Machine Learning and Economics

Adam Giles

Financial Conduct Authority

27/09/2018

Summary

This talk will cover:

- What is machine learning
- Using prediction for policy
- Applications to causal problems

What is machine learning?

Definitions

From a computer scientist[1] in 1959:

> *Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed.*

...but that's pretty vague. In practice, three broad areas:

- **Supervised Learning**: using data to fit models and predict outcomes. Useful in economics as a policy tool, and to get the most out of datasets for causal estimation problems.
- **Unsupervised Learning**: finding structure in data. Useful in economics to enable analysis of unstructured data — especially text.
- **Reinforcement Learning**: optimising a payoff function. Useful in economics to simulate strategic behaviour.

We're going to talk about supervised learning because (so far!) it has the broadest applications in economics and econometrics.

[1] Arthur Samuelson

Why has this become popular?

Most of the methods used today are not new:

- Neural Networks: **1961**
- Classification and Regression Trees: **1984**
- LASSO: **1996**
- Random Forests: **2001**

The difference is technology:

- availability of data
- computing power

Trivia: The most well known machine learning method, neural networks, had some econometric interest in the 1990s. Halbert White, Jeffrey Wooldridge, and others published quite a lot before they fell out of favour for being too hard to fit.

Differences with econometrics

Machine Learning

- Focus on $\hat{Y}$
- Minimise prediction *error*
- Care about:
  - out of sample performance
  - speed

Econometrics

- Focus on $\hat{\beta}$
- Minimise *bias* (usually)
- Care about:
  - formal statistical properties
  - inference

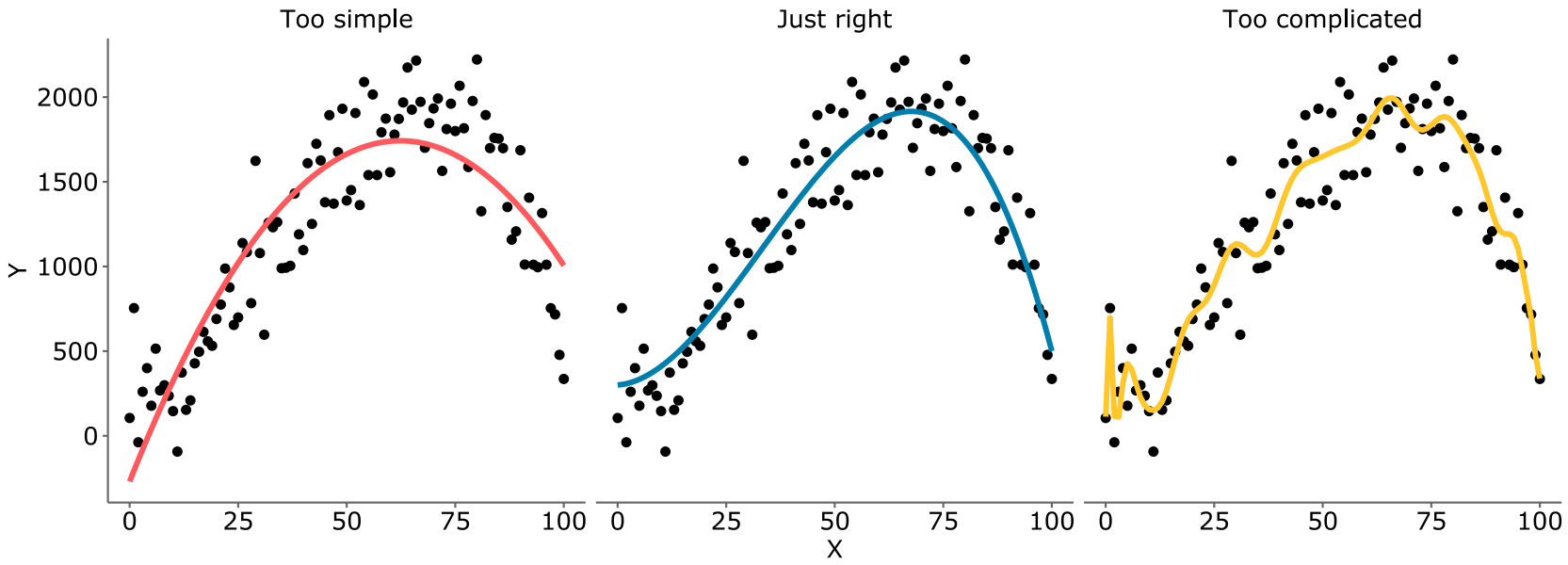
Like many applied branches of statistics, names for things are also quite different. Eg,

- models are "trained" rather than estimated
- dummy/indicator variables are "one-hot encodings"
- regressors are "features"
- coefficients in linear models are (sometimes) "weights"
- limited dependent variable problems are "classification"
- and so on...

