So let’s get started.

Welcome everyone.

It is my great pleasure to introduce our speaker today, Dr. Edward Kennedy, who is an assistant professor at the Department of Statistics and Data Science at Carnegie Mellon University.

Dr. Kennedy got his MA in statistics and PhD in biostatistics from University of Pennsylvania. He’s an expert in methods for causal inference, missing data and machine learning, especially in settings involving high dimensional and complex data structures. He has also been collaborating on statistical applications in criminal justice, health services, medicine and public policy.

Today’s going to share with us his recent work in the space of heterogeneous causal effect estimation.

Welcome Edward, the floor is yours.

Thanks so much, (clears throat) yeah, thanks for the invitation. I’m happy to talk to everyone today about this work I’ve been thinking about for the last year or so. Sort of excited about it.

Yeah, so it’s all about doubly robust estimation. Maybe before I start, I don’t know what the standard approach is for questions, but I’d be more than happy to take any questions throughout the talk and I can always sort of adapt and focus more on different parts of the room,
what people are interested in. I’m also trying to get used to using Zoom, I’ve been teaching this big lecture course so I think I can keep an eye on the chat box too if people have questions that way, feel free to just type something in. Okay. So yeah, this is sort of standard problem non-causal inference but I’ll give some introduction. The kind of classical target that people go after in causal inference problems is what’s often called the average treatment effect. So this tells you the mean outcome if everyone was treated versus if everyone was untreated, for example. So this is, yeah, sort of the standard target. We know quite a bit about estimating this parameter under no unmeasured confounding kinds of assumptions. So just as a just sort of point this out, a lot of my work is sort of focused on the statistics of causal inference, how to estimate causal parameters well in flexible non-parametric models. So we know quite a bit about this average treatment effect parameter. There are still some really interesting open problems, even for this sort of most basic parameter, which I’d be happy to talk to people about, this is just one number, it’s an overall summary of how people respond to treatment, on average.
It can obscure potentially important heterogeneity.

So for example, very extreme case would be where half
the population is seeing a big benefit
from treatment and half is seeing severe harm,
then you would completely miss this
by just looking at the average treatment effect.
So this motivates going beyond this,
maybe looking at how treatment effects can vary
across subject characteristics.

