

[Cross Entropy]


1. 정보 이론에서의 엔트로피

정보 이론(혹은 통계학적 의미)에서의 엔트로피는 "어떤 사건이 정보적인 측면에서 얼마나 중요한지를 반영하는 로그 지표의 기댓값"이라고 정의된다.

1948년 Claude Shannon이 정보량을 계량화하는 방법으로서 제안한 것이다. Shannon은 1) 확률이 큰 사건일수록 그 사건에서 얻을 수 있는 정보량은 적어지며, 2) 독립적인 두 사건이 발생할 때 해당 사건들로부터 얻을 수 있는 정보량은 더해져야 한다고 생각했다.

이를 바탕으로 그는 한 사건에서 얻을 수 있는 정보량을 확률과 확률의 역수에 로그를 취한 값을 곱한 형태로 정의했다. 그리고 (자연스럽게) 모든 사건들로부터 얻을 수 있는 정보량은 각각의 사건에서 얻을 수 있는 정보량의 평균으로 정의했다.

교재의 엔트로피 식을 살펴 보자.

 image-20200623091033650

- discrete한 random variable X에 대해,
- 각각의 사건이 발생할 확률과,
- 그 확률의 역수 값에 로그를 취한 것을 곱한 뒤,
- 모두 더하면 **정보량**을 구할 수 있다.

정보의 가치

$$\log \frac{1}{p_i}$$

정보적인 측면에서 얼마나 중요한지를 반영한 부분이 위에 나타나 있다. 확률에 역수를 취했기 때문에, 확률 값(p_i)이 작아지면 전체 로그 값은 커지고, 확률 값이 커지면 전체 로그 값은 작아진다.

낮은 확률값을 가질수록, 즉, 드물게 발생하는 사건일수록 귀한 정보라고 간주하는 것이다. 즉, **정보 가치는 확률과 반비례**한다.

모든 정보는, 우리가 알고 있을 때 다른 사람이 몰라야 가치가 있다. 드물게 발생하는 사건에 대한 정보를 얻어야 진짜 중요한 정보를 얻는 것이다. 이 개념을 식으로 반영한 것일 뿐이다.

독립적인 사건의 정보량

본래 확률 이론에서 독립적인 두 사건 A, B가 모두 발생할 확률은, 각 사건이 발생할 확률의 곱으로 정의된다. 그러나 두 사건이 모두 발생할 때 정보량은 각각의 곱이 아니라 각각의 합이 되어야 한다.

요컨대, 동전을 던져 앞면이 2번 나오는 사건에 대한 정보량은 동전을 던져 앞면이 1번 나오는 정보량의 2배가 되어야 한다는 원리이다.

_출처: [ratsgo's blog](#)

위에서 Shannon은 한 사건의 정보량을 그 사건이 발생할 확률에 로그 확률을 곱한 것으로 정의했다. 따라서 로그 계산 법칙에 의해 로그의 곱은 합으로 나타낼 수 있게 되므로, 독립적인 두 사건이 발생했을 때의 두 정보량의 합은 로그 확률의 표현될 수 있다.

전체 정보량

이제 정보량(I)을 한 사건이 발생할 확률의 로그 확률로 정의하면 Shannon의 정보 가치 및 정보량에 대한 아이디어를 모두 표현할 수 있음을 확인했다.

이제 어떤 사건은 일어날 수도 있고, 일어나지 않을 수도 있기 때문에, 정보량의 기댓값을 구하면 전체 정보량의 기댓값을 알 수 있게 된다. 일반적인 확률 이론에서의 기댓값 공식을 따라 정보량의 기댓값을 구하자.

$$Entropy = \sum (p_i \times \log \frac{1}{p_i})$$

정보 엔트로피

Shannon이 정의한 정보량 식이 정보 엔트로피(entropy)라는 이름을 갖게된 것은, 그 식의 형태가 볼츠만 엔트로피 식과 동일한 형태이기 때문이다. (열역학의 엔트로피 식은 강의 범위를 넘기 때문에 기록하지 않는다.)

열역학에서 엔트로피 식은 유용하지 않은 에너지의 흐름을 설명할 때 사용된다. 어떤 사건이 발생할 때 유용한(혹은 중요한) 정보가 무엇인지를 나타내고자 했던 Shannon의 정보량 개념이, 정보의 흐름 측면에서 열역학의 '엔트로피'와 접목될 수 있는 부분이다. 따라서 Shannon의 정보량은 **정보 엔트로피**라 불리게 되었다.