Fitting predictive models

The challenge is striking a balance between flexibly modelling the data *without* ending up modelling the noise in the sample you're estimating on

This is the **bias-variance tradeoff**



Machine learning methods

Over the next couple of slides we'll talk through the principles of two key supervised machine learning methods:

- Penalised linear models
- Regression trees

Linear models

You want to fit a model like this:

$$Y = X\beta + \epsilon$$

General approach to this is to maximise the fit of the model to the sample data. Or,

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}\, L(\beta|X, Y)$$

For some loss function L. (Eg OLS, Logit, Probit, ...)

Penalised linear models

Models like OLS will tend to assign some predictive power to all variables in X

If some variables in X aren't really relevant to explain Y, then the model will be overfit

Penalised regression models address this by adding another term to the minimisation problem:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left(L(\beta | X, Y) + \lambda P(\beta) \right)$$

Where P is some function that increases with the size of the coefficients

This forces a tradeoff between assigning predictive power to a variable and improving the model's in-sample fit. This is a form of **regularisation**

Penalty terms

What to use for P? Three main options:

- LASSO: sum of absolute values of coefficients
  - $\sum |\beta_j|$
  - Penalizes all coefficient values equally
- Ridge: sum of squared coefficients
  - $\sum \beta_j^2$
  - Penalizes large coefficient values more harshly, and small coefficients less harshly
- Elastic Net: weighted sum of LASSO and Ridge.
  - $\alpha \sum |\beta_j| + (1 - \alpha) \sum \beta_j^2$
  - Penalizes small and large coefficient values more harshly

LASSO

LASSO is "Least Absolute Shrinkage and Selection Operator"

The kink it creates in the penalty function around zero results in a "*sparse*" model - ie, some coefficients can be exactly zero

This means you can fit models with more covariates than observations(!)

In practice lots of linear models economists fit are at least approximately sparse. Especially if you take into account "derived regressors"; transforms, interactions and so on. Example:

- Linear model with 10 raw regressors in X
- 2 functional transforms (linear and log, say)
- up to 3rd interaction between transformed variables
- You now have $C_3^{20} + C_2^{20} + 20 = 1350$ regressors

You can use LASSO for model selection, then run (say) OLS on the selected variables (usually called post-LASSO)

Regression Trees

More general problem. You want to model:

$$Y = G(X) + \epsilon$$

Without saying much about the form of G.

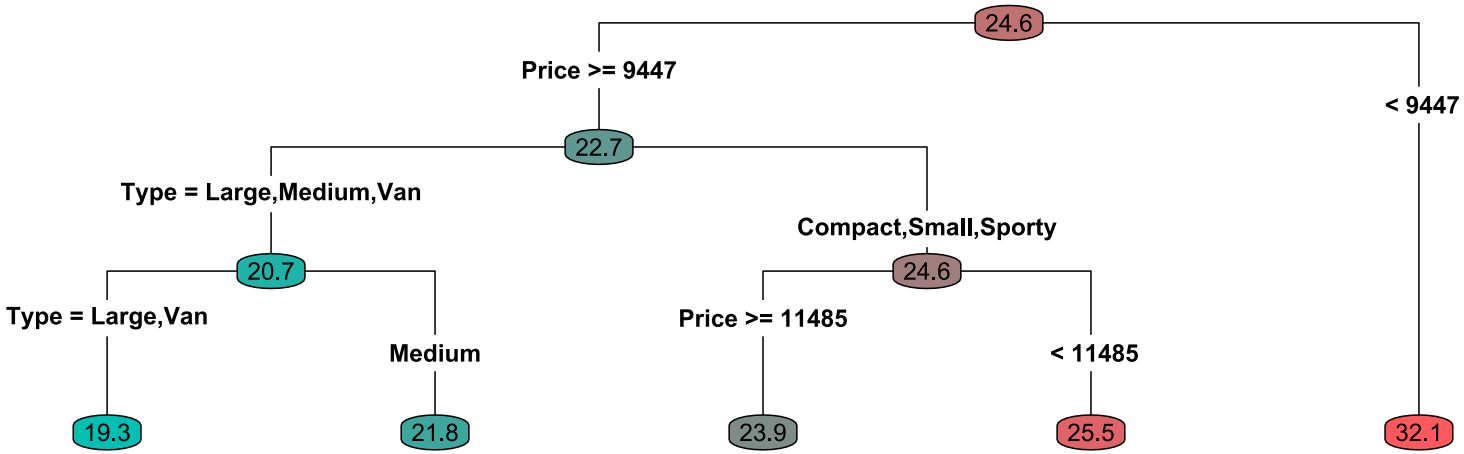
Regression Trees approach this with **recursive partitioning**.

