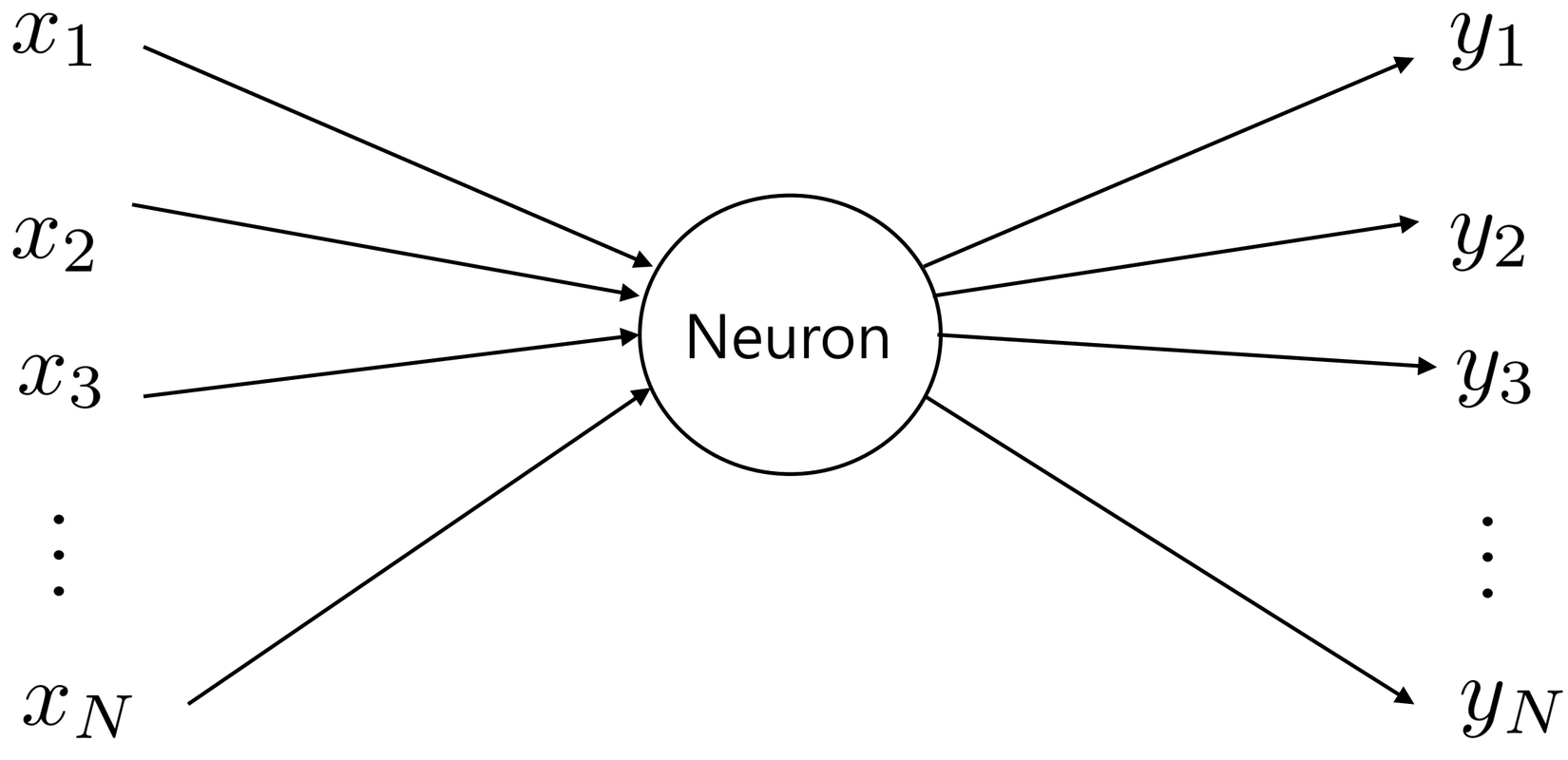
→ 뉴런: 다중 입력 처리 + 단일 출력 + Activation Function



