241004 4회분량

개요

- 카페설명
  - 3기 부원 스터디 게시판이 생김
  - 팀 소개, 구성원, 주제 간단하게 작성
  - 스터디 내용 작성
  - 양식: 운영진 스터디 글 참고해서 간단하게

▼ 1. 깃과 깃허브

깃허브: 개발자들 협업 도구이자 가장 유명하고 거대한, 무료 깃 저장소

깃: 리누스 토르발스가 개발한 버전관리 시스템

(bash 설치, 레포지토리 생성)

cd > clone > add > commit > push

▼ 2. 지도 학습

`given (x1, y1), (x2, y2), … , (xi, yi)`

learn a function f(x) to predict y given x

- 함수 f(x)를 학습하는 것.
- 한 데이터에 (x, y)로 구성
- …

회귀와 분류의 직관적 이해

- classification:
  - 이산적인 범주나 클래스를 예측하는 문제
  - 확률 모델을 사용해 각 클래스에 대한 확률을 계산하여 예측
  - 예: 스팸 메시지
- Regression:

- 연속적인 수치 값을 예측하는 문제
- 예: 주택 가격 예측

회귀

- 입력 X를 받아서 출력 Y(연속적)를 예측하는 것
- 단순회귀식: Y^ = β^_0 + β^_1X
- 다중회귀식: Y^ = β^_0 + β^_1X_1 + β^_2X_2 + ... + β^_kX_k
- β^_0는 절편값
- ...

분류

- 입력 X 받아서 출력 Y(이산적)를 예측하는 것
- 이산적인 범주(클래스) 예측 문제. 확률을 계산하여 예측
- P(y = C_k | x) = 1 / (1 + e^-(w1×1+w2×2+...+wnxn+b)) *<< !!다중회귀식과 유사!!*
- P(y = 1 | x) = 1 / (1 + e^-z)

▼ 3. 회귀와 분류 종류

회귀

- Linear Regression - 선형 회귀
  - x: 입력변수
  - y: 출력값
  - w: 입력변수의 가중치
  - b: 절편값, 바이어스
    - x값은 데이터에 따라 정해져 있고, 결국 w값(파라미터)을 맞추는 게 관건
  - 손실함수: 예측한 값이랑 실제 값 사이의 차이를 수치적으로 알 수 있게 해줌
    - MSE = (1/N) ∑^N_(i=1) (y_i - (w^Tx_i + b))^2
    - N: 데이터 수
    - y_i : 실제 값
    - x_i : 입력 값

        - w^Tx_i + b: 예측된 값
        - ...
- Polynomial Regression - 다항 회귀
    - y = w_0 + w_1x + w_2x^2 + w_3x^3 + ... + w_dx^d
    - x: 입력 변수
    - y: 출력 변수
    - w_0, w_1, ... , w_d: 가중치
    - d: 차수
    - 동일하게 선형 회귀 범주에 속함.
    - 차수가 작고 높고의 문제는 과도적합, 과소적합 문제와 직결
- Ridge/Lasso - 규제
    - 과도적합 방지 용도. 말만 회귀지 그냥 규제임
    - J(w) = (1/N) ∑^N_(i=1) (y_i - (w^Tx_i + b))^2 + λ (1/N) ∑^n_(j=1) (w^2_j)
        - = MSE + λ (1/N) ∑^n_(j=1) (w^2_j)
    - ...
    - L1:
        - 주로 feature selection에 사용
        - 변수 많을 때, 중요 변수만 남기는 방식
        - 절대값이 선형적으로 증가하는 효과 부여
            - 그래서 가중치 0이 되는 경우 생김
    - L2:
        - 모든 feature 다 쓰면서 가중치 조절해서 모델 과도적합 방지
        - 제곱값이 가중치 크면 더 큰 손실, 작으면 크게 변화 X
            - 그래서 0에 가까워지기만 하고 딱 0은 되지 않음