All right, so why should we care about this?
Why should we care how treatment effects vary in this way?
So often when I talk about this,
people’s minds go immediately to optimal treatment
regimes,
which is certainly an important part of this problem.
So that means trying to find out who’s benefiting
from treatment and who is not or who’s being harmed.
And then just in developing
a treatment policy based on this,
where you treat the people who benefit,
but not the people who don’t.
This is definitely an important part
of understanding heterogeneity,
but I don’t think it’s the whole story.
So it can also be very useful just
to understand heterogeneity from a theoretical perspective,
not just trying to optimally assign
the system
that you’re studying and not only that,
0:03:53.17 –> 0:03:55.48 the current treatment that’s available, 
0:03:55.48 –> 0:03:56.85 but if you find, for example, 
0:03:56.85 –> 0:04:00.53 that there are portions of the subject population 
0:04:00.53 –> 0:04:02.81 that are not responding to the treatment, 
0:04:02.81 –> 0:04:05.45 maybe you should then go off and try and develop 
0:04:05.45 –> 0:04:09.003 a treatment that would better aim at these people. 
0:04:09.91 –> 0:04:11.58 So lots of different reasons why you might care 
0:04:11.58 –> 0:04:12.413 about heterogeneity, 
0:04:12.413 –> 0:04:16.33 including devising optimal policies, 
0:04:16.33 –> 0:04:17.633 but not just that. 
0:04:18.89 –> 0:04:21.41 And this really plays a big role across lots 
0:04:21.41 –> 0:04:24.13 of different fields as you can imagine. 
0:04:24.13 –> 0:04:28.89 We might want to target policies based on how people 
0:04:28.89 –> 0:04:32.033 are responding to a drug or a medical treatment. 
0:04:33 –> 0:04:36.45 We’ll see a sort of political science example here. 
0:04:36.45 –> 0:04:39.31 So this is just a picture of what you should maybe think 
0:04:39.31 –> 0:04:42.31 about as we’re talking about this problem 
0:04:42.31 –> 0:04:43.95 with heterogeneous treatment effects. 
0:04:43.95 –> 0:04:45.75 So this is a timely example. 
0:04:45.75 –> 0:04:46.84 So it’s looking at the effect 
0:04:46.84 –> 0:04:48.633 of canvassing on voter turnout. 
0:04:49.84 –> 0:04:52.97 So this is the effect of being sort of reminded 
0:04:52.97 –> 0:04:55.24 in a face-to-face way to vote 
0:04:55.24 –> 0:04:56.94 that there’s an election coming up 
0:04:58 –> 0:05:00.37 and how this effect varies with age. 
0:05:00.37 –> 0:05:03.5 And so I’ll come back to where this plot came from 
0:05:03.5 –> 0:05:06.63 and the exact sort of data structure and analysis, 
0:05:06.63 –> 0:05:09.33 but just as a picture to sort of make things concrete. 
0:05:10.91 –> 0:05:14.8 It looks like there might be some sort of positive effect 
0:05:14.8 –> 0:05:16.49 of canvassing for younger people, 
0:05:16.49 –> 0:05:18.02 but not for older people,
there might be some non-linearity. So this might be useful for a number of reasons. You might not want to target the older population with canvassing, because it may not be doing anything, you might want to try and find some other way to increase turnout for this group right. Or you might just want to understand sort of from a psychological, sociological, theoretical perspective, what kinds of people are responding to this sort of thing? And so this is just one simple example you can keep in mind. So what’s the state of the art for this problem? So in this talk, I’m going to focus on this conditional average treatment effect here. So it’s the expected difference if people of type X were treated versus not expected difference in outcomes. This is kind of the classic or standard parameter that people think about now in the heterogeneous treatment effects literature, there are other options you could think about risk ratios, for example, if outcomes are binary. A lot of the methods that I talk about today will have analogs for these other regions, but there are lots of fun, open problems to explore here. How to characterize heterogeneous treatment effects when you have timeframe treatments, continuous treatments, of cool problems to think about. But anyways, this kind of effect where we have
a binary treatment and some set of covariates,
there’s really been this proliferation of proposals
in recent years for estimating this thing
in a flexible way that goes beyond just fitting
a linear model and looking at some interaction terms.
(clears throat)
So I guess I’ll refer to the paper for a lot
of these different papers that have thought about this.
People have used, sort of random forests
and tree based methods basing out
of a regression trees, lots of different variants
for estimating this thing.
So there’ve been lots of proposals,
lots of methods for estimating this,
but there’s some really big theoretical gaps
in this literature.
So one, yeah, this is especially true
when you can imagine that this conditional effect
might be much more simple
or sparse or smooth than the rest
of the data generating process.
So you can imagine you have some
potentially complex propensity score describing
the mechanism by which people are treated
based on their covariates.
You have some underlying regression functions
that describe this outcome process,
how their outcomes depend on covariates,
whether they’re treated or not.
These could be very complex and messy objects,
but this CATE might be simpler.
And in this kind of regime, there’s very little known. I’ll talk more about exactly what I mean by this in just a bit. So one question is, how do we adapt to this kind of structure? And there are really no strong theoretical benchmarks in this world in the last few years, which means we have all these proposals, which is great, but we don’t know which are optimal or when or if they can be improved in some way. What’s the best possible performance that we could ever achieve at estimating this quantity in the non-parametric model without adding assumptions? So these kinds of questions are basically entirely open in this setup. The point of this work is really to try and push forward to answer some of these questions. So one is just to provide more flexible estimators of this guy and specifically to show, give stronger error guarantees on estimating this. So that we can use a really diverse set of methods for estimating this thing in a doubly robust way and still have some rigorous guarantees about how well we’re doing. That part is more practical. It’s more about giving a method that people can actually implement and practice that’s pretty straight forward.
0:09:40.62 –> 0:09:43.69 it looks like a two stage progression procedure
0:09:43.69 –> 0:09:46.339 and being able to say something about this
0:09:46.339 –> 0:09:51.339 that’s model free and and agnostic about both
0:09:51.63 –> 0:09:53.16 the underlying data generating process
0:09:53.16 –> 0:09:56.99 and what methods we’re using to construct the estimator.
0:09:56.99 –> 0:09:59.32 This was lacking in the previous literature.
0:09:59.32 –> 0:10:02.91 So that’s one side of this work, which is more practical.
0:10:02.91 –> 0:10:04.913 I think I’ll focus more on that today,
0:10:06.34 –> 0:10:08.51 but we can always adapt as we go,
0:10:08.51 –> 0:10:10.16 if people are interested in other stuff.
0:10:10.16 –> 0:10:12.31 I’m also going to talk a bit about an analysis of this,
0:10:12.31 –> 0:10:15.013 just to show you sort of how it would work in practice.
0:10:16.34 –> 0:10:17.41 So that’s one part of this work.
0:10:17.41 –> 0:10:21.64 The second part is more theoretical and it says,
0:10:21.64 –> 0:10:23.86 so I don’t want to just sort of construct
0:10:23.86 –> 0:10:27.31 an estimator that has the nice error guarantees,
0:10:27.31 –> 0:10:29.23 but I want to try and figure out what’s
0:10:29.23 –> 0:10:31.22 the best possible performance I could ever get
0:10:31.22 –> 0:10:33.723 at estimating these heterogeneous effects.
0:10:35.64 –> 0:10:38.7 This turns out to be a really hard problem
0:10:38.7 –> 0:10:40.233 with a lot of nuance,
0:10:41.77 –> 0:10:43.27 but that’s sort of the second part
0:10:43.27 –> 0:10:45.84 of the talk which maybe is a little tackle
0:10:45.84 –> 0:10:48.363 in a bit less time.
0:10:49.92 –> 0:10:50.94 So that’s kind of big picture.
0:10:50.94 –> 0:10:52.853 I like to give the punchline of the talk at the start,
0:10:52.853 –> 0:10:56.453 just so you have an idea of what I’m going to be covering.
0:10:57.39 –> 0:11:01.34 And yeah, so now let’s go into some details.
0:11:01.34 –> 0:11:03.32 So we’re going to think about this sort
0:11:03.32 –> 0:11:05.56 of classic causal inference data structure,
where we have $n$ iid observations, we have covariates $X$, which are $D$ dimensional, binary treatment for now, all the methods that I’ll talk about will work without any extra work in the discrete treatment setting if we have multiple values. The continuous treatment setting is more difficult it turns out. And some outcome $Y$ that we care about. All right, so there are a couple of characters in this talk that will play really important roles. So we’ll have some special notation for them. This is the chance of being treated, given your covariates. So some people might be more or less likely to be treated depending on their baseline covariates, $X$. Muse of a, this will be an outcome regression function. So it’s your expected outcome given your covariates and given your treatment level. And then we’ll also later on in the talk use this ada, which is just the marginal outcome regression. So without thinking about treatment, just how the outcome varies on average as a function of $X$. And so those are the three main characters in this talk, we’ll be using them throughout. So under these standard causal assumptions there’s a really amazing group at Yale that are focused on dropping these assumptions.
So lots of cool work to be done there, but we’re going to be using them today. So consistency, we’re roughly thinking this means there’s no interference, this is a big problem in causal inference, but we’re going to say that my treatments can affect your outcomes, for example. We’re going to think about the case where everyone has some chance at receiving treatment, both treatment and control, and then we have no unmeasured confounding. So we’ve collected enough sufficiently relevant covariates that once we conditioned on them, look within levels of the covariates, the treatment is as good as randomized. So under these three assumptions, this conditional effect on the left-hand side here can just be written as a difference in regression functions. It’s just the difference in the regression function under treatment versus control, sort of super simple parameter right. So I’m going to call this thing Tau. This is just the regression under treatment minus the regression under control. So you might think, we know a lot about how to estimate regression functions non-parametrically they’re really nice, min and max lower bounds that say we can’t do better uniformly across the model without adding some assumptions or some extra structure. The fact that we have a difference
in regression doesn't seem like it would make things more complicated than just the initial regression problem, but it turns out it really does, it’s super interesting, this is one of the parts of this problem that I think is really fascinating.

So just by taking a difference in regressions, you completely change the nature of this problem from the standard non-parametric regression setup. So let’s get some intuition for why this is the case. So why isn’t it optimal just to estimate the two regression functions and take a difference, for example?

So let’s think about a simple data generating process where we have just a one dimensional covariate, it’s uniform on minus one, one, we have a simple step function propensity score and then we’re going to think about a regression function, both under treatment and control that looks like some kind of crazy polynomial from this Gyorfi textbook, I’ll show you a picture in just a minute. The important thing about this polynomial is that it’s non-smooth, it has a jump, has some kinks in it and so it will be hard to estimate, but we’re taking both the regression function under treatment and the regression function under control to be equal, they’re equal to this same hard estimate polynomial function.
And so that means the difference is really simple, it’s just zero, it’s the simplest conditional effect you can imagine, not only constant, but zero. You can imagine this probably happens a lot in practice.

where we have treatments that are not extremely effective for everyone in some complicated way.

So the simplest way you would estimate this conditional effect is just take an estimate of the two regression functions and take a difference. Sometimes I’ll call this plugin estimator. There’s this paper by Kunzel and colleagues, call it the T-learner.

So for example, we can use smoothing splines, estimate the two regression functions and take a difference. And maybe you can already see what’s going to go wrong here.

So these individual regression functions by themselves are really hard to estimate. They have jumps and kinks, they’re messy functions. And so when we try and estimate these with smoothing splines, for example, we’re going to get really complicated estimates that have some bumps, It’s hard to choose.

When we take the difference, we’re inheriting the sort of complexity of the individual regression functions. and so it’s not doing a good job of exploiting this simple structure in the conditional effect.
This is sort of analogous to this intuition that people have that interaction terms might be smaller or less worrisome than sort of main effects in a regression model. Or you can think of the muse as sort of main effects and the differences as like an interaction.