원래 Shannon은 정보량을 비트(bit)로 표현한다고 했기 때문에, 정보량을 나타내는 로그확률에서 로그의 밑은 2가 되어야 한다. 그러나 이 부분까지는 진행하지 않는다.

참고2

엔트로피를 통해 정보 불순도를 알아 보자. Decision Tree 알고리즘에서의 엔트로피, 정보 불순도와 연관 지어 복습하자.

엔트로피 식	엔트로피와 정보 불순도
<p><u>Entropy</u> ← controls how a DT decides where to split the data</p> <p>definition: measure of <u>impurity</u> in a bunch of examples</p> <p>entropy = $\sum_i -p_i \log_2(p_i)$</p> <p>sum over all classes → fraction of examples in class i</p>	<p><u>Entropy</u> ← controls how a DT decides where to split the data</p> <p>definition: measure of <u>impurity</u> in a bunch of examples</p> <p>intuition</p> <p>all examples are <u>same class</u> → entropy = 0</p> <p>examples are <u>evenly split</u> between classes → entropy = 1.0</p>

<https://www.youtube.com/watch?v=NHAatuG0T3Q>

2. 교차 엔트로피

교차 엔트로피는 서로 다른 두 확률 분포 P, Q에 대한 정보 엔트로피를 정의한다.

교재의 식을 보자.

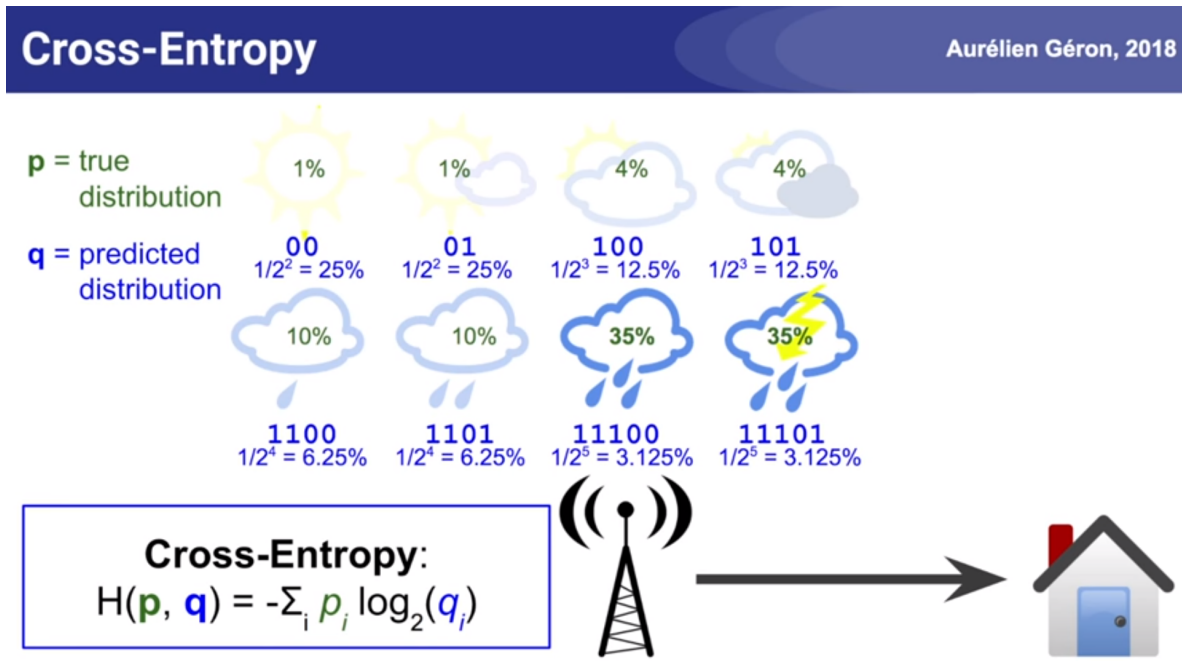
$$\text{Cross Entropy (CE)} = - \sum_i p_i \log(q_i) = H(p, q)$$

엔트로피 식에서 \log 안에 들어가는 부분만 q 로 달라졌다. 엔트로피는 엔트로피이되 두 확률분포가 교차로 곱해진다는 측면에서 '크로스' 엔트로피라는 이름을 갖게 된 듯하다.

왜 사용하는가?

굳이 다른 확률분포를 곱하는 까닭은, (실제 분포를 모르는 상황에서) 실제 분포를 가정한 후 예측한 분포에 따른 정보 획득의 유용성을 나타내기 위함이다.