Basics of a neural network

■ 인공 신경망

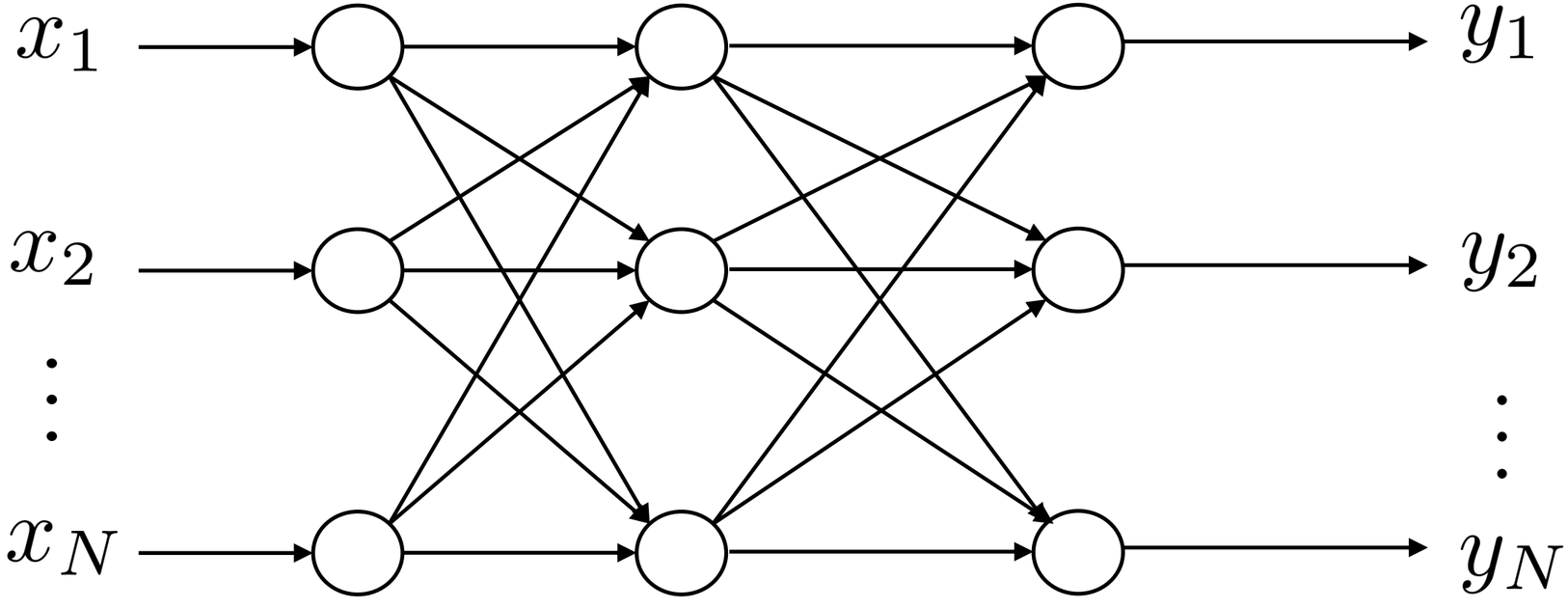
→ 하나의 뉴런은 대개 아래와 같이 대표됨



Basics of a neural network

■ 인공 신경망

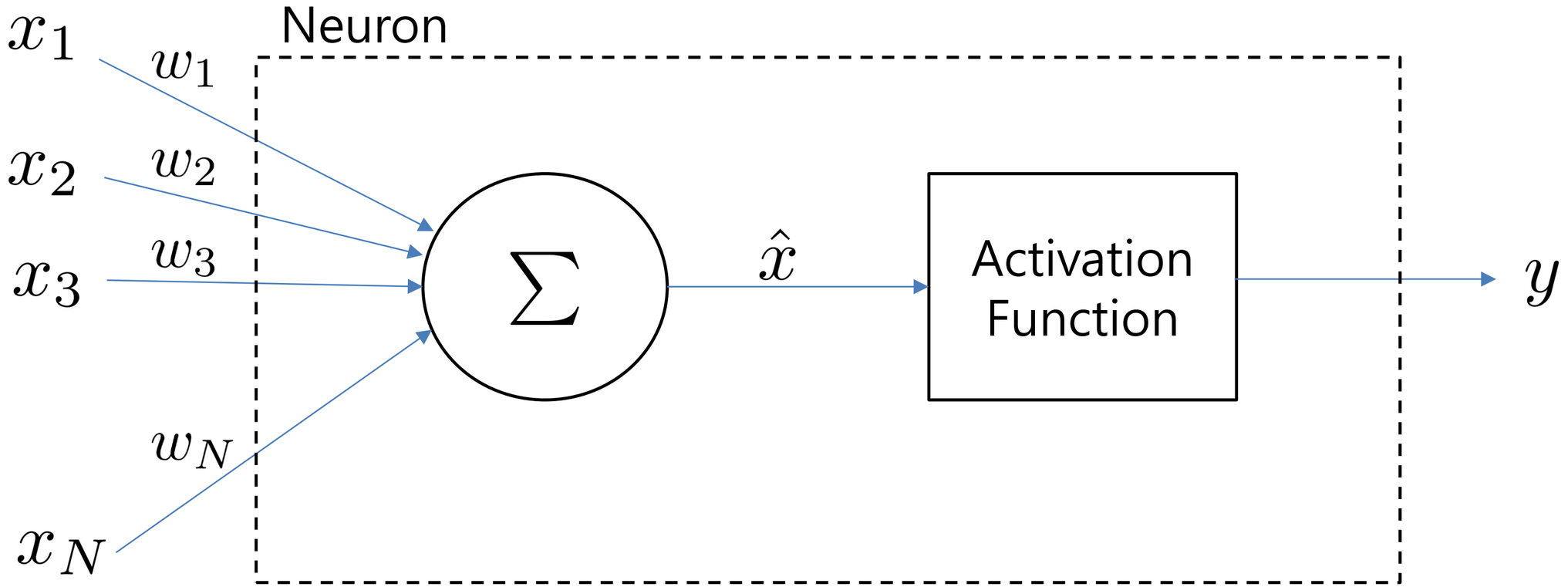
→ 다중 뉴런을 구성하면 아래와 같이 대표됨



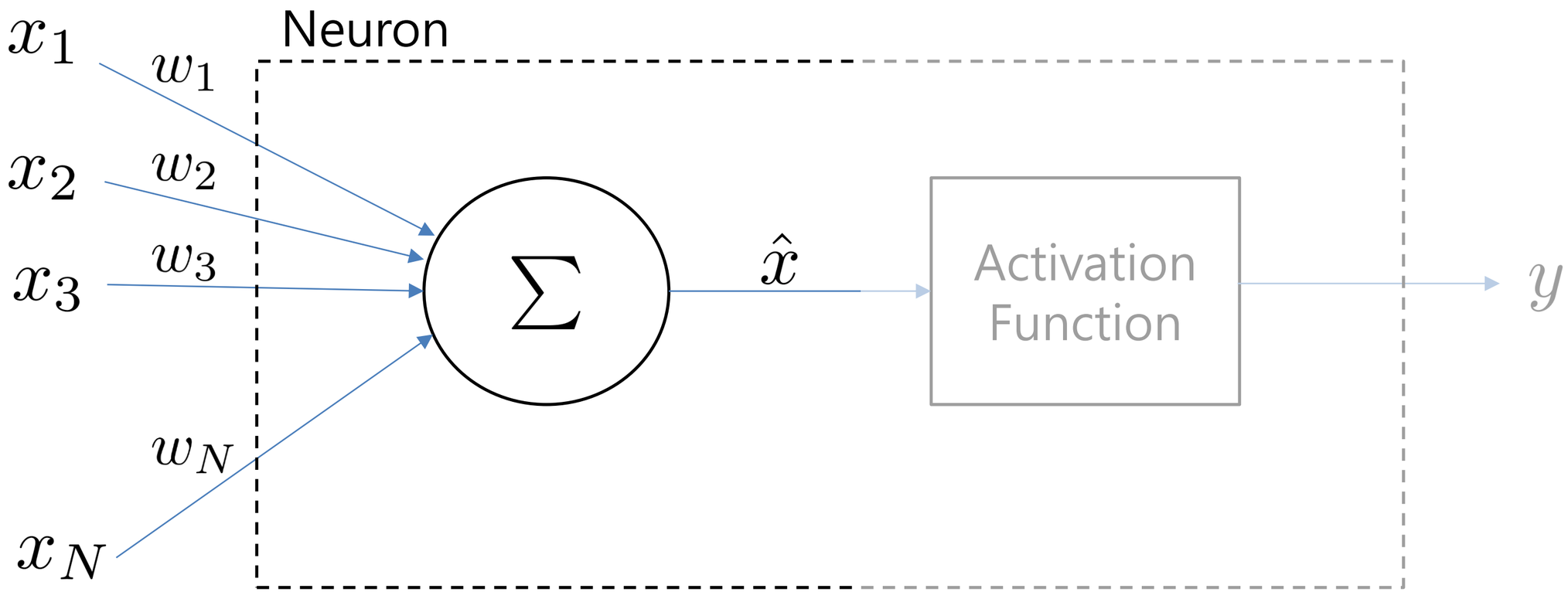
Basics of a neural network

■ 인공 신경망

→ 싱글 뉴런의 공식

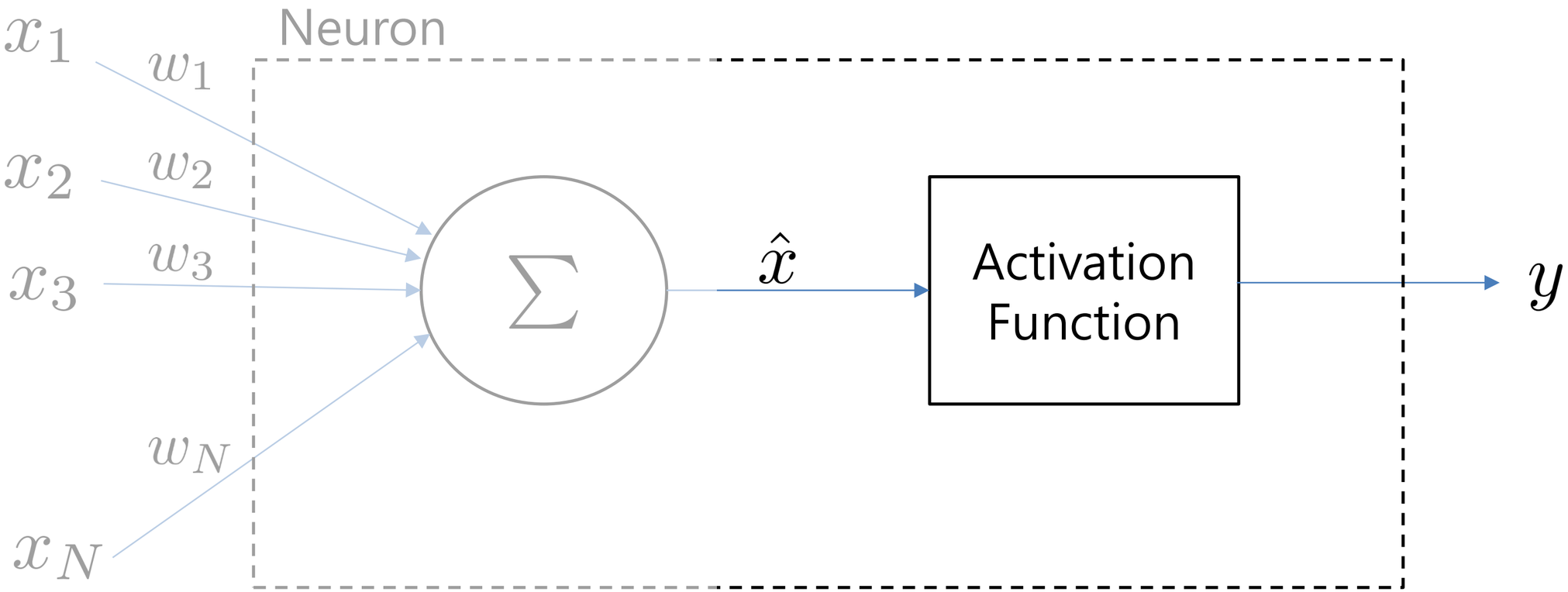


Basics of a neural network



$$\hat{x} = x_1 w_1 + x_2 w_2 + x_3 w_3 + \cdots + x_N w_N + \theta = \sum_{n=1}^{N} x_n w_n + \theta$$

Basics of a neural network



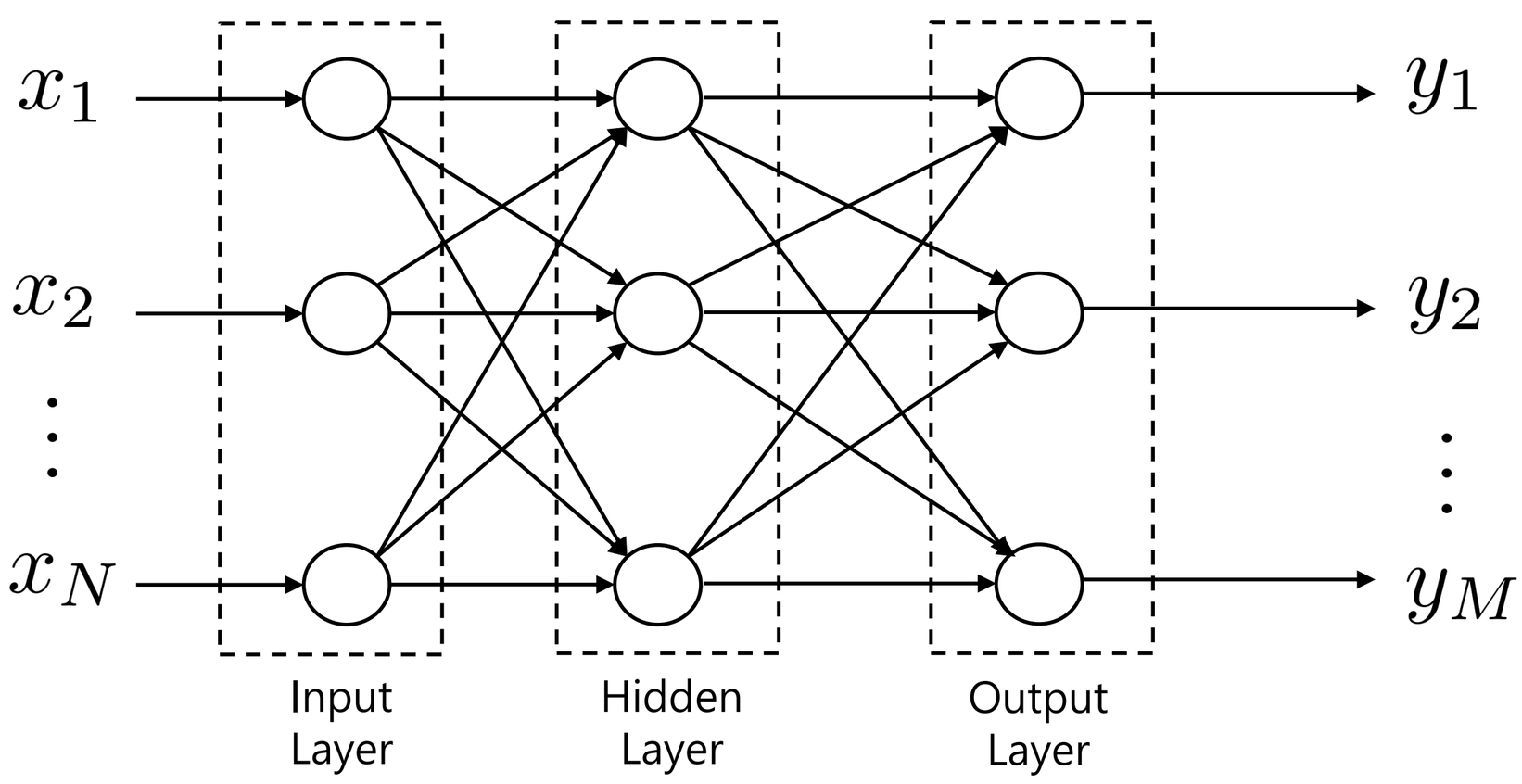
$$y = f(\hat{x})$$

$f(\cdot)$: Activation Function

Basics of a neural network

■ 인공 신경망

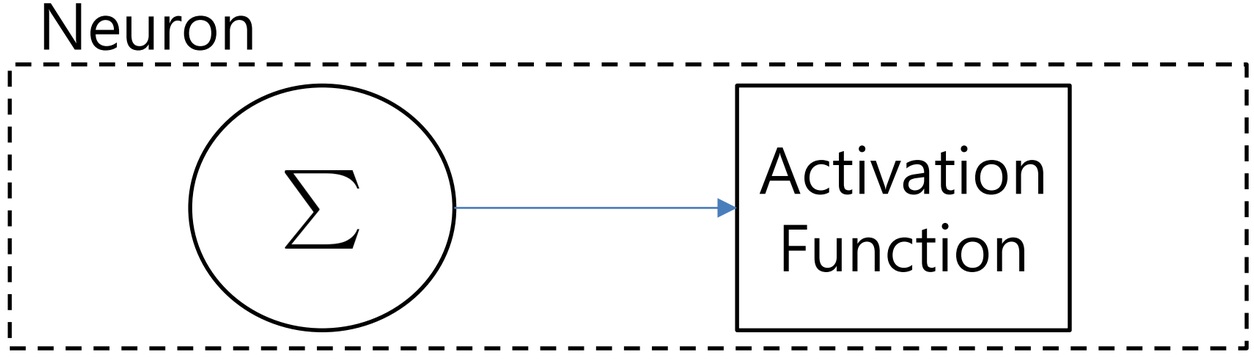
→ 다중 뉴런 중첩 = 멀티 레이어 퍼셉트론 (MLP)

