# Decision Rule

made by Jay Hong

Check

# Decision Rule

1. Intro
2. OneR
3. Sequential Covering
4. Bayesian Rule List
5. 장단점
6. 실습

## 1) 개념

- ✓ 의사결정 규칙이란?
  - ✓ IF-THEN 문
  - ✓ 조건-예상
  - ✓ IF (오늘 비가온다) AND (오늘이 4월이다) THEN 내일 비가 온다.

## 1) 의사결정 규칙이란?

- ✓ IF-THEN 문
- ✓ 조건 - 예상

- ✓ 예시
  - ✓ IF (오늘 비가온다) AND (오늘이 4월이다) THEN (내일 비가 온다.)
  - ✓ IF (100평 이상이다) AND (정원이 있다) THEN (집값이 비싸다.)

- ✓ 유용한 결정규칙은 “Support(지지도)”와 “Accuracy(정확도)”로 요약가능

## 2) 지지도란?

- ✓ 규칙이 적용되는 인스턴스의 비율

- ✓ 예시
  - ✓ IF (size=BIG) AND (location=GOOD) THEN (value=HIGH)
  - ✓ 1,000가구의 집 중 100채가 IF에 해당하며, 84가구가 정말로 (value=HIGH)이다.

    => 지지도 = 100/1,000 = 10%
- ✓ 특징
  - ✓ 실제로 (value=HIGH)인지는 관심이 없음

## 3) 정확도란?

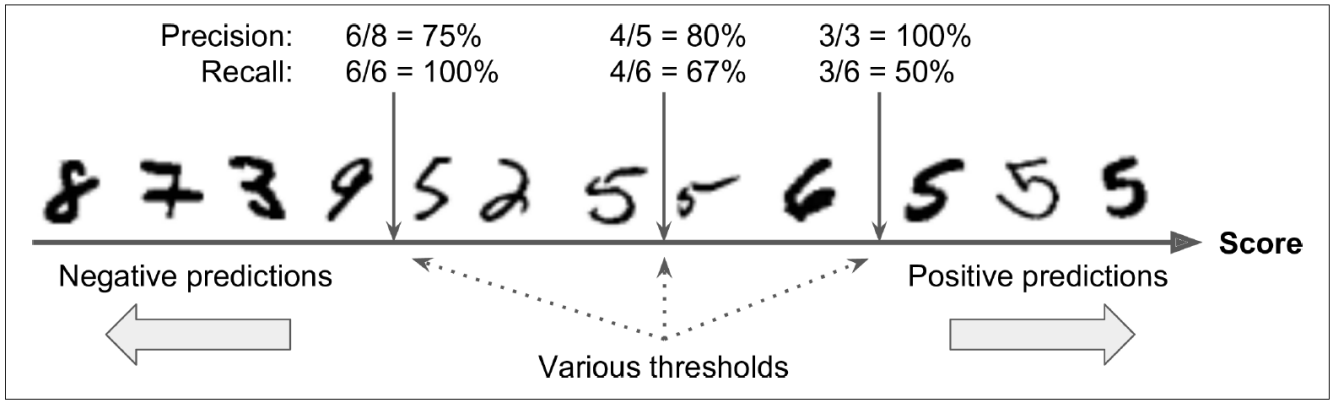
- ✓ 규칙이 얼마나 정확한가

- ✓ 예시
  - ✓ IF (size=BIG) AND (location=GOOD) THEN (value=HIGH)
  - ✓ 1,000가구의 집 중 100채가 IF에 해당하며, 84가구가 정말로 (value=HIGH)이다.

    => 정확도 = 84/100 = 84%

# 4) 지지도 & 정확도

- ✓ Trade-Off 관계
  - ✓ 한 쪽이 올라가면 다른 한 쪽이 떨어지는 관계
  - ✓ 지지도를 높이기 위해 조건을 간단히 하면 정확도가 떨어지고
    정확도를 높이기 위해 조건을 복잡히 하면 지지도가 떨어지는 관계




=> “절충안”이 필요

- ✓ 높은 정확도를 유지하면서 지지도도 가지고 있어야 함.

