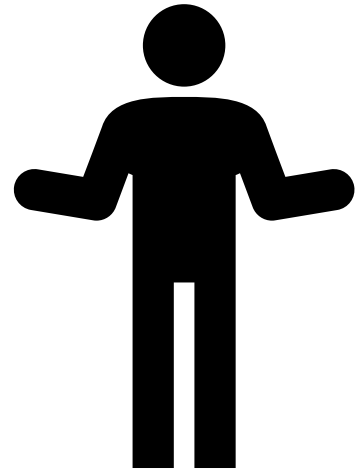
Causal prediction for medical decision making: Methods and practice

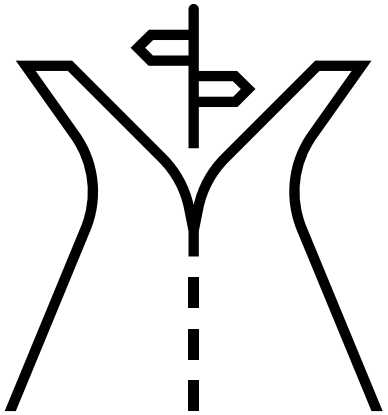
Estimation for causal prediction

Karla Diaz-Ordaz,

Deparment of Statistical Science

University College London





[Day 3, morning]

Recap prediction “under intervention” models

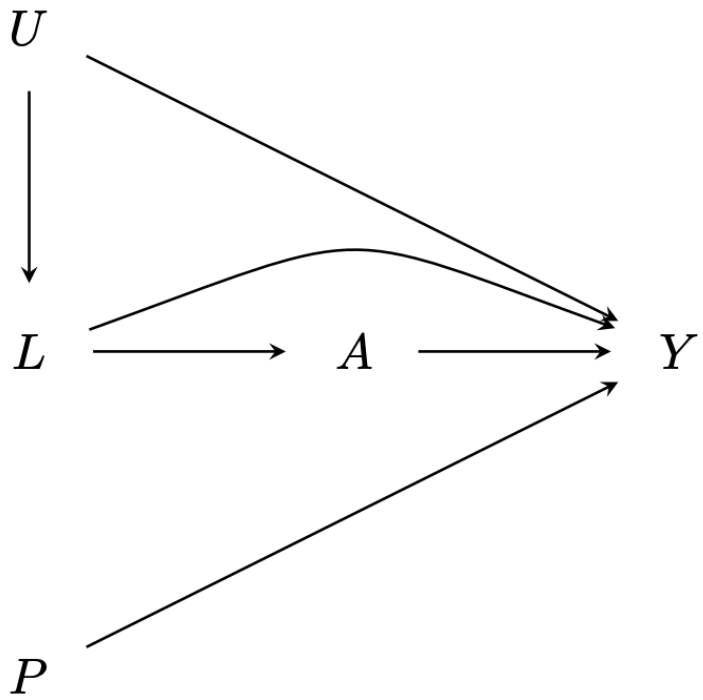
- Counterfactual predictions are risk estimates under possible (hypothetical) treatments
- if the training data are observational, we must do something to control for the confounding in the $A - Y$ relationship.
- We assume that L is a sufficient adjustment set, i.e.:

$$Y^1 \perp A|L$$

- The estimand of interest (prediction under a hypothetical intervention a) is a function of patients’ characteristics X^* (predictive of the outcome, can belong L or P

$$\mu_1(X^*) = E[Y^1|X^*]$$

- We distinguish the variables we want to condition on for the prediction estimand X^* and variables we need to incorporate to control for confounding L.