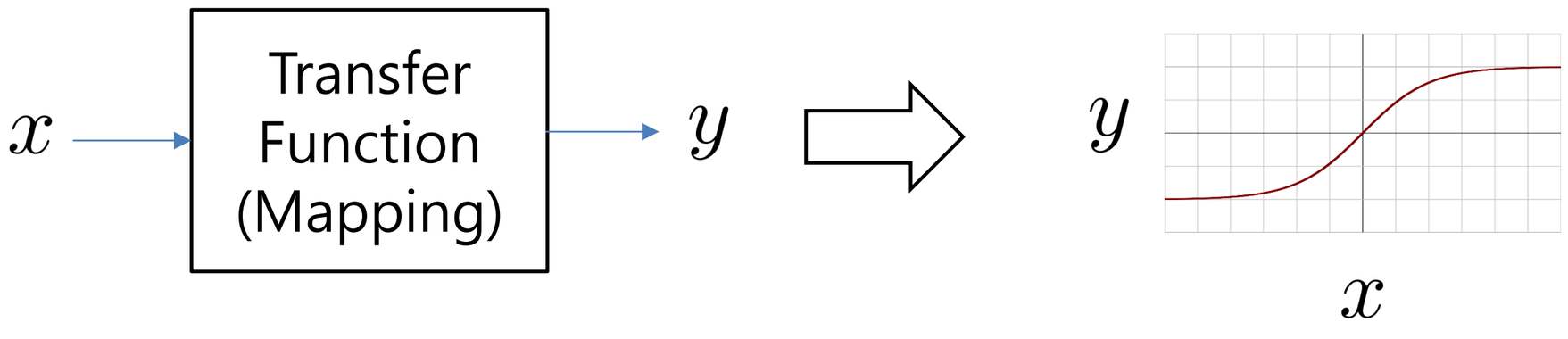


Basics of a neural network

■ 인공 신경망

→ 멀티 레이어 퍼셉트론 (MLP)내 Activation Function

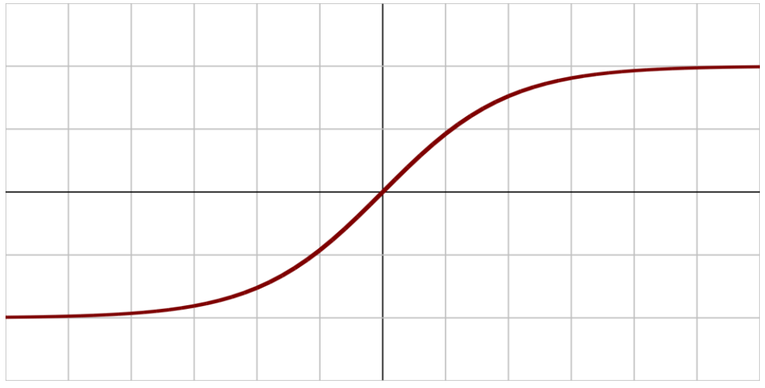




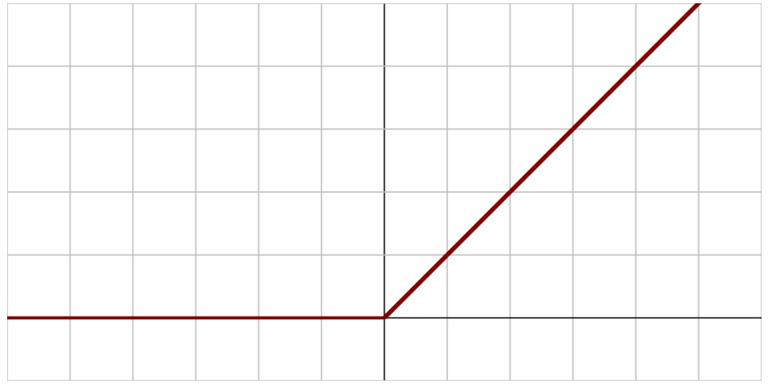
Basics of a neural network

■ 인공 신경망

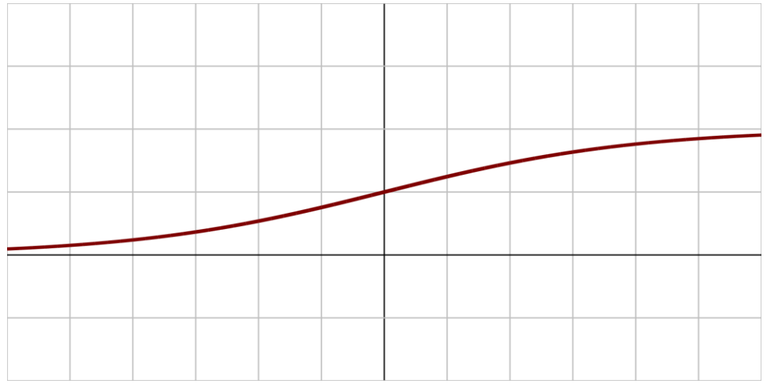
→ Activation Function 종류



tanh



Relu



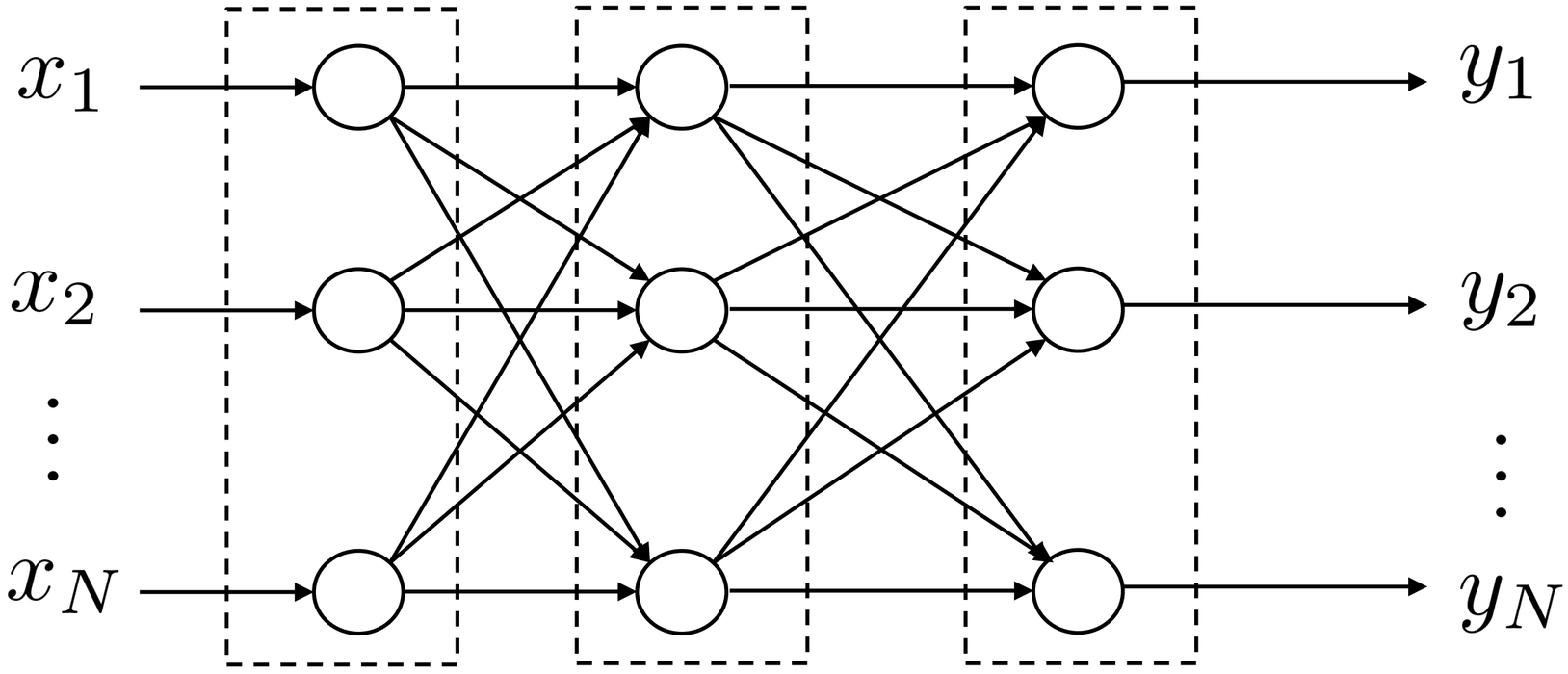
Sigmoid

Basics of a neural network

- 인공 신경망

  → 각각의 노드 주변의 Weight에 대한 랜덤 초기화 필요

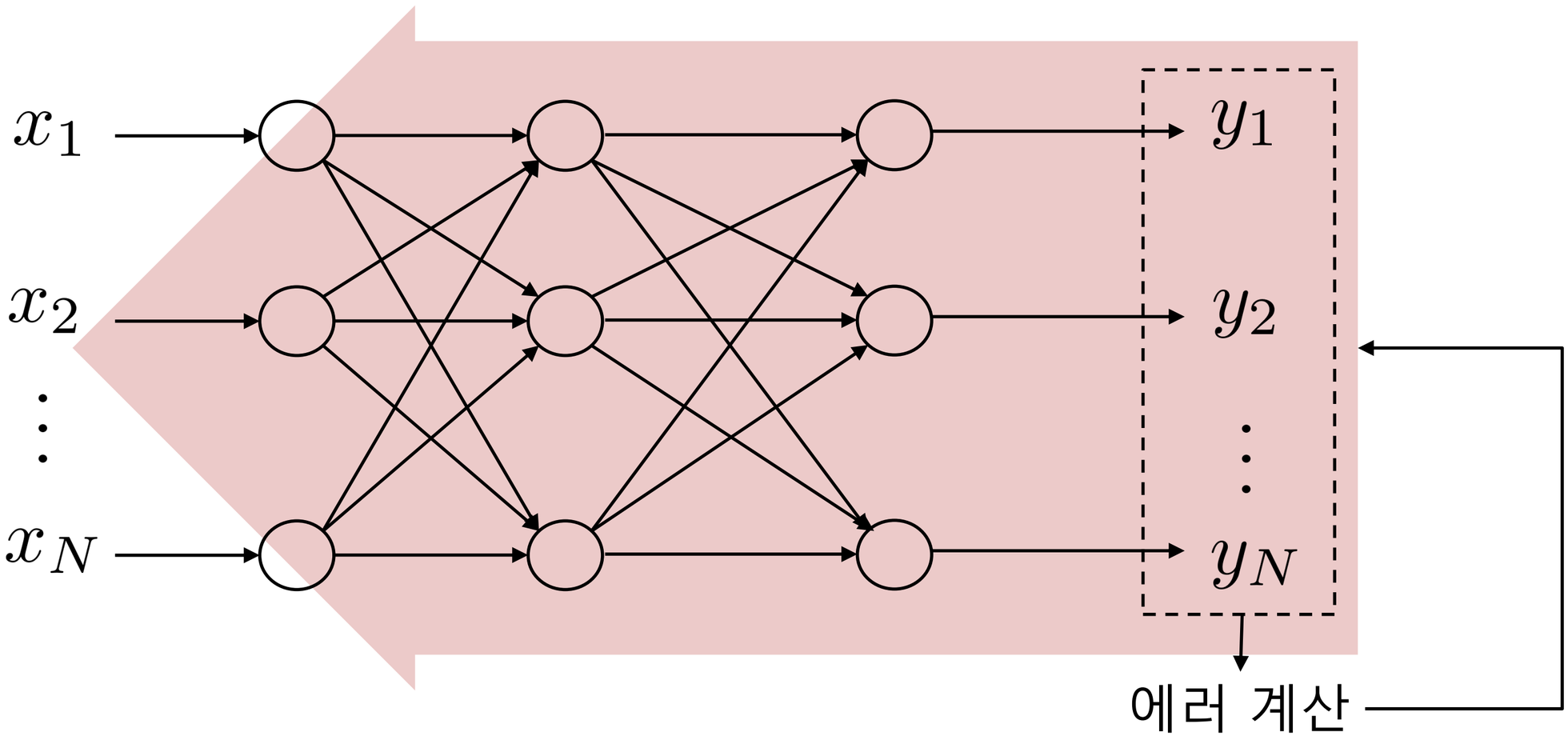
  → 특히 입력 데이터 범위를 [0,1]로 변경하는 것이 성능 향상에 도움



Basics of a neural network

■ 인공 신경망

→ Error에 대한 Back-propagation 과정



Basics of a neural network

■ 인공 신경망

→ Error에 대한 Back-propagation 과정

→ 노드 k에서의 에러 계산 과정

$$E = \frac{1}{2}\sum_k |\hat{y}_k - y_k|^2$$

$$\downarrow$$

$$\frac{dE}{dy_k} = -\sum_k |\hat{y}_k - y_k|$$

$$\rightarrow \quad \frac{dE}{dw_1} = \frac{dE}{dy_1}\frac{dy_1}{dw_1}$$

$$\downarrow$$

$$\hat{w}_1 = w_1 - l \cdot \frac{dE}{dw_1}$$

Required User Input

■ 필수적인 유저 입력 요소 및 값

→ Number of Hidden Layers

→ Size of Hidden Layer

→ Number of Output Nodes

→ choice of predictors

→ Learning rate

→ Momentum (0~2)

Exploring the relationship between predictors and responses

- Neural networks의 다른 닉네임은 블랙박스임
  → 결과값을 가지고 내부 구조가 어떻게 설계되었는지 판단이 불가능
  → 가장 비판을 많이 받는 부분이 이 해당 부분임

- 다만 네트워크 구조가 파악한 데이터간의 상관관계를 Validation 데이터 기반 sensitivity 분석을 통해서 파악 가능

Advantages and Weaknesses of Neural Networks

■ 주요 장단점

→ 가장 최고의 장점은 좋은 예측 성능

→ 노이즈 포함된 데이터에 적응성이 강하고 굉징히 섬세하고 복잡한 데이터간의 관계도를 도출해낼수 있음.

→ 앞서 설명한 바와 같이 관계구조상에서의 시사점을 얻기가 어려운 점이며 따라서 블랙박스라는 닉네임이 붙음.

→ 몇가지 고려사항과 위험요소들이 있음: 첫번째는 범위를 벗어나는 추정의 불가임.

→ 두번째의 경우, 신경망의 경우 변수 선택 매커니즘이 없음. 따라서 Predictor들에 대한 조심스런 선택이 필요. classification and regression trees 혹은 추가적인 dimension reduction techniques 등이 키 Predictor들을 찾는데 쓰임

→ 세번째는 훈련용도의 충분한 데이터가 있어야 신경망 구조의 유연성이 담보된다는 점이다. 작은 데이터 수만 가지고는 성능이 제대로 나올수가 없다. 특히 분류의 경우가 큰 문제임

Advantages and Weaknesses of Neural Networks

■ 주요 장단점

→ 네번째의 경우 weight 값들을 얻는 과정에 있어 해당 값들이 전역 최적점을 찾는 대신에 지역 최적점을 찾는데에 있다. Learning rate나 Momentum등을 적절히 활용하여 토대로 최적점을 찾을수 있으나 해당 활용들이 최적점을 찾는데 있어서 Guarantee가 안된다는 점이다.

→ 마지막으로 실용적 고려를 위해서는 연산 시간을 봐야된다는 점에 있다. 신경망의 경우 작동 시간이 타 기법들에 비해서 훨신 더 오래걸리며 Predictor들이 많을수록 더 길어긴다. real–time or near–real–time 예측을 위해서 작동시간(runtime)이 측정되어야 한다.

Sample Example 1

■ 뉴럴넷 코드 예제 1

```
from sklearn.neural_network import MLPClassifier
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
X, y = make_classification(n_samples=100, random_state=1)
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y,
                                                    random_state=1)
clf = MLPClassifier(random_state=1, max_iter=300).fit(X_train, y_train)
clf.predict_proba(X_test[:1])
clf.predict(X_test[:5, :])
clf.score(X_test, y_test)
```

Sample Example 2

■ 뉴럴넷 코드 예제 2

```
from sklearn.neural_network import MLPRegressor
from sklearn.datasets import make_regression
from sklearn.model_selection import train_test_split
X, y = make_regression(n_samples=200, random_state=1)
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    random_state=1)
regr = MLPRegressor(random_state=1, max_iter=500).fit(X_train, y_train)
regr.predict(X_test[:2])
regr.score(X_test, y_test)
```

Sample Example 3

- 합성곱 뉴럴넷 간단예제

  →Verysimplecnn.py

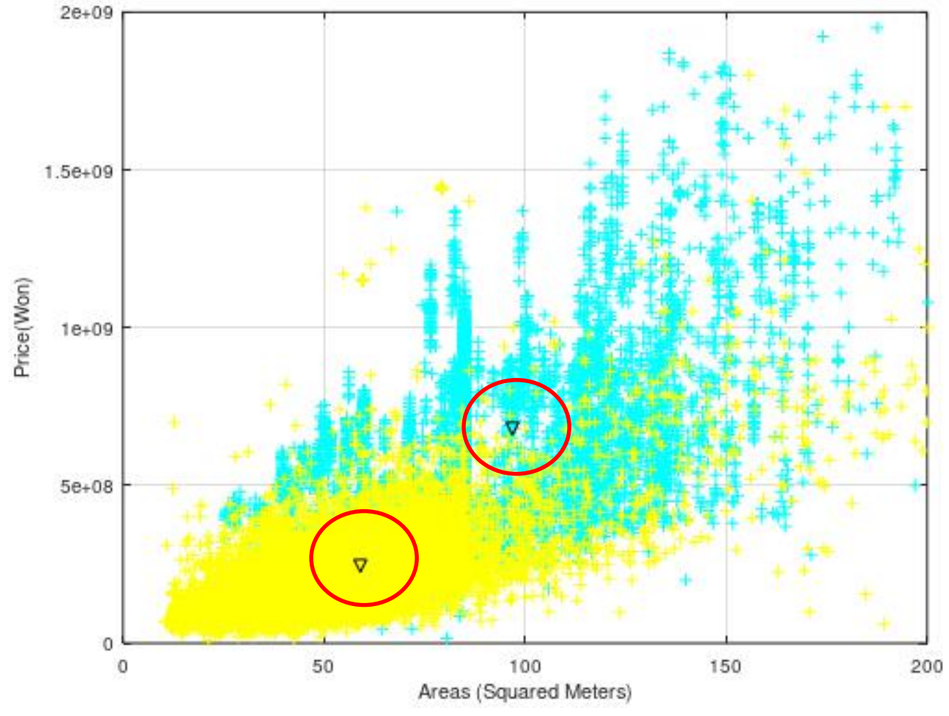
Source: https://towardsdatascience.com/build-your-own-convolution-neural-network-in-5-mins-4217c2cf964f

Unsupervised Learning

- Unsupervised learning

  → a type of machine learning in which the algorithm is not provided with any pre-assigned labels or scores for the training data.