## 4) 지지도 & 정확도

- ✓ 절충안을 만드는 중 발생할 수 있는 문제
  - ✓ 규칙이 겹치는 경우
    - ✓ 집 값을 예측하기 위해 2개 이상의 규칙이 적용되었을 때, 모순된 예측을 한다면?
      => Decision List, Decision Set
  - ✓ 적용되는 규칙이 없는 경우
    - ✓ 어떠한 규칙도 없을 때는 어떤 예측 결과를 내야하지?
      => Default Rule

## 5) 의사결정 규칙의 과거 문제

- ✓ 규칙이 겹칠 수 있다.
  - ✓ 2개 이상의 규칙이 적용되었을 때, 서로 모순된 예측
  - ✓ 해결 : Decision List, Decision Set

- ✓ 적용되는 규칙이 없을 수 있다.
  - ✓ 개념 : 어떠한 규칙도 없을 때의 예측 결과
  - ✓ 해결 : Default Rule

## 5) 의사결정 규칙의 과거 문제

- ✓ 규칙이 겹칠 수 있다.
  - ✓ 2개 이상의 규칙이 적용되었을 때, 서로 모순된 예측
  - ✓ 해결 : Decision List, Decision Set

- ✓ 적용되는 규칙이 없을 수 있다.
  - ✓ 개념 : 어떠한 규칙도 없을 때의 예측 결과
  - ✓ 해결 : Default Rule

## 5) 의사결정 규칙의 과거 문제

- ✓ Decision List

  - ✓ 개념 :
    의사결정 규칙에 순서를 부여

  - ✓ 예시 :
    1번 규칙 True => 1번 규칙 사용
    1번 규칙 False => 2번 규칙 True => 2번 규칙 사용
    1번 규칙 False => 2번 규칙 False => 3번 규칙 True => 3번 규칙 사용
    …

  - ✓ 특징 :
    과거 문제 중 “중복 규칙 문제“ 해결

## 5) 의사결정 규칙의 과거 문제

- ✓ Decision Set
  - ✓ 개념 :
    의사결정 규칙에 가중치 부여
  - ✓ 특징 :
    해석할 때 잠재적인 어려움이 있음
    과거 문제 중 “중복 규칙 문제“ 해결

## 5) 의사결정 규칙의 과거 문제

- ✓ Default Rule
  - ✓ 개념 :
    아무런 규칙이 없을 때 적용 가능한 규칙
    가장 빈번한 클래스를 사용

## 6) 그 외의 3가지 대표 의사결정 규칙

- ✓ OneR
  - ✓ 개념 :
    단일 특성 값에서 규칙을 학습
    단순성, 해석력이 좋아 벤치마크로 사용

- ✓ Sequential Covering(순차적 적용)
  - ✓ 개념 :
    새로운 규칙에서 배우는 데이터 포인트 제거하며
    규칙을 반복적으로 학습

- ✓ Bayesian Rule Lists (베이즈 규칙 목록)
  - ✓ 개념 :
    미리 알아낸 자주 나오는 패턴을 Bayeisan 통계를 사용하여 의사결정 목록으로 결합

## 1) 개념 및 방법

- ✓ 개념
  - ✓ Holte(1993) 제언
  - ✓ 가장 단순한 규칙 유도 알고리즘
  - ✓ OneR = One Rule 이나, 사실 2개 이상의 규칙을 사용

- ✓ 방법
  - ✓ 적절한 간격을 선택해 연속된 Feature 값을 별개의 것으로 이산화
  - ✓ 각 Feature에 대해
    - ✓ Feature와 결과 사이의 Cross-Table을 만듦
    - ✓ Feature 값에 대해 특정 Feature가 있는 인스턴스의 가장 빈번한 Class를 예측하는 규칙을 만듦
    - ✓ Feature에 대한 오류 계산
  - ✓ 오류가 가장 낮은 특성 선택

# 2) 예시

| location | size | pets | value |
|---|---|---|---|
| good | small | yes | high |
| good | big | no | high |
| good | big | no | high |
| bad | medium | no | medium |
| good | medium | only cats | medium |
| good | small | only cats | medium |
| bad | medium | yes | medium |
| bad | small | yes | low |
| bad | medium | yes | low |
| bad | small | no | low |

✓ 적절한 간격을 선택해 연속된 Feature 값을 별개의 것으로 이산화

<location>

| | value=low | value=medium | value=high |
|---|---|---|---|
| location=bad | 3 | 2 | 0 |
| location=good | 0 | 2 | 3 |

