

언플러그드 인공지능 수학 수업1

October 10, 2021

1 언플러그드 인공지능 수학 수업

1.1 이론적 배경

1.1.1 머신러닝?

- 지도학습: 문제와 답이 모두 있는 경우.
- 비지도학습: 문제만 있고 답은 직접 찾게 하는 경우.
- 강화학습: 행동마다 보상을 주는 경우.

1.1.2 지도학습: 의사결정나무, 랜덤포레스트, 로지스틱 회귀, 최근접 이웃법(kNN), 서포트벡터머신(SVM) 등

의사결정나무

1. 의사결정나무란?

- 의사결정규칙을 나무구조로 나타내 전체자료를 몇개의 소집단으로 나누어 예측하는 분석 방법. 분류 함수를 이용해 나무 구조로 분리함
- 계산 결과가 직접 드러나기 때문에 분류 이유를 투명하게 보여줄 수 있어 의학분야나 신용평가 등에 사용됨

2. 의사결정나무를 만드는 과정

- 의사 결정 나무 성장(가지 분할) → 가지치기 → 타당성 평가 → 해석/예측

3. 가지 분할

- 순수도/불순도 : 부모보다 자식의 가지들의 순수도가 높아지도록(불순도가 낮아지도록) 분리함
- 불순도 척도: 카이제곱통계량, 지니 계수, 엔트로피, 분산 감소량
- 카이제곱 통계량 계산법: $\chi^2 = (\text{실제도수} - \text{기대도수})^2 / (\text{기대도수})$ 의 합 (기대도수는 전체 비율을 의미)
- 지니계수 계산법: $1 - \sum_{l=1}^k P_l^2$
- 엔트로피 지수 계산법: $-\sum_{l=1}^k P_l \log_2 P_l$

4. 가지치기

- 너무 큰 나무는 과대적합 너무작은 나무는 과소적합의 위험성이 있어 가지치기 작업을 함.
- 최대 깊이를 넘어서거나 마디에 속하는 자료가 일정수 이하일 때 분할을 멈추는 과정.
- Prepruning(reduced-error pruning, c4.5 pruning) / Postpruning(subtree replacement, subtree raising)
- 과대적합: 학습 데이터에 대한 성능은 좋지만 실제 데이터에서는 성능이 떨어지는 경우

5. 분류 방법: c4.5 GainRatio, C-SEP(GiniIndex), CHAID, G-statistics, MDL, CART

앙상블 기법

- 여러가지 분석 모형의 예측, 분류 결과를 종합해 최적화된 결과를 내는 방법
- 다중 모델 조합, 분류기 조합이 있다.
- 앙상블 기법에는 대표적으로 배깅, 부스팅, 랜덤포레스트가 있다.
- 일반적으로 기존 모델보다 같거나 더 나은 결과를 예측한다.

랜덤포레스트

- 랜덤 포레스트는 서로 다른 부트스트랩 데이터로 의사결정나무를 여러 개 모은 것이다.
- 부트스트랩이란 랜덤 복원 추출로 만든 샘플을 의미한다.
- 의사결정나무의 결과값을 다수결로 종합해 예측, 분류값을 결정한다.

1.1.3 비지도 학습

군집화(클러스터링)

- 분류는 사전에 분류 라벨을 주므로 지도학습이지만 군집은 사전에 정의된 범주가 없이 데이터의 특성을 분석해 군집화해야한다는 차이점이 있다.
- 데이터간 거리 개념을 도입. 주어진 데이터를 탐색하며 군집 내 거리를 최소화하고 군집 간 거리는 최대화하는 방식으로 분류.

비계층적 군집화

- 사전에 k개의 클러스터가 있다고 가정하고 시작.
- K평균 알고리즘, 혼합 분포 군집, 가우시안 혼합모델(GMM), EM알고리즘, DBSCAN, SOM 알고리즘 등
- K평균 알고리즘 과정: k개 랜덤 선택 → 자료를 가장 가까운 군집에 할당 → 군집 중심값 갱신 → 중심에 변화가 거의 없을 때까지 반복
- K평균 알고리즘은 이상값에 민감하게 반응하기 때문에 이상값을 미리 제거해야 한다.

계층적 군집화

- 병합적 방법/ 분할적 방법 군집화

군집간 거리 측정 방법: 최단연결법, 최장연결법, 중심연결법, 평균연결법, 와드연결법

1. 평균연결법: 모든 항목에 대한 거리의 평균을 구하는 방법. 이상치에 덜 민감하나 계산 복잡도가 매우 높다.
2. 와드연결법: 군집내 오차제곱합을 계산하여 거리를 구함.

