Entropy

Refer : https://hyunw.kim/blog/2017/10/27/KL_divergence.html

정보량

알파벳 한글자 추측

$$2^{\text{질문개수}} = 26$$

$$\text{질문개수} = \log_2 26 \approx 4.7$$

6글자인 경우

$$6 \times 4.7 = 28.2$$

$\therefore$ 질문개수 $= \log_2$(가능한 결과의 수)

$$\text{정보}(H) = n\log(s) = \log(s^n)$$

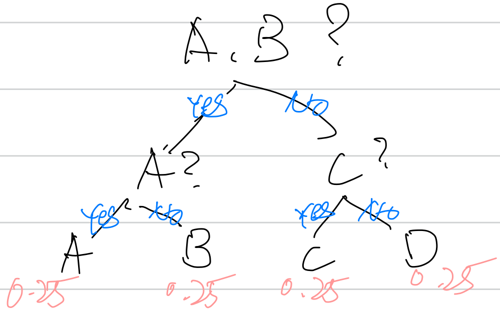
n: 문자의 개수 (6글자 → 6)
s: 선택 경우의 수 (알파벳은 26개이므로 26)

Entropy

기계 X는 A, B, C, D를 각각 0.25 확률로 출력

기계 Y는 〃 0.5, 0.125, 0.125, 0.25

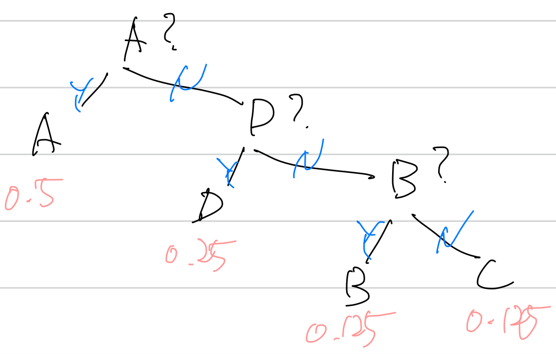
X가 출력하는 문자를 알기위한 최소 질문 => 2개.
($\log_2 4$)



But Y ?

=> 단순한 log로 계산 불가능

A에 확률 X A를 출력하기 위한 최소질문 개수.



$P(A) \cdot 1 + P(B) \cdot 3 + P(C) \cdot 3 + P(D) \cdot 2$
$= 1.75$

Y가 X보다 더 적은 정보량 생산

불확실(랜덤성)이 클수록 정보량도 크다.

Shannon은 불확실성의 측정을 Entropy 라고 부름.

이를 H 라고 표현, 단위는 bit를 사용.

결론확수는 발생 확률과 연관.

$A: 0.5 = \frac{1}{2}$ $B: 0.125 = \frac{1}{8}$ $C: 0.125 = \frac{1}{8}$ $D: 0.25 = \frac{1}{4}$

$\log_2\left(\frac{1}{\frac{1}{2}}\right) = 1$ $\log_2\left(\frac{1}{\frac{1}{8}}\right) = 3$ $\log_2\left(\frac{1}{\frac{1}{4}}\right) = 2$

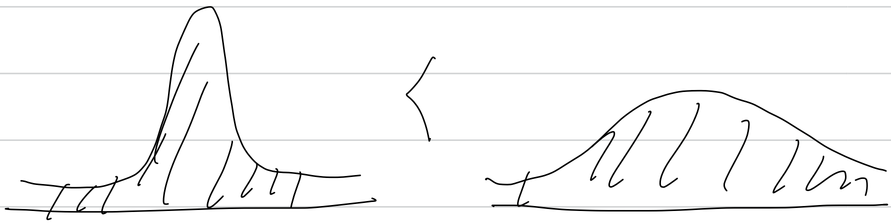
가능한 결과의 수는 그 사건 발생확률의 역수.

$\therefore H = \sum (\text{사건발생확률}) \times \log_2\left(\frac{1}{\text{사건발생확률}}\right)$

$= \sum_i P_i \cdot \log_2\left(\frac{1}{P_i}\right)$

$= -\sum_i P_i \cdot \log_2 P_i$

Entropy는 가능한 모든 사건이 같은 확률로 일어날때 최대값을 갖는다.