<size>

| | value=low | value=medium | value=high |
|---|---|---|---|
| size=big | 0 | 0 | 2 |
| size=medium | 1 | 3 | 0 |
| size=small | 2 | 1 | 1 |

<pets>

| | value=low | value=medium | value=high |
|---|---|---|---|
| pets=no | 1 | 1 | 2 |
| pets=only cats | 0 | 2 | 0 |
| pets=yes | 2 | 1 | 1 |

✓ Feature와 결과 사이의 Cross-Table을 만듦

# 2) 예시

<location>

| | value=low | value=medium | value=high |
|---|---|---|---|
| location=bad | 3 | 2 | 0 |
| location=good | 0 | 2 | 3 |

<size>

| | value=low | value=medium | value=high |
|---|---|---|---|
| size=big | 0 | 0 | 2 |
| size=medium | 1 | 3 | 0 |
| size=small | 2 | 1 | 1 |

<pets>

| | value=low | value=medium | value=high |
|---|---|---|---|
| pets=no | 1 | 1 | 2 |
| pets=only cats | 0 | 2 | 0 |
| pets=yes | 2 | 1 | 1 |

- ✓ Feature 값에 대해 특정 Feature가 있는 인스턴스의 가장 빈번한 Class를 예측하는 규칙을 만듦
- ✓ Feature에 대한 오류 계산

- ✓ location
  - ✓ rule :
    if location = bad then value = low (3/5 acc)
    if location = good then value = high (3/5 acc)
  - ✓ error : 4/10

- ✓ size
  - ✓ rule :
    if size = big then value = high (2/2 acc)
    if size = medium then value = medium (3/4 acc)
    if size = small then value = low (2/4 acc)
  - ✓ error : 3/10

- ✓ pets
  - ✓ rule : …
  - ✓ error : 4/10

- ✓ 최종 rule = size의 rule과 동일

# 3) 특징

<location>

| | value=low | value=medium | value=high |
|---|---|---|---|
| location=bad | 3 | 2 | 0 |
| location=good | 0 | 2 | 3 |

<size>

| | value=low | value=medium | value=high |
|---|---|---|---|
| size=big | 0 | 0 | 2 |
| size=medium | 1 | 3 | 0 |
| size=small | 2 | 1 | 1 |

<pets>

| | value=low | value=medium | value=high |
|---|---|---|---|
| pets=no | 1 | 1 | 2 |
| pets=only cats | 0 | 2 | 0 |
| pets=yes | 2 | 1 | 1 |

- ✓ Feature 값에 대해 특정 Feature가 있는 인스턴스의 가장 빈번한 Class를 예측하는 규칙을 만듦
- ✓ Feature에 대한 오류 계산

- ✓ location
  - ✓ rule :
    if location = bad then value = low (3/5 acc)
    if location = good then value = high (3/5 acc)
  - ✓ error : 4/10

- ✓ size
  - ✓ rule :
    if size = big then value = high (2/2 acc)
    if size = medium then value = medium (3/4 acc)
    if size = small then value = low (2/4 acc)
  - ✓ error : 3/10

- ✓ pets
  - ✓ rule : …
  - ✓ error : 4/10

- ✓ 최종 rule = size의 rule과 동일

## 3) 특징

- ✓ level이 많으면 예측하기 좋기에 Feature로 선택될 가능성이 높음
  => 과적합 문제로 연결
  => Train-validation으로 검증하자.

