Quant II

Machine Learning and Optimization

Ye Wang

4/25/2018

Machine learning and optimization in social sciences

- How could machine learning be applied to social sciences?
  - Variable creation: train a model and use it to code data
  - Methods: prediction and heterogeneity
- Optimization has two meanings:
  - Causal inference as an optimization problem
  - Optimize your R code

Variable creation

- The problem: transform texts, images, audio, or even video into variables

Variable creation

- The problem: transform texts, images, audio, or even video into variables
- A common solution: 1. hire workers to code part of it, 2. train a model, 3. let the machine do the rest

Variable creation

- The problem: transform texts, images, audio, or even video into variables
- A common solution: 1. hire workers to code part of it, 2. train a model, 3. let the machine do the rest
- Example I: forest on Google maps (Burgess et al., 2012)

Variable creation

- The problem: transform texts, images, audio, or even video into variables
- A common solution: 1. hire workers to code part of it, 2. train a model, 3. let the machine do the rest
- Example I: forest on Google maps (Burgess et al., 2012)
- Example II: topics in Chinese social media posts (King et al., 2013, 2014)

Variable creation

- The problem: transform texts, images, audio, or even video into variables
- A common solution: 1. hire workers to code part of it, 2. train a model, 3. let the machine do the rest
- Example I: forest on Google maps (Burgess et al., 2012)
- Example II: topics in Chinese social media posts (King et al., 2013, 2014)
- Example III: use images to detect protests (Zhang and Pan, 2019; Donghyeon et al., 2019)

Variable creation

- The problem: transform texts, images, audio, or even video into variables
- A common solution: 1. hire workers to code part of it, 2. train a model, 3. let the machine do the rest
- Example I: forest on Google maps (Burgess et al., 2012)
- Example II: topics in Chinese social media posts (King et al., 2013, 2014)
- Example III: use images to detect protests (Zhang and Pan, 2019; Donghyeon et al., 2019)
- Example IV: audio and video processing (Know and Lucas, 2018)

Variable creation

- The problem: transform texts, images, audio, or even video into variables
- A common solution: 1. hire workers to code part of it, 2. train a model, 3. let the machine do the rest
- Example I: forest on Google maps (Burgess et al., 2012)
- Example II: topics in Chinese social media posts (King et al., 2013, 2014)
- Example III: use images to detect protests (Zhang and Pan, 2019; Donghyeon et al., 2019)
- Example IV: audio and video processing (Know and Lucas, 2018)
- Example V: identify Russian bots on Twitter (Stukal et al., 2017)

Variable creation

- The problem: transform texts, images, audio, or even video into variables
- A common solution: 1. hire workers to code part of it, 2. train a model, 3. let the machine do the rest
- Example I: forest on Google maps (Burgess et al., 2012)
- Example II: topics in Chinese social media posts (King et al., 2013, 2014)
- Example III: use images to detect protests (Zhang and Pan, 2019; Donghyeon et al., 2019)
- Example IV: audio and video processing (Know and Lucas, 2018)
- Example V: identify Russian bots on Twitter (Stukal et al., 2017)
- What if it is too expensive? Active learning (Miller et al., 2019)

Methods: prediction

- Some relationships in causal inference can be non-causal
- We just need to fit/predict it with a high accuracy
  - Example I: Propensity score
  - Example II: Stage one of IV
  - Example III: Response surface

Methods: prediction

- Some relationships in causal inference can be non-causal
- We just need to fit/predict it with a high accuracy
  - Example I: Propensity score
  - Example II: Stage one of IV
  - Example III: Response surface

- What if we just throw a bunch of variables into the first stage?

Methods: prediction

- Some relationships in causal inference can be non-causal
- We just need to fit/predict it with a high accuracy
  - Example I: Propensity score
  - Example II: Stage one of IV
  - Example III: Response surface

- What if we just throw a bunch of variables into the first stage? - Bias in both estimation and inference (Cattaneo et al., 2019)

Example: double selection

```
## Loading required package: hdm
```

```
## [1] 100 100
```

```
## [1] 100   3
```

Methods: heterogeneity

- Heterogeneity just means a relationship between the effect and the moderators: $TE = f(\mathbf{X})$.

Methods: heterogeneity

- Heterogeneity just means a relationship between the effect and the moderators: $TE = f(\mathbf{X})$.
- It is thus a machine learning problem.