분류

- Perceptron

    - f(x) = {
        - 1 if x ≥ 0
        - 0 if x < 0
    - 결국 이진 분류에 사용
    - 굉장히 초창기 분류 문제
    - y = ∑^n_(i=1) (...뭐였더라)
- SVM (support vector machine)
    - 최대 마진을 찾는 것. 두 클래스 잘 구분하는 **결정경계** 만드는 것이 목적
        - 마진: 두 클래스의 간격
    - Decision Hyperplane: 두 클래스 분리하는 선. 2차원에선 선, 3차원에선 평면
    - Margine: 가장 가까운 데이터포인트 + Decision Hyperplane에 가장 가까운 데이터포인트
    - d = |w^Tx_0 + b| / ||w|| *<< 선*
    - M = 1 / ||w|| *<< 마진*
        - w: decision hyperplane의 방향
        - ||w||: normal vector의 크기
        - x_0: 데이터 포인트
        - b: 상수값
    - normal vector: 고차원 공간에서 데이터 분리하는 평면. 그 평면에 수직으로 향하는 벡터
    - 결국 d는 x_0와 decision hyperplane 사이의 거리를 의미
    - ||w||값 커지면 margin값 감소 (서로 반비례) > 분자를 1로 고정하여 표현
    - decision hyperplane 정의 식: wx + b = 0
- ~~으어어살려줘~~
- Logistic/softmax Regression
    - Logistic Regression
        - z = w^Tx + b

        - σ(z) = 1 / (1 + e^-z)
        - y^ = {
            - 1 if σ(z) ≥ 0.5
            - 0 if σ(z) < 0.5
        - 선형회귀와 비슷하게 선형 결합 형태
        - 하지만 출력값을 시그모이드 함수 사용하여 확률값으로 변환
    - Softmax Regression
        - z_k = w^T_k x + b_k
        - P(y = k | x) = e^zk / (∑^K_(j=1) e^zj)
            - P(y = C_k | x) = ... 와 비슷함. 연장선이라 이해하기
        - 결국 z_k의 확률값을 반환
        - 모든 클래스의 확률 합 = 1
        - 동일한 선형 결합 형태
        - 그래서 로지스틱 회귀의 연장선
            - 클래스가 2개라면 로지스틱 회귀와 완전동일

▼ 4. 트리모델 (4중2)

Tree Model

- Decision Tree Classifier
    - 데이터 특성 기준으로 의사결정 규칙 만들어서 분류하는 모델
    - 루트 노드: 맨 처음 나누는 지점
    - 결정 노드: 데이터 특성에 따라 분기되는 지점
    - 리프 노드: 최종 클래스 결정
    - Gini(D) = 1 - ∑^C_(i=1) P^2_i
    - Gini: 불순도 지표 (데이터가 얼마나 혼합되어있는지)
    - 0이면 완벽하게 한 클래스에 속한 것. 0.5에 가까우면 클래스 여러 개 혼합된 상태
    - P_i: 클래스 i에 속할 확률
    - C: 클래스 개수

    - Entropy(D) = -1 - ∑^C_(i=1) P_i logP_i
    - ...
- Decision Tree Regressor
    - 데이터를 여러 구간으로 나눠서 각 구간 내에서 예측값 내는 것
    - 각 노드 개념은 동일
    - 하지만 리프도느에서 평균이나 중앙값을 최종 예측값으로 내는 차이 존재
    - Gini, Entropy 대신 MSE나 분산을 지표로 주로 사용
    - Variance(D) = (1/n) ∑^n_(i=1) (y_i - y^-)^2
- ㄴ 두 트리의 공통점
    - 트리가 너무 깊을 때 과도적합 주의
    - 트리가 너무 얕을 때 과소적합 주의
    - 해결:
        - 가지 깊이를 제한하거나 가지 몇 개를 제거
        - 앙상블(Random Forest, Gradient Boosting) 기법 적용
        - Cross Validation으로 살펴보기

▼ 과제ㄱ

- 회귀 문제에서의 손실함수 더 찾아볼 것
- 수업 내용 정리
- 오늘 나온 내용, sklearn 라이브러리 활용 후 적용
- 분류문제 데이터: IRIS
- 회귀문제 데이터: Diabetes
    - 폴더명: 2주차_3기_OOO
    - 한 폴더 안에
    - 손실함수 더 찾은것 & 수업 내용 정리, .ipynb 파일
    - PR

다음주 목요일까지 (10/10)