20200720

PART 4.2-4.6

데이터 전처리 : 범주형 데이터 세트 나누기, 스케일 맞추기, 유용한 특성 선택

ESAA 1기 유찬미, ESAA 2기 장현정, ESAA 2기 정혜안

목차

4.2

범주형 데이터 다루기

범주형데이터

순서가 있는 특성 ex) 티셔츠 사이즈 XL>L>M

순서가 없는 특성 ex) 티셔츠 컬러 빨강? 파랑? 노랑?

순서 특성 매핑이 필요한 이유?

학습 알고리즘이 순서 특성을 올바르게 인식하기 위해서

순서 특성 매핑 하는 법

문자열 값 -> 정수
XL>L>M -> 3>2>1

클래스 레이블 인코딩이 필요한 이유?

많은 머신 러닝 라이브러리가
클래스 레이블이 정수로 인코딩되어 있을 거라 기대하기 때문에

클래스 레이블 인코딩 하는 법

class1, class2 -> 0, 1
클래스 레이블 => 순서가 없음을 기억

blue = 0, green = 1, red = 2
문제 : red>green>blue 이런 순서가 생김
해결책 : 원-핫 인코딩

원-핫 인코딩?

순서 없는 특성에 들어 있는 고유한 값마다 새로운 더미 특성을 만드는 것
ex) blue -> blue=1, green=0, red=0
green -> blue=0, green=1, red=0
red -> blue=0, green=0, red=1

원-핫 인코딩에서의 다중 공선성 문제

color		color_blue	color_green	color_red
1	→	0	1	0
2		0	0	1
0		1	0	0

-> 특성 간의 상관관계가 높은 다중 공선성이 발생할 수 있음

color		color_green	color_red
1	→	1	0
2		0	1
0		0	0

-> 이는 특성 열 하나를 삭제하여 상관관계를 감소시킬 수 있음

4.3

데이터셋을 훈련 세트와 테스트 세트로 나누기

실제 모델에 투입 전 테스트 세트와 예측을 비교함

-> 편향되지 않은 성능을 측정하기 위해서

실전에서 가장 많이 사용하는 비율

보통의 데이터셋)

훈련 세트 : 테스트 세트 = 60:40 or 70:30 or 80:20

대용량의 데이터셋)

훈련 세트 : 테스트 세트 = 90:10 or 99:1

4.4

특성 스케일 맞추기

특성 스케일 조정의 중요성

첫 번째 특성의 스케일 : 1~10
두 번째 특성의 스케일 : 1~100,000

아달린의 제곱 오차 함수의 경우

두 번째 특성에 대한 큰 오차에 맞추어 가중치 최적화

k-최근접 이웃의 경우

샘플 간의 거리 계산 시 두 번째 특성 축에 좌우될 것

특성 스케일 조정의 대표적인 방법

1) 정규화

- 최소-최대 스케일 변환의 특별한 경우로 특성의 스케일을 [0, 1] 범위에 맞추는 것

$$x_{norm}^{(i)} = \frac{x^{(i)} - x_{\min}}{x_{\max} - x_{\min}}$$

$x_{norm}^{(i)}$: 새로운 값

$x^{(i)}$: 샘플

$x_{\min}$: 특성 중에서 가장 작은 값

$x_{\max}$: 특성 중에서 가장 큰 값

- 범위가 정해진 값이 필요할 때 유용하게 사용할 수 있는 기법

특성 스케일 조정의 대표적인 방법

2) 표준화

- 특성의 평균을 0에 맞추고 표준편차를 1로 만들어 정규 분포와 같은 특징을 가지도록 만들어 가중치를 더 쉽게 학습할 수 있도록 만드는 것

$$x_{std}^{(i)} = \frac{x^{(i)} - \mu_x}{\sigma_x}$$

μ_x : 어떤 특성의 샘플 평균

σ_x : 샘플 평균에 해당하는 표준 편차

- 경사 하강법 같은 최적화 알고리즘에 널리 사용
- 이상치 정보가 유지되어 정규화에 비해 알고리즘이 이상치에 덜 민감

4.5

유용한 특성 선택

4.5 유용한 특성 선택

- 모델의 성능이 test set보다 train set에서 높음 ➔ **과대적합** (모델이 train set에 너무 잘 맞춰져 있어서 새로운 데이터에서 일반화 힘듦)
- 과대적합을 피하고 train data 밖에서도 높은 성능을 보이기 위해서는 **모델의 복잡도를 낮출 필요**가 있음
- 모델이 복잡하다는 것은 불필요한 항이 많아졌다는 것을 의미함

