

밑바닥부터 시작하는 데이터 과학



4회차

- 차원 축소
- 기계학습 기초
- k-NN
- 나이브베이지스
- 자연어처리



1. 차원 축소

차원 축소 (Dimensionality Reduction)

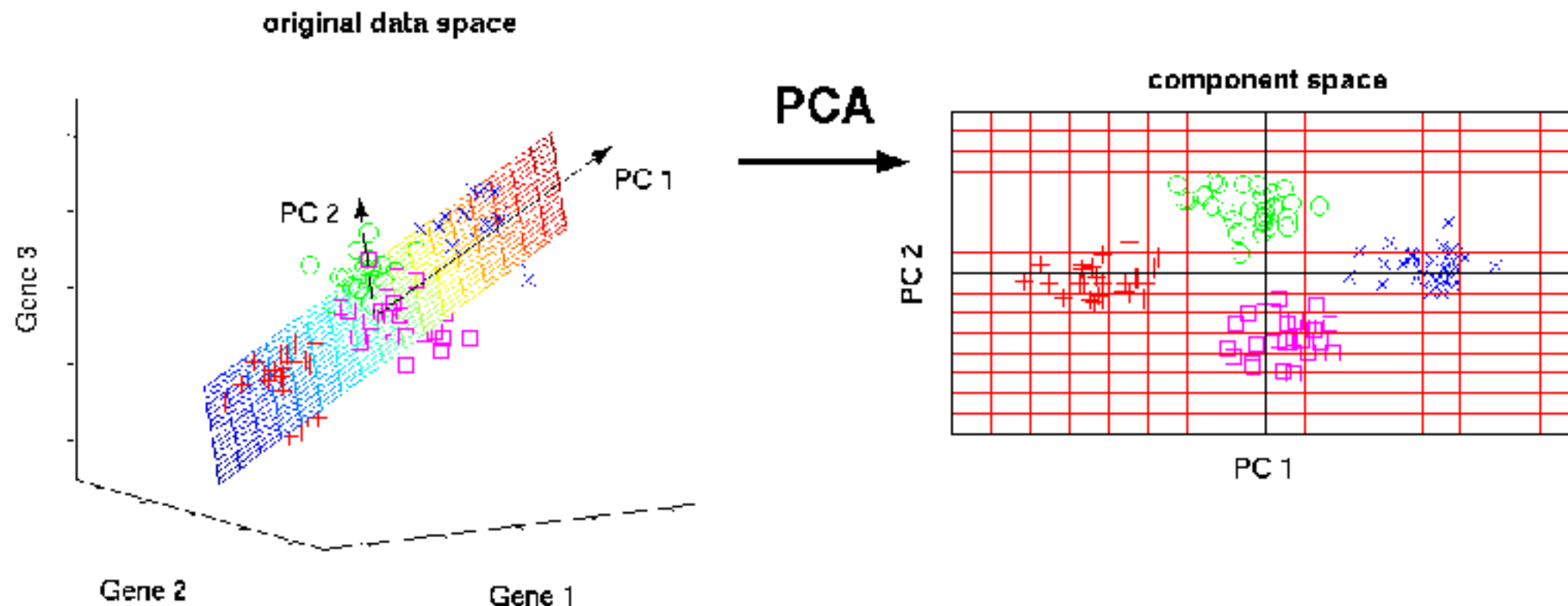
데이터의 의미를 제대로 표현하는 특징을 추려내는 것

1. 차원 축소

주성분 분석 (PCA)

주성분분석은 데이터의 분포를 가장 잘 표현하는 성분을 찾아주는 것이다.

- 실제 우리가 관찰한 특징들 (data space)중 에서 숨겨져있는 진짜 특징 (latent space)를 찾는 것.
 - 데이터는 'linear'하다고 가정.
 - 서로 직교하는 좌표들 간의 조합으로 만들어진다. (즉, 가장 주요한 두 성분은 직교한다)



1. 차원축소

주성분 분석 (PCA)

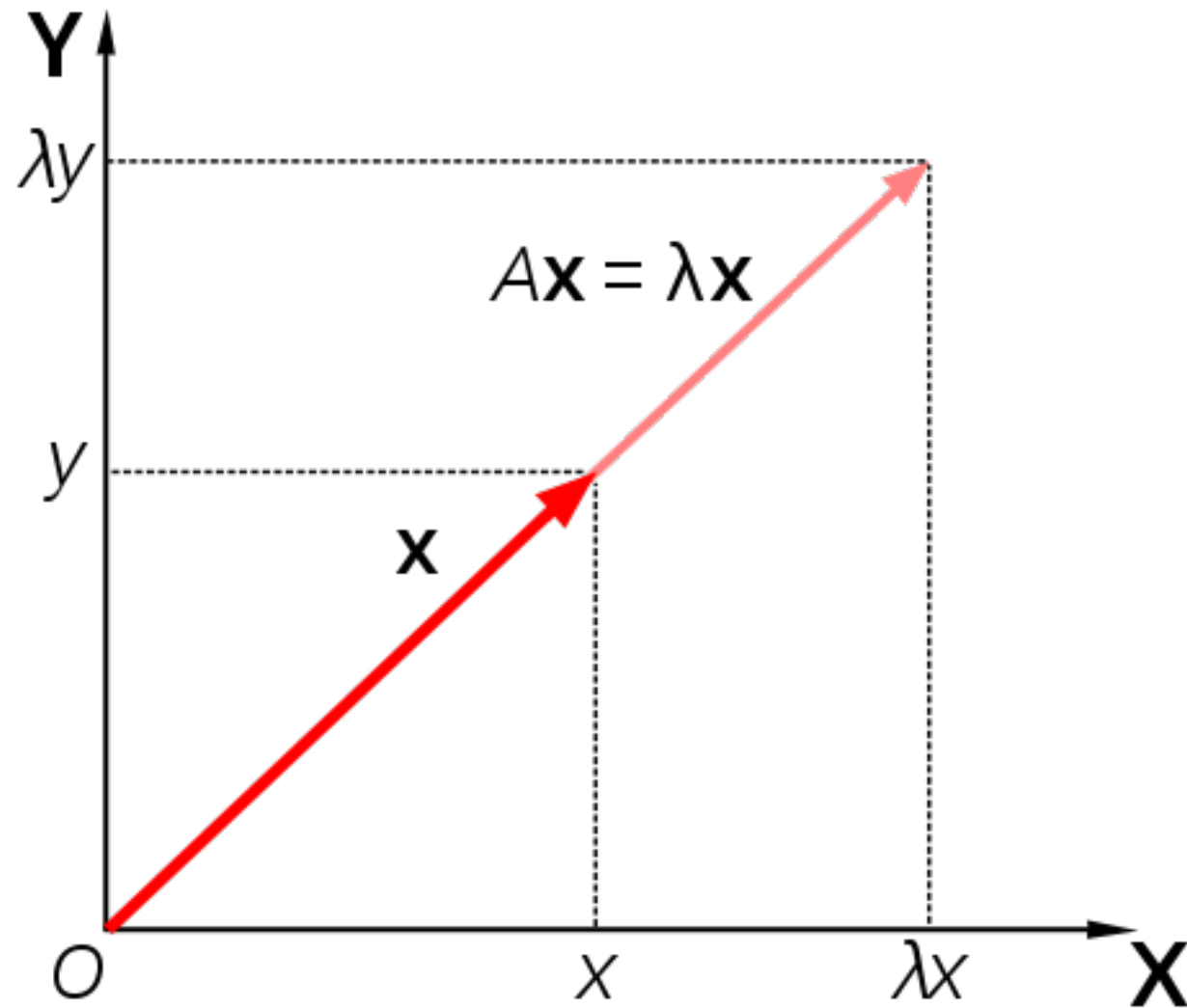
주성분분석은 데이터의 분포를 **가장 잘 표현하는 성분**은 **분산이 가장 큰 벡터**이다.