So here’s a picture of this data in the simple motivating example. We've got treated people on the left and untreated people on the right and this gray line is the true, that messy, weird polynomial function that we're thinking about. So here’s a jump and there’s a couple of kinks here and there’s confounding. So treated people are more likely to have larger Xs, untreated people are more likely to have smaller Xs.

So what happens here is the function is sort of a bit easier to estimate on the right side. And so for treated people, we're going to take a sort of larger bandwidth, get a smoother function. For untreated people, it’s harder to estimate on the left side and so we’re going to need a small bandwidth to try and capture this jump, for example, this discontinuity. And so what’s going to happen is when you take a difference of these two regression estimates, these black lines are just the standard smoothing that spline estimates that you’re getting are with one line of code, when you take a difference.
you’re going to get something that’s very complex and messy and it’s not doing a good job of recognizing that the regression functions are the same under treatment and control.

So what else could we do? This maybe points to this fact that the plugin estimator breaks, it doesn’t do a good job of exploiting a structure, but what other options do we have?

So let’s say that we knew the propensity scores. So for just simplicity, say we were in a trial, for example, an experiment, where we randomized everyone to treat them with some probability that we knew. In that case, we could construct a pseudo outcome, which is just like an inverse probability weighted outcome, which has exactly the right conditional expectation, its conditional expectation is exactly equal to that conditional effect. And so when you did a non-parametric regression of the pseudo outcome on X, it would be like doing an oracle regression of the true difference in potential outcomes, it has exactly the same conditional expectation. And so this sort of turns this hard problem into a standard non-parametric regression problem. Now this isn’t a special case where we knew the propensity scores for the rest of the talk we’re gonna think about what happens when we don’t know these, what can we say?
So here’s just a picture of what we get in the setup.

So this red line is this really messy plug in estimator that we get that’s just inheriting that complexity of estimating the individual regression functions.

And then these black and blue lines are IPW and doubly robust versions that exploit this underlying smoothness and simplicity of the heterogeneous effects, the conditional effects.

So this is just a motivating example to help us get some intuition for what’s going on here.

These results are sort of standard in this problem, we’ll come back to some simulations later on.

And so now our goal is going to study the error of the sort of inverse weighted kind of procedure, but a doubly robust version.

We’re going to give some new model free error guarantees, which let us use very flexible methods and it turns out we’ll actually get better areas than what were achieved previously in literature.

Even when focusing specifically on some particular method.

And then again, we’re going to see how well can we actually do estimating this conditional effect in this problem.

Might be a good place to pause and see if people have any questions.

Okay.

Feel free to shout out any questions or stick them on the chat if any come up.
a pretty simple two-stage doubly robust estimator, which I’m going to call the DR-learner, this is following this nomenclature that’s become kind of common in the heterogeneous effects literature where we have letters and then a learner. So I’m calling this the DR-Learner, but this is not a new procedure, but the version that I’m going to analyze has some variances, but it was actually first proposed by Mike Vanderlande in 2013, was used in 2016 by Alex Lucca and Mark Vanderlande. So they proposed this, but they didn’t give specific error bounds. I think relatively few people know about these earlier papers because this approach was then sort of rediscovered in various ways after that in the following years, typically in these later versions, people use very specific methods for estimating, for constructing the estimator, which I’ll talk about in detail in just a minute, for example, using kernel kind of methods, Local polynomials and this paper used a sort of series or spline and regression. These papers are nice ways of doing doubly robust estimation, but they had a couple of drawbacks, which we’re going to try and build on in this work. So one is, we’re going to try not to commit
0:22:05.44 –> 0:22:07.04 to using any particular methods.
0:22:07.04 –> 0:22:10.53 We’re going to see what we can say about error guarantees,
0:22:14.5 –> 0:22:16.17 And then we’re going to see
0:22:16.17 –> 0:22:19.44 if we can actually weaken the sort of assumptions
0:22:19.44 –> 0:22:22.44 that we need to get oracle type behavior.
0:22:22.44 –> 0:22:25.02 So the behavior of an estimator that we would see
0:22:25.02 –> 0:22:27.61 if we actually observed the potential outcomes
0:22:27.61 –> 0:22:29.18 and it turns out we’ll be able to do this,
0:22:29.18 –> 0:22:32.133 even though we’re not committing to particular methods.
0:22:33.61 –> 0:22:35.31 There’s also a really nice paper by Foster
0:22:35.31 –> 0:22:37.6 and Syrgkanis from last year,
0:22:37.6 –> 0:22:41.3 which also considered a version of this DR-learner
0:22:41.3 –> 0:22:44.054 and they had some really nice model agnostic results,
0:22:44.054 –> 0:22:45.72 but they weren’t doubly robust.
0:22:45.72 –> 0:22:48.12 So, in this work we’re going to try
0:22:48.12 –> 0:22:50.953 and doubly robust defy these these results.
0:22:53.51 –> 0:22:57.08 So that’s the sort of background and an overview.
0:22:57.08 –> 0:23:00.55 So let’s think about what this estimator is actually doing.
0:23:00.55 –> 0:23:02.9 So here’s the picture of this,
0:23:02.9 –> 0:23:05.01 what I’m calling the DR-learner.
0:23:05.01 –> 0:23:07.81 So we’re going to do some interesting sample splitting
0:23:07.81 –> 0:23:10.49 here and later where we split our sample
0:23:10.49 –> 0:23:12.82 in the three different groups.
0:23:12.82 –> 0:23:15.88 So one’s going to be used for nuisance training
0:23:15.88 –> 0:23:17.68 for estimating the propensity score.
0:23:19.32 –> 0:23:20.88 And then I’m also going to estimate
0:23:20.88 –> 0:23:25.88 the regression functions, but in a separate fold.
0:23:25.91 –> 0:23:28.76 So I’m separately estimating my propensity score
and regression functions.

This turns out to not be super crucial for this approach.

It actually is crucial for something I’ll talk about later in the talk,

this is just to give a nicer error bound.

So the first stage is we estimate these nuisance functions,

the propensity scores and the regressions.

And then we go to this new data that we haven’t seen yet,

our third fold of split data and we construct a pseudo outcome.

Pseudo outcome looks like this, it’s just some combination,

it’s like an inverse probability weighted residual term plus something like the plug-in estimator of the conditional effect.

So it’s just some function of the propensity score estimates and the regression estimates.

If you’ve used doubly robust estimators before you’ll recognize this as what we average when we construct a usual doubly robust estimator of the average treatment effect.

And so intuitively instead of averaging this year, we’re just going to regress it on covariates, that’s exactly how this procedure works.

So it’s pretty simple, construct the pseudo outcome, which we typically would average estimate the ate, now, we’re just going to do a regression of this thing on covariates in our third sample.