Entropy에서 Cross entropy 까지

$$H = \sum_i P_i \log_2 \frac{1}{P_i} = -\sum_i P_i \log_2 P_i$$

기계 X → P라고 명칭, (0.25, 0.25, 0.25, 0.25)

Y → Q (0.5, 0.125, 0.125, 0.25)

P 전략을 Q에 적용한다면?

entropy = 0.5 × 2 + (0.125 × 2) × 2 + 0.25 × 2 = 2.

entropy 가 증가.

이것이 바로 cross entropy!

Cross entropy는 어떤 문제에 대해 특정전략을 쓸때 예상되는 질문개수에 대한 기대값

확률분포 (Q) — 0.25라고 가정

즉, 해당문제에 대해 최적의 전략을 사용할때 CE의 값이 최소화 된다.

⇒ 확률분포로 된 어떤 문제 p에 대해 확률분포로 된 어떤 전략 q를 사용할때의 질문개수의 기대값이 CE

Cross entropy 의 쓰임새

$$H(p, q) = \sum_i p_i \log_2 \frac{1}{q_i} = -\sum_i p_i \log_2 q_i$$

$$-\int p(x) \log q(x)\, dx$$

p_i: 목표값 (참값) (0.5, 0.125, 0.125, 0.25)

q_i; 현재 학습한 값. (0.25, 0.25, 0.25, 0.25)

p_i에 가까워질수록 CE는 작아짐

Binary classification과 cross entropy

$$-y \log \hat{y} - (1-y)\log(1-\hat{y})$$

우리가 logistic regression에서 보는 cost function입니다. 이 식은 사실 정확하게 cross entropy 식입니다. 하나하나 뜯어보겠습니다. Binary classification에서는 0 또는 1로 두 가지 class를 구분합니다. 이를 수식으로 표현하면 $y \in \{0, 1\}$ 입니다. 우리가 어떤 대상이 1이라고 predict(또는 분류)하는 확률 $q_{y=1}$ 을 $\hat{y}$로 놓겠습니다: $q_{y=1} = \hat{y}$. 그렇다면 어떤 대상을 0으로 predict하는 확률 $q_{y=0}$ 은 $(1-\hat{y})$ 가 됩니다(1 또는 0, 두 가지 결과만 있으므로). 그리고 실제로 어떤 대상이 0 또는 1일 확률(참값) $p_{y=1}, p_{y=0}$ 은 각각 y와 $(1-y)$ 가 됩니다. 이제 좀 보이는 것 같습니다. 이 내용을 지금까지 우리가 해왔던 것처럼 정리를 해보면 결국 이런 상황인 것입니다.

$$p = [\, y, 1-y \,]$$
$$q = [\, \hat{y}, 1-\hat{y} \,]$$

따라서 logistic regression의 cost function은 단순히 우리가 위에서 봤던 cross entropy의 sigma($\sum$)를 풀어쓴 것에 불과했습니다. [Wikipedia 참고]

Cross entropy와 Log loss

[What’s an intuitive way to think of cross entropy? 발췌]

Cross entropy는 log loss로 불리기도 합니다. 왜냐하면 cross entropy를 최소화하는 것은 log likelihood를 최대화하는 것과 같기 때문입니다. 찬찬히 살펴보겠습니다. 어떤 데이터가 0 또는 1로 predict될 확률은 $\hat{y}, 1-\hat{y}$이므로 그 데이터의 likelihood 식을 이렇게 세워볼 수 있습니다. (y가 0 또는 1의 값만 가질 때 가능합니다)

$$\hat{y}^{y}(1-\hat{y})^{(1-y)}$$

$y=1$일 때 $\hat{y}$를 최대화시켜야 하고, $y=0$일 때는 $(1-\hat{y})$를 최대화시켜야 합니다. 여기에 $\log$를 씌워보겠습니다. 그러면 지수들이 내려와서 이렇게 됩니다: $maximize \;\; y\log\hat{y} + (1-y)\log(1-\hat{y})$

$$= minimize \;\; -y\log\hat{y} - (1-y)\log(1-\hat{y})$$

최소화 해야 하는 식은 우리가 방금 살펴본 cross entropy와 똑같은 식입니다. 이런 이유로 cross entropy를 log **loss**라고 부르기도 합니다. 왜냐하면 cross entropy 값이 커지면 log likelihood가 작아지기 때문에 cross entropy값을 작게 해야 합니다. 또 나아가 자연스럽게 cross entropy는 **negative log likelihood**로 불리기도 합니다.