분산이 가장 큰 벡터를 찾기 위해서는 **공분산 행렬**, **eigen value**, **eigen vector**에 대해 알아야 한다.

- **공분산 행렬** : 벡터의 각 차원 간의 공분산을 나타낸 행렬
- **eigen vector (고유벡터)** : 어떤 행렬(A)와의 선형 변환의 결과가 자기 자신의 상수배가 되는 0이 아닌 벡터
- **eigen value (고유값)** : 그때의 상수배
 - $Ax = \lambda x$ 가 성립할 때, (A = 정방행렬 ($n \times n$))
 - λ : A의 eigen value
 - x : λ 에 대응되는 eigen vector
 - x 와 λ 는 최대 n 개까지 존재할 수 있다.
 - **기하학적 의미**
 - 행렬 A의 eigen vector는 선형 변환 A에 의해 방향은 보존되고 스케일 (scale)만 변화되는 방향 벡터를 나타내고, eigen value은 그 eigen vector의 변화되는 스케일 정도를 나타내는 값이다.

1. 차원축소

주성분 분석 (PCA)



- 람다 : A 의 eigen value
- x : 람다에 대응하는 eigen vector

1. 차원 축소

주성분 분석 (PCA)

PCA 계산

- 데이터 : $z_i = (x_i, \dots, x_p)$ $i = 1, \dots, n$
- w 에 대해 z_i 를 프로젝트 시킨 h_i 가 있을 때, PCA는
- h_i 의 크기를 최대화 시킬 수 있는 w 를 찾는 문제와 동일해진다.

분산 계산

- $z_i \cdot w$ 의 분산을 σ_w^2 라고 하자.

$$\sigma_w^2 = \frac{1}{n} \sum_i (z_i \cdot w)^2 - \left(\frac{1}{n} \sum_i (z_i \cdot w) \right)^2$$

$$= \frac{1}{n} \sum_i (z_i \cdot w)^2$$

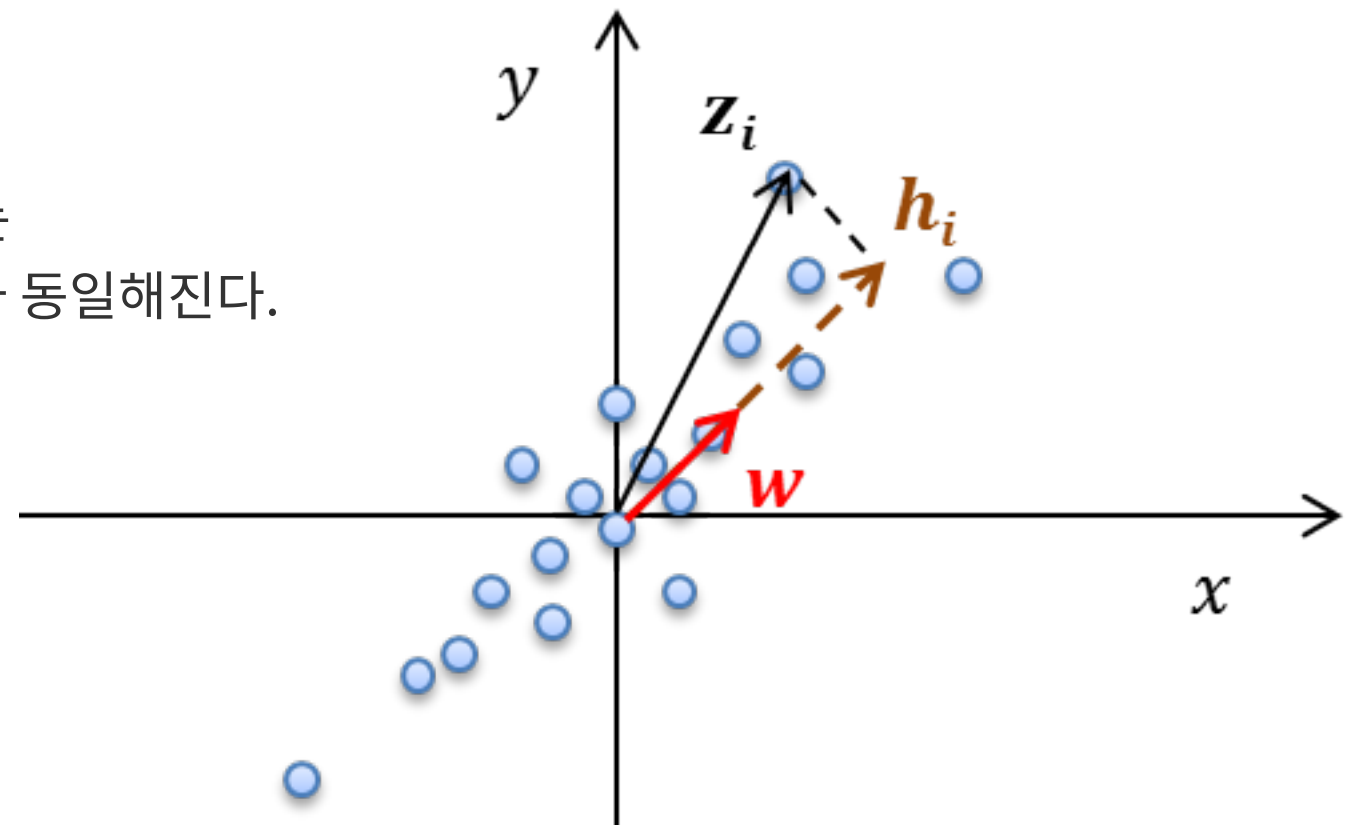
$$= \frac{1}{n} (Zw)^T (Zw)$$

$$= \frac{1}{n} w^T Z^T Z w$$

$$= w^T \frac{Z^T Z}{n} w$$

$$= w^T C w$$

- z_i 들의 평균이 0이 되도록 centering을 한 후라고 생각하면,
- Z/n 으로 잡은 C 는 z_i 들의 공분산 행렬이 된다.



