6.S897/HST.956 Machine Learning for Healthcare

Lecture 15: Causal Inference Part II

Instructors: David Sontag, Peter Szolovits

1 Review

1.1 Potential Outcomes

- Each unit (individual) x_i has two potential outcomes:
  - $Y_0(x_i)$ is the potential outcome had the unit not been treated: "control outcome"
  - $Y_1(x_i)$ is the potential outcome had the unit been treated: "treated outcome"
- Conditional Average Treatment Effect (CATE) for unit x_i:

$$CATE(x_i) = \mathbb{E}_{Y_1 \sim p(Y_1|x_i)}[Y_1|x_i] - \mathbb{E}_{Y_0 \sim p(Y_0|x_i)}[Y_0|x_i]$$

- Average Treatment Effect (ATE):

$$ATE = \mathbb{E}_{x \sim p(x)}[CATE(x)]$$

1.2 Two common approaches to counterfactual inference

1. Covariate Adjustment
2. Propensity scores

These two are the result of doing reduction from causal inference to Machine Learning. We will see why this is the case from future sections.

2 Covariate Adjustment

- Explicitly model the relationship between treatment, confounders, and outcome.
  - Covariates/features are $x_1, ..., x_d$ and treatment T
  - Regression model $f(x, T)$ is a function of the covariates and the treatment
  - Output y of the regression model is the outcome
- Under ignorability:

$$CATE(x) = \mathbb{E}_{x \sim p(x)}[\mathbb{E}[Y_1|T = 1, x] - \mathbb{E}[Y_0|T = 0, x]]$$

- Fit a model $f(x, t) \approx \mathbb{E}[Y_t|T = t, x]$, then:

$$\widehat{CATE}(x_i) = f(x_i, 1) - f(x_i, 0)$$

2.1 Covariate Adjustment with Linear Models

Assume that

$$Y_t(x) = \beta x + \gamma t + \epsilon_t$$

and that

$$\mathbb{E}[\epsilon_t] = 0$$

where $Y_t(x)$ is blood pressure, x is age, and t is medication. We have that $Y_t(x)$ is a linear function of both x and t. We can now calculate CATE:

$$\begin{aligned} CATE(x) &= \mathbb{E}_{p(Y_1|x)}[Y_1] - \mathbb{E}_{p(Y_0|x)}[Y_0] \\ &= \mathbb{E}_{\epsilon_0,\epsilon_1}[\beta x + \gamma + \epsilon_1 - \beta x - \epsilon_0] \\ &= \gamma + \mathbb{E}[\epsilon_1] - \mathbb{E}[\epsilon_0] \\ &= \gamma \end{aligned}$$

We can also easily calculate ATE:

$$ATE = \mathbb{E}_{p(x)}[CATE(x)] = \gamma$$

and observe that ATE is equal to the coefficient of the linear model.

For causal inference, our goal is to estimate the coefficients (γ) well, not necessarily predict the output of the model ($Y_t(x)$) well. This is a key difference that is often highlighted in the different focuses of the machine learning and statistics communities. In machine learning, the goal is often to produce the best predictions possible. We will try to reduce the held-out error of our model. In the context of this problem, we would want to predict the factual outcome $Y_t(x)$. On the other hand, in statistics, the goal is *identification* of the parameter γ rather than *prediction*. The main goal is to understand the causal structure of the problem, rather than predicting the value of an outcome. When estimating linear models, we care about the linear coefficients and confidence intervals regarding these coefficients. Now we will explore what happens if the model is not, in fact, linear.

2.2 What happens if the true model is not linear?

Suppose our true data generating process is non linear. Specifically for $x \in \mathbb{R}$:

$$\begin{aligned} Y_t(x) &= \beta x + \gamma t + \delta x^2 \\ ATE &= \mathbb{E}[Y_1 - Y_0] = \gamma \end{aligned}$$

However, we hypothesize that the model is linear:

$$\hat{Y}_t(x) = \hat{\beta} x + \hat{\gamma} t$$

Now, our estimate for γ could be arbitrarily large or small, depending on the value of δ, as demonstrated in the following result which we show without derivation:

$$\hat{\gamma} = \gamma + \delta \frac{\mathbb{E}[xt]\mathbb{E}[x^2] - \mathbb{E}[t^2]\mathbb{E}[x^2 t]}{\mathbb{E}[xt]^2 - \mathbb{E}[x^2]\mathbb{E}[t^2]}$$

