Chapter 6

Cross-Validation Based Conformal Methods

yoonseo choi

October 13, 2025

목차

목차

복습

Full Conformal

- **장점:** 모든 데이터를 활용하여 통계적으로 효율적 (좁은 예측 구간)
- **단점:** 새로운 Test Point마다 모델을 반복적으로 학습해야 하므로 계산 비용이 매우 높음

Split Conformal

- **장점:** 모델 학습이 한 번만 필요하여 계산적으로 매우 효율적
- **단점:** 데이터를 Training/Calibration으로 분할하여 통계적 효율성 감소 (넓은 예측 구간)

핵심 질문

Split Conformal의 통계적 비효율성을 개선하면서, Full Conformal의 막대한 계산 비용을 피할 수 있는 중간 지점은 없을까?

해결책: Cross-Validation (CV)

Cross-Validation type 방법

- cross conformal
- CV+
- jackknife+
- 위의 세 가지 방법의 정의, 커버리지 보장에 대해 알아보자.

목차

Split Conformal 복습

1. 전체 데이터 $[n]$을 Training set $\mathcal{D}_{[n]\setminus I}$와 Calibration set $\mathcal{D}_I$로 분할
2. $\mathcal{D}_{[n]\setminus I}$으로 모델 학습, $\mathcal{D}_I$으로 score 계산.
   - $s_i = s((X_i, Y_i); \mathcal{D}_{[n]\setminus I})$ for $i \in I$
   - 새로운 테스트 데이터 (X_{n+1}, y)에 대한 score는 $s_{n+1} = s((X_{n+1}, y); \mathcal{D}_{[n]\setminus I})$
3. s_{n+1}가 calibration set으로 만든 score들의 $(1-\alpha)(1+1/|I|)$분위수보다 **작으면** y를 예측구간에 포함

Split Conformal Prediction Interval

$$\mathcal{C}(X_{n+1}) = \{y : s_{n+1} \leq \text{Quantile}\left(\{s_i\}_{i\in I}, (1-\alpha)(1+1/|I|)\right)\}$$

새로운 관점: 토너먼트(Tournament)

- Split CP의 예측 구간을 토너먼트 관점으로 재해석해보자.

토너먼트로서의 Split Conformal

$$\mathcal{C}(X_{n+1}) = \left\{ y : \sum_{i \in I} \mathbb{I}\{s_{n+1} > s_i\} < (1-\alpha)(|I|+1) \right\}$$

- score를 구하는 각 point(calibration set과 test point)=팀
- 각 points들의 score=힘
- 경기 규칙: 두 팀이 경기할 때, 점수가 더 높은 팀이 '승리'

test point가 split CP 예측구간에 포함된다
=팀 test point가 Calibration 팀들과의 경기에서
$(1-\alpha)(|I|+1)$ 판 **미만으로** 승리했다

목차

Cross-Conformal

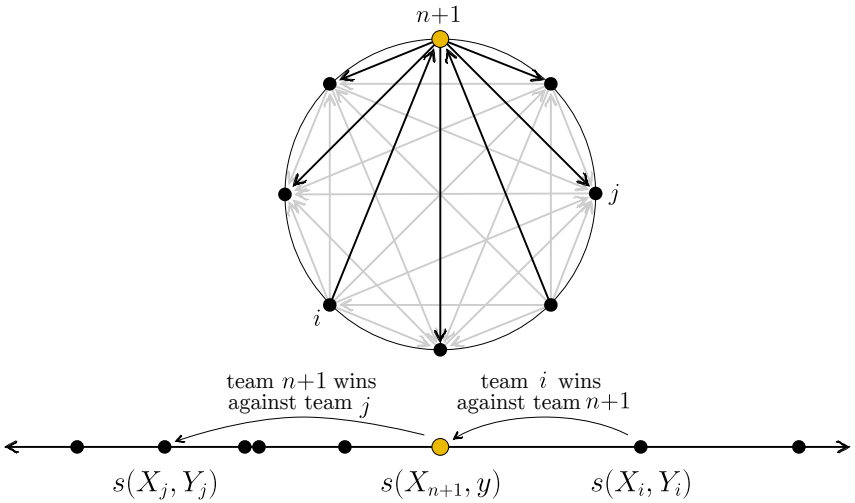
- Split Conformal의 토너먼트 개념을 Cross-Validation으로 확장해보자.
- CC 예측구간 만드는 방법
  1. 전체 데이터 $[n]$을 K개의 partition fold $I_1, \ldots, I_K$로 분할
  2. 각 폴드 $k = 1, \ldots, K$에 대해 다음을 반복:
     1. I_k는 calibration set, 나머지 데이터
     2. $\mathcal{D}_{[n]\setminus I_k}$로 모델 학습
     3. Test point와 I_k의 데이터 포인트들 간 토너먼트 진행
  3. 모든 K개의 토너먼트 결과를 합산하여 최종 예측 셋을 결정

$$\mathcal{C}(X_{n+1}) = \left\{ y : \sum_{k=1}^{K} \sum_{i \in I_k} \mathbb{I}\{ s((X_{n+1}, y); \mathcal{D}_{[n]\setminus I_k}) > s((X_i, Y_i); \mathcal{D}_{[n]\setminus I_k}) \} < (1-\alpha)(n+1) \right\}$$

