TV, 라디오, 신문 . 이 세자기에 대한 광고료.

X가 이 세가지에 대한 광고료. (input 값 의미)

Y가 output값 의미.

input값이 여러 개 있다면, 

이걸 수식으로 쭉쭉 쓰게 된다면,  이런 식으로 구성.

여기서 epsilon은 오류값.

텍스트, 도표, 라인, 그래프이(가) 표시된 사진

자동 생성된 설명

위의 식을 이 그림에 적용하면,

교육에 따른 소득이 왼쪽 그래프와 같이 나타난다면,

f 함수를 추정해야 한다. 간략하게 그냥 f 함수 시물레이션 돌린게 오른쪽 그림의 파란 선.

여기서 검은 선은 오류(epsilon)

파란선 위에 점은 파란선 위가 +, 파란선 아래는 – 라서 결국엔 이거 전부 더해지면 평균적으로 0에 가까운 값을 낸다.

그리고 epsilon이 0에 가깝다는 것은, 점 하나하나를 기준으로 overfitting, underfitting이 balance 있게 수행하고 있음을 의미한다.

종이접기, 디자인이(가) 표시된 사진

중간 신뢰도로 자동 생성된 설명

이건 X 값이 여러 개(교육, 경력).

그냥 여기서는 다차원 적인 f로 나타날 수도 있다고만 하고 넘어감.

Prediction(예측)

1. X(입력) 값은 쉽게 얻을 수 있는데, Y(출력)값은 쉽게 얻을 수 없다. 예를 들어, X를 환자의 여러 변수(키, 나이) 이런거라고 한다면, 이에 대한 부작용(Y)는 쉽게 얻을 수 없다. 그래서 f^을 이용해서 출력 예측값(Y^)을 얻으려고 하는건데, 이를 식으로 나타내면 Y^ = f^(X)

여기서 보통 f^은 블랙박스 취급. 이 함수를 맞춰야 하는게 관건.

1. 줄일 수 있는 오류(reducible error): 실제함수 f랑 예측함수 f^이 얼마나 잘 일치하는지.

못 줄이는 오류(irreducible error): f^이 f랑 완벽하게 추정 잘했다고 해도, epsilon 영향 때문에, 완벽하게 맞을 수는 없다. 결국 epsilon이 irreducible 하다 라는 말.

1. Irreducible error가 0 보다 큰 이유: 아주 미세한 측정 차이(키 깔창껴서 0.5올라가고, 나이 출생신고 늦게해서 젊어지고 그런거) 땜에 어쩔 수 없다.(noise 의미)
2. 이 식에서 Y가 실제 output 값. F(x)가 실제 함수, f^(x)가 추정 함수. 전체 식이 MSE.

여기서 epsilon값을 밖으로 빼면, 폰트, 텍스트, 화이트, 라인이(가) 표시된 사진

자동 생성된 설명 이렇게 구성.

Var(e)는 e에 의해 발생하는 변동성 의미. 결국 오차는 아무리 잘 없애려 해도 존재하기 마련(irreducible) 이다. # 분산 🡺 variance

Inference(추론)

추론하는 목적: 어떤 예측변수(X)가 출력변수(Y)와 관련이 있는가🡺 feature extract

각 X와 Y의 관계는 무엇인가 🡺 어떤건(ex. X1과 y) 양의 상관관계, 어떤건(ex. X2와 y) 음의 상관관계

Linear하게 단순하게 바꿀 수 있는가 🡺 복잡하면 Linear model은 적합하지 않다.

이런걸 맞추기 위해 ‘추론’이 필요하다.

추론 예시:

1. 어떤 매체가 판매와 연관이 있는가?
2. 어느 매체가 가장 높은 판매 증가를 불러일으켰는가?
3. TV 광고가 늘어났을 때, 판매량이 얼마나 증가했는가?

## 결국에는 inference와 prediction 차이는 f값을 딱 알아 맞추는게 prediction, x값 하나하나 다 뜯어보는게 inference(ex. Feature extract)

## 데이터 분석 쪽이 추론. 어떤 모델링 쪽은 예측.

F 함수를 추정하는 두가지 방법 설명하겠다.

