10.3 교차엔트로피와 쿨백-라이블러 발산

이 절에서는 교차엔트로피와 쿨백-라이블러 발산을 공부한다. 교차엔트로피는 분류문제의 성능을 평가하는데 유용하다. 쿨백-라이블러 발산은 교차엔트로피를 응용한 것으로 두 확률분포의 모양이 얼마나 유사한지를 평가한다.

교차엔트로피

두 확률분포 p, q의 **교차엔트로피(cross entropy)** $H[p, q]$는

이산확률분포의 경우에는 다음처럼 정의한다.

$$H[p, q] = -\sum_{k=1}^{K} p(y_k) \log_2 q(y_k) \tag{10.3.1}$$

연속확률분포의 경우에는 다음처럼 정의한다.

$$H[p, q] = -\int_{y} p(y) \log_2 q(y)\, dy \tag{10.3.2}$$

교차엔트로피는 지금까지 공부한 엔트로피, 결합엔트로피, 조건부엔트로피와 다르게 확률변수가 아닌 확률분포를 인수로 받는다는 점에 주의하라.

예제

다음과 같은 두 분포가 있을 때 교차엔트로피는 0.25다.

$$p = [1/4, 1/4, 1/4, 1/4] \tag{10.3.3}$$

$$q = [1/2, 1/4, 1/8, 1/8] \tag{10.3.4}$$

In [1]:

```
- 1/4 * np.log2(1/2) - 1/4 * np.log2(1/4) - 1/4 * np.log2(1/8) - 1/4 * np.log2(1/8)
```

Out[1]:

2.25

교차엔트로피를 사용한 분류성능 측정

교차엔트로피는 분류모형의 성능을 측정하는데 사용된다. Y가 0 또는 1이라는 값만 가지는 이진분류문제를 예로 들어보자.

p는 X값이 정해졌을 때 정답인 Y의 확률분포다. 이진분류문제에서 Y는 0또는 1이다. 따라서 p는

- 정답이 $Y = 1$일 때,

$$p(Y = 0) = 0, p(Y = 1) = 1 \tag{10.3.5}$$

- 정답이 $Y=0$일 때

$$p(Y=0)=1, p(Y=1)=0 \tag{10.3.6}$$

이다.

분포 q는 X값이 정해졌을 때 예측값의 확률분포다. 모수가 μ인 베르누이분포라고 가정한다.

$$q(Y=0)=1-\mu, q(Y=1)=\mu \tag{10.3.7}$$

따라서 확률분포 p와 q의 교차엔트로피는

- 정답이 $Y=1$일 때,

$$H[p,q]=-\cancel{p(Y=0)}\log_2 q(Y=0)-p(Y=1)\log_2 q(Y=1)=-\log_2 \mu \tag{10.3.8}$$

- 정답이 $Y=0$일 때,

$$H[p,q]=-p(Y=0)\log_2 q(Y=0)-\cancel{p(Y=1)}\log_2 q(Y=1)=-\log_2(1-\mu) \tag{10.3.}$$

가 된다.

이 값은 분류성능이 좋을수록 작아지고 분류성능이 나쁠수록 커진다. 이유는 다음과 같다.

- $Y=1$일 때는 μ가 작아질수록 즉, 예측이 틀릴수록 $-\log_2 \mu$의 값도 커진다.
- $Y=0$일 때는 μ가 커질수록, 즉 예측이 틀릴수록 $-\log_2(1-\mu)$의 값도 커진다.

따라서 교차엔트로피값은 예측의 틀린정도를 나타내는 오차함수의 역할을 할 수 있다.

N개의 학습 데이터 전체에 대해 교차엔트로피 평균을 구하면 다음 식으로 표현할 수 있다. 이 값을 **로그손실(log-loss)**이라고도 한다.

$$\text{log loss}=-\frac{1}{N}\sum_{i=1}^{N}\left(y_i\log_2 \mu_i+(1-y_i)\log_2(1-\mu_i)\right) \tag{10.3.10}$$

같은 방법으로 이진분류가 아닌 다중분류에서도 교차엔트로피를 오차 함수로 사용할 수 있다. 다중분류문제의 교차엔트로피 손실함수를 **카테고리 로그손실(categorical log-loss)**이라고 한다.

$$\text{categorical log loss}=-\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\left(\mathbb{I}(y_i=k)\log_2 p(y_i=k)\right) \tag{10.3.11}$$

위 식에서 $\mathbb{I}(y_i=k)$는 y_i가 k인 경우에만 1이고 그렇지 않으면 0이 되는 지시함수(indicator function)다.
$p(y_i=k)$는 분류모형이 계산한 $y_i=k$일 확률이다.

사이킷런(Scikit-Learn) 패키지의 metrics 서브패키지는 로그손실을 계산하는 `log_loss` 함수를 제공한다.

