WEBVTT

1 00:00:00.000 --> 00:00:02.610 - So let's get started.

2 00:00:02.610 --> 00:00:04.130 Welcome everyone.

3 00:00:04.130 --> 00:00:07.360 It is my great pleasure to introduce our speaker today,

4 00:00:07.360 --> 00:00:11.030 Dr. Edward Kennedy, who is an assistant professor

5 00:00:11.030 --> 00:00:13.150 at the Department of Statistics and Data Science

6 00:00:13.150 --> 00:00:14.763 at Carnegie Mellon University.

7 00:00:15.790 --> 00:00:17.850 Dr. Kennedy got his MA in statistics

8 00:00:17.850 --> 00:00:21.460 and PhD in biostatistics from University of Pennsylvania.

9 00:00:21.460 --> 00:00:24.210 He's an expert in methods for causal inference,

10 00:00:24.210 --> 00:00:25.690 missing data and machine learning,

11 00:00:25.690 --> 00:00:27.360 especially in settings involving

12 00:00:27.360 --> 00:00:30.810 high dimensional and complex data structures.

13 00:00:30.810 --> 00:00:33.680 He has also been collaborating on statistical applications

14 00:00:33.680 --> 00:00:36.270 in criminal justice, health services,

15 00:00:36.270 --> 00:00:38.260 medicine and public policy.

16 00:00:38.260 --> 00:00:39.970 Today's going to share with us his recent work

17 00:00:39.970 --> 00:00:43.430 in the space of heterogeneous causal effect estimation.

18 00:00:43.430 --> 00:00:45.600 Welcome Edward, the floor is yours.

19 00:00:45.600 --> 00:00:46.850 - [Edward] Thanks so much, (clears throat)

20 00:00:46.850 --> 00:00:48.500 yeah, thanks for the invitation.

21 00:00:48.500 --> 00:00:51.010 I'm happy to talk to everyone today about this work

22 00:00:51.010 --> 00:00:54.768 I've been thinking about for the last year or so.

23 00:00:54.768 --> 00:00:57.030 Sort of excited about it.

24 00:00:57.030 --> 00:00:59.400 Yeah, so it's all about doubly robust estimation

25 00:00:59.400 --> 00:01:01.150 of heterogeneous treatment effects.

26 00:01:03.320 --> 00:01:04.270 Maybe before I start,

27 00:01:04.270 --> 00:01:06.730 I don't know what the standard approach is for questions,

28 00:01:06.730 --> 00:01:07.910 but I'd be more than happy to take

29 00:01:07.910 --> 00:01:09.647 any questions throughout the talk

30 00:01:09.647 --> 00:01:13.030 and I can always sort of adapt and focus more

31 00:01:13.030 --> 00:01:13.863 on different parts of the room,

32 00:01:13.863 --> 00:01:15.433 what people are interested in.

33 00:01:17.300 --> 00:01:20.780 I'm also trying to get used to using Zoom,

34 00:01:20.780 --> 00:01:22.740 I've been teaching this big lecture course

35 00:01:22.740 --> 00:01:25.760 so I think I can keep an eye on the chat box too

36 00:01:25.760 --> 00:01:26.900 if people have questions that way,

37 00:01:26.900 --> 00:01:28.700 feel free to just type something in.

38 00:01:29.810 --> 00:01:30.643 Okay.

39 00:01:30.643 --> 00:01:34.180 So yeah, this is sort

40 00:01:34.180 --> 00:01:35.640 of standard problem non-causal inference

41 00:01:35.640 --> 00:01:37.240 but I'll give some introduction.

42 00:01:38.310 --> 00:01:41.240 The kind of classical target that people go after

43 00:01:41.240 --> 00:01:43.840 in causal inference problems is what's

44 00:01:43.840 --> 00:01:45.820 often called the average treatment effect.

45 00:01:45.820 --> 00:01:47.880 So this tells you the mean outcome if everyone

46 00:01:47.880 --> 00:01:51.143 was treated versus if everyone was untreated, for example.

47 00:01:53.320 --> 00:01:57.420 So this is, yeah, sort of the standard target.

48 00:01:57.420 --> 00:02:01.080 We know quite a bit about estimating this parameter

49 00:02:01.080 --> 00:02:04.123 under no unmeasured confounding kinds of assumptions.

50 00:02:05.290 --> 00:02:09.690 So just as a just sort of point this out,

51 00:02:09.690 --> 00:02:11.030 so a lot of my work is sort of focused

52 00:02:11.030 --> 00:02:12.470 on the statistics of causal inference,

53 00:02:12.470 --> 00:02:14.590 how to estimate causal parameters

54 00:02:14.590 --> 00:02:17.090 well in flexible non-parametric models.

55 00:02:17.090 --> 00:02:17.923 So we know quite a bit

56 00:02:17.923 --> 00:02:20.380 about this average treatment effect parameter.

57 00:02:20.380 --> 00:02:23.190 There are still some really interesting open problems,

58 00:02:23.190 --> 00:02:24.970 even for this sort of most basic parameter,

59 00:02:24.970 --> 00:02:26.400 which I'd be happy to talk to people about,

60 00:02:26.400 --> 00:02:31.060 but this is just one number, it's an overall summary

61 00:02:31.060 --> 00:02:33.850 of how people respond to treatment, on average.

62 00:02:33.850 --> 00:02:37.563 It can obscure potentially important heterogeneity.

63 00:02:38.470 --> 00:02:42.950 So for example, very extreme case would be where half

64 00:02:42.950 --> 00:02:44.900 the population is seeing a big benefit

65 00:02:44.900 --> 00:02:48.870 from treatment and half is seeing severe harm,

66 00:02:48.870 --> 00:02:50.320 then you would completely miss this

67 00:02:50.320 --> 00:02:52.900 by just looking at the average treatment effect.

68 00:02:52.900 --> 00:02:54.980 So this motivates going beyond this,

69 00:02:54.980 --> 00:02:57.500 maybe looking at how treatment effects can vary

70 00:02:57.500 --> 00:02:59.593 across subject characteristics.

71 00:03:01.440 --> 00:03:03.090 All right, so why should we care about this?

72 00:03:03.090 --> 00:03:05.900 Why should we care how treatment effects vary in this way?

73 00:03:05.900 --> 00:03:09.420 So often when I talk about this,

74 00:03:09.420 --> 00:03:12.460 people's minds go immediately to optimal treatment regimes,

75 00:03:12.460 --> 00:03:15.830 which is certainly an important part of this problem.

76 00:03:15.830 --> 00:03:19.190 So that means trying to find out who's benefiting

77 00:03:19.190 --> 00:03:21.840 from treatment and who is not or who's being harmed.

78 00:03:21.840 --> 00:03:23.870 And then just in developing

79 00:03:23.870 --> 00:03:25.750 a treatment policy based on this,

80 00:03:25.750 --> 00:03:27.160 where you treat the people who benefit,

81 00:03:27.160 --> 00:03:29.270 but not the people who don't.

82 00:03:29.270 --> 00:03:30.490 This is definitely an important part

83 00:03:30.490 --> 00:03:31.890 of understanding heterogeneity,

84 00:03:31.890 --> 00:03:33.280 but I don't think it's the whole story.

85 00:03:33.280 --> 00:03:36.390 So it can also be very useful just

86 00:03:36.390 --> 00:03:39.080 to understand heterogeneity from a theoretical perspective,

87 00:03:39.080 --> 00:03:40.515 just to understand the system

88 00:03:40.515 --> 00:03:43.890 that you're studying and not only that,

89 00:03:43.890 --> 00:03:48.890 but also to help inform future treatment development.

90 00:03:49.730 --> 00:03:53.170 So not just trying to optimally assign

91 00:03:53.170 --> 00:03:55.480 the current treatment that's available,

92 00:03:55.480 --> 00:03:56.850 but if you find, for example,

93 00:03:56.850 --> 00:04:00.530 that there are portions of the subject population

94 00:04:00.530 --> 00:04:02.810 that are not responding to the treatment,

95 00:04:02.810 --> 00:04:05.450 maybe you should then go off and try and develop

96 00:04:05.450 --> 00:04:09.003 a treatment that would better aim at these people.

97 00:04:09.910 --> 00:04:11.580 So lots of different reasons why you might care

98 00:04:11.580 --> 00:04:12.413 about heterogeneity,

99 00:04:12.413 --> 00:04:16.330 including devising optimal policies,

100 00:04:16.330 --> 00:04:17.633 but not just that.

101 00:04:18.890 --> 00:04:21.410 And this really plays a big role across lots

102 00:04:21.410 --> 00:04:24.130 of different fields as you can imagine.

103 00:04:24.130 --> 00:04:28.890 We might want to target policies based on how people

104 00:04:28.890 --> 00:04:32.033 are responding to a drug or a medical treatment.

105 00:04:33.000 --> 00:04:36.450 We'll see a sort of political science example here.

106 00:04:36.450 --> 00:04:39.310 So this is just a picture of what you should maybe think

107 00:04:39.310 --> 00:04:42.310 about as we're talking about this problem

108 00:04:42.310 --> 00:04:43.950 with heterogeneous treatment effects.

109 00:04:43.950 --> 00:04:45.750 So this is a timely example.

110 00:04:45.750 --> 00:04:46.840 So it's looking at the effect

111 00:04:46.840 --> 00:04:48.633 of canvassing on voter turnout.

112 00:04:49.840 --> 00:04:52.970 So this is the effect of being sort of reminded

113 00:04:52.970 --> 00:04:55.240 in a face-to-face way to vote

114 00:04:55.240 --> 00:04:56.940 that there's an election coming up

115 00:04:58.000 --> 00:05:00.370 and how this effect varies with age.

116 00:05:00.370 --> 00:05:03.500 And so I'll come back to where this plot came from

117 00:05:03.500 --> 00:05:06.630 and the exact sort of data structure and analysis,

118 00:05:06.630 --> 00:05:09.330 but just as a picture to sort of make things concrete.

119 00:05:10.910 --> 00:05:14.800 It looks like there might be some sort of positive effect

120 00:05:14.800 --> 00:05:16.490 of canvassing for younger people,

121 00:05:16.490 --> 00:05:18.020 but not for older people,

122 00:05:18.020 --> 00:05:20.630 there might be some non-linearity.

123 00:05:20.630 --> 00:05:23.470 So this might be useful for a number of reasons.

124 00:05:23.470 --> 00:05:26.770 You might not want to target the older population

125 00:05:26.770 --> 00:05:29.590 with canvassing, because it may not be doing anything,

126 00:05:29.590 --> 00:05:31.710 you might want to try and find some other way

127 00:05:31.710 --> 00:05:34.363 to increase turnout for this group right.

128 00:05:36.010 --> 00:05:37.580 Or you might just want to understand sort

129 00:05:37.580 --> 00:05:41.270 of from a psychological, sociological,

130 00:05:41.270 --> 00:05:42.573 theoretical perspective,

131 00:05:43.830 --> 00:05:46.853 what kinds of people are responding to this sort of thing?

132 00:05:48.600 --> 00:05:50.960 And so this is just one simple example

133 00:05:50.960 --> 00:05:52.493 you can keep in mind.

134 00:05:54.290 --> 00:05:57.600 So what's the state of the art for this problem?

135 00:05:57.600 --> 00:06:00.120 So in this talk, I'm going to focus

136 00:06:00.120 --> 00:06:02.180 on this conditional average treatment effect here.

137 00:06:02.180 --> 00:06:04.500 So it's the expected difference

138 00:06:04.500 --> 00:06:08.170 if people of type X were treated versus

139 00:06:08.170 --> 00:06:10.570 not expected difference in outcomes.

140 00:06:10.570 --> 00:06:13.810 This is kind of the classic or standard parameter

141 00:06:13.810 --> 00:06:15.710 that people think about now

142 00:06:15.710 --> 00:06:18.540 in the heterogeneous treatment effects literature,

143 00:06:18.540 --> 00:06:21.030 there are other options you could think

144 00:06:21.030 --> 00:06:23.780 about risk ratios, for example, if outcomes are binary.

145 00:06:24.920 --> 00:06:26.460 A lot of the methods that I talk about today

146 00:06:26.460 --> 00:06:29.530 will have analogs for these other regions,

147 00:06:29.530 --> 00:06:32.580 but there are lots of fun, open problems to explore here.

148 00:06:32.580 --> 00:06:35.110 How to characterize heterogeneous treatment effects

149 00:06:35.110 --> 00:06:38.524 when you have timeframe treatments, continuous treatments,

150 00:06:38.524 --> 00:06:40.480 of cool problems to think about.

151 00:06:40.480 --> 00:06:44.270 But anyways, this kind of effect where we have

152 00:06:44.270 --> 00:06:46.793 a binary treatment and some set of covariates,

153 00:06:47.700 --> 00:06:51.100 there's really been this proliferation of proposals

154 00:06:51.100 --> 00:06:53.340 in recent years for estimating this thing

155 00:06:53.340 --> 00:06:56.550 in a flexible way that goes beyond just fitting

156 00:06:56.550 --> 00:06:59.503 a linear model and looking at some interaction terms.

157 00:07:01.111 --> 00:07:01.944 (clears throat)

158 00:07:01.944 --> 00:07:06.944 So I guess I'll refer to the paper for a lot

159 00:07:07.030 --> 00:07:11.620 of these different papers that have thought about this.

160 00:07:11.620 --> 00:07:13.810 People have used, sort of random forests

161 00:07:13.810 --> 00:07:17.010 and tree based methods basing out

162 00:07:17.010 --> 00:07:19.710 of a regression trees, lots of different variants

163 00:07:19.710 --> 00:07:21.240 for estimating this thing.

164 00:07:21.240 --> 00:07:22.510 So there've been lots of proposals,

165 00:07:22.510 --> 00:07:24.360 lots of methods for estimating this,

166 00:07:24.360 --> 00:07:27.530 but there's some really big theoretical gaps

167 00:07:27.530 --> 00:07:28.480 in this literature.

168 00:07:29.790 --> 00:07:32.430 So one, yeah, this is especially true

169 00:07:32.430 --> 00:07:36.480 when you can imagine that this conditional effect

170 00:07:36.480 --> 00:07:37.890 might be much more simple

171 00:07:37.890 --> 00:07:40.640 or sparse or smooth than the rest

172 00:07:40.640 --> 00:07:41.920 of the data generating process.

