

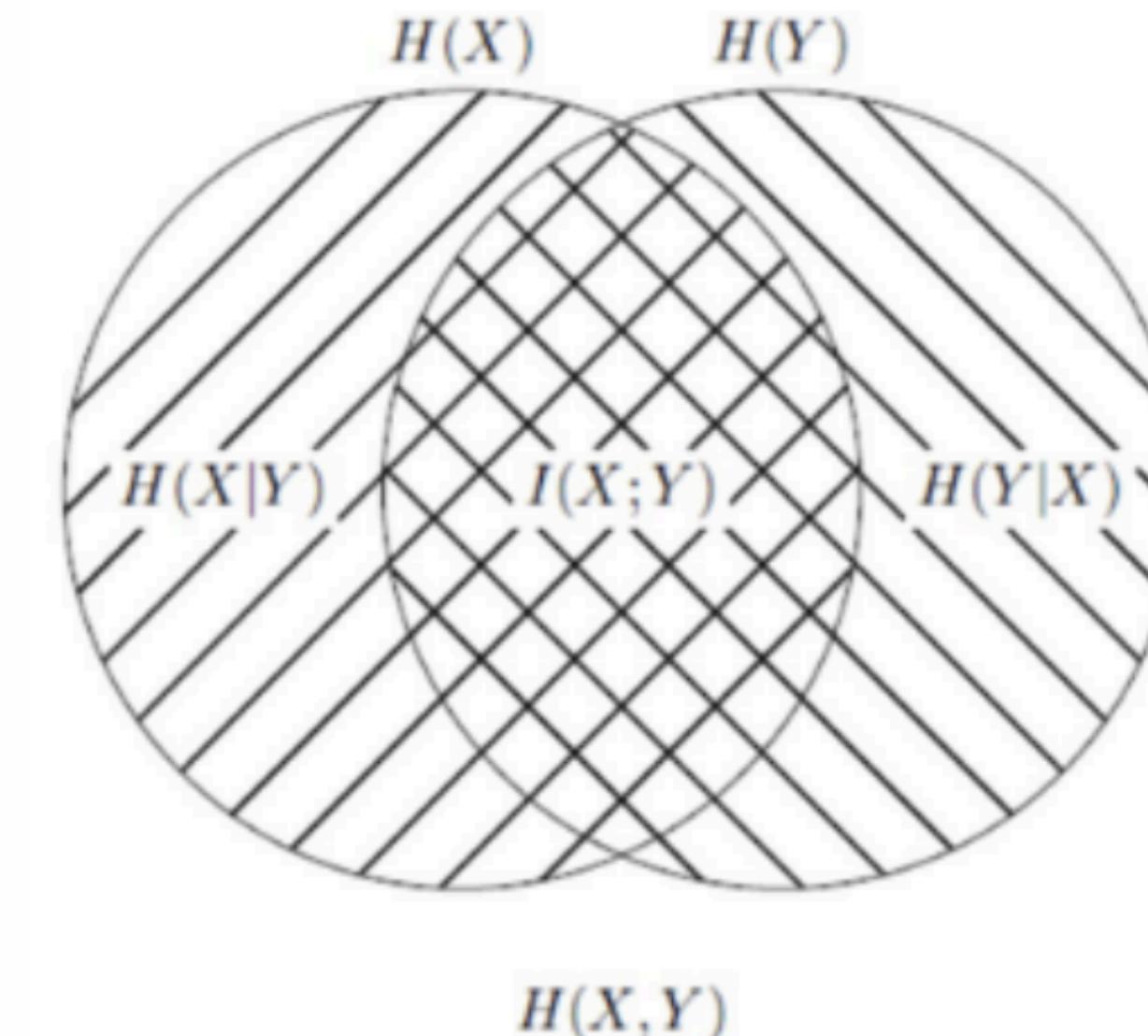
# Mutual Information

## 상호정보량

두 확률 변수 간 의존성을 나타내는 척도

- 두 변수 간 의존성이 클 때 해당 변수를 선택

$$\begin{aligned} I(X;Y) &= H(X) + H(Y) - H(X, Y) \\ &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \end{aligned}$$



