모델 훈련

박종민(jijupax@gmail.com)

May 19, 2020

학습 목표

- 선형 회귀를 알아봅니다.
  - 직접 계산할 수 있는 공식을 사용
  - 경사 하강법을 사용
- 비선형 데이터셋을 훈련시킬 수 있는 다항 회귀를 알아봅니다.
- 과대접합을 감지하는 방법을 알아봅니다.
- 과대적합을 감소시킬 수 있는 규제 방법을 알아봅니다.
- 로지스틱 회귀와 소프트맥스 회귀를 알아봅니다.

$$\begin{aligned} \hat{y} &= \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n \\ &= h_\theta(\mathbf{x}) = \theta \cdot \mathbf{x} \end{aligned} \tag{1}$$

- θ는 선형 회귀 모델의 파라미터이고, 각 특성과 곱해집니다.
- 식 1과 같이 입력 특성의 가중치 합과 편향의 합으로 표현할 수 있습니다.

$$\mathrm{MSE}(\mathbf{X}, h_\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(\theta^T \mathbf{x}^{(i)} - y^{(i)}\right)^2 \qquad (2)$$

- 회귀 모델을 훈련시키는 방법은 식 2와 같이 비용 함수인 평균 제곱 오차(MSE)를 구해 그 값을 최소화하는 것입니다.

$$\hat{\theta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y} \tag{3}$$

- 비용 함수를 최소화하는 θ를 찾기 위한 해적적인 방법을 식 3 과 같이 정규방정식이라고 합니다.

```python
theta_best = np.linalg.inv(X_b.T.dot(X_b)).dot(X_b.T).dot(y)

from sklearn.linear_model import LinearRegression
lin_reg = LinearRegression()
lin_reg.fit(X, y)

theta_best_svd, ... = np.linalg.lstsq(X_b, y, rcond=1e-6)
```

- numpy를 사용하여 식 3을 직접 계산 하는 것이 가능합니다.
- sklearn의 `LinearRegression` 또는 `numpy.linalg.lstsq` 함수를 사용해도 됩니다.

Gradient Descent

비용 함수를 최소화하기 위해 반복해서 파라미터를 조정해가는 방법입니다.

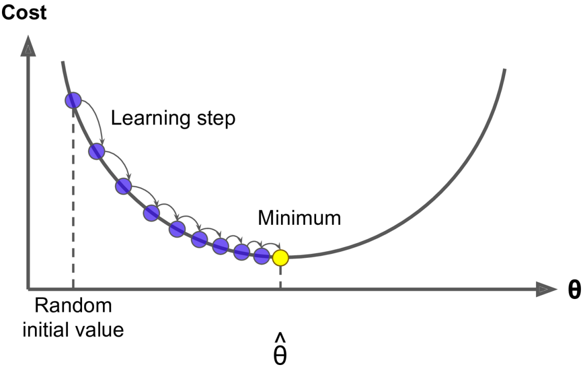

Figure 1: 경사 하강법

경사 하강법

- 파라미터 벡터 θ에 대해 비용 함수의 현재 그레디언트를 계산하여 감소하는 방향으로 진행합니다.
- θ를 임의의 값으로 시작해서 한 번에 조금씩 비용 함수가 감소되는 방향으로 진행합니다.
- 학습률의 크기에 따라 알고리즘의 수렴 속도가 틀리고, 클 경우 발산하거나 작을 경우 지역 최소값에 빠지기도 합니다.

$$\frac{\partial}{\partial \theta_j}\mathrm{MSE}(\boldsymbol{\theta}) = \frac{2}{m}\sum_{i=1}^{m}(\boldsymbol{\theta}^T\mathbf{x}^{(i)} - y^{(i)})\, x_j^{(i)}$$
$$\nabla_{\boldsymbol{\theta}}\,\mathrm{MSE}(\boldsymbol{\theta}) = \frac{2}{m}\mathbf{X}^T(\mathbf{X}\boldsymbol{\theta} - \mathbf{y}) \tag{4}$$

- 매 경사 하강법 스텝에서 전체 훈련 세트 **X**에 대해 계산하는 방법입니다.
- 식 4는 배치 경사 하강법의 비용함수 그레디언트 벡터를 나타내는 것입니다.
- 경사 하강법의 스텝: $\boldsymbol{\theta}^{(\text{다음 스텝})} = \boldsymbol{\theta} - \eta\nabla_{\boldsymbol{\theta}}\,\mathrm{MSE}(\boldsymbol{\theta})$

