K-mooc, cau, 딥러닝의 깊이 있는 이해를 위한 머신러닝, 최종원 교수님
제출자 최규형 (choigww)

중간고사 과제

1. Linear Regression Model의 학습 방법은 왜 Linear Classification Model에 부적합한가

1.1 Linear Regression Model이 해결하려는 문제

Linear Regressor는 0, 1 등 discrete하게 구분된 소수의 라벨을 구하는 것이 아닌, 0에서 100 사이의 무수히 많은 연속적인 실수 라벨을 갖을 때 적용하는 방법이다. 예를 들어, 어떤 사람의 교육 기간을 특징 x값으로, 연수입을 라벨 y값으로 가질 때의 상관관계를 표현하는 문제에 적합하다.

1.2 Linear Regression Model의 학습 방법

Linear Regression Model은 연속적인 실수 라벨에 대하여 최대한 정확한 예측치를 출력하는 것을 목표로 한다. 이러한 문제를 해결하기 위해 Linear Regression Model은 학습의 척도로서 실제 라벨 값과 예측치 간의 차이를 최소화하는 Least Square를 가장 기본적인 형태로 채택한다.

$$f(w) = \sum_{i=1}^{n} (wx_i - y_i)^2$$

따라서 Linear Regression Model은 Least Square 수식에 대한 최소값을 구하는 것을 목적으로 한다. 위 수식에 대한 최소값을 알아내기 위해서, w(가중치/파라미터)에 대하여 미분을 1회 적용한 값이 0이 되는 지점을 파악해야 한다.

w는 특징의 차원 수(d)만큼 존재하므로, 각 값에 대하여 총 d번의 미분을 진행한다. 이는 우리가 알고 싶은 Least Square 값이 단순히 특정한 특징 차원에 대해서가 아닌, 모든 특징들이 이루는 차원 공간에서 실측치와 예측치의 차이를 최소화하는 w의 집합을 구하고자 하기 때문이다.

$$f(w_1, w_2, \dots, w_d) = \frac{1}{2}\left(\sum_{j=1}^{d} w_j x_{ij}\right)^2 - \left(\sum_{j=1}^{d} w_j x_{ij}\right) y_i + \frac{1}{2} y_i^2$$

$$\frac{\partial}{\partial w_k} f(w_1, w_2, \dots, w_d) = \left(\sum_{j=1}^{d} w_j x_{ij}\right) x_{ik} - y_i x_{ik} = \left(\mathbf{w}^{\mathrm{T}} \mathbf{x}_i - y_i\right) x_{ik}$$

따라서 아래 식을 성립시키는 w가 우리가 구하고자 하는 w가 되며, 이를 Linear Regression 모델의 파라미터라고 한다.

$$\nabla f(\mathbf{w}) = \mathbf{A}\mathbf{w} - \mathbf{b} = \mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w} - \mathbf{X}^{\mathrm{T}}\mathbf{y} = \mathbf{0}$$

Linear Regression Model은 연속적인 실수값을 예측하는 문제 뿐 아니라, classification 문제에 대해서도 사용 가능하다. 연속적인 실수 값을 범위로 무리지어 불연속적인 라벨을 할당하는 방법이 있는데, 예를 들면 0부터 1 사이의 범위를 10개 구간으로 등분한 10개의 라벨을 지정하는 것이다.

그러나 위 예시의 경우 0.01과 0.09이라는 특징값의 정보량 차이는 무시되어 동일 라벨로 취급된다. 이를 해결하기 위해서는 범위 구간을 촘촘하게 나누어야 한다. 그러나 모델 특성상 실수 라벨 값의 범위를 제한할 수 없고, 라벨의 숫자가 지나치게 커질 경우 Linear Regression Model의 효율성이 나빠지게 된다.

1-3. Linear Classification Model이 해결하려는 문제

Linear Classifier는 Linear Regressor와 같이 w*x의 형태로 예측값을 계산하되, f(w*x)의 형태로서 함수를 추가 적용해 라벨을 구분하는 것을 목표로 한다. 예를 들면, 프리딕션의 결과(w*x)를 0.0을 기준으로 하여 0.0 초과하면 +1, 0.0 미만이면 -1을 출력한다.