딥러닝: ANN DNN CNN RNN GAN

1. 퍼셉트론: 단일 인공신경망. 뉴런의 구조를 본 따 만들었으며 활성화 함수로 있다.
2. ANN: 다중 인공신경망
3. DNN: ANN의 은닉층을 많이 늘린 것(2개 이상), 오류 역전파 → 역전파의 원리 학습
4. CNN: 합성곱 신경망, 필터행렬곱 컨볼루션(압축), 풀링, 문장분류 이미지인식 → 압축 원리 학습
5. RNN: 순환 신경망, 이전 노드 학습결과 다음 노드에 반영, 연속 데이터, 음성이나 자연어처리
6. GAN: 적대적 생성 신경망, 생성기와 판별기가 서로 경쟁하면서 학습.

1.1.4 머신러닝을 무료로 체험해 볼 수 있는 교육용 사이트&프로그램

1. 구글 티처블 머신 <https://teachablemachine.withgoogle.com/>
2. IBM 머신러닝 for KIDS <https://machinelearningforkids.co.uk/>
3. WEKA <https://svn.cms.waikato.ac.nz/svn/weka/>

2 언플러그드 인공지능 수학 강의 지도안1

2.1 의사결정나무를 이용한 언플러그드 수학 수업

2.1.1 수업 개요

1. 의사결정나무가 무엇인지 소개하기
2. 조별활동1: 주어진 예시 데이터로 직접 의사결정나무를 만들기. 분할 할 때 어떤 기준으로 했는지 기록할 것.
3. 의사결정나무 테스트 해보기
4. 조별발표: 만들어진 의사결정나무와 분할한 방법 그리고 정확도를 발표하기
5. 발문1: 어떻게 해야 효과적/효율적으로 가지 분할을 할 수 있을까?
6. 조별활동2: 의사결정나무를 어떻게하면 효율적으로 만들 수 있는지 토의하고 발표하기
7. 발견하기: 카이제곱통계량, 지니계수, 엔트로피의 계산 알려주기
8. 발문2: 과대적합문제 - 학습데이터는 잘 맞는데 실제로는 왜 안맞을까?
9. 조별활동3: 만들어진 의사결정나무를 불순도 척도 중 1개를 택하여 계산해보기
10. 관찰하기: 주어진 코드를 실행한 실제 결과 확인하기
11. 관찰하기: 그래프로 보는 의사결정 트리 - 축에 수직으로 자른다. 한계는? 방법은?
12. 제시하기 : 랜덤 포레스트에 대한 간략한 설명
13. 발견하기 : 조별로 만든 의사결정 트리로 다수결에 의한 랜덤 포레스트 시도하기
14. 랜덤 포레스트가 더 좋은 결과를 내는 이유 토의해보기
15. 배운 내용 정리 및 마무리

2.1.2 평가방법

- 자기평가, 동료평가, 교사관찰평가
 1. 조별 활동 참여도
 2. 발문에 대한 토의 적극성
- 교사평가
 1. 발표 적극성
 2. 의사결정나무의 완성여부
 3. 의사결정나무 불순도 척도 계산의 이해도

3 언플러그드 인공지능 수학 수업 - 의사결정나무 학습지 자료

3.1 의사결정나무 만들어보기

다음은 질환A를 판단하기 위한 변수 X, Y에 대한 자료이다.

Target의 값이 0이면 위험군, 1이면 비위험군이라고 한다.

다음 데이터에서 중복을 허용하여 무작위로 24개를 뽑은 후 환자의 질병여부를 판단하는 의사결정나무를 만들어보자.

난수 생성 사이트 :<https://kr.piliapp.com/random/number/>

id	X	Y	Target
0	12	7	0
1	15	6	0
2	10	12	0
3	20	13	0
4	23	12	1
5	17	7	1
6	19	19	1
7	13	16	0
8	10	14	0
9	19	11	1
10	12	11	0
11	26	15	1
12	11	15	0
13	7	11	0
14	15	18	0
15	16	12	1
16	26	12	1
17	21	9	1
18	15	10	1
19	10	10	0
20	16	15	0
21	24	17	1
22	22	18	1
23	15	13	1