Our estimate could potentially extremely wrong. While this framework can be very useful for linear models, it is important to realize that it can be extremely wrong if the underlying model is not actually linear. The biostatistics community has attempted to add in nonlinearities (interaction terms between variables) by growing the model. They start with a linear model and slowly add in nonlinearities to see if it produces a better fit. In addition to this example, causality is using a range of nonlinear methods from machine learning:

- Random forests and Bayesian trees
- Gaussian processes
- Neural networks

2.3 Examples of Nonlinear Methods

2.3.1 Gaussian Processes

Gaussian processes allow us to examine the full distribution and compare confidence intervals between two treatments. Figure 1 shows two graphs, each of which contains points from treated and control populations, as well as regression curves and confidence intervals.

As an example, let us think of a naive way to determine whether a patient x should receive a given treatment. We can compute their CATE and then if $\widehat{CATE}(x) > \alpha$, give them the treatment. Otherwise, do not give them the treatment. However, this policy could be wrong and we do not know how certain we are of the decision. Instead, we would like a decision rule that quantitatively characaterizes our uncertainty by defining a support region. For instance, if $\mathbb{P}(CATE(x) > \alpha) > 0.9$, then give the patient treatment, otherwise decide that you don't have enough information to make an informed decision.

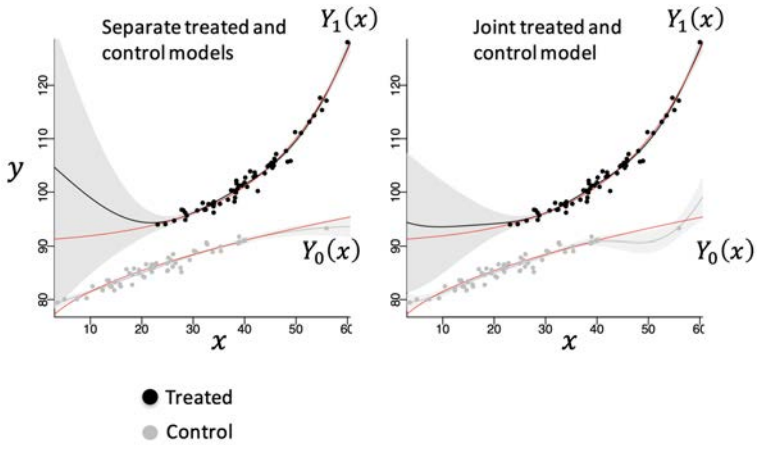

Figure 1: Example Gaussian Processes. On the right, we have a single Gaussian Process for concatenated (x, t). This model shares parameters across potential outcomes. The graph on the right shows two Gaussian processes, one for t=1 and one for t=0. One advantage of sharing parameters as in the joint model, is that we can learn from less data.

2.3.2 Neural Networks

We can use neural networks to learn non-linear models. One example architecture is shown in Figure 2 [SJS17]. In the Figure 1, we apply several nonlinear layers on our input x and then apply a treatment layer Φ. Note that we share models in the beginning to learn the joint representation. After that, we use separate layers to get different outcomes. Another important thing to note is that we apply treatment after we convolve the input x because treatment features often get lost if we use them with the input x when x has strong features.

2.4 Matching

- Idea: Find each unit's long-lost counterfactual identical twin and check up on his outcome. In practice, we can do this by identifying another unit (x_1) who is very similar (by some distance metric) to our selected unit (x_0) but belongs to the other treatment group and then assess the outcome of the other unit x_1. We can think of this as an approximation of determining the counterfactual for our unit of interest x_0. We use this for estimating CATE and ATE. ***

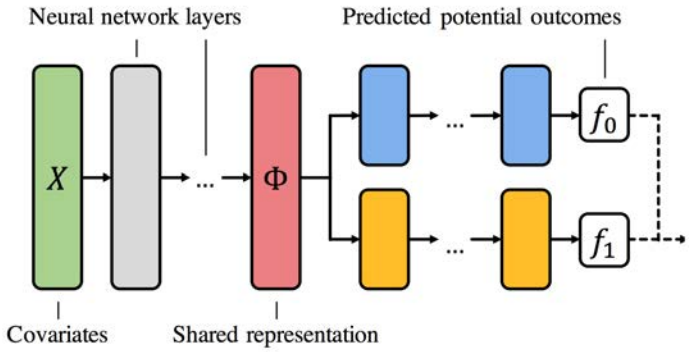

Figure 2: Example Neural Network

2.4.1 1-NN Matching

- Let $d(\cdot, \cdot)$ be a metric between x's. For each i, define

$$j(i) = \underset{j:t_j \neq t_i}{\operatorname{argmin}} d(x_j, x_i)$$

  where $j(i)$ is the nearest counterfactual neighbor of i.