# Mutual Information

---

## Entropy(엔트로피)

한 변수의 불확실성  
(- 놀라움의 정도)

$$H(X) = - \sum_x P(x) \log P(x)$$

## Joint Entropy

두 변수를 동시에 아는데  
필요한 정보량

$$H(X, Y) = - \sum_{x,y} P(x, y) \log P(x, y)$$

## Conditional Entropy

Y를 알고 있을 때 X의  
남은 불확실성

$$H(X|Y) = - \sum_{x,y} P(x, y) \log P(x|y)$$

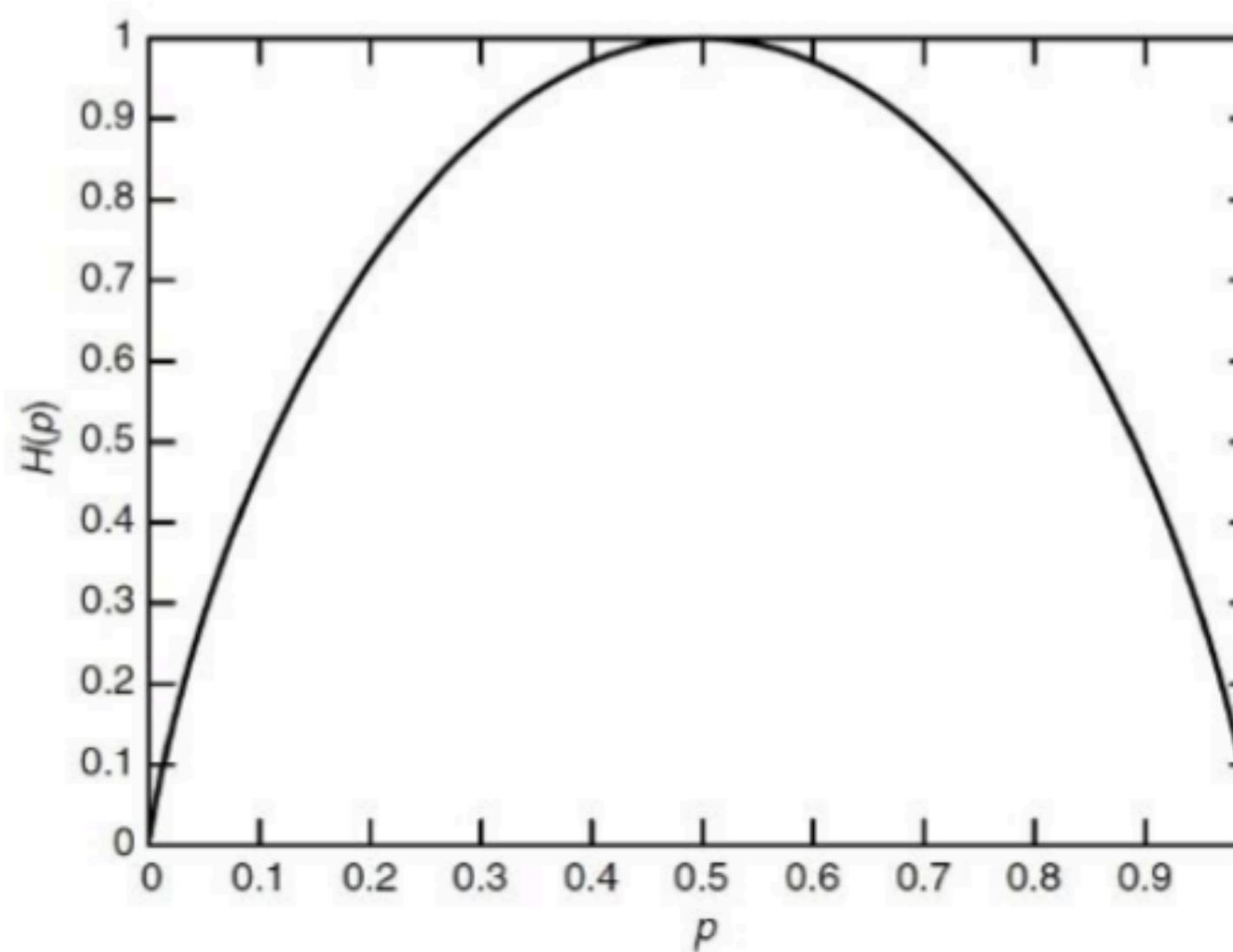
# Mutual Information

---

Ex) 해가 동쪽에서 뜬다.  $\rightarrow p(\text{동쪽}) = 1.0$

해가 서쪽에서 뜬다.  $\rightarrow p(\text{서쪽}) = 0.0$

확실한 사건!!!! = 정보량이 없음 = 불확실하지 않음



## 상호정보량을 이용한 변수 선택 과정

1. 각 feature와 target 간의 상호정보량 계산
2. 계산된 상호정보량을 기준으로 정렬
3. 임계값을 기준으로 상위 순위의 특성을 선택

# Mutual Information

---

## 장점

- 다른 feature selection보다 정확함
- 비선형적 관계도 분석 가능
- 희소한 데이터도 적용 가능

## 단점

- 계산 비용이 매우 큼

# Mutual Imformation

---

월(Month)	결제 패턴(Pattern)
1	A
2	B
1	C
3	A
2	B
3	C

# Mutual Information

---

	1	2	3	주변빈도
A	1	0	1	2
B	0	2	0	2
C	1	0	1	2
주변빈도	2	2	2	6

	1	2	3	
A	$P(1,A)$	$P(2,A)$	$P(3,A)$	$P(A)$
B	$P(1,B)$	$P(2,B)$	$P(3,B)$	$P(B)$
C	$P(1,C)$	$P(2,C)$	$P(3,C)$	$P(C)$
	$P(1)$	$P(2)$	$P(3)$	

	1	2	3	
A	$1/6$	0	$1/6$	$1/3$
B	0	$1/3$	0	$1/3$
C	$1/6$	0	$1/6$	$1/3$
	$1/3$	$1/3$	$1/3$	

# Mutual Information

---

	1	2	3	