일반화 오차(과대적합)를 감소하기 위한 방법

1. 더 많은 train data 수집 → 불가능한 경우 많음
   :데이터의 양이 적을 경우, 해당 데이터의 특정 패턴이나 노이즈까지 쉽게 암기하게 되므로 과적합 현상이 발생할 가능성이 높아짐.
2. **모델 규제를 통해 복잡도 제한:** L1규제, L2규제 등을 통해
3. Parameter 수가 적은 간단한 모델 선택
4. **데이터의 차원 축소:** Feature selection 등을 통해

4.5.1 모델 복잡도 제한을 위한 L1 규제와 L2 규제

- **규제**: 모델을 단순하게 하고 과대적합의 위험을 감소시키기 위해 모델에 제약을 가하는 것. 가중치의 모든 원소를 0에 가깝게 하여 모든 특성이 출력에 주는 영향을 최소한으로 만듦.

- 기존의 비용함수에 규제를 추가한 후 비용함수를 최소화하는 방향으로 계산하면 모델의 복잡도를 줄일 수 있음
- 비용 함수를 최소화하기 위해서는 가중치 w들의 값이 작아져야 함
- 학습 과정에서 큰 가중치에 대해서 그에 상응하는 큰 패널티를 부과하여 과대적합을 억제. 과대적합은 가중치 매개변수의 값이 커서 발생하는 경우가 많기 때문
- 학습하는 동안 적용할 규제의 강도는 하이퍼파라미터(λ 등)가 결정
- λ 가 크다면 모델이 훈련 데이터에 대해서 적합한 매개 변수를 찾는 것보다 규제를 위해 추가된 항들을 작게 유지하는 것을 우선한다는 의미

4.5.2 L2 규제

L2규제 (L2 regularization)

$$L2 : \|\boldsymbol{w}\|_2^2 = \sum_{j=1}^{m} w_j^2 \quad \rightarrow \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

: 기존의 비용함수에 규제항 추가 (Ridge 회귀)

- 모든 가중치 w들의 제곱합을 비용 함수에 추가
- 규제가 없는 비용함수에 비해 규제항을 추가한 비용함수는 가중치 값을 아주 작게 만듦
- 목적: train data에서 비용함수를 최소화하는 가중치 값의 조합을 찾는 것

4.5.3 L1 규제

L1 규제 (L1 regularization)

$$L1: \|w\|_1 = \sum_{j=1}^{m} |w_j| \quad \rightarrow \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

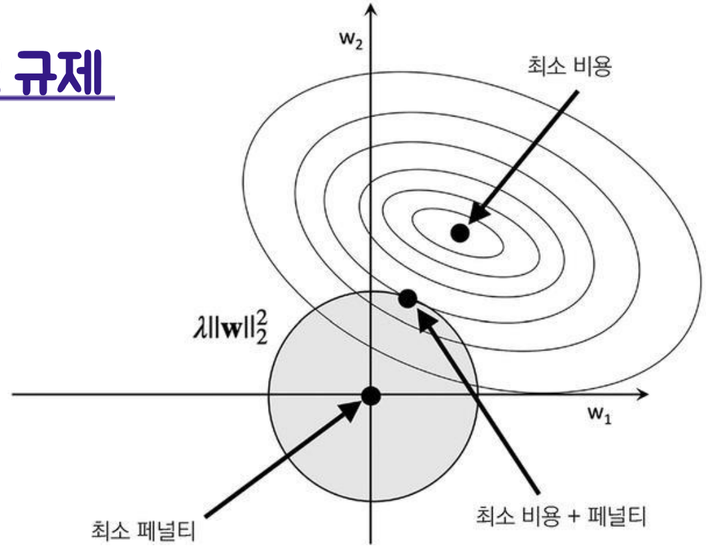
: 기존의 비용함수에 규제항 추가 (Lasso 회귀)