3.2 의사결정 나무의 성능 측정

직접 만든 의사결정나무에 테스트 데이터를 넣어 검증해보고 아래에 맞았는지 기록해보자

id	X	Y	예측한 Target	실제 Target	예측 성공 여부
0					
1					
2					
3					
4					
5					

3.3 의사결정나무 각 단계의 카이제곱통계량, 지니계수, 엔트로피 중 1가지 계산해보기

계산기를 활용해 직접 만든 의사결정나무의 각 단계별로 카이제곱통계량, 지니계수, 엔트로피 중 1가지를 선택해 계산해보자

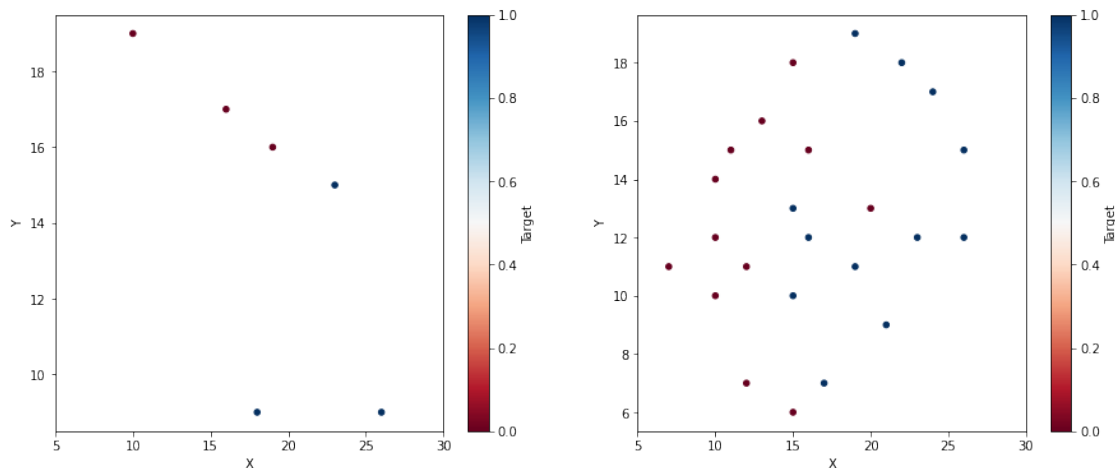
4 언플러그드 인공지능 수학 수업 - 실제 시뮬레이션 자료

```
[2]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

df1 = pd.read_csv('Decision_tree_Example.csv')
pd.set_option('display.max_rows',None)
train = df1.sample(n=24,random_state=2)
test = df1.drop(train.index)
fig, ax = plt.subplots(figsize=(15,6), ncols=2)
xticks=np.arange(df1.min()['X'],df1.max()['X']+1,0.1)
yticks=np.arange(df1.min()['Y'],df1.max()['Y']+1,0.1)

train.reset_index(drop=True, inplace=True)
train.to_csv('tree_train_set.csv')
test.plot.scatter(x='X',y='Y',c='Target', xlim=[5,30],colormap='RdBu', ax=ax[0])
train.plot.scatter(x='X',y='Y',c='Target', xlim=[5,30],colormap='RdBu',
↪ax=ax[1])
```

```
[2]: <AxesSubplot:xlabel='X', ylabel='Y'>
```



```
[4]: from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn import tree

fig2, ax2 = plt.subplots(figsize=(24,10), ncols=2)

train_data = train.iloc[:,0:2].to_numpy()
train_target = train.iloc[:,2:3].to_numpy()
test_data = test.iloc[:,0:2].to_numpy()
clf = DecisionTreeClassifier(random_state=0, criterion='gini', max_depth=5,
    ↳min_samples_split=2, min_samples_leaf=1)
clf = clf.fit(train_data, train_target)
prediction = clf.predict(test_data)
print(f"예측결과 : {prediction}")
print(f"실제값: {test['Target'].to_numpy()}")

xx, yy = np.meshgrid(xticks, yticks)
Z = clf.predict(np.c_[xx.ravel(), yy.ravel()]).reshape(xx.shape)
fig2.tight_layout(h_pad=0.5, w_pad=0.5, pad=2.5)
cs = ax2[0].contourf(xx, yy, Z, cmap=plt.cm.RdBu, alpha=0.3)
train.plot.scatter(x='X',y='Y',c='Target', colormap='RdBu', ax=ax2[0])

tree.plot_tree(clf, filled=True)
plt.show()
```

예측결과 : [0 0 0 1 1 1]

실제값: [0 0 0 1 1 1]

