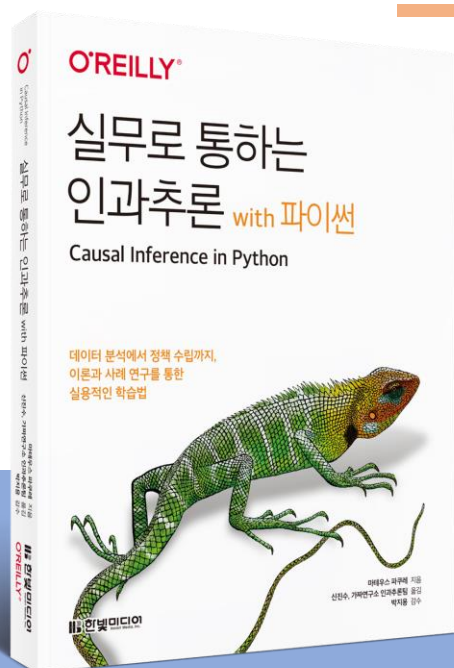


『실무로 통하는 인과추론 with 파이썬』 특강

머신러닝으로 인과추론



김준영 (Columbia University DBMI RA)

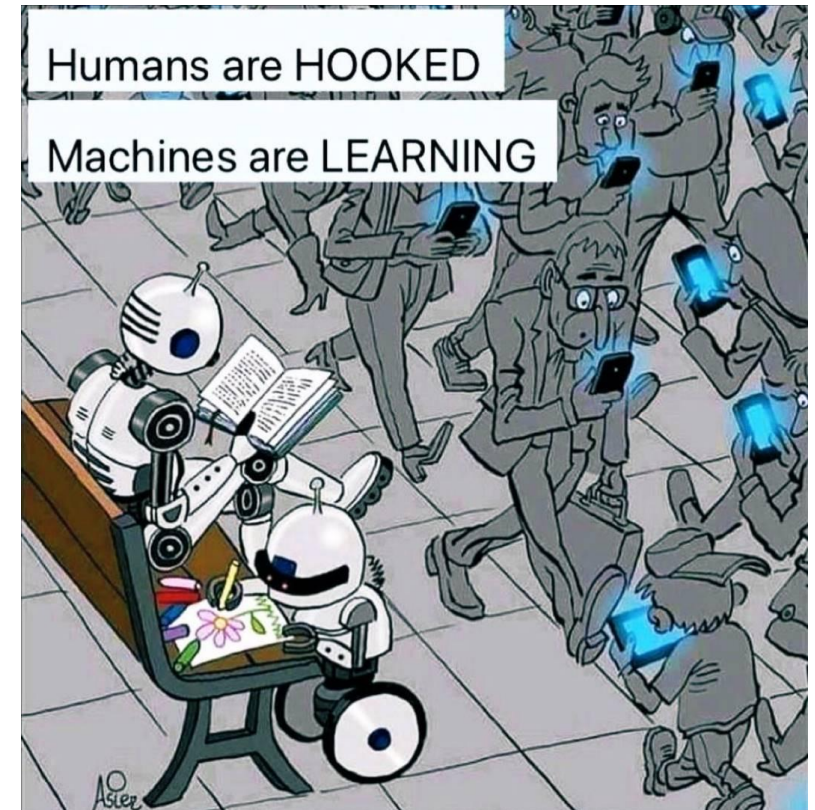
Phenotype-driven gene prioritization

Screening systematic review with NLP

통계학 석사 졸업

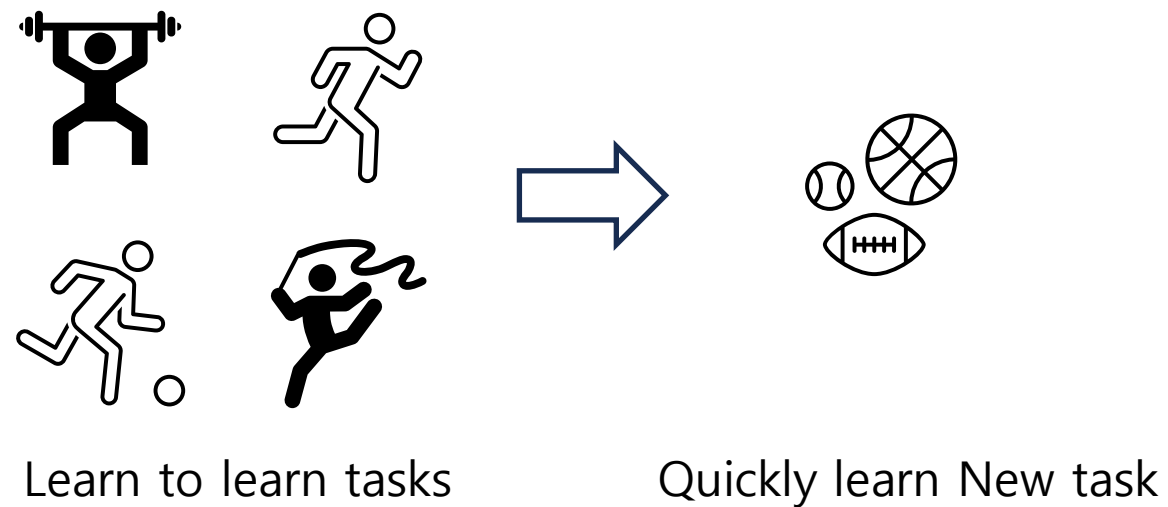
응용통계학 학사 졸업

머신러닝 / 농구



1. 메타러너
2. 실습
3. 요약

Meta Learner



지난 시간의 CATE...

