Machine Learning Methods Economists Should Know About*

Susan Athey† Guido W. Imbens‡

September 2018

Abstract

We discuss the relevance of the recent machine learning literature for economics and econometrics. First we discuss the differences in goals, methods and settings between the ML literature and the traditional econometrics and statistics literatures. Then we discuss some specific methods from the machine learning literature that we view as important for empirical researchers in economics. These include supervised learning methods for regression and classification, unsupervised learning methods as well as matrix completion methods. Finally, we highlight newly developed methods at the intersection of ML and econometrics, methods that typically perform better than either off-the-shelf ML or more traditional econometric methods when applied to particular classes of problems, problems that include causal inference for average treatment effects, optimal policy estimation, and estimation of the counterfactual effect of price changes in consumer choice models.

*We are grateful to Sylvia Klosin for comments.

†Professor of Economics, Graduate School of Business, Stanford University, SIEPR, and NBER, athey@stanford.edu.

‡Graduate School of Business and Department of Economics, Stanford University, SIEPR, and NBER. Electronic correspondence: imbens@stanford.edu.

1 Introduction

In the abstract of his provocative 2001 paper in *Statistical Science* the Berkeley statistician Leo Breiman writes about the difference between a model-based versus an algorithmic approach to statistics:

> "There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown." Breiman [2001b], p199.

He goes on to claim that:

> " The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems. Algorithmic modeling, both in theory and practice, has developed rapidly in fields outside statistics. It can be used both on large complex data sets and as a more accurate and informative alternative to data modeling on smaller data sets. If our goal as a field is to use data to solve problems, then we need to move away from exclusive dependence on data models and adopt a more diverse set of tools. Breiman [2001b], p199.

This is no longer true. The statistics community has by and large accepted the Machine Learning (ML) revolution that Breiman refers to as the algorithm modeling culture, and many textbooks discuss ML methods alongside more traditional statistical methods, *e.g.,* Hastie et al. [2009] and Efron and Hastie [2016]. Although the adoption of these methods in economics has been slower, they are now beginning to be widely used in empirical work, and are the topic of a rapidly increasing methodological literature. In this paper we want to make the case that economists and econometricians also, as Breiman writes referring to the statistics community, "need to move away from exclusive dependence on data models and adopt a more diverse set of tools." We discuss some of the specific tools that empirical researchers would benefit from, and which we feel should be part of the standard graduate curriculum in econometrics if, as Breiman writes, and we agree with, "our goal as a field is to use data to solve problems," if, in other words, we view econometrics as in essence, decision making under uncertainty (*e.g.,* Chamberlain [2000]), and if we wish to enable students to

be able to communicate effectively with researchers in other fields where these methods are routinely being adopted. Although relevant more generally, the methods developed in the ML literature have been particularly successful in "big data" settings, where we observe information on a large number of units, or many pieces of information on each unit, or both.

Why has the acceptance of ML methods been so much slower in economics compared to the broader statistics community? A large part of it may be the culture as Breiman refers to it. Economics journals emphasize the use of methods with formal properties of a type that many of the ML methods do not naturally deliver. This includes large sample properties of estimators and tests, including consistency, Normality, and efficiency. In contrast, the focus in the machine learning literature is often on working properties of algorithms in specific settings, with at best guarantees of error rates, but with fewer theoretical results of the type traditionally reported in econometrics papers. There are no formal results that show that for supervised learning problems deep learning or neural net methods are superior to regression trees or random forests, and it appears unlikely that general results for such comparisons will soon be available, if ever.

Although the ability to construct valid large sample confidence intervals is important in many cases, one should not out-of-hand dismiss methods that cannot not deliver them (or possibly, that can not yet deliver them), if these methods have other advantages. The demonstrated ability to outperform alternative methods on specific data sets in terms of out-of-sample predictive power is valuable in practice, although such performance is rarely explicitly acknowledged as a goal, or assessed, in econometrics. As Mullainathan and Spiess [2017] highlights, some problems are appropriately cast as prediction problems, and assessing their goodness of fit on a test set may be sufficient for the purposes of the analysis. In other cases, the output of a prediction problem is an input to the primary analysis of interest, and statistical analysis of the prediction component is not needed. On the other hand, there are also many settings where it is important to provide valid confidence intervals for a parameter of interest, such as an average treatment effect. The confidence interval may be used in decisions about whether to implement the treatment. We argue that in the future, as ML tools are more widely adopted, researchers should articulate clearly the goals of their analysis and why certain properties of algorithms and estimators may or may not be important.