The idea is to keep splitting the data into increasingly "pure" subsets.

This can be used for continuous outcomes ("regression") or logit/multinomial logit type problems ("classification").

Regression Tree Example

This is a model of vehicle MPG as a function of: Price, Country of origin, Reliability rating, and Vehicle type



The model has found Price and Vehicle Type the only useful variables to partition on

Regularising trees

In theory best to grow tree fully, then prune branches that didn't improve fit enough

In practice much faster and not much worse performance-wise to rely on stopping rules. Eg,

- Minimum number of samples in leaf node
- Minimum improvement in fit
- Maximum depth threshold

The example on the previous page just used the default stopping rules from the package used to estimate it (`rpart`).

For a an excellent visual introduction to regression tree modelling see here

Hyperparameters

Essentially all supervised learning methods have the same formulation:

- A function that fits to the data
- A regularisation component that constrains complexity

The parameters that control the regularisation can't be estimated with in-sample fit alone. In our examples:

Penalised regression:

- The choice of λ
- The choice of P

Regression trees:

- Which stopping rules to use
- Calibration of stopping rules

We need a way to estimate these parameters — "cross-validation"

Cross-validation

Cross-validation is sample splitting. Typical method:

1. Split data into 5 **"folds"**[1]
2. For each fold, fit the model on the combination of the other 4, and predict for this fold
3. Measure model performance on the set of out of sample predictions

Intuition: You want to maximise out of sample performance, so mimic that

Using this method for evaluation, you can search for regularisation parameters that provide the best possible out of sample performance

Despite being intuitive, the formal properties of this approach to tuning parameters are not known for many models. LASSO is an exception.

1: There's nothing special about 5. Most people use 3-10. With small numbers of observations you might even use N, this is called Leave-One-Out-Cross-Validation or LOOCV.

Parameter search

Approaches to parameter search:

- **Grid**: specify steps across range of parameters, cross-validate all combinations
- **Random**: specify distributions across parameters, repeatedly sample new parameters and cross-validate
- **Bayesian Optimisation**: specify priors across parameters, cross-validate selection, update priors and test the next combination

Grids are the most common approach; can be very time consuming with many parameters.

Meta-methods

Bootstrap Aggregation ("bagging"): fit many models across bootstrapped samples of observations. Average across individual estimates for final prediction. Very common with tree models — a "Random Forest" is bagged trees.

Boosting: fit sequence of simple models to the residuals of prior models. Very powerful. Boosted regression trees win many machine learning competitions.

Stacking: Build many different types of model, store predictions. Then feed these as inputs to another model to produce the final estimate. Not common in production environments, but wins many contests.

Pipelines: in practice your data analysis is probably a flow consisting of many parameters (eg, where to winsorize a variable) that aren't estimated. Building the entire process as a pipeline enables you to optimize over all those choices as well through cross-validation.

...and combinations of the above

Some practical advice

- Use Python or R — Stata, SAS, Matlab etc just don't compare
  - The Python package `scikit-learn` in particular is excellent
  - If you've not used either before, Python is easier
- You can get very far with modern linear models
- Random Forests are powerful and easy to implement. Regression Trees are not very good
- Boosting is incredibly powerful and fairly easy to implement
  - The packages `xgboost`, `lightgbm` (Microsoft), and `catboost` (Yandex) all do this very well
- Neural Networks are rarely worth the extra effort with the type of data most economists use
  - They're really superior as soon as you have data like images, sound, raw text

Using predictions for policy

Causation vs Prediction

Economists are generally very interested in causality. And we should be!

But lots of policy problems have a prediction component to them as well:

- **Causal** question: how do consumers get into financial distress?
- **Predictive** question: which consumers will become financially distressed?

For an excellent introduction to this topic see Kleinberg et al. (2015) in the references.