173 00:07:41.920 --> 00:07:45.230 So you can imagine you have some

174 00:07:45.230 --> 00:07:48.660 potentially complex propensity score describing

175 00:07:48.660 --> 00:07:50.300 the mechanism by which people are treated

176 00:07:50.300 --> 00:07:51.360 based on their covariates.

177 00:07:51.360 --> 00:07:54.010 You have some underlying regression functions

178 00:07:54.010 --> 00:07:55.642 that describe this outcome process,

179 00:07:55.642 --> 00:08:00.642 how their outcomes depend on covariates,

180 00:08:00.680 --> 00:08:02.280 whether they're treated or not.

181 00:08:02.280 --> 00:08:05.400 These could be very complex and messy objects,

182 00:08:05.400 --> 00:08:08.240 but this CATE might be simpler.

183 00:08:08.240 --> 00:08:11.510 And in this kind of regime, there's very little known.

184 00:08:11.510 --> 00:08:13.500 I'll talk more about exactly what I mean

185 00:08:13.500 --> 00:08:14.623 by this in just a bit.

186 00:08:17.300 --> 00:08:18.133 So one question is,

187 00:08:18.133 --> 00:08:20.970 how do we adapt to this kind of structure?

188 00:08:20.970 --> 00:08:25.677 And there are really no strong theoretical benchmarks

189 00:08:25.677 --> 00:08:28.173 in this world in the last few years,

190 00:08:30.100 --> 00:08:32.800 which means we have all these proposals,

191 00:08:32.800 --> 00:08:35.650 which is great, but we don't know which are optimal

192 00:08:35.650 --> 00:08:39.633 or when or if they can be improved in some way.

193 00:08:41.320 --> 00:08:43.620 What's the best possible performance

194 00:08:43.620 --> 00:08:45.610 that we could ever achieve at estimating

195 00:08:45.610 --> 00:08:47.300 this quantity in the non-parametric model

196 00:08:47.300 --> 00:08:48.550 without adding assumptions?

197 00:08:48.550 --> 00:08:50.010 So these kinds of questions are basically

198 00:08:50.010 --> 00:08:52.433 entirely open in this setup.

199 00:08:53.460 --> 00:08:55.470 So the point of this work is really to try

200 00:08:55.470 --> 00:08:58.913 and push forward to answer some of these questions.

201 00:08:59.931 --> 00:09:04.931 There are two kind of big parts of this work,

202 00:09:04.970 --> 00:09:06.923 which are in a paper on archive.

203 00:09:08.560 --> 00:09:12.240 So one is just to provide more flexible estimators

204 00:09:12.240 --> 00:09:16.683 of this guy and specifically to show,

205 00:09:17.540 --> 00:09:20.193 give stronger error guarantees on estimating this.

206 00:09:22.575 --> 00:09:26.930 So that we can use a really diverse set of methods

207 00:09:26.930 --> 00:09:29.240 for estimating this thing in a doubly robust way

208 00:09:29.240 --> 00:09:31.500 and still have some rigorous guarantees

209 00:09:31.500 --> 00:09:33.990 about how well we're doing.

210 00:09:33.990 --> 00:09:35.180 So that part is more practical.

211 00:09:35.180 --> 00:09:37.300 It's more about giving a method

212 00:09:37.300 --> 00:09:38.470 that people can actually implement

213 00:09:38.470 --> 00:09:40.620 and practice that's pretty straight forward,

214 00:09:40.620 --> 00:09:43.690 it looks like a two stage progression procedure

215 00:09:43.690 --> 00:09:46.339 and being able to say something about this

216 00:09:46.339 --> 00:09:51.339 that's model free and and agnostic about both

217 00:09:51.630 --> 00:09:53.160 the underlying data generating process

218 00:09:53.160 --> 00:09:56.990 and what methods we're using to construct the estimator.

219 00:09:56.990 --> 00:09:59.320 This was lacking in the previous literature.

220 00:09:59.320 --> 00:10:02.910 So that's one side of this work, which is more practical.

221 00:10:02.910 --> 00:10:04.913 I think I'll focus more on that today,

222 00:10:06.340 --> 00:10:08.510 but we can always adapt as we go,

223 00:10:08.510 --> 00:10:10.160 if people are interested in other stuff.

224 00:10:10.160 --> 00:10:12.310 I'm also going to talk a bit about an analysis of this,

225 00:10:12.310 --> 00:10:15.013 just to show you sort of how it would work in practice.

226 00:10:16.340 --> 00:10:17.410 So that's one part of this work.

227 00:10:17.410 --> 00:10:21.640 The second part is more theoretical and it says,

228 00:10:21.640 --> 00:10:23.860 so I don't want to just sort of construct

229 00:10:23.860 --> 00:10:27.310 an estimator that has the nice error guarantees,

230 00:10:27.310 --> 00:10:29.230 but I want to try and figure out what’s

231 00:10:29.230 --> 00:10:31.220 the best possible performance I could ever get

232 00:10:31.220 --> 00:10:33.723 at estimating these heterogeneous effects.

233 00:10:35.640 --> 00:10:38.700 This turns out to be a really hard problem

234 00:10:38.700 --> 00:10:40.233 with a lot of nuance,

235 00:10:41.770 --> 00:10:43.270 but that’s sort of the second part

236 00:10:43.270 --> 00:10:45.840 of the talk which maybe is a little tackle

237 00:10:45.840 --> 00:10:48.363 in a bit less time.

238 00:10:49.920 --> 00:10:50.940 So that’s kind of big picture.

239 00:10:50.940 --> 00:10:52.853 I like to give the punchline of the talk at the start,

240 00:10:52.853 --> 00:10:56.453 just so you have an idea of what I’m going to be covering.

241 00:10:57.390 --> 00:11:01.340 And yeah, so now let’s go into some details.

242 00:11:01.340 --> 00:11:03.320 So we’re going to think about this sort

243 00:11:03.320 --> 00:11:05.560 of classic causal inference data structure,

244 00:11:05.560 --> 00:11:09.800 where we have n iid observations, we have covariates X,

245 00:11:09.800 --> 00:11:13.710 which are D dimensional, binary treatment for now,

246 00:11:13.710 --> 00:11:16.010 all the methods that I’ll talk about will work

247 00:11:16.930 --> 00:11:20.610 without any extra work in the discrete treatment setting

248 00:11:20.610 --> 00:11:23.290 if we have multiple values.

249 00:11:23.290 --> 00:11:24.380 The continuous treatment setting

250 00:11:24.380 --> 00:11:26.730 is more difficult it turns out.

251 00:11:26.730 --> 00:11:29.293 And some outcome Y that we care about.

252 00:11:30.520 --> 00:11:33.400 All right, so there are a couple of characters

253 00:11:33.400 --> 00:11:36.670 in this talk that will play really important roles.

254 00:11:36.670 --> 00:11:39.390 So we’ll have some special notation for them.

255 00:11:39.390 --> 00:11:41.750 So PI of X, this is the propensity score.

256 00:11:41.750 --> 00:11:44.020 This is the chance of being treated,

257 00:11:44.020 --> 00:11:45.750 given your covariates.

258 00:11:47.020 --> 00:11:48.670 So some people might be more or less likely

259 00:11:48.670 --> 00:11:52.363 to be treated depending on their baseline covariates, X.

260 00:11:53.760 --> 00:11:56.890 Muse of a, this will be an outcome regression function.

261 00:11:56.890 --> 00:11:59.610 So it’s your expected outcome given your covariates

262 00:11:59.610 --> 00:12:01.940 and given your treatment level.

263 00:12:01.940 --> 00:12:04.450 And then we’ll also later on in the talk use this ada,

264 00:12:04.450 --> 00:12:06.160 which is just the marginal outcome regression.

265 00:12:06.160 --> 00:12:07.940 So without thinking about treatment,

266 00:12:07.940 --> 00:12:11.823 just how the outcome varies on average as a function of X.

267 00:12:13.929 --> 00:12:16.280 And so those are the three main characters in this talk,

268 00:12:16.280 --> 00:12:18.400 we’ll be using them throughout.

269 00:12:18.400 --> 00:12:20.990 So under these standard causal assumptions

270 00:12:20.990 --> 00:12:23.463 of consistency, positivity, exchangeability,

271 00:12:24.490 --> 00:12:26.670 there’s a really amazing group

272 00:12:26.670 --> 00:12:31.200 at Yale that are focused on dropping these assumptions.

273 00:12:31.200 --> 00:12:33.840 So lots of cool work to be done there,

274 00:12:33.840 --> 00:12:35.830 but we’re going to be using them today.

275 00:12:35.830 --> 00:12:38.390 So consistency, we’re roughly thinking

276 00:12:38.390 --> 00:12:39.840 this means there’s no interference,

277 00:12:39.840 --> 00:12:41.580 this is a big problem in causal inference,

278 00:12:41.580 --> 00:12:43.152 but we’re going to say

279 00:12:43.152 --> 00:12:46.650 that my treatments can affect your outcomes, for example.

280 00:12:46.650 --> 00:12:48.010 We're going to think about the case where everyone

281 00:12:48.010 --> 00:12:50.730 has some chance at receiving treatment,

282 00:12:50.730 --> 00:12:52.440 both treatment and control,

283 00:12:52.440 --> 00:12:54.000 and then we have no unmeasured confounding.

284 00:12:54.000 --> 00:12:57.070 So we've collected enough sufficiently relevant covariates

285 00:12:57.070 --> 00:12:58.620 that once we conditioned on them,

286 00:12:58.620 --> 00:13:00.030 look within levels of the covariates,

287 00:13:00.030 --> 00:13:01.980 the treatment is as good as randomized.

288 00:13:03.480 --> 00:13:05.650 So under these three assumptions,

289 00:13:05.650 --> 00:13:08.530 this conditional effect on the left-hand side here

290 00:13:08.530 --> 00:13:10.790 can just be written as a difference in regression functions.

291 00:13:10.790 --> 00:13:12.840 It's just the difference in the regression function

292 00:13:12.840 --> 00:13:14.930 under treatment versus control,

293 00:13:14.930 --> 00:13:16.873 sort of super simple parameter right.

294 00:13:17.760 --> 00:13:19.550 So I'm going to call this thing Tau.

295 00:13:19.550 --> 00:13:22.340 This is just the regression under treatment minus

296 00:13:22.340 --> 00:13:23.790 the regression under control.

297 00:13:26.950 --> 00:13:29.100 So you might think, we know a lot about

298 00:13:29.100 --> 00:13:32.820 how to estimate regression functions nonparametrically

299 00:13:32.820 --> 00:13:35.740 they're really nice, min and max lower bounds

300 00:13:35.740 --> 00:13:39.580 that say we can't do better uniformly across the model

301 00:13:40.530 --> 00:13:43.633 without adding some assumptions or some extra structure.

302 00:13:45.530 --> 00:13:46.560 The fact that we have a difference

303 00:13:46.560 --> 00:13:47.720 in regression doesn't seem like

304 00:13:47.720 --> 00:13:49.960 it would make things more complicated

305 00:13:49.960 --> 00:13:52.570 than just the initial regression problem,

306 00:13:52.570 --> 00:13:54.650 but it turns out it really does,

307 00:13:54.650 --> 00:13:55.590 it's super interesting,

308 00:13:55.590 --> 00:13:57.060 this is one of the parts of this problem

309 00:13:57.060 --> 00:14:00.030 that I think is really fascinating.

310 00:14:00.030 --> 00:14:02.580 So just by taking a difference in regressions,

311 00:14:02.580 --> 00:14:05.540 you completely change the nature of this problem

312 00:14:05.540 --> 00:14:08.040 from the standard non-parametric regression setup.

313 00:14:10.410 --> 00:14:13.930 So let's get some intuition for why this is the case.

314 00:14:13.930 --> 00:14:17.320 So why isn't it optimal just to estimate

315 00:14:17.320 --> 00:14:18.490 the two regression functions

316 00:14:18.490 --> 00:14:20.073 and take a difference, for example?

317 00:14:20.980 --> 00:14:23.060 So let's think about a simple data generating process

318 00:14:23.060 --> 00:14:25.920 where we have just a one dimensional covariate,

319 00:14:25.920 --> 00:14:28.600 it's uniform on minus one, one,

320 00:14:28.600 --> 00:14:31.520 we have a simple step function propensity score

321 00:14:31.520 --> 00:14:32.353 and then we're going to think

322 00:14:32.353 --> 00:14:35.360 about a regression function, both under treatment

323 00:14:35.360 --> 00:14:37.200 and control that looks like some kind

324 00:14:37.200 --> 00:14:40.470 of crazy polynomial from this Gyorfi textbook,

325 00:14:40.470 --> 00:14:42.520 I'll show you a picture in just a minute.

326 00:14:44.190 --> 00:14:47.110 The important thing about this polynomial

327 00:14:47.110 --> 00:14:50.410 is that it's non-smooth, it has a jump,

328 00:14:50.410 --> 00:14:55.410 has some kinks in it and so it will be hard to estimate,

329 00:14:55.720 --> 00:14:59.710 in general, but we're taking both

330 00:14:59.710 --> 00:15:01.210 the regression function under treatment

331 00:15:01.210 --> 00:15:03.160 and the regression function under control

332 00:15:03.160 --> 00:15:05.560 to be equal, they're equal to this same hard

333 00:15:05.560 --> 00:15:07.040 to estimate polynomial function.

334 00:15:07.040 --> 00:15:09.610 And so that means the difference is really simple,

335 00:15:09.610 --> 00:15:12.380 it's just zero, it's the simplest conditional effect

336 00:15:12.380 --> 00:15:15.320 you can imagine, not only constant, but zero.

337 00:15:15.320 --> 00:15:18.210 You can imagine this probably happens a lot in practice

338 00:15:18.210 --> 00:15:21.810 where we have treatments that are not extremely effective

339 00:15:21.810 --> 00:15:23.763 for everyone in some complicated way.

340 00:15:26.406 --> 00:15:29.280 So the simplest way you would estimate

341 00:15:29.280 --> 00:15:31.840 this conditional effect is just take an estimate

342 00:15:31.840 --> 00:15:34.850 of the two regression functions and take a difference.

343 00:15:34.850 --> 00:15:36.950 Sometimes I'll call this plugin estimator.