  → Example: Anomaly Detection, Neural Networks, **Clustering**



Clustering/Cluster Analysis

- Cluster Analysis

  → Goal: To segment the data into a set of homogeneous clusters of obs.

  Two approaches:

  → hierarchical clustering: sequentially clustering based on distances between obs. and between clusters

  → k-means clustering

Clustering/Cluster Analysis

- Example: Public utilities

Company	Fixed	RoR	Cost	Load	Demand	Sales	Nuclear	Fuel Cost
Arizona Public Service	1.06	9.2	151	54.4	1.6	9,077	0	0.628
Boston Edison Co.	0.89	10.3	202	57.9	2.2	5,088	25.3	1.555
Central Louisiana Co.	1.43	15.4	113	53	3.4	9,212	0	1.058
Commonwealth Edison Co.	1.02	11.2	168	56	0.3	6,423	34.3	0.7
Consolidated Edison Co. (NY)	1.49	8.8	192	51.2	1	3,300	15.6	2.044
Texas Utilities Co.	1.16	11.7	104	54	−2.1	13,507	0	0.636
Wisconsin Electric Power Co.	1.2	11.8	148	59.9	3.5	7,287	41.1	0.702
United Illuminating Co.	1.04	8.6	204	61	3.5	6,650	0	2.116
Virginia Electric & Power Co.	1.07	9.3	174	54.3	5.9	10,093	26.6	1.306

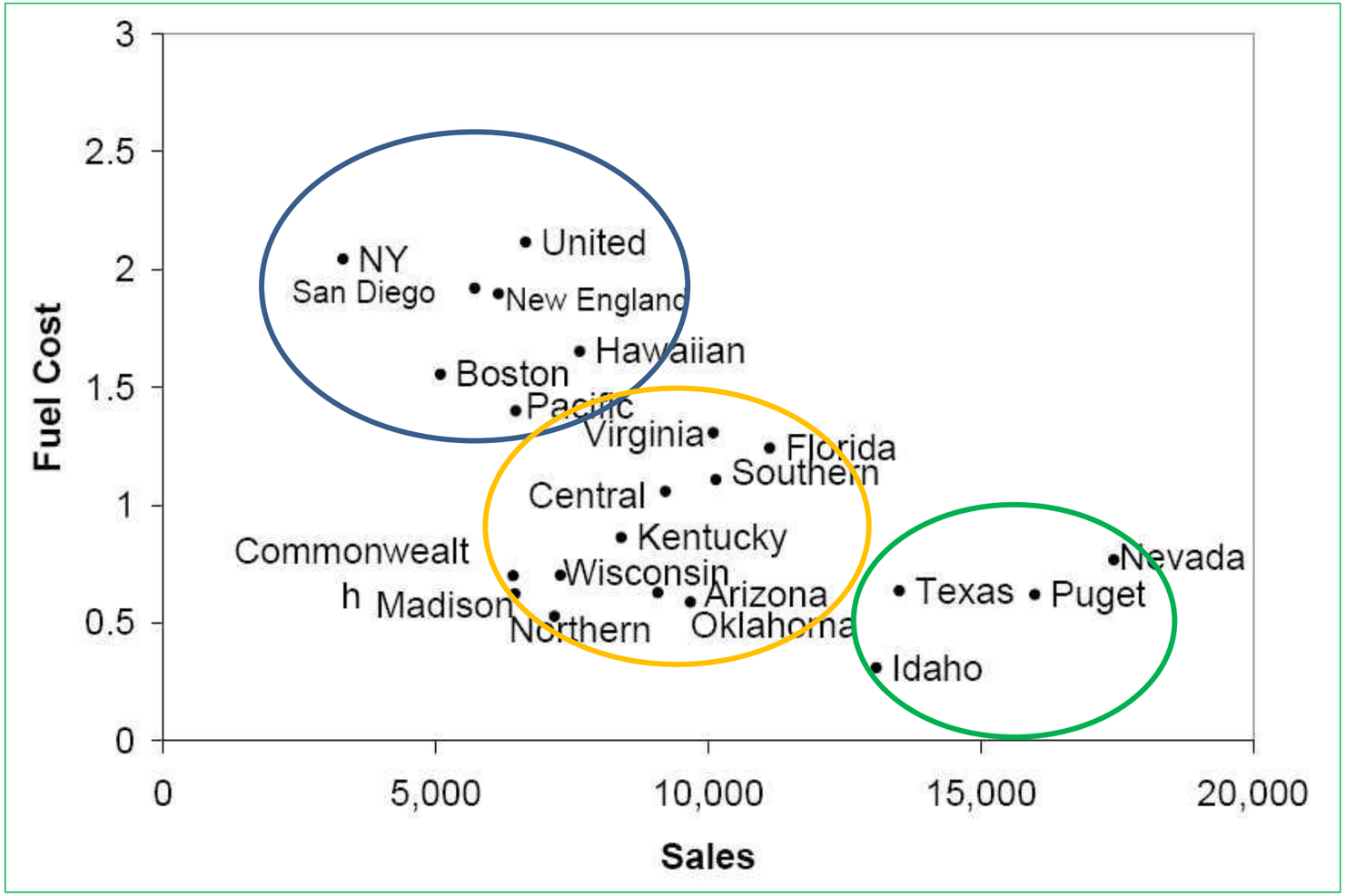
22 US public utilities, 8 variables – to form groups of similar utilities

→ may be useful to predict the cost impact of deregulation

→ 각 cluster의 대표적인 하나의 record에 대해서만 분석해도 됨

Clustering/Cluster Analysis

- Example: Public utilities



Clustering/Cluster Analysis

- In prior example, clustering was done by eye
- Multiple dimensions require formal algorithm with

  → a distance measure

  → a way to use the distance measure in forming clusters
- Two algorithms:

  → Hierarchical methods

  → Nonhierarchical methods

Measuring distance between two records

- Euclidean distance

$$d_{ij} = \sqrt{(x_{i1} - x_{j1})^2 + \cdots + (x_{ip} - x_{jp})^2}$$

- Normalizing (standardizing) – making z-score

$$x \rightarrow z = \frac{x - \bar{x}}{s}$$

Measuring distance between two records

ORIGINAL AND NORMALIZED MEASUREMENTS FOR SALES AND FUEL COST

Company	Sales	Fuel Cost	NormSales	NormFuel
Arizona Public Service	9,077	0.628	0.0459	−0.8537
Boston Edison Co.	5,088	1.555	−1.0778	0.8133
Central Louisiana Co.	9,212	1.058	0.0839	−0.0804
Commonwealth Edison Co.	6,423	0.7	−0.7017	−0.7242
Consolidated Edison Co. (NY)	3,300	2.044	−1.5814	1.6926

Clustering/Cluster Analysis

- Other distance measures for numerical data

  → Distance measure의 선택이 매우 중요한 역할

  → Euclidean distance는 highly scale dependent → (normalizing?) unequal weighting을 하고 싶을 땐?

  → measurement들 간의 relationship을 고려하지 못함 -> strongly correlated되었다면?

  → outlier에 sensitive

Measuring distance between two records

Additional distance metrics

- Correlation-based similarity (similarity than distance):

$$d_{ij} = 1 - r_{ij}^2$$

where r_{ij} = correlation coefficient of record i and j

→ actually, dissimilarity

- Manhattan distance (city block):Euclidean distance

$$d_{ij} = \sum_{m=1}^{p} |x_{im} - x_{jm}|$$

Measuring distance between two records

Additional distance metrics

- Statistical distance (Mahalanobis distance): considered the correlation between measurements

$$d_{ij} = \sqrt{(\mathbf{x_i} - \mathbf{x_j})^{\mathbf{T}} \mathbf{S}^{-\mathbf{1}} (\mathbf{x_i} - \mathbf{x_j})}$$

Measuring distance between two records

Distance measures for categorical data

■ When binary values such as:

Ex.

p = 1 0 0 0 0 0 0 0 0 0

q = 0 0 0 0 0 0 1 0 0 1

		Record *j* 0	Record *j* 1	
Record *i*	0	*a*	*b*	$a+b$
	1	*c*	*d*	$c+d$
		$a+c$	$b+d$	*p*

■ Simple Matching coefficient = $\frac{a+d}{p}$

■ Jaquard's coefficient = $\frac{d}{b+c+d}$

두 record가 여러 predictor에서 비교할 대상이 없어서 "similar"로 판정되는 경우를 방지하기 위함 (ex. "콜벳(스포츠카)을 가지고 있는가"의 비교에서, 두 사람이 모두 Yes이면 "similar"라고 볼 수 있으나 모두 No라고 해서 "similar"로 보기는 어려움)

Measuring distance between two clusters

- Distance between clusters A and B:

→ Minimum distance (single linkage): min(distance(A_i , B_j))
: 가장 가까운 두 record 간의 거리

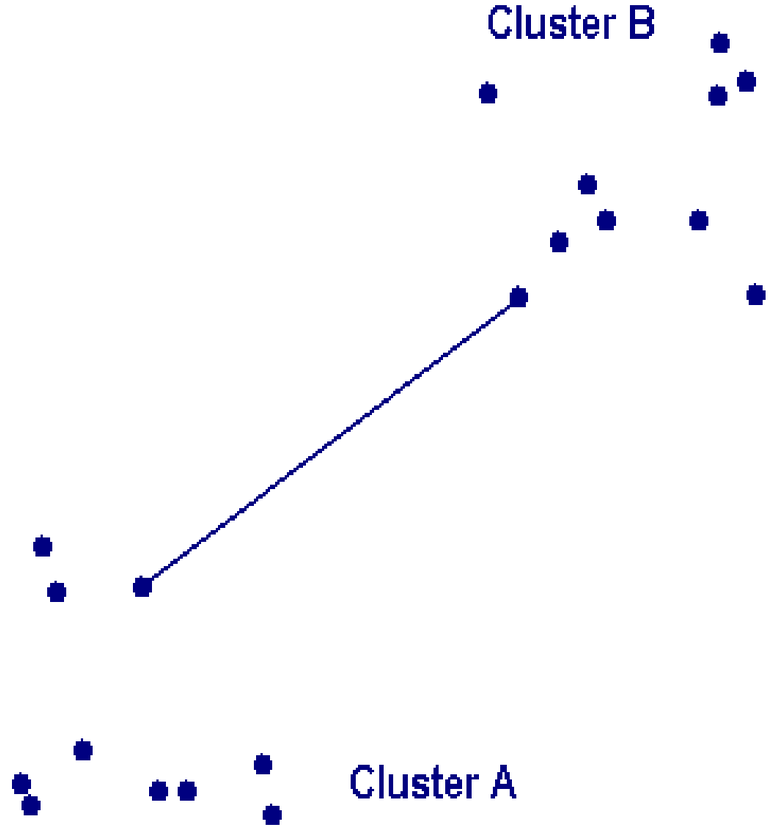
→ Maximum distance (complete linkage): max(distance(A_i , B_j))
: 가장 먼 두 record 간의 거리

→ Average distance (average linkage): Avg(distance(A_i ,B_j))
: 가능한 모든 두 record 간의 거리의 평균

→ Centroid distance: distance(X_A, X_B)
: 두 cluster의 centroid 간의 거리, 단 centroid = 각 성분들의 평균으로 이루어진 벡터

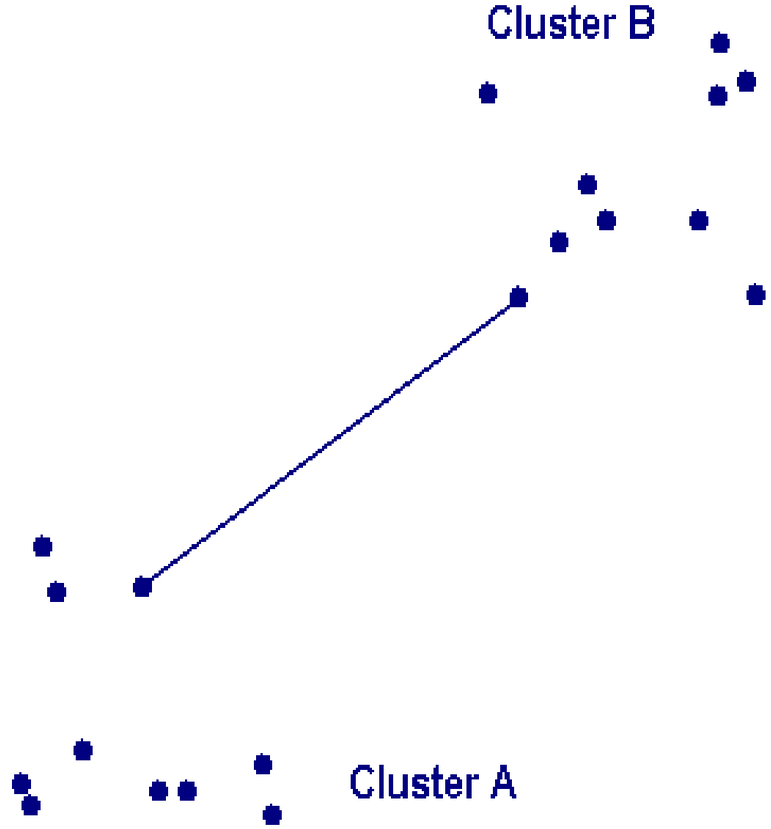
Simple Linkage Algorithm

- Link based on the distance based on the smallest distance between objects from clusters.