- ✓ 두 특성 값이 동일할 때는
  첫 번째 특성값 선택 or chi-square test의 p-value로 해결

# 1) 개념 및 방법

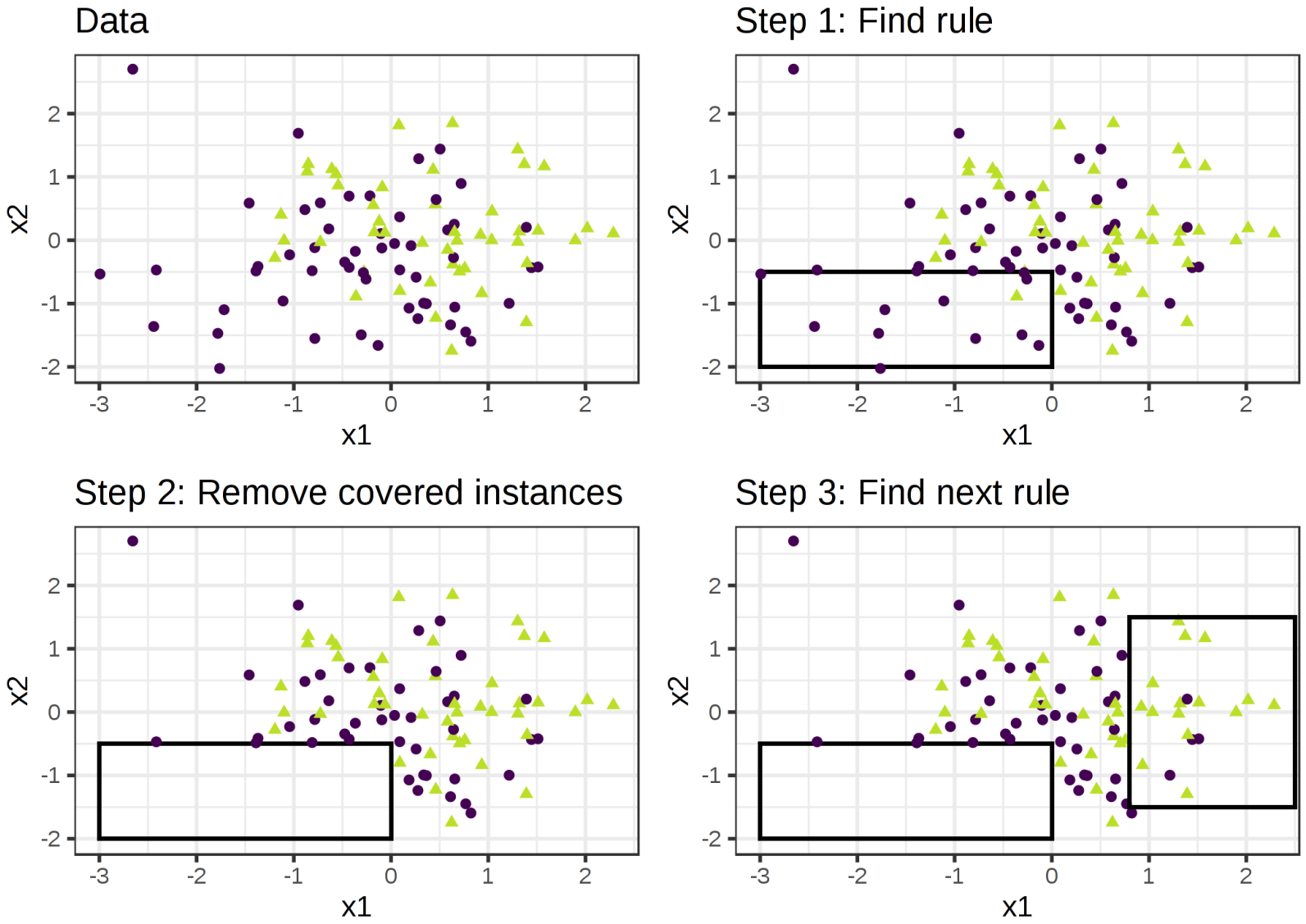
- ✓ 개념
  전체 데이터 집합 규칙을 다루는 Decision List(set)을 만들기 위해
  단일한 규칙을 반복적으로 학습하는 절차

- ✓ 방법
  - ✓ 빈 Decision List 생성

  - ✓ 규칙 r 학습
  - ✓ r을 list에 추가
  - ✓ r로 처리한 데이터 포인트 삭제
  - ✓ 남은 데이터로 r 학습

  - ✓ 중지 규칙 or 데이터가 없으면 중지

## 2) 예시


Data
Step 1: Find rule
Step 2: Remove covered instances
Step 3: Find next rule
x1
x2

## 1) 개념 및 방법

✓ 방법

1. 의사결정 규칙의 조건으로 사용할 수 있는 frequent pattern 채굴
2. 미리 지정된 규칙의 선택 항목에서 의사결정 목록을 학습

# 1) 개념 및 방법

✓ 패턴 채굴

$$Support(x_j = A) = \frac{1}{n}\sum_{i=1}^{n} I(x_j^{(i)} = A)$$

A : feature
n : 데이터 포인트의 수
I : Indicator Function
즉, i번째 instance가 특정 Feature일 경우 1 아니면 0에 대한 평균