$$C = \begin{pmatrix} \text{cov}(x,x) & \text{cov}(x,y) \\ \text{cov}(x,y) & \text{cov}(y,y) \end{pmatrix} = \begin{pmatrix} \frac{1}{n} \sum (x_i - m_x)^2 & \frac{1}{n} \sum (x_i - m_x)(y_i - m_y) \\ \frac{1}{n} \sum (x_i - m_x)(y_i - m_y) & \frac{1}{n} \sum (y_i - m_y)^2 \end{pmatrix}$$

1. 차원축소

주성분 분석 (PCA)

분산을 최대화 시키는 w 계산

- 구하고자 하는 문제는 w 가 단위벡터($w^T w = 1$, 길이가 1인 벡터)라는 조건을 만족하면서 $w^T C w$ 를 최대로 하는 w 를 구하는 constrained optimization 문제로 볼 수 있으며 Lagrange multiplier λ 를 도입하여 다음과 같이 최적화 문제로 식을 세울 수 있다.

$$u = w^T C w - \lambda (w^T w - 1)$$

- 이 때, u 를 최대로 하는 w 는 u 를 w 로 편미분한 $\partial u / \partial w$ 를 0으로 하는 값이다.

$$\frac{\partial u}{\partial w} = 2Cw - 2\lambda w = 0$$
$$Cw = \lambda w$$

- 즉, z_i 에 대한 공분산 행렬 C 의 eigenvector가 z_i 의 분산을 최대로 하는 방향벡터임을 알 수 있다. 또한 여기서 구한 w 를 분산 식에 대입하면 $\sigma_{w^2} = w^T \lambda w = \lambda$ 가 되므로 w 에 대응하는 eigenvalue λ 가 w 방향으로의 분산의 크기임을 알 수 있다.

2. 기계학습 기초

기계학습 (Machine Learning)

2. 기계학습 기초

기계학습 기초

기계 학습의 방법

- 지도 학습 (Supervised Learning)
 - 학습에 사용될 데이터에 정답이 포함되어 있는 경우
- 비지도 학습 (Unsupervised Learning)
 - 학습에 사용될 데이터에 정답이 포함되어 있지 않은 경우
- 준 지도 학습 (Semi - supervised Learning)
 - 데이터의 일부에만 정답이 포함되어 있는 경우
- 온라인 학습 (online learning)
 - 새로 들어오는 데이터를 통해 모델을 끊임없이 조정하는 경우

모델 (Model)

- 다양한 변수간의 수학적 혹은 확률적 관계를 표현

하이퍼파라미터 (Hyperparameter)

- 기계학습 모델의 parameter
 - 베이저안 통계에서의 하이퍼파라미터와는 전혀 다르다.

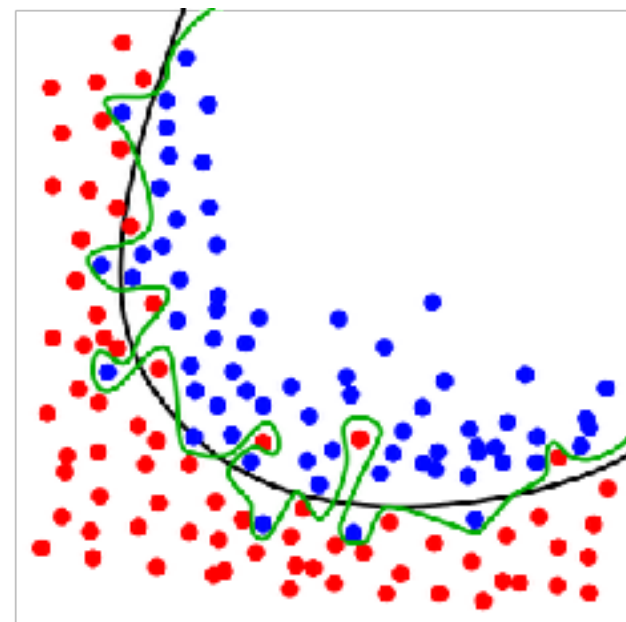


2. 기계학습 기초

기계학습 기초

오버피팅 문제

- 주어진 데이터에만 과하게 적합한 (overfitting) 모델이 만들어지는 문제
 - 새로 관찰된 데이터에 대해서는 전혀 맞지 않는 모델이 되어버린다.
- 이를 해소하기 위해서는 **학습용(train) 데이터**와 **검증용(test) 데이터**를 분리해서 모델링을 한다. (**validation** 이라고 한다)



정확도

- 기계학습 방법론에 따라 모델의 성능을 평가하기 위한 평가 지표가 주어진다.
- 그 중, 분류모델은 어떤 문제에 관한 모델인지에 따라 다른 방식으로 평가하는게 용이하다.
- **일반적으로 알려진 정확도 (Accuracy)**
 - 정답을 맞춘 개수 / 전체 개수
 - 예시) 암 진단 모델 (전체 환자 100명 중, 암 환자 2명 인 경우)
 - 모든 환자를 다 암환자가 아니라고 판단할 경우, 정확도는 98%가 된다.