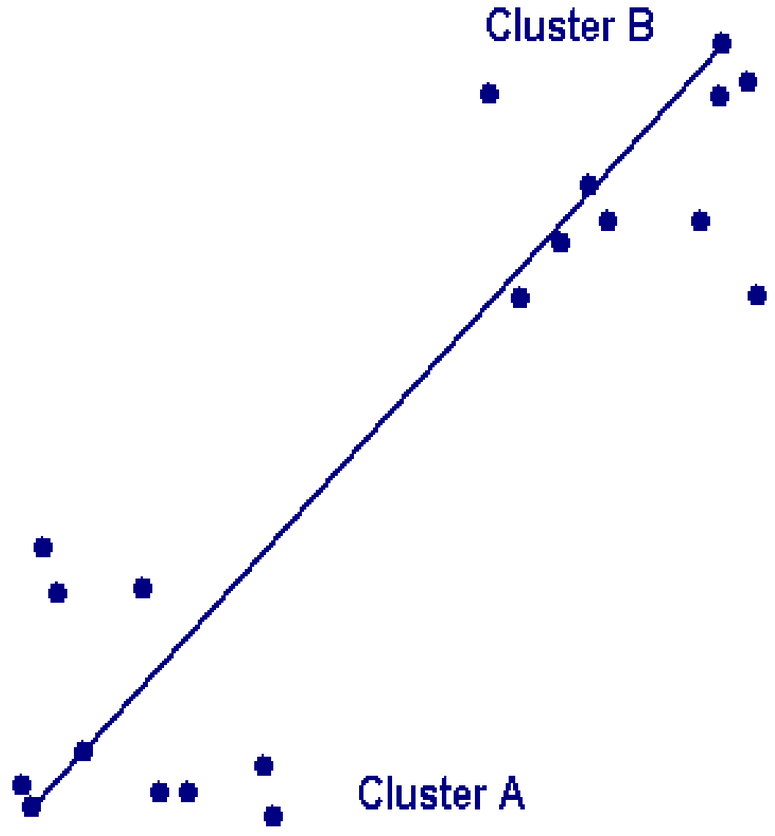
Simple Linkage Algorithm

- Link based on the distance based on the smallest distance between objects from clusters.



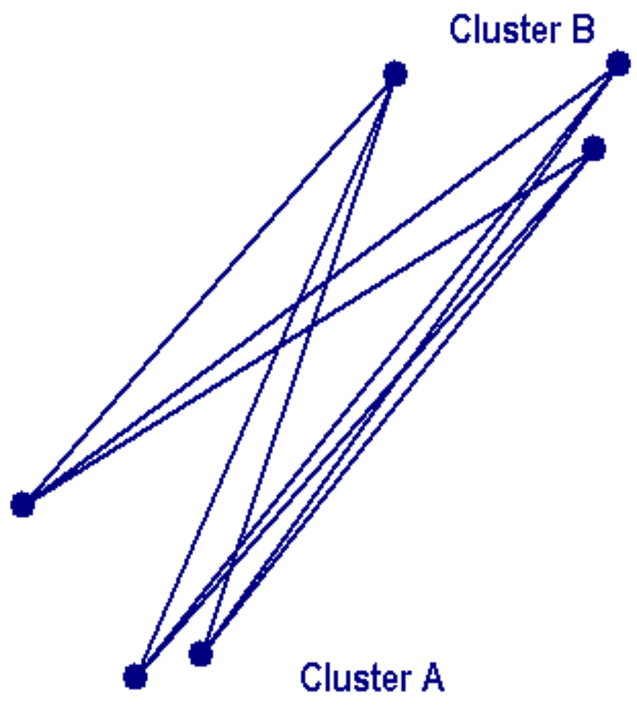
Complete Linkage Algorithm

- Link based on the distance based on the furthest distance between objects from clusters.



Average Linkage Clustering

- The average distance is calculated from the distance between each point in a cluster and all other points in another cluster. The two clusters with the lowest average distance are joined together to form a new cluster.



Measuring distance between two clusters

	Arizona	Boston	Central	Commonwealth	Consolidated
Arizona	0				
Boston	2.01	0			
Central	0.77	1.47	0		
Commonwealth	0.76	1.58	1.02	0	
Consolidated	3.02	1.01	2.43	2.57	0

- Cluster A = {Arizona, Boston}
- Cluster B = {Central, Commonwealth, Consolidated}
  - → Minimum distance = 0.76
  - → Maximum distance = 3.02
  - → Average distance = 1.44
  - → Centroid distance = 0.38

Measuring distance between two clusters

- 어느 distance를 쓸 것인가? -> domain knowledge가 필요
  → Minimum distance -> Cluster들이 chain-형태일 때
  → (ex. 줄 맞추어 심은 작물이나 수로를 따라 번지는 유행병의 특성, 광물 탐사 등)

- Max 또는 Average distance -> cluster가 spherical이거나 그 특성을 알기 힘들 때
  → (ex. 다양한 속성에 의해 고객분류 시, default choice)

Hierarchical Clustering

- Agglomerative Methods

  → Begin with n-clusters (each record is its own cluster)