344 00:15:38.040 --> 00:15:40.610 There's this paper by Kunzel and colleagues,

345 00:15:40.610 --> 00:15:41.710 call it the T-learner.

346 00:15:43.340 --> 00:15:45.700 So for example, we can use smoothing splines,

347 00:15:45.700 --> 00:15:49.430 estimate the two regression functions and take a difference.

348 00:15:49.430 --> 00:15:52.250 And maybe you can already see what's going to go wrong here.

349 00:15:52.250 --> 00:15:54.250 So these individual regression functions

350 00:15:54.250 --> 00:15:56.653 by themselves are really hard to estimate.

351 00:15:57.530 --> 00:16:00.970 They have jumps and kinks, they're messy functions

352 00:16:00.970 --> 00:16:03.420 And so when we try and estimate these

353 00:16:03.420 --> 00:16:05.440 with smoothing splines, for example,

354 00:16:05.440 --> 00:16:08.340 we're going to get really complicated estimates

355 00:16:08.340 --> 00:16:10.550 that have some bumps, It's hard to choose

356 00:16:10.550 --> 00:16:13.790 the right tuning parameter, but even if we do,

357 00:16:13.790 --> 00:16:16.080 we're inheriting the sort of complexity

358 00:16:16.080 --> 00:16:18.250 of the individual regression functions.

359 00:16:18.250 --> 00:16:19.180 When we take the difference,

360 00:16:19.180 --> 00:16:20.100 we're going to see something

361 00:16:20.100 --> 00:16:22.470 that is equally complex here

362 00:16:22.470 --> 00:16:24.830 and so it's not doing a good job of exploiting

363 00:16:24.830 --> 00:16:27.393 this simple structure in the conditional effect.

364 00:16:29.510 --> 00:16:33.350 This is sort of analogous to this intuition

365 00:16:33.350 --> 00:16:36.420 that people have that interaction terms might

366 00:16:36.420 --> 00:16:41.040 be smaller or less worrisome than sort

367 00:16:41.040 --> 00:16:43.410 of main effects in a regression model.

368 00:16:43.410 --> 00:16:45.157 Or you can think of the muse as sort of main effects

369 00:16:45.157 --> 00:16:47.313 and the differences as like an interaction.

370 00:16:48.890 --> 00:16:50.700 So here's a picture of this data

371 00:16:50.700 --> 00:16:53.070 in the simple motivating example.

372 00:16:53.070 --> 00:16:54.700 So we've got treated people on the left

373 00:16:54.700 --> 00:16:56.620 and untreated people on the right

374 00:16:56.620 --> 00:17:00.350 and this gray line is the true, that messy,

375 00:17:00.350 --> 00:17:02.750 weird polynomial function that we're thinking about.

376 00:17:02.750 --> 00:17:05.810 So here's a jump and there's a couple

377 00:17:05.810 --> 00:17:08.700 of kinks here and there's confounding.

378 00:17:08.700 --> 00:17:12.822 So treated people are more likely to have larger Xs,

379 00:17:12.822 --> 00:17:15.540 untreated people are more likely to have smaller Xs.

380 00:17:15.540 --> 00:17:18.530 So what happens here is the function is sort

381 00:17:18.530 --> 00:17:21.870 of a bit easier to estimate on the right side.

382 00:17:21.870 --> 00:17:24.140 And so for treated people, we're going to take a sort

383 00:17:24.140 --> 00:17:27.500 of larger bandwidth, get a smoother function.

384 00:17:27.500 --> 00:17:29.700 For untreated people, it's harder to estimate

385 00:17:29.700 --> 00:17:31.490 on the left side and so we're going to need

386 00:17:31.490 --> 00:17:34.370 a small bandwidth to try and capture this jump,

387 00:17:34.370 --> 00:17:36.123 for example, this discontinuity.

388 00:17:37.760 --> 00:17:40.010 And so what's going to happen is when you take a difference

389 00:17:40.010 --> 00:17:42.360 of these two regression estimates, these black lines

390 00:17:42.360 --> 00:17:45.560 are just the standard smoothing that spline estimates

391 00:17:45.560 --> 00:17:48.260 that you're getting are with one line of code,

392 00:17:48.260 --> 00:17:50.060 using the default bandwidth choices.

393 00:17:50.060 --> 00:17:50.893 When you take a difference,

394 00:17:50.893 --> 00:17:51.726 you're going to get something

395 00:17:51.726 --> 00:17:54.680 that's very complex and messy and it's not doing

396 00:17:54.680 --> 00:17:57.547 a good job of recognizing that the regression functions

397 00:17:57.547 --> 00:17:59.713 are the same under treatment and control.

398 00:18:02.520 --> 00:18:04.360 So what else could we do?

399 00:18:04.360 --> 00:18:06.440 This maybe points to this fact that

400 00:18:06.440 --> 00:18:08.910 the plugin estimator breaks,

401 00:18:08.910 --> 00:18:10.630 it doesn't do a good job of exploiting a structure,

402 00:18:10.630 --> 00:18:12.600 but what other options do we have?

403 00:18:12.600 --> 00:18:15.480 So let's say that we knew the propensity scores.

404 00:18:15.480 --> 00:18:18.680 So for just simplicity, say we were in a trial,

405 00:18:18.680 --> 00:18:20.840 for example, an experiment,

406 00:18:20.840 --> 00:18:22.690 where we randomized everyone to treat them

407 00:18:22.690 --> 00:18:25.550 with some probability that we knew.

408 00:18:25.550 --> 00:18:28.180 In that case, we could construct a pseudo outcome,

409 00:18:28.180 --> 00:18:30.960 which is just like an inverse probability weighted outcome,

410 00:18:30.960 --> 00:18:34.700 which has exactly the right conditional expectation,

411 00:18:34.700 --> 00:18:36.640 its conditional expectation is exactly equal

412 00:18:36.640 --> 00:18:38.620 to that conditional effect.

413 00:18:38.620 --> 00:18:40.960 And so when you did a non-parametric regression

414 00:18:40.960 --> 00:18:42.990 of the pseudo outcome on X,

415 00:18:42.990 --> 00:18:45.390 it would be like doing an oracle regression

416 00:18:45.390 --> 00:18:47.210 of the true difference in potential outcomes,

417 00:18:47.210 --> 00:18:49.610 it has exactly the same conditional expectation.

418 00:18:50.490 --> 00:18:53.060 And so this sort of turns this hard problem

419 00:18:53.060 --> 00:18:56.300 into a standard non-parametric regression problem.

420 00:18:56.300 --> 00:18:57.810 Now this isn't a special case where we knew

421 00:18:57.810 --> 00:18:59.290 the propensity scores for the rest

422 00:18:59.290 --> 00:19:00.660 of the talk we're gonna think about what happens

423 00:19:00.660 --> 00:19:02.983 when we don't know these, what can we say?

424 00:19:03.940 --> 00:19:06.590 So here's just a picture of what we get in the setup.

425 00:19:07.770 --> 00:19:10.270 So this red line is this really messy plug in estimator

426 00:19:10.270 --> 00:19:12.650 that we get that's just inheriting that complexity

427 00:19:12.650 --> 00:19:15.060 of estimating the individual regression functions

428 00:19:15.060 --> 00:19:18.640 and then these black and blue lines are IPW

429 00:19:18.640 --> 00:19:21.555 and doubly robust versions that exploit

430 00:19:21.555 --> 00:19:25.280 this underlying smoothness and simplicity

431 00:19:25.280 --> 00:19:28.250 of the heterogeneous effects, the conditional effects.

432 00:19:31.690 --> 00:19:33.500 So this is just a motivating example

433 00:19:33.500 --> 00:19:36.573 to help us get some intuition for what's going on here.

434 00:19:38.740 --> 00:19:40.790 So these results are sort of standard in this problem,

435 00:19:40.790 --> 00:19:43.340 we'll come back to some simulations later on.

436 00:19:43.340 --> 00:19:47.630 And so now our goal is going to study the error

437 00:19:47.630 --> 00:19:51.430 of the sort of inverse weighted kind of procedure,

438 00:19:51.430 --> 00:19:53.490 but a doubly robust version.

439 00:19:53.490 --> 00:19:57.430 We're going to give some new model free error guarantees,

440 00:19:57.430 --> 00:19:59.216 which let us use very flexible methods

441 00:19:59.216 --> 00:20:03.260 and it turns out we'll actually get better areas

442 00:20:03.260 --> 00:20:07.770 than what were achieved previously in literature,

443 00:20:07.770 --> 00:20:11.560 even when focusing specifically on some particular method.

444 00:20:11.560 --> 00:20:12.790 And then again, we're going to see,

445 00:20:12.790 --> 00:20:14.990 how well can we actually do estimating

446 00:20:14.990 --> 00:20:17.963 this conditional effect in this problem.

447 00:20:20.710 --> 00:20:22.810 Might be a good place to pause

448 00:20:22.810 --> 00:20:24.793 and see if people have any questions.

449 00:20:33.000 --> 00:20:33.942 Okay.

450 00:20:33.942 --> 00:20:34.775 (clears throat)

451 00:20:34.775 --> 00:20:37.110 Feel free to shout out any questions

452 00:20:37.110 --> 00:20:39.223 or stick them on the chat if any come up.

453 00:20:41.910 --> 00:20:44.682 So we're going to start by thinking about

454 00:20:44.682 --> 00:20:48.110 a pretty simple two-stage doubly robust estimator,

455 00:20:48.110 --> 00:20:49.740 which I'm going to call the DR-learner,

456 00:20:49.740 --> 00:20:52.940 this is following this nomenclature that's become kind

457 00:20:52.940 --> 00:20:55.970 of common in the heterogeneous effects literature

458 00:20:56.950 --> 00:20:59.000 where we have letters and then a learner.

459 00:21:00.660 --> 00:21:02.100 So I'm calling this the DR-Learner,

460 00:21:02.100 --> 00:21:04.110 but this is not a new procedure,

461 00:21:04.110 --> 00:21:05.680 but the version that I'm going to analyze

462 00:21:05.680 --> 00:21:08.440 has some variances, but it was actually first proposed

463 00:21:08.440 --> 00:21:12.870 by Mike Vanderlande in 2013, was used in 2016

464 00:21:12.870 --> 00:21:15.900 by Alex Lucca and Mark Vanderlande.

465 00:21:15.900 --> 00:21:16.733 So they proposed this,

466 00:21:16.733 --> 00:21:18.783 but they didn't give specific error bounds.

467 00:21:20.705 --> 00:21:22.740 I think relatively few people know

468 00:21:22.740 --> 00:21:25.170 about these earlier papers because this approach

469 00:21:25.170 --> 00:21:28.070 was then sort of rediscovered in various ways

470 00:21:28.070 --> 00:21:30.350 after that in the following years,

471 00:21:30.350 --> 00:21:32.450 typically in these later versions,

472 00:21:32.450 --> 00:21:34.850 people use very specific methods for estimating,

473 00:21:37.230 --> 00:21:38.290 for constructing the estimator,

474 00:21:38.290 --> 00:21:41.250 which I'll talk about in detail in just a minute,

475 00:21:41.250 --> 00:21:43.620 for example, using kernel kind of methods,

476 00:21:43.620 --> 00:21:47.778 Local polinomials and this paper used

477 00:21:47.778 --> 00:21:50.333 a sort of series or spline and regression.

478 00:21:51.840 --> 00:21:52.819 So.

479 00:21:52.819 --> 00:21:53.652 (clears throat)

480 00:21:53.652 --> 00:21:56.380 These papers are nice ways

481 00:21:56.380 --> 00:21:57.660 of doing doubly robust estimation,

482 00:21:57.660 --> 00:22:00.340 but they had a couple of drawbacks,

483 00:22:00.340 --> 00:22:03.180 which we're going to try and build on in this work.

484 00:22:03.180 --> 00:22:05.440 So one is, we're going to try not to commit

485 00:22:05.440 --> 00:22:07.040 to using any particular methods.

486 00:22:07.040 --> 00:22:10.530 We're going to see what we can say about error guarantees,

487 00:22:10.530 --> 00:22:12.823 just for generic regression procedures.

488 00:22:14.500 --> 00:22:16.170 And then we're going to see

489 00:22:16.170 --> 00:22:19.440 if we can actually weaken the sort of assumptions

490 00:22:19.440 --> 00:22:22.440 that we need to get oracle type behavior.

491 00:22:22.440 --> 00:22:25.020 So the behavior of an estimator that we would see

492 00:22:25.020 --> 00:22:27.610 if we actually observed the potential outcomes

493 00:22:27.610 --> 00:22:29.180 and it turns out we'll be able to do this,

494 00:22:29.180 --> 00:22:32.133 even though we're not committing to particular methods.

495 00:22:33.610 --> 00:22:35.310 There's also a really nice paper by Foster

496 00:22:35.310 --> 00:22:37.600 and Syrgkanis from last year,

497 00:22:37.600 --> 00:22:41.300 which also considered a version of this DR-learner

498 00:22:41.300 --> 00:22:44.054 and they had some really nice model agnostic results,

499 00:22:44.054 --> 00:22:45.720 but they weren't doubly robust.

500 00:22:45.720 --> 00:22:48.120 So, in this work we're going to try

501 00:22:48.120 --> 00:22:50.953 and doubly robust defy these these results.

502 00:22:53.510 --> 00:22:57.080 So that's the sort of background and an overview.

503 00:22:57.080 --> 00:23:00.550 So let's think about what this estimator is actually doing.

504 00:23:00.550 --> 00:23:02.900 So here's the picture of this,

505 00:23:02.900 --> 00:23:05.010 what I'm calling the DR-learner.

506 00:23:05.010 --> 00:23:07.810 So we're going to do some interesting sample splitting

507 00:23:07.810 --> 00:23:10.490 here and later where we split our sample

508 00:23:10.490 --> 00:23:12.820 in the three different groups.

509 00:23:12.820 --> 00:23:15.880 So one's going to be used for nuisance training

510 00:23:15.880 --> 00:23:17.680 for estimating the propensity score.

511 00:23:19.320 --> 00:23:20.880 And then I'm also going to estimate

512 00:23:20.880 --> 00:23:25.880 the regression functions, but in a separate fold.

513 00:23:25.910 --> 00:23:28.760 So I'm separately estimating my propensity score

514 00:23:28.760 --> 00:23:30.360 and regression functions.

515 00:23:30.360 --> 00:23:35.220 This turns out to not be super crucial for this approach.