So we can write our estimator this way.
This e hat in notation just means some generic regression estimator. So one of the crucial points in this work, so I’m not going to, I want to see what I can say about the error of this estimator without committing to a particular estimator. So if you want to use random forests in that last stage, I want to be able to tell you what kind of error to expect or if you want to use linear regression or whatever procedure you like, the goal would be to give you some nice error guarantee. So (indistinct), and you should think of it as just your favorite regression estimator. So we take the suit outcome, we regress it on covariates, super simple, just create a new column in your dataset, which looks like this pseudo outcome. And then treat that as the outcome in your second stage regression. So here we’re going to get let’s say we split our sample into half for the second stage regression, we would get an over two kind of we’d be using half our sample for the second stage regression. You can actually just swap these samples in the you’ll get back the full sample size errors. So it would be as if you had used the full sample size all at once. That’s called Cross Fitting, it’s becoming sort of popular in the last couple of years.
So here’s a schematic of what this thing is doing. We split our data in the thirds, use one third testing to estimate the propensity score, another third to estimate the regression functions, and then we do a second stage regression of that pseudo outcome on covariates. So pretty easy, you can do this in three lines of code. Okay.

And now our goal is to say something about the error of this procedure, being completely agnostic about how we estimate these propensity scores, that regression functions and what procedure we use in this third stage or second stage. And it turns out we can do this by exploiting the sample splitting can come up with a strong guarantee that actually gives you smaller errors than what appeared in the previous literature when people focused on specific methods. And the main thing is we’re really exploiting the sample splitting. And then the other tool that we’re using is we’re assuming some stability condition on that second stage estimator, which is really mild, I’ll tell you what it is right now. So you say that regression estimator is stable, if when you add some constant to the outcome and then do a regression, you get something
that’s the same as if you do the regression and then add some constant. So it’s pretty intuitive, if a method didn’t satisfy this, it would be very weird and actually for the proof, we don’t actually need this to be exactly equal. So adding a constant pre versus post regression shouldn’t change things too much. You don’t have to have it be exactly equal, it still works if it’s just equal up to the error in the second stage regression. So that’s the first stability condition. The second one is just that if you have two random variables with the same conditional expectation, then the mean squared error is going to be the same up to constants. Again, any procedure that didn’t satisfy these two assumptions would be very bizarre. It’s a very mild stability conditions. And that’s essentially all we need. So now our benchmark here is going to be an oracle estimator that instead of doing a regression with the pseudo, it does a regression with the actual potential outcomes, Y, one, Y, zero. So we can think about the mean squared error of this estimator, so I’m using mean squared error, just sort of for simplicity and convention, you could think about translating this
0:28:37.8 –> 0:28:39.74 to other kinds of measures of risk.
0:28:39.74 –> 0:28:42.80 That would be an interesting area for future work.
0:28:43.77 –> 0:28:46.7 So this is the oral, our star is the Oracle
0:28:46.7 –> 0:28:47.93 the mean squared error.
0:28:47.93 –> 0:28:49.93 It’s the mean squared error you’d get for estimating
0:28:49.93 –> 0:28:52.4 the conditional effect if you actually saw
0:28:52.4 –> 0:28:53.55 the potential outcomes.
0:28:55.24 –> 0:28:57.02 So we get this really nice, simple result,
0:28:57.02 –> 0:28:59.45 which says that the mean squared error
0:28:59.45 –> 0:29:03.01 of that DR-learner procedure that uses the pseudo
outcomes,
0:29:03.01 –> 0:29:05.68 it just looks like the Oracle means squared error,
0:29:05.68 –> 0:29:08.13 plus a product of mean squared errors in estimating
0:29:08.13 –> 0:29:10.58 the propensity score and the regression function.
0:29:11.711 –> 0:29:16.39 It resembles the kind of doubly robust error results
0:29:16.39 –> 0:29:18.2 that you see for estimating average treatment effects,
0:29:18.2 –> 0:29:21.063 but now we have this for conditional effects.
0:29:22.77 –> 0:29:25.16 The proof technique is very different here compared
0:29:25.16 –> 0:29:29.03 to what is done in the average effect case.
0:29:29.03 –> 0:29:32.37 But the proof is actually very, very straightforward.
0:29:32.37 –> 0:29:35.13 It’s like a page long, you can take a look in the paper,
0:29:35.13 –> 0:29:37.68 it’s really just leaning on this sample splitting
0:29:37.68 –> 0:29:40.743 and then using stability in a slightly clever way.
0:29:42.47 –> 0:29:44.968 But the most complicated tool uses is just
0:29:44.968 –> 0:29:49.4 some careful use of the components
0:29:49.4 –> 0:29:51.613 of the estimator and iterated expectation.
0:29:52.51 –> 0:29:55.483 So it’s really a pretty simple proof, which I like.
0:29:57.4 –> 0:29:59.03 So yeah, this is the main result.
0:29:59.03 –> 0:30:01.98 And again, we’re not assuming anything beyond
0:30:01.98 –> 0:30:03.95 this mild stability here, which is nice.
0:30:03.95 –> 0:30:07.1 So you can use whatever regression procedures you like.
And this will tell you something about the error that relates to the Oracle error that you would get if you actually observed the potential outcomes. So this is model free method-agnostic, it's also a finite sample down, there's nothing asymptotic here. This means that the mean squared error is upper bounded up to some constant times this term on the right. So there's no end going to infinity or anything here either. So the other crucial point of this is because we have a product of mean squared errors, you have the kind of usual doubly robust story. So if one of these is small, the product will be small, potentially more importantly, if they're both kind of modest sized because both, maybe the propensity score and the regression functions are hard to estimate the product will be potentially quite a bit smaller than the individual pieces. And this is why this is showing you that sort of plugging approach, which would really just be driven by the mean squared error for estimating the regression functions can be improved by quite a bit, especially if there's some structure to exploit in the propensity scores. Yeah, so in previous work people used specific methods. So they would say I'll use maybe series estimators or current estimators and then the error bound was actually bigger.
than what we get here.
So this it’s a little surprising that you can get a smaller error bound under weaker assumptions,
but this is a nice advantage of the sample splitting trick here.
Now that you have this nice error bound you can plug in sort of results from any of your favorite estimators.
So we know lots about mean squared error for estimating regression functions.
So for example, you think about smooth functions.
So these are functions that are close to their tailored, approximations, the strict definition,
which may be I’ll pass in the interest of time.
Then you can say, for example, if PI is alpha smooth,
so it has alpha partial derivatives with the highest order Lipschitz then we know that you can estimate that a propensity score
with the mean squared error that looks like n to the minus two alpha over two alpha plus D,
this is the usual non-parametric regression.
You can say the same thing for the regression functions.
If they’re beta smooth, then we can estimate them at the usual non-parametric rate,
then we could say, okay, suppose the conditional effect,
and gamma, it can’t be smaller than beta, it has to be at least as smooth.
0:33:05.39 –> 0:33:07.51 as the regression functions and in practice,
0:33:07.51 –> 0:33:09.22 it could be much more smooth.
0:33:09.22 –> 0:33:12.44 So for example, in the case where the CATE is just zero
0:33:12.44 –> 0:33:17.17 or constant, Gamma’s like infinity, infinitely smooth.
0:33:17.17 –> 0:33:20.25 Then if we use a second stage estimator that’s optimal
0:33:20.25 –> 0:33:22.15 for estimating Gamma smooth functions,
0:33:22.15 –> 0:33:25.28 we can just plug in the error rates
0:33:25.28 –> 0:33:26.33 that we get and see
0:33:26.33 –> 0:33:28.39 that we get a mean squared error bound
0:33:28.39 –> 0:33:30.39 that looks like the Oracle rate.
0:33:30.39 –> 0:33:33.25 This is the rate we would get if we actually observed
0:33:33.25 –> 0:33:34.698 the potential outcomes.
0:33:34.698 –> 0:33:37.29 And then we get this product of mean squared errors.
0:33:37.29 –> 0:33:39.64 And so whenever this product, it means squared errors
0:33:39.64 –> 0:33:42.01 is smaller than the Oracle rate,
0:33:42.01 –> 0:33:45.77 then we’re achieving the Oracle rate up to constants,
0:33:45.77 –> 0:33:46.89 the same rate that we would get
0:33:46.89 –> 0:33:49.083 if we actually saw Y one minus Y zero.
0:33:49.083 –> 0:33:52.799 And so you can work out the conditions,
0:33:52.799 –> 0:33:56.07 what you need to make this term smaller than this one,
0:33:56.07 –> 0:33:57.53 that’s just some algebra
0:33:57.53 –> 0:34:01.773 and it has some interesting structure.
0:34:02.62 –> 0:34:07.05 So if the average smoothness of the two nuisance
0:34:07.05 –> 0:34:09.42 the propensity score and the regression function
0:34:09.42 –> 0:34:14.02 is greater than D over two divided by some inflation
0:34:14.02 –> 0:34:17.53 factor, then you can say that you’re achieving
0:34:17.53 –> 0:34:19.48 the same rate as this Oracle procedure.
0:34:22.23 –> 0:34:26.66 So the analog of this for the average treatment effect
0:34:26.66 –> 0:34:27.63 or the result you need