2. 기계학습 기초

기계학습 기초

분류모델의 성능 평가

- Confusion matrix

- 양성인데, 양성으로 제대로 검출된것은 True Positive (TP)
- 음성인데 음성으로 제대로 검출된것은 True Negative (TN)
- 양성인데 음성으로 잘못 검출된것은 False Negative (FN)
 - 실제로는 암 환자인데, 정상으로 판별한 경우
- 음성인데 양성으로 잘못 검출된것은 False Positive (FP)
 - 정상인데 암환자로 판별한 경우

| | | Predicted | | |
|----------|----------|-----------|----------|---|
| | | Positive | Negative | |
| Observed | Positive | TP | FN | P |
| | Negative | FP | TN | N |

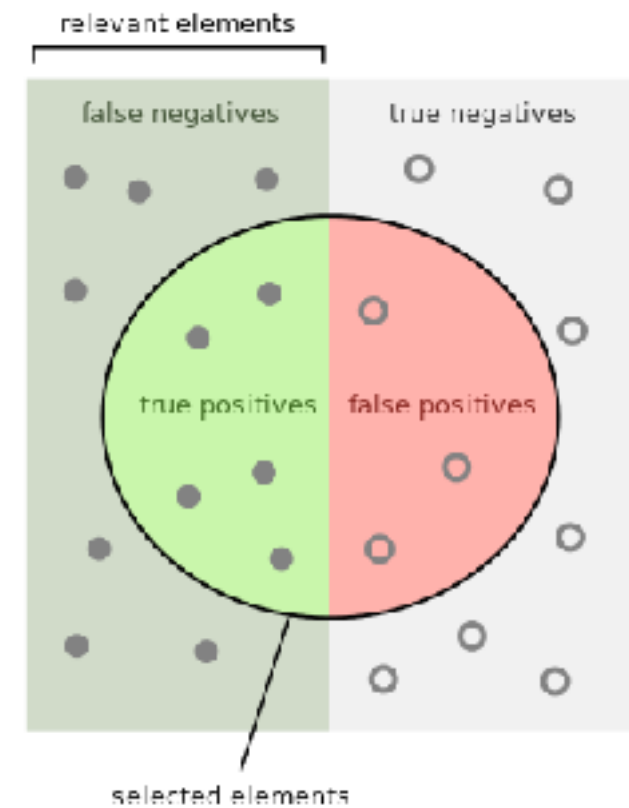
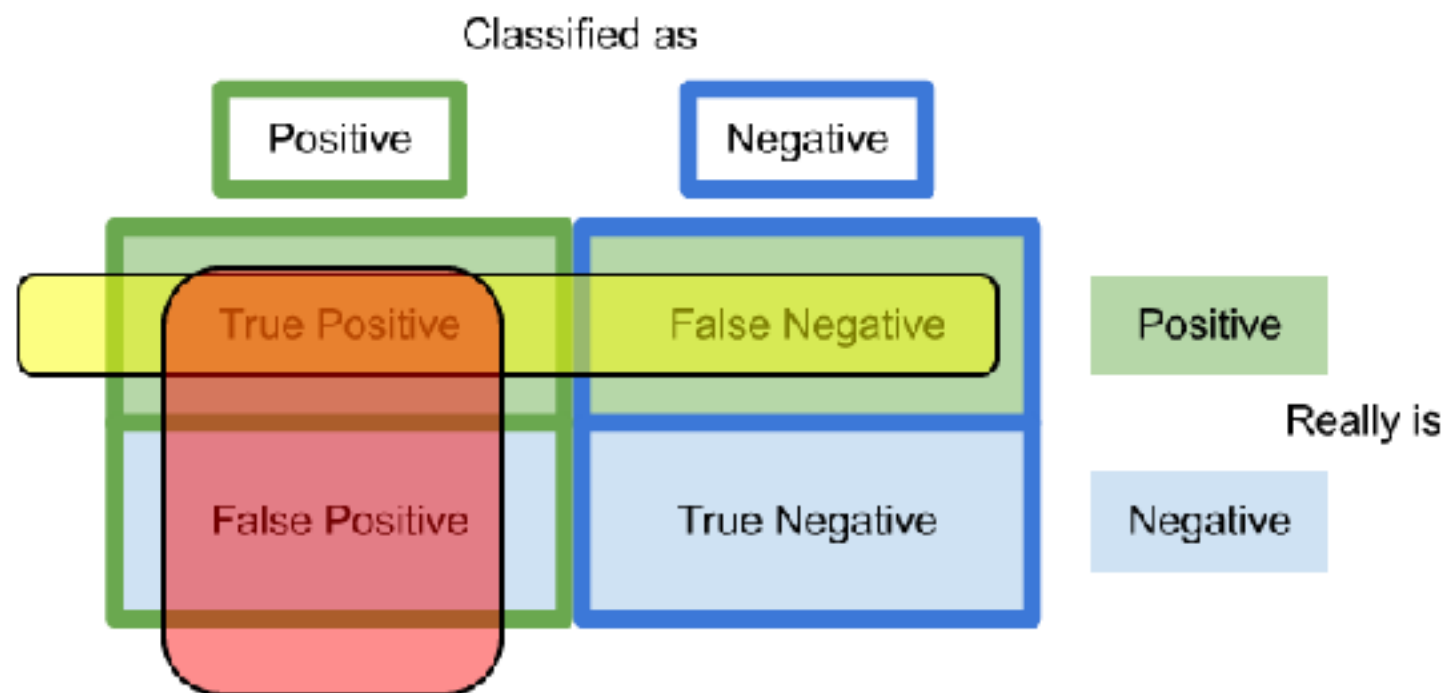
- Recall 과 Precision

- **Accuracy** : $(\text{True positive} + \text{True negative}) / (\text{TP} + \text{FN} + \text{FP} + \text{TN})$
- **Recall** : $\text{True positive} / (\text{True positive} + \text{False negative})$
 - 암 환자인데 암 환자라 맞춘 수(TP) / (암 환자인데 암 환자라 맞춘 수(TP) + 암 환자인데 암 환자가 아니라고 맞춘 수(FN))
 - 예시의 경우에는 recall이 0%가 된다. $(0 / (0 + 2))$
- **Precision** : $\text{True positive} / (\text{True positive} + \text{False positive})$
 - 암 환자인데 암 환자라 맞춘 수(TP) / (암 환자인데 암 환자라 맞춘 수(TP) + 암 환자 아닌데 암 환자라고 맞춘 수(FP))
 - 예시의 경우에는 precision은 계산 불가 $(0 / (0 + 0))$.

2. 기계학습 기초

기계학습 기초

분류모델의 성능 평가



How many selected items are relevant?