확률적 경사 하강법

- 매 스텝에서 딱 1개의 샘플을 무작위로 선택하여 그래디언트를 계산하는 방법입니다.
- 매우 적은 데이터를 처리하므로 배치 경사 하강법에 비해 훨씬 빠르고, 메모리를 적게 차지합니다.
- 무작위로 샘플을 선택하므로 전역 최소값 주변을 맴돌면서 수렵하게 됩니다.
- 학습 스케쥴을 조절하여 주변을 맴도는 문제를 해결할 수 있습니다.

미니 배치 경사 하강법

- 미니배치라고 부르는 임의의 작은 샘플 세트에 대해 그래디언트를 계산하는 방법입니다.
- 미니배치 크기를 적당히 조절하면 확률적 경사 하강법보다 덜 불규칙하게 수렴하게 됩니다.
- GPU를 사용한 병렬 계산에 유리한 방법입니다.

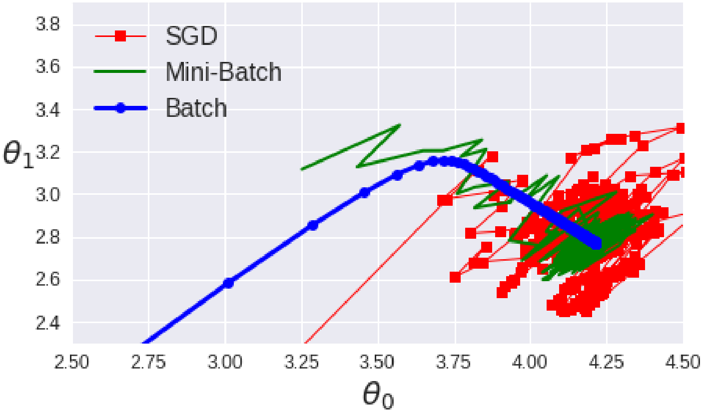

Figure 2: 파라미터 공간에 표시된 경사 하강법의 경로

선형 회귀 알고리즘 비교

알고리즘	m이 클 때	외부 메모리 학습 지원	n이 클 때	하이퍼파라미터 수	스케일조정 필요
정규방정식	빠름	No	느림	0	No
SVD	빠름	No	느림	0	No
GD	느림	No	빠름	2	Yes
SGD	빠름	Yes	빠름	≥ 2	Yes
Mini-batch	빠름	Yes	빠름	≥ 2	Yes

Table 1: 선형 회귀를 사용한 알고리즘 비교

- 각 특성의 거듭 제곱을 새로운 특성으로 추가하여 확장된 특성을 포함한 데이터셋에 대해 선형 모델을 훈련 시키는 방법입니다.

```
from sklearn.preprocessing import PolynomialFeatures
poly_ = PolynomialFeatures(degree=2, include_bias=False)
X_poly = poly_.fit_trasform(X)

lin_reg = LinearRegression()
lin_reg.fit(X_poly, y)
```