25
for the standard doubly robust estimate,
or the average treatment effect
is that the average smoothness is greater than $D$ over two.
So here we don’t have $D$ over two,
we have $D$ over two over one plus $D$ over gamma.
So this is actually giving you a sort
of a lower threshold for achieving Oracle rates
in this problem.
So, because it’s a harder problem,
we need weaker conditions
on the nuisance estimation to behave like an Oracle
and how much weaker those conditions
are, depends on the dimension of the covariates
and the smoothness of the conditional effect.
So if we think about the case where the conditional effect
is like infinitely smooth,
then we recovered the usual condition that we need
for the doubly robust estimator to be root
and consistent as greater than $D$ over two.
But when dimension is for some non-trivial smoothness,
then we’re somewhere in between sort of when
a plugin is optimal and this nice kind of parametric setup.
So this is just a picture of the rates here
which is useful to keep in mind.
Here on the x-axis, we have the smoothness
of the nuisance functions.
You can think of this as the average smoothness
of the propensity score in regression functions.
And again, in this holder smooth model, which is a common model people use in non-parametrics, the more smooth things are the easier it is to estimate them. And then here we have the mean squared error for estimating the conditional effect. So here is the minimax lower bounce, this is the best possible mean squared error that you can achieve for the average treatment effect. This is just to kind of anchor our results and think about what happens relative to this nicer, simpler parameter, which is just the overall average and not the conditional average. So here is the Oracle rate. This is the rate that we would achieve in this problem if we actually observed the potential outcomes. So it’s lower than Rudin, it’s a bigger error. Here’s what you would get with the plugin. This is just really inheriting the complexity and estimating the regression functions individually.
It doesn’t capture this CATE smoothness and so you need the regression functions to be sort of infinitely smoother or as smooth as the CATE to actually get Oracle efficiency with the plugin estimator. It’s this plugin as big errors, if we use this DR-learner approach, we close this gap substantially. So we can say that we’re hitting this Oracle rate. Once we have a certain amount of smoothness of the nuisance functions and in between we get an error that looks something like this. So this is just a picture of this row results showing, the improvement of the DR-learner approach here over a simple plug estimator. So yeah, just the punchline here is this simple two-stage doubly robust approach can do a good job adapting to underlying structure in the conditional effect, even when the nuisance stuff, the propensity scores and the underlying regression functions are more complex or less smooth in this case. This is just talking about the relation to the average treatment effect conditions, which I mentioned before. So you can do the same thing for any generic regression methods you like. So in the paper, I do this for smooth models and sparse models, which are common in these non-parametric settings.
where you have high dimensional Xs and you believe that some subset of them are the ones that matter.

So I’ll skip past this, if you’re curious though, all the details are in the paper.

So you can say, what kind of sparse should be doing need in the propensity score in regression functions to be able to get something that behaves like an Oracle that actually saw the potential outcomes from the start.

You can also do the same kind of game where you compare this to what you need for the average treatment effect.

You can also do the same kind of game where you compare this to what you need for the average treatment effect.

Yeah, happy to talk about this offline or afterwards people have questions.

So there’s also a nice kind of side result which I think I’ll also go through quickly here.

From all this, is just a general Oracle inequality for regression when you have some estimated outcomes. So in some sense, there isn’t anything really special in our results that has to do with this particular pseudo outcome.

The proof that we have here works for any second stage or any two-stage sort of regression procedure where you first estimate some nuisance stuff, create a pseudo outcome that depends on this estimated stuff and then do a regression of the pseudo outcome on some set of covariates.

And so a nice by-product of this work, as you get a kind of similar error bound.
for just generic regression with pseudo outcomes.

This comes up in a lot of different problems, actually.

So one is when you want just a partly conditional effect.

So maybe I don’t care about how effects vary

with all the Xs, but just a subset of them,

then you can apply this result.

I have a paper with a great student, Amanda Costin,

who studied a version of this

regression with missing outcomes.

Again, these look like nonparametric regression problems

where you have to estimate some pseudo outcome
dose response curve problems, conditional IV effects,
partially linear IVs.

So there are lots of different variants where you need
do some kind of two-stage regression procedure like
this.

Again, you just need a stability condition
and you need some sample splitting
and you can give a similar kind of a nice rate result
that we got for the CATE specific problem,
but in generic pseudo outcome progression problem.

So we’ve got about 15 minutes,

I have some simulations,

which I think I will go over quickly.

So we did this in a couple simple models,

one, a high dimensional linear model.

It’s actually a logistic model where
we have 500 covariates and 50
of them have non-zero coefficients.