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

3. KNN (K- nearest neighbor)

K-NN (K- nearest neighbor)

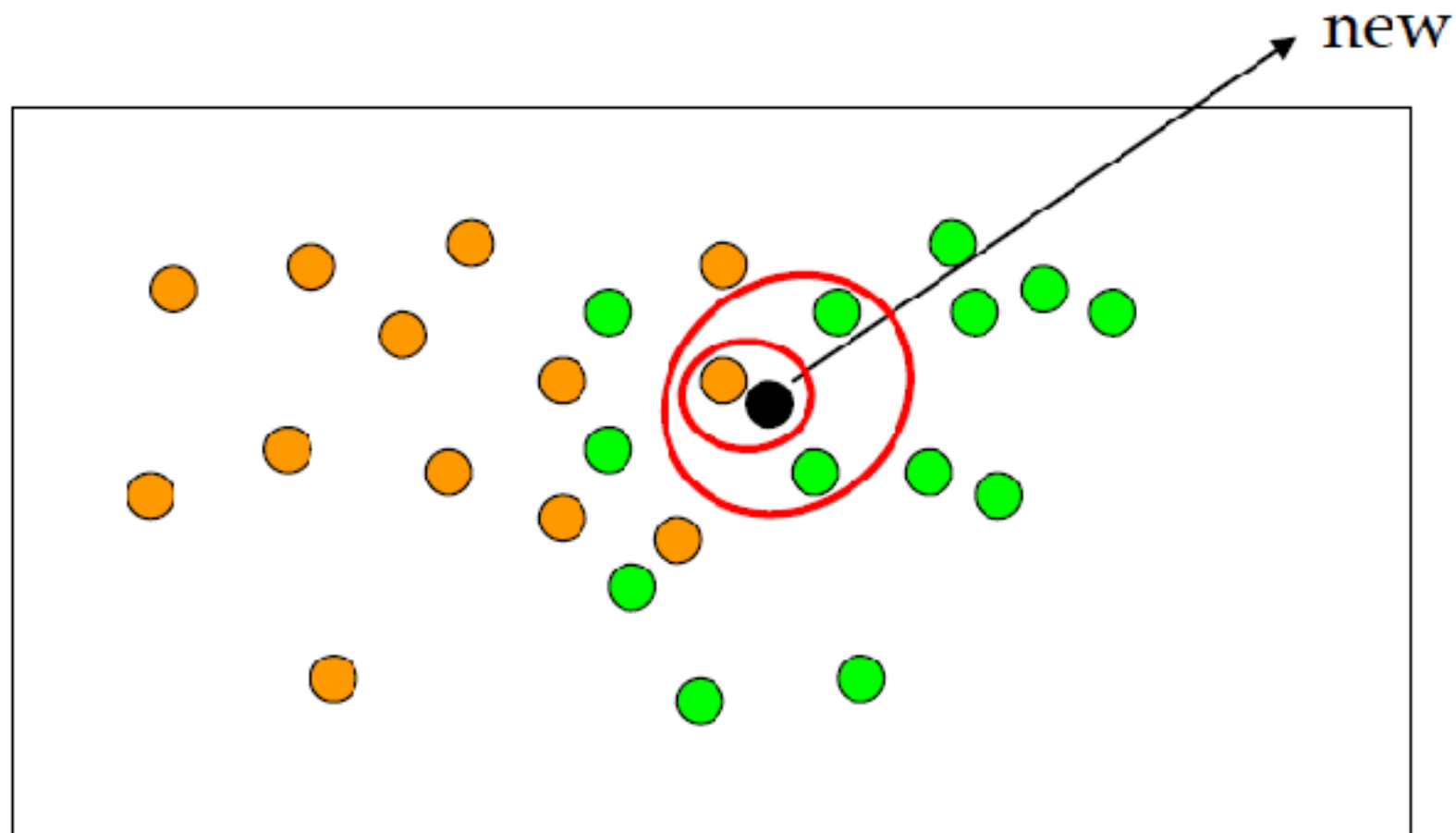
새로운 데이터가 주어졌을 때 기존 데이터 가운데 가장 가까운 k개 이웃의 정보로 새로운 데이터를 예측하는 방법론

3. KNN (K- nearest neighbor)

K-NN

수학적인 가정없이 새로운 데이터가 들어왔을 때, k개의 가장 가까운 (=이웃) 기존 데이터로 새로운 데이터의 결과값을 추론한다. (k -means와는 완전히 다른 모델임)

- k = 1 이면 new = 오렌지색
 - k = 3 이면 new = 녹색
- 아주 간단한 모델이며 분류 (classification), 회귀 (regression)에 모두 적용 가능함