- $t_i = 1$, unit i is treated:

$$\widehat{CATE}(x_i) = y_i - y_{j(i)}$$

- $t_i = 0$, unit i is control:

$$\widehat{CATE}(x_i) = y_{j(i)} - y_i$$

- We can generalize the two equations above into the following single equation:

$$\widehat{CATE}(x_i) = (2t_i - 1)(y_i - y_{j(i)})$$

- As usual, we can take the expected value, or average, of CATE in order to find the ATE:

$$\widehat{ATE} = \frac{1}{n} \sum_{i=1}^{n} \widehat{CATE}(x_i)$$

2.4.2 Properties of Matching

- Interpretable, especially in small-sample regime
- Nonparametric
- Heavily reliant on the underlying metric
- Could be misled by features which don't affect the outcome
- Matching is not used much in practice, but there are compelling reasons to use it. For instance, in healthcare, Nigam Shah at Stanford implemented a form of matching via the Green Button approach. This method searches electronic medical records for patients similar to a patient of interest, in terms of background, health history, and responsiveness to treatment. By finding nearest neighbors of this patient, doctors can determine how other, similar, patients react to treatments and can use this information to guide the care of their patient. A benefit of this approach is the high level of interpretability.

- 1-NN matching can have the same problems as kNN in machine learning. For instance, Euclidean distance doesn't work well as a distance metric in high dimensions. Also, this method will not work well if there are not manyt samples. If there are no individuals with treatment T_0 near a particular individual with T_1, we do not have a good counterfactual example.

2.4.3 Covariate Adjustment and Matching

- Matching is equivalent to covariate adjustment with two 1-nearest neighbor classifiers:

$$\hat{Y}_1(x) = y_{NN_1(x)}, \hat{Y}_0(x) = y_{NN_0(x)}$$

  where $y_{NN_t(x)}$ is th nearest-neighbor of x among units with treatment assignment $t = 0, 1$

- 1-NN matching is in general inconsistent, though only with small bias.

3 Propensity score re-weighting

Propensity score re-weighting (henceforth *PSR*)is another tool to estimate ATE.

3.1 Main idea

The main idea behind this method is that we turn observational study into a pseudo-randomized trial by re-weighting the sample weights. This is similar to statistical methods such as *Importance Sampling*. *PSR* is helpful to deal with imbalanced datasets. Mathematically, if we see that $p(x|t = 0)$ and $p(x|t = 1)$ are vastly different, we want to add weight functions w_0, w_1 to our sample points such that $p(x|t = 0)w_0(x) \approx p(x|t = 1)w_1(x)$. Intuitively, we can see how and why *PSR* assigns weights from Figure 3. Basically we want to add more weights to red points that are inside blue points and vise versa. The reason is that they represent more unique features.

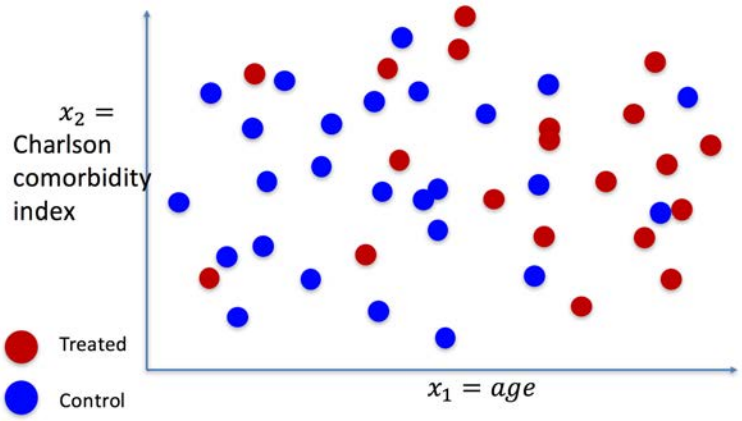

Figure 3: Inverse propensity score re-weighting

3.2 Propensity score algorithm

Propensity score algorithm is an algorithm that estimates ATE using PSR. First of all, propensity score is defined as $p(T = t|x)$. Therefore propensity score is the probability that the patient would receive treatment t given that patient's data. We can use many Machine Learning algorithms to get an estimation for $p(T = t|x)$. Therefore given the data $(x_1, t_1, y_1), \ldots, (x_n, t_n, y_n)$, our algorithm to estimate ATE is:

1. Estimate $p(T=t|x)$ using any Machine Learning algorithm
2. $\widehat{ATE} = \frac{1}{n}\sum_{i,t_i=1}\frac{y_i}{\hat{p}(t_i=1|x_i)} - \frac{1}{n}\sum_{i,t_i=0}\frac{y_i}{\hat{p}(t_i=0|x_i)}$

Note that we are multiplying each y_i by its' inverse propensity score. This is what we mean by re-weighting. In the special case of randomized trial where $p(T=t|x)=0.5$, our expression for ATE reduces to:

$$\widehat{ATE} = \frac{2}{n}\sum_{i,t_i=1} y_i - \frac{2}{n}\sum_{i,t_i=0} y_i$$

The additional multiplication coefficient 2 is still fine as the summation involves roughly around $\frac{n}{2}$, not n terms. Now let's derive the above formula for ATE. First recall that:

$$ATE = \mathbb{E}_{x\sim p(x)}[\mathbb{E}[Y_1|x,T=1] - \mathbb{E}[Y_0|x,T=0]]$$

But we have only samples for:

$$\mathbb{E}_{x\sim p(x|T=1)}[\mathbb{E}[Y_1|x,T=1]]$$
$$\mathbb{E}_{x\sim p(x|T=0)}[\mathbb{E}[Y_0|x,T=0]]$$

But using Bayes Theorem, we know that:

$$p(x) = p(x|T=1) * \frac{p(T=1)}{p(T=1|x)}$$

$$p(x) = p(x|T=0) * \frac{p(T=0)}{p(T=0|x)}$$

Note that denominators are corresponding propensity scores! Then we can rewrite expectations as:

$$\mathbb{E}_{x\sim p(x)}[\mathbb{E}[Y_1|x,T=1]] = \mathbb{E}_{x\sim p(x|T=1)}[\frac{p(T=1)}{p(T=1|x)}\mathbb{E}[Y_1|x,T=1]]$$

$$\mathbb{E}_{x\sim p(x)}[\mathbb{E}[Y_1|x,T=0]] = \mathbb{E}_{x\sim p(x|T=0)}[\frac{p(T=0)}{p(T=0|x)}\mathbb{E}[Y_0|x,T=0]]$$

To close this discussion, there are couple things to keep in mind when using propensity algorithm:

- If $p(T=t|x)$ is known, then propensity scores re-weighting is consistent
- Usually the propensity score is unknown and must be estimated
- If theres not much overlap in the date , propensity scores become non-informative and easily miscalibrated.
- Weighting can create large variance and large errors for small propensity scores
- We can only calculate ATE

4 More Ideas and Methods

4.1 Natural experiments

- The idea of natural experiments is to look for observational data in which the desired treatment happened to be given to some members of the population and not given to other members.

- As an example, suppose we want to study how stress during pregnancy affects later child development. We can't conduct a randomized controlled trial, so instead we can look for a natural experiment in which otherwise similar populations were split into "treatment" (i.e. stress during pregnancy) and "control" (i.e. no extra stress during pregnancy).
- The Cuban missile crisis of October 1962 caused increased levels of stress because people were afraid a nuclear war would break out. We could compare children who were in utero during the crisis with children from immediately before and after.

4.2 Instrumental Variables

- An instrumental variable is a variable which affects treatment assignment but not the outcome.
- We could use instrumental variables to answer the question: are private schools better than public schools? Again, we can't conduct a RCT here because we can't force people which school to go to. However, we could randomly give out vouchers to some students, giving them an opportunity to attend private schools. In this case, the voucher assignment is the instrumental variable.

5 Conclusion

We discussed two approaches to use machine learning for causal inference:

1. Predict outcome given features and treatment, then use resulting model to impute counterfactuals (covariate adjustment)
2. Predict treatment using features (propensity score), then use to reweight outcome or stratify the data

It is also important to think through causal graphs to see whether problem is setup appropriately and whether assumptions hold before doing any reductions to Machine Learning.

References

[SJS17] Shalit, Johansson, and Sontag. Estimating individual treatment effect: Generalization. *Health Affairs*, 2017.

6.S897 / HST.956 Machine Learning for Healthcare

Spring 2019