$$\text{CATE} = E[Y_1 - Y_0 | X]$$

특정 조건에서 처치 (Treatment)가 결과에 미치는 평균적인 효과
(Treatment의 heterogeneity effect: 이질적 처치효과)

이를 구하는데 있어서 현실적인 문제들이 존재

1. Unconfoundness 조건과 positivity 가정 성립 전제
2. 분산이 증가하는 경우도 많음

따라서 CATE를 추정하기 위한 도구가 필요

1. 메타러너

메타러너는

다양한 머신러닝 모델과 기법을 활용하여 CATE를 추정하기 위한 전략적인 방법론들

다양한? 방법들?

⇒ 언제나 가장 좋은 머신러닝 모델은 존재하지 않기에 상황에 맞게 적용할 필요

여러 단계로 쪼개서 단계별로 추정하는 방법

- Missing value imputation
- Non-parametric HTE estimation (flexible한 장점)

보편적인 알고리즘 STEP

1. Base learner를 통해 통제 집단과 처리 집단의 조건부 기댓값을 계산한다 (COM)
2. 통제 집단과 처리 집단의 조건부 기댓값의 차이를 계산한다 (CATE)

특징: Base 모델이 여러 형태를 가질 수 있음 (회귀, 트리, 가우시안 프로세스 등등 일반 ML모델)

1-1 S-learner

S-learner는

하나의 모델로 처치와 공변량을 동시에 input으로 사용

- 하나의 모델을 쓰기 때문에 가장 간단함
- 처치 변수가 이산형/연속형 모두 사용 가능
- 처치 변수를 feature에 추가해 결과 예측
- Single Learner

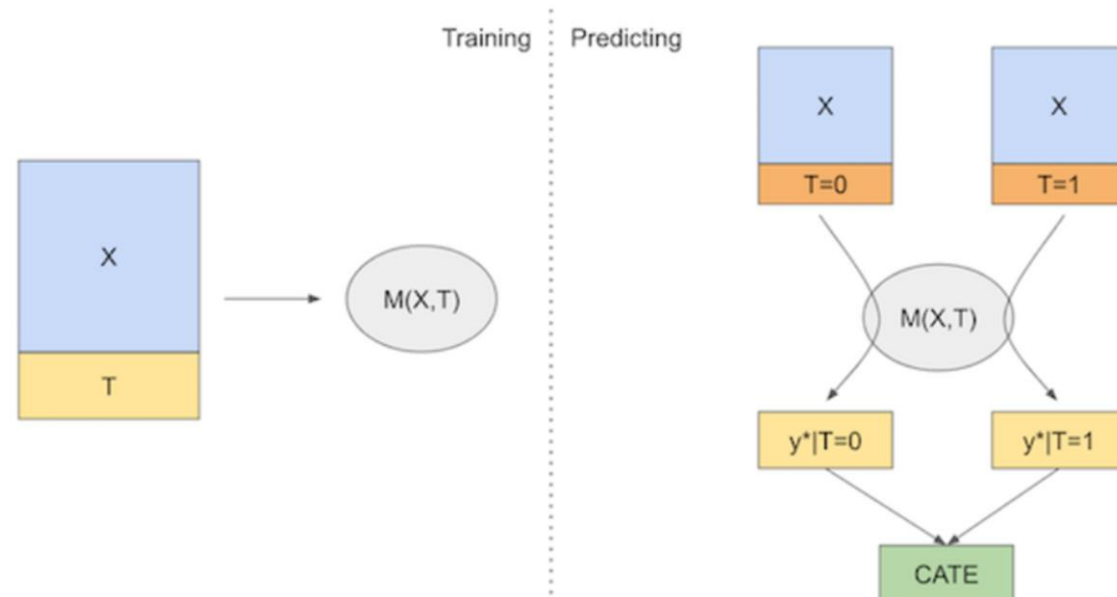
모든 predictor들을 사용

=> single estimator를 base learner로 사용할 수 있음

Algorithm SI 2 S-learner

- 1: **procedure** S-LEARNER(X, Y, W)
- 2: $\hat{\mu} = M(Y \sim (X, W))$
- 3: $\hat{\tau}(x) = \hat{\mu}(x, 1) - \hat{\mu}(x, 0)$

$M(Y \sim (X, W))$ is the notation for estimating $(x, w) \mapsto \mathbb{E}[Y|X = x, W = w]$ while treating W as a 0,1-valued feature.