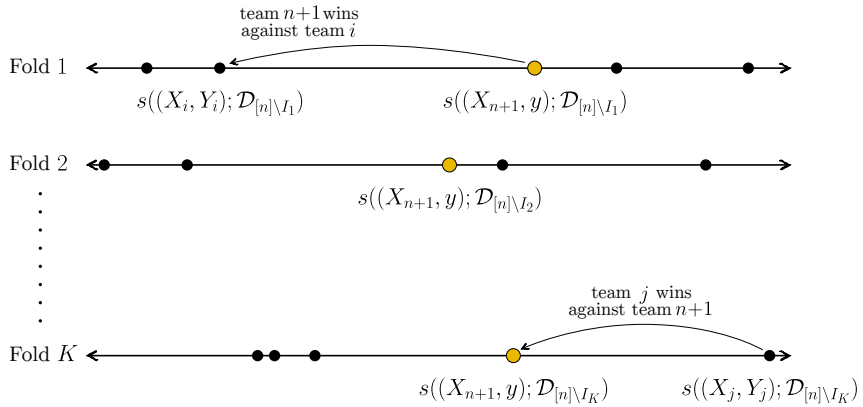
- split CP와의 차이점: test point의 힘이 **상대방이 속한 폴드** k에 따라 계속 변함!

Cross-Conformal

- split CP와의 차이점: test point의 힘이 **상대방이 속한 폴드** k에 따라 계속 변함!



(a) split CP의 토너먼트



(b) CC의 토너먼트

- $K = n$인 경우를 **Leave-One-Out Cross-Conformal** 이라고 한다.

CC의 Coverage Guarantee

결론부터 말하면...

Cross-Conformal의 marginal coverage는 목표 수준인 $1-\alpha$가 아닌 $\approx 1-2\alpha$ 수준으로 보장된다.

- 이에 관한 정리 3개를 보자.
  1. **Theorem 6.1:** p-value averaging 기반. K가 작을 때 유리함.
  2. **Theorem 6.2:** Tournament-matrix 기반. K가 클 때 유리함.
  3. **Corollary 6.3:** 위의 두 정리를 합친 것.

Theorem 6.1 (p-value averaging)

Theorem 6.1

데이터 $(X_1, Y_1), \ldots, (X_{n+1}, Y_{n+1})$이 exchangeable 하다면, K-fold Cross-Conformal은 다음을 만족한다:

$$\mathbb{P}(Y_{n+1} \in \mathcal{C}(X_{n+1})) \geq 1 - 2\alpha - 2(1-\alpha) \cdot \frac{1 - 1/K}{n/K + 1}$$

- **증명 아이디어**
  1. 각 폴드 k에서 p-value p_k를 정의(p. 15 참고)
  $$p_k(x, y) = \frac{1 + \sum_{i \in I_k} \mathbb{I}s((x, y); \mathcal{D}_{[n]\setminus I_k}) \leq s((X_i, Y_i); \mathcal{D}_{[n]\setminus I_k})}{1 + n/K}.$$
  2. 알고 있는 사실 1: 데이터가 exchangeable하므로 Cor 2.6 적용됨. 즉, $\mathbb{P}(p_k(X_n + 1, Y_n + 1) \leq t) \leq t$)
  3. 알고 있는 사실 2: 여러 p-value를 평균한 값은 원래 p-value보다 보수적(최대 2배)인 p-value가 됨 ($\mathbb{P}(\frac{1}{K}\sum p_k \leq t) \leq 2t$).
  4. CC의 정의는 p_k로 표현할 수 있으므로, 위의 사실을 사용하여 쉽게 증명됨

Theorem 6.2 (Tournament-matrix)

Theorem 6.2

$(X_1, Y_1), \ldots, (X_{n+1}, Y_{n+1}), (X_{n+2}, Y_{n+2}, \ldots, (X_{n/K}, Y_{n/K})$가 exchangeable하고 score function가 symmetry을 만족하면 CC는 다음을 만족한다:

$$\mathbb{P}(Y_{n+1} \in \mathcal{C}(X_{n+1})) \geq 1 - 2\alpha - 2(1-\alpha) \cdot \frac{1 - K/n}{K+1}$$

lemma 6.4: Tournament Lemma

Let $A \in \{0,1\}^{N \times N}$ satisfy $A_{ij} + A_{ji} \leq 1$ for all i, j. Then for any $t \in [0,1]$,

$$\sum_{i=1}^{N} \mathbb{I} \sum_{j=1}^{N} A_{ij} \geq N(1-t) \leq 2tN.$$

Proof of Theorem 6.2: 핵심 요약 (1/3)

Step 1: 토너먼트 구성 (Tournament Construction)

가상 토너먼트 설정

- 가상의 test point들(K/n-1개)을 추가하여, 총 $K+1$개의 fold가 참가.
- 다른 fold에 있는 팀끼리만 경기를 함.
- 두 팀 i, j가 경기할 때의 점수를 담는 행렬 S, 이를 바탕으로 승패 정보를 저장하는 행렬 A를 다음과 같이 정의한다:

$$S_{ij} = \begin{cases} s\big((X_i, Y_i); \mathcal{D}_{[n+n/K]\setminus(I_k \cup I_\ell)}\big), & \text{if } i,j \text{ are in different folds: } i \in I_k,\ j \in I_\ell,\ k \neq \ell, \\ 0, & \text{if } i,j \text{ are in the same fold.} \end{cases}$$

$$A_{ij} = \mathbb{I}(S_{ij} > S_{ji})$$

- 즉, S_{ij}는 i, j가 속해있는 fold들을 제외한 데이터셋으로 훈련한 후 구한 i번째 point의 score, A_{ij}는 팀 i와 팀 j의 싸움에서 팀 i가 이겼으면 1, 아니면 0.