Confounding-adjustment and predictor sets

- Recall, $\mu_1(X^*) = E[Y^1|X^*]$ and $Y^1 \perp A|L$
- They are three distinct settings
  a. $L \subseteq X^*$: the adjustment set L is equal or contained in the set of predictors X^* OR
  b. there are variables we need to adjust for, which we don't want in our prediction models, i.e. $L \not\subset X^*$, equiv. there are variables L not in X^* (i.e. $L \backslash X^*$ non-empty)
     - because clinically actionable (clinically meaningful)
     - or fairness concerns (e.g. do not want a causal prediction model that varies by ethnicity)
  c. *Restricted covariate availability at runtime* only the subset X^* is available at *runtime* (e.g pragmatic to only input a few variables, $L \backslash X^*$ is expensive to collect): if we're not careful, it leads to (*runtime*) confounding

runtime : time of model deployment, when new (causal) predictions are obtained in the population of interest

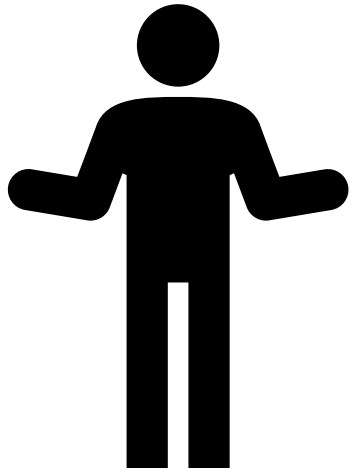
Constructing estimators for $E[Y^1|X^*]$

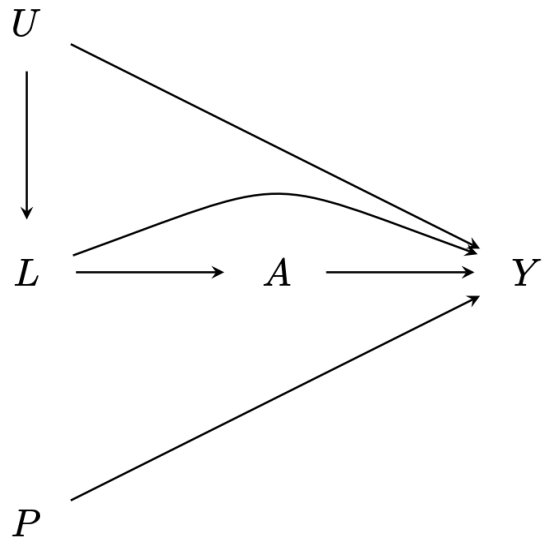
- we want to develop a good prediction model for $E[Y^1|X^*]$
- in other words, we want an estimator $\mu_1(X^*)$ which is unbiased

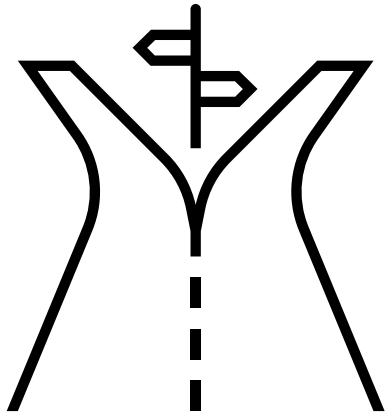
Challenges:

- there is confounding
- we need to estimate $E[Y^1|X^*]$ without bias – even when there is covariate "restrictions" at deployment (runtime confounding)
- model training can only be done on the ***observed treated,*** yet predictions must be good for the whole population (out-of sample)

Setting a: Developing a prediction model for $E[Y^1|X^*]$ when $L \subseteq X^*$







Estimators when $L \subseteq X^*$

- Under the identifying assumptions, if $Y^a \perp A|L$, we can write

$$\mu_1(X^*) = E[Y^1|X^*] = E[Y|A = 1, L, X^* \backslash L]$$

  where I wrote $X^* = L \cup X^* \backslash L$ explicitly. Recall $X^* \backslash L \subset P$

- This suggest the following estimation strategy

Outcome regression

1. Develop a model for the outcome dependent on L, and all other variables in X^* using the treated

$$Q_1(X^*) = E[Y|A = 1, L, X^* \backslash L]$$

   e.g for continuous outcome Y: $Q_1(X^*) = \beta_0 + \beta_1 L_1 + \beta_2 L_2 + \cdots + \beta_p P_1 + \cdots$

   - Assumes model is correct
   - extrapolation is an issue

Dickerman, Dahabreh, Cantos, et al. Predicting counterfactual risks under hypothetical treatment strategies: an application to HIV. *European Journal of Epidemiology* 2022. https://doi.org/10.1007/s10654-022-00855-8