We just used the default lasso fitting in our
and compared plugin estimators
0:41:34.41 –> 0:41:36.84 to the doubly robust approach that we talked
0:41:36.84 –> 0:41:38.51 about and then also an ex-learner
0:41:38.51 –> 0:41:43.51 which is some sort of variants of the plug-in approach
0:41:43.81 –> 0:41:45.933 that was proposed in recent years.
0:41:46.89 –> 0:41:48.91 And the basic story is you get sort
0:41:48.91 –> 0:41:50.24 of what the theory predicts.
0:41:50.24 –> 0:41:54.117 So the DR-learner does better than these plug-in types
0:41:54.117 –> 0:41:57.82 of approaches in this setting.
0:41:57.82 –> 0:42:00.347 The nuisance functions are hard to estimate
0:42:00.347 –> 0:42:02.23 and so you don’t see a massive gain over,
0:42:02.23 –> 0:42:03.62 for example, the X-Learner,
0:42:03.62 –> 0:42:04.83 you do see a pretty massive gain
0:42:04.83 –> 0:42:06.323 over the simple plugin.
0:42:08.41 –> 0:42:09.65 And we’re a bit away
0:42:09.65 –> 0:42:13.07 from this Oracle DR-learner approach here,
0:42:13.07 –> 0:42:16.43 so that means great errors is relatively different.
0:42:16.43 –> 0:42:18.82 This is telling us that the nuisance stuff is hard
0:42:18.82 –> 0:42:21.003 to estimate in this simulation set up.
0:42:22.3 –> 0:42:23.63 Here’s another simulation based
0:42:23.63 –> 0:42:25.973 on that plot I showed you before.
0:42:27.75 –> 0:42:30.97 And so here, I’m actually estimating the propensity scores,
0:42:30.97 –> 0:42:33.07 but I’m constructing the estimates myself
0:42:33.07 –> 0:42:35.45 so that I can control the rate of convergence
0:42:35.45 –> 0:42:38.57 and see how things change across different error rates
0:42:38.57 –> 0:42:40.62 for estimating with propensity score.
0:42:40.62 –> 0:42:41.52 So here’s what we see.
0:42:41.52 –> 0:42:44.08 So on the x-axis here,
0:42:44.08 –> 0:42:47.99 we have how well we’re estimating the propensity score.
0:42:47.99 –> 0:42:49.62 So this is a convergence rate
0:42:49.62 –> 0:42:52.44 for the propensity score estimator.
Y-axis, we have the mean squared error and then this red line is the plugin estimator, it's doing really poorly. It's not capturing this underlying simplicity of the conditional effects.

It's really just inheriting that difficulty in estimating the regression functions. Here's the X-learner, it's doing a bit better than the plugin, but it's still not doing a great job capturing the underlying simplicity and the conditional effect. This dotted line is the Oracle.

So this is what you would get if you actually observed the potential outcomes. This is doing a bit better than the X-learner, and here we see what we expect.

I'm just using smoothing splines everywhere, just defaults in R, it's like three lines of code, all the code's in the paper, too, if you want to play around with this.

And when it's really hard to estimate the propensity score, it's just a hard problem and we don't do much better than the X-learner. We still get some gain over the plugin in this case, well at all, you start seeing some pretty big gains by doing this doubly robust approach and at some point we start to roughly match the Oracle actually.
As soon as we’re getting something like into the quarter rates in this case, we’re getting close to the Oracle. So maybe I’ll just show you an illustration and then I’ll talk about the second part of the talk and very briefly if people have, want to talk about that, I’d be more than happy to. So here’s a study, which I actually learned about from Peter looking at effects of canvassing on voter turnout, so this is this timely study. Here’s the paper, there are almost 20,000 voters across six cities here. They’re randomly encouraged to vote in these local elections that people would go and talk to them face to face. You remember what that was like pre-pandemic. Here’s a script of the sort of canvassing that they did, just saying, reminding them of the election, giving them a reminder to vote. Hopefully I’m doing this for you as well, if you haven’t voted already. And so what’s the data we have here? We have a number of covariates things like city, party affiliation, some measures of the past voting history, age, family size, race. Again, the treatment is whether they work randomly contact is actually whether they were randomly assigned some cases, people couldn’t be contacted in the setup. So we’re just looking at intention
0:45:26.37 -> 0:45:28.11 to treat kinds of effects.
0:45:28.11 -> 0:45:30.21 And then the outcome is whether people voted
0:45:30.21 -> 0:45:32.25 in the local election or not.
0:45:32.25 -> 0:45:34.94 So just as kind of a proof of concept,
0:45:34.94 -> 0:45:37.05 I use this DR-learner approach,
0:45:37.05 -> 0:45:42.05 I just use two folds and use random forest separator
0:45:42.05 -> 0:45:54.97 So here's a histogram of the conditional effect estimates.
0:45:54.97 -> 0:46:02.64 So there's sort of a big chunk, a little bit above zero,
0:46:02.64 -> 0:46:04.26 but then there is some heterogeneity
0:46:04.26 -> 0:46:06.84 around that in this case.
0:46:06.84 -> 0:46:07.63 So there are some people
0:46:07.63 -> 0:46:12.49 who maybe seem especially responsive to canvassing,
0:46:12.49 -> 0:46:14.517 maybe some people who are going to know it
0:46:14.517 -> 0:46:17.267 and actually some are less likely to vote, potentially.
0:46:18.23 -> 0:46:21.06 This is a plot of the effect estimates
0:46:21.06 -> 0:46:24.43 from this DR-learner procedure,
0:46:22.35 -> 0:46:27.28 how this would work in practice across
0:46:24.43 -> 0:46:29.56 to potentially important covariate.
0:46:29.56 -> 0:46:33.9 So here's the age of the voter and then the party
0:46:33.9 -> 0:46:38.77 and the color here represents the size and direction
0:46:38.77 -> 0:46:41.38 of the CATE estimate of the conditional effect estimates,
0:46:41.38 -> 0:46:46.09 so blue is canvassing is having a bigger effect
0:46:46.09 -> 0:46:50.97 on voting in the next local election.
0:46:49.7 -> 0:46:54.213 Red means less likely to vote due to canvassing.
0:46:55.34 -> 0:46:58.73 So you can see some interesting structure here just briefly,
0:46:58.73 -> 0:47:00.58 the independent people,
0:47:00.58 –> 0:47:03.1 it seems like the effects are closer to zero.
0:47:03.1 –> 0:47:06.95 Democrats maybe seem more likely to be positively affected,
0:47:06.95 –> 0:47:10.58 maybe more so among younger people.
0:47:10.58 –> 0:47:12.33 It’s just an example of the kind of
0:47:13.24 –> 0:47:15.68 sort of graphical visualization stuff you could do
0:47:15.68 –> 0:47:17.08 with this sort of procedure.
0:47:18.31 –> 0:47:20.61 This is the plot I showed before, where here,
0:47:20.61 –> 0:47:22.19 we’re looking at just how the conditional
0:47:23.97 –> 0:47:25.3 And you can see some evidence
0:47:25.3 –> 0:47:28.49 that younger people are to canvassing.
0:47:32.92 –> 0:47:36.483 Older people, less evidence that there’s any response.
0:47:42.61 –> 0:47:45.743 I should stop here and see if people have any questions.
0:47:51.33 –> 0:47:53.15 - So Edward, can I ask a question?
0:47:53.15 –> 0:47:54.57 - Of course yeah.
0:47:54.57 –> 0:47:57.3 - I think we've discussed about point estimation.
0:47:57.3 –> 0:47:58.79 Does this approach also allows
0:47:58.79 –> 0:48:00.99 for consistent variance estimation?
0:48:00.99 –> 0:48:04.41 - Yeah, that’s a great question.
0:48:04.41 –> 0:48:07.58 Yeah, I haven’t included any of that here,
0:48:07.58 –> 0:48:09.513 but if you think about that.
0:48:10.69 –> 0:48:14.863 This Oracle result that we have.
0:48:16.761 –> 0:48:19.49 If these errors are small enough,
0:48:19.49 –> 0:48:21.57 so under the kinds of conditions that we talked about,
0:48:21.57 –> 0:48:25.64 then we’re getting an estimate of it looks like an Oracle
0:48:25.64 –> 0:48:28.62 has to meet or of the potential outcomes on the covari-
ates.
0:48:28.62 –> 0:48:30.63 And that means that as long as these are small enough,
0:48:30.63 –> 0:48:33.1 we could just port over any inferential tools
0:48:33.1 –> 0:48:35.496 that we like from standard non-parametric regression
0:48:35.496 –> 0:48:38.09 treating our pseudo outcomes as if they were
0:48:38.09 –> 0:48:41.29 the true existential outcomes, yeah.
0:48:41.29 –> 0:48:43.135 That’s a really important point,
0:48:43.135 –> 0:48:44.01 I’m glad you mentioned that.
0:48:44.01 –> 0:48:45.07 - Thanks.
0:48:45.07 –> 0:48:47.44 - So inference is more complicated
0:48:47.44 –> 0:48:49.59 and nuanced than non-parametric regression,
0:48:50.596 –> 0:48:54.823 but any inferential tool could be used here.
0:48:55.79 –> 0:48:57.27 - So operationally, just to think
0:48:57.27 –> 0:48:59.36 about how to operationalize the variance estimation
0:48:59.36 –> 0:49:02.82 also, does that require the cross fitting procedure
0:49:02.82 –> 0:49:05.91 where you’re swapping your D one D two
0:49:05.91 –> 0:49:08.403 in the estimation process and then?
0:49:09.55 –> 0:49:10.87 - Yeah, that’s a great question too.
0:49:10.87 –> 0:49:11.81 So not necessarily,
0:49:11.81 –> 0:49:14.14 so you could just use these folds
0:49:14.14 –> 0:49:17.13 for nuisance training and then go to this fold
0:49:17.13 –> 0:49:19.207 and then just forget that you ever used this data
0:49:21.31 –> 0:49:22.36 The drawback there would be,
0:49:22.36 –> 0:49:24.768 you’re only using a third of your data.
0:49:24.768 –> 0:49:26.48 If you really want to make full use
0:49:26.48 –> 0:49:27.67 of the sample size using
0:49:27.67 –> 0:49:30.97 the cross fitting procedure would be ideal,
0:49:30.97 –> 0:49:32.15 but the inference doesn’t change.
0:49:32.15 –> 0:49:34.58 So if you do cross fitting,
0:49:34.58 –> 0:49:35.91 you would at the end of the day,
0:49:35.91 –> 0:49:39.34 you’d get an out of sample CATE estimate
0:49:39.34 –> 0:49:42.144 for every single row in your data, every subject,
0:49:42.144 –> 0:49:44.44 but just where that CATE was built from other,
0:49:45.44 –> 0:49:47.07 the nuisance stuff for that estimate
0:49:47.07 –> 0:49:49.5 was built from other samples.
But at the end of the day, you’d get one big column with all these out of sample CATE estimates and then you could just use whatever inferential tools you like there.