Covariate 영향이 처치 영향보다 강하면, 처리 효과를 0으로 하는 경향을 지님

⇒ 처치 변수를 삭제 할 가능성

⇒ Regularization이 클 수록 문제가 더 커짐

1-2 T-learner

T-learner는

처치 효과를 버리는 S-learner의 문제를 해결하기 위한 방법으로
각 처치에 따른 counterfactual prediction 수행

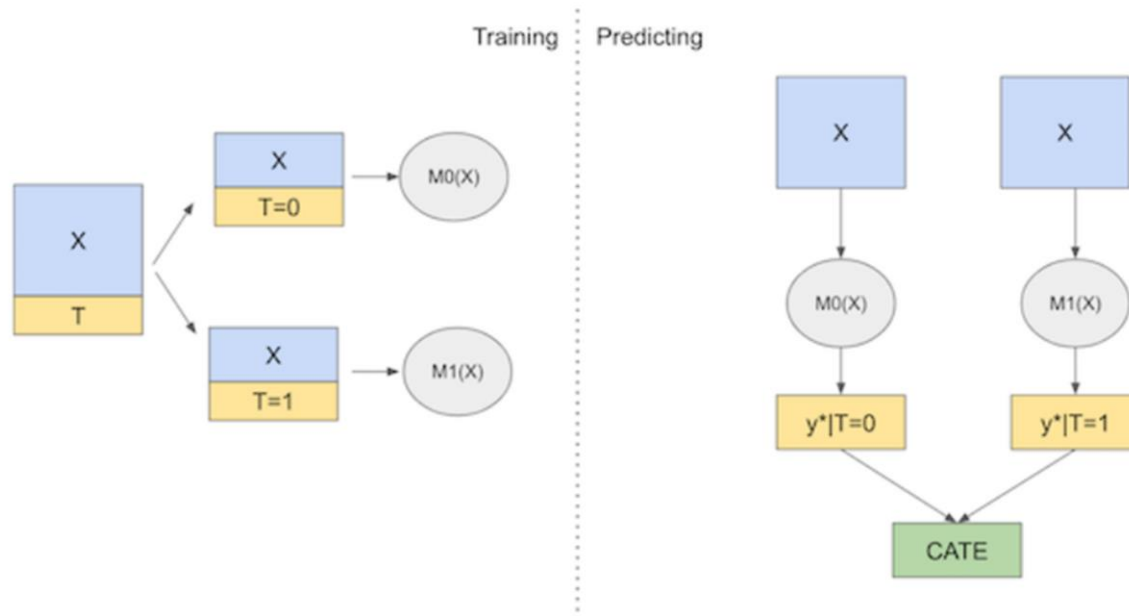
- Shared structure -> 처치에 따라 다른 모델을 사용
- 학습을 처음부터 쪼개서 함 -> 약한 처치 효과 변수 삭제 문제 극복
- Selection bias가 크기 때문에 간단한 모델일 수록 용이
- Two Learners

Algorithm 1 T-learner

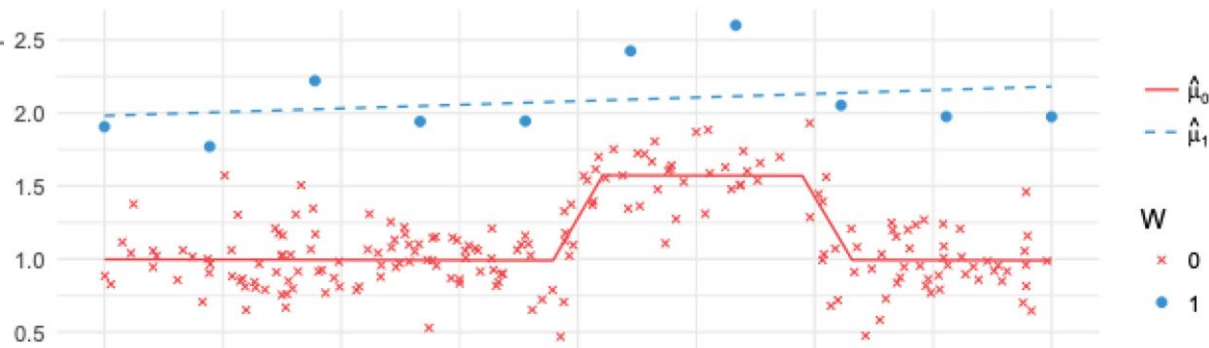
```
1: procedure T-LEARNER( $X, Y, W$ )  
2:    $\hat{\mu}_0 = M_0(Y^0 \sim X^0)$   
3:    $\hat{\mu}_1 = M_1(Y^1 \sim X^1)$   
  
4:    $\hat{\tau}(x) = \hat{\mu}_1(x) - \hat{\mu}_0(x)$ 
```

M_0 and M_1 are here some, possibly different, machine-learning/regression algorithms.

Regularization에 의한 편향 문제는 여전히 존재
(비선형 모델인데 처치 효과는 상수?)
데이터가 불균형 할 경우 모델 성능 저하



(a) Observed Outcome & First Stage Base Learners



1-3 X-learner

X-learner는

T-learner의 upgrade 버전으로

실험 설계가 불균형하거나, 더 복잡한 경우 T-learner 보다 좋음

처치 효과로 모델을 한번 더 돌린 후 가중 평균 추정

- 데이터 불균형에 Robust
- 개별 효과 추정에 집중 -> individual 효과 추정에 적합

Algorithm 3 X-learner

1: **procedure** X-LEARNER(X, Y, W, g)