Prediction for measurement

As policymakers we often care about outcomes that might be difficult or expensive to measure

While we have access to detailed administrative data, the actual thing we want to measure is not included — for example, consumer vulnerability

But if we can gather data on this outcome for a sample of those we observe administrative data for, we can fit a predictive model from administrative data to the outcomes, and estimate what we want to measure for the rest of the population

This could be used purely as a measurement device, or to target policies across the population as a whole

This is popular and successful in development economics. For example using luminosity data from satellites (Henderson, Storeygard, and Weil (2012)) or mobile phone records (Blumenstock, Cadamuro, and On (2015)) to measure economic growth and poverty.

Prediction for allocation

We have substantial but finite capacity for supervision and enforcement

Using past records on firm outcomes and the information we hold about them, we can predict the risk of a firm making some kind of conduct violation

We can use those risk predictions to allocate supervision and enforcement resources

This type of approach has been shown to be effective in various policy settings:

- Medicine
- Policing and law and order
- Social policy
- ...and so on. See the references for examples

Prediction for intervention

In settings where outcomes develop over time, prediction may enable early intervention

In retail financial services markets for example, negative outcomes like serious financial difficulty may be predicatable before they actually happen

While prediction alone wouldn't tell us what an effective remedy would be, being able to act *before* something significant happens would expand the range of options available to deal with it

Making mistakes

Access to large detailed datasets can be deceptive[1]

Even well validated models can fail when taken to new data — "old" statistical modelling problems don't go away. Eg,

- Unrepresentative sampling
- Changes in environment
- Bias in the raw data

Adverse selection in credit or insurance are good examples of this type of problem

The issue here is really about understanding what your source data is, and its relevance to whatever you're interested in predicting

1: Meng (2018) develops various ideas about the tradeoff between data quality and data quantity. Even the abstract is a good read.

Fairness (I)

Many applications of prediction in economic policy relate to people

Machine learning models are optimised for prediction — they don't care about anything else. But you probably do!

In particular, in focusing on raw predictive power machine learning models may:

- effectively proxy for particular protected characteristics that are correlated with the outcome and other controls
- find that some groups are easier to predict outcomes for than others

That can have implications for the fairness of outcomes they produce

- Fairness constrained prediction is a very active area of research
- Tools like aequitas are also becoming available to audit off the shelf models

But this can still be nuanced. Fuster, et al. (2017) use an application to US mortgage data to explore the impact of improved statistical technology. In their case more effective methods result in increased access to credit for all groups but more unequal outcomes between groups.

Fairness (II)

Excluding protected characteristics from the modelling does not solve the fairness problem. For example:

- Gender cannot be used to price car insurance
- Men and women have (on average) different preferences over car colours
- Is it fair to use car colour to price car insurance?

Fairness is hard to define in terms of statistical properties. Which do you care about more:

- False negatives?
- False positives?

Competing definitions may well be impossible to reconcile within one problem. See Kleinberg, Mullainathan, and Raghavan (2017)

Applications to causal problems

Introduction

We've talked about machine learning applications to prediction

But all those problems are cases where we care about $\hat{Y}$, and the $\hat{\beta}$s (or equivalents) were effectively nuisance parameters

Machine learning methods are not designed for us to make any real inferences about βs alone

We need to apply them carefully to use them for the types of (quasi-)causal problem that economists care about, where parameters rather than outcomes are of interest

Motivation

> There are two types of statisticians: those who do causal inference and those who lie about it [1, 2]

1: Larry Wasserman
2: Courtesy of Edward Kennedy's twitter; @edwardhkennedy

Why you need to be careful

Many methods do not provide any means of making inferences about the parameters

Even where they do, interpretation is very difficult — how would you interpret a penalised linear model?

The regularisation in particular causes problems:

- some variables might be regularised away not because they have no relationship with $\hat{Y}$, but because their relationship is weak
- where variables are correlated, they are substitutes from a prediction perspective

ML-powered causal estimation

But sometimes the βs we want to estimate involve some kind of predictive step

This is actually very common in causal inference/estimation problems that come up often in microeconomics:

- **Instrumental Variables**: main step is fitting relationship from exogenous instrument to endogenous covariate
- **Propensity Score Matching**: main step is estimating $Pr(Treated|X)$
- **Regression adjustment**: estimate counterfactual *outcome* using data for other treatment group
- **Doubly-robust methods**: combination of propensity score matching and regression adjustment
- **Difference-in-Difference**: same principle as regression adjustment

In these settings you can, with some caveats, essentially plug in whatever machine learning methods you like to do these parts of the estimation.