Thanks.

- So, just got a few minutes.
- So maybe I’ll just give you a high level kind of picture of the stuff in the second part of this talk which is really about pursuing the fundamental limits of conditional effect estimation.

So what’s the best we could possibly do here? This is completely unknown, which I think is really fascinating.

So if you think about what we have so far, we’ve given these sufficient conditions under which this DR-learner is Oracle efficient, but a natural question here is what happens when those mean squared error terms are too big and so we can’t say that we’re getting the Oracle rate anymore.

Then you might say, okay, is this a bug with the DR-learner? Maybe I could have adapted this in some way to actually do better or maybe I’ve reached the limits of how well I can do for estimating the effect.

It doesn’t matter if I had gone to a different estimator, think I would’ve had the same kind of error.

So this is the goal of this last part of the work. So here we use a very different estimator, it’s built using this R-learner idea, which is reproducing RKHS extension of this.
classic double residual regression method of Robinson, which is really cool.

This is actually from 1988, so it's a classic method.

And so we study a non-parametric version of this built from local polynomial estimators.

And I'll just give you a picture.

It's quite a bit more complicated than that dr. Learner procedure.

So we again use this triple sample splitting and here it's actually much more crucial.

If you didn't use that triple sample splitting for the dr learner, you'd just get a slightly different Arab bound, but here it's actually really important.

We estimate propensity scores and regression functions.

Now the marginal regression functions, we combine these to get weights, Colonel weights.

We also combine them to get residuals.

So treatment residuals and outcome residuals.

This is like what you would get for this re Robinson procedure from econ.

Then we do instead of a regression of outcome residuals on treatment residuals, we do a weighted nonparametric regression of these residuals on the treatment residuals.

So that's the procedure a little bit more complicated. And again, this is,
I think there are ways to make this work well practically, but the goal of this work is really to try and figure out what's the best possible mean squared error that we could achieve. It's less about a practical method, more about just understanding how hard the conditional effect estimation problem is. And so we actually show that a generic version as long as you estimate the propensity scores and the regression functions with linear smoothers, with particular bias and various properties, which are standard in nonparametrics, you can actually get better mean squared error.