위 예시를 기준으로 Least Square 척도를 Linear Classification Model에 적용한다면, 라벨은 -1과 1로 정해져 있으나 예측치의 범위는 제한이 없으므로 에러가 과도하게 측정되는 문제가 생긴다. 동일한 라벨이더라도 라벨 1을 갖는 예측치 0.1에 비하여 라벨 1을 갖는 예측치 9081은 지나치게 높은 오차를 지니게 된다. 이 경우, 모델은 예측치 9081에 대한 오차를 최소화시키는 w를 찾으려 할 것이다.

2. 분류 문제에 대한 Linear Classification Model 학습 최적화 방법

위와 같은 문제는 예측치 범위를 제한할 수 없는 Linear Model의 한계에 기인한다. 이 문제를 해결하기 위하여 등장한 것이 0-1 Loss Function이다. Least Square의 결과에 대하여 0-1 Loss Function을 적용하면, Model은 예측치와 실제값이 동일할 때 0, 동일하지 않을 때 1을 출력한다.

$$\|\hat{y} - y\|_0$$

그러나 0-1 Loss Function은 모든 구간에서 기울기가 1이며, 이는 미분했을 때 항상 0이 됨을 의미한다. 미분이 불가능할 경우 Least Square 방식(normal equation)으로 오차 최소화 문제를 풀 수 없게 된다. 이러한 미분 불가능 문제를 해결하기 위해 나온 것이 Max Function이다.

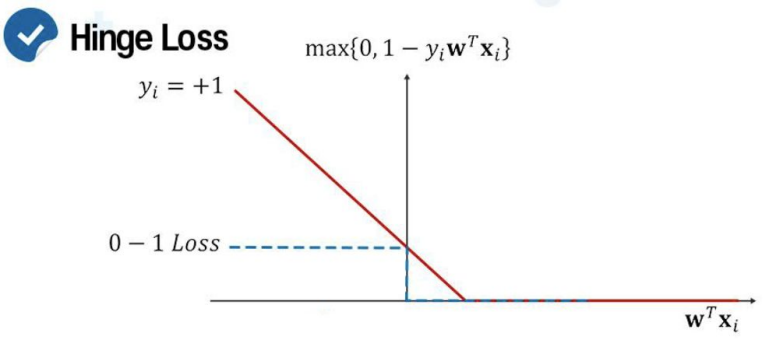
$$\max\{0, -y_i \mathbf{w}^T \mathbf{x}_i\}$$

Max Function은 예측치와 실제값이 동일하면 항상 0을 출력하고, 부호가 다르다면 항상 0보다 큰 값을 출력한다. 미분 불가능 문제를 해결하고, 틀린 것에 대해서만 w를 최소화하기 위해 고안되었다. 그러나, 모든 w가 0이 되면 Max Function의 값도 0이 되므로 최적의 w가 0이 되는 방향으로 학습을 진행하는 degenerate 문제가 발생했다.

$$y_i w^T x_i > 0 \text{ with } y_i w^T x_i > 1$$

w가 모두 0이 될 때 Loss Function이 최소값을 출력하지 않도록, 0-1 Loss Function을 기반으로 개선한 것이 Hinge Loss Function이다. Hinge Loss Function은 다음과 같은 특징을 갖는다.

1. Hinge Loss Function은 0-1 Loss Function보다 항상 크거나 같음
2. 정확하게 0-1 Loss Function과 일치하는 지점 또는 구간이 존재함
3. 그 외에는 모두 Hinge Loss가 0-1 Loss보다 큰 영역을 가지게 됨



이러한 Hinge Loss Function을 한층 더 개선한 것이 바로 Logistic Loss Function이다. Logistic Loss Function은 Max Function에 log-sum-exponential을 적용한 함수다. 2개 값에 대하여 exponential을 적용, 큰 값이 대부분의 비중을 차지하는 결과값을 출력하는 것을 기본 원리로 한다.

Logistic Loss Function은 모든 영역에 대하여 미분이 가능하며, 아래 그래프와 같이 Hinge Loss Function을 한 번 더 추정(approximation)한 형태라고 말할 수 있다. 이 외, w*x를 0과 1 사이로 강제하는 Sigmoid Function 등을 기반으로 Linear한 방식의 확률적 구분기를 만들 수도 있다.

$$f(w) = \sum_{i=1}^{n} \log(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i))$$