516 00:23:35.220 --> 00:23:37.430 It actually is crucial for something I'll talk

517 00:23:37.430 --> 00:23:38.920 about later in the talk,

518 00:23:38.920 --> 00:23:40.970 this is just to give a nicer error bound.

519 00:23:42.530 --> 00:23:45.200 So the first stage is we estimate these nuisance functions,

520 00:23:45.200 --> 00:23:47.910 the propensity scores and the regressions.

521 00:23:47.910 --> 00:23:52.620 And then we go to this new data that we haven't seen yet,

522 00:23:52.620 --> 00:23:55.730 our third fold of split data

523 00:23:55.730 --> 00:23:58.110 and we construct a pseudo outcome.

524 00:23:58.110 --> 00:24:01.020 Pseudo outcome looks like this, it's just some combination,

525 00:24:01.020 --> 00:24:04.320 it's like an inverse probability weighted residual term

526 00:24:04.320 --> 00:24:06.820 plus something like the plug-in estimator

527 00:24:06.820 --> 00:24:08.810 of the conditional effect.

528 00:24:08.810 --> 00:24:11.790 So it's just some function of the propensity score estimates

529 00:24:11.790 --> 00:24:13.993 and the regression estimates.

530 00:24:15.020 --> 00:24:17.079 If you've used doubly robust estimators

531 00:24:17.079 --> 00:24:19.160 before you'll recognize this as what

532 00:24:19.160 --> 00:24:21.450 we average when we construct

533 00:24:21.450 --> 00:24:24.100 a usual doubly robust estimator

534 00:24:24.100 --> 00:24:25.460 of the average treatment effect.

535 00:24:25.460 --> 00:24:27.720 And so intuitively instead of averaging this year,

536 00:24:27.720 --> 00:24:30.040 we're just going to regress it on covariates,

537 00:24:30.040 --> 00:24:32.700 that's exactly how this procedure works.

538 00:24:32.700 --> 00:24:35.700 So it's pretty simple, construct the pseudo outcome,

539 00:24:35.700 --> 00:24:38.600 which we typically would average estimate the ate,

540 00:24:38.600 --> 00:24:40.550 now, we're just going to do a regression

541 00:24:40.550 --> 00:24:43.423 of this thing on covariates in our third sample.

542 00:24:44.730 --> 00:24:46.940 So we can write our estimator this way.

543 00:24:46.940 --> 00:24:49.350 This e hat in notation just means

544 00:24:49.350 --> 00:24:51.833 some generic regression estimator.

545 00:24:52.983 --> 00:24:54.550 So one of the crucial points in this work,

546 00:24:54.550 --> 00:24:57.180 so I'm not going to, I want to see what I can say

547 00:24:57.180 --> 00:25:00.930 about the error of this estimator without committing

548 00:25:00.930 --> 00:25:02.030 to a particular estimator.

549 00:25:02.030 --> 00:25:04.780 So if you want to use random forests in that last stage,

550 00:25:04.780 --> 00:25:06.840 I want to be able to tell you what kind

551 00:25:06.840 --> 00:25:08.840 of error to expect or if you want

552 00:25:08.840 --> 00:25:09.800 to use linear regression

553 00:25:09.800 --> 00:25:13.110 or whatever procedure you like,

554 00:25:13.110 --> 00:25:15.620 the goal would be to give you some nice error guarantee.

555 00:25:15.620 --> 00:25:17.600 So (indistinct), and you should think of it as just

556 00:25:17.600 --> 00:25:19.350 your favorite regression estimator.

557 00:25:20.683 --> 00:25:22.220 So we take the suit outcome,

558 00:25:22.220 --> 00:25:24.730 we regress it on covariates, super simple,

559 00:25:24.730 --> 00:25:26.590 just create a new column in your dataset,

560 00:25:26.590 --> 00:25:28.200 which looks like this pseudo outcome.

561 00:25:28.200 --> 00:25:30.360 And then treat that as the outcome

562 00:25:30.360 --> 00:25:31.960 in your second stage regression.

563 00:25:35.140 --> 00:25:38.540 So here we're going to get let's say we split

564 00:25:38.540 --> 00:25:41.810 our sample into half for the second stage regression,

565 00:25:41.810 --> 00:25:43.380 we would get an over two kind

566 00:25:43.380 --> 00:25:45.530 of we'd be using half our sample

567 00:25:45.530 --> 00:25:47.680 for the second stage regression.

568 00:25:47.680 --> 00:25:49.470 You can actually just swap these samples

569 00:25:49.470 --> 00:25:53.860 in the you'll get back the full sample size errors.

570 00:25:53.860 --> 00:25:55.940 So it would be as if you had used

571 00:25:55.940 --> 00:25:58.313 the full sample size all at once.

572 00:25:59.470 --> 00:26:00.820 That's called Cross Fitting,

573 00:26:00.820 --> 00:26:03.200 it's becoming sort of popular in the last couple of years.

574 00:26:03.200 --> 00:26:05.510 So here's a schematic of what this thing is doing.

575 00:26:05.510 --> 00:26:07.220 So we split our data in the thirds,

576 00:26:07.220 --> 00:26:09.016 use one third testing

577 00:26:09.016 --> 00:26:10.020 to estimate the propensity score,

578 00:26:10.020 --> 00:26:12.370 another third to estimate the regression functions,

579 00:26:12.370 --> 00:26:14.470 we use those to construct a pseudo outcome

580 00:26:14.470 --> 00:26:15.960 and then we do a second stage regression

581 00:26:15.960 --> 00:26:18.699 of that pseudo outcome on covariates.

582 00:26:18.699 --> 00:26:21.853 So pretty easy, you can do this in three lines of code.

583 00:26:24.550 --> 00:26:26.160 Okay.

584 00:26:26.160 --> 00:26:27.510 And now our goal is to say something

585 00:26:27.510 --> 00:26:29.110 about the error of this procedure,

586 00:26:29.110 --> 00:26:30.460 being completely agnostic about

587 00:26:30.460 --> 00:26:32.020 how we estimate these propensity scores,

588 00:26:32.020 --> 00:26:34.070 that regression functions and what procedure

589 00:26:34.070 --> 00:26:36.463 we use in this third stage or second stage.

590 00:26:39.730 --> 00:26:41.810 And it turns out we can do this by exploiting

591 00:26:41.810 --> 00:26:45.700 the sample splitting can come up with a strong guarantee

592 00:26:45.700 --> 00:26:47.820 that actually gives you smaller errors than

593 00:26:47.820 --> 00:26:49.970 what appeared in the previous literature

594 00:26:49.970 --> 00:26:52.440 when people focused on specific methods.

595 00:26:52.440 --> 00:26:55.347 And the main thing is we're really exploiting

596 00:26:55.347 --> 00:26:56.563 the sample splitting.

597 00:26:57.640 --> 00:26:58.980 And then the other tool that we're using

598 00:26:58.980 --> 00:27:01.950 is we're assuming some stability condition

599 00:27:01.950 --> 00:27:03.330 on that second stage estimator,

600 00:27:03.330 --> 00:27:05.180 that's the only thing we assume here.

601 00:27:06.170 --> 00:27:10.140 It's really mild, I'll tell you what it is right now.

602 00:27:10.140 --> 00:27:13.143 So you say that regression estimator is stable,

603 00:27:14.230 --> 00:27:18.310 if when you add some constant to the outcome

604 00:27:18.310 --> 00:27:20.620 and then do a regression, you get something

605 00:27:20.620 --> 00:27:22.230 that's the same as if you do the regression

606 00:27:22.230 --> 00:27:23.580 and then add some constant.

607 00:27:24.880 --> 00:27:25.770 So it's pretty intuitive,

608 00:27:25.770 --> 00:27:27.350 if a method didn't satisfy this,

609 00:27:27.350 --> 00:27:28.540 it would be very weird

610 00:27:29.635 --> 00:27:31.620 and actually for the proof,

611 00:27:31.620 --> 00:27:34.690 we don't actually need this to be exactly equal.

612 00:27:34.690 --> 00:27:38.020 So adding a constant pre versus post regression

613 00:27:38.020 --> 00:27:39.790 shouldn't change things too much.

614 00:27:39.790 --> 00:27:41.940 You don't have to have it be exactly equal,

615 00:27:42.896 --> 00:27:44.290 it still works if it's just equal up

616 00:27:44.290 --> 00:27:48.563 to the error in the second stage regression.

617 00:27:51.900 --> 00:27:54.750 So that's the first stability condition.

618 00:27:54.750 --> 00:27:56.850 The second one is just that if you have

619 00:27:56.850 --> 00:27:59.630 two random variables with the same conditional expectation,

620 00:27:59.630 --> 00:28:00.560 then the mean squared error is going

621 00:28:00.560 --> 00:28:02.550 to be the same up to constants.

622 00:28:02.550 --> 00:28:05.240 Again, any procedure

623 00:28:05.240 --> 00:28:06.893 that didn't satisfy these two assumptions

624 00:28:06.893 --> 00:28:09.053 would be very bizarre.

625 00:28:11.200 --> 00:28:12.920 It's a very mild stability conditions.

626 00:28:12.920 --> 00:28:15.160 And that's essentially all we need.

627 00:28:15.160 --> 00:28:20.160 So now our benchmark here is going to be an oracle estimator

628 00:28:20.530 --> 00:28:23.450 that instead of doing a regression with the pseudo,

629 00:28:23.450 --> 00:28:26.280 it does a regression with the actual potential outcomes,

630 00:28:26.280 --> 00:28:27.293 Y, one, Y, zero.

631 00:28:29.600 --> 00:28:31.400 So we can think about the mean squared error

632 00:28:31.400 --> 00:28:33.510 of this estimator, so I'm using mean squared error,

633 00:28:33.510 --> 00:28:35.750 just sort of for simplicity and convention,

634 00:28:35.750 --> 00:28:37.800 you could think about translating this

635 00:28:37.800 --> 00:28:39.740 to other kinds of measures of risk.

636 00:28:39.740 --> 00:28:42.803 That would be an interesting area for future work.

637 00:28:43.770 --> 00:28:46.700 So this is the oral, our star is the Oracle

638 00:28:46.700 --> 00:28:47.930 the mean squared error.

639 00:28:47.930 --> 00:28:49.930 It's the mean squared error you'd get for estimating

640 00:28:49.930 --> 00:28:52.400 the conditional effect if you actually saw

641 00:28:52.400 --> 00:28:53.550 the potential outcomes.

642 00:28:55.240 --> 00:28:57.020 So we get this really nice, simple result,

643 00:28:57.020 --> 00:28:59.450 which says that the mean squared error

644 00:28:59.450 --> 00:29:03.010 of that DR-learner procedure that uses the pseudo outcomes,

645 00:29:03.010 --> 00:29:05.680 it just looks like the Oracle means squared error,

646 00:29:05.680 --> 00:29:08.130 plus a product of mean squared errors in estimating

647 00:29:08.130 --> 00:29:10.580 the propensity score and the regression function.

648 00:29:11.711 --> 00:29:16.390 It resembles the kind of doubly robust error results

649 00:29:16.390 --> 00:29:18.200 that you see for estimating average treatment effects,

650 00:29:18.200 --> 00:29:21.063 but now we have this for conditional effects.

651 00:29:22.770 --> 00:29:25.160 The proof technique is very different here compared

652 00:29:25.160 --> 00:29:29.030 to what is done in the average effect case.

653 00:29:29.030 --> 00:29:32.370 But the proof is actually very, very straightforward.

654 00:29:32.370 --> 00:29:35.130 It's like a page long, you can take a look in the paper,

655 00:29:35.130 --> 00:29:37.680 it's really just leaning on this sample splitting

656 00:29:37.680 --> 00:29:40.743 and then using stability in a slightly clever way.

657 00:29:42.470 --> 00:29:44.968 But the most complicated tool uses is just

658 00:29:44.968 --> 00:29:49.400 some careful use of the components

659 00:29:49.400 --> 00:29:51.613 of the estimator and iterated expectation.

660 00:29:52.510 --> 00:29:55.483 So it's really a pretty simple proof, which I like.

661 00:29:57.400 --> 00:29:59.030 So yeah, this is the main result.

662 00:29:59.030 --> 00:30:01.980 And again, we're not assuming anything beyond

663 00:30:01.980 --> 00:30:03.950 this mild stability here, which is nice.

664 00:30:03.950 --> 00:30:07.100 So you can use whatever regression procedures you like.

665 00:30:07.100 --> 00:30:09.040 And this will tell you something about the error

666 00:30:09.040 --> 00:30:12.490 how it relates to the Oracle error that you would get

667 00:30:12.490 --> 00:30:15.043 if you actually observed the potential outcomes.

668 00:30:18.350 --> 00:30:20.880 So this is model free method-agnostic,

669 00:30:20.880 --> 00:30:22.570 it's also a finite sample down,

670 00:30:22.570 --> 00:30:24.710 there's nothing asymptotic here.

671 00:30:24.710 --> 00:30:27.990 This means that the mean squared error is upper bounded up

672 00:30:27.990 --> 00:30:30.950 to some constant times this term on the right.

673 00:30:30.950 --> 00:30:34.133 So there's no end going to infinity or anything here either.

674 00:30:38.740 --> 00:30:40.540 So the other crucial point of this is

675 00:30:40.540 --> 00:30:43.063 because we have a product of mean squared errors,

676 00:30:44.010 --> 00:30:46.050 you have the kind of usual doubly robust story.

677 00:30:46.050 --> 00:30:48.800 So if one of these is small, the product will be small,

678 00:30:49.780 --> 00:30:51.640 potentially more importantly, if they're both kind

679 00:30:51.640 --> 00:30:54.920 of modest sized because both, maybe the propensity score

680 00:30:54.920 --> 00:30:57.250 and the regression functions are hard to estimate

681 00:30:57.250 --> 00:31:01.270 the product will be potentially quite a bit smaller

682 00:31:01.270 --> 00:31:04.370 than the individual pieces.

683 00:31:04.370 --> 00:31:07.850 And this is why this is showing you that that sort

684 00:31:07.850 --> 00:31:09.650 of plugging approach, which would really just be driven

685 00:31:09.650 --> 00:31:10.977 by the mean squared error for estimating

686 00:31:10.977 --> 00:31:15.030 the regression functions can be improved by quite a bit,

687 00:31:15.030 --> 00:31:16.770 especially if there's some structure to exploit

688 00:31:16.770 --> 00:31:18.363 in the propensity scores.