붓꽃 분류문제에서 꽃받침 길이(sepal length) 5.6cm를 기준으로 세토사와 베르시칼라 종을 분류한 결과는 다음과 같다.

In [2]:

```
from sklearn.datasets import load_iris

iris = load_iris()
idx = np.in1d(iris.target, [0, 1])
X = iris.data[idx, :]
y = iris.target[idx]
df = pd.DataFrame(X, columns=iris.feature_names)
df["y"] = iris.target[idx]
df["y_hat"] = (df["sepal length (cm)"] > 5.4).astype(int)
df.tail()
```

Out[2]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	y	y_hat
95	5.7	3.0	4.2	1.2	1	1
96	5.7	2.9	4.2	1.3	1	1
97	6.2	2.9	4.3	1.3	1	1
98	5.1	2.5	3.0	1.1	1	0
99	5.7	2.8	4.1	1.3	1	1

로그손실값을 계산하면 3.8이다.

In [3]:

```
from sklearn.metrics import log_loss

log_loss(df["y"], df["y_hat"])
```

Out[3]:

3.799305383311686

연습 문제 10.3.1

(1) 붓꽃 데이터에서 꽃받침의 길이(sepal length)의 최솟값과 최댓값 구간을 0.5 간격으로 나누어 각각의 값을 기준값으로 하였을 때 로그손실이 어떻게 변하는지 그래프로 그려라. 종으로는 세토사와 베르시칼라만 사용한다.

(2) 꽃받침의 길이를 특징으로 사용하였을 때 어떤 값을 기준값으로 하는 것이 가장 좋은가?

(3) 꽃받침의 폭(sepal width)에 대해 위의 분석을 실시하라. 이 때는 기준값이 어떻게 되는가?

(4) 꽃받침의 길이(sepal length)와 꽃받침의 폭(sepal width) 중 하나를 특징으로 선택해야 한다면 어떤 것을 선택해야 하는가?

쿨백-라이블러 발산

쿨백-라이블러 발산(Kullback-Leibler divergence)은 두 확률분포 $p(y)$, $q(y)$의 분포모양이 얼마나 다른지를 숫자로 계산한 값이다. $KL(p||q)$로 표기한다.

이산확률분포에 대해서는 다음처럼 정의한다.

$$\begin{aligned} KL(p||q) &= H[p,q] - H[p] \\ &= \sum_{i=1}^{K} p(y_i) \log_2 \left(\frac{p(y_i)}{q(y_i)} \right) \end{aligned} \tag{10.3.12}$$

연속확률분포에 대해서는 다음처럼 정의한다.

$$\begin{aligned} KL(p||q) &= H[p,q] - H[p] \\ &= \int p(y) \log_2 \left(\frac{p(y)}{q(y)} \right) dy \end{aligned} \tag{10.3.13}$$

쿨백-라이블러 발산은 교차엔트로피에서 기준이 되는 p분포의 엔트로피 값을 뺀 값이므로 상대엔트로피(relative entropy)라고도 한다. 그 값은 항상 양수이며 두 확률분포 $p(x)$, $q(x)$가 완전히 같을 경우에만 0이 된다.

$$\begin{aligned} KL(p||p) &= H[p,p] - H[p] \\ &= \int p(y) \log_2 \left(\frac{p(y)}{p(y)} \right) dy \\ &= 0 \end{aligned} \tag{10.3.14}$$

역으로 쿨백-라이블러 발산이 0이면 두 확률분포는 같은 확률분포다. 증명은 생략한다.

$$KL(p||q) = 0 \;\leftrightarrow\; p = q \tag{10.3.15}$$

쿨백-라이블러 발산은 거리(distance)가 아니라 확률분포 q가 기준확률분포 p와 얼마나 다른지를 나타내는 값이므로 두 확률분포의 위치가 달라지면 일반적으로 값이 달라진다는 점에 주의해야 한다.

$$KL(p||q) \neq KL(q||p) \tag{10.3.16}$$

가변길이 인코딩과 쿨백-라이블러 발산

4개의 글자 A, B, C, D로 씌여진 다음과 같은 문서를 가변길이 인코딩하는 경우를 생각하자.

In [4]:

```
N = 200
p = [1/2, 1/4, 1/8, 1/8]
doc0 = list("".join([int(N * p[i]) * c for i, c in enumerate("ABCD")]))
np.random.shuffle(doc0)
doc = "".join(doc0)
doc
```

Out[4]:

```
'DBCADDAAAAABAACABABCAAACAAABBACACCDAABDACAAAACABABBABAADDAADBAAABDBABDBABCCBBDABACA
CABADCBDBBABDAAAABABBCABDAAADADAAABAABAAABCABDCBDAACDAAAAABBCAADBBBADCAAAAAAABABAAAA
ABAAAAABCDDDCBABBABBCABCAAAACABAA'
```

이 문서를 구성하는 글자의 확률분포는 다음과 같다.