A major theme of this review is that although there are cases where using simple of-the-shelf algorithms from the ML literature can be effective (see Mullainathan and Spiess [2017]

for a number of examples), there are also many cases where this is not the case. Often the ML techniques require careful tuning and adaptation to effectively address the specific problems economists are interested in. Perhaps the most important type of adaptation is to exploit the structure of the problems, *e.g.,* the causal nature of many estimands, the endogeneity of variables, the configuration of data such as panel data, the nature of discrete choice among a set of substitutable products, or the presence of credible restrictions motivated by economic theory, such as monotonicity of demand in prices. Statistics and econometrics have traditionally put much emphasis on these structures, and developed insights to exploit them. Exploiting these insights, both substantive and statistical, which, in a different form, is also seen in the careful tuning of ML techniques for specific problems such as image recognition, can greatly improve their performance. Another type of adaptation involves changing the optimization criteria of machine learning algorithms to prioritize considerations from causal inference, such as controlling for confounders or discovering treatment effect heterogeneity. Finally, techniques such as sample splitting (using different data to select models than to estimate parameters) and orthogonalization (e.g. Chernozhukov et al. [2016a]) can be used to improve the performance of machine learning estimators, in some cases leading to desirable properties such as asymptotic normality of machine learning estimators (e.g. Athey et al. [2017d]).

In this paper, we provide a list of tools that we feel should be be part of the empirical economists' toolkit and covered in the core econometrics graduate courses. Of course, this is a subjective list, and given the speed with which this literature is developing, the list will rapidly evolve. First on our list is nonparametric regression, or in the terminology of the ML literature, supervised learning for regression problems. This problem is familiar to economists, and there is a large set of methods preceeding the current ML literature that can serve to put the latter into context. Second, supervised learning for classification problems, or closely related, but not quite the same, nonparametric regression for discrete response models. This is the area where ML methods have perhaps had their biggest successes. Third, unsupervised learning, or clustering analysis. Fourth, we analyze estimates of heterogeneous treatment effects and optimal policies mapping from individuals' observed characteristics to treatments. Fifth, we discuss ML approaches to experimental design, where bandit approaches are starting to revolutionize effective experimentation especially in online settings. Sixth, we discuss the matrix completion problem, including its application to causal panel data models. Seventh, and closely related, we discuss the adaptation of matrix com-

pletion methods to problems of consumer choice among a discrete set of products. Finally, we discuss the analysis of text data.

We note that there are a few other recent reviews of ML methods aimed at economists, often with more empirical examples and references to applications than we discuss here. Varian [2014] is an early discussion of methods that economists should know about. Mullainathan and Spiess [2017] focus on the benefits of supervised learning methods for regression, and discuss the prevalence of problems in economics where prediction methods are appropriate. Athey [2017] and Athey et al. [2017c] provides a broader perspective with more emphasis on recent developments in adapting ML methods for causal questions and general implications for economics. In the computer science and statistics literatures there are also a number of excellent textbooks, with different levels of accessibility to researchers with a social science background, including Efron and Hastie [2016], which is a highly readable and highly recommended introduction, Hastie et al. [2009], which is a more comprehensive text from a statistics perspective, and Alpaydin [2009], and Knox [2018], which take more of a computer science perspective.

2 Econometrics and Machine Learning: Goals, Methods, and Settings

In this section we introduce some of the general themes of this paper. What are the differences in the goals and concerns of traditional econometrics and the machine learning literature, and how do these goals and concerns affect the choices between specific methods?

2.1 Goals

The traditional approach in econometrics is to specify a target, an estimand, that is a functional of a joint distribution of the data. Often the target is a parameter of a statistical model that describes the distribution of a set of variables (typically conditional on some other variables) in terms of a set of parameters, which can be a finite or infinite set. Given a random sample from the population of interest the parameter of interest and the nuisance parameters are estimated by finding the parameter values that best fit the full sample, using an objective function such as the sum of squared errors, or the likelihood function. The focus is on the quality of the estimators of the target, traditionally measured through large sample efficiency. Often there is also interest in constructing confidence intervals. Researchers

typically report point estimates and standard errors.

In contrast, in the ML literature the focus is typically on making good decisions. Often this takes the form of constructing algorithms for predicting some variables given others, or classifying units on the basis of limited information. A canonical problem is to classify handwritten digits on the basis of pixel values. The goal is to develop algorithms that will delivery high-quality decisions in a wide variety of new cases. (A widely cited paper, Wu et al. [2008], has the title "Top 10 algorithms in data mining").

In a very simple example, suppose we model the conditional distribution of some outcome Y_i given a vector-valued regressor or feature X_i. Suppose we are confident that

$$Y_i|X_i \sim \mathcal{N}(\alpha + \beta^\top X_i, \sigma^2).$$