Example: outcome regression

- Consider the CVD risk prediction tool for type 1 diabetes, hypothetical statins intervention
- Y =LDL cholesterol at the post-treatment
- predictors X^*= sex, age, and baseline LDL
- sufficient adjustment set: Suppose $L =$ baseline LDL

outcome regression

1. Develop a model for the outcome on the treated

$$Q_1(X^*) = \beta_0 + \beta_1 age + \beta_2 sex + \beta_3\ ldl\_base | A = 1$$

- Under the assumption that

$$ldl\_post^a \perp Obs\_Statin |\ ldl\_base$$

this is a valid causal prediction model

2. use this to predict the desired conditional potential outcomes for all at deployment

Example continued

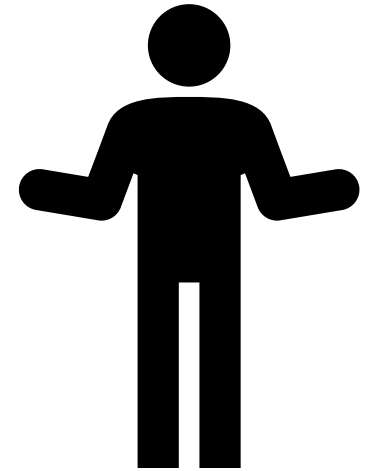
- Now, suppose we learn that the level of physical activity ($motion$) is a confounder, and therefore must be included in the adjustment set, i.e. $L =$ baseline LDL and motion
- we do not want to include "$motion$" in the final causal prediction model, which only depends on baseline LDL, age and sex

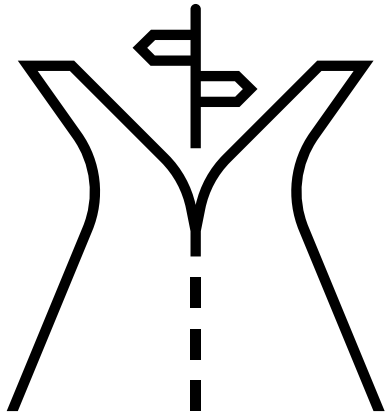
Now we are in that setting where $L \not\subset X^*$, as $motion$ is in L but not in X^*

We need to modify our outcome modelling approach:

G-computation

Setting b: Developing a prediction model for $E[Y^1|X^*]$ when $L \not\subset X^*$





G-computation when $L \not\subset X^*$

- If $Y^a \perp A|L$ holds, but L not all in X^* (ie, there are vars in $L\backslash X^*$), we have
$$E[Y^1|X^*] = E[E[Y|A=1, X^*, L\backslash X^*]|X^*]$$
- if we have access to all $L\backslash X^*$ at deployment, we can use the following strategy

G-computation

1. Model the outcome dependent on confounders and predictors of interest $Q_1(L, X^*) = E[Y|A=1, L\backslash X^*, X^*]$
2. use this to predict the conditional potential outcomes for all at deployment, $\widehat{Q_1}(X^* = x, L\backslash X^* = l)$
3. Marginalise over $L\backslash X^*$ (integrated the unwanted $L\backslash X^*$ out).

- The causal prediction of interest (for simplicity, assuming $L\backslash X^*$ is discrete)
$$\widehat{\mu_1}(X^*) = \sum_l \widehat{Q_1}(X^* = x, L\backslash X^* = l)\widehat{\Pr}(L\backslash X^* = l)$$
- In general, (i.e. if $L\backslash X^*$ contains continuous variables), marginalizing over $L\backslash X^*$ would be difficult, and numerical techniques may be necessary

Example continued

- $Y =$ LDL cholesterol at the post−treatment as the outcome
- predictors X^*= sex, age, baseline LDL
- $L =$ baseline LDL and motion
- $L \not\subset X^*$, as $motion$ is in L but not in X^*: $L \backslash X^* = motion$