Proof of Theorem 6.2: 핵심 요약 (1/3)

Step 1: 토너먼트 구성 (Tournament Construction)

- **Lemma 6.4 (Tournament Lemma)**을 적용하면, 다음과 같은 식을 얻을 수 있다.

$$\sum_{i=1}^{n+n/K} \mathbb{I}\sum_{j=1}^{n+n/K} A_{ij} \geq (1-\alpha)(n+1) \leq 2\left(\alpha + (1-\alpha)\cdot\frac{n/K-1}{n+n/K}\right)\cdot(n+n/K).$$

Proof of Theorem 6.2

Step 2: 토너먼트와 예측셋 연결

Miscoverage와 토너먼트의 관계

- for any $k \in [K]$ and $i \in I_k$

$$s((X_{n+1}, y); \mathcal{D}_{[n]\setminus I_k}) > s((X_i, Y_i); \mathcal{D}_{[n]\setminus I_k}) \iff A_{n+1,i} = 1$$

- 따라서

$$Y_{n+1} \notin \mathcal{C}(X_{n+1}) \iff \sum_{i=1}^{n} A_{n+1,i} \geq (1-\alpha)(n+1)$$

Proof of Theorem 6.2

Step 3: 교환가능성 적용

1. If $i, j \in I_k$ for some $k \in [K+1]$, then $\sigma(i), \sigma(j) \in I_\ell$ for some $\ell \in [K+1]$ 라는 조건(6.5)을 만족하는 σ를 생각해보자. 또한 permutation을 적용한 토너먼트 행렬 A를 다음과 같이 정의한다: $(A_\sigma)_{ij} = A_{\sigma(i),\sigma(j)}$
2. 위의 (6.5)조건은 동치가 되고, exchangeable 조건때문에 (6.5)를 만족하는 σ에 대해,

$$A = A(\mathcal{D}_{n+n/K}) \overset{d}{=} A((\mathcal{D}_{n+n/K})_\sigma)$$

이다.

3. 또한, score function이 symmetric이므로

$$A_\sigma = A((\mathcal{D}_{n+n/K})_\sigma)$$

임. 따라서

$$A \overset{d}{=} A_\sigma$$

Proof of Theorem 6.2

Step 3: 교환가능성 적용

1. $\mathrm{P}(Y_{n+1} \notin \mathcal{C}(X_{n+1}))$

$$= \mathrm{P}\left(\sum_{j=1}^{n+n/K} A_{\sigma(n+1),j} \geq (1-\alpha)(n+1)\right)$$

$$= \mathrm{P}\left(\sum_{j=1}^{n+n/K} A_{ij} \geq (1-\alpha)(n+1)\right)$$

$$= 1 \frac{}{n+n/K \sum_{i=1}^{n+n/K} \mathbb{P}\left(\sum_{j=1}^{n+n/K} A_{ij} \geq (1-\alpha)(n+1)\right)}$$

$$= \mathrm{E}\left[\frac{1}{n+n/K} \sum_{i=1}^{n+n/K} \mathbb{I}\left(\sum_{j=1}^{n+n/K} A_{ij} \geq (1-\alpha)(n+1)\right)\right] \leq$$

$$2\alpha + 2(1-\alpha) \cdot \frac{n/K-1}{n+n/K} (\because \mathit{lemma} 6.4)$$

두 정리의 결합 (Corollary 6.3)

Corollary 6.3

Theorem 6.2의 가정 하에서, Cross-Conformal은 다음을 만족한다:

$$\mathbb{P}(Y_{n+1} \in \mathcal{C}(X_{n+1})) \geq 1 - 2\alpha - 2(1-\alpha) \cdot \min\left\{\frac{1 - 1/K}{n/K + 1}, \frac{1 - K/n}{K+1}\right\}$$

왜 Coverage가 $1 - 2\alpha$인가? (The factor of 2)

coverage가 $1 - \alpha$에 가깝게 보장되었던 full CP와 비교해보자.

"Factor of 2" 문제의 원인: Transitivity 실패

- **Full Conformal의 경우:**
  - 모든 점수 S_i는 **동일한** 데이터셋 $\mathcal{D}_{n+1}$을 기반으로 계산됨.
  - 따라서 모든 데이터 포인트들 사이에 일관된 순서가 존재
  - 이는 토너먼트의 **추이성(Transitivity)**을 보장: 만약 A가 B를 이기고, B가 C를 이기면, C는 A를 이길 수 없음 ($A \to B, B \to C \implies A \to C$).
- **Cross-Conformal의 경우:**
  - 각 경기(A vs B, B vs C, C vs A)는 **서로 다른** 모델(다른 fold를 제외하고 학습된)을 사용하여 진행.
  - 이로 인해 추이성이 깨질 수 있음. 즉, $A \to B, B \to C$ 이지만 $C \to A$ 인 순환 관계(cycle)가 발생할 수 있음.

결론

이러한 transitivity의 실패 가능성이 coverage guarantee를 1-2α로 만든다.