689 00:31:22.960 --> 00:31:25.800 Yeah, so in previous work people used specific methods.

690 00:31:25.800 --> 00:31:27.528 So they would say I'll use

691 00:31:27.528 --> 00:31:31.467 maybe series estimators or current estimators

692 00:31:31.467 --> 00:31:33.520 and then the error bound was actually bigger

693 00:31:33.520 --> 00:31:35.510 than what we get here.

694 00:31:35.510 --> 00:31:38.320 So this it's a little surprising that you can get

695 00:31:38.320 --> 00:31:40.130 a smaller error bound under weaker assumptions,

696 00:31:40.130 --> 00:31:42.050 but this is a nice advantage

697 00:31:42.050 --> 00:31:44.313 of the sample splitting trick here.

698 00:31:48.870 --> 00:31:51.670 Now that you have this nice error bound you can plug

699 00:31:51.670 --> 00:31:54.500 in sort of results from any of your favorite estimators.

700 00:31:56.010 --> 00:31:59.050 So we know lots about mean squared error

701 00:31:59.050 --> 00:32:01.030 for estimating regression functions.

702 00:32:01.030 --> 00:32:03.350 And so you can just plug in what you get here.

703 00:32:03.350 --> 00:32:05.800 So for example, you think about smooth functions.

704 00:32:07.150 --> 00:32:11.060 So these are functions and hold their classes intuitively

705 00:32:11.060 --> 00:32:13.080 these are functions that are close to their tailored,

706 00:32:13.080 --> 00:32:16.930 approximations, the strict definition,

707 00:32:16.930 --> 00:32:20.363 which may be I'll pass in the interest of time.

708 00:32:21.610 --> 00:32:25.070 Then you can say, for example, if PI is alpha smooth,

709 00:32:25.070 --> 00:32:28.630 so it has alpha partial derivatives

710 00:32:29.510 --> 00:32:32.790 with the highest order Lipschitz then we know

711 00:32:32.790 --> 00:32:36.760 that you can estimate that a propensity score

712 00:32:36.760 --> 00:32:37.803 with the mean squared error that looks like

713 00:32:37.803 --> 00:32:41.120 n to the minus two alpha over two alpha plus D,

714 00:32:41.120 --> 00:32:43.530 this is the usual non-parametric regression

715 00:32:43.530 --> 00:32:44.480 mean squared error.

716 00:32:46.384 --> 00:32:49.330 You can say the same thing for the regression functions.

717 00:32:49.330 --> 00:32:51.150 If they're beta smooth, then we can estimate them

718 00:32:51.150 --> 00:32:52.873 at the usual non-parametric rate,

719 00:32:52.873 --> 00:32:55.153 n to the minus two beta over two beta plus D.

720 00:32:56.050 --> 00:32:56.890 Then we could say,

721 00:32:56.890 --> 00:32:59.470 okay, suppose the conditional effect,

722 00:32:59.470 --> 00:33:03.840 Tau is gamma smooth, and gamma, it can't be smaller

723 00:33:03.840 --> 00:33:05.390 than beta, it has to be at least as smooth

724 00:33:05.390 --> 00:33:07.510 as the regression functions and in practice,

725 00:33:07.510 --> 00:33:09.220 it could be much more smooth.

726 00:33:09.220 --> 00:33:12.440 So for example, in the case where the CATE is just zero

727 00:33:12.440 --> 00:33:17.170 or constant, Gamma's like infinity, infinitely smooth.

728 00:33:17.170 --> 00:33:20.250 Then if we use a second stage estimator that's optimal

729 00:33:20.250 --> 00:33:22.150 for estimating Gamma smooth functions,

730 00:33:23.740 --> 00:33:25.280 we can just plug in the error rates

731 00:33:25.280 --> 00:33:26.330 that we get and see

732 00:33:26.330 --> 00:33:28.390 that we get a mean squared error bound

733 00:33:28.390 --> 00:33:30.390 that looks like the Oracle rate.

734 00:33:30.390 --> 00:33:33.250 This is the rate we would get if we actually observed

735 00:33:33.250 --> 00:33:34.698 the potential outcomes.

736 00:33:34.698 --> 00:33:37.290 And then we get this product of mean squared errors.

737 00:33:37.290 --> 00:33:39.640 And so whenever this product, it means squared errors

738 00:33:39.640 --> 00:33:42.010 is smaller than the Oracle rate,

739 00:33:42.010 --> 00:33:45.770 then we're achieving the Oracle rate up to constants,

740 00:33:45.770 --> 00:33:46.890 the same rate that we would get

741 00:33:46.890 --> 00:33:49.083 if we actually saw Y one minus Y zero.

742 00:33:51.240 --> 00:33:52.799 And so you can work out the conditions,

743 00:33:52.799 --> 00:33:56.070 what you need to make this term smaller than this one,

744 00:33:56.070 --> 00:33:57.530 that's just some algebra

745 00:33:59.550 --> 00:34:01.773 and it has some interesting structure.

746 00:34:02.620 --> 00:34:07.050 So if the average smoothness of the two nuisance functions,

747 00:34:07.050 --> 00:34:09.420 the propensity score and the regression function

748 00:34:09.420 --> 00:34:14.020 is greater than D over two divided by some inflation factor,

749 00:34:14.020 --> 00:34:17.530 then you can say that you're achieving

750 00:34:17.530 --> 00:34:19.480 the same rate as this Oracle procedure.

751 00:34:22.230 --> 00:34:26.660 So the analog of this for the average treatment effect

752 00:34:26.660 --> 00:34:27.630 or the result you need

753 00:34:27.630 --> 00:34:29.990 for the standard doubly robust estimate,

754 00:34:29.990 --> 00:34:31.426 or the average treatment effect

755 00:34:31.426 --> 00:34:34.470 is that the average smoothness is greater than D over two.

756 00:34:34.470 --> 00:34:35.700 So here we don't have D over two,

757 00:34:35.700 --> 00:34:38.983 we have D over two over one plus D over gamma.

758 00:34:40.270 --> 00:34:43.850 So this is actually giving you a sort

759 00:34:43.850 --> 00:34:48.160 of a lower threshold for achieving Oracle rates

760 00:34:48.160 --> 00:34:49.230 in this problem.

761 00:34:49.230 --> 00:34:50.810 So, because it's a harder problem,

762 00:34:50.810 --> 00:34:52.100 we need weaker conditions

763 00:34:52.100 --> 00:34:55.030 on the nuisance estimation to behave like an Oracle

764 00:34:56.030 --> 00:34:58.120 and how much weaker those conditions

765 00:34:58.120 --> 00:35:00.242 are, depends on the dimension of the covariates

766 00:35:00.242 --> 00:35:03.850 and the smoothness of the conditional effect.

767 00:35:03.850 --> 00:35:06.040 So if we think about the case where the conditional effect

768 00:35:06.040 --> 00:35:07.400 is like infinitely smooth,

769 00:35:07.400 --> 00:35:10.000 so this is almost like a parametric problem.

770 00:35:10.000 --> 00:35:12.660 Then we recovered the usual condition that we need

771 00:35:12.660 --> 00:35:14.430 for the doubly robust estimator to be root

772 00:35:14.430 --> 00:35:17.693 and consistent as greater than D over two.

773 00:35:19.920 --> 00:35:24.263 But when dimension is for some non-trivial smoothness,

774 00:35:25.560 --> 00:35:27.720 then we're somewhere in between sort of when

775 00:35:27.720 --> 00:35:31.343 a plugin is optimal and this nice kind of parametric setup.

776 00:35:33.630 --> 00:35:37.280 So this is just a picture of the rates here

777 00:35:37.280 --> 00:35:39.210 which is useful to keep in mind.

778 00:35:39.210 --> 00:35:42.670 So here on the x-axis, we have the smoothness

779 00:35:42.670 --> 00:35:44.112 of the nuisance functions.

780 00:35:44.112 --> 00:35:45.750 You can think of this as the average smoothness

781 00:35:45.750 --> 00:35:49.180 of the propensity score in regression functions.

782 00:35:49.180 --> 00:35:52.200 And again, in this holder smooth model,

783 00:35:52.200 --> 00:35:55.440 which is a common model people use in nonparametrics,

784 00:35:55.440 --> 00:35:56.500 the more smooth things are

785 00:35:56.500 --> 00:35:58.343 the easier it is to estimate them.

786 00:35:59.760 --> 00:36:01.870 And then here we have the mean squared error

787 00:36:01.870 --> 00:36:03.773 for estimating the conditional effect.

788 00:36:06.435 --> 00:36:08.710 So here is the minimax lower bounce,

789 00:36:08.710 --> 00:36:10.580 this is the best possible mean squared error

790 00:36:10.580 --> 00:36:13.850 that you can achieve for the average treatment effect.

791 00:36:13.850 --> 00:36:16.120 This is just to kind of anchor our results

792 00:36:16.120 --> 00:36:18.740 and think about what happens relative to this nicer,

793 00:36:18.740 --> 00:36:21.030 simpler parameter, which is just the overall average

794 00:36:21.030 --> 00:36:22.630 and not the conditional average.

795 00:36:23.700 --> 00:36:25.880 So once you hit a certain smoothness in this case,

796 00:36:25.880 --> 00:36:28.363 it's five, so this is looking at

797 00:36:28.363 --> 00:36:31.120 a 20 dimensional covariate case where

798 00:36:32.420 --> 00:36:34.900 the CATE smoothness is twice the dimension

799 00:36:34.900 --> 00:36:36.233 just to fix ideas.

800 00:36:37.663 --> 00:36:41.670 And so once we hit this smoothness of five,

801 00:36:41.670 --> 00:36:43.040 so we have five partial derivatives,

802 00:36:43.040 --> 00:36:47.050 then it's possible to achieve a Rudin rate.

803 00:36:47.050 --> 00:36:49.940 So this is into the one half for estimating

804 00:36:49.940 --> 00:36:51.880 the average treatment effect.

805 00:36:51.880 --> 00:36:55.040 Rudin rates are never possible for conditional effects.

806 00:36:55.040 --> 00:36:57.690 So here's the Oracle rate.

807 00:36:57.690 --> 00:36:59.500 This is the rate that we would achieve in this problem

808 00:36:59.500 --> 00:37:02.010 if we actually observed the potential outcomes.

809 00:37:02.010 --> 00:37:05.193 So it's lower than Rudin, it's a bigger error.

810 00:37:07.740 --> 00:37:09.510 Here's what you would get with the plugin.

811 00:37:09.510 --> 00:37:13.290 This is just really inheriting the complexity

812 00:37:13.290 --> 00:37:15.390 and estimating the regression functions individually,

813 00:37:15.390 --> 00:37:17.610 it doesn't capture this CATE smoothness

814 00:37:17.610 --> 00:37:19.600 and so you need the regression functions

815 00:37:19.600 --> 00:37:21.500 to be sort of infinitely smoother or as smooth

816 00:37:21.500 --> 00:37:24.673 as the CATE to actually get Oracle efficiency

817 00:37:24.673 --> 00:37:26.853 with the plugin estimator.

818 00:37:27.760 --> 00:37:29.240 It's this plugin as big errors,

819 00:37:29.240 --> 00:37:32.040 if we use this DR-learner approach,

820 00:37:32.040 --> 00:37:35.850 we close this gap substantially.

821 00:37:35.850 --> 00:37:38.970 So we can say that we're hitting this Oracle rate.

822 00:37:38.970 --> 00:37:40.560 Once we have a certain amount of smoothness

823 00:37:40.560 --> 00:37:44.430 of the nuisance functions and in between

824 00:37:44.430 --> 00:37:47.363 we get an error that looks something like this.

825 00:37:48.590 --> 00:37:51.290 So this is just a picture of this row results showing,

826 00:37:52.530 --> 00:37:55.790 graphically, the improvement of the DR-learner approach

827 00:37:55.790 --> 00:37:57.913 here over a simple plug estimator.

828 00:38:03.360 --> 00:38:05.770 So yeah, just the punchline here is

829 00:38:05.770 --> 00:38:09.460 this simple two-stage doubly robust approach

830 00:38:09.460 --> 00:38:12.340 can do a good job adapting to underlying structure

831 00:38:12.340 --> 00:38:14.070 in the conditional effect,

832 00:38:14.070 --> 00:38:15.820 even when the nuisance stuff,

833 00:38:15.820 --> 00:38:16.850 the propensity scores

834 00:38:16.850 --> 00:38:18.280 and the underlying regression functions

835 00:38:18.280 --> 00:38:21.223 are more complex or less smooth in this case.

836 00:38:23.920 --> 00:38:26.290 This is just talking about the relation

837 00:38:26.290 --> 00:38:27.920 to the average treatment effect conditions,

838 00:38:27.920 --> 00:38:29.393 which I mentioned before.

839 00:38:31.740 --> 00:38:34.350 So you can do the same thing for any generic

840 00:38:34.350 --> 00:38:35.490 regression methods you like.

841 00:38:35.490 --> 00:38:37.880 So in the paper, I do this for smooth models

842 00:38:37.880 --> 00:38:39.320 and sparse models, which are common

843 00:38:39.320 --> 00:38:40.660 in these non-parametric settings,

844 00:38:40.660 --> 00:38:42.970 where you have high dimensional Xs

845 00:38:42.970 --> 00:38:45.050 and you believe that some subset

846 00:38:45.050 --> 00:38:48.020 of them are the ones that matter.

847 00:38:48.020 --> 00:38:50.030 So I'll skip past this, if you're curious though,

848 00:38:50.030 --> 00:38:51.700 all the details are in the paper.

849 00:38:51.700 --> 00:38:53.360 So you can say, what kind of sparse should

850 00:38:53.360 --> 00:38:55.390 be doing need in the propensity score

851 00:38:55.390 --> 00:38:56.730 in regression functions to be able

852 00:38:56.730 --> 00:38:59.250 to get something that behaves like an Oracle

853 00:38:59.250 --> 00:39:02.050 that actually saw the potential outcomes from the start.

854 00:39:03.790 --> 00:39:05.210 You can also do the same kind of game

855 00:39:05.210 --> 00:39:06.380 where you compare this to what you need

856 00:39:06.380 --> 00:39:08.030 for the average treatment effect.