Double selection

A very common problem in econometrics is this:

$$Y = D\theta + X\beta + \epsilon$$

Where θ is the parameter of interest, and X contains things you want to control for because you think they *might* either:

- Be correlated with D and Y so confound the estimate of θ
- Influence Y and so including them would improve precision

Chernozhukov et al. (2015) show that the following procedure gives valid estimates and confidence intervals for θ.

1. Regress Y on X using LASSO
2. Regress D on X using LASSO
3. Regress Y on D and all X variables selected in 1. or 2. using OLS

This is a very powerful result! Given a large set of potentially relevant control variables you can very robustly select a linear model *and* make valid inferences about θ

Back to the 1930s

The Frisch-Waugh-Lovell theorem shows that you can decompose a problem like the one on the previous slide like so:

1. Regress Y on X, store residuals
2. Regress D on X, store residuals
3. Regress the residuals from 1. on the residuals from 2.

The coefficient from 3. is numerically equivalent to what would you get from an OLS regression of Y on D and X together

This is usually called estimation via partialling out

Double machine learning

Chernozhukov et al. (2017) shows that the the Frisch-Waugh-Lovell result will hold for any 'high-quality' estimator in steps 1. and 2., not just OLS.

This allows you to apply it to treatment/causal problems like this:

$$Y = D\theta + G(X) + \epsilon$$

$$D = P(X) + \eta$$

And estimate G and P with any high-quality machine learning method you want.[1]

1: high-quality is (roughly) shorthand for an estimator that is consistent, converges at at least $n^{\frac{1}{4}}$, and uses cross-fitting to prevent overfitting G or P to the sample. The full definition is in the paper but is heavy going.

Heterogenous treatment effects

When testing or evaluating policies it is common to look at their average effect.

In reality policies affect people in different ways, but robustly estimating the heterogenous effects is very hard. Just trying lots of different subgroup estimates causes serious problems with inference.[1]

Recent reseach tries to address these problems by adapting machine learning methods to target this type of problem, rather than raw prediction. This is a very active area:

- Work by Susan Athey and coauthors focuses on adapting tree and forest models to directly estimate treatment effects conditional on covariates
- Work by Victor Chernozhukov and coauthors looks at estimating simpler approximations of true heterogenous effects using generic machine learning methods
- Work by Xinkun Nie and Stefan Wager proposes a two step procedure to estimate heterogenous effects even with observational rather than experimental data

1: In some disciplines (eg, medicine) this problem is so severe that journals require pre-registering analysis plans to try and ensure robustness of results

Optimal policies

A key reason for trying to understand effect heterogeneity is trying to build optimal policies

If we can understand when treatments do or don't work for different types of person or situation, we can combine them into a super-treatment that works better than any individual treatment

Sometimes that might mean only selectively implementing a policy — it's possible for a policy to have a population effect of zero, but still work well for some subgroup

This type of approach would present many challenges in implementation. Eg,

- How would you convert conditional average treatment effects from complex models into a rule...?
- Could it update over time...?

Conclusions

Conclusions

As economists you're well equipped to understand and apply supervised ML

Prediction has direct applications in policy settings:

- Doing it well isn't necessarily easy
- Fairness is an important consideration

With large or complicated datasets, supervised ML methods can help identify causal relationships, including heterogenous effects

Questions?

Adam Giles

Technical Specialist, Economics

Adam.Giles@fca.org.uk

FINANCIAL CONDUCT AUTHORITY

References: recommended

Athey, S. (2018) The Impact of Machine Learning on Economics in: *The Economics of Artificial Intelligence: An Agenda*. University of Chicago Press (forthcoming)

Mullainathan, S. & Speiss, J. (2015) Machine Learning: An Applied Econometric Approach, *Journal of Economic Perspectives*. 31 (2) pp. 87-106. Available from: 10.1257/jep.31.2.87

Varian, H. (2014) Big Data: New Tricks for Econometrics *Journal of Economic Perspectives*, 28 (2) pp. 3-28. Available from: 10.1257/jep.28.2.3

Kleinberg, J., J. Ludwig, S. Mullainathan, and Z. Obermeyer. (2015). Prediction Policy Problems, *American Economic Review*, 105 (5) pp. 491-95. Available from: 10.1257/aer.p20151023

Belloni, A., Chernozhukov, V. and Hansen, C., (2014) High-Dimensional Methods and Inference on Structural and Treatment Effects *Journal of Economic Perspectives*, 28 (2) pp. 29-50. Available from 10.1257/jep.28.2.29

References: machine learning methods

James, G., Witten, D., Hastie, T., Tibshirani, R., (2013) *An Introduction to Statistical Learning with Applications in R* Springer

Hastie, T., Tibshirani, R., Friedman, J., (2009) *The Elements of Statistical Learning* 2nd Edition Springer

References: prediction for policy (I)

Henderson, J. Vernon, Storeygard, A., and Weil., D. N. (2012) Measuring Economic Growth from Outer Space. *American Economic Review* 102 (2) pp. 994–1028.