2: $\hat{\mu}_0 = M_1(Y^0 \sim X^0)$

3: $\hat{\mu}_1 = M_2(Y^1 \sim X^1)$

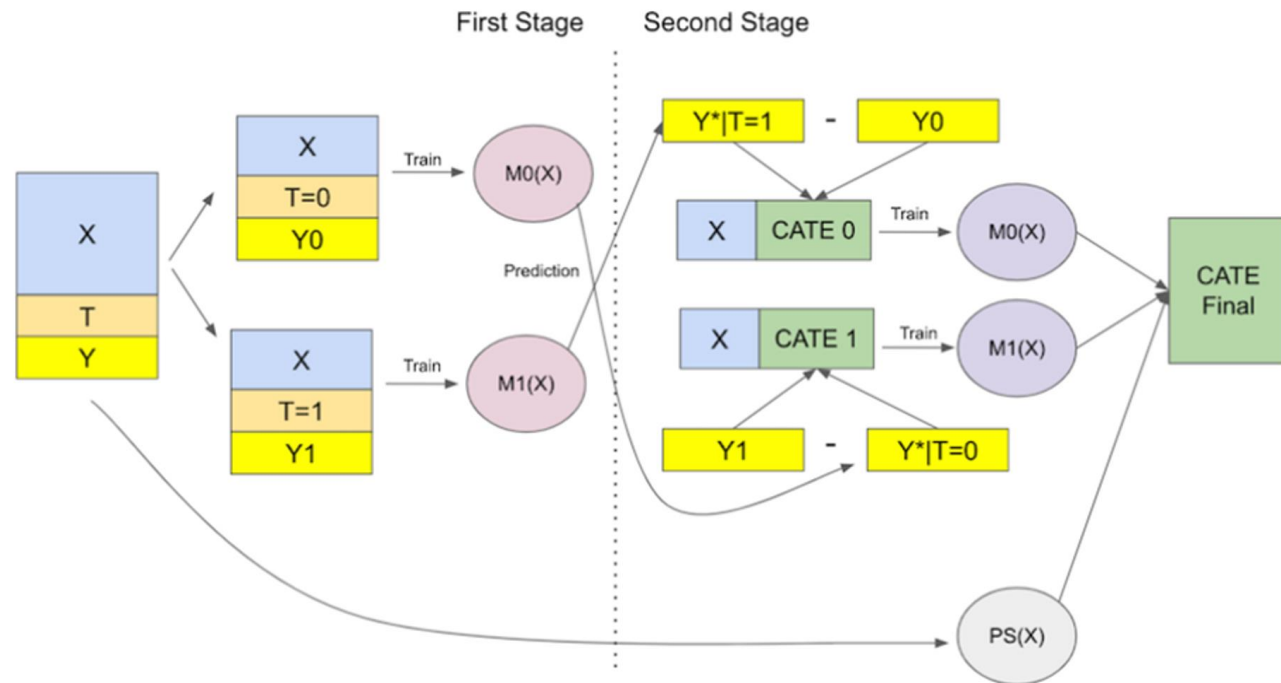
4: $\tilde{D}_i^1 = Y_i^1 - \hat{\mu}_0(X_i^1)$

5: $\tilde{D}_i^0 = \hat{\mu}_1(X_i^0) - Y_i^0$

6: $\hat{\tau}_1 = M_3(\tilde{D}^1 \sim X^1)$

7: $\hat{\tau}_0 = M_4(\tilde{D}^0 \sim X^0)$

8: $\hat{\tau}(x) = g(x)\hat{\tau}_0(x) + (1 - g(x))\hat{\tau}_1(x)$



1-4 R-learner

R-learner는

잔차를 통해 결과 변수와 처리 변수 간 상관성 제거,
두 단계의 처리를 수행하여 CATE 추정

- 데이터 불균형에 Robust
- 비선형 관계에서 유리
- 연속형 처리에서도 사용 가능

Algorithm 4: R-learner

1. procedure R-LEARNER(X, Y, T, W)

2. $\hat{Y}(X) = M_Y(X)$ (Outcome model: Predict Y without T)

3. $\hat{T}(X) = M_T(X)$ (Propensity model: Predict T using X)

