CHAPTER4. 모델 훈련

- 선형 회귀
- 경사 하강법
- 다항 회귀
- 학습 곡선
- 규제가 있는 선형모델
- 로지스틱 회귀

4.1 선형 회귀 (Linear Regeression)

4.1.1 선형 회귀 모델

$$\hat{y} = h_\theta(X) = \theta^T \cdot x$$

- $h_\theta(X)$는 모델 Parameter(모수) θ를 사용한 가설(hypothesis) 함수입니다.
- θ는 편향 θ_0 에서 θ_n까지 모델의 각 변수별 Parameter(모수)를 담은 벡터입니다.
- θ^T는 θ의 전치입니다. (열 벡터가 아닌 행 백터)
- x는 x_0 에서 x_n까지 담고 있는 샘플의 특성 벡터입니다. x_0은 항상 1입니다.

4.1.2 성능 측정 지표 : 평균제곱오차 (Mean Square Error, MSE)

$$MSE(X, h_\theta) = \frac{1}{m}\sum_{i=1}^{m}(\theta^T \cdot x^{(i)} - y^{(i)})^2$$

- 회귀 모델에서 가장 많이 사용되는 성능 측정 지표는 RMSE입니다.
- 하지만 실제로는 RMSE보다 MSE(평균제곱오차)를 최소화 하는것이 같은 결과를 내면서 더 간단합니다.
- 간단한 이유는 어떤 함수를 최소화 하는 것은 그 함수의 제곱근을 최소화 하는것과 같기 때문입니다.

4.1.3 정규방정식 (Normal Equation)

- 선형 회귀의 목표는 최적의 θ를 찾는 것입니다.
- 최적의 θ를 찾기 위해선 비용함수 (Cost Fucntion)을 최소화해야 합니다.
- 비용함수를 최소화하는 방법에는 크게 정규방정식 / 경사하강법을 이용합니다.

$$\hat{\theta} = (X^T \cdot X)^{-1} \cdot X^T \cdot y$$

- $\hat{\theta}$은 비용함수를 최소화하는 θ값입니다.
- y는 $y^{(1)}$ 부터 $y^{(m)}$ 까지 포함하는 타깃 벡터입니다.
- 정규방정식의 학습된 선형 회귀 모델은 예측이 매우 빠릅니다.
- 예측 계산 복잡도는 샘플 수와 특성 수에 선형적입니다.

4.2 경사하강법 (Gradient Descent, GD)

- 경사하강법은 여러 종류의 문제에서 최적의 해법을 찾을 수 있는 매우 일반적인 최적화 알고리즘입니다.
- 경사하강법의 기본 아이디어는 비용함수를 최소화하기 위해 반복해서 파라미터를 조정해가는 것입니다.
- 볼록함수(x = θ, y = cost)에서 임의의 점을 찍고 그 함수의 꼭지점으로 점을 내려보냅니다.
- 볼록함수의 꼭지점이 cost가 최소값이 되는데 이때의 θ가 최적의 파라미터입니다.
- 경사하강법에서 중요한 파라미터는 스텝의 크기로 학습률(Learning rate) 하이퍼파라미터로 결정됩니다.
- 학습률이 너무 작으면 꼭지점으로 가는데 너무 오래걸리고, 학습률이 너무 크면 꼭지점을 지나칠 수 있습니다.
- 경사하강법은 특성이 매우 많고 훈련 샘플이 너무 많아 메모리에 담을 수 없을 때 적합합니다.

- 경사하강법의 스텝 (η = 학습률 (Learning rate))

$$\theta^{(next\ step)} = \theta - \eta \nabla_\theta\ MSE(\theta)$$

- 비용함수의 그래디언트 벡터

$$\nabla_\theta\ MSE(\theta) = \frac{2}{m} X^T \cdot (X \cdot \theta - y)$$

4.2.1 배치 경사하강법 (Batch Gradient Descent)

- 매 경사하강법 스텝에서 전체훈련 세트 X에 대해 계산하는 방법입니다.
- 안정성이 높은 장점이 있지만 시간이 오래걸리는 단점이 있습니다.

4.2.2 학률적 경사 하강법

- 매 스텝에서 딱 한개의 샘플을 무작위로 선택하고 그 하나에 대한 그레디언트를 계산하는 방법입니다.
- 매우 큰 훈련세트에서 학습이 가능하고 시간이 빠른 장점이 있지만 안정성이 낮은 단점이 있습니다.
- 일반적으로 한 반복에서 m번 되풀이되고, 이때 각 반복을 에포크 (epoch)라고 합니다.

4.2.3 미니배치 경사하강법 (Mini Batch Gradient Descent)

- 미니배치라 부르는 임의의 작은 샘플 세트에 대해 Gradient Descent를 계산하는 방법입니다.
- GPU를 사용하면 성능이 향상됩니다.