Blumenstock, J., Cadamuro, G., & On, R. (2015) Predicting poverty and wealth from mobile phone metadata. *Science*, 350 (6264) pp. 1073-1076.

Alaa, A. M., and van der Schaar, M. (2018) Prognostication and Risk Factors for Cystic Fibrosis via Automated Machine Learning. *Scientific reports*, 8 (1), 11242.

McCue, C., Parker, A., McNulty, P., and McCoy, D. (2004) Doing More with Less: Data Mining in Police Deployment Decisions *United States Department of Justice Violent Crime Newsletter* Spring 2004, pp. 4–5

References: prediction for policy (II)

Ghosh, D. (2007) Predicting vulnerability of indian women to domestic violence incidents *Research and Practice in Social Sciences*, 3 (1) pp. 48-72.

Meng, Xiao-Li. (2018) Statistical paradises and paradoxes in big data (I): Law of large populations, big data paradox, and the 2016 US presidential election. *Annals of Applied Statistics* 12 (2), pp. 685-726. Available from: 10.1214/18-AOAS1161SF

Fuster, A., Goldsmith-Pinkham, P., Ramadorai, A., and Walther, A. (2018) Predictably Unequal? The Effects of Machine Learning on Credit Markets *Working paper*. Available from: 10.2139/ssrn.3072038

Kleinberg, J., Mullainathan, S., and Raghavan, M., (2017) Inherent Trade-Offs in the Fair Determination of Risk Scores *Proceedings of the 8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, 43:1-43:23. Available from: 10.4230/LIPIcs.ITCS.2017.43

References: causal inference (I)

Belloni, A., Chen, D., Chernozhukov, V., and Hansen, C. (2012) Sparse Models and Methods for Optimal Instruments With an Application to Eminent Domain *Econometrica*, 80 (6) pp. 2369-2429. Available from: 10.3982/ECTA9626

Belloni, A., Chernozhukov, V., and Hansen, C. (2014) Inference on Treatment Effects after Selection among High-Dimensional Controls *Review of Economic Studies*, 81 (2) pp. 608-650. Available from: 10.1093/restud/rdt044

Frisch, R. and Waugh, F. (1933) Partial Time Regressions as Compared with Individual Trends *Econometrica*, 1 (4) pp. 387-401. Available from: 10.2307/1907330

Lovell, M. (1963) Seasonal Adjustment of Economic Time Series and Multiple Regression Analysis *Journal of the American Statistical Association*, 58 (304) pp. 993-1010. Available from: 10.2307/2283327

Chernozhukov, V. et al. (2017) Double/debiased machine learning for treatment and structural parameters *The Econometrics Journal*, 21 (1) pp. C1-C68. Available from: 10.1111/ectj.12097

References: causal inference (II)

Chernozhukov, V., Duflo, E., Demirer, M., and Fernandez-Val, I. (2018) Generic Machine Learning Inference on Heterogenous Treatment Effects in Randomized Experiments *Working paper*

Athey, S., and Imbens, G. (2016) Recursive partitioning for heterogenous causal effects *Proceedings of the National Academy of Sciences*, 113 (27) pp. 7353-7360. Available from: 10.1073/pnas.1510489113

Athey, S., and Wager, S. (2018) Estimation and Inference of Heterogeneous Treatment Effects using Random Forests *Journal of the American Statistical Association*. forthcoming. Available from: 10.1080/01621459.2017.1319839

Nie, X., and Wager, S. (2018) Quasi-Oracle Estimation of Heterogeneous Treatment Effects *Preprint*. Available from: arXiv:1712.04912v2

Hill, J. (2011) Bayesian Nonparametric Modeling for Causal Inference *Journal of Computational and Graphical Statistics*, 20 (1) pp. 217-240. Available from: 10.1198/jcgs.2010.08162