- 가중치 w들의 절대값 합계를 비용 함수에 추가
- L2 규제와 대조적으로 L1 규제는 보통 희소한 특성 벡터를 생성.
- 관련 없는 특성이 많은 고차원 데이터셋의 경우 이런 희소성이 도움이 됨
- L2 규제와는 달리 어떤 가중치는 실제로 0이 됨. 즉, 모델에서 완전히 제외되는 특성이 생김.

  ➔ 일부 계수를 0으로 만듦으로써 모델을 이해하기 쉬워지고, 모델의 가장 중요한 특성이 무엇인지 드러남
- L1 규제는 어떤 특성들이 모델에 영향을 주고 있는지를 정확히 판단하고자 할 때 유용
- 그러나 L2 규제가 L1 규제에 비해 더 안정적이라 일반적으로는 L2규제가 더 많이 사용

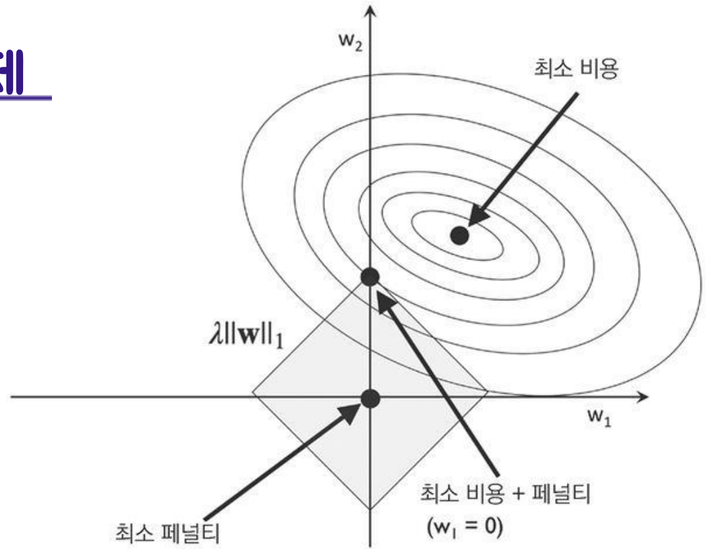
목표는 규제가 없는 비용과 페널티 항의 합을 최소화하는 것!!