알고리즘	훈련샘플수가 클때	외부메모리 학습지원	특성수가 클때	하이퍼파라미터 수	스케일 조정 필요	사이킷 런
정규방정식	빠름	NO	느림	0	NO	LinearRegression
배치 경사하강법	느림	NO	빠름	2	YES	n/a
확률적 경사하강법	빠름	YES	빠름	≥ 2	YES	SGDRegressor
미니배치 경사하강법	빠름	YES	빠름	≥ 2	YES	n/a

4.3 다항 회귀 (Polynomial Regression)

- 가지고 있는 데이터가 직선이 아닌 복잡한 형태일 경우 사용하는 방법이 다항 회귀 입니다.
- 제곱이나 세제곱 등을 추가시켜 비선형 데이터에도 적용 가능하도록 합니다.
- 다항회귀를 통해 차수를 높이면 모델의 정확도는 높아지지만 Overfitting (과최적화)의 위험이 있습니다.

4.4 학습 곡선 (Learning Curve)

- 다항회귀는 Overffiting의 가능성을 높이는데 실제 모델이 얼마나 Oveffing이 됐는지 확인하는 방법입니다.
- 학습 곡선은 Train Data와 Test Data의 모델 성능을 훈련 세트 크기의 함수로 나타냅니다.
- Train Data에서 크기가 다른 서브 Data를 만들어 모델을 여러 번 훈련시킵니다.
- 과소적합 모델의 경우는 Train / Test 곡선이 수평한 구간을 만들고 꽤 높은 오차에서 매우 가까이 근접해 있습니다.
- 과대적합 모델의 경우는 Train의 오차가 선형회귀모델보다 훨씬 낮고 두 곡선 사이에 공간이 있습니다.

*** 편향 / 분산 트레이드 오프**

1) 편향 (과소적합)

- 일반화 오차 중에서 편향은 잘못된 가정으로 인한 것입니다.
- 예를 들면 데이터가 2차인데 선형으로 가정하는 경우 등이 있고 편향이 큰 모델은 Train Data에 과소적합되기 쉽습니다.

2) 분산 (과대적합)

- 훈련 데이터에 있는 작은 변동에 모델이 과도하게 민감하기 때문에 나타납니다.
- 자유도가 높은 모델이 높은 분산을 가지기 쉬워 훈련 데이터에 과대적합되는 경향이 있습니다.

3) 줄일 수 없는 오차

- 데이터 자체에 있는 노이즈 때문에 발생합니다.
- 해결하는 유일한 방법은 데이터에서 노이즈를 제거하는 것입니다.

→ 모델이 복잡해지면 분산이 늘어나고 편향은 줄어듭니다. 반대로 모델이 단순해지면 분산이 감소하고 편향이 커집니다. 그래서 트레이드오프라 합니다.

4.5 규제가 있는 선형 모델

- 과대적합을 막는 가장 효율적인 방법은 모델을 규제하는 것입니다.
- 일반적으로 회귀모델을 규제하기 위해서 가중치를 제한하는 방법이 있습니다.

4.5.1 릿지 (Ridge)

- 가중치들의 제곱합을 최소화하는 것을 추가적인 제약조건으로 하는 방법입니다.
- 기본적으로 사용하는 방법입니다.

$$J(\theta) = MSE(\theta) + \alpha\frac{1}{2}\sum_{i=1}^{n}\theta_i^2$$

- α를 통해 모델을 얼마나 규제할지 조정할 수 있습니다.
- α가 크면 규제가 커지고 α가 작으면 규제가 작아집니다.

4.5.2 라쏘 (Lasso)

- 가중치의 절대값의 합을 추가적인 제약조건으로 하는 방법입니다.
- 실제 쓰이는 특성의 개수가 얼마 없을 때 사용합니다.

$$J(\theta) = MSE(\theta) + \alpha\frac{1}{2}\sum_{i=1}^{n}|\theta_i|$$

- 규제의 의미는 덜 중요한 특성의 가중치를 완전히 제거하는 것입니다.
- 자동으로 특성을 선택하고 희소모델을 만듭니다.

4.5.3 엘라스틱 넷 (Elastic Net)

- 릿지와 라쏘를 결합한 형태입니다.
- 특성 수가 훈련샘플 수보다 크거나 특성 몇개가 강하게 연관되어 있을 때 사용합니다.

$$J(\theta) = MSE(\theta) + r \cdot \alpha \frac{1}{2} \sum_{i=1}^{n} |\theta_i| + \frac{(1-r)}{2} \cdot \alpha \frac{1}{2} \sum_{i=1}^{n} \theta_i^2$$

- $r = 1$ 이면 라쏘, $r = 0$ 이면 릿지를 의미합니다.

4.5.4 조기 종료

- 에러가 최솟값에 도달하면 바로 훈련을 중지하는 방법입니다.