질의 응답

- Q. 98페이지 하단부에 추가 가정이 없는 한, cross-conformal의 miscoverage rate이 2 alpha가 될 수 있다고 나와 있습니다. 어떤 가정을 추가하면 alpha 수준으로 내릴 수 있을까요?
- A. 알고리즘 안정성이 보장되면 alpha 수준으로 내릴 수 있을 거라 생각합니다. CC의 miscoverage rate이 2 alpha까지 올라간 이유는 transitivity가 실패했기 때문인데, 알고리즘 안정성이 보장되면, 즉 모델 훈련에 사용한 데이터가 달라져도 모델의 적합 결과가 크게 달라지지 않는다면 서로 다른 훈련 데이터셋으로 모델을 적합했다고 하더라도 그 결과가 크게 달라지지 않아서 transitivity를 조금 더 보장하게 되어 miscoverage rate이 내려가지 않을까 생각합니다.

목차

고전적인 CV 예측 구간

- CC는 잠깐 버려두고...전통적인 CV방법을 떠올려보자.

1. 데이터를 K-fold로 분할.
2. 각 폴드 k에 대해, 해당 폴드를 제외한 데이터로 모델 $\hat{f}_{-I_k}$를 학습.
3. 각 데이터 i에 대해 out-of-sample 잔차 $S_i = |Y_i - \hat{f}_{-I_{k(i)}}(X_i)|$를 계산. (여기까지는 CC랑 동일)
4. 잔차들의 $(1-\alpha)$-quantile인 $\hat{q}_{CV}$를 계산. 즉,

$$\hat{q}_{CV} = \text{Quantile}\left((S_i)_{i\in[n]}, (1-\alpha)\right)$$

5. **전체 데이터로 재학습**한 모델 $\hat{f}$를 이용하여 최종 예측 구간을 생성:

Standard CV Interval

$$C(X_{n+1}) = \hat{f}(X_{n+1}) \pm \hat{q}_{CV}$$

- S_i들이 근사적으로 $\hat{f}$의 unbiased estimater이므로 일반적으로 잘 작동함. 하지만...

표준 CV의 실패 사례 (Example 6.5)

표준 CV는 assumption-free coverage를 보장하지 못함.

- **가상의 불안정한 알고리즘 A을 생각해보자.**
  - training set이 **짝수** 개이면 $\hat{f}(x) = 0$
  - training set이 **홀수** 개이면 $\hat{f}(x) = 1$
- **아래와 같은 상황이라면...**
  - 실제 Y의 값은 항상 0 ($\mathbb{P}(Y = 0) = 1$).
  - 전체 데이터 수 n과 각 fold의 크기 n/K가 모두 **홀수**.
- **결과:**
  1. 잔차 계산 시, training set은 $n - n/K$개 (짝수) $\implies \hat{f}_{-I_k}(x) = 0$.
     따라서 모든 잔차 $S_i = |0 - 0| = 0 \hat{q}_{CV} = 0$
  2. 최종 예측 시, 학습 데이터는 n개 (홀수) $\implies \hat{f}(x) = 1$.
  3. 최종 예측 구간: $C(X_{n+1}) = 1 \pm 0 = \{1\}$.
  4. 실제 참값은 0이므로, **Coverage Probability = 0**.

문제의 원인과 해결책

표준 CV의 근본적인 문제

잔차 $S_i = |Y_i - \hat{f}_{-I_{k(i)}}(X_i)|$와 우리가 추정하려는 test error $|Y_{n+1} - \hat{f}(X_{n+1})|$는 서로 다른 분포에서 나옴.

- S_i: $n - n/K$개의 데이터로 학습된 모델의 오차.
- Test error: n개의 데이터로 학습된 모델의 오차.

해결책: CV+ 와 Jackknife+

- 핵심 아이디어: 전체 데이터로 학습한 $\hat{f}(X_{n+1})$ 대신, 잔차 계산에 사용된 leave-one-out 모델들의 예측값 $\hat{f}_{-I_k}(X_{n+1})$을 사용해서 예측 구간을 만들자!
- 비교 대상의 분포가 같아져서 문제 해결됨.

K-fold CV+ 공식 (Eq. 6.13)

- 기존의 CV 예측구간을 다시 나타내면

$$\mathcal{C}(X_{n+1}) = \Big[-\text{Quantile}\left(\{-(\hat{f}(X_{n+1}) - S_i)\}_{i\in[n]}, (1-\alpha)\right), \\ \text{Quantile}\left(\{\hat{f}(X_{n+1}) + S_i\}_{i\in[n]}, (1-\alpha)\right)\Big]$$

- 여기에서 $\hat{f}$ 대신 $\hat{f}_{-I_{k(i)}}$을, (1-α) 대신 (1-2α)를 사용한게 CV+
- $S_i = |Y_i - \hat{f}_{-I_{k(i)}}(X_i)|$는 기존과 동일하게 계산.