아래 그림에서처럼, P는 각 사건의 정확한 분포를, Q는 P를 예측하는 분포라고 생각해 보자.



출처: <https://www.youtube.com/watch?v=ErfnhcEV1O8&t=391s>

그러면 크로스 엔트로피 식에 의해 계산된 값은 실제 사건이 일어날 확률 p 를 q 로 예측했을 때의 획득된 정보량을 나타낸다고 볼 수 있다.

어쨌든, 크로스 엔트로피 식을 사용하면 실제 분포를 가정하는 예측 분포가 있고, 그 예측한 분포에 따라 획득한 정보의 유용성이 얼마나 되는지 계산할 수 있다. 이 식 자체의 수학적 작동 원리에 대해 너무 깊이 파고 들지는 말자. (제발!)

손실 함수로서의 Cross Entropy

머신러닝 혹은 딥러닝에서 Cross Entropy가 왜 손실함수로 쓰이는지를 이해해야 한다.

머신러닝 모델을 통해 실제 확률 분포 P 를 예측하려고 한다. 머신러닝 모델이 P 를 예측하기 위해 만들어진 분포는 Q 이다. 그러면, 실제 확률 분포 상에서 어떤 사건이 발생했을 때 정보량은 $1/p$ 가 되지만, 모델링을 통해 획득한 정보량은 $1/q$ 가 된다. 따라서 크로스엔트로피 식에 의해 p 와 $1/q$ 를 곱하면, 모델링을 통해 실제 사건을 예측했을 때의 교차 정보량을 구할 수 있게 된다!

머신러닝 모델링에는 학습에 사용할 실제 관측값(p)이 있고, 모델링을 통해 예측한 값(q)이 있다. 교재의 예시를 보자.

$$y^t = [1, 0, 0] \quad \leftarrow \text{True value (p)}$$

$$y^p = [0.7, 0.1, 0.2] \quad \leftarrow \text{Prediction value (q)}$$

3 label 분류 문제를 풀어야 하고, 실제 라벨 값은 1이다. 그렇다면 라벨 1이라는 사건이 발생할 실제 확률(p_1)은 1, 라벨 2라는 사건이 발생할 실제 확률(p_2)은 0, 라벨 3이라는 사건이 발생할 확률(p_3)은 0 이 된다.

머신러닝 모델링을 진행했다. 모델은 학습 데이터를 바탕으로 열심히 학습을 진행한 후, 실제 확률에 대한 예측 분포 Q 를 만들어낼 것이다. 그리고 이 Q 라는 분포를 기반으로 모델이 예측해 내기로는, 라벨 1이라는 사건이 발생할 확률(q_1)이 0.7, 라벨 2라는 사건이 발생할 확률(q_2)이 0.1, 라벨 3이라는 사건이 발생할 확률(q_3)이 0.2이라고 한다.

이 때의 크로스 엔트로피는 다음과 같이 계산될 수 있다.

$$\log \frac{1}{0.7} \times 1 + \log \frac{1}{0.1} \times 0 + \log \frac{1}{0.2} \times 0 = 0.35667494393$$

크로스 엔트로피가 손실함수로서 사용될 수 있는 이유를 알아 보기 위해, 모델이 해당 값을 완전히 틀리게 예측했을 경우와, 완전히 맞게 예측했을 경우를 가정하고 크로스 엔트로피를 계산해 보자.

- 완전히 틀린 경우

$$\begin{aligned} y^t &= [1.0 \ 0.0 \ 0.0] \\ y^p &= [0.0 \ 1.0 \ 0.0] \\ CE &= \log \frac{1}{0.0} \times 1 + \log \frac{1}{1.0} \times 0 + \log \frac{1}{0.0} \times 0 = \inf \end{aligned}$$

모델이 실제 값과 완전히 다르게 예측한 경우는 크로스 엔트로피 값이 무한대가 나온다.

- 완전히 맞은 경우

$$\begin{aligned} y^t &= [1.0 \ 0.0 \ 0.0] \\ y^p &= [1.0 \ 0.0 \ 0.0] \\ CE &= \log \frac{1}{1.0} \times 1 + \log \frac{1}{0.0} \times 0 + \log \frac{1}{0.0} \times 0 = 0 \end{aligned}$$

모델이 실제 값과 완전히 같게 예측한 경우는 크로스 엔트로피 값이 0이 나온다. 이 경우는 크로스 엔트로피 값이 실제 엔트로피 값과 동일해 진다.