4. Residualization for outcome:

$$\tilde{Y} = Y - \hat{Y}(X)$$

5. Residualization for treatment:

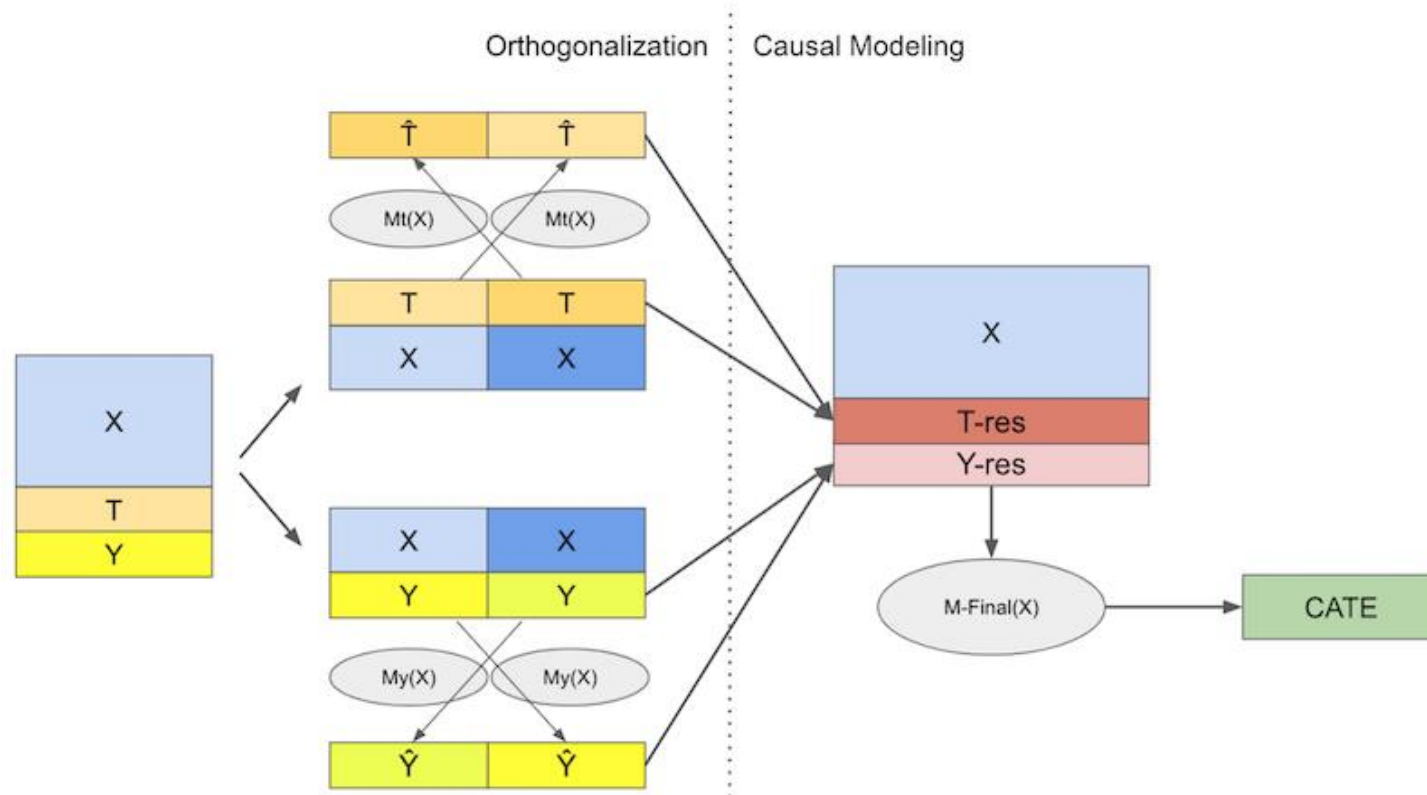
$$\tilde{T} = T - \hat{T}(X)$$

6. CATE estimation using residuals:

$$\hat{\tau}(X) = M_{\tau}(\tilde{T}, \tilde{Y}) \quad (\text{Learn treatment effect from residuals})$$

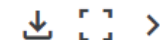
7. Final CATE estimate:

$$\hat{\tau}(X) = \frac{\tilde{Y}}{\tilde{T}} \quad (\text{Or with a flexible model based on residuals})$$



2-1 실습

german_credit_data.csv (49.69 kB)



Detail Compact Column

10 of 10 columns ▾

#	# Age	Δ Sex	# Job	Δ Housing	Δ Saving ac...	Δ Checking ...
0	67	male	2	own	NA	little
1	22	female	2	own	little	moderate
2	49	male	1	own	little	NA
3	45	male	2	free	little	little
4	53	male	2	free	little	little

이산형: T-learner / X-learner
성별에 따른 대출 승인 여부의 차이가 있을지?

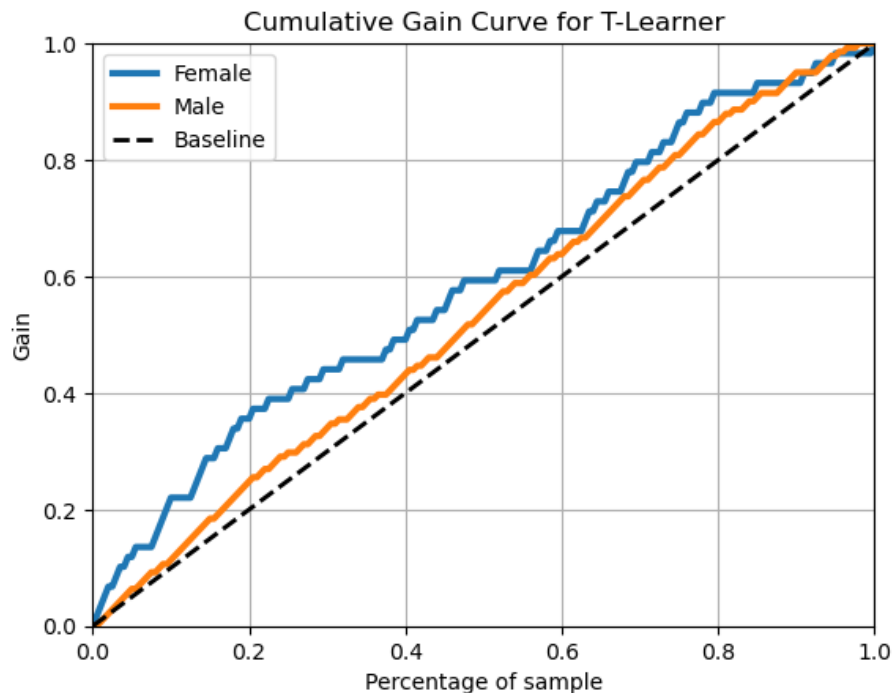
Lending Club

Lending Club is a peer-to-peer Lending company based in the US. They match people looking to invest money with people looking to borrow money. When investors invest their money through Lending Club, this money is passed onto borrowers, and when borrowers pay their loans back, the capital plus the interest passes on back to the investors. It is a win for everybody as they can get typically lower loan rates and higher investor returns.