G-computation

1. Develop a model for the outcome on the treated

$$Q_1(L, X^*) = \beta_0 + \beta_1 age + \beta_2 sex + \beta_3\, ldl_{base} + \beta_4 motion | A = 1$$

2. use $\widehat{Q_1}(L \backslash X^*, X^*)$ to predict the expected conditional potential outcomes for deployment set, conditional on levels of $motion$
3. average these predictions over $L \backslash X^* = motion$. As $motion$ is binary, this is:

$$\widehat{\mu_1}(X^*) = \widehat{Q_1}(X^*, motion = 1)\frac{\#(motion=1)}{n} + \widehat{Q_1}(X^*, motion = 0)\frac{\#(motion=0)}{n}$$

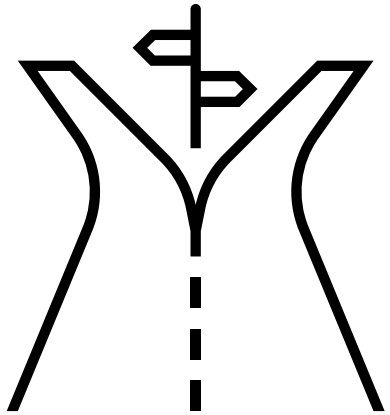
Challenges when L only measured in the training set

- G-computation requires all the variables in the adjustment set are available at time of deployment
- Suppose only a subset $Z \subset L$ is available at runtime,

$$E[Y \mid A = 1, Z] \quad \text{is not } E[Y^1 | Z]$$

- so, a G-computation $Q_1(Z, X^*) = E_z \{E[Y|A = 1, Z, X^*]|X^*\}$ does not target the right counterfactual quantity
- *runtime confounding:* if we don't have access to all L at runtime (deployment), and we can't do an unconfounded G-computation
- The amount of bias will depend on how much residual confounding there is (after adjusting only for Z)

Setting c: Developing a causal prediction model for $E[Y^1|X^*]$ when $L \not\subset X^*$ **only** measured in the training set



Plug-in estimator $L \not\subset X^*$ only measured in train set

- Recall $E[E[Y|A=1, X^*, L\backslash X^*]|X^*]$, this motivates the following strategy
  1. Model the outcome dependent on confounders and predictors of interest $Q_1(L, X^*) = E[Y|A=1, L\backslash X^*, X^*]$
  2. use this to predict $\widehat{Q_1}(L\backslash X^*, X^*)$ still for the training data
  3. Run $m_1^{PL}(X^*)$ a second-stage model (the outer expectation) with $\widehat{Q_1}(L\backslash X^*, X^*)$ as the dependent variable, on X^*
  4. use the trained $\widehat{m}_1^{PL}(X^*)$ this to predict the desired conditional potential outcomes $\widehat{\mu_1}(X_i^*)$ for all at deployment

  This strategy needs an extra model $m_1^{PL}(X^*)$ (assumed to be correct)
- This can also be applied when $L \not\subset X^*$ is available at runtime.
- **Technical:** We use different splits of the train data to learn $Q_1(L\backslash X^*, X^*)$ and $m_1^{PL}(X^*)$ to avoid potential overfitting

Coston, A., Kennedy, E. H., & Chouldechova, A. (2020). Counterfactual Predictions under Runtime Confounding. In Advances in Neural Information Processing Systems (Vol. 33, pp. 4150–4162) (Algorithm 2)

Example continued

- $Y =$ LDL cholesterol at the post−treatment as the outcome
- predictors X^*= sex, age, baseline LDL
- $L =$ baseline LDL and $motion \notin X^*$,