857 00:39:10.610 --> 00:39:12.930 Yeah, happy to talk about this offline

858 00:39:12.930 --> 00:39:15.263 or afterwards people have questions.

859 00:39:18.330 --> 00:39:20.690 So there's also a nice kind of side result

860 00:39:20.690 --> 00:39:23.333 which I think I'll also go through quickly here.

861 00:39:24.480 --> 00:39:29.020 From all this, is just a general Oracle inequality

862 00:39:29.020 --> 00:39:31.170 for regression when you have some estimated outcomes.

863 00:39:31.170 --> 00:39:33.574 So in some sense, there isn't anything really special

864 00:39:33.574 --> 00:39:37.220 in our results that has to do

865 00:39:37.220 --> 00:39:38.600 with this particular pseudo outcome.

866 00:39:38.600 --> 00:39:43.380 So, the proof that we have here works

867 00:39:43.380 --> 00:39:46.199 for any second stage or any two-stage sort

868 00:39:46.199 --> 00:39:47.730 of regression procedure

869 00:39:47.730 --> 00:39:49.940 where you first estimate some nuisance stuff,

870 00:39:49.940 --> 00:39:51.530 create a pseudo outcome that depends

871 00:39:51.530 --> 00:39:53.720 on this estimated stuff and then do a regression

872 00:39:53.720 --> 00:39:56.333 of the pseudo outcome on some set of covariates.

873 00:39:57.613 --> 00:39:59.900 And so a nice by-product of this work,

874 00:39:59.900 --> 00:40:02.240 as you get a kind of similar error bound

875 00:40:02.240 --> 00:40:06.630 for just generic regression with pseudo outcomes.

876 00:40:06.630 --> 00:40:09.500 This comes up in a lot of different problems, actually.

877 00:40:09.500 --> 00:40:14.500 So one is when you want just a partly conditional effect.

878 00:40:15.010 --> 00:40:16.760 So maybe I don't care about how effects vary

879 00:40:16.760 --> 00:40:18.990 with all the Xs, but just a subset of them,

880 00:40:18.990 --> 00:40:20.270 then you can apply this result.

881 00:40:20.270 --> 00:40:22.990 I have a paper with a great student, Amanda Costin,

882 00:40:22.990 --> 00:40:25.570 who studied a version of this

883 00:40:28.170 --> 00:40:30.020 regression with missing outcomes.

884 00:40:30.020 --> 00:40:33.000 Again, these look like nonparametric regression problems

885 00:40:33.000 --> 00:40:35.990 where you have to estimate some pseudo outcome

886 00:40:35.990 --> 00:40:39.950 dose response curve problems, conditional IV effects,

887 00:40:39.950 --> 00:40:40.930 partially linear IVs.

888 00:40:40.930 --> 00:40:42.560 So there are lots of different variants where you need

889 00:40:42.560 --> 00:40:47.560 to do some kind of two-stage regression procedure like this.

890 00:40:50.840 --> 00:40:52.420 Again, you just need a stability condition

891 00:40:52.420 --> 00:40:53.690 and you need some sample splitting

892 00:40:53.690 --> 00:40:57.045 and you can give a similar kind of a nice rate result

893 00:40:57.045 --> 00:41:01.010 that we got for the CATE specific problem,

894 00:41:01.010 --> 00:41:03.683 but in generic pseudo outcome progression problem.

895 00:41:07.430 --> 00:41:09.750 So we've got about 15 minutes,

896 00:41:09.750 --> 00:41:11.810 I have some simulations,

897 00:41:11.810 --> 00:41:14.303 which I think I will go over quickly.

898 00:41:15.400 --> 00:41:16.980 So we did this in a couple simple models,

899 00:41:16.980 --> 00:41:19.810 one, a high dimensional linear model.

900 00:41:19.810 --> 00:41:21.890 It's actually a logistic model where

901 00:41:21.890 --> 00:41:24.640 we have 500 covariates and 50

902 00:41:24.640 --> 00:41:26.563 of them have non-zero coefficients.

903 00:41:28.120 --> 00:41:32.180 We just used the default lasso fitting in our

904 00:41:32.180 --> 00:41:34.410 and compared plugin estimators

905 00:41:34.410 --> 00:41:36.840 to the doubly robust approach that we talked

906 00:41:36.840 --> 00:41:38.510 about and then also an ex-learner

907 00:41:38.510 --> 00:41:43.510 which is some sort of variants of the plug-in approach

908 00:41:43.810 --> 00:41:45.933 that was proposed in recent years.

909 00:41:46.890 --> 00:41:48.910 And the basic story is you get sort

910 00:41:48.910 --> 00:41:50.240 of what the theory predicts.

911 00:41:50.240 --> 00:41:54.117 So the DR-learner does better than these plug-in types

912 00:41:54.117 --> 00:41:57.820 of approaches in this setting.

913 00:41:57.820 --> 00:42:00.347 The nuisance functions are hard to estimate

914 00:42:00.347 --> 00:42:02.230 and so you don't see a massive gain over,

915 00:42:02.230 --> 00:42:03.620 for example, the X-Learner,

916 00:42:03.620 --> 00:42:04.830 you do see a pretty massive gain

917 00:42:04.830 --> 00:42:06.323 over the simple plugin.

918 00:42:08.410 --> 00:42:09.650 And we're a bit away

919 00:42:09.650 --> 00:42:13.070 from this Oracle DR-learner approach here,

920 00:42:13.070 --> 00:42:16.430 so that means great errors is relatively different.

921 00:42:16.430 --> 00:42:18.820 This is telling us that the nuisance stuff is hard

922 00:42:18.820 --> 00:42:21.003 to estimate in this simulation set up.

923 00:42:22.300 --> 00:42:23.630 Here's another simulation based

924 00:42:23.630 --> 00:42:25.973 on that plot I showed you before.

925 00:42:27.750 --> 00:42:30.970 And so here, I'm actually estimating the propensity scores,

926 00:42:30.970 --> 00:42:33.070 but I'm constructing the estimates myself

927 00:42:33.070 --> 00:42:35.450 so that I can control the rate of convergence

928 00:42:35.450 --> 00:42:38.570 and see how things change across different error rates

929 00:42:38.570 --> 00:42:40.620 for estimating with propensity score.

930 00:42:40.620 --> 00:42:41.520 So here's what we see.

931 00:42:41.520 --> 00:42:44.080 So on the x-axis here,

932 00:42:44.080 --> 00:42:47.990 we have how well we're estimating the propensity score.

933 00:42:47.990 --> 00:42:49.620 So this is a convergence rate

934 00:42:49.620 --> 00:42:52.440 for the propensity score estimator.

935 00:42:52.440 --> 00:42:54.210 Y-axis, we have the mean squared error

936 00:42:54.210 --> 00:42:56.270 and then this red line is the plugin estimator,

937 00:42:56.270 --> 00:42:57.420 it's doing really poorly.

938 00:42:57.420 --> 00:42:59.440 It's not capturing this underlying simplicity

939 00:42:59.440 --> 00:43:00.470 of the conditional effects.

940 00:43:00.470 --> 00:43:03.380 It's really just inheriting that difficulty

941 00:43:03.380 --> 00:43:05.450 in estimating the regression functions.

942 00:43:05.450 --> 00:43:07.120 Here's the X-learner, it's doing a bit better

943 00:43:07.120 --> 00:43:09.570 than the plugin, but it's still not doing

944 00:43:09.570 --> 00:43:12.190 a great job capturing the underlying simplicity

945 00:43:12.190 --> 00:43:14.330 and the conditional effect.

946 00:43:14.330 --> 00:43:16.290 This dotted line is the Oracle.

947 00:43:16.290 --> 00:43:17.500 So this is what you would get

948 00:43:17.500 --> 00:43:19.820 if you actually observed the potential outcomes.

949 00:43:19.820 --> 00:43:22.900 And then the black line is the DR-learner,

950 00:43:22.900 --> 00:43:24.310 this two-stage procedure here,

951 00:43:24.310 --> 00:43:26.350 I'm just using smoothing splines everywhere,

952 00:43:26.350 --> 00:43:29.370 just defaults in R, it's like three lines of code,

953 00:43:29.370 --> 00:43:30.560 all the code's in the paper, too,

954 00:43:30.560 --> 00:43:33.300 if you want to play around with this.

955 00:43:33.300 --> 00:43:35.370 And here we see what we expect.

956 00:43:35.370 --> 00:43:37.570 So when it's really hard to estimate the propensity score,

957 00:43:37.570 --> 00:43:40.250 it's just a hard problem and we don't do

958 00:43:40.250 --> 00:43:43.510 much better than the X-learner.

959 00:43:43.510 --> 00:43:46.110 We still get some gain over the plugin in this case,

960 00:43:47.140 --> 00:43:50.370 but as soon as you can estimate the propensity score

961 00:43:50.370 --> 00:43:54.390 well at all, you start seeing some pretty big gains

962 00:43:54.390 --> 00:43:56.280 by doing this doubly robust approach

963 00:43:56.280 --> 00:43:58.420 and at some point we start to roughly match

964 00:43:58.420 --> 00:44:00.383 the Oracle actually.

965 00:44:01.600 --> 00:44:02.710 As soon as we’re getting something like

966 00:44:02.710 --> 00:44:04.070 into the quarter rates in this case,

967 00:44:04.070 --> 00:44:05.683 we’re getting close to the Oracle.

968 00:44:10.470 --> 00:44:12.290 So maybe I’ll just show you an illustration

969 00:44:12.290 --> 00:44:15.100 and then I’ll talk about the second part of the talk

970 00:44:15.100 --> 00:44:16.850 and very briefly if people have,

971 00:44:16.850 --> 00:44:19.507 want to talk about that,

972 00:44:19.507 --> 00:44:22.540 offline, I’d be more than happy to.

973 00:44:22.540 --> 00:44:24.620 So here’s a study, which I actually learned about

974 00:44:24.620 --> 00:44:27.390 from Peter looking at effects of canvassing

975 00:44:27.390 --> 00:44:30.070 on voter turnout, so this is this timely study.

976 00:44:30.070 --> 00:44:35.070 Here’s the paper, there are almost 20,000 voters

977 00:44:35.580 --> 00:44:37.400 across six cities here.

978 00:44:37.400 --> 00:44:41.650 They’re randomly encouraged to vote

979 00:44:41.650 --> 00:44:44.537 in these local elections that people would go

980 00:44:44.537 --> 00:44:47.150 and talk to them face to face.

981 00:44:47.150 --> 00:44:50.210 You remember what that was like pre-pandemic.

982 00:44:50.210 --> 00:44:54.570 Here’s a script of the sort of canvassing that they did,

983 00:44:54.570 --> 00:44:57.526 just saying, reminding them of the election,

984 00:44:57.526 --> 00:44:59.850 giving them a reminder to vote.

985 00:44:59.850 --> 00:45:01.840 Hopefully I’m doing this for you as well,

986 00:45:01.840 --> 00:45:03.770 if you haven’t voted already.

987 00:45:03.770 --> 00:45:07.170 And so what’s the data we have here?

988 00:45:07.170 --> 00:45:09.680 We have a number of covariates things like city,

989 00:45:09.680 --> 00:45:11.470 party affiliation, some measures

990 00:45:11.470 --> 00:45:15.020 of the past voting history, age, family size, race.

991 00:45:15.020 --> 00:45:19.410 Again, the treatment is whether they work randomly contact

992 00:45:19.410 --> 00:45:22.180 is actually whether they were randomly assigned some cases,

993 00:45:22.180 --> 00:45:24.730 people couldn't be contacted in the setup.

994 00:45:24.730 --> 00:45:26.370 So we're just looking at intention

995 00:45:26.370 --> 00:45:28.110 to treat kinds of effects.

996 00:45:28.110 --> 00:45:30.210 And then the outcome is whether people voted

997 00:45:30.210 --> 00:45:32.250 in the local election or not.

998 00:45:32.250 --> 00:45:34.940 So just as kind of a proof of concept,

999 00:45:34.940 --> 00:45:37.050 I use this DR-learner approach,

1000 00:45:37.050 --> 00:45:42.050 I just use two folds and use random forest separator

1001 00:45:42.340 --> 00:45:45.143 for the first stage regressions and the second stage.

1002 00:45:47.290 --> 00:45:50.070 And actually for one part of the analysis,

1003 00:45:50.070 --> 00:45:52.870 I used generalized additive models in that second stage.

1004 00:45:56.100 --> 00:45:59.970 So here's a histogram of the conditional effect estimates.

1005 00:45:59.970 --> 00:46:02.640 So there's sort of a big chunk, a little bit above zero,

1006 00:46:02.640 --> 00:46:04.260 but then there is some heterogeneity

1007 00:46:04.260 --> 00:46:06.840 around that in this case.

1008 00:46:06.840 --> 00:46:07.673 So there are some people

1009 00:46:07.673 --> 00:46:12.490 who maybe seem especially responsive to canvassing,

1010 00:46:12.490 --> 00:46:14.517 maybe some people who are going to know it

1011 00:46:14.517 --> 00:46:17.267 and actually some are less likely to vote, potentially.

1012 00:46:18.230 --> 00:46:21.060 This is a plot of the effect estimates

1013 00:46:21.060 --> 00:46:22.350 from this DR-learner procedure,

1014 00:46:22.350 --> 00:46:24.430 just to see what they look like, 1015 00:46:24.430 --> 00:46:27.280 how this would work in practice across

1016 00:46:27.280 --> 00:46:29.560 to potentially important covariate.

1017 00:46:29.560 --> 00:46:33.900 So here's the age of the voter and then the party

1018 00:46:33.900 --> 00:46:38.770 and the color here represents the size and direction

1019 00:46:38.770 --> 00:46:41.380 of the CATE estimate of the conditional effect estimates,

1020 00:46:41.380 --> 00:46:45.130 so blue is canvassing is having a bigger effect

1021 00:46:45.130 --> 00:46:49.700 on voting in the next local election.

1022 00:46:49.700 --> 00:46:54.213 Red means less likely to vote due to canvassing.

1023 00:46:55.340 --> 00:46:58.730 So you can see some interesting structure here just briefly,

1024 00:46:58.730 --> 00:47:00.580 the independent people,

1025 00:47:00.580 --> 00:47:03.100 it seems like the effects are closer to zero.