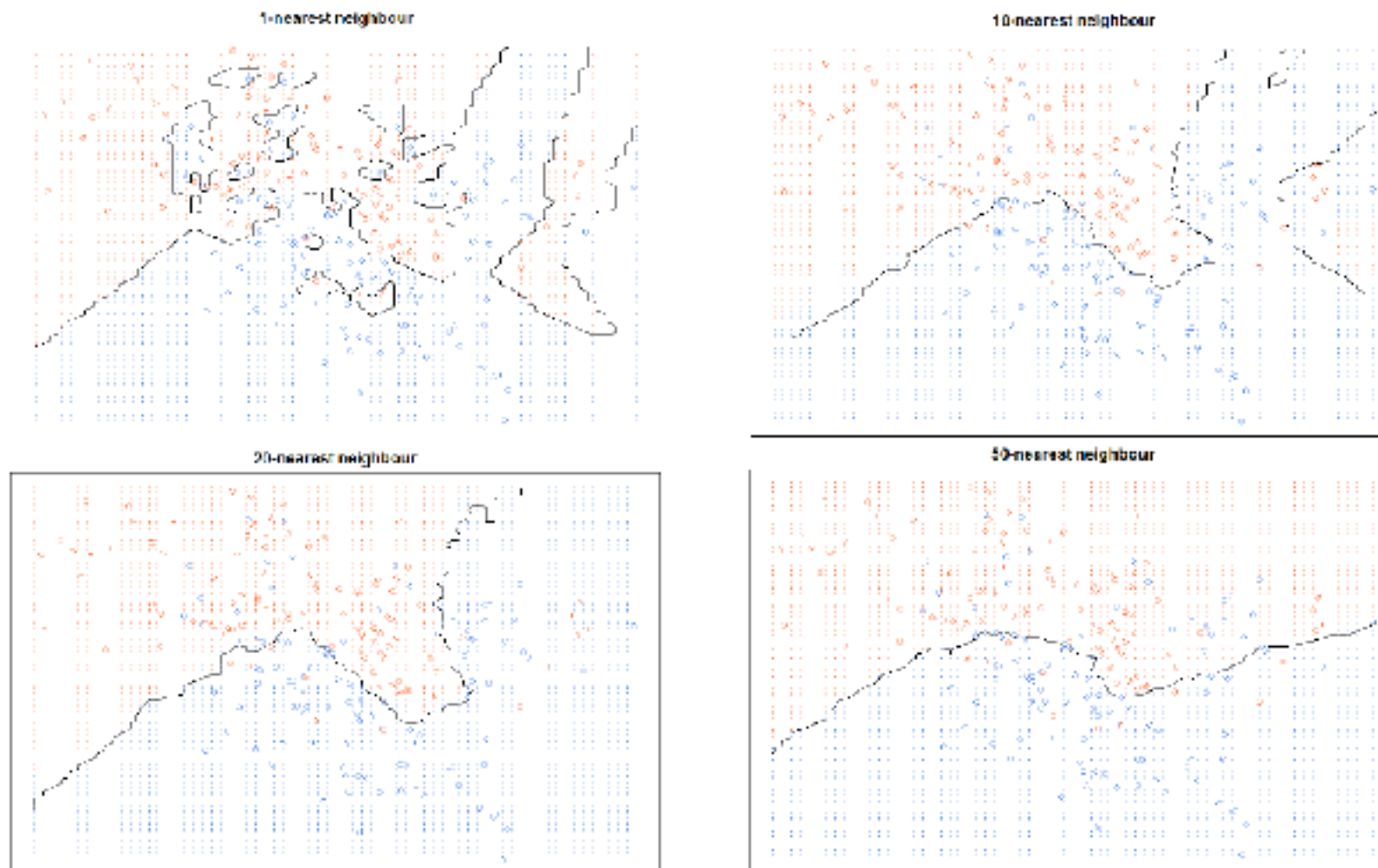


3. KNN (K- nearest neighbor)

K-NN

하이퍼파라미터 : k (이웃), 거리 측정 방법

- k가 지나치게 작을 경우 : 지역적특성을 지나치게 반영함 (overfitting)
- k가 지나치게 클 경우 : 과하게 정규화한다 (underfitting)

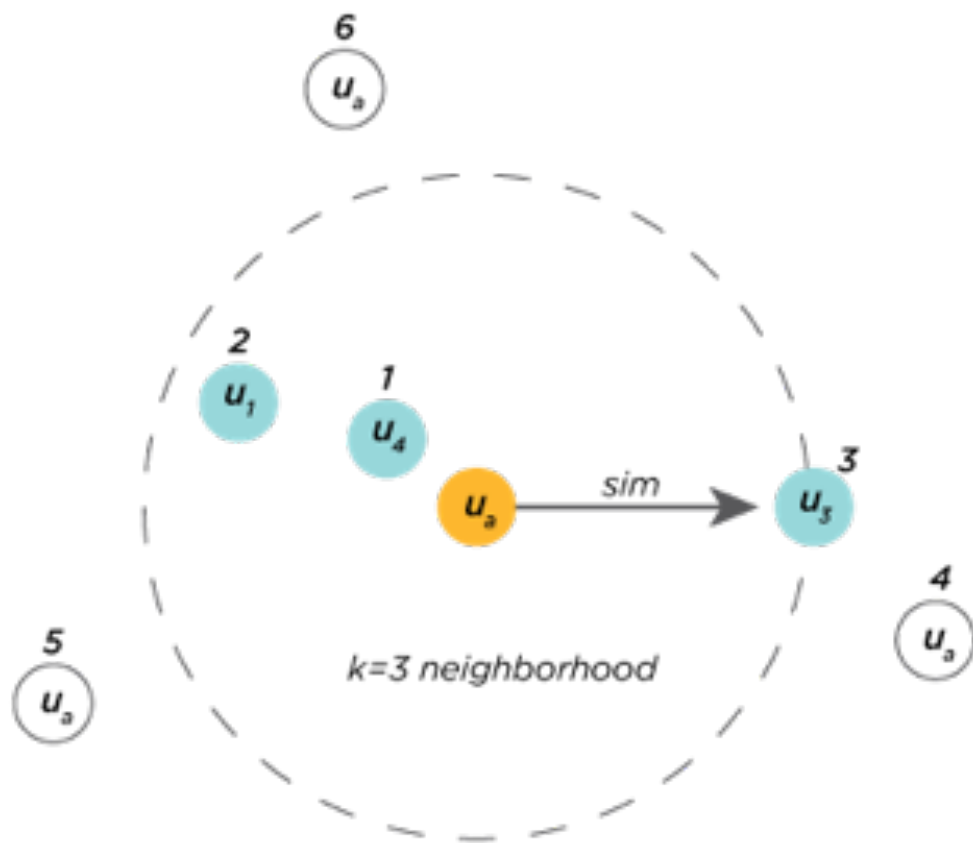


3. KNN (K- nearest neighbor)

K-NN

활용 예시 (추천 시스템)

- User Based Collaborative Filtering
- 거리상으로 가까운 유저의 item 정보를 바탕으로 추천



| | i_1 | i_2 | i_3 | i_4 | i_5 | i_6 |
|-------|-------|-------|-------|-------|-------|-------|
| u_3 | ? | ? | 4.0 | 3.0 | ? | ? |
| u_1 | ? | 4.0 | 4.0 | 2.0 | 1.0 | 2.0 |
| u_2 | 3.0 | ? | ? | ? | 5.0 | 1.0 |
| u_5 | 3.0 | ? | ? | 3.0 | 2.0 | 2.0 |
| u_4 | 4.0 | ? | ? | 2.0 | 1.0 | 1.0 |
| u_5 | 1.0 | 1.0 | ? | ? | ? | ? |
| u_6 | ? | 1.0 | ? | ? | 1.0 | 1.0 |
| | 3.5 | 4.0 | | | 1.3 | |