Parameteric method

1. 폰트, 텍스트, 타이포그래피, 서예이(가) 표시된 사진

   자동 생성된 설명여기서는 선형 모델을 예시로 들었는데, f(x)값 추정하는 건 매우 쉬워진다. B\_0~B\_p 까지의 값만 추정하면된다. # 첫번째 단계.
2. 그 다음에 모델 training 시켜서 B\_0~B\_p값 찾아야 한다. 결국 실제 Y값이랑 최대한 근사하게 만드는게 목적. # 모델 fitting 시키는 방법 중 하나가 최소제곱법(Ordinary least squares)

Parametric method 장점:

파라미터(B\_0~B\_p) 계수만 추정하면 됨.

이래서 매우 간단함.

단점:

데이터 형태 자체가 linear 하지 않을 수 있다. 그래서 f가 적합하지 않는 문제 존재.

그래서 최대한 유연한 모델(다차원 말하는 거)을 써야 하는데, 그럼 또 overfitting의 문제 발생.

Non parametric method:

f를 미리 가정하지 않음. 데이터 형태에 최대한 근접하게 적합 하게 끔 하면서 불규칙 하게 하지 않도록 f를 추정함.

Parametric 방식은 f를 처음 잘못 가정하면 부정확한 모델 만들지만, non parametric 방식은 이런 위험을 피할 수 있다.

단점: 파라미터(B\_0~B\_p)를 parametric 처럼 단순화 하지 않아서 매우 많은 데이터가 필요.

디자인이(가) 표시된 사진

자동 생성된 설명(2.3)

큐브, 디자인이(가) 표시된 사진

자동 생성된 설명(2.4)

디자인, 큐브이(가) 표시된 사진

중간 신뢰도로 자동 생성된 설명(2.5)

여기서 2.3이 실제 f

2.4가 parametric 방식 사용한 f 함수

2.5가 non parametric 방식 사용한 f 함수 인데,

실제 2.3과 2.5가 매우 유사한 것을 볼 수 있다.

그림, 예술, 디자인이(가) 표시된 사진

자동 생성된 설명(2.6)

2.6은 non parametric이 overfitting된 문제. 새로운 데이터가 들어오면 잘 못 맞추는 문제 발생.

정리: parametric 방식은 데이터 작을 때 효율적. 좀 부정확하다(underfitting)는 단점 존재.

Non parametric 방식은 매우 다양한 형태의 함수에(데이터에 맞출 수 있는) 맞출 수 있지만, 매우 많은 데이터 필요 & overfitting 문제 발생.

텍스트, 스크린샷, 라인, 디스플레이이(가) 표시된 사진

자동 생성된 설명

유연성(flexibility), interpretability(해석 가능성) 트레이드 오프

Linear regression 같은 경우는 선형 함수만 생성해서 덜 유연하다.

아까 봤던 2.6 같은 방식은 다양한 함수 추정 가능해서 훨씬 더 유연하다.

추론(inference)같은 방식은 해석 가능해야 한다. 그래서 덜 유연한 방식이 더 적합.

선형 모델 같은 경우는 Y랑 X 관계 쉽계 이해 가능해서 추론에 더 적합.

유연한 방법은 다양한 함수 f를 추정하기 좋지만, 해석이 그만큼 더 어렵다.(ex. 딥러닝, 블랙박스 모델) # 곡선 형태가 너무 복잡하게 구성되어 있어서 더 그렇다.

추론이 목표라면 해석 가능해야 해서 덜 유연한게 적합.

예측이 목표면 유연한 모델이 더 적합. # 물론 overfitting 문제도 있어서 이거 잘 고려해서 시행할 것.

지도 학습(supervised learning):

 예측 입력 변수 x\_i가 주어지면,

이에 대응하는 Y\_i가 주어짐.

미래 관측치(데이터)에 대한 Y를 정확히 예측하거나, 변수들 간의 관계 이해하는 것이 관건.

Ex. Linear regresson, logistic regression, SVM 등등

비지도 학습(unsupervised learning):

예측 입력 변수 x\_i만 주어지고, Y\_i는 안 주어짐(라벨링 안되어 있는거 생각)

데이터간의 관계나 패턴 찾는 것이 관건.