Plug-in G-computation

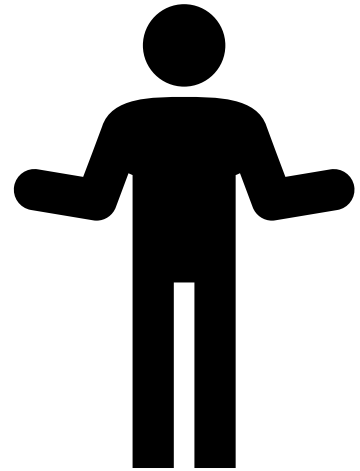
1. Develop a model for the outcome on the treated
   $$Q_1(L, X^*) = \beta_0 + \beta_1 age + \beta_2 sex + \beta_3\, ldl_{base} + \beta_4 motion | A = 1$$
2. use $\widehat{Q_1}(L, X^*)$ to predict the pseudo-outcomes for training set
3. Regress $m_1^{PL}(X^*) := \widehat{Q_1}(L, X^*) \sim X^*$ on the train set
4. using this 2nd-stage model $\hat{m}_1^{PL}(X^*)$, obtain causal predictions in the deployment

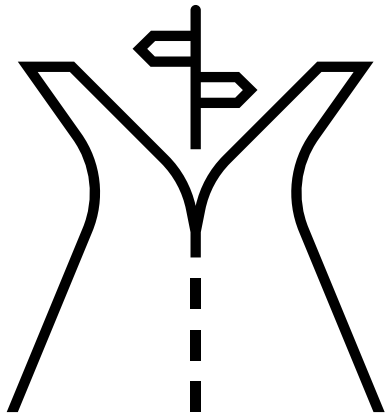
```
m1cond<-lm(LDL_fup~age+sex_male+LDL_base+motion,
           data=datatrain[datatrain$statin==1,])
pred.cond.y1<-predict(m1cond, newdata=datatrain)
stage2_mody1<-lm(pred.cond.y1~age+sex_male+LDL_base, data=datatrain)
pred.y1.plugin<-pred.y1<-predict(stage2_mody1, newdata=deployment)
```

- G-computation assumes the conditional outcome model is correctly specified and needs a marginalisation step requiring all the de-confounders be available at runtime
- The plug-in estimator, requires two models which need to be correctly specified
- **Example continued:** let's see how two different strategies compare, with a cond. outcome model with and without interactions with $motion$
- First two columns **G-comp**, last two **plug-in estimator**

y1.marg.motion	y1.marg.motion2	pred.y1	pred.y1.2
2.488437409	2.488032056	2.308159512	2.61069846
0.956757842	1.058869509	0.972578094	1.21641988
-0.889658032	-0.645756188	-1.017793501	-0.62230145
0.961395656	1.021194662	0.793766824	1.03946839
3.225442113	3.101144584	3.257955054	3.28501181
1.113290077	1.235906662	1.066240478	1.40111523
-0.657655985	-0.413190462	-0.810126582	-0.37498305
0.745415395	0.863156418	0.465490215	0.86102739
-0.733692058	-0.505436918	-0.841197269	-0.47023660

Developing a causal prediction model for $E[Y^1|X^*]$ based on IPW (all cases!)





IPW strategy to develop a $\mu_1(X^*)$

- If $Y^a \perp A|L$, regardless of whether $L \subseteq X^*$ or $L \not\subset X^*$, or where L is available, we can use an IPW identification strategy

$$\mu_1(X^*) = E[Y^1|X^*] = E\left[\frac{A}{Pr(A=1|L)}Y|X^*\right]$$

- This suggest the following

1. specify a model $\pi(L)$ for $Pr(A=1|L)$ (propensity score) in the train data
2. obtain weights $\frac{1}{\hat{\pi}(L)}$ in the training data
3. develop a weighted model $m_1^{ipw}(X^*)$ on the treated in the train data
4. Using the trained model $\hat{m}_1^{ipw}(X^*)$, obtain causal predictions $\widehat{\mu_1}(X_i^*)$ in the deployment

- assumes positivity $0 < P(A=1|L=l) < 1$, for all l.
- relies on the PS being correctly specified