L2 규제



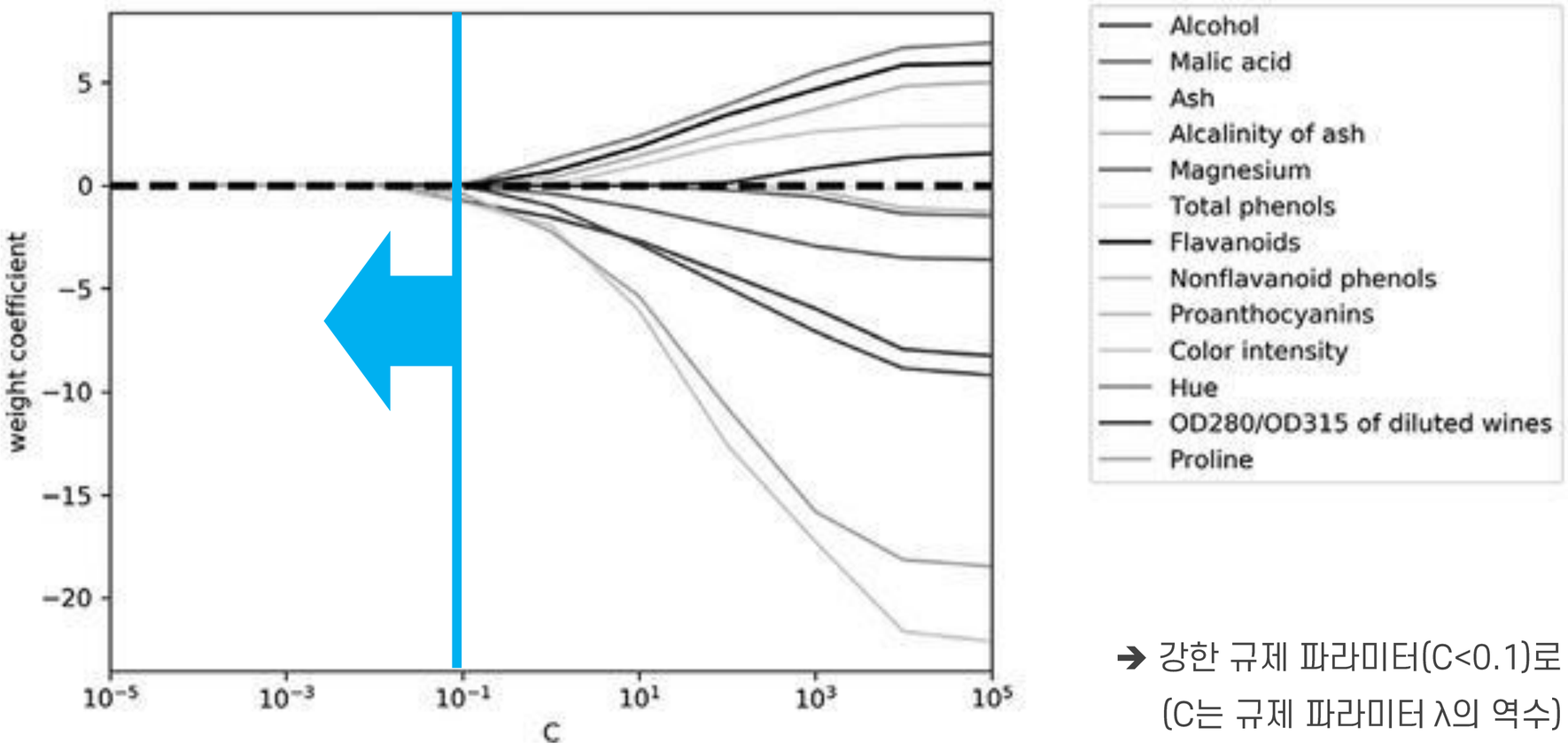
- L2 규제항 → 회색 원
- 가중치 값은 규제 예산을 초과할 수 없으므로 가중치값의 조합이 회색 원 밖에 놓일 수 없음.
- 패널티가 주어진 상황에서 최선은 L2 회색 공과 규제가 없는 비용 함수의 등고선이 만나는 지점
- 규제 파라미터 λ가 커질수록 회색공이 작아짐

L1 규제



- L1 규제항 → 회색 다이아몬드
- w1 = 0일 때 비용 함수의 등고선이 L1 다이아몬드와 만남
- L1 규제의 등고선은 날카롭기 때문에 비용 함수의 포물선과 L1 다이아몬드의 경계가 만나는 최적점은 축에 가깝게 위치할 가능성이 높음 (가중치가 0이 됨)

4.5.3 L1 규제를 사용한 희소성



➔ 강한 규제 파라미터(C<0.1)로 모델을 제약하면 모든 가중치가 0이 됨
(C는 규제 파라미터 λ의 역수)

4.5.4 순차 특성 선택 알고리즘

과대적합을 피하는 다른 방법: **차원 축소**

- 특성 선택 (feature selection): 원본 특성에서 일부를 선택
- 특성 추출 (feature extraction): 일련의 특성에서 얻은 정보로 새 특성 만듦

순차 특성 선택 (Sequential Feature Selection

- d차원의 특성 공간을 k차원의 특성 부분 공간으로 축소 (k<d)
- 목적: 주어진 문제에 가장 관련이 높은 특성 부분 집합을 자동으로 선택하는 것
- 관계없는 특성이나 잡음을 제거하여 계산 효율성을 높이고 모델의 일반화 오차를 줄임
- 대표적으로 순차 후진 선택 (Sequential Backward Selection, SBS) 알고리즘

4.5.4 순차 특성 선택 알고리즘

순차 후진 선택 (Sequential Backward Selection, SBS)