4.6 로지스틱 회귀 (Logistic Regression)

- 샘플이 특정 클래스에 속할 확률을 추정하는데 사용합니다.
- 확률이 50% 이상이면 클래스에 속하고 50%보다 작으면 속하지 않다고 판단합니다.

4.6.1 로지스틱 회귀모델의 확률추정

$$\hat{p} = h_\theta(x) = \sigma(\theta^T \cdot x)$$

- 로지스틱 함수

$$\sigma(t) = \frac{1}{1 + exp(-t)}$$

- 로지스틱 함수는 0과 1사이의 값을 출력하는 시그모이드 함수입니다. (S자 형태)

4.6.2 훈련과 비용 함수

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} [y^{(i)} \ log(\hat{p}^{(i)}) + (1 - y^{(i)}) \ log(1 - \hat{p}^{(i)})]$$

4.6.3 소프트맥스 회귀 (Softmax Regerssion)

- 여러 개의 이진분류문제를 훈련시켜 연결하지 않고 직접 다중 클래스를 지원하도록 해줍니다.

$$\hat{p_k} = \sigma(S(X))_k = \frac{exp(S_k(X))}{\sum_{j=1}^{k} exp(s_j(X))}$$

- k 는 클래스 수를 의미합니다.

- $\sigma(S(X))_k$는 클래스에 속할 확률을 의미합니다.
- $s_j(X)$는 클래스의 점수를 의미합니다.

$$\hat{y} = argmax\ \sigma(S(X))_k = argmax\ S_k(X) = argmax\ ((\theta^{(k)})^T X)$$

- 각 클래스의 확률을 비교해 가장 큰 확률의 클래스를 선택합니다.

4.6.4 엔트로피

1) 엔트로피 (Entropy)

- 확률 분포들이 가지는 확신의 정도를 수치로 표현하는 것을 의미합니다.
- 물리학에선 상태가 분산되어 있는 정도를 엔트로피로 정의합니다.
- 고루 분산되어 있으면 엔트로피가 높고 하나의 상태로 몰려있으면 엔트로피가 낮습니다.

1)-1 이산형 엔트로피

$$H[Y] = -\sum_{k=1}^{K} p(y_k) \log_2 p(y_k)$$

1)-2 연속형 엔트로피

$$H[Y] = -\int p(y) \log_2 p(y)\ dy$$

2) 엔트로피 성질

- 엔트로피의 최소값은 0입니다. (하나의 값이 나올 확률이 1일 때)
- 엔트로피의 최대값은 이산확률 변수의 클래스에 따라 다릅니다.
- 클래스의 개수가 2^K 이면 최대값은 K입니다.

$$H = -\frac{2^K}{2^K} \log_2 \frac{1}{2^K} = K$$

3) 엔트로피와 정보량

- 엔트로피가 0이면 확률변수는 결정론적이므로 확률변수의 표본값이 변화하지 않습니다.
- 따라서 확률 변수의 표본값을 관측한다고 해도 우리가 얻을 수 있는 추가 정보는 없습니다.

- 엔트로피가 크면 확률변수의 표본값이 가질 수 있는 실질적인 경우의 수가 증가합니다.
- 따라서 표본값을 실제로 관측하기 전까지는 아는것이 없다는 의미입니다.
- 이는 확률변수의 표본값이 우리에게 가져다 줄 수 있는 정보량이 많다는 뜻입니다.

4) 크로스 엔트로피

- 크로스 엔트로피는 주로 분류 문제의 목표값 분포와 예측값 분포를 비교하는데 사용합니다.

4)-1 크로스 엔트로피 이산형

$$H[p,q] = -\sum_{k=1}^{K} p(y_k)\log_2 q(y_k)$$

4)-2 크로스 엔트로피 연속형

$$H[p,q] = -\int p(y)\log_2 q(y)dy$$

4)-3 크로스 엔트로피 비용함수

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k} y_k^{(i)} log(\hat{p_k}^{(i)})$$

4)-4 크로스 엔트로피 Gradient Vector

$$\nabla_{\theta^{(k)}} J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(\hat{p_k}^{(i)} - y_k^{(i)})\ X^{(i)}$$

5) 쿨백-라이블러 발산 (Kullback - Leibler divergence)

- 쿨백-라이블러 발산은 두 확률분포 $P(y), q(X)$의 차이를 정량화하는 방법입니다.

$$KL(p||q) = H[p,q] - H[p] \ = \int p(y)\log_2\left(\frac{p(y)}{q(y)}\right)dy$$

- 쿨백-라이블러 발산은 크로스 엔트로피에서 대상이 되는 분포의 엔트로피를 뺀 값입니다.
- 따라서 상대 엔트로피라고도 불리며 값은 항상 양수입니다.
- 그리고 두 확률분포가 완전히 같으면 0이 됩니다.