Methods: heterogeneity

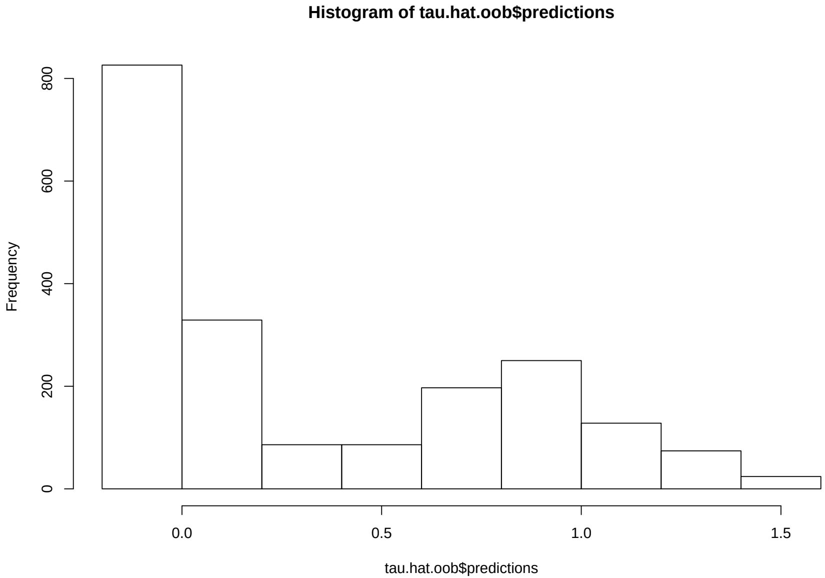
- Heterogeneity just means a relationship between the effect and the moderators: $TE = f(\mathbf{X})$.
- It is thus a machine learning problem.
- Example I: Trees and forests

Methods: heterogeneity

- Heterogeneity just means a relationship between the effect and the moderators: $TE = f(\mathbf{X})$.
- It is thus a machine learning problem.
- Example I: Trees and forests
- Example II: X-learner

Example: causal forest

```
## Loading required package: grf
```



Methods: new directions

- Experiment + gradient descent (Wager and Kuang, 2019)

Methods: new directions

- Experiment + gradient descent (Wager and Kuang, 2019)
- What is the optimal price offered to drivers for UBER?

Methods: new directions

- Experiment + gradient descent (Wager and Kuang, 2019)
- What is the optimal price offered to drivers for UBER?
- Drivers decide whether to work after observing the price (an inverted U-shape profit curve).

Methods: new directions

- Experiment + gradient descent (Wager and Kuang, 2019)
- What is the optimal price offered to drivers for UBER?
- Drivers decide whether to work after observing the price (an inverted U-shape profit curve).
- Assign each driver $p_i = p_0 + \varepsilon_i$.

Methods: new directions

- Experiment + gradient descent (Wager and Kuang, 2019)
- What is the optimal price offered to drivers for UBER?
- Drivers decide whether to work after observing the price (an inverted U-shape profit curve).
- Assign each driver $p_i = p_0 + \varepsilon_i$.
- Calculate the slope of tangent at p_0, and do gradient descent to find p_1.

Methods: new directions

- Experiment + gradient descent (Wager and Kuang, 2019)
- What is the optimal price offered to drivers for UBER?
- Drivers decide whether to work after observing the price (an inverted U-shape profit curve).
- Assign each driver $p_i = p_0 + \varepsilon_i$.
- Calculate the slope of tangent at p_0, and do gradient descent to find p_1.
- Repeat until convergence.

Optimization

- Optimization is underlying many extant methods: weighting, matching, RD, SC, etc.

Optimization

- Optimization is underlying many extant methods: weighting, matching, RD, SC, etc.
- Most estimators generate estimates by weighting the outcome variable.

Optimization

- Optimization is underlying many extant methods: weighting, matching, RD, SC, etc.
- Most estimators generate estimates by weighting the outcome variable.
- We want to balance the accuracy and parsimony of weighting.

Optimization

- Optimization is underlying many extant methods: weighting, matching, RD, SC, etc.
- Most estimators generate estimates by weighting the outcome variable.
- We want to balance the accuracy and parsimony of weighting.
- It is a machine learning problem as well as an optimization problem.