모델이 실제 값과 완전히 같게 예측한 경우, 크로스 엔트로피 값이 실제 엔트로피 값과 동일해진다는 사실에 주의하자.

결국 머신러닝 모델링은 예측한 분포를 통해 실제 분포의 엔트로피를 구하고자 하는 과정이다. 요컨대, 틀릴 수 있는 정보(모델의 예측값)를 가지고 실제 정보의 정보량을 계산하는 것이란 의미다. 직관적으로 이해했을 때, 머신러닝 모델링은 틀릴 수 있는 정보를 가지고 있으므로, 실제 분포에서 정보의 양보다 더 많은 정보를 갖게 된다. 그렇기 때문에 머신러닝 모델링을 통해 구한 교차 엔트로피는 실제 엔트로피보다 항상 크거나 같을 수밖에 없다.

그리고 머신러닝의 모델링 차원에서 이해하자면, 학습이 잘 되어 예측값과 실제 값이 비슷해질수록 크로스 엔트로피 값은 엔트로피 값과 동일해 지게 된다.

- 예측 값과 실제 값이 완전히 다른 경우 크로스 엔트로피의 값은 무한대이다.
- 예측값과 실제 값이 비슷할 경우, 크로스 엔트로피는 엔트로피 값에 근접해 가게 된다.
- 마침내 예측값과 실제 값이 완전히 일치하면, 크로스 엔트로피 값은 엔트로피 값과 같아 진다.

결국 크로스 엔트로피와 엔트로피의 차이가 머신러닝 학습의 정도를 나타낸다. 그렇기 때문에 크로스 엔트로피 값을 계산해서, gradient descent, back propagation 등 학습 과정에서의 손실 함수로 사용할 수 있게 된다.

기존의 손실함수인 MSE와 비교해서 살펴 보면, 모델의 정확도가 상승할수록 CE와 MSE가 모두 감소하는 것을 알 수 있다. 그리고 일반적으로 분류 문제에서는 MSE보다 CE를 사용하는 것이 더 좋다고 알려져 있다(나중에 더 참고 : [ratsgo's blog](https://ratsgo.github.io)).

🔥 Cross Entropy (CE)와 Mean Square Error (MSE)

- 아래 예시에서 정확도가 상승할수록 CE와 MSE가 감소하는 것을 알 수 있다. 따라서 CE나 MSE를 최소화하도록 학습하면 정확도가 높은 y 값을 추정할 수 있다.

Case A : 정확도가 높은 경우

실제 Class	예측값	예측 Class	정확도	CE	MSE
1	0.7585	1	1	0.2764	0.0583
0	0.0384	0	1	0.0391	0.0015
0	0.4525	0	1	0.6024	0.2048
0	0.0510	0	1	0.0523	0.0026
0	0.0388	0	1	0.0396	0.0015
0	0.3825	0	1	0.4821	0.1463
1	0.5514	1	1	0.5953	0.2012
0	0.2020	0	1	0.2257	0.0408
1	0.7961	1	1	0.2281	0.0416
0	0.2712	0	1	0.3164	0.0736
합계			100.00%	2.8575	0.2779

평균	10.99	0.57
표준편차	5.30	0.16

정확도	CE	MSE
10%	20.5406	0.7735
20%	11.0685	0.6361
30%	12.8731	0.6910
40%	8.4569	0.5625
50%	9.7229	0.5795
60%	15.9780	0.7478
70%	14.1874	0.5933
80%	10.2063	0.5197
90%	3.8865	0.3374
100%	3.0000	0.3000

Normalized	
CE	MSE
1.8016	1.2696
0.0144	0.3948
0.3549	0.7442
-0.4783	-0.0737
-0.2395	0.0344
0.9408	1.1058
0.6029	0.1222
-0.1482	-0.3459
-1.3407	-1.5067
-1.5079	-1.7447

Case B : 정확도가 낮은 경우

실제 Class	예측값	예측 Class	정확도	CE	MSE
1	0.1130	0	0	2.1799	0.7867
1	0.2856	0	0	1.2530	0.5103
1	0.6215	1	1	0.4756	0.1433
1	0.9345	1	1	0.0678	0.0043
1	0.1314	0	0	2.0299	0.7545
0	0.7214	1	0	1.2781	0.5205
1	0.0803	0	0	2.5225	0.8459
1	0.2698	0	0	1.3101	0.5332
0	0.5344	1	0	0.7644	0.2856
1	0.5979	1	1	0.5144	0.1617
합계			30.00%	12.3957	0.6742