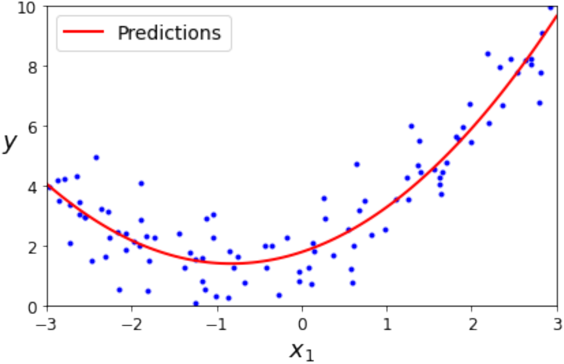

Figure 3: 다항 회귀 모델의 예측

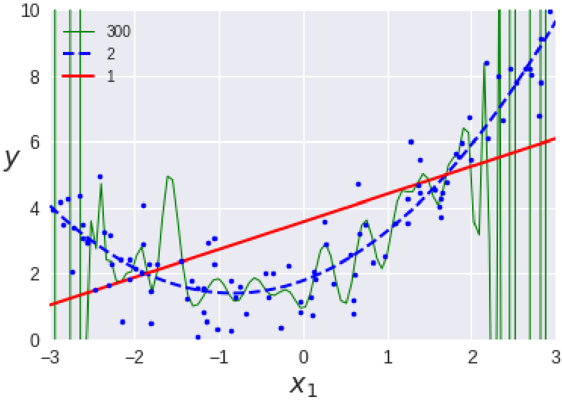

Figure 4: 고차 다항 회귀

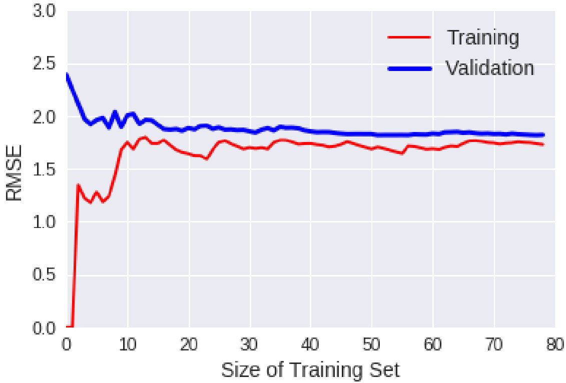

Figure 5: 선형 회귀 학습 곡선

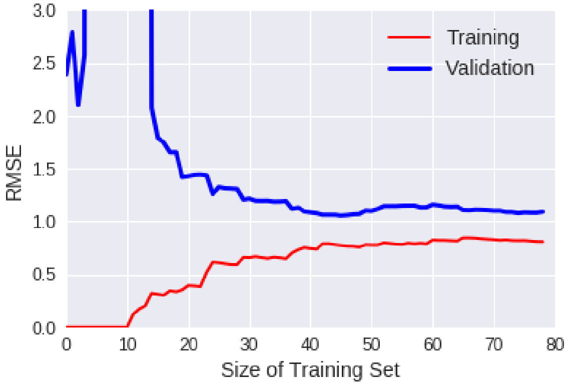

Figure 6: 다항 회귀 학습 곡선

모델의 일반화 오차

- 편향
  - 잘못된 가정으로 발생됩니다.
  - 과소적합이 되기 쉬습니다.
- 분산
  - 훈련 데이터에 있는 작은 변동에 모델이 과도하게 민감하기 때문에 발생합니다.
  - 자유도가 높은 모델이 높은 분산을 가지기 쉬워 과대 적합 경향이 있습니다.
- 줄일 수 없는 오차
  - 데이터 자체에 있는 노이즈 때문에 발생합니다.
  - 데이터에서 노이즈를 제거하는 방법이 유일합니다.
- 모델의 복잡도가 커지면 분산이 늘어나고 편향은 줄어듭니다.
- 모델의 복잡도가 줄어들면 분산은 작아지고 편향이 커집니다.

$$J(\boldsymbol{\theta}) = \mathrm{MSE}(\boldsymbol{\theta}) + \alpha \frac{1}{2} \sum_{i=1}^{n} \theta_i^2 \tag{5}$$

- 규제항 $\alpha \sum_{i=1}^{n} \theta_i^2$이 비용 함수에 추가된 것입니다.
- 규제항은 훈련하는 동안에만 비용 함수에 추가 되고, 테스트 시엔 사용하지 않습니다.
- $\alpha = 0$이면 선형 회귀, α가 아주 크면 모든 가중치가 거의 0에 가까워지고 데이터의 평균을 지나는 수평선이 됩니다.

$$J(\boldsymbol{\theta}) = \text{MSE}(\boldsymbol{\theta}) + \alpha \sum_{i=1}^{n} |\theta_i| \tag{6}$$

- Least Absolute Shrinkage and Selection Operator
- 규제항 $\alpha \sum_{i=1}^{n} |\theta_i|$이 비용 함수에 추가된 것입니다.
- 덜 중요한 특성의 가중치를 완전히 제거하는 특징이 있으므로 희소 모델을 만듭니다.