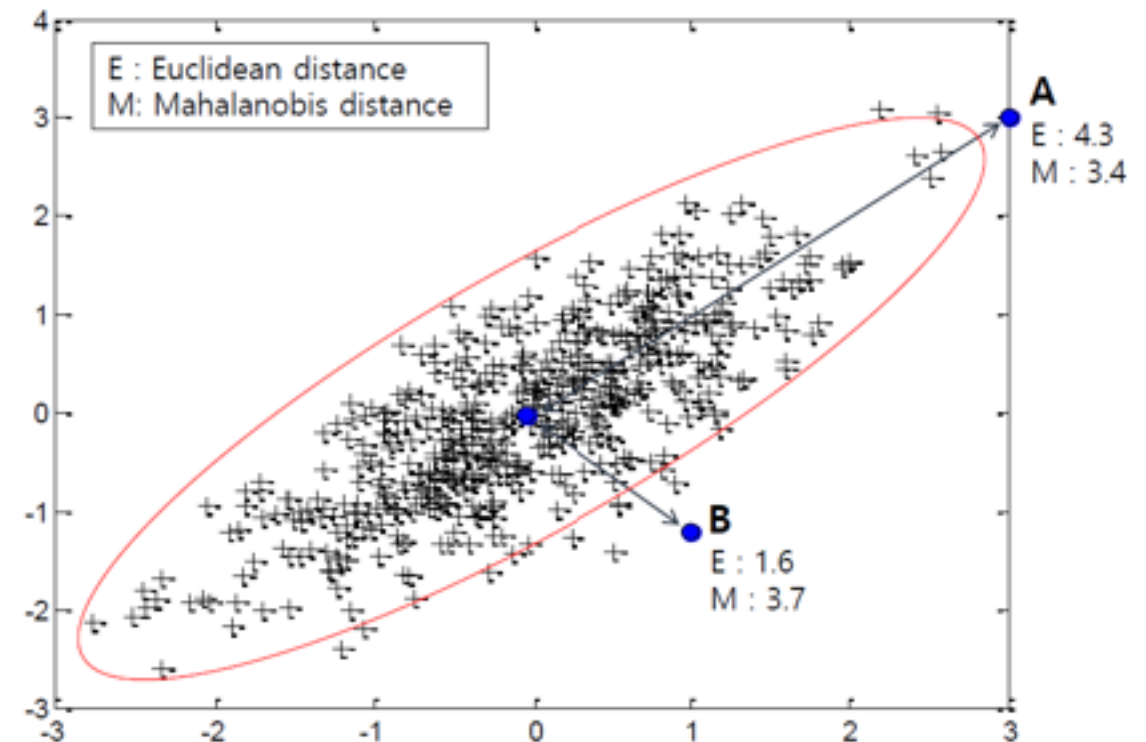
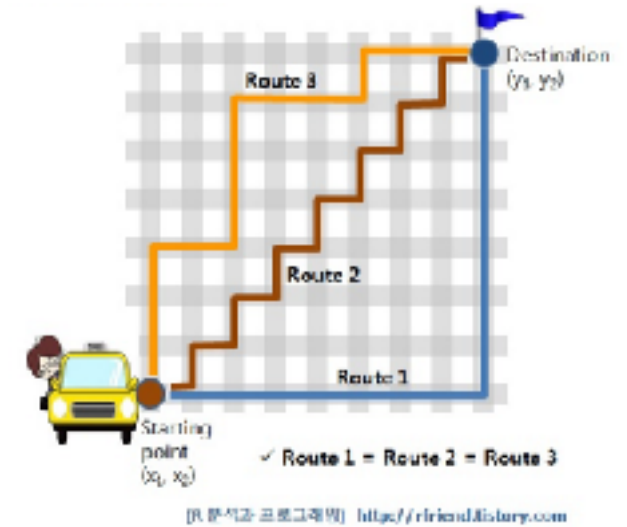
Recommendations: i_2, i_1

3. KNN (K- nearest neighbor)

K-NN

거리 지표

- Euclidean Distance
 - 두 관측치의 최단 거리를 의미
- Manhattan Distance
 - A에서 B로 이동할 때 각 좌표축 방향으로만 이동할 경우에 계산되는 거리
- Mahalanobis Distance
 - 변수 내 분산, 변수간 공분산을 모두 반영하여 거리를 계산하는 방식



4. Naive Bayes

Naive Bayes

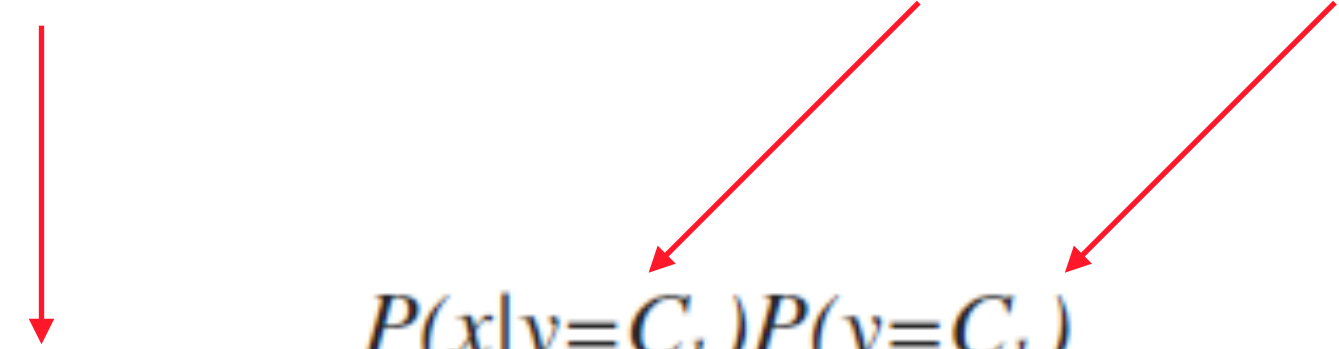
조건부 독립이라는 간단한(naive) 가정을 통해 데이터를 분류하는 방법론

4. Naive Bayes

Naive Bayes

나이브 베이즈 모델은 베이즈 정리를 이용하여, 분류하고자 하는 대상의 각 분류별 확률을 측정하여, 그 확률이 큰 쪽으로 분류하는 방법을 취한다.

- 분류하고자 하는 대상의 확률(사후 확률, posteriori)을 계산하기 위해 likelihood, 사전확률 (prior)이 필요하다.

$$P(y=C_k|x) = \frac{P(x|y=C_k)P(y=C_k)}{P(x)}$$


4. Naive Bayes

Naive Bayes

| 사전 확률 계산

- 주어진 데이터를 바탕으로 전체 데이터에서 각 클래스별 비율을 계산한다.
 - 전체 데이터 : 100
 - 클래스1 : 10
 - $P(\text{클래스1}) = 10/100$
 - 클래스2 : 90
 - $P(\text{클래스2}) = 90/100$

| Likelihood 계산

- 모든 x 가 상호 독립이라는 가정하에 각 클래스에 대한 조건부 확률을 구한다.
 - **likelihood 확률 분포**
 - 베르누이 분포
 - 다항분포
 - 정규분포
 - x 가 연속변수일 때는 정규분포, 여러 개의 값을 가진 카테고리값인 경우 (ex. 단어) 다항분포를 가정한다.

$$P(x|y=C_k) = \prod_{i=1}^P P(x_i | y=C_k)$$

5. 자연어처리 소개

자연어 처리 (Natural Language Processing)

- 인간이 발화하는 언어 현상을 기계적으로 분석해서 **컴퓨터**가 이해할 수 있는 형태로 만드는 **자연 언어** 이해 혹은 그러한 형태를 다시 인간이 이해할 수 있는 언어로 표현하는 제반 기술을 의미

5. 자연어처리 소개

자연어 처리

데이터 과학과 자연어 처리

- 인터넷 사용이 급증하면서 문서의 생성자가 폭발적으로 증가함.
- 인터넷 정보를 효율적으로 수집/활용하고자 자연어처리 연구가 활발히 진행됨
- 최근에는 딥러닝 기술이 급격히 발달하면서 자연어 처리를 넘어 자연어를 이해 (NLU)하려는 노력이 시도되고있다.