Kullback-Leibler divergence

$$H(P,q) = -\sum P_i \log q_i$$

$$= -\sum P_i \log q_i \overbrace{-\sum P_i \log P_i}^{H(P)} + \sum P_i \log P_i$$

$$= H(P) + \sum P_i \log P_i - \sum P_i \log q_i$$

$$= H(P) + \sum P_i \log \frac{P_i}{q_i}$$

P의 entropy에 이만큼 더해진게 cross entropy

P와 q분포의 정보량 차이 (KL-Div)

$$KL(P\|q) = H(P,q) - H(P)$$

$$D_{KL}(P|q) \,/\, KL(P\|q) = \begin{cases} \sum P_i \log \frac{P_i}{q_i} \quad \text{or} \quad -\sum P_i \log \frac{q_i}{P_i} \\ \int P(x) \log \frac{P(x)}{q(x)} \quad \text{or} \quad -\int P(x) \log \frac{q(x)}{P(x)} dx \end{cases}$$

Cross entropy 최소화 하는 것은, 어차피 H(P)는 고정된 상수이기 때문에 KL-div 를 최소화 하는거랑 같다.

KL-divergence의 특성

① $KL(P||Q) \geq 0$
② $KL(P||Q) \neq KL(Q||P)$ ⇒ 거리개념이 아님.

① KL-div은 CE에서 entropy를 빼는것.
$H(P,Q)$ $H(P)$

$H(P,Q)$의 lower bound는 $H(P)$

Jensen's Inequality로 증명가능

② 거리 개념이 아니다. (비대칭)

$$KL(P||Q) = H(P,Q) - H(P)$$
$$\neq H(Q,P) - H(Q) = KL(Q||P)$$

$$\therefore KL(P||Q) \neq KL(Q||P)$$

하지만 Jensen-shannon divergence는 대칭이어 거리개념으로 사용가능.

Jesen-Shannon divergence.

$$JSD(P\|Q) = \frac{1}{2}KL(P\|M) + \frac{1}{2}KL(Q\|M)$$

where, $M = \frac{1}{2}(P+Q)$

KL-div 2가지를 구하고 평균을 쓰는 방식

KL-divergence와 log likelihood

우리가 전체를 알 수 없는 분포 $p(x)$ 에서 추출되는 데이터를 우리가 모델링하고 싶다고 가정해보겠습니다. 우리는 이 분포에 대해 어떤 학습 가능한 parameter θ의 parametric distribution $q(x|\theta)$ 를 이용해 근사시킨다고 가정해보겠습니다. 이 θ 를 결정하는 방법 중 하나는 바로 $p(x)$와 $q(x|\theta)$ 사이의 KL-divergence 를 최소화시키는 θ 를 찾는 것입니다. 우리가 $p(x)$ 자체를 모르기 때문에 이는 직접 해내기는 불가능합니다. 하지만 우리는 이 $p(x)$에서 추출된 몇 개의 샘플 데이터(training set)는 압니다(x_n, for $n = 1, \ldots, N$). 따라서 $p(x)$에 대한 기댓값은 그 샘플들의 평균을 통해 구할 수 있습니다.

$$KL(p\|q) \simeq \frac{1}{N}\sum_{n=1}^{N}\{-\ln q(x_n|\theta) + \ln p(x_n)\}$$

$\ln \frac{p(x)}{q(x)}$

$\ln p(x_n)$ 는 θ에 대해 독립이고, $-\ln q(x_n|\theta)$ 는 training set으로 얻은 $q(X|\theta)$ 분포 하에서 θ 에 대한 negative log likelihood 입니다. 그러므로 KL-divergence를 minimize하는 것이 likelihood를 maximize하는 것과 같다는 것을 알 수 있습니다. [Pattern Recognition and Machine Learning, C.M. Bishop 참조]

정리

Cross entropy는 negative log likelihood와 같습니다. 그래서 cross entropy를 minimize하는 것이 log likelihood를 maximize하는 것과 같습니다. 그리고 확률분포 p, q 에 대한 cross entropy는 $H(p) + KL(p|q)$ 이므로 KL-divergence를 minimize하는 것 또한 결국 log likelihood를 maximize하는 것과 같습니다.