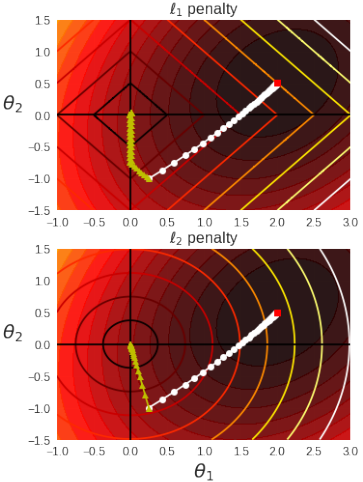

Figure 7: 라쏘 대 릿지 규제

$$J(\boldsymbol{\theta}) = \mathrm{MSE}(\boldsymbol{\theta}) + r\alpha \sum_{i=1}^{n} |\theta_i| + \frac{1-r}{2}\alpha \sum_{i=1}^{n} \theta_i^2, \quad 0 \le r \le 1 \quad (7)$$

- 릿지 회귀와 라쏘 회귀를 절충한 모델입니다.
- 혼합 비율 r을 사용해 조절하여 $r = 0$이면 릿지 회귀이고 $r = 1$이면 라쏘 회귀입니다.
- 릿지 회귀가 기본이 되지만 실제 쓰이는 특성이 몇 개뿐이라면 라쏘 회귀나 엘라스틱넷을 씁니다.
- 특성 수가 훈련 샘플 수보다 많거나 특성 몇개가 강하게 연관되어 있다면 엘라스틱넷을 씁니다.

조기 종료

- 검증 에러가 최소값에 도달하면 바로 훈련을 중지하는 방법입니다.
- 검증 에러가 증가하기 시작하면 과대 적합이 시작하는 것을 의미합니다.
- 검증 에러가 일정 시간 동안 최소값보다 크다는 확신이 들 때 학습을 멈추고 최소였을 때의 모델 파라미터로 되돌립니다.
- Beautiful free lunch, *Geoffrey Hinton*

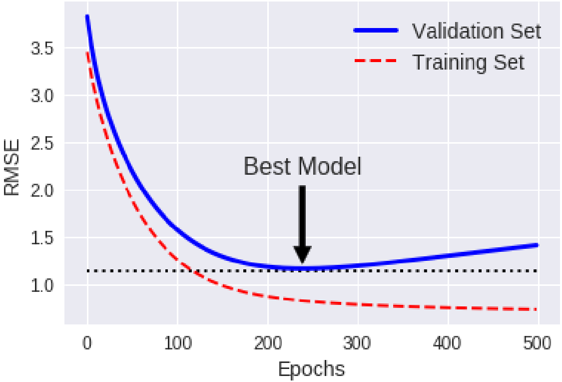

Figure 8: 라쏘 대 릿지 규제

- 샘플이 특정 클래스에 속할 확률을 추정하는데 사용합니다.
- 추정 확률이 50%가 넘으면 모델은 그 샘플이 해당 클래스에 속한다고 예측합니다.(양성 클래스)
- 아니면 클래스에 속하지 않는다고 예측합니다.(음성 클래스)

$$\hat{y} = \begin{cases} 0 & \hat{p} < 0.5\text{일 때} \\ 1 & \hat{p} \geq 0.5\text{일 때} \end{cases} \tag{8}$$

$$\hat{p} = h_{\boldsymbol{\theta}}(\mathbf{x}) = \sigma(\boldsymbol{\theta}^T \mathbf{x}) \tag{9}$$

$$\sigma(t) = \frac{1}{1 + \exp(-t)} \tag{10}$$

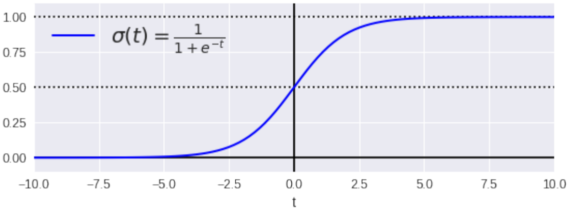

Figure 9: 시그모이드 함수

$$c(\boldsymbol{\theta}) = \begin{cases} -\log(\hat{p}) & y = 1\text{일 때} \\ -\log(1-\hat{p}) & y = 0\text{일 때} \end{cases} \tag{11}$$

$$J(\boldsymbol{\theta}) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} log\left(\hat{p}^{(i)}\right) + (1-y^{(i)}) log\left(1-\hat{p}^{(i)}\right)\right] \tag{12}$$

- 훈련의 목적은 $y = 1$에 대해서 높은 확률을 추정하고, $y = 0$에 대해서 낮은 확률을 추정하는 모델의 파라미터 벡터 θ를 찾는 것입니다.
- 정규방정식과 같은 로지스틱 회귀 비용 함수의 최소값을 계산하는 공식은 없습니다.
- $J(\boldsymbol{\theta})$는 볼록 함수 이므로 경사 하강법으로 전역 최소값을 찾을 수 있습니다.