텍스트, 지도, 도표이(가) 표시된 사진

자동 생성된 설명

이거에 대한 예시로 클러스터링. 위의 그림.

왼쪽 그룹은 군집(그룹) 이 매우 명확해서 클러스터링 쉬운데, 오른쪽은 그룹 나누는게 좀 어려움.

준지도 학습(semi-supervised learning):

몇 개만 라벨링(y\_i값) 나와있고, 몇 개는 y\_i값 없는 것.

이 책에서는 안 다룸.

양적 변수(quantitative variables):

숫자로 표현되는 변수. Ex. 나이, 키, 소득, 주식가격

질적 변수(qualitative variables, categorical variables):

범주나 클래스로 구분되는 변수. Ex. 결혼 여부, 옷 브랜드, 채무 불이행 여부

Regression 문제는 quantitative 에 주로 사용.

Classification 문제는 qualitative 문제에 주로 사용.

양적 변수, 질적 변수 두 문제에 통계적 학습 다 적용 가능한데, qualitative 변수는 작업 전에 작업 필요(one hot encoding, labelEncoder 등등)

이제부터 모델 적합도 보는 방식 설명하겠다.

폰트, 텍스트, 화이트, 타이포그래피이(가) 표시된 사진

자동 생성된 설명

예측값(f^x\_i)이랑 y\_i의 실제 값 차이 제곱해서 평균내는 성능 측정 방식.

위에꺼는 train MSE 라고 하는데, 훈련 데이터 사용해서 모델 fit 시킨 후에, 성능 측정할 때 쓰는 것.

폰트, 타이포그래피, 텍스트, 화이트이(가) 표시된 사진

자동 생성된 설명

이건 위의 MSE랑 똑 같은 식인데, testMSE라고, 새로운(test)데이터 사용할 때 모델 성능 측정하는 식.

Test MSE가 더 중요하다. 당연히.

# trianMSE가 작다고 모델이 잘 작동하는지는 모르는 거기 때문에, test MSE가 더 중요하다.

도표, 라인, 텍스트, 그래프이(가) 표시된 사진

자동 생성된 설명

왼쪽 그림에서 초록색 선 보면, 유연성이 매우 커서 데이터 대변을 매우 잘함.s

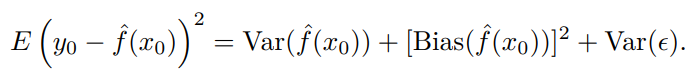
근데 오른쪽 그림보면 검은선(train MSE)는 줄어드는데, 빨간선(test MSE)는 폭발적 증가.

이게 overfitting 됐다는 것을 의미.

Cross validation 잠깐 언급. 실 데이터에는 test 데이터가 없어서, Train 데이터 잘개 나눠서 test MSE 측정하는 방식인데 나중에 다룬다고 넘어감.

편향-분산 트레이드오프(Bias-variance trade-off):

모델의 test MSE(모델의 유연성)가 어떻게 편향과 분산에 영향을 받는지.



Var(f^x\_0)는 모델이 다른 훈련 데이터셋 사용했을 때, f^x\_0가 얼마나 변하는지 나타냄.

유연한 모델은 분산이 더 큼.

## 여기서 Bias값에 제곱이 붙는 이유는 예측값과 실제값 차이(평균적 오차 나타냄) 나타내기 위해서, 편향 자체가 단일 값(크기)라서 이거 제곱해서 에러가 기여하는 정도 계산 하기 위함.

분산은 어떤 예측 값의 변동성 나타내는 값. 분산 자체가 MSE의 구성 요소라서 굳이 제곱 안함.

Bias(f^x\_0)는 실제 데이터를 단순한 모델로 갖다 붙일 때 발생하는 오차.

Linear regression 같은 모델은 편향이 큼.

끝에 Var€는 그냥 오차. 이건 못 줄이는 노이즈.

Var(분산)는 새로운 데이터를 썼을 때, 얼마나 변동하는지 나타내는 것.

더 유연한 모델은 분산이 커짐.

Bias는 모델이 데이터 대변 잘 못해 발생하는 오차.

Linear regression 같은 비선형 모델은 편향값이 큼.

