

- Statistic/Math
- 1 중심극한정리는 왜 유용한걸까요?
- 2 엔트로피(entropy)에 대해 설명해주세요. 가능하면 Information Gain도요.
- Machine Learning
- 3 ROC 커브에 대해 설명해주실 수 있으신가요?
- 4 K-means의 대표적 의미론적 단점은 무엇인가요? (계산량 많다는것 말고)
- 5 머신러닝(machine)적 접근방법과 통계(statistics)적 접근방법의 둘간에 차이에 대한 견해가 있나요?
- Deep Learning
- 6 GD가 Local Minima 문제를 피하는 방법과 찾은 해가 Global Minimum인지 아닌지 알 수 있는 방법은 무엇이 있을까요?
- 7 Training 세트와 Test 세트를 분리하는 이유와 Validation 세트가 따로 있는 이유는?
- 8 Test 세트가 오염되었다는 말의 뜻은 무엇인가?
- Python
- 9 What does \*args, \*\*kwargs mean? And why would we use it?
- Algorithm
- 10 Problem - Programmers - 피로도

1. 중심 극한 정리 : 모집단이 '평균이  $\mu$ 이고 표준편차가  $\sigma$ 인 임의의 분포'를 이룬다고 할 때, 이 모집단에 추출된 표본의 크기가 충분히 ( $n \geq 30$ ) 크다면 표본 값이 이루는 분포는 '평균이  $\mu$ 이고 표준편차가  $\frac{\sigma}{\sqrt{n}}$ 인 정규분포'에 근접한다.

장점: 모집단의 분포형태에 상관없이 널리 적용이 가능

2. 엔트로피 : Example들의 집합에서 혼합성을 나타내는 척도

골고루 섞여 있다(무질서)  $\Rightarrow$  High Entropy, 하나의 속성을 가진 것으로만 이루어져 있다(순수)  $\Rightarrow$  Low Entropy

Information Gain : 어떤 분류를 통해서 information (정보)에 대한 gain(이익)이 생겼는지를 나타내는 척도

High I.G  $\Rightarrow$  변별력이 좋다 (어떤 속성을 선택함으로써 데이터를 잘 구분함)

3. ROC 커브 : 모델을 평가할 때 쓰이는 지표로 민감도와 (1 - 특이도)로 그려지는 곡선

예측 클래스 (predict class)

		False	True	
실제 클래스 (Actual Class)	False	TN (True Negative)	FP (False Positive)	: 정답
	True	FN (False Negative)	TP (True Positive)	: 오답

• 민감도 (Sensitivity) : 실제 양성(True) 중 예측한 양성 비율 ( $\frac{TP}{TP+FN}$ )

• 특이도 (Specificity) : 실제 음성(False) 중 예측한 음성 비율 ( $\frac{TN}{TN+FP}$ )

ROC 커브가 좌상단에 가까운 경우, 좋은 모델이라 평가한다.

4. 단점

- K값과 centroid (Cluster의 중심점)를 임의로 정해야 하며, 첫 centroid에 따라 cluster 결과가 민감하게 변한다.
- Outlier (경계선)에 민감하다. 그에 따라 centroid가 정해지기 때문.
- 원형 또는 구형의 cluster만 찾을 수 있다. 원형이 아닐 때는 정확한 결과를 도출하지 못한다.

5. 머신러닝적 접근 방법 : 모델의 예측을 높이는 것이 목적이기 때문에 다양한 피처를 바탕으로 높은 예측을 달성하는 것이 목적 (예측)

통계학적 접근 방법 : 분포와 가정을 통해 신뢰가능하고 검증한 모델을 만드는 것이 목적으로 모형을 단순하게 구성한다. (분석)

6. Local Minima 해결 방법: Momentum 최적화, 네스테로프 가속, AdaGrad, RMSProp, Adam, Nadam...

· Momentum 최적화 ( $\theta$ : 가중치,  $J(\theta)$ : cost func,  $\eta$ : learning rate,  $\nabla_{\theta} J(\theta)$ : Gradient)

기본 경사하강법:  $\theta \leftarrow \theta - \eta \nabla_{\theta} J(\theta)$

Momentum:  $m \leftarrow \beta m - \eta \nabla_{\theta} J(\theta)$ ,  $\theta \leftarrow \theta + m$  ↖ 관찰

· 네스테로프 가속 경사

$m \leftarrow \beta m - \eta \nabla_{\theta} J(\theta + \beta m)$ ,  $\theta \leftarrow \theta + m$

· AdaGrad: 현재 최적점으로 가도록 파라미터의 변화를 감지하여 더 정확한 방향으로 향하게 한다.

$\theta_i \leftarrow \theta_i - \eta \cdot \frac{\nabla_{\theta} J(\theta_i)}{\sqrt{s_i + \epsilon}}$  ↓ Scale 조정

7. Train, Test  $\Rightarrow$  Training Set의 Overfitting 방지

Validation Set  $\Rightarrow$  모델의 성능을 평가하기 위해 (모델 선택에 관여)

8. Test Set의 운영

· Train Set와 유사

· Not Random (편향됨)

9. \* args: 매개변수의 수를 모를 때, list & tuple의 argument를 전달할 때 사용

\* kwargs: 매개변수의 수를 모를 때, 키워드를 지정해 dictionary의 argument를 전달할 때 사용