K-fold CV+ Prediction Interval

$$\mathcal{C}(X_{n+1}) = \Big[-\text{Quantile}\left(\{-(\hat{f}_{-I_{k(i)}}(X_{n+1}) - S_i)\}_{i\in[n]}, (1-\alpha)(1+1/n)\right), \\ \text{Quantile}\left(\{\hat{f}_{-I_{k(i)}}(X_{n+1}) + S_i\}_{i\in[n]}, (1-\alpha)(1+1/n)\right)\Big]$$

- $K = n$인 특수한 경우를 **Jackknife+**라고 부름.* 책에 오타 있음

질의 응답

- Q. 6.12와 6.13 (또는 6.14)의 range차이가 symmetric interval인지 아닌지 차이라고 적혀있는데 (p.101 6.3.4 도입부) 약간 햇갈리네요. 6.14는 symmetric이 아님이 바로 보이는데 (quantile 값이 다름) 6.13은 symmetric한게 아닌가요?
- A. 6.14 jackknife+의 예측구간의 분위수가 잘못 써진 것으로 보입니다.
- A. symmetric interval이 아니라는 말은, 전체 훈련 데이터셋으로 훈련한 모델로 구한 예측값인 $\hat{f}(X_{n+1})$을 중심으로 symmetric이 아니라는 뜻이고, CV+와 jackknife+로 구한 예측구간은 symmetric인건 맞습니다.

CV+ 구간의 해석

- CV+로 만들어진 구간은 더 이상 $\hat{f}(X_{n+1})$를 중심으로 한 대칭적인 구간이 아님.
- 심지어 $\hat{f}(X_{n+1})$가 구간에 포함되지 않을 수도 있음! 하지만...

Proposition 6.6

$\alpha \leq 0.5$일 때, CV+ 예측 구간은 항상 leave-one-fold-out 예측값들의 중앙값을 포함한다.

$$\hat{f}^{\text{med}}(X_{n+1}) = \text{Median}\left(\{\hat{f}_{-I_k}(X_{n+1})\}_{k\in[K]}\right) \in \mathcal{C}(X_{n+1})$$

- $\alpha \leq 0.5$ 조건과 $S_i \geq 0$임을 이용해서 $\hat{f}^{\text{med}}(X_{n+1})$가 $\mathcal{C}(X_{n+1})$의 상한보다 작고, 하한보다 큼을 보이면 됨.

목차

두 방법론의 관계 (Proposition 6.7)

- 표면적으로 CV+와 Cross-Conformal(CC)은 달라보임. 하지만...
  - CV+: 항상 구간을 반환.
  - CC: 분리된 집합을 반환할 수도 있음.

Proposition 6.7

Cross-Conformal에서 score function을 $s((x,y);\mathcal{D}) = |y - \hat{f}_{\mathcal{D}}(x)|$로 정의하면, Cross-Conformal 예측 셋은 항상 CV+ 예측 구간에 포함된다.

$$\mathcal{C}^{\mathrm{CC}}(X_{n+1}) \subseteq \mathcal{C}^{\mathrm{CV+}}(X_{n+1})$$

- 즉, CV+가 CC보다 더 conservative할 수 있음. (예측구간을 더 넓게 잡음)
- 실제로는 두 방법의 결과가 동일하게 나오는 경우가 많음.

CV+의 Coverage Guarantee (Corollary 6.8)

- Proposition 6.7의 포함 관계 덕분에, Cross-Conformal에 대한 coverage guarantee가 CV+에도 즉시 적용됨.

Corollary 6.8

Jackknife+와 CV+는 다음의 marginal coverage를 만족한다.

- **Jackknife+ ($K = n$):**

$$\mathbb{P}(Y_{n+1} \in \mathcal{C}(X_{n+1})) \geq 1 - 2\alpha$$

- **K-fold CV+:**

$$\mathbb{P}(Y_{n+1} \in \mathcal{C}(X_{n+1})) \geq 1 - 2\alpha - 2\sqrt{n}$$

여전히 존재하는"Factor of 2"

CV+ 역시 이론적으로는 목표치보다 2배 넓은 오차율(2α)을 보장.

목차

Training-Conditional Coverage

- **질문:** CC/CV+/jackknife+의 training-conditional coverage가 보장되는가?
  $\Rightarrow$ CC/CV+는 보장이 되긴 되지만 여전히 factor of 2 문제가 있음
  $\Rightarrow$ 심지어 jackknife+는 coverage가 0이 될 수도 있음.
- **Theorem 6.9 (K-fold CC/CV+):**
  - 각 폴드의 크기 n/K가 충분히 크다면, CC에 대해서
    $\alpha_P(\mathcal{D}_n)=\mathbb{P}(Y_{n+1} \notin \mathcal{C}(X_{n+1})|\mathcal{D}_n) \geq 1-\alpha$ 라 할 때,
    $\mathbb{P}\left(\alpha_P(\mathcal{D}_n) \leq 2\alpha + \sqrt{\frac{2\log(K/\delta)}{n/K}}\right) \geq 1-\delta$
  - 즉, 높은 확률로 CC의 training conditional coverage가 2α에 가깝게 보장됨.
  - CV+는 CC보다 conservative하므로 CV+에 대해서도 thm 6.9가 성립함.

Training-Conditional Coverage

- **증명 아이디어**
  1. **알고 있는 사실 1:** $\mathcal{D}_{[n]\setminus I_k}$가 주어졌을 때 $(S_i)_{i\in I_k}$의 경험적 분포와 참 분포가 거의 비슷하다. (DKW 부등식)
  2. 폴드 k에서 p-value p_k를 정의(S_i의 참 분포 사용)

$$p_k^*(x,y) = \mathbb{P}_{S\sim P_k}\big(S \geq s((x,y);\mathcal{D}_{[n]\setminus I_k}) \mid \mathcal{D}_n\big).$$

  3. **알고 있는 사실 2:** 여러 p-value를 평균한 값은 원래 p-value보다 보수적(최대 2배)인 p-value가 됨 ($\mathbb{P}(\frac{1}{K}\sum p_k^* \leq t) \leq 2t$).
  4. 위의 사실들을 이용하면 경험적 분포에 대한 확률 상한을 구할 수 있음.