A	1/6	0	1/6	1/3
B	0	1/3	0	1/3
C	1/6	0	1/6	1/3
	1/3	1/3	1/3	

$$\begin{aligned} I(\text{Month, Pattern}) &= P(1, A) \log_2 \left( \frac{P(1, A)}{(P(1) * P(A))} \right) + P(1, C) \log_2 \left( \frac{P(1, C)}{(P(1) * P(C))} \right) + P(2, B) \log_2 \left( \frac{P(2, B)}{(P(2) * P(B))} \right) \\ &\quad + P(3, A) \log_2 \left( \frac{P(3, A)}{(P(3) * P(A))} \right) + P(3, C) \log_2 \left( \frac{P(3, C)}{(P(3) * P(C))} \right) \\ &= 0.097 \end{aligned}$$

# Mutual Information

---

## 유방암 데이터

```
1 from sklearn.datasets import load_breast_cancer  
2  
3 data = load_breast_cancer()  
4 X = data.data  
5 y = data.target  
0.4s
```

## 유방암 데이터

```
1 from sklearn.feature_selection import mutual_info_classif, SelectKBest
2
3 # k개의 feature 선택
4 k = 10
5
6 # 상호정보량 계산
7 mi = mutual_info_classif(X, y)
8
9 # k개의 feature 선택
10 selector = SelectKBest(mutual_info_classif, k=k)
11 X_new = selector.fit_transform(X, y)
12
13 # 상호정보량이 가장 높은 feature 10개 출력
14 topk_indices = mi.argsort()[:-1][-10:]
15 topk_features = data.feature_names[topk_indices]
16 print("Top {} Features:".format(k))
17
18 for feature in topk_features:
19     print("- {}".format(feature))
```