  → Keep joining records into clusters until one cluster is left

  → Most popular

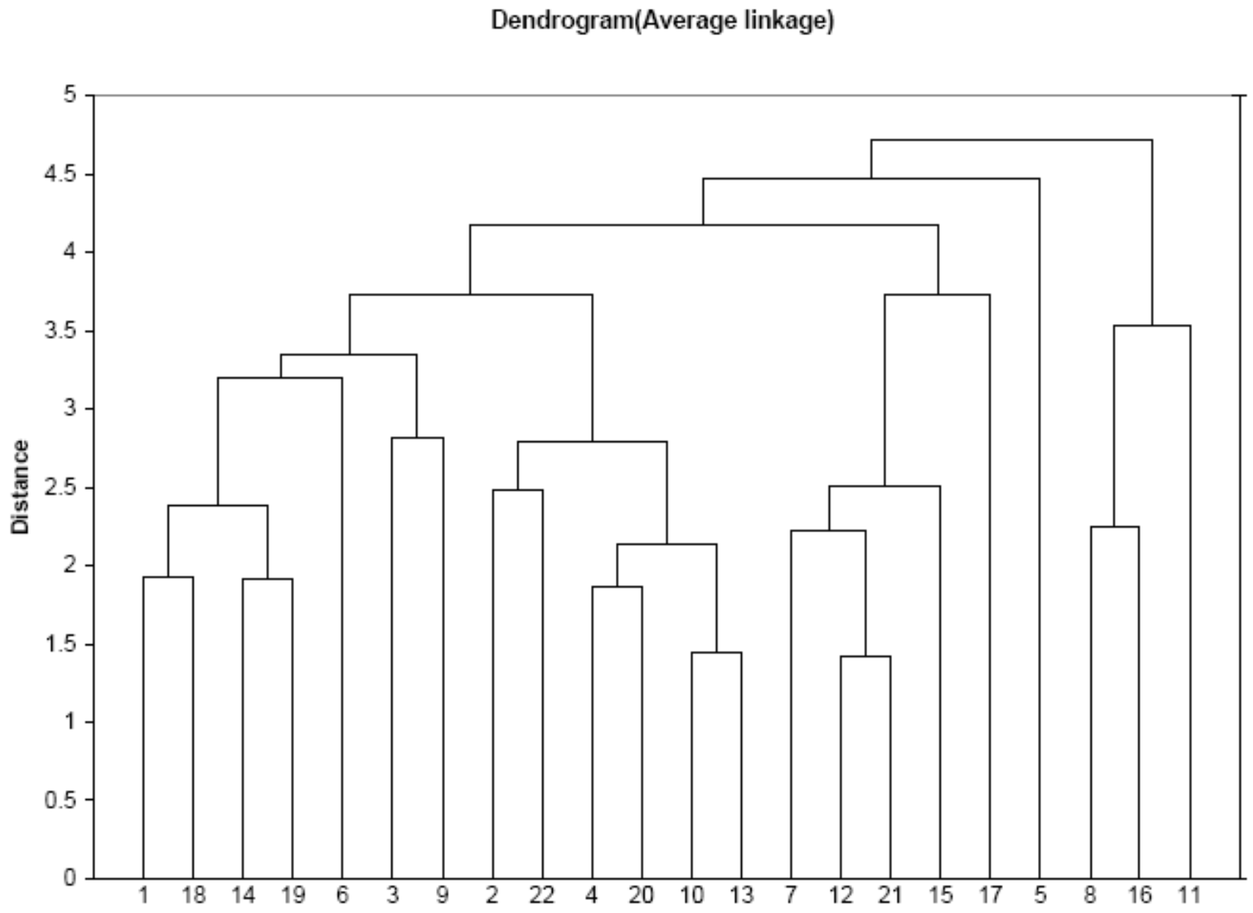
- Divisive Methods

  → Start with one all-inclusive cluster

  → Repeatedly divide into smaller clusters

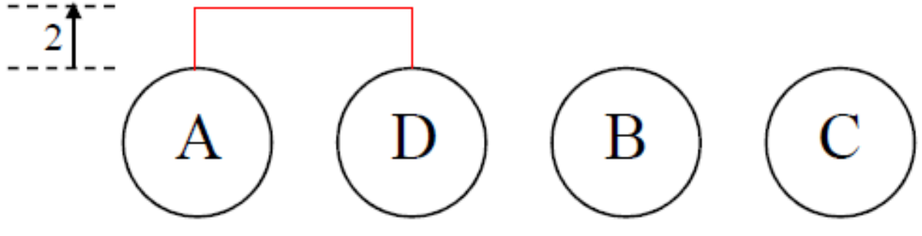
Hierarchical Clustering

- A Dendrogram shows the cluster hierarchy



Hierarchical Clustering Algorithm

Current Clusters

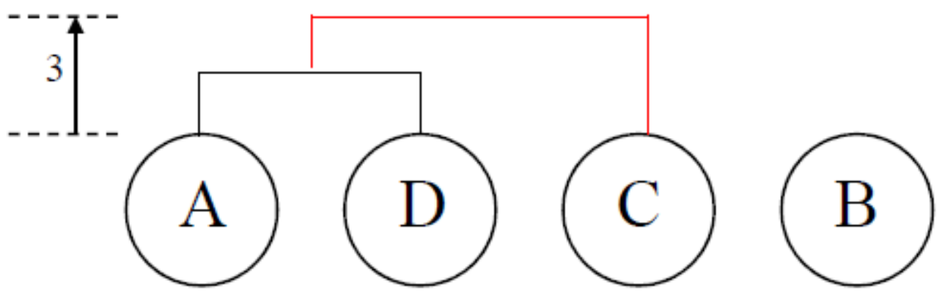


Distance Matrix

Dist	A	B	C	D
A		20	7	2
B			10	25
C				3
D				

Hierarchical Clustering Algorithm

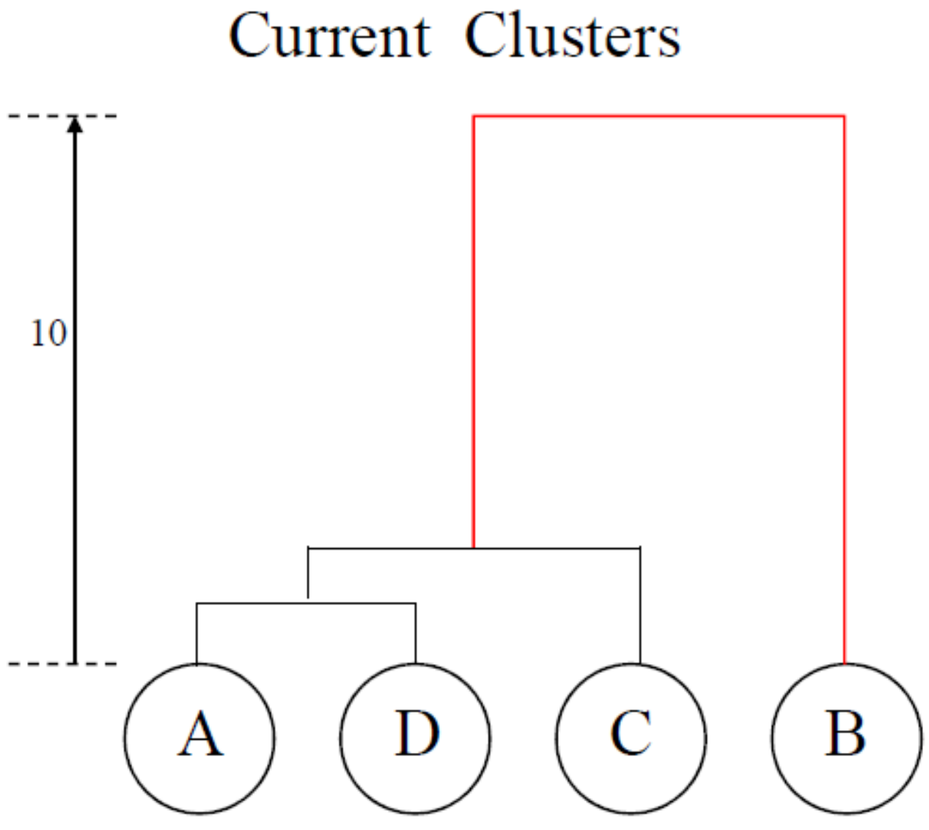
Current Clusters



Distance Matrix

Dist	AD	B	C	
AD		20	3	
B			10	
C				

Hierarchical Clustering Algorithm

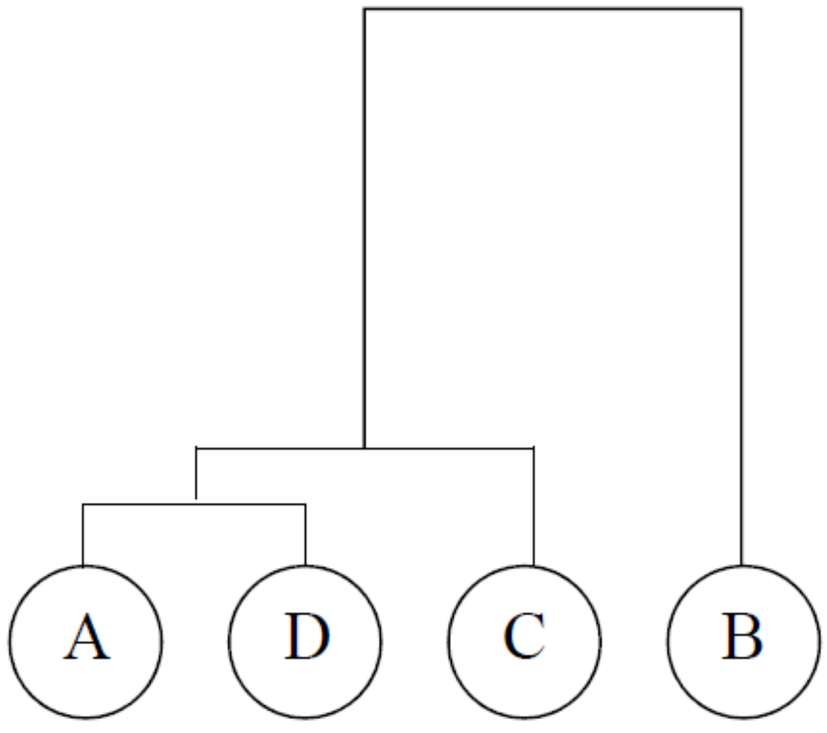


Distance Matrix

Dist	AD C	B		
AD C		10		
B				

Hierarchical Clustering Algorithm

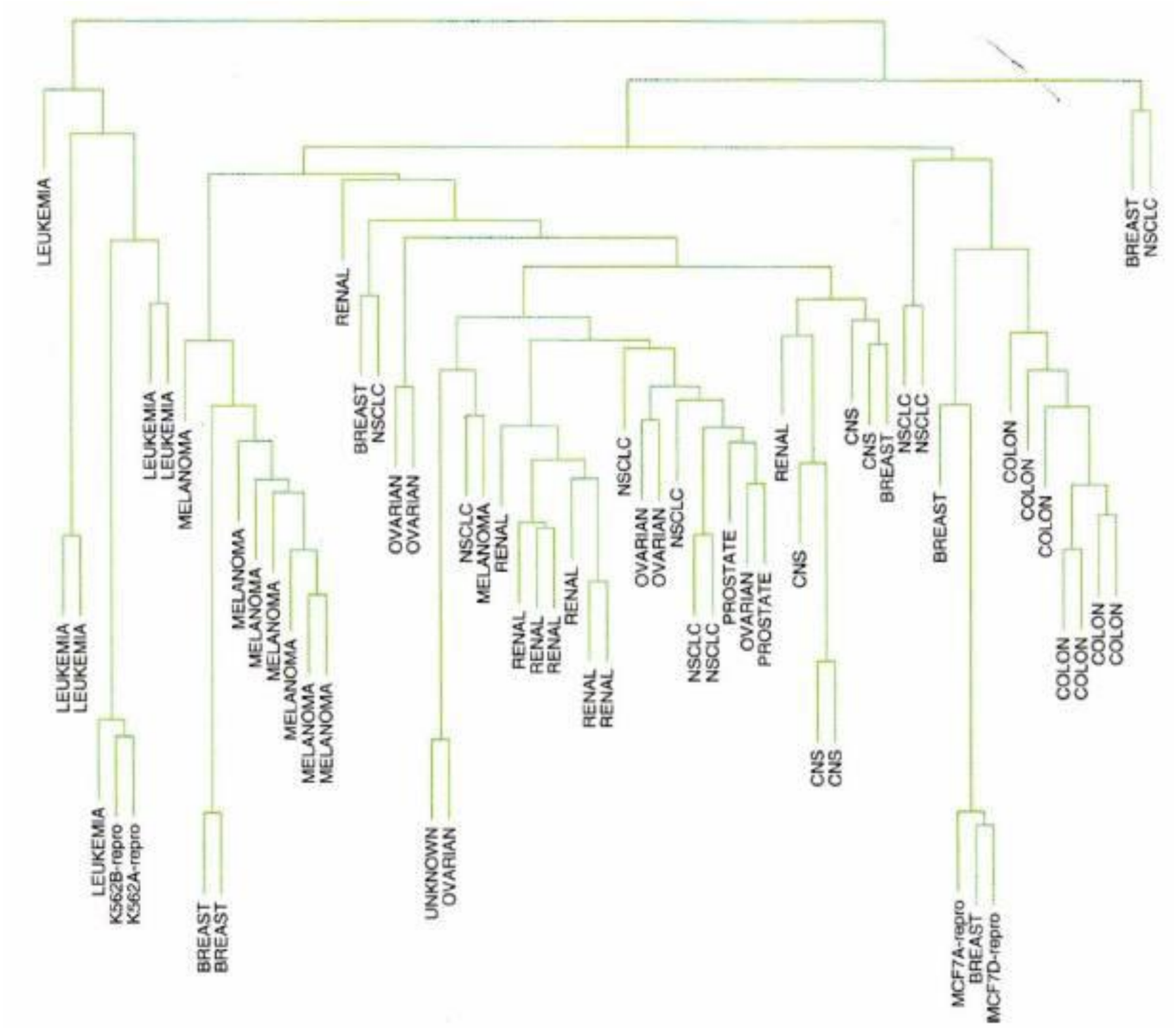
Final Result



Distance Matrix

Dist	AD CB			
AD CB				

Resulting Hierarchical Clustering



Hierarchical (agglomerative) clustering

- Ex. 5 utilities, 2 measures, distance matrix in Table 14.3

Company	NormSales	NormFuel
Arizona Public Service	0.0459	−0.8537
Boston Edison Co.	−1.0778	0.8133
Central Louisiana Co.	0.0839	−0.0804
Commonwealth Edison Co.	−0.7017	−0.7242
Consolidated Edison Co. (NY)	−1.5814	1.6926

	Arizona	Boston	Central	Commonwealth	Consolidated
Arizona	0				
Boston	2.01	0			
Central	0.77	1.47	0		
Commonwealth	0.76	1.58	1.02	0	
Consolidated	3.02	1.01	2.43	2.57	0

→ {Arizona}와 {Commonwealth}를 merge (∵ closest in Euclidean)

→ 4개 cluster를 가지고 distance matrix를 다시 구성

Hierarchical (agglomerative) clustering

- Minimum distance (single linkage)의 경우 새로운 4×4 matrix using single linkage

	Arizona–Commonwealth	Boston	Central	Consolidated
Arizona–Commonwealth	0			
Boston	min(2.01,1.58)	0		
Central	min(0.77,1.02)	1.47	0	
Consolidated	min(3.02,2.57)	1.01	2.43	0

→ {Arizona, Commonwealth}와 {Central}을 merge

→ 이런 식으로 계속 merge - 체인 형태를 띠는 cluster가 만들어짐

Hierarchical (agglomerative) clustering

■ Maximum distance (compete linkage)의 경우

→ min → max로 바꾸면 됨

→ 초기단계에 좁은 범위의 record들이 합쳐지는 경향

→ spherical 형태의 cluster가 만들어짐

■ Average distance (average linkage)의 경우

→ min → avg로 바꾸면 됨

	Arizona–Commonwealth	Boston	Central	Consolidated
Arizona–Commonwealth	0			
Boston	min(2.01,1.58)	0		
Central	min(0.77,1.02)	1.47	0	
Consolidated	min(3.02,2.57)	1.01	2.43	0

Hierarchical (agglomerative) clustering

- Centroid distance (average group linkage)의 경우

  → Each cluster is represented by the vector of means

  → Average distance에서는 모든 pair간의 거리의 평균을 계산해야 하지만

  → Centroid distance에서는 group mean간의 거리 한 번만 계산

Hierarchical (agglomerative) clustering

■ Ward's method

→ 작은 cluster와 record들을 merge해 나가는 것은 동일

→ record들을 clustering할 때 발생하는 "lost of information"을 고려함

record + … + record -> cluster (– information)

ESS(error sum of squares): 각 group의 제곱오차합(SSE)

ex. 2,2,2,2,5,6,6,0,0,0 하나의 group으로 하여 평균인 2.5를 대표값으로 한다면 ESS=50.5

(2,2,2,2), (5), (6,6), (0,0,0) 4개 group으로 한다면 ESS=0+0+0+0=0

■ Loss of info가 가장 적게 증가하는 방향을 선택하는 것임

→ Cluster가 동일 크기의 convex 모양을 띠는 경향

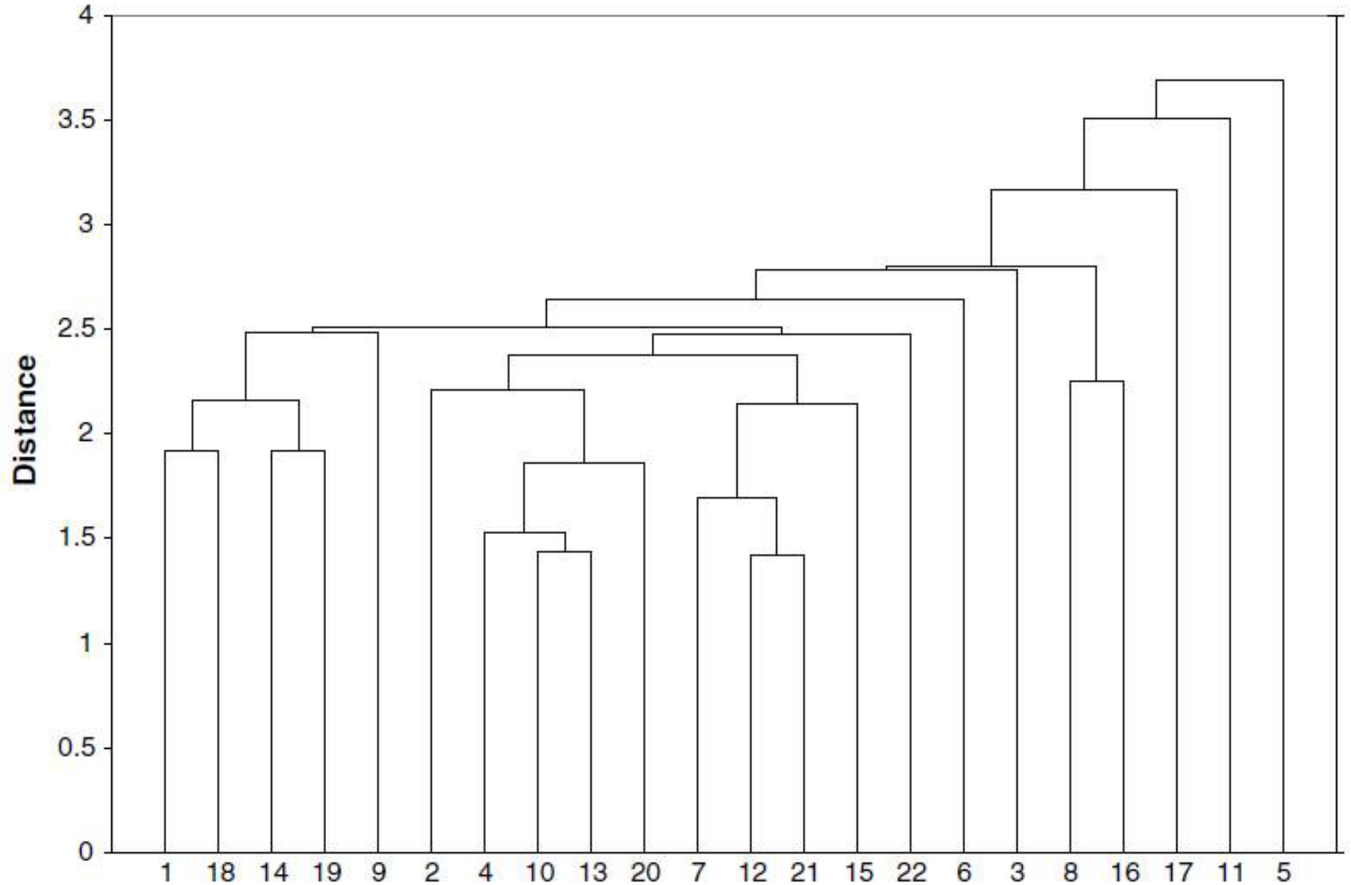
Hierarchical (agglomerative) clustering

- The Hierarchical Clustering Steps (Using Agglomerative Method)
  - → Start with n clusters (each record is its own cluster)
  - → Merge two closest records into one cluster
  - → At each successive step, the two clusters closest to each other are Merged