그 외에 Apriori, FP-Growth 등이 있음

✓ Apriori 패턴 채굴

| itemsets | support |
|---|---|
| 기저귀 | 3/4 = 75% |
| 버터 | 1/4 = 25% |
| 맥주 | 2/4 = 50% |
| 빵 | 1/4 = 25% |
| 떡 | 1/4 = 25% |
| 사이다 | 1/4 = 25% |

| itemsets | support | Confidence |
|---|---|---|
| 기저귀 → 맥주 | 2/4 = 50% | 2/3 = 66.7% |
| 맥주 → 기저귀 | 2/4 = 50% | 2/2 = 100% |

## 1) 개념 및 방법

- ✓ BRL 학습
  rule의 길이 (짧을수록 좋음)
  rule의 수 (적을수록 좋음)

- ✓ 방법
  1. 사전 분포에서 무작위로 추출한 최초 의사결정 목록 생성
  2. 규칙 추가, 전환 ,삭제를 통해 리스트 반복 수정 => 사후 분포를 따르도록 함
  3. 사후 분포에 따라 확률이 가장 높은 목록에서 의사결정 목록 선택

## 1) 개념 및 방법

✓ BRL 학습

$$\underbrace{p(d|x,y,A,\alpha,\lambda,\eta)}_{posteriori} \propto \underbrace{p(y|x,d,\alpha)}_{likelihood} \cdot \underbrace{p(d|A,\lambda,\eta)}_{priori}$$

$p(d|x,y,A,\alpha,\lambda,\eta)$

d : 의사결정 목록
x : feature
y : 사전 채굴 조건 집합
알파 : (1,1)에서 가장 잘 고정된 양수 및 음수 클래스에 대한 사전 근사수
람다 : 의사결정 목록의 이전 예상 길이
에타 : 사전 예상 조건 수

=> 위의 모든걸 가지고 사후 분포로부터 특정 의사결정 목록 d를 샘플링 하겠다.

# 1) 개념 및 방법

✓ BRL 학습

$$\underbrace{p(d|x,y,A,\alpha,\lambda,\eta)}_{posteriori} \propto \underbrace{p(y|x,d,\alpha)}_{likelihood} \cdot \underbrace{p(d|A,\lambda,\eta)}_{priori}$$

$p(y|x,d,\alpha)$

d : 의사결정 목록
x : feature
y : 사전 채굴 조건 집합
알파 : (1,1)에서 가장 잘 고정된 양수 및 음수 클래스에 대한 사전 근사수

=> 의미
y의 우도
y가 디리클레 다항분포에 의해 생성될 때, d가 데이터를 얼마나 잘 설명하는가.

# 1) 개념 및 방법

✓ BRL 학습

$$\underbrace{p(d|x,y,A,\alpha,\lambda,\eta)}_{posteriori} \propto \underbrace{p(y|x,d,\alpha)}_{likelihood} \cdot \underbrace{p(d|A,\lambda,\eta)}_{priori}$$

$p(d|A,\lambda,\eta)$

d : 의사결정 목록
A : feature
람다 : 의사결정 목록의 이전 예상 길이
에타 : 사전 예상 조건 수

=> 의미
의사결정 목록의 사전 분포

✓ 즉,
의사결정 목록의 사전분포를 잘 따르고, 그에 따른 우도가 높으면 => 사후 확률이 높다.
이런 d를 고르자.

## 1) 장점

- ✓ IF-THEN 구조
  - ✓ 해석하기 쉽다.
  - ✓ 의사결정 나무처럼 표현할 수 있다.
- ✓ 빠르다.
- ✓ 단조로운 변화에 강력(Robust)함
- ✓ 이상치에 강력함

## 2) 단점

- ✓ IF-THEN 구조
  - ✓ 회귀 방법을 고려하지 않는다.
  - ✓ 특성값도 범주형 변수여야 함

  ⇒ 데이터의 loss

- ✓ 과적합 되기 쉬움

- ✓ 선형 관계를 설명하기 어려움

https://colab.research.google.com/drive/1uxYTcx2G-S2Eo_LuTF01iloDUqkL4NM0?usp=sharing