1026 00:47:03.100 --> 00:47:06.950 Democrats maybe seem more likely to be positively affected,

1027 00:47:06.950 --> 00:47:10.580 maybe more so among younger people.

1028 00:47:10.580 --> 00:47:12.330 It's just an example of the kind of

1029 00:47:13.240 --> 00:47:15.680 sort of graphical visualization stuff you could do

1030 00:47:15.680 --> 00:47:17.080 with this sort of procedure.

1031 00:47:18.310 --> 00:47:20.610 This is the plot I showed before, where here,

1032 00:47:20.610 --> 00:47:22.190 we're looking at just how the conditional

1033 00:47:22.190 --> 00:47:23.970 effect varies with age.

1034 00:47:23.970 --> 00:47:25.300 And you can see some evidence

1035 00:47:25.300 --> 00:47:28.490 that younger people are to canvassing.

1036 00:47:32.920 --> 00:47:36.483 Older people, less evidence that there's any response.

1037 00:47:42.610 --> 00:47:45.743 I should stop here and see if people have any questions.

1038 00:47:51.330 --> 00:47:53.150 - So Edward, can I ask a question?

1039 00:47:53.150 --> 00:47:54.570 - Of course yeah.

1040 00:47:54.570 --> 00:47:57.300 - I think we've discussed about point estimation.

1041 00:47:57.300 --> 00:47:58.790 Does this approach also allows

1042 00:47:58.790 --> 00:48:00.990 for consistent variance estimation?

1043 00:48:00.990 --> 00:48:04.410 - Yeah, that's a great question.

1044 00:48:04.410 --> 00:48:07.580 Yeah, I haven't included any of that here,

1045 00:48:07.580 --> 00:48:09.513 but if you think about that.

1046 00:48:10.690 --> 00:48:14.863 This Oracle result that we have.

1047 00:48:16.761 --> 00:48:19.490 If these errors are small enough,

1048 00:48:19.490 --> 00:48:21.570 so under the kinds of conditions that we talked about,

1049 00:48:21.570 --> 00:48:25.640 then we're getting an estimate of it looks like an Oracle

1050 00:48:25.640 --> 00:48:28.620 has to meet or of the potential outcomes on the covariates.

1051 00:48:28.620 --> 00:48:30.630 And that means that as long as these are small enough,

1052 00:48:30.630 --> 00:48:33.100 we could just port over any inferential tools

1053 00:48:33.100 --> 00:48:35.496 that we like from standard non-parametric regression

1054 00:48:35.496 --> 00:48:38.090 treating our pseudo outcomes as if they were

1055 00:48:38.090 --> 00:48:41.290 the true existential outcomes, yeah.

1056 00:48:41.290 --> 00:48:43.135 That's a really important point,

1057 00:48:43.135 --> 00:48:44.010 I'm glad you mentioned that.

1058 00:48:44.010 --> 00:48:45.070 - Thanks.

1059 00:48:45.070 --> 00:48:47.440 - So inference is more complicated

1060 00:48:47.440 --> 00:48:49.590 and nuanced than non-parametric regression,

1061 00:48:50.596 --> 00:48:54.823 but any inferential tool could be used here.

1062 00:48:55.790 --> 00:48:57.270 - So operationally, just to think

1063 00:48:57.270 --> 00:48:59.360 about how to operationalize the variance estimation

1064 00:48:59.360 --> 00:49:02.820 also, does that require the cross fitting procedure

1065 00:49:02.820 --> 00:49:05.910 where you're swapping your D one D two

1066 00:49:05.910 --> 00:49:08.403 in the estimation process and then?

1067 00:49:09.550 --> 00:49:10.870 - Yeah, that's a great question too.

1068 00:49:10.870 --> 00:49:11.810 So not necessarily,

1069 00:49:11.810 --> 00:49:14.140 so you could just use these folds

1070 00:49:14.140 --> 00:49:17.130 for nuisance training and then go to this fold

1071 00:49:17.130 --> 00:49:19.207 and then just forget that you ever used this data

1072 00:49:19.207 --> 00:49:21.310 and just do variance estimation here.

1073 00:49:21.310 --> 00:49:22.360 The drawback there would be,

1074 00:49:22.360 --> 00:49:24.768 you're only using a third of your data.

1075 00:49:24.768 --> 00:49:26.480 If you really want to make full use

1076 00:49:26.480 --> 00:49:27.670 of the sample size using

1077 00:49:27.670 --> 00:49:30.970 the cross fitting procedure would be ideal,

1078 00:49:30.970 --> 00:49:32.150 but the inference doesn't change.

1079 00:49:32.150 --> 00:49:34.580 So if you do cross fitting,

1080 00:49:34.580 --> 00:49:35.910 you would at the end of the day,

1081 00:49:35.910 --> 00:49:39.340 you'd get an out of sample CATE estimate

1082 00:49:39.340 --> 00:49:42.144 for every single row in your data, every subject,

1083 00:49:42.144 --> 00:49:44.444 but just where that CATE was built from other,

1084 00:49:45.440 --> 00:49:47.070 the nuisance stuff for that estimate

1085 00:49:47.070 --> 00:49:49.500 was built from other samples.

1086 00:49:49.500 --> 00:49:51.530 But at the end of the day, you'd get one big column

1087 00:49:51.530 --> 00:49:53.760 with all these out of sample CATE estimates

1088 00:49:53.760 --> 00:49:54.640 and then you could just use

1089 00:49:54.640 --> 00:49:57.283 whatever inferential tools you like there.

1090 00:50:00.250 --> 00:50:01.083 - Thanks.

1091 00:50:07.360 --> 00:50:09.930 - So, just got a few minutes.

1092 00:50:09.930 --> 00:50:11.830 So maybe I'll just give you a high level kind of picture

1093 00:50:11.830 --> 00:50:14.230 of the stuff in the second part of this talk

1094 00:50:14.230 --> 00:50:18.540 which is really about pursuing the fundamental limits

1095 00:50:18.540 --> 00:50:20.010 of conditional effect estimation.

1096 00:50:20.010 --> 00:50:22.970 So what's the best we could possibly do here?

1097 00:50:22.970 --> 00:50:24.750 This is completely unknown,

1098 00:50:24.750 --> 00:50:27.290 which I think is really fascinating.

1099 00:50:27.290 --> 00:50:29.254 So if you think about what we have so far,

1100 00:50:29.254 --> 00:50:32.810 so far, we've given these sufficient conditions under

1101 00:50:32.810 --> 00:50:35.293 which this DR-learner is Oracle efficient,

1102 00:50:36.170 --> 00:50:38.040 but a natural question here is what happens

1103 00:50:38.040 --> 00:50:40.480 when those mean squared error terms are too big

1104 00:50:40.480 --> 00:50:42.010 and so we can't say that we're getting

1105 00:50:42.010 --> 00:50:43.793 the Oracle rate anymore.

1106 00:50:45.230 --> 00:50:46.063 Then you might say,

1107 00:50:46.063 --> 00:50:50.450 okay, is this a bug with the DR-learner?

1108 00:50:50.450 --> 00:50:52.310 Maybe I could have adapted this in some way

1109 00:50:52.310 --> 00:50:56.008 to actually do better or maybe I've reached the limits

1110 00:50:56.008 --> 00:50:59.760 of how well I can do for estimating the effect.

1111 00:50:59.760 --> 00:51:02.790 It doesn't matter if I had gone to a different estimator,

1112 00:51:02.790 --> 00:51:04.933 think I would've had the same kind of error.

1113 00:51:06.820 --> 00:51:11.810 So this is the goal of this last part of the work.

1114 00:51:11.810 --> 00:51:14.220 So here we use a very different estimator.

1115 00:51:14.220 --> 00:51:17.090 It's built using this R-learner idea,

1116 00:51:17.090 --> 00:51:22.090 which is reproducing RKHS extension of this

1117 00:51:22.210 --> 00:51:24.420 classic double residual regression method

1118 00:51:24.420 --> 00:51:26.770 of Robinson, which is really cool.

1119 00:51:26.770 --> 00:51:30.973 This is actually from 1988, so it's a classic method.

1120 00:51:32.530 --> 00:51:34.620 And so we study a non-parametric version

1121 00:51:34.620 --> 00:51:37.880 of this built from local polynomial estimators.

1122 00:51:37.880 --> 00:51:40.220 And I'll just give you a picture

1123 00:51:40.220 --> 00:51:41.160 of what the estimator is doing.

1124 00:51:41.160 --> 00:51:42.520 It's quite a bit more complicated

1125 00:51:42.520 --> 00:51:44.920 than that dr. Learner procedure.

1126 00:51:44.920 --> 00:51:47.480 So we again use this triple sample splitting

1127 00:51:47.480 --> 00:51:49.790 and here it's actually much more crucial.

1128 00:51:49.790 --> 00:51:52.560 So if you didn't use that triple sample splitting

1129 00:51:52.560 --> 00:51:53.393 for the dr learner,

1130 00:51:53.393 --> 00:51:55.423 you'd just get a slightly different Arab bound,

1131 00:51:55.423 --> 00:51:57.060 but here it's actually really important.

1132 00:51:57.060 --> 00:51:59.760 I'd be happy to talk to people about why specifically.

1133 00:52:01.120 --> 00:52:04.190 So one part of the sample we estimate propensity scores

1134 00:52:04.190 --> 00:52:05.023 and another part of the sample.

1135 00:52:05.023 --> 00:52:07.510 We estimate propensity scores and regression functions.

1136 00:52:07.510 --> 00:52:09.900 Now the marginal regression functions,

1137 00:52:09.900 --> 00:52:13.060 we combine these to get weights, Colonel weights.

1138 00:52:13.060 --> 00:52:15.440 We also combine them to get residuals.

1139 00:52:15.440 --> 00:52:17.520 So treatment residuals and outcome residuals.

1140 00:52:17.520 --> 00:52:18.800 This is like what you would get

1141 00:52:18.800 --> 00:52:22.603 for this re Robinson procedure from econ.

1142 00:52:23.660 --> 00:52:25.500 Then we do instead of a regression

1143 00:52:25.500 --> 00:52:28.470 of outcome residuals on treatment residuals,

1144 00:52:28.470 --> 00:52:31.560 we do a weighted nonparametric regression

1145 00:52:31.560 --> 00:52:34.270 of these residuals on the treatment residuals.

1146 00:52:34.270 --> 00:52:37.880 So that's the procedure a little bit more complicated.

1147 00:52:37.880 --> 00:52:39.240 And again, this is,

1148 00:52:39.240 --> 00:52:42.450 I think there are ways to make this work well practically,

1149 00:52:42.450 --> 00:52:44.280 but the goal of this work is really to try

1150 00:52:44.280 --> 00:52:45.610 and figure out what's the best possible

1151 00:52:45.610 --> 00:52:47.360 mean squared error that we could achieve.

1152 00:52:47.360 --> 00:52:50.710 It's less about a practical method,

1153 00:52:50.710 --> 00:52:52.230 more about just understanding how hard

1154 00:52:52.230 --> 00:52:55.980 the conditional effect estimation problem is.

1155 00:52:55.980 --> 00:52:58.820 And so we actually show that a generic version

1156 00:52:58.820 --> 00:53:00.563 of this procedure,

1157 00:53:01.660 --> 00:53:03.033 as long as you estimate the propensity scores

1158 00:53:03.033 --> 00:53:05.060 and the regression functions with linear smoothers,

1159 00:53:05.060 --> 00:53:08.270 with particular bias and various properties,

1160 00:53:08.270 --> 00:53:10.910 which are standard in nonparametrics,

1161 00:53:10.910 --> 00:53:13.380 you can actually get better mean squared error.

1162 00:53:13.380 --> 00:53:15.230 Then for the dr. Learner,

1163 00:53:15.230 --> 00:53:18.620 we'll just give you a sense of what this looks like.

1164 00:53:18.620 --> 00:53:22.460 So you get something that looks like an Oracle rate plus

1165 00:53:22.460 --> 00:53:26.900 something like the squared bias from the new synced,

1166 00:53:26.900 --> 00:53:30.483 from the propensity score and regression functions.

1167 00:53:31.810 --> 00:53:35.760 So before you had the product of mean squared errors,

1168 00:53:35.760 --> 00:53:37.570 now we have the square of the bias

1169 00:53:37.570 --> 00:53:40.310 of the two procedures, the mean squared error,

1170 00:53:40.310 --> 00:53:43.290 and the propensity score in the regression function.

1171 00:53:43.290 --> 00:53:46.180 And this gives you, this opens the door to under smoothing.

1172 00:53:46.180 --> 00:53:49.340 So this means that you can estimate the propensity score

1173 00:53:49.340 --> 00:53:52.400 and the regression functions in a suboptimal way.

1174 00:53:52.400 --> 00:53:54.130 If you actually just care about the,

1175 00:53:54.130 --> 00:53:55.980 these functions by themselves.

1176 00:53:55.980 --> 00:53:59.070 So you drive down the bias that blows up

1177 00:53:59.070 --> 00:54:00.310 the variance a little bit,

1178 00:54:00.310 --> 00:54:02.600 but it turns out not to affect the conditional effect

1179 00:54:02.600 --> 00:54:05.023 estimate too much if you do it in the right way.

1180 00:54:06.040 --> 00:54:06.873 And so if.

1181 00:54:06.873 --> 00:54:08.853 You, if you do this, you get.

1182 00:54:11.313 --> 00:54:12.290 A rate that looks like this,

1183 00:54:12.290 --> 00:54:15.267 you get an Oracle rate plus into the minus two S over D.

1184 00:54:15.267 --> 00:54:17.640 And this is strictly better than what we got

1185 00:54:17.640 --> 00:54:18.693 with the dr. Learner.

1186 00:54:19.963 --> 00:54:21.030 (clears throat)

1187 00:54:21.030 --> 00:54:23.070 You can do the same game where you see sort

1188 00:54:23.070 --> 00:54:26.630 of when the Oracle rate is achieved here, it's achieved.

1189 00:54:26.630 --> 00:54:29.400 If the average smoothness of the nuisance functions

1190 00:54:29.400 --> 00:54:31.390 is greater than D over four.

1191 00:54:31.390 --> 00:54:34.140 And then here, the inflation factor is also changing.