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\big(S_i \geq s((X_{n+1},Y_{n+1});\mathcal{D}_{[n]\setminus I_k})\big) \leq \alpha \;\middle|\; \mathcal{D}_n\right) \leq 2\alpha + \sqrt{\frac{2\log(K/\delta)}{n/K}}$$

  5. CC의 정의에 따라

$$Y_{n+1} \notin \mathcal{C}(X_{n+1}) \;\Rightarrow\; \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\big(S_i \geq s((X_{n+1},Y_{n+1});\mathcal{D}_{[n]\setminus I_k})\big) \leq \alpha$$

- **Theorem 6.10 (Jackknife+):**
  Jackknife+에서 training conditional coverage는 **실패할 수 있다**.

$$\mathbb{P}\left(\alpha_P(\mathcal{D}_n) = 1\right) \geq \alpha - \mathcal{O}\left(\sqrt{\frac{\log n}{n}}\right),$$

- **증명 아이디어**
  jackknife+에서 training conditional coverage가 실패하는 A를 만들자.
  1. A를 데이터셋이 주어지면

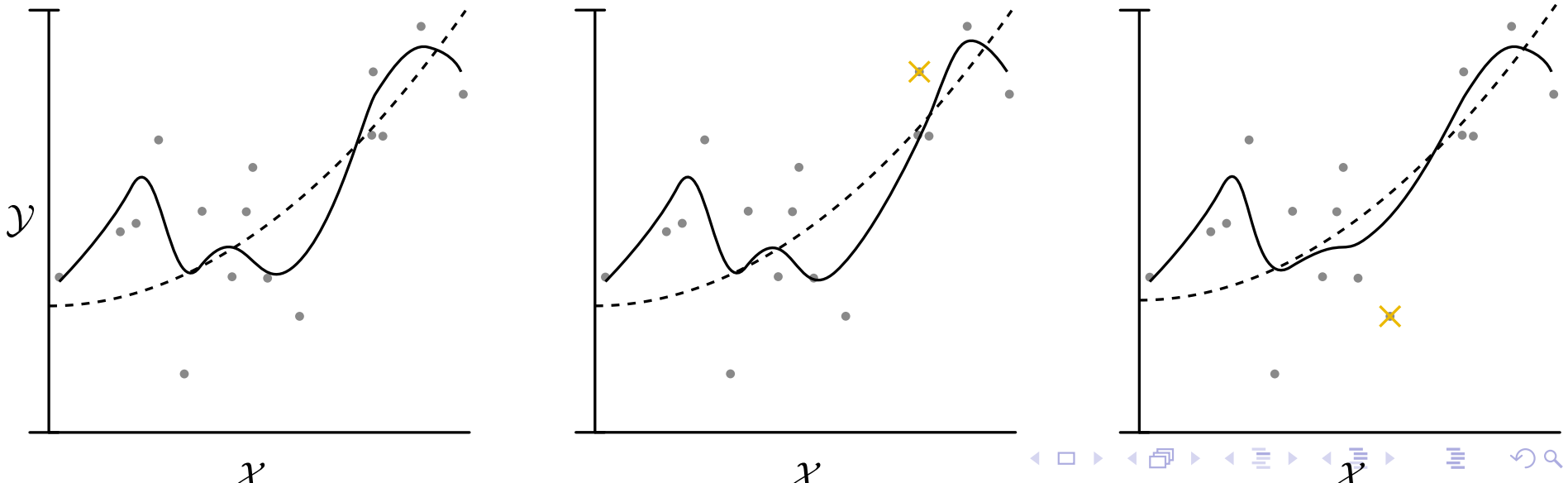
$$\hat{f}(x) = \begin{cases} 0, & \text{if mod}\left(a(x) + \sum_{j=1}^{n-1} a(x_j), n\right) < N, \\ 2B, & \text{otherwise.} \end{cases}$$

     를 반환하는 알고리즘으로 설정한다면, A는 값이 하나만 달라져도 $\hat{f}$가 엄청 달라지는 불안정한 알고리즘이 됨.
  2. 이후, thm 4.3과 동일하게 증명하면 됨.

목차

실패의 원인: Algorithmic Instability

- Jackknife/CV 방식이 실패하는 근본적인 원인은 알고리즘의 **불안정성(instability)**.
- **불안정한 알고리즘:** 학습 데이터에서 단 하나의 데이터 포인트가 추가되거나 제거될 때, 모델의 예측이 크게 변하는 알고리즘. (e.g., Example 6.5의 짝/홀수 모델)
- **안정적인 알고리즘(Stable Algorithm):** 학습 데이터의 작은 변화가 모델 예측에 큰 영향을 미치지 않는 알고리즘. (e.g., Ridge Regression, K-NN)



안정성 가정 하의 Jackknife 보장 (Thm 6.12)