소프트맥스 회귀

$$s_k(\mathbf{x}) = (\boldsymbol{\theta}^{(k)})^T \mathbf{x} \tag{13}$$

$$\hat{p}_k = \sigma\left(\mathbf{s}(\mathbf{x})\right)_k = \frac{\exp\left(s_k(\mathbf{x})\right)}{\sum_{j=1}^{K} \exp\left(s_j(\mathbf{x})\right)} \tag{14}$$

- 로지스틱 회귀 모델이 다중 클래스를 수행할 수 있도록 일반화된 것입니다.
- 샘플 $\mathbf{x}$가 주어지면 각 클래스 k에 대한 점수 $S_K(\mathbf{x})$를 계산하고, 그 점수에 소프트맥스 함수를 적용하여 각 클래스의 확률을 추정합니다.
- 한 번에 하나의 클래스만 예측 가능합니다.

크로스 엔트로피

$$J(\boldsymbol{\Theta}) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log\left(\hat{p}_k^{(i)}\right) \tag{15}$$

$$\nabla_{\boldsymbol{\theta}^{(k)}} J(\boldsymbol{\Theta}) = \frac{1}{m}\sum_{i=1}^{m}\left(\hat{p}_k^{(i)} - y_k^{(i)}\right)\mathbf{x}^{(i)} \tag{16}$$

- 모델이 타겟 클래스에 대해서 높은 확률을 추정하도록 만들기 위해 비용 함수를 최소화하는 전략으로 사용합니다.
- 크로스 엔트로피는 추정된 클래스의 확률이 타겟 클래스에 얼마나 잘 맞는지 측정하는 용도로 사용됩니다.

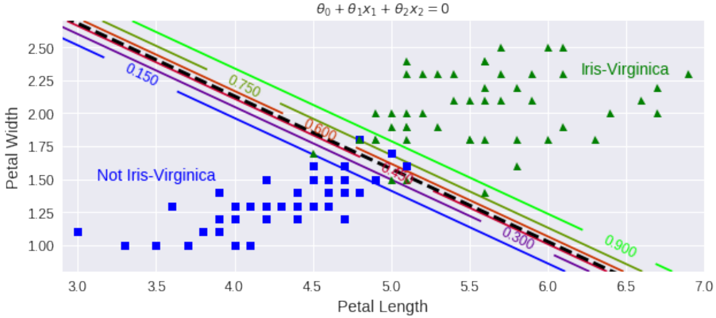

Figure 10: 선형 회귀 결정 경계

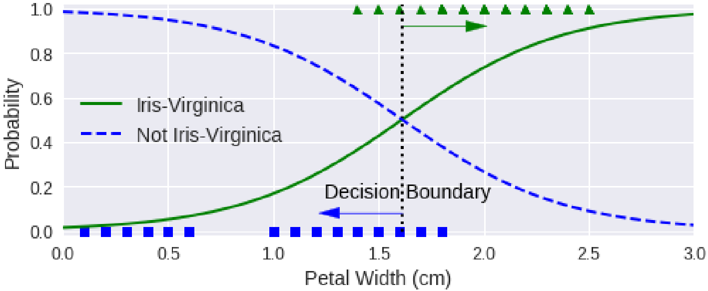

Figure 11: 로지스틱 회귀 결정 경계

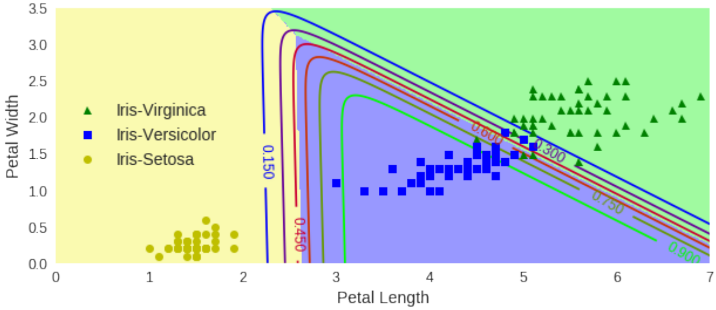

Figure 12: 소프트맥스 회귀 결정 경계

참고 사항

- 실습: https://github.com/rickiepark/handson-ml2
- 핸즈온 머신러닝 1판: https://bit.ly/2Tgfd2w