Example continued — IPW

Y =LDL cholesterol at the post−treatment as the outcome

predictors X^*= sex, age, baseline LDL

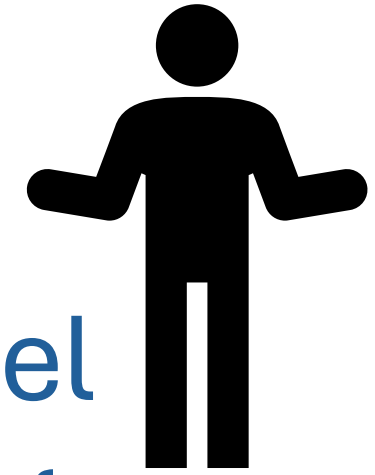
L = baseline LDL and motion

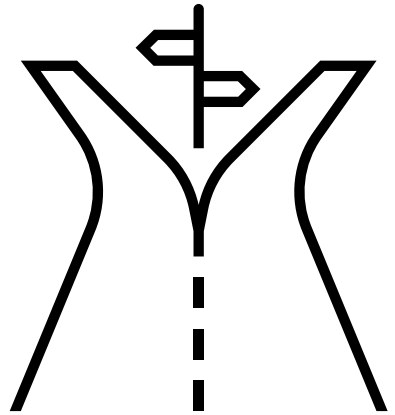
```
#IPW
ps.model<-glm(statin~ LDL_base+motion,
   data=datatrain, family=binomial())
ps<-predict(ps.model, type = "response",newdata=datatrain)
w<-1/ps

mody1.w<-lm(LDL_fup~age+sex_male+LDL_base, data=datatrain, weights = w)
print(summary(mody1.w))
pred.y1.ipw<-predict(mody1.w, newdata=deployment)
```

Technical: we could use stabilised IPW, to improve issues of large weights: stabilised weights work by multiplying the standard IPW weights by the overall (marginal) probability of the treatment.

Developing a causal prediction model for $E[Y^1|X^*]$ using machine learning





Constructing machine learning estimators

- we want a model $m_1(X^*)$ that can be used at deployment without access to all of L and doesn't suffer from runtime confounding bias
- Plug-in and IPW estimators
  - They assume either their models are correctly specified
- we would like to attenuate dependence on model misspecification
- Naïve use of data-adaptive estimation for these conditional expectations leads to **plug-in bias**

Debiased Machine Learning estimators

- ideally our estimator $m_1(X^*)$ must minimise (feasible) counterfactual mean square error

$$E[\{Y^1 - \mu_1(X^*)\}^2]$$

Challenges:

- $E[Y^1|X^*]$ may be an infinite-dimensional "parameter"
- want to use machine learning /data-adaptive estimation, without plug-in bias
- dealing with runtime confounding
- **Technical:** well-developed theory for de-biasing plug-in estimators (via efficient influence functions) only applies to pathwise differentiable parameters – we need the theory of "orthogonal loss functions"

Orthogonal loss functions: DR learner

- double robust (DR) m^{DR} estimator of the (empirical) counterfactual prediction error is that one that minimises

$$\frac{1}{n}\sum_{i=1}^{n}\left[\frac{A}{\pi(L)}\{Y - Q_1(L, X^*)\} + \{Q_1(L, X^*) - m_1{}^{DR}(X^*)\}\right]^2 + \Lambda(m_1{}^{DR})$$

  where $Q_1(L, X^*) = E[Y|A = 1, L, X^*]$ is the outcome model, $\pi(L)$ the PS, and $\Lambda(m_1{}^{DR})$ is a penalization term needed to avoid overly complex m^{DR}

- made feasible by estimating nuisance models $\hat{\pi}(L)$ and $\widehat{Q_1}(L, X)$
- using standard ML algorithms, we can find $\widehat{m_1}^{DR}(X^*)$ by "regressing"

  pseudo-outcome(AIPW) $\frac{A}{\hat{\pi}(L)}\{Y - \widehat{Q_1}(L, X^*)\} + \widehat{Q_1}(L, X^*)$ on X^*

- causal predictions $\widehat{\mu_1}(X_i^*)$ for i in deployment obtained using trained model $\widehat{m_1}^{DR}$
- **Technical:** needs cross-fitting— split the data, use one part to estimate nuisance models and another to run the regression $m_1{}^{DR}$ of the pseudo-outcome on predictors (then swap and aggregate)