유연성 낮은 모델: 편향 크고 분산 낮고,

유연성 높은 모델: 분산 크고, 편향 낮고. (overfitting문제 발생 큼)

그래서 trade off 그래프를 보는건데, 편향 줄이면 분산 증가, 분산 줄이면 편향 증가.

이런거 때문에 적절한 선을 찾아야 한다.

도표, 텍스트, 라인, 그래프이(가) 표시된 사진

자동 생성된 설명(2.9)

도표, 라인, 그래프, 지도이(가) 표시된 사진

자동 생성된 설명(2.10)

도표, 텍스트, 지도, 라인이(가) 표시된 사진

자동 생성된 설명(2.11)

도표, 텍스트, 라인, 그래프이(가) 표시된 사진

자동 생성된 설명

첫번째 그림 보면, 2.9모델 복잡하니까 var 값 높고, bias 값은 작고, MSE도 당연히 높아지고,

두번째 그림 보면, 2.10 모델도 쓸데없이 복잡해서 var값 높고, bias 값 작고,

세번째 그림은 2.10 모델 딱 적합해서 전부 이상적인 형태 띄고 있는 거 보임.

이게 regression에서 편향-분산 trade off 보는 이유.

분류문제에서도 위 trade off랑 비슷하게 베이즈 분류기(Bayes classifier)가 있다.

폰트, 도표, 라인, 텍스트이(가) 표시된 사진

자동 생성된 설명 분류 문제 ‘**train** 오류율’.

여기서 I는 i번째 데이터 잘못 분류하면 1, 정확히 분류하면 0 반환.

N은 데이터 수, y\_i는 실제 값, y^\_i는 예측 값.

폰트, 타이포그래피, 텍스트, 서예이(가) 표시된 사진

자동 생성된 설명 **test** 오류율

위의 식이랑 똑같은데, 사용하는 곳이 다름.

Y\_0는 실제 클래스, y^\_0 예측 클래스.

좋은 분류기는 이 값이 가장 낮은 분류기.

## 여기까지가 분류 문제 오류값 측정 식.

폰트, 화이트, 텍스트, 타이포그래피이(가) 표시된 사진

자동 생성된 설명베이즈 분류기

Input값 X가 X\_0(input 변수 첫번째)일 때, output값 Y가 jdp 속할 확률 계산.

Ex. Pr(Y = 1|X = x0) > 0.5 라면, 이건 1로 분류.

여기서 0.5 값은 Bayes decision boundary인데, 여기서는 50%로 둬서 0.5

베이즈 오류율 측정법.

실제 데이터는 클래스 겹치는 문제 빈번하기 때문에 0이 될 수 없다.

여기서 irreducible error라는게 나왔는데, 데이터 자체의 노이즈 없앨 수 없는 것 같이 베이즈 오류율 0 될수 없다는 말.

KNN(k-nearest Neighbors):

KNN은 베이즈 분류기 대신해서 사용 가능한 방법.

폰트, 텍스트, 화이트, 서예이(가) 표시된 사진

자동 생성된 설명

앞에 식은 베이즈 분류기. 뒤의 식은 KNN식.

N\_0는 x\_0랑 가장 가까운 k개의 이웃.

K = 3이라고 하면, x\_0랑 가까운 3개 훈련 데이터 선택. 두개 파란색이고 하나 주황색이면 파란색으로 분류.

K값 크면, 선형적으로.

K값 작으면, 너무 유연해져서 데이터 노이즈까지 따라가는(비선형적)으로 돼서 overfitting 문제 발생.

K =1, 매우 유연, overfitting 문제 발생. 낮은 편향, 높은 분산.

K = 100, 매우 linear. 높은 편향, 낮은 분산.

텍스트, 도표, 라인, 그래프이(가) 표시된 사진

자동 생성된 설명

지금 이거 보면, K값 낮아짐(분수꼴 잘 볼 것)에 따라 train 오류는 낮아지는데, test 오류는 올라가는 걸 볼 수 있음.

딱 중간지점 10 정도가 적당해 보임.

## regression 문제랑 마찬가지로, classification 문제에서도 적절한 유연성 수준 선택하는게 성능에 좋다.