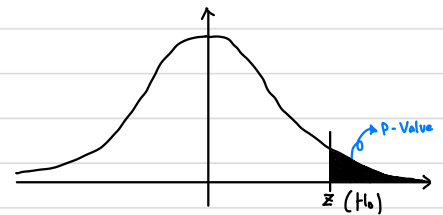


- Statistic/Math
- 1 신뢰 구간의 정의는 무엇인가요?
- 2 p-value를 모르는 사람에게 설명한다면 어떻게 설명하실 건가요?
- Machine Learning
- 3 텍스트 더미에서 주제를 추출해야 합니다. 어떤 방식으로 접근해 나가시겠나요?
- 4 SVM은 무엇이고 왜 반대로 차원을 확장시키는 방식으로 동작할까요? SVM은 왜 좋을까요?
- 5 다른 좋은 머신 러닝 대비, 오래된 기법인 나이브 베이즈(naive bayes)의 장점을 옹호해보세요.
- Deep Learning
- 6 요즘 Sigmoid 보다 ReLU를 많이 쓰는데 그 이유는?
- 7 ReLU로 어떻게 곡선 함수를 근사하나? ReLU의 문제점은?
- 8 Bias는 왜 존재할까?
- Python
- 9 What is __init__?
- Algorithm
- 10 Problem - Programmers - 소수 찾기

1. 신뢰 구간 : 모수가 실제로 포함될 것이라고 예측되는 범위

2. p-value : 귀무 가설이 맞다고 가정할 때 얻은 결과보다 극단적인 결과가 실제로 관측될 확률

↳ 모집단 특성에 대해 옳다고 제안하는 주장



3. LDA 활용 (토픽 모델링의 알고리즘)

· LDA는 '문서들은 토픽들의 혼합으로 구성되어 있으며, 토픽들은 화물 분포에 기반해 단어들을 생성한다'고 가정

· 데이터가 주어지면 LDA는 문서가 생성되는 과정을 역추적하여 토픽 추정

· LDA는 각 문서의 토픽 분포와 각 토픽 내의 단어 분포를 추정

LDA 예시

문서1 : 저는 사과랑 바나나를 먹어요

문서2 : 우리는 귀여운 강아지가 좋아요

문서3 : 저의 깜찍하고 귀여운 강아지가 바나나를 먹어요

<각 문서의 토픽 분포>

문서1 : 토픽 A 100%

문서2 : 토픽 B 100%

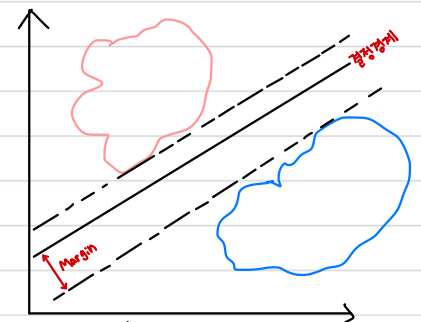
문서3 : 토픽 B 60%, 토픽 A 40%

<각 토픽의 단어 분포>

토픽A : 사과 20%, 바나나 40%, 먹어요 40%, 귀여운 0%, 강아지 0%, 깜찍하고 0%, 좋아요 0%

토픽B : 사과 0%, 바나나 0%, 먹어요 0%, 귀여운 33%, 강아지 33%, 깜찍하고 16%, 좋아요 16%

LDA는 토픽의 제목을 정해주지 않지만, 이 시점에서 알고리즘의 사용자는 위 결과로부터 두 토픽이 각각 과일에 대한 토픽과 강아지에 대한 토픽이라고 판단해볼 수 있다.



<SVM>

4. SVM : 주어진 데이터가 어느 카테고리에 속하는지 판단하는 이진 선형 분류 모델

Kernel trick: 저차원 공간을 고차원 공간으로 매핑해주는 작업

⇒ Why? 저차원에서 구하기 힘든 non linear Seperable line을 비교적 쉬운 고차원에서 linear Seperable line을 구한 후,

저차원 공간으로 매핑하여 non linear Seperable line을 구한다.

SVM 장점

- 범주, 수치 예측에 사용되며, 오목 데이터에 대한 영향이 적다. 과대적합되는 경우가 적고, 신경망보다 사용하기 쉬움
- 결정 평면을 저장하기 때문에 memory 소모가 적다.

5. 나이브 베이즈 (Naive Bayes): 데이터셋의 모든 특징들이 동등하고 독립적이라 가정할 때, 베이즈 정리를 이용하여 어떤 데이터가 특정 클래스에 속하는지 분류하는 알고리즘

장점

- 간단하고 빠르게 효율적인 알고리즘
- 메모리를 적게 차지한다.
- 잡음과 누락 데이터를 잘 처리한다.
- 훈련 할 때 데이터 크기에 상관 없이 잘 작동한다.
- 예측을 위한 추정 확률을 쉽게 얻을 수 있다.

단점

- 모든 특징들이 동등하고 독립적이라 가정이 잘못된 경우가 있다.
- 수치, 특징이 많은 데이터셋에 이상적이지 않다.
- 추정된 확률이 예측된 클래스보다 덜 신뢰할 만하다.

6. Sigmoid 에서 $x \rightarrow \infty$ or $x \rightarrow -\infty$ 에서 기울기 소실이 발생하여 back propagation 과정에서 문제가 발생한다.

7. $ReLU = \max(0, x)$ 에서 확인할 수 있는 것처럼, $y=0$ 과 $y=x$ 의 선형 함수 결합 형태다. 두 결합의 합성 함수를 통해 다양한 곡선(비선형)으로 근사할 수 있다.

· 문제점

$x < 0$ 에서 0 값을 가져 back propagation 과정에서 기울기에 0이 곱해지는 Dying ReLU 현상이 발생한다. 이를 해결하기 위해 Leaky ReLU 나 평행이동을 통해 해결할 수 있다.

함수값의 중심이 0이 아니다. (기함수 or 우함수) 이때 발생하는 문제점은 back propagation 과정에서 가중치의 업데이트가 단방향으로만 발생해 비효율성을 초래한다.

8. 하나의 뉴런에서 활성화 함수를 거쳐 최종적으로 출력되는 값을 조절하는 역할

9. `__init__`: 객체에서 생성자 역할로 실행될 때 처음에 호출되는 함수

10.

```
def isPrime(num): (소수 판정 함수)
    if(num <= 1): return False
    for i in range(2, int(math.sqrt(num)) + 1):
        if num % i == 0:
            return False
    return True

def solution(numbers):
    prime_candidates = [] (소수 후보)
    answer = []
    for i in range(1, len(numbers) + 1):
        prime_candidates += [''.join(k) for k in permutations(numbers, i)]
    prime_candidates = [int(n) for n in prime_candidates]
    for i in prime_candidates:
        if(isPrime(i)):
            answer.append(i)
```