DR learner

1. Initial step : Nuisance training, using one part of training data D_1
   a. Train 'propensity score' estimates $\widehat{\pi}(L)$
   b. learn the conditional outcome model $\widehat{Q_1}(L, X^*)$ in the treated
2. Using the other part of the training data, D_2 construct the pseudo-outcome $\psi = \frac{A}{\widehat{\pi}(L)}\{Y - \widehat{Q_1}(L, X^*)\} + \widehat{Q_1}(L, X^*)$
3. On D_2, "regress" ψ on the predictors of interest X^* using an ML algorithm m_1 of choice to obtain the trained model $\widehat{m_1}^1(X^*)$
4. Cross-fitting (reverse roles of D_1 and D_2), and get the trained model $\widehat{m_1}^2$
5. Get fitted values $\widehat{m_1}^v({X_i}^*)$ on every i person in the deployment set for each trained model in fold v. The final prediction is the average of these

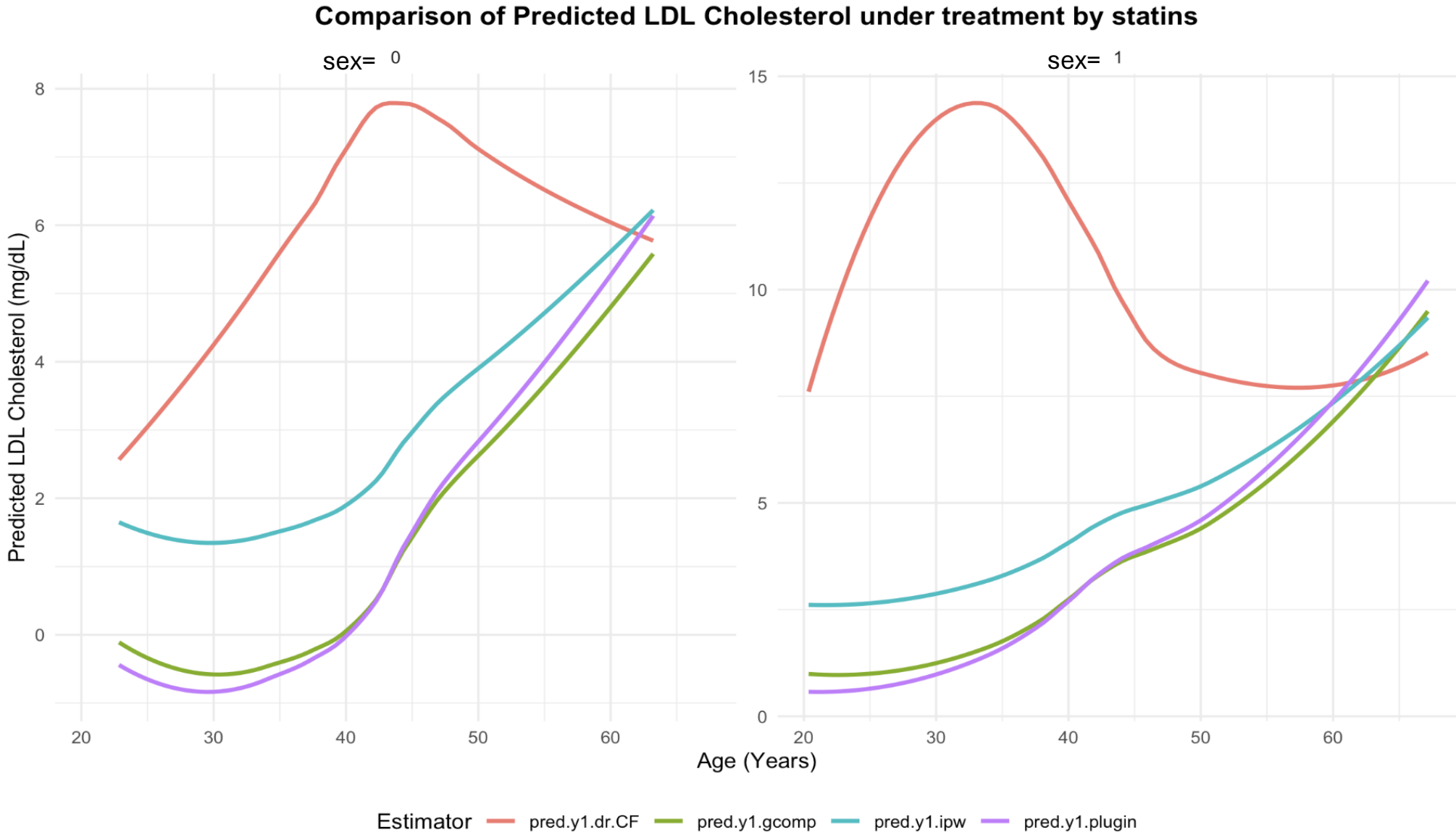
$$\widehat{\mu_1}^{DR}({X_i}^*) = \frac{1}{\#folds} \sum_{v-folds} \widehat{m_1}^v({X_i}^*)$$