The Lending Club dataset contains complete loan data for all loans issued through the 2007-2015, including the current loan status (Current, Late, Fully Paid, etc.) and latest payment information. Features (aka variables) include credit scores, number of finance inquiries, address including zip codes and state, and collections among others. Collections indicates whether the customer has missed one or more payments and the team is trying to recover their money. The file is a matrix of about 890 thousand observations and 75 variables.

연속형: S-learner / R-learner
할부금에 따른 대출 승인 여부의 차이가 있을지?

2-2 실습: 성별에 따른 차이

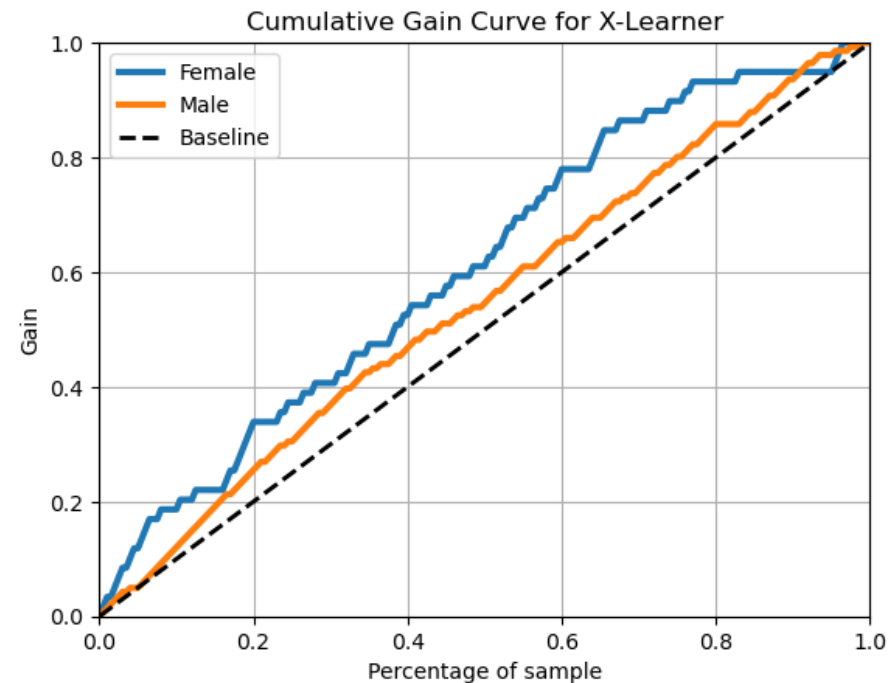


```
model_control = LogisticRegression(random_state=42)
model_treated = LogisticRegression(random_state=42)

model_control.fit(X_train[T_train == 0], y_train[T_train == 0])
model_treated.fit(X_train[T_train == 1], y_train[T_train == 1])

p0 = model_control.predict_proba(X_test)[: , 1]
p1 = model_treated.predict_proba(X_test)[: , 1]

cate_t_learner = p1 - p0
```



```
model_control = LogisticRegression(random_state=42)
model_treated = LogisticRegression(random_state=42)
model_control.fit(X_train[T_train == 0], y_train[T_train == 0])
model_treated.fit(X_train[T_train == 1], y_train[T_train == 1])

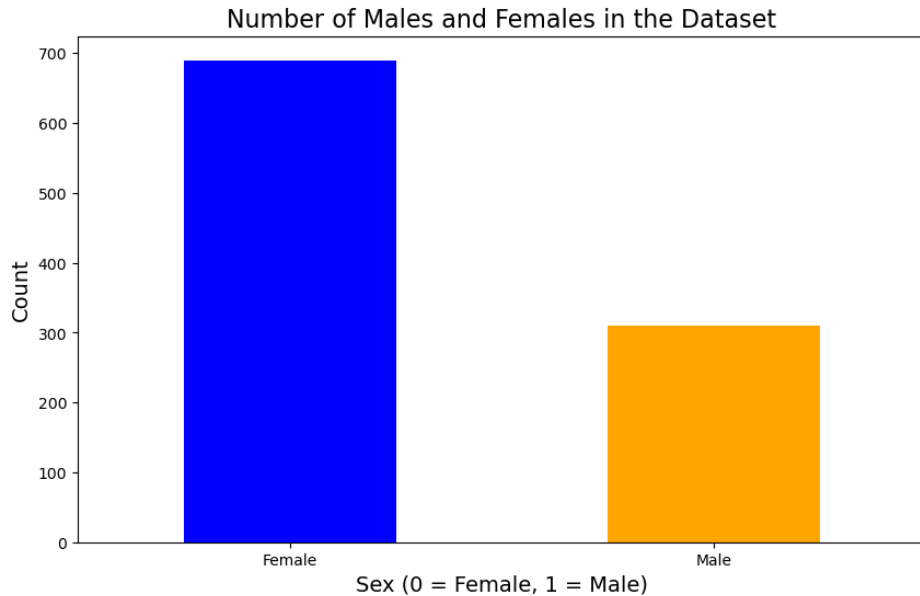
p1_imputed = model_treated.predict_proba(X_train[T_train == 0])[:, 1]
tau_control = p1_imputed - model_control.predict_proba(X_train[T_train == 0])[:, 1]
p0_imputed = model_control.predict_proba(X_train[T_train == 1])[:, 1]
tau_treated = model_treated.predict_proba(X_train[T_train == 1])[:, 1] - p0_imputed

weight_control = len(T_train[T_train == 0]) / len(T_train)
weight_treated = len(T_train[T_train == 1]) / len(T_train)

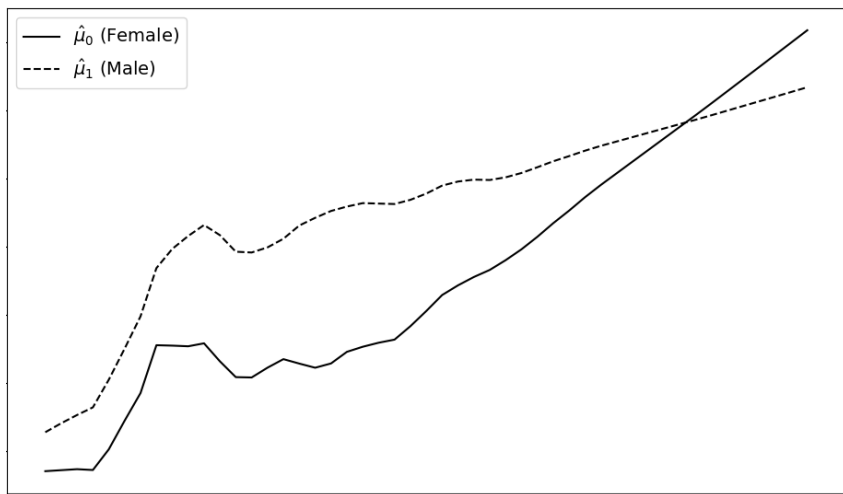
cate_x_learner = np.zeros(len(X_test))
cate_x_learner[T_test == 0] = tau_control.mean() * weight_control
cate_x_learner[T_test == 1] = tau_treated.mean() * weight_treated

p0 = model_control.predict_proba(X_test)[: , 1]
p1 = model_treated.predict_proba(X_test)[: , 1]
```