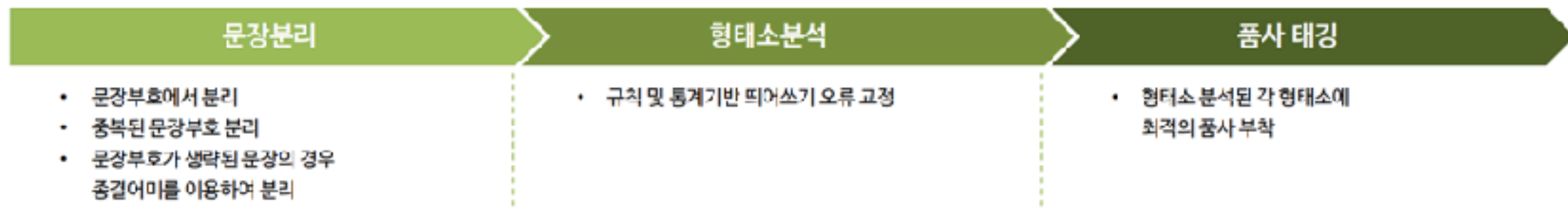
자연어 처리의 일반적인 과정

- 문장분리
- 형태소 분석
- 품사 태깅
- 패러프레이징
- 구문단위화
- 구문분석

5. 자연어처리 소개

자연어 처리

| 자연어 처리의 일반적인 과정

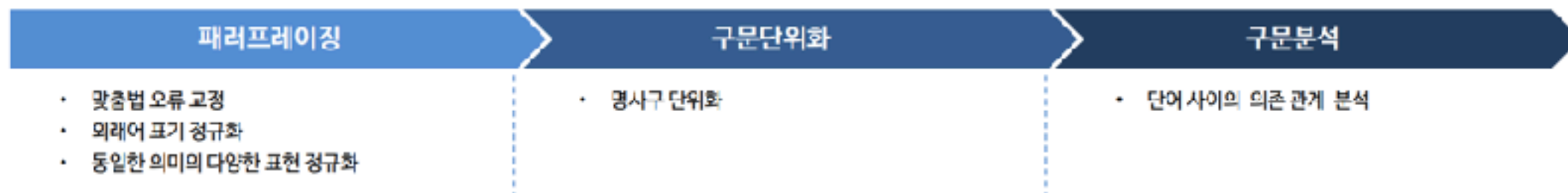


| 기능 | 목적 | 예시 |
|--------|-----------------------------------|--|
| 문장 분리 | 전체 텍스트를 문장부호(!,?) 기준으로 분리 | 문장1 : 내가 아이폰을 샀는데, 디자인은 맘에 드는데 카메라가 영 좋지 않네요. 문장2 : 하지만 아이폰이 갤럭시보다 디자인, 그림감, 색감 등이 좋네요. |
| 형태소 분석 | 분석 가능한 모든 형태소 조합을 분석 | <div>내가</div> <div>아이폰을</div> <div>샀는데,</div> <div>디자인은</div> <div>맘에</div> <div>드는데</div> <div>카메라가</div> <div>영</div> <div>좋지</div> <div>않네요</div> <div>내/p+가/j</div> <div>아이폰/n+을/j</div> <div>사/v+았/pe+는데/e+./cm</div> <div>디자인/n+은/j</div> <div>맘/n+에/j</div> <div>내/n+가/j</div> <div>내가/v+아/e</div> <div>들/v+는데/e</div> <div>카메라/n+가/j</div> <div>영/ad</div> <div>영/n</div> <div>좋/aj+지/e</div> <div>않/x+네요/e</div> <div>않/aj+네요/e</div> |
| 품사 태깅 | 분석가능한 형태소 조합들 중 가장 분석에 적합한 품사로 태깅 | <div>내가</div> <div>아이폰을</div> <div>샀는데,</div> <div>디자인은</div> <div>맘에</div> <div>드는데</div> <div>카메라가</div> <div>영</div> <div>좋지</div> <div>않네요</div> <div>내/p+가/j</div> <div>아이폰/n+을/j</div> <div>사/v+았/pe+는데/e+./cm</div> <div>디자인/n+은/j</div> <div>맘/n+에/j</div> <div>들/v+는데/e</div> <div>카메라/n+가/j</div> <div>영/ad</div> <div>영/n</div> <div>좋/aj+지/e</div> <div>않/x+네요/e</div> |

5. 자연어처리 소개

자연어 처리

자연어 처리의 일반적인 과정



| 기능 | 목적 | 예시 |
|--------|---|--|
| 패러프레이징 | 정규화 사전을 기반으로 분석이 어려운 표현을 분석하기 쉬운 표현으로 정규화 | <div>내가</div> <div>내/p+가/j</div> <div>아이폰을</div> <div>아이폰/n+을/j</div> <div>샀는데,</div> <div>사/v+았/pe+는데/e+,/cm</div> <div>디자인은</div> <div>디자인/n+은/j</div> <div>맘에</div> <div>맘/n+에/j</div> <div>좋</div> <div>좋/aj+은데/e</div> <div>드는데</div> <div>들/v+는데/e</div> |
| 구문단위화 | 명사구를 하나의 분석단위로 처리 (e.g. 좋은제품) | <div>내가</div> <div>[내/p+가/j]</div> <div>아이폰을</div> <div>[아이폰/n+을/j]</div> <div>샀는데,</div> <div>[사/v+았/pe+는데/e+,/cm]</div> <div>디자인은</div> <div>[디자인/n+은/j]</div> <div>맘에</div> <div>[좋/aj+은데/e]</div> <div>드는데</div> <div></div> |
| 구문분석 | 하나의 문장을 단어 사이의 관계를 통해 구문단위로 분석 | <div>내가</div> <div>[내/p+가/j]</div> <div>아이폰을</div> <div>[아이폰/n+을/j]</div> <div>샀는데,</div> <div>[사/v+았/pe+는데/e+,/cm]</div> <div>디자인은</div> <div>[디자인/n+은/j]</div> <div>맘에</div> <div>[좋/aj+은데/e]</div> <div>드는데</div> <div></div> |

* p:대명사, j:조사, n:명사, v:동사, ad:부사, aj:형용사, x:보조동사, pe:선어말어미, e:이미, cm:임포

5. 자연어처리 소개

자연어 처리

파이썬과 자연어처리

- **Konlpy**
 - 한글 형태소 분석기들이 패키징되어있다.
- **Gensim**
 - 자연어 처리분석 알고리즘들의 모음
 - C++ 로 개발되어 처리속도가 매우 빠르다.



다음에
또!
같이!
만나요!