Dendrogram, from bottom up, illustrates the process

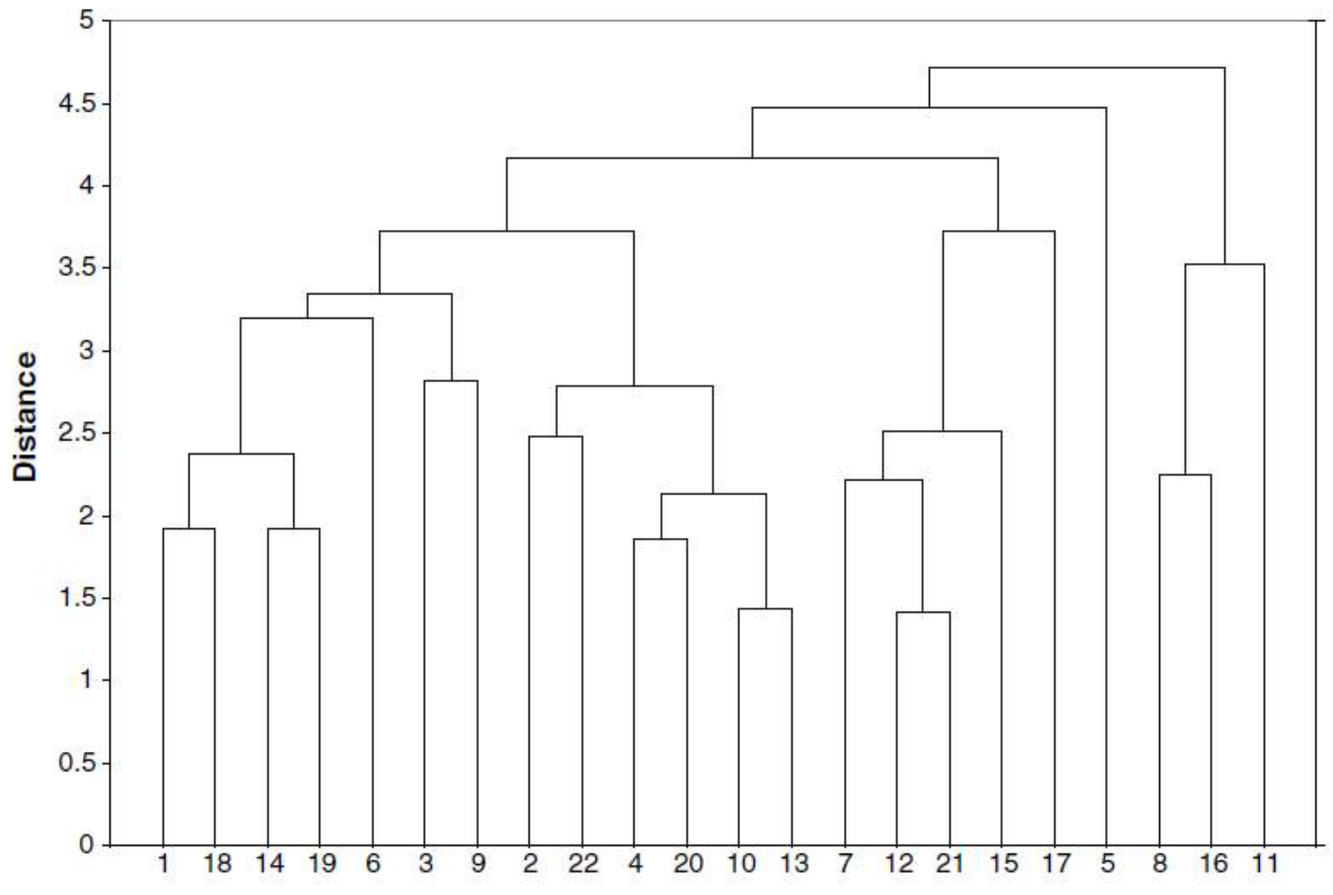
Hierarchical (agglomerative) clustering

22 utilities, 8 normalizing measurements, Euclidean distance, single linkage

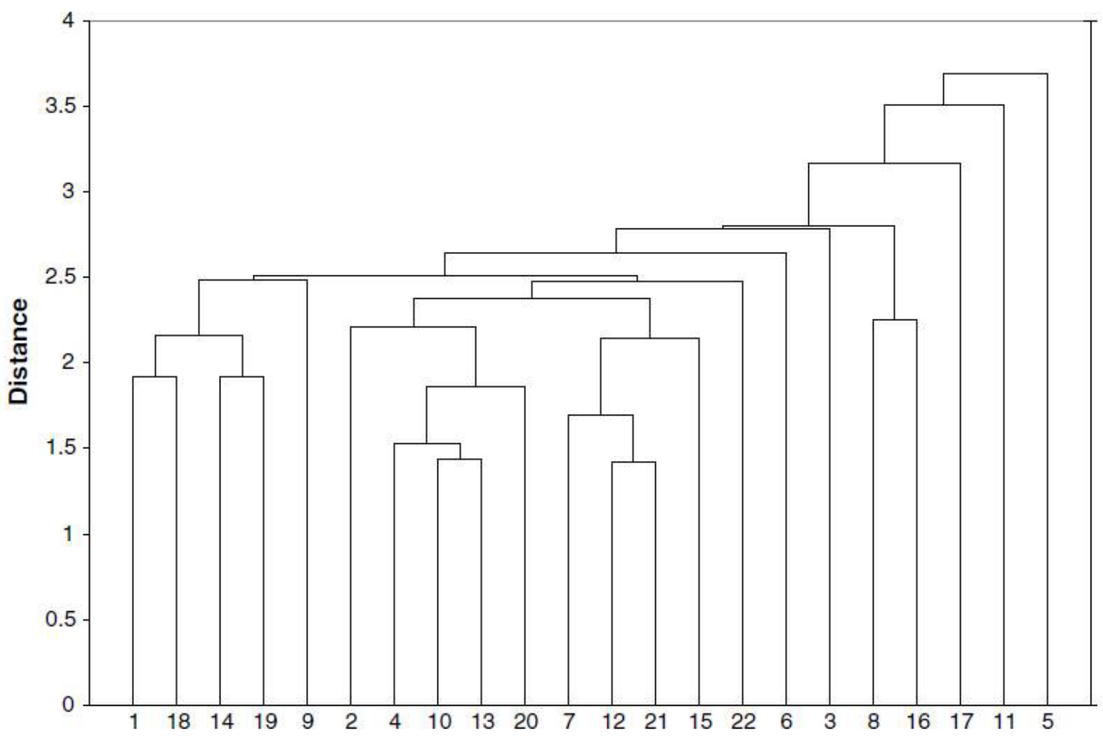


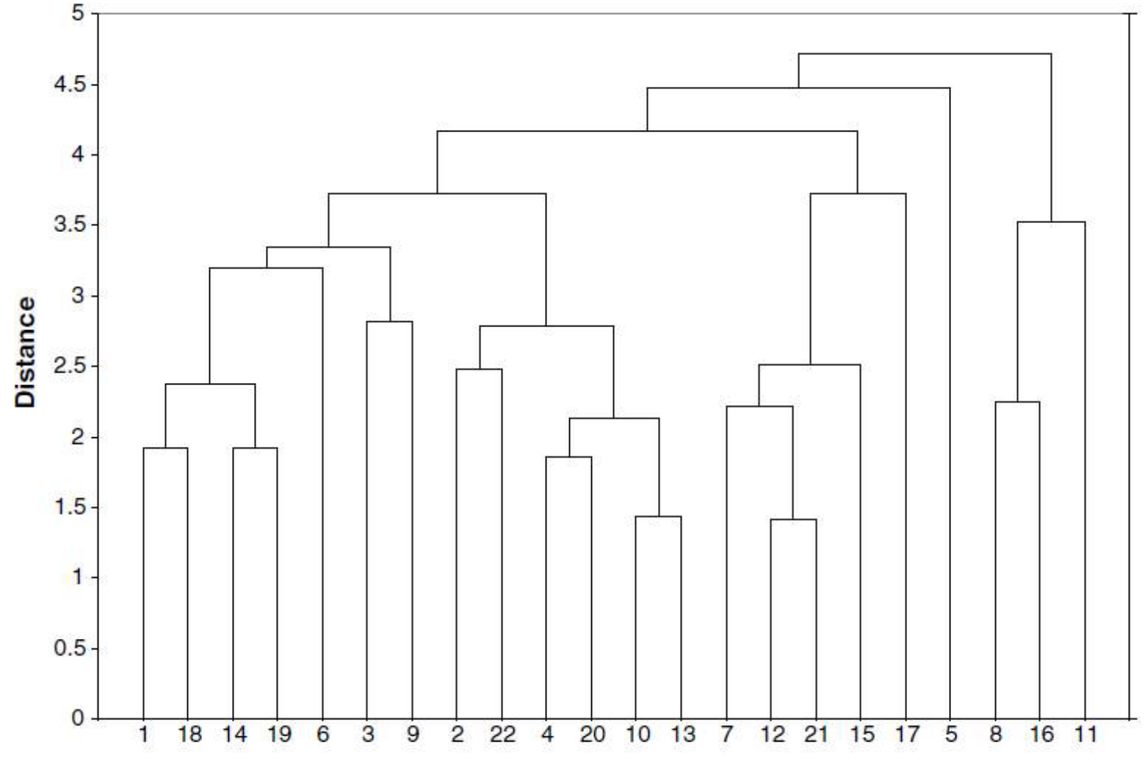
Hierarchical (agglomerative) clustering

Average linkage를 사용했을 때의 dendrogram



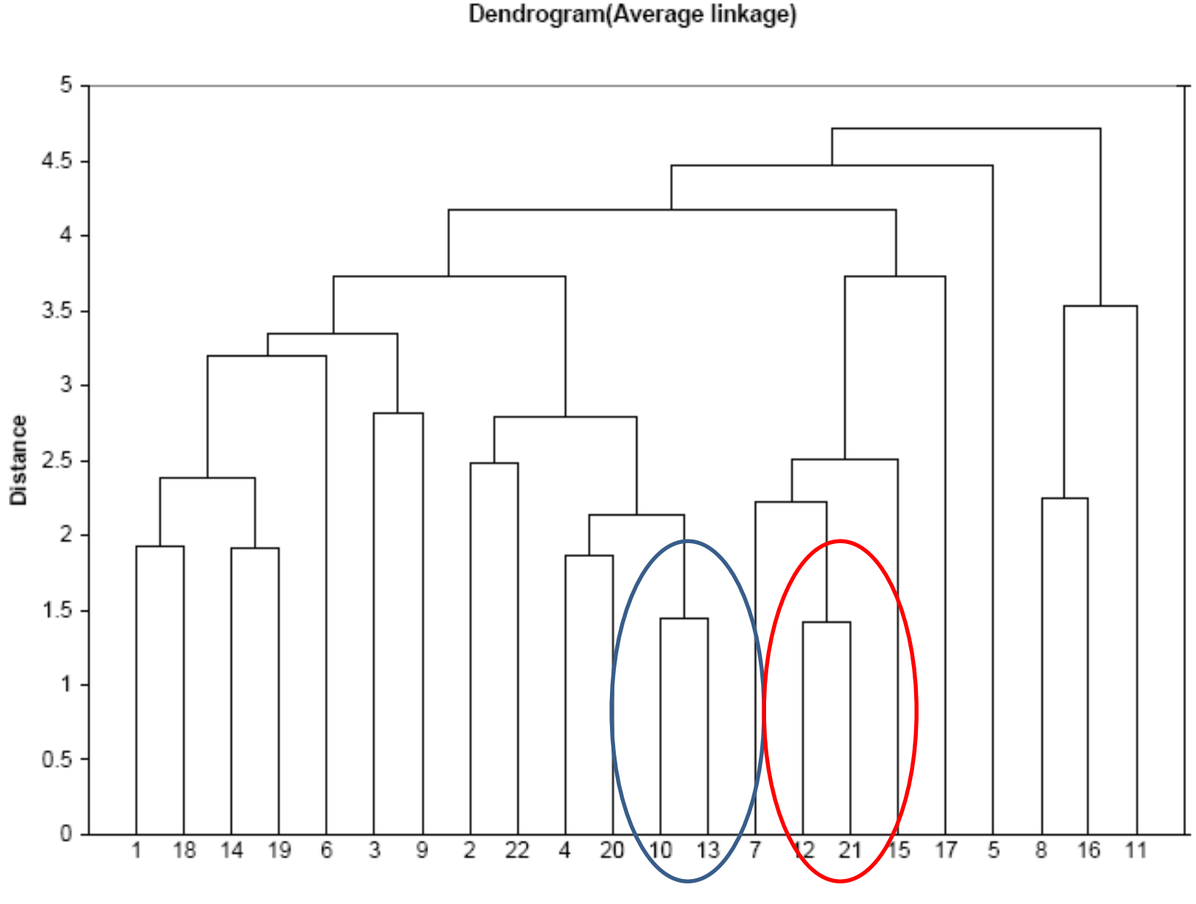
Hierarchical (agglomerative) clustering





Hierarchical (agglomerative) clustering

Reading the Dendrogram: Lines connected lower down are merged earlier



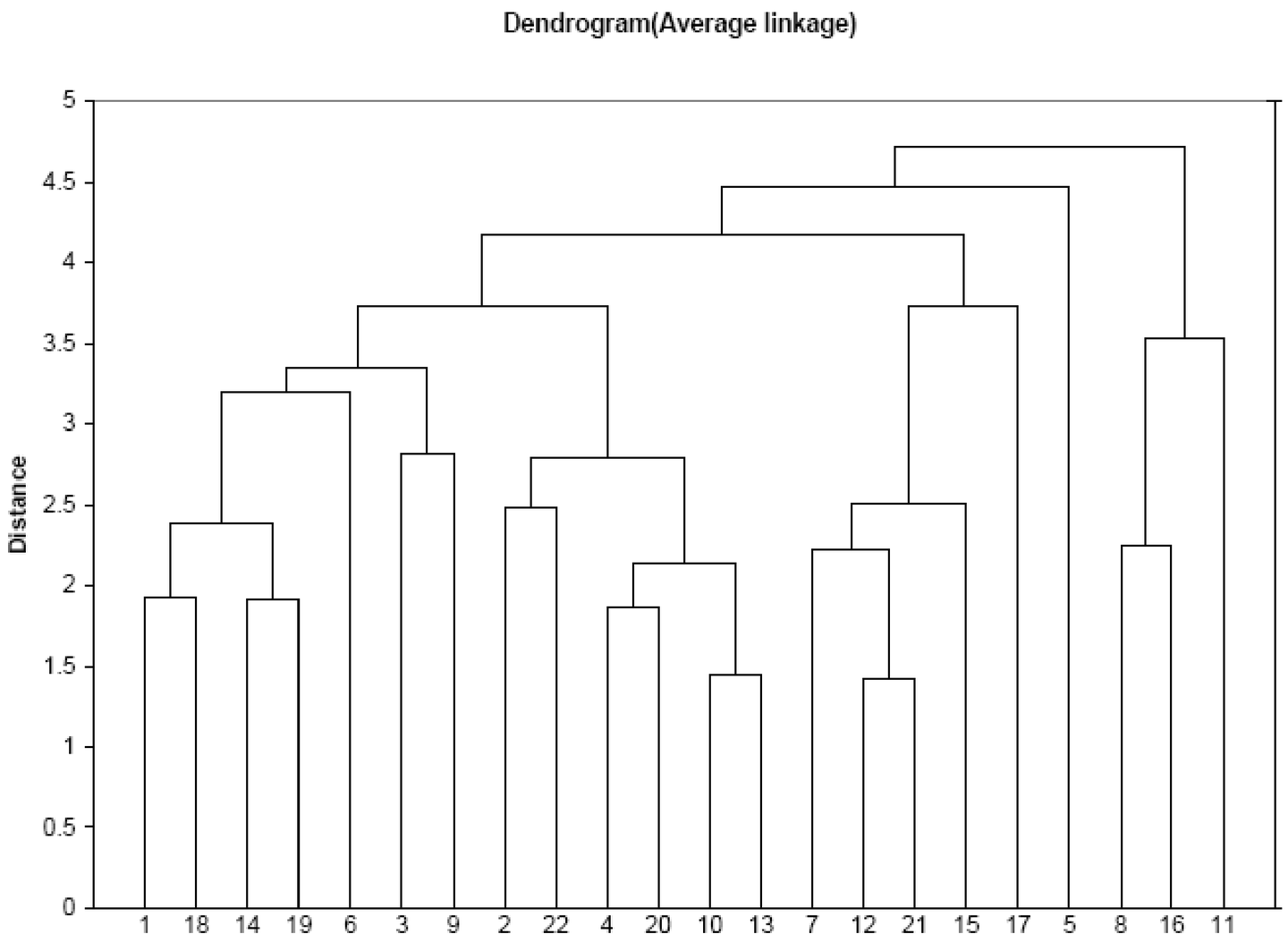
Hierarchical (agglomerative) clustering

- Determining number of clusters: For a given “distance between clusters”, a horizontal line intersects the clusters that are that far apart, to create clusters

  → E.g., at distance of 4.6 (red dashed line in next slide), data can be reduced to 2 clusters

  → At distance of 3.6 (green solid line) data can be reduced to 6 clusters

Hierarchical (agglomerative) clustering



Hierarchical (agglomerative) clustering

- Validating clusters - 얻어진 cluster가 meaningful한가?

- Cluster interpretability: cluster 해석을 위해 각 cluster의 특성을 분석
  - → 각 cluster에서 measurements의 summary statistics (mean, max, min 등)들을 계산해 봄
  - → cluster 분석에서 사용하지 않은 다른 특성(변수)들이 있는지 검토
  - → 각 cluster의 특징을 나타낼 수 있는 이름을 붙여봄E.g., at distance of 4.6 (red dashed line in next slide), data can be reduced to 2 clusters

Hierarchical (agglomerative) clustering

- Cluster stability: input을 약간만 바꾸어도 cluster assignment가 크게 달라지는가? 즉, Data를 둘로 나누어 한쪽 data에서의 clustering이 다른쪽 data에도 잘 적용되는가 검토

  → Partition A에 대해 clustering 수행

  → A의 각 cluster centroid들을 사용하여 partition B의 각 record를 assign

  → 둘로 나누지 않았을 때의 경우와 비교하여 consistent한가를 평가

Hierarchical (agglomerative) clustering

- Cluster separation: variation의 비교

  → between clusters의 variation 대 within cluster variation ratio를 검토하여 cluster separation이 적절한지 검토

Hierarchical (agglomerative) clustering

- Limitations of hierarchical clustering

  → Need huge storage for distance matrix

  → 일단 한 번 잘 못 allocated된 record는 다시 reallocate 될 수 있는 기회가 없음

  → might be unstable: reordering or dropping some records may lead to a very different results

  → single linkage나 complete linkage는 distance metric(either Euclidean or statistical distance)에 대해 robust하지만, avg linkage의 경우 크게 influenced되어 very different results

  → outlier에 sensitive

Non-Hierarchical Clustering: K-means algorithms

- k-Means Clustering

  → 미리 cluster 개수 k 를 정해두고 cluster 내(within-cluster)의 dispersion 을 최소화하면서 각 record를 이 중 하나에 assign하는 방식

  → Within-cluster dispersion = cluster centroid로부터의 distance의 합

  → k -means algorithm:

  1. Start with k initial clusters (or, randomly select k centroids)
  2. 각 record는 closest centroid가 있는 cluster에 re-assign
  3. cluster의 centroid를 다시 계산한 후 go to step 2
  4. record 이동을 하면 cluster dispersion이 증가할 경우 (또는 record 의 이동이 더 이상 없을 경우) stop

Non-Hierarchical Clustering: K-means algorithms

- Ex. five utilities and two measurements

  → 1. Initially, let k =2

  → 2. Let A = {Arizona, Boston} and B = {Central, Commonwealth, Consolidated} (임의로 나눈 것임)

→ 3. Then their centroids are:

= (-0.516.-0.020), = (-0.733,0.296)

Company	NormSales	NormFuel
Arizona Public Service	0.0459	−0.8537
Boston Edison Co.	−1.0778	0.8133
Central Louisiana Co.	0.0839	−0.0804
Commonwealth Edison Co.	−0.7017	−0.7242
Consolidated Edison Co. (NY)	−1.5814	1.6926

Non-Hierarchical Clustering: K-means algorithms