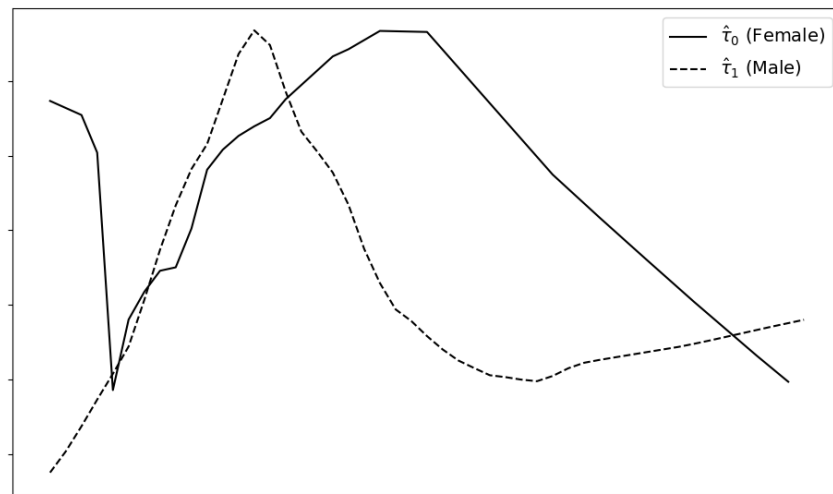
2-2 실습: 성별에 따른 차이



< T-learner >

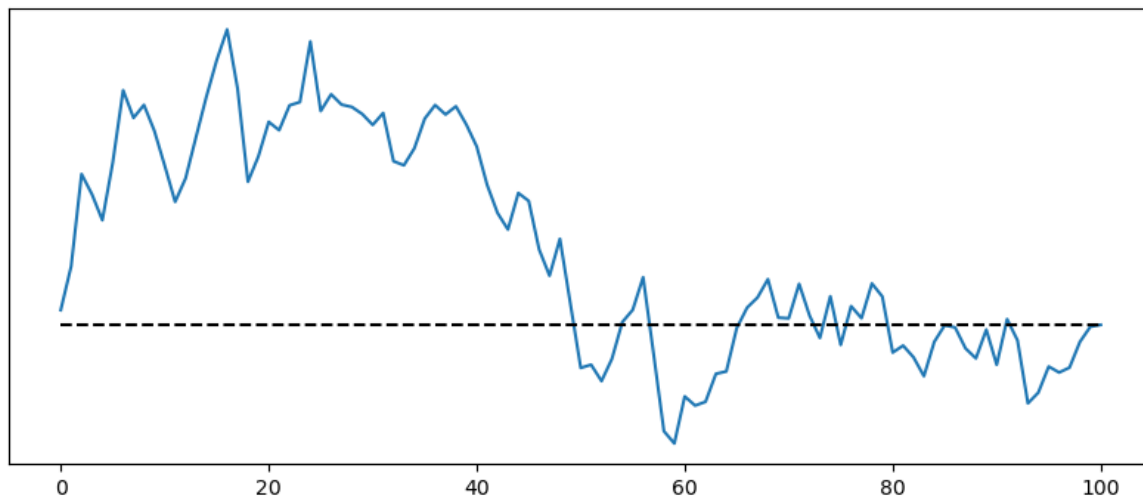


< X-learner >

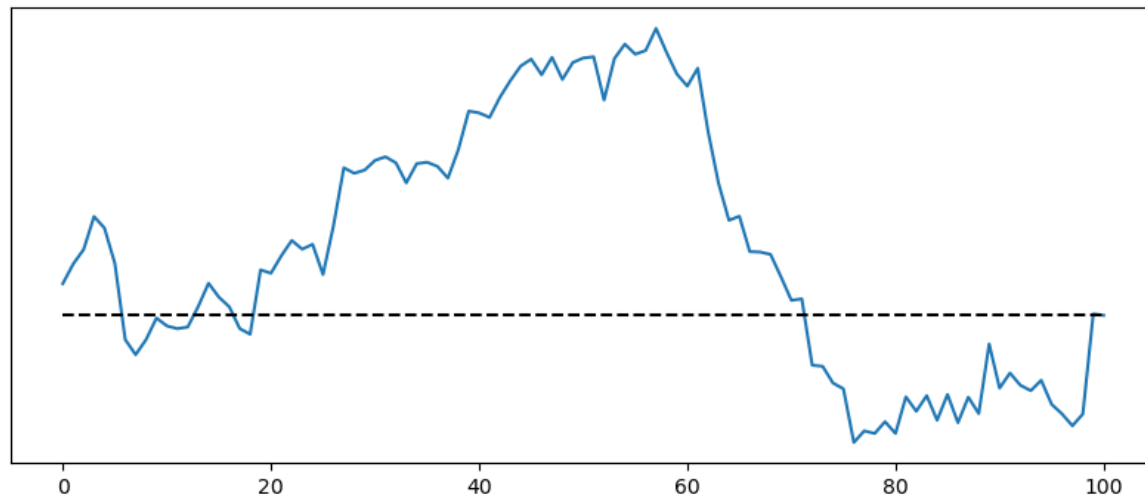


2-3 실습: 할부금에 따른 차이

S-Learner



R-Learner



```
s_learner_model = LogisticRegression(random_state=42)
s_learner_model.fit(X_train_with_treatment, y_train)

X_test_with_treatment[:, -1] = 1
p1 = s_learner_model.predict_proba(X_test_with_treatment)[:, 1]

X_test_with_treatment[:, -1] = 0
p0 = s_learner_model.predict_proba(X_test_with_treatment)[:, 1]

cate_s_learner = p1 - p0
```

```
denoise_m = LogisticRegression()
y_res = y_train - cross_val_predict(denoise_m, X_train_scaled, y_train, cv=5)

debias_m = LinearRegression()
t_res = T_train - cross_val_predict(debias_m, X_train_scaled, T_train, cv=5)

y_star = y_res / t_res
w = t_res ** 2

cate_model = LGBMRegressor()
cate_model.fit(X_train_scaled, y_star, sample_weight=w)





test_r_learner_pred = X_test.copy()
test_r_learner_pred['cate'] = cate_model.predict(X_test_scaled)

test_s_learner_pred = X_test.copy()
test_s_learner_pred['cate'] = cate_model.predict(X_test_scaled)
```

* Fairness analysis를 위해 전처리 과정에서 데이터를 많이 단순화 시켰습니다.

3 요약





S-learner

Feature X	Treatment T	Y(1)	Y(0)