- 알고리즘이 안정적이라는 가정을 추가하면, "factor of 2" 문제를 해결하고 목표 coverage를 거의 달성할 수 있음

Definition 6.11: Algorithmic Stability

알고리즘 A가 안정적이라는 것은, 데이터 포인트 하나를 제외하고 학습한 모델($\hat{f}_{-i}$)과 전체 데이터로 학습한 모델($\hat{f}$)의 예측값 차이가 작은 확률(δ)을 제외하고는 작은 값(ϵ) 이하임을 의미한다.

$$\mathbb{P}(|\hat{f}(X_{n+1}) - \hat{f}_{-i}(X_{n+1})| \leq \epsilon) \geq 1 - \delta$$

Theorem 6.12

알고리즘 A가 안정적이라면, 약간 확장된(inflated) **표준 Jackknife** 구간 $C(X_{n+1}) = \hat{f}(X_{n+1}) \pm (\hat{q}_{CV} + \epsilon)$은 다음을 만족한다:

$$\mathbb{P}(Y_{n+1} \in C(X_{n+1})) \geq 1 - \alpha - 2\sqrt{\delta} - \frac{1}{n+1}$$

thm 6.12 증명

1. For each $i \in [n+1]$, define a fitted model $\tilde{f}_{-i} = \mathcal{A}(\mathcal{D}_{[n+1]\setminus\{i\}})$. Define also a modified leave-one-out residual $\tilde{S}_i = |Y_i - \tilde{f}_{-i}(X_i)|$,for each $i \in [n+1]$.
2. Since $\mathcal{A}$ is a symmetric algorithm, $\tilde{S}_1, \ldots, \tilde{S}_{n+1}$ are exchangeable. Therefore
$$\mathbb{P}\left(\tilde{S}_{n+1} \leq \mathit{Quantile}(\tilde{S}_1, \ldots, \tilde{S}_{n+1}; 1-\alpha')\right) \geq 1-\alpha',$$
for any $\alpha' \in [0,1]$.($\because$ Fact 2.15 (ii))
3. By the Replacement Lemma (Lemma 3.4), together with the fact that $\tilde{S}_{n+1} = |Y_{n+1} - \hat{f}(X_{n+1})|$ by definition, we then have
$$\mathbb{P}\left(|Y_{n+1} - \hat{f}(X_{n+1})| \leq \mathit{Quantile}\big((\tilde{S}_i)_{i\in[n]}; (1-\alpha')(1+1/n)\big)\right) \geq 1-\alpha'.$$
4. On the other hand, by definition of the inflated jackknife prediction interval $\mathcal{C}(X_{n+1})$, we have
$$Y_{n+1} \in \mathcal{C}(X_{n+1}) \iff |Y_{n+1} - \hat{f}(X_{n+1})| \leq \hat{q}_{\mathrm{CV}} + \epsilon.$$

5. Then,

$$\mathbb{P}(Y_{n+1} \in \mathcal{C}(X_{n+1})) = \mathbb{P}(|Y_{n+1} - \hat{f}(X_{n+1})| \leq \hat{q}_{\mathsf{CV}} + \epsilon)$$

$$= \mathbb{P}(\tilde{S}_{n+1} \leq \hat{q}_{\mathsf{CV}} + \epsilon)$$

and

$$\mathbb{P}(\tilde{S}_{n+1} \leq Q)$$

$$= \mathbb{P}(\tilde{S}_{n+1} \leq Q \text{ and } Q \leq \hat{q}_{cv} + \epsilon) + \mathbb{P}(\tilde{S}_{n+1} \leq Q \text{ and } Q > \hat{q}_{cv} + \epsilon)$$

$$\leq \mathbb{P}(\tilde{S}_{n+1} \leq \hat{q}_{cv} + \epsilon) + \mathbb{P}(Q > \hat{q}_{cv} + \epsilon)$$

Therefore,

$$\mathbb{P}(Y_{n+1} \in \mathcal{C}(X_{n+1}))$$

$$\geq 1 - \alpha' - \mathbb{P}\left(\hat{q}_{\mathsf{CV}} + \epsilon < \mathit{Quantile}\left((\tilde{S}_i)_{i\in[n]}; (1-\alpha')(1+1/n)\right)\right).$$

6. Plugging in our definition of $\hat{q}_{\mathrm{CV}}$, our last step is to bound the probability $\mathbb{P}(S_{(k)} + \epsilon < \tilde{S}_{(k')})$ where $k = \lceil (1-\alpha)n \rceil$ while $k' = \lceil (1-\alpha')(n+1) \rceil$. By definition of the order statistics, $S_{(k)} + \epsilon < \tilde{S}_{(k')} \Rightarrow \tilde{S}_i > S_i + \epsilon$ for at least $k - k' + 1$ many $i \in [n]$.

$$
\begin{aligned}
\mathbb{P}(S_{(k)} + \epsilon < \tilde{S}_{(k')}) &\leq \mathbb{P}\left(\sum_{i \in [n]} \mathbb{I}(\tilde{S}_i > S_i + \epsilon) \geq k - k' + 1 \right) \\
&\leq \mathbb{P}\left(\sum_{i \in [n]} \mathbb{I}(|\tilde{f}_{-i}(X_i) - \hat{f}_{-i}(X_i)| > \epsilon) \geq k - k' + 1 \right) \\
&\leq \frac{\mathbb{E}\left[\sum_{i \in [n]} \mathbb{I}(|\tilde{f}_{-i}(X_i) - \hat{f}_{-i}(X_i)| > \epsilon) \right]}{k - k' + 1} \quad \text{by Markov's inequality} \\
&= \frac{\sum_{i \in [n]} \mathbb{P}(|\tilde{f}_{-i}(X_i) - \hat{f}_{-i}(X_i)| > \epsilon)}{k - k' + 1} \\
&\leq \frac{n\delta}{k - k' + 1} \quad \text{by algorithmic stability}
\end{aligned}
$$