- Ex. five utilities and two measurements
  → 4. Calculate distances
  → 5. Reallocate Boston(A →B) and Central and Commonwealth(B→A)
  A={Arizona, Central, Commonwealth} and B={Boston, Consolidated}

	Distance from Centroid A	Distance from Centroid B
Arizona	1.0052	1.3887
Boston	1.0052	0.6216
Central	0.6029	0.8995
Commonwealth	0.7281	1.0207
Consolidated	2.0172	1.6341

Non-Hierarchical Clustering: K-means algorithms

- Ex. five utilities and two measurements

  → 4. Calculate distances

  → 5. Reallocate Boston(A →B) and Central and Commonwealth(B→A)

  A={Arizona, Central, Commonwealth} and B={Boston, Consolidated}

  → 현재 A={Arizona, Central, Commonwealth} and B={Boston, Consolidated}

	Distance from Centroid A	Distance from Centroid B
Arizona	1.0052	1.3887
Boston	1.0052	0.6216
Central	0.6029	0.8995
Commonwealth	0.7281	1.0207
Consolidated	2.0172	1.6341

Non-Hierarchical Clustering: K-means algorithms

- Ex. five utilities and two measurements
  - → 6. Recalculate centroids: xA = (-0.191.-0.553), xB = (-1.33,1.253)
  - → 7. New distance of each records is given by

	Distance from Centroid A	Distance from Centroid B
Arizona	0.3827	2.5159
Boston	1.6289	0.5067
Central	0.5463	1.9432
Commonwealth	0.5391	2.0745
Consolidated	2.6412	0.5067

Non-Hierarchical Clustering: K-means algorithms

- Cluster Quality

  → 22 utilities, 8 measurements, 6 case

Cluster centers

Cluster	Fixed	RoR	Cost	Load_factor	Demand	Sales	Nuclear	Fuel
Cluster-1	1.112	11.480001	177.200001	55.380002	3.76	7487.399702	38.280034	0.7716
Cluster-2	0.755001	6.949994	154.500005	56.700001	7.749996	11577.49951	4.149999	1.344
Cluster-3	1.2	10.7	221.666487	57.800002	6.566652	12493.01588	-0.000008	0.597
Cluster-4	1.185	12.400001	120.833197	54.650001	0.799999	10456.00045	3.750008	0.8765
Cluster-5	1.49	8.8	192.000002	51.20002	0.999999	3300.012277	15.600001	2.044
Cluster-6	1.048	9.920001	184.600002	62.14001	2.300001	6400.400459	5.240004	1.724

Non-Hierarchical Clustering: K-means algorithms

- Cluster Quality

  → Centroid 간의 distance

Distance between cluster centers	Cluster-1	Cluster-2	Cluster-3	Cluster-4	Cluster-5	Cluster-6
Cluster-1	0	4090.309917	5005.961483	2969.338323	4187.479063	1087.549967
Cluster-2	4090.309917	0	917.995763	1122.041163	8277.584943	5177.193266
Cluster-3	5005.961483	917.995763	0	2039.52432	9193.06908	6092.733626
Cluster-4	2969.338323	1122.041163	2039.52432	0	7156.353693	4056.109579
Cluster-5	4187.479063	8277.584943	9193.06908	7156.353693	0	3100.434146
Cluster-6	1087.549967	5177.193266	6092.733626	4056.109579	3100.434146	0

Non-Hierarchical Clustering: K-means algorithms

- within-cluster dispersion

Data summary (In Original coordinates)

Cluster	#Obs	Average distance in cluster
Cluster-1	5	1042.936117
Cluster-2	2	5863.533146
Cluster-3	3	2724.981548
Cluster-4	6	1241.097807
Cluster-5	1	0.012277017
Cluster-6	5	624.4372161
Overall	22	1622.067124

Non-Hierarchical Clustering: K-means algorithms

- ESS에 의한 cluster validation

  → For a given k (> 1),

  → Ratio=

  Sum of squared distances (k>1: multiple clusters)

  Sum of squared distances to the mean (k=1: all in one cluster)

→ If ratio ≈ 1, clustering is not effective.

→ If ratio ≈ 0, well-separated groups.

Non-Hierarchical Clustering: K-means algorithms

- Choosing k

  → 초기 record의 allocate: 사전지식이 있을 경우 이를 사용하지만 no info 인 경우는?

What other unsupervised learnings exist

- Unsupervised Learning
  - → K-means clustering
  - → Gaussian Mixture
  - → Isolation Forest
  - → Autoencoders, GANs
  - → Expectation-Maximization Algorithms