1192 00:54:34.140 --> 00:54:35.430 So before we had,

1193 00:54:35.430 --> 00:54:38.420 we needed the smoothness to be greater than D over two,

1194 00:54:38.420 --> 00:54:39.930 over one plus D over gamma.

1195 00:54:39.930 --> 00:54:43.233 Now we have D over four over one plus or two gamma.

1196 00:54:44.680 --> 00:54:45.990 So this is a weaker condition.

1197 00:54:45.990 --> 00:54:48.180 So this is telling us that there are settings

1198 00:54:48.180 --> 00:54:51.530 where that dr. Lerner is not Oracle efficient,

1199 00:54:51.530 --> 00:54:53.370 but there exists an estimator, which is,

1200 00:54:53.370 --> 00:54:56.200 and it looks like this estimator

1201 00:54:56.200 --> 00:54:57.033 I had described here,

1202 00:54:57.033 --> 00:54:58.520 this regression on residuals thing.

1203 00:55:01.770 --> 00:55:02.603 So that's the story.

1204 00:55:02.603 --> 00:55:03.436 You can actually,

1205 00:55:03.436 --> 00:55:04.780 you can actually beat this dr. Lerner.

1206 00:55:04.780 --> 00:55:08.280 And now the question is, okay, what happens?

1207 00:55:08.280 --> 00:55:09.270 One, what happens

1208 00:55:09.270 --> 00:55:11.280 when we're not achieving the Oracle rate here,

1209 00:55:11.280 --> 00:55:13.030 can you still do better?

1210 00:55:13.030 --> 00:55:16.393 A second question is can anything, yeah.

1211 00:55:18.660 --> 00:55:20.120 Can anything achieve the Oracle rate

1212 00:55:20.120 --> 00:55:22.260 under weaker conditions than this?

1213 00:55:22.260 --> 00:55:24.970 And so I haven't proved anything about this yet.

1214 00:55:24.970 --> 00:55:27.853 It turns out to be somewhat difficult,

1215 00:55:29.160 --> 00:55:32.900 but I conjecture that this, this condition is mini max.

1216 00:55:32.900 --> 00:55:34.410 So I don't think any,

1217 00:55:34.410 --> 00:55:36.160 any estimator could ever be Oracle efficient

1218 00:55:36.160 --> 00:55:40.250 under weaker conditions than what this estimator is.

1219 00:55:40.250 --> 00:55:41.780 So this is just a picture of the results again.

1220 00:55:41.780 --> 00:55:44.590 So here's, it's the same setting as before here,

1221 00:55:44.590 --> 00:55:48.057 we have the plugin estimator that dr. Learner.

1222 00:55:48.057 --> 00:55:51.400 And here's what we get with this.

1223 00:55:51.400 --> 00:55:52.560 I call it the LPR learner.

1224 00:55:52.560 --> 00:55:55.040 It's a local polynomial version of the, our learner.

1225 00:55:55.040 --> 00:55:57.670 And so we're, actually getting quite a bit smaller rates.

1226 00:55:57.670 --> 00:56:01.640 We're hitting the Oracle rate under Meeker conditions

1227 00:56:01.640 --> 00:56:03.470 on the smoothness.

1228 00:56:03.470 --> 00:56:08.304 Now, the question is whether we can fill this gap anymore,

1229 00:56:08.304 --> 00:56:09.137 and this is unknown.

1230 00:56:09.137 --> 00:56:11.883 This is one of the open questions in causal inference.

1231 00:56:14.070 --> 00:56:17.990 So yeah, I think in the interest of time,

1232 00:56:17.990 --> 00:56:20.810 I'll skip to the discussion section here.

1233 00:56:20.810 --> 00:56:22.260 We can actually fill the gap a little bit

1234 00:56:22.260 --> 00:56:26.060 with some extra, extra tuning.

1235 00:56:26.060 --> 00:56:26.953 Just interesting.

1236 00:56:28.690 --> 00:56:29.550 Okay.

1237 00:56:29.550 --> 00:56:30.545 Yeah.

1238 00:56:30.545 --> 00:56:32.270 So this last part is really about just pushing the limits,

1239 00:56:32.270 --> 00:56:35.410 trying to figure out what the best possible performance is.

1240 00:56:35.410 --> 00:56:36.450 Okay.

1241 00:56:36.450 --> 00:56:37.750 So just to wrap things up,

1242 00:56:38.590 --> 00:56:40.850 right we gave some new results here

1243 00:56:40.850 --> 00:56:43.470 that let you be very flexible with

1244 00:56:43.470 --> 00:56:46.000 the kinds of methods that you want to use.

1245 00:56:46.000 --> 00:56:48.980 They do a good job of exploiting this Cate structure

1246 00:56:48.980 --> 00:56:52.523 when it's there and don't lose much when it's not.

1247 00:56:53.620 --> 00:56:55.820 So we have this nice model, free Arab bound.

1248 00:56:56.730 --> 00:56:58.890 We also kind of for free to get

1249 00:56:58.890 --> 00:57:03.460 this nice general Oracle inequality did

1250 00:57:03.460 --> 00:57:05.690 some investigation of the best possible rates

1251 00:57:05.690 --> 00:57:06.523 of convergence,

1252 00:57:06.523 --> 00:57:07.560 the best possible mean squared error

1253 00:57:07.560 --> 00:57:09.310 for estimating conditional effects,

1254 00:57:10.540 --> 00:57:13.560 which again was unknown before.

1255 00:57:13.560 --> 00:57:15.210 These are the weekend weak cause conditions

1256 00:57:15.210 --> 00:57:16.730 that have appeared,

1257 00:57:16.730 --> 00:57:18.580 but it's still not entirely known whether

1258 00:57:18.580 --> 00:57:21.713 they are mini max optimal or not.

1259 00:57:22.890 --> 00:57:24.490 So, yeah, big picture goals.

1260 00:57:24.490 --> 00:57:26.460 We want some nice flexible tools,

1261 00:57:26.460 --> 00:57:28.020 strong guarantees when it pushed forward,

1262 00:57:28.020 --> 00:57:30.390 our understanding of this problem.

1263 00:57:30.390 --> 00:57:32.350 I hope I've conveyed that there are lots of fun,

1264 00:57:32.350 --> 00:57:34.180 open problems here to work out

1265 00:57:34.180 --> 00:57:36.880 with important practical implications.

1266 00:57:36.880 --> 00:57:38.350 Here's just a list of them.

1267 00:57:38.350 --> 00:57:41.530 I'd be happy to talk more with people at any point,

1268 00:57:41.530 --> 00:57:43.890 feel free to email me a big part is applying

1269 00:57:43.890 --> 00:57:46.200 these methods in real problems.

1270 00:57:46.200 --> 00:57:48.530 And yeah, I should stop here,

1271 00:57:48.530 --> 00:57:52.580 but feel free to email the, the papers on archive here.

1272 00:57:52.580 --> 00:57:54.630 I'd be happy to hear people's thoughts.

1273 00:57:54.630 --> 00:57:55.463 Yeah.

1274 00:57:55.463 --> 00:57:56.296 Thanks again for inviting me.

1275 00:57:56.296 --> 00:57:57.760 It was fun.

1276 00:57:57.760 --> 00:57:58.593 - Yeah.

1277 00:57:58.593 --> 00:57:59.426 Thanks Edward.

1278 00:57:59.426 --> 00:58:02.050 That's a very nice talk and I think we're hitting the hour,

1279 00:58:02.050 --> 00:58:03.660 but I want to see in the audience

1280 00:58:03.660 --> 00:58:05.420 if we have any questions.

1281 00:58:05.420 --> 00:58:06.253 Huh.

1282 00:58:12.820 --> 00:58:13.653 All right.

1283 00:58:13.653 --> 00:58:15.730 If not, I do have one final question

1284 00:58:15.730 --> 00:58:16.563 if that's okay.

1285 00:58:16.563 --> 00:58:17.560 - Yeah, of course.

1286 00:58:17.560 --> 00:58:21.250 - And so I think there is a hosted literature

1287 00:58:21.250 --> 00:58:22.730 on flexible outcome modeling

1288 00:58:22.730 --> 00:58:26.150 to estimate conditional average causal effect,

1289 00:58:26.150 --> 00:58:28.297 especially those baits and non-parametric tree models

1290 00:58:28.297 --> 00:58:29.690 (laughs)

1291 00:58:29.690 --> 00:58:30.940 that are getting popular.

1292 00:58:31.820 --> 00:58:35.870 So I am just curious to see if you have ever thought

1293 00:58:35.870 --> 00:58:37.510 about comparing their performances,

1294 00:58:37.510 --> 00:58:40.000 or do you think there are some differences

1295 00:58:40.000 --> 00:58:42.250 between those sweats based

1296 00:58:42.250 --> 00:58:43.810 in non-parametric tree models versus

1297 00:58:43.810 --> 00:58:45.790 the plug-in estimator?

1298 00:58:45.790 --> 00:58:48.490 We compared in a simulation study here?

1299 00:58:48.490 --> 00:58:49.476 - Yeah.

1300 00:58:49.476 --> 00:58:50.309 I think of them

1301 00:58:50.309 --> 00:58:52.810 as really just versions of that plugin estimator

1302 00:58:52.810 --> 00:58:54.720 that use a different regression procedure.

1303 00:58:54.720 --> 00:58:58.281 There may be ways to tune plugins to try

1304 00:58:58.281 --> 00:59:01.360 and exploit this special structure of the Cate.

1305 00:59:01.360 --> 00:59:02.460 But if you’re really just looking

1306 00:59:02.460 --> 00:59:04.670 at the regression functions individually,

1307 00:59:04.670 --> 00:59:06.880 I think these would be susceptible to the same kinds

1308 00:59:06.880 --> 00:59:09.040 of issues that we see with the plugin.

1309 00:59:09.040 --> 00:59:09.873 Yeah.

1310 00:59:09.873 --> 00:59:10.706 That’s a good one.

1311 00:59:10.706 --> 00:59:11.539 - I see.

1312 00:59:11.539 --> 00:59:12.830 Yep.

1313 00:59:12.830 --> 00:59:16.640 So I want to see if there’s any further questions

1314 00:59:16.640 --> 00:59:19.293 from the audience to dr. Kennedy.

1315 00:59:21.494 --> 00:59:22.894 (indistinct)

1316 00:59:22.894 --> 00:59:26.260 - I was just wondering if you could speak a little more,

1317 00:59:26.260 --> 00:59:29.097 why the standard like naming orthogonality results

1318 00:59:29.097 --> 00:59:31.213 or can it be applicable in this setup?

1319 00:59:32.130 --> 00:59:33.073 - [Edward] Yeah.

1320 00:59:33.073 --> 00:59:33.906 (clears throat)

1321 00:59:33.906 --> 00:59:34.739 Yeah.

1322 00:59:34.739 --> 00:59:35.572 That’s a great question.

1323 00:59:35.572 --> 00:59:39.840 So one way to S to say it is that these effects,

1324 00:59:41.890 --> 00:59:42.740 these conditional effects

1325 00:59:42.740 --> 00:59:44.503 are not Pathwise differentiable.

1326 00:59:46.080 --> 00:59:49.950 And so these kinds of there's some distinction

1327 00:59:49.950 --> 00:59:51.160 between naming orthogonality

1328 00:59:51.160 --> 00:59:52.140 and pathways differentiability,

1329 00:59:52.140 --> 00:59:53.210 but maybe we can think about them

1330 00:59:53.210 --> 00:59:54.910 as being roughly the same for now.

1331 00:59:56.580 --> 00:59:59.200 So yeah, all the standards in my parametric

1332 00:59:59.200 --> 01:00:01.140 theory breaks down here

1333 01:00:01.140 --> 01:00:03.710 because of this lack of pathways differentiability so the,

1334 01:00:03.710 --> 01:00:04.870 all the efficiency bounds that

1335 01:00:04.870 --> 01:00:06.823 we know and love don't apply,

1336 01:00:08.550 --> 01:00:10.960 but it turns out that there's some kind

1337 01:00:10.960 --> 01:00:14.550 of analogous version of this that works for these things.

1338 01:00:14.550 --> 01:00:17.610 I think of them as like infinite dimensional functional.

1339 01:00:17.610 --> 01:00:20.300 So instead of like the ate, which is just a number,

1340 01:00:20.300 --> 01:00:22.340 this is like a curve,

1341 01:00:22.340 --> 01:00:25.450 but it has the same kinds of like functional structure

1342 01:00:25.450 --> 01:00:27.660 in the sense that it's combining regression functions

1343 01:00:27.660 --> 01:00:29.417 or our propensity scores in some way.

1344 01:00:29.417 --> 01:00:32.530 And we don't care about the individual components.

1345 01:00:32.530 --> 01:00:34.130 We care about their combination.

1346 01:00:36.170 --> 01:00:38.070 So yeah, the standard stuff doesn't work just

1347 01:00:38.070 --> 01:00:39.710 because it's, we're outside of this route

1348 01:00:39.710 --> 01:00:43.900 in Virginia, roughly, but there are, yeah,

1349 01:00:43.900 --> 01:00:46.030 there's analogous structure and there's tons

1350 01:00:46.030 --> 01:00:47.840 of important work to be done, 1351 01:00:47.840 --> 01:00:52.000 sort of formalizing this and extending

1352 01:00:53.290 --> 01:00:55.390 that's a little vague, but hopefully that.

1353 01:01:01.920 --> 01:01:02.753 - All right.

1354 01:01:02.753 --> 01:01:04.653 So any further questions?

1355 01:01:08.440 --> 01:01:09.273 - Thanks again.

1356 01:01:09.273 --> 01:01:10.402 And yeah.

1357 01:01:10.402 --> 01:01:12.460 If any questions come up, feel free to email.

1358 01:01:12.460 --> 01:01:13.596 - Yeah.

1359 01:01:13.596 --> 01:01:14.429 If not,

1360 01:01:14.429 --> 01:01:15.840 I'll let smoke unless that doctors can be again.

1361 01:01:15.840 --> 01:01:16.920 And I'm sure he'll be happy

1362 01:01:16.920 --> 01:01:18.730 to answer your questions offline.

1363 01:01:18.730 --> 01:01:20.010 So thanks everyone.

1364 01:01:20.010 --> 01:01:20.843 I'll see you.

1365 01:01:20.843 --> 01:01:21.790 We'll see you next week.

1366 01:01:21.790 --> 01:01:22.623 - Thanks a lot.