Then for the dr. Learner, we'll just give you a sense of what this looks like. So you get something that looks like an Oracle rate plus something like the squared bias from the new synced from the propensity score and regression functions. So before you had the product of mean squared errors, now we have the square of the bias of the two procedures, the mean squared error, and the propensity score in the regression function. And this gives you, this opens the door to under smoothing. So this means that you can estimate the propensity score and the regression functions in a suboptimal way. If you actually just care about the, these functions by themselves. So you drive down the bias that blows up the variance a little bit,
0:54:00.31 –> 0:54:02.6 but it turns out not to affect the conditional effect
0:54:02.6 –> 0:54:05.023 estimate too much if you do it in the right way.
0:54:06.04 –> 0:54:06.873 And so if.
0:54:06.873 –> 0:54:08.853 You, if you do this, you get.
0:54:11.313 –> 0:54:12.29 A rate that looks like this,
0:54:12.29 –> 0:54:15.267 you get an Oracle rate plus into the minus two S over D.
0:54:15.267 –> 0:54:17.64 And this is strictly better than what we got
0:54:17.64 –> 0:54:18.693 with the dr. Learner.
0:54:19.963 –> 0:54:21.03 (clears throat)
0:54:21.03 –> 0:54:23.07 You can do the same game where you see sort
0:54:23.07 –> 0:54:26.63 of when the Oracle rate is achieved here, it’s achieved.
0:54:26.63 –> 0:54:29.4 If the average smoothness of the nuisance functions
0:54:29.4 –> 0:54:31.39 is greater than D over four.
0:54:31.39 –> 0:54:34.14 And then here, the inflation factor is also changing.
0:54:34.14 –> 0:54:35.43 So before we had,
0:54:35.43 –> 0:54:38.42 we needed the smoothness to be greater than D over two,
0:54:38.42 –> 0:54:39.93 over one plus D over gamma.
0:54:39.93 –> 0:54:43.233 Now we have D over four over one plus or two gamma.
0:54:44.68 –> 0:54:45.99 So this is a weaker condition.
0:54:45.99 –> 0:54:48.18 So this is telling us that there are settings
0:54:48.18 –> 0:54:51.53 where that dr. Lerner is not Oracle efficient,
0:54:51.53 –> 0:54:53.37 but there exists an estimator, which is,
0:54:53.37 –> 0:54:56.2 and it looks like this estimator
0:54:56.2 –> 0:54:57.033 I had described here,
0:54:57.033 –> 0:54:58.52 this regression on residuals thing.
0:55:01.77 –> 0:55:02.603 So that’s the story.
0:55:02.603 –> 0:55:03.436 You can actually,
0:55:03.436 –> 0:55:04.78 you can actually beat this dr. Lerner.
0:55:04.78 –> 0:55:08.28 And now the question is, okay, what happens?
0:55:08.28 –> 0:55:09.27 One, what happens
0:55:09.27 –> 0:55:11.28 when we’re not achieving the Oracle rate here,
0:55:11.28 –> 0:55:13.03 can you still do better?
0:55:13.03 –> 0:55:16.393 A second question is can anything, yeah.
0:55:18.66 –> 0:55:20.12 Can anything achieve the Oracle rate
0:55:20.12 –> 0:55:22.26 under weaker conditions than this?
0:55:22.26 –> 0:55:24.97 And so I haven’t proved anything about this yet.
0:55:24.97 –> 0:55:27.853 It turns out to be somewhat difficult,
0:55:29.16 –> 0:55:32.9 but I conjecture that this, this condition is mini max.
0:55:32.9 –> 0:55:34.41 So I don’t think any,
0:55:34.41 –> 0:55:36.16 any estimator could ever be Oracle efficient
0:55:36.16 –> 0:55:40.25 under weaker conditions than what this estimator is.
0:55:40.25 –> 0:55:41.78 So this is just a picture of the results again.
0:55:41.78 –> 0:55:44.59 So here’s, it’s the same setting as before here,
0:55:44.59 –> 0:55:48.057 we have the plugin estimator that dr. Learner.
0:55:48.057 –> 0:55:51.4 And here’s what we get with this.
0:55:51.4 –> 0:55:52.56 I call it the LPR learner.
0:55:52.56 –> 0:55:55.04 It’s a local polynomial version of the, our learner.
0:55:55.04 –> 0:55:57.67 And so we’re, actually getting quite a bit smaller rates.
0:55:57.67 –> 0:56:01.64 We’re hitting the Oracle rate under Meeker conditions
0:56:01.64 –> 0:56:03.47 on the smoothness.
0:56:03.47 –> 0:56:08.304 Now, the question is whether we can fill this gap
anymore,
0:56:08.304 –> 0:56:09.137 and this is unknown.
0:56:09.137 –> 0:56:11.883 This is one of the open questions in causal inference.
0:56:14.07 –> 0:56:17.99 So yeah, I think in the interest of time,
0:56:17.99 –> 0:56:20.81 I’ll skip to the discussion section here.
0:56:20.81 –> 0:56:22.26 We can actually fill the gap a little bit
0:56:22.26 –> 0:56:26.06 with some extra, extra tuning.
0:56:29.55 –> 0:56:30.545 Yeah.
So this last part is really about just pushing the limits, trying to figure out what the best possible performance is.

Okay.

So just to wrap things up, right we gave some new results here that let you be very flexible with the kinds of methods that you want to use. They do a good job of exploiting this Cate structure when it’s there and don’t lose much when it’s not. So we have this nice model, free Arab bound. We also kind of for free to get this nice general Oracle inequality did some investigation of the best possible rates of convergence, the best possible mean squared error when estimating conditional effects, which again was unknown before. These are the weekend weak cause conditions that have appeared. It’s still not entirely known whether they are mini max optimal or not. So, yeah, big picture goals. We want some nice flexible tools, strong guarantees when it pushed forward, our understanding of this problem. I hope I’ve conveyed that there are lots of fun open problems here to work out with important practical implications. Here’s just a list of them. I’d be happy to talk more with people at any point,
feel free to email me a big part is applying these methods in real problems. And yeah, I should stop here, but feel free to email the papers on archive here. I’d be happy to hear people’s thoughts. Yeah. Thanks again for inviting me. It was fun. Yeah. Thanks Edward. That’s a very nice talk and I think we’re hitting the hour, but I want to see in the audience if we have any questions. Huh. All right. If not, I do have one final question if that’s okay. Yeah, of course. And so I think there is a hosted literature on flexible outcome modeling to estimate conditional average causal effect, especially those baits and non-parametric tree models (laughs) between those sweats based in non-parametric tree models versus the plug-in estimator?
We compared in a simulation study here?

- Yeah.

I think of them as really just versions of that plugin estimator that use a different regression procedure.

There may be ways to tune plugins to try and exploit this special structure of the Cate. But if you're really just looking at the regression functions individually, I think these would be susceptible to the same kinds of issues that we see with the plugin.

That's a good one.

So I want to see if there's any further questions from the audience to dr. Kennedy.

I was just wondering if you could speak a little more, why the standard like naming orthogonality results or can it be applicable in this setup? Yeah. (clears throat) Yeah.

That's a great question. So one way to say it is that these effects, these conditional effects are not Pathwise differentiable. And so these kinds of there's some distinction between naming orthogonality and pathways differentiability.
but maybe we can think about them as being roughly the same for now.

So yeah, all the standards in my parametric theory breaks down here because of this lack of pathways differentiability so the, all the efficiency bounds that we know and love don’t apply, but it turns out that there’s some kind of analogous version of this that works for these things. I think of them as like infinite dimensional functional. So instead of like the ate, which is just a number, this is like a curve, but it has the same kinds of like functional structure in the sense that it’s combining regression functions or our propensity scores in some way. And we don’t care about the individual components. We care about their combination.

So yeah, the standard stuff doesn’t work just because it’s, we’re outside of this route in Virginia, roughly, but there are, yeah, there’s analogous structure and there’s tons of important work to be done, sort of formalizing this and extending that’s a little vague, but hopefully that. All right. So any further questions? Thanks again. And yeah, if any questions come up, feel free to email. Yeah. If not,
I'll let smoke unless that doctors can be again.
And I'm sure he'll be happy
to answer your questions offline.
So thanks everyone.
I'll see you.
We'll see you next week.
- Thanks a lot.