	T=1	●	
	T=1	●	
	T=0		●
	T=0		●

Step 1. Learn $\mu(X, T) = E[Y|X, T]$

Step 2. $\tau = \mu(X, T = 1) - \mu(X, T = 0)$

T-learner





Feature X	Treatment T	Y(1)	Y(0)
	T=1	●	
	T=1	●	
	T=0		●
	T=0		●

Step 1. Learn $\mu_1(X) = E[Y(1)|X]$

Step 1. Learn $\mu_0(X) = E[Y(0)|X]$

Step 2. $\tau = \mu_1(X) - \mu_0(X)$

X-learner

Feature X	Treatment T	Y(1)	Y(0)
	T=1	●	
	T=1	●	
	T=0		●
	T=0		●

Step 1. $\mu_1(X) = \text{Learn } E[Y(1)|X]$

Step 1. $\mu_0(X) = \text{Learn } E[Y(0)|X]$

Y(1)	Y(0)	τ
●	○	● - ○
●	○	● - ○
○	●	○ - ●
○	●	○ - ●

Estimate

Step 2. Learn $\tau_1(X) = E[Y(1) - \mu_0(X)|X]$

Step 2. Learn $\tau_0(X) = E[\mu_1(X) - Y(0)|X]$

Step 3. $\tau = e(x)\tau_0 + (1 - e(x))\tau_1$

1. 처치 효과가 큰 경우 S-learner로 충분, 그렇지만 편향성을 지니는 단점 \Rightarrow 처치 변수 삭제 가능성
2. 처치 효과에 따라 모델을 만드는 T-learner는 샘플이 충분히 크다면 좋은 결과를 줌, 그렇지만 샘플이 작을 경우 overfit가능성 높음
3. X-learner = 2개 stage + propensity score
4. T-learner와 X-learner는 이산형 처치 변수에서만 사용 가능
5. R-learner는 잔차화를 통해 공변량 영향을 제거한 후 CATE 추정 (연속형 처치 변수도 사용 가능)

감사합니다

Reference

- 인과추론의 데이터 과학: <https://www.youtube.com/@causaldata science>
- Causal Inference for The Brave and True: <https://matheusfacure.github.io/python-causality-handbook/landing-page.html>