$$p(Y=A) = \frac{1}{2},\; p(Y=B) = \frac{1}{4},\; p(Y=C) = \frac{1}{8},\; p(Y=D) = \frac{1}{8} \tag{10.3.17}$$

In [5]:

```
from collections import Counter

p = np.array(list(Counter(doc).values())) / len(doc)
p
```

Out[5]:

```
array([0.125, 0.25 , 0.125, 0.5  ])
```

한글자당 인코딩된 글자수는 분포 q의 엔트로피인 1.75가 된다.

$$\begin{aligned}\sum_{k=1}^{K} p(y_i)\log_2 p(y_i) &= -\frac{1}{2}\cdot\log_2\frac{1}{2} + -\frac{1}{4}\cdot\log_2\frac{1}{4} + -\frac{1}{8}\cdot\log_2\frac{1}{8} + -\frac{1}{8}\cdot\log_2\frac{1}{8} \\ &= \frac{1}{2}\cdot 1 + \frac{1}{4}\cdot 2 + \frac{1}{8}\cdot 3 + \frac{1}{8}\cdot 3 \\ &= 1.75 \end{aligned} \tag{10.3.18}$$

In [6]:

```
sp.stats.entropy([1/2, 1/4, 1/8, 1/8], base=2)
```

Out[6]:

```
1.75
```

실제로 인코딩된 글자수에서 확인할 수 있다.

In [7]:

```
vl_encoder = {"A": "0", "B": "10", "C": "110", "D": "111"}
vl_encoded_doc = "".join([vl_encoder[c] for c in doc])
len(vl_encoded_doc) / len(doc)
```

Out[7]:

```
1.75
```

그런데 가변길이 인코딩이 아니라 고정길이 인코딩을 사용한다는 것은 다음과 같은 분포를 가정한 것과 같다.

$$q(Y=A)=\frac{1}{4},\; q(Y=B)=\frac{1}{4},\; q(Y=C)=\frac{1}{4},\; q(Y=D)=\frac{1}{4} \tag{10.3.19}$$

실제로 한글자당 인코딩된 글자수는 다음과 같이 계산할 수 있다.

$$\begin{aligned}\sum_{k=1}^{K} p(y_i)\log_2 q(y_i) &= -\frac{1}{2}\cdot\log_2\frac{1}{4} + -\frac{1}{4}\cdot\log_2\frac{1}{4} + -\frac{1}{8}\cdot\log_2\frac{1}{4} + -\frac{1}{8}\cdot\log_2\frac{1}{4} \\ &= \frac{1}{2}\cdot 2 + \frac{1}{4}\cdot 2 + \frac{1}{8}\cdot 2 + \frac{1}{8}\cdot 2 \\ &= 2 \end{aligned} \tag{10.3.20}$$

In [8]:

```
encoder = {"A": "00", "B": "01", "C": "10", "D": "11"}
encoded_doc = "".join([encoder[c] for c in doc])
len(encoded_doc) / len(doc)
```

Out[8]:

2.0

쿨백-라이블러 발산은 잘못된 분포 q로 인코딩되었을 때의 한글자당 인코딩된 글자수와 원래의 분포 p를 사용하였을 때 한글자당 인코딩된 글자수의 차이인 0.25와 같다.

$$
\begin{aligned}
KL(p||q) &= \sum_{i=1}^{K} p(y_i) \log_2 \left(\frac{p(y_i)}{q(y_i)} \right) \\
&= -\sum_{i=1}^{K} p(y_i) \log_2 q(y_i) - (-\sum_{i=1}^{K} p(y_i) \log_2 p(y_i))) \\
&= H[p,q] - H[p] \\
&= 2.0 - 1.75 = 0.25
\end{aligned}
\tag{10.3.21}
$$

즉, 확률분포 q의 모양이 확률분포 p와 다른 정도를 정량화한 것이다.

사이파이의 stats 서브패키지에서 제공하는 `entropy` 함수는 두 개의 확률분포를 인수로 넣으면 쿨백-라이블러 발산을 계산해준다. `base` 인수를 2로 설정하는 것을 잊지 말아야 한다.

In [9]:

```
sp.stats.entropy([1/2, 1/4, 1/8, 1/8], [1/4, 1/4, 1/4, 1/4], base=2)
```

Out[9]:

0.24999999999999997

연습 문제 10.3.2

A, B, C, D, E, F, G, H의 8글자로 이루어진 문서가 있고 각각의 글자가 나올 확률이 다음과 같다고 가정하자.

$$
\left\{ \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64} \right\} \tag{10.3.22}
$$

(1) 위 확률분포와 균일확률분포의 쿨백-라이블러 발산값을 구하라.

(2) 이 문서를 가변길이 인코딩을 할 때와 고정길이 인코딩을 할 때 한글자당 인코딩된 글자수를 비교하라.