Note: It can result on predictions outside of the natural range of the outcome due to weights used in the pseudo-outcome!

Example continued — DR-learner code

```
### DR learner
folds=2
m.model<-list(folds)
#Cross-fitting index
N<-dim(datatrain)[1]
split <- floor(N / folds)
if (N %% folds != 0) {
  s <- c(rep(1:folds, split), 1:(N - split * folds))
}else{s <- c(rep(1:folds, split))}
for (k  in 1:folds){
  dt_k<-datatrain[s!=k,]
  dtk<-datatrain[s==k,]
  Q.model=ranger(LDL_fup~age+sex_male+LDL_base+motion, data=dt_k[dt_k$statin==1,])
  ps.model= ranger(statin~ LDL_base+motion, data=dt_k)
  Q=predict(Q.model, data= dtk, type = "response")$predictions
  P=predict(ps.model,data= dtk, type = "response")$predictions
  w=1/P
  dtk$ypseudo =  w*(dtk$LDL_fup-Q)+Q
  #pseudo-outcome regression
  m.model[[k]]<-ranger(ypseudo~ age+sex_male+LDL_base, data=dtk)
}
m1.f1<-predict(m.model[[1]], data= deployment, type = "response")$predictions
m1.f2<-predict(m.model[[2]], data= deployment, type = "response")$predictions
pred.y1.dr.CF<-(m1.f1+m1.f2)/2
```

Example continued — comparison



Uncertainty quantification

- We want prediction intervals for counterfactual predictions, **without** assuming the model is correct
- We can use **conformal inference:** uses a model's past experience to determine precise levels of confidence in new predictions
- **Technical:** *Standard CoI:* assuming exchangeability, the distribution of the residuals in the training

$$r_i = |Y_i - \hat{m}(Z_i)|$$

approximates the distribution of the residuals for the deployment population

- 95% prediction intervals are defined by the 95-th centile of the r distribution
- **Technical:** with **confounding, to regain exchangeability**
  - re-weight the residuals by IPW
  - or doubly-robust *Lei & Candès https://academic.oup.com/jrsssb/article/83/5/911/7056131*

Summary

- Prediction models under hypothetical interventions need to control for confounding
- in high-dimensional confounding settings, , “outcome regression” and G-computation can be problematic due to extrapolation
- in addition, is runtime confounding settings, “outcome regression” and G-computation cannot remove the confounding
- IPW is intuitive, easy to implement and works well in both high-dim adjustment set and runtime confounding
- Neither strategy is not robust to mis-specification of the models involved

Summary and final remarks

- using ML may result directly into plug-in G-comp or IPW leads to plug-in bias
- orthogonal-loss estimators are the soluion, ie DR learner
- DR-learner has limitations, the transformed pseudo -outcome may result on predictions outside the space
- Vansteelandt & Morzywolek have developed a so called “imputation” learner which avoid this (this is a targeted learning (in infinite dims) approach https://arxiv.org/abs/2311.09423)
- There is a close relation with causal inference methods for estimating conditional average treatment effects (CATE) — R-learner, Causal forests, DR-learner in longitudinal settings (https://arxiv.org/pdf/2306.16297)
- **But** for prediction under interventions we are interested in absolute risks, not just risk differences