- 계산 효율성을 향상하기 위해 모델 성능을 가능한 적게 희생하면서 초기 특성의 부분 공간으로 차원을 축소
- 전체 특성을 모두 선택한 뒤 순차적으로 목표하는 특성 개수가 될 때까지 하나씩 특성 제거
- 각 단계에서 어떤 특성을 제거할지 판단하기 위해 최소화할 기준 함수를 정의
- 기준함수는 특성 제거 전후의 모델 성능 차이를 계산
- 각 단계에서는 제거했을 때 성능 손실이 최대가 되는 특성 제거 (기준값이 가장 큰 특성 제거)

<순차 후진 선택의 단계>

1. 알고리즘을 k=d로 초기화. d는 전체 특성 공간 X_d의 차원.
2. 조건 x-=arg max J(X_k -x)를 최대화하는 특성 x-를 결정. (x ∈ X_k)
3. 특성 집합에서 특성 x-를 제거 $X_{k-1} := X_k - x^-; k := k-1$
4. 다시 2단계로 돌아가 특성 하나씩 제거. k가 목표하는 특성 개수가 되면 종료

4.6

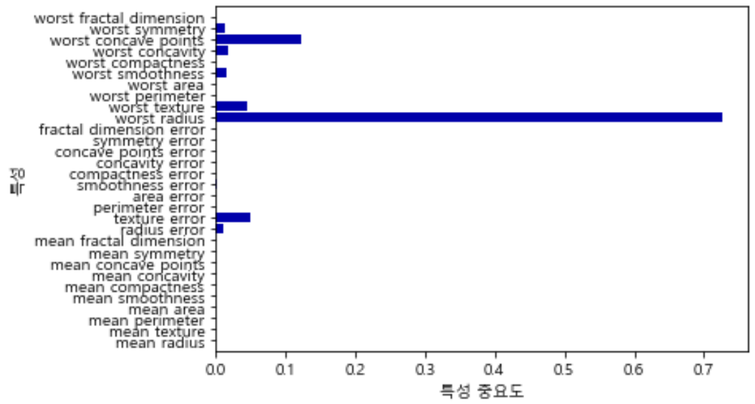
랜덤 포레스트와 특성 중요도 사용

4.6 랜덤 포레스트의 특성 중요도 사용

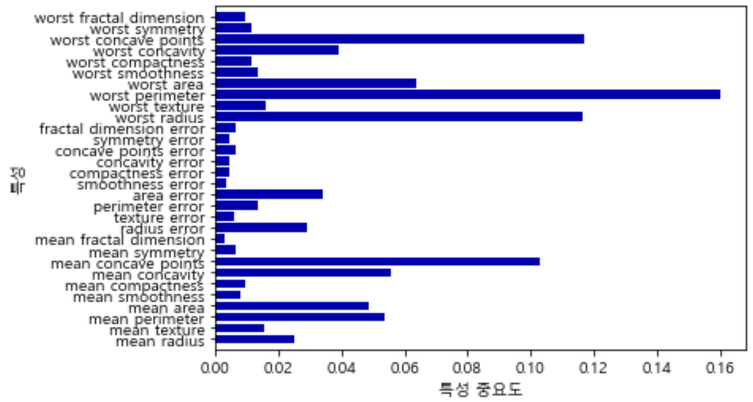
특성 선택하는 다른 방법: **랜덤 포레스트**의 Feature selection

- 앙상블에 참여한 모든 결정 트리에서 계산한 평균적인 불순도 감소로 특성 중요도를 측정할 수 있음
- 특징: 랜덤 포레스트에서 두 개 이상의 특성이 매우 상관관계가 높다면 하나의 특성은 매우 높은 순위를 갖지만 다른 특성 정보는 완전히 잡아내지 못할 수 있음. 따라서 특성 중요도 값을 해석하는 것보다 모델의 예측 성능에 관심이 있을 경우에 사용하기 적절
- 단일 결정 트리보다 훨씬 많은 특성이 0이상의 중요도를 가짐

단일 결정 트리



랜덤 포레스트



발표를 들어주셔서

감사합니다 :)