7. Finally, by construction, $k - k' + 1 \geq (1-\alpha)n - (1-\alpha')(n+1)$, and so combining everything, we obtain

$$\mathbb{P}(Y_{n+1} \in C(X_{n+1})) \geq 1 - \alpha' - \frac{n\delta}{(1-\alpha)n - (1-\alpha')(n+1)}.$$

Choosing $\alpha' = \alpha + \sqrt{\delta} + \frac{1}{n+1}$ completes the proof.

안정성: 검증 가능한가?

- K-NN과 같은 일부 간단한 알고리즘을 제외하면, 주어진 알고리즘이 안정적인지를 이론적으로 증명하거나, 제한된 데이터로 경험적으로 **검증하는 것은 일반적으로 불가능**.
- 때문에 안정성 가정 하에 만족하는 방법을 (ex: Jackknife) assumption-free 방법으로 보기 어려움.

목차

Chapter 6 요약

- **Cross-Conformal (CC):** Split Conformal을 CV로 확장한 개념. 계산 효율성과 통계적 효율성의 균형을 맞춤.
  - 단, 이론적 coverage는 $\approx 1 - 2\alpha$로 보장됨 (Transitivity 실패 때문).
- **CV+ & Jackknife+:** 전통적인 CV의 실패 사례를 보완하기 위해 수정된 방법.
  - CC 예측 셋을 항상 포함하며, 유사한 $1 - 2\alpha$ coverage를 보장받음.
  - 예측 구간이 앙상블 예측의 중앙값을 포함하는 등 해석에 이점이 있음.
- **Algorithmic Stability:**
  - 이 가정이 추가되면, (표준) Jackknife 방법이 목표 coverage $1 - \alpha$에 가까운 보장을 얻을 수 있음.
  - 하지만 안정성 자체는 검증이 어려운'가정'으로 남음.

질의 응답

- Q. Cross validation fold의 개수를 K(1<=K<=n)라 할 때, K=1일 때의 cross-conformal prediction이 앞 단원들에서 계속 등장했던 split conformal prediction과 일치한다고 이해했습니다. 그러나 제가 이해한 것이 맞다면 K=n일 때의 cross-conformal prediction (leave-one-out cross-conformal prediction)은 full conformal prediction과는 다른 것으로 알고 있는데, 이 두 알고리즘이 computational/statistical efficiency 측면이나 marginal/conditional coverage guarantee 등의 이론적 성질과 관련된 측면에서 구체적으로 어떤 차이가 있는지, 그리고 서로와 비교하여 각각 어떤 장/단점이 있는지 궁금합니다.
- A. 우선, K=1일 때의 CC와 split CP는 차이가 있습니다. K=1일 때의 CC는 전체 데이터셋이 곧 calibration set이 되고 훈련데이터셋은 존재하지 않는 것이고, 반면 split CP는 전체 데이터셋을 train set과 calibration set으로 나누고, train set으로 모델 훈련하고 calibration set으로 score를 계산하여 이를 test point의 score랑 비교해서 예측 구간을 만듭니다. 그리고 CC는 모든 fold가 calibration set 역할을 한번씩 수행합니다.

질의 응답

- A. K=n일 때의 CC는 모델을 총 n번 적합하고, 새로운 y에 대해서는 score만 계산하면 되지만, full CP는 새로운 y에 대해서 계속 모델을 다시 적합해야하므로 보통 K=n일 때의 계산량이 더 적습니다.
- A. K=n일 때의 CC는 n-1개의 데이터를 이용해 모델을 적합하고, full CP는 n+1개의 데이터를 이용해 모델을 적합하므로 통계적 효율성은 비슷하다고 생각합니다.

질의 응답

- Q. Thm 6.9와 6.10을 보면 jackknife+을 쓰면 training-conditional 보장이 다른 방법에 비해 안좋은 것 같은데, jackknife+를 쓰는 이유가 무엇인가요? jackknife+의 장점이 궁금합니다.
- A. 훈련할 때 CV+보다 많은 데이터(n-1개)를 쓰니, 통계적 효율성이 CV+보다 좋을 수 있습니다.
- Q. Jackknife/CV 에서 interval은 score function이 absolute difference로 두어져 있는 것 같은데 해당하는 score가 다른 함수 ex. diff square 등이더라도 이러한 방법과 커버리지 보장이 적용될 수 있을까요?
- A. Jackknife/CV는 원래도 커버리지 보장이 안되었는데, Jackknife+/CV+의 경우 score함수가 바뀌면 커버리지 보장이 안될거라 생각합니다. Jackknife+/CV+의 marginal coverage가 1-2 α 수준에서 보장되는 이유는, CC 구간 안에 이들이 포함되기 때문인데 이를 증명하는 과정에서 절대값 함수임이 주요 단서로 쓰이기 때문에...만약 다른 score 함수를 쓰고 싶다면 CC를 쓰면 될 거 같습니다.