Optimization

- Optimization is underlying many extant methods: weighting, matching, RD, SC, etc.
- Most estimators generate estimates by weighting the outcome variable.
- We want to balance the accuracy and parsimony of weighting.
- It is a machine learning problem as well as an optimization problem.
- The most common approach: convex optimization:

$$
\begin{aligned}
&\min\ f(\mathbf{x}) \\
&\text{s.t. } g_k(\mathbf{x}) \leq 0 \\
&h_k(\mathbf{x}) = 0
\end{aligned}
$$

Optimization

- Optimization is underlying many extant methods: weighting, matching, RD, SC, etc.
- Most estimators generate estimates by weighting the outcome variable.
- We want to balance the accuracy and parsimony of weighting.
- It is a machine learning problem as well as an optimization problem.
- The most common approach: convex optimization:

$$
\begin{aligned}
&\min \; f(\mathbf{x}) \\
&\text{s.t. } g_k(\mathbf{x}) \leq 0 \\
&h_k(\mathbf{x}) = 0
\end{aligned}
$$

- How to solve? Simplex method.

Optimization

Example I: entropy balancing

```
## ##
## ## ebal Package: Implements Entropy Balancing.

## ## See http://www.stanford.edu/~jhain/ for additional in

## Converged within tolerance

## treatment mean

##        x1        x2        x3
## 0.7806424 0.6152861 0.5820738

## weighted control mean

##        x1        x2        x3
## 0.7761886 0.6148558 0.5801141

## unweighted control mean
```

Optimization

Let's do it using convex optimization!

```
## Loading required package: CVXR

##
## Attaching package: 'CVXR'

## The following object is masked from 'package:stats':
##
##     power

## treatment mean

##        x1        x2        x3
## 0.8122392 0.4125846 0.5765168

## weighted control mean

##        x1        x2        x3
## 0.8122392 0.4125846 0.5765168
```

Optimization

- Example II: optimal bandwidth of RD

$$\hat{\tau} = \sum_{i=1}^{n} \hat{\gamma}_i Y_i, \quad \hat{\gamma} = \operatorname{argmin}_{\gamma} \left\{ \sum_{i=1}^{n} \gamma_i^2 \sigma_i^2 + I_B^2(\gamma) \right\},$$

$$I_B(\gamma) := \sup_{\mu_0(\cdot), \mu_1(\cdot)} \left\{ \sum_{i=1}^{n} \gamma_i \mu_{W_i}(X_i) - (\mu_1(c) - \mu_0(c)) : |\mu_w''(x)| \leq B \text{ for all } w,\, x \right\}.$$

Optimization

- **Example III**: evolutionary tree

Optimization

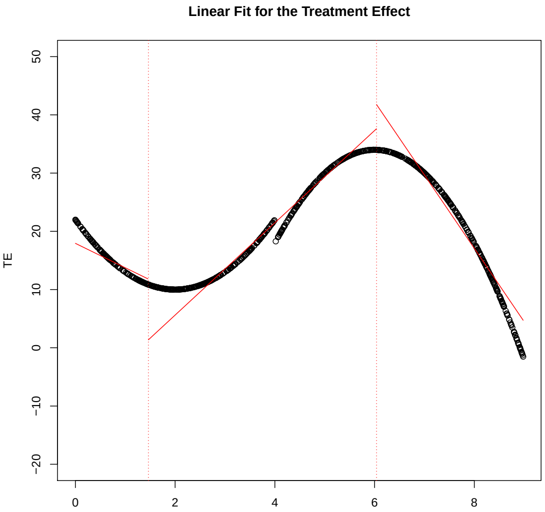
- **Example III**: evolutionary tree

Optimization

- Example III: evolutionary tree
  How to optimally partition an interval?

Optimization

- Example III: evolutionary tree
  How to optimally partition an interval?
  Generate 10,000 partitions and let them "evolve"...



Optimize your R code

- The R Inferno (Don't grow matrices!)

Optimize your R code

- The R Inferno (Don't grow matrices!)
- Parallel computing

Optimize your R code

- The R Inferno (Don't grow matrices!)
- Parallel computing
- Rcpp

Optimize your R code

- The R Inferno (Don't grow matrices!)
- Parallel computing
- Rcpp
- NYU clusters

Example: parallel computing

```
## Loading required package: doParallel

## Loading required package: foreach

## Loading required package: iterators

## Loading required package: parallel

## Loading required package: microbenchmark

## Parallel computing with 4 cores...

## Time difference of 4.251918 secs

## Time difference of 0.7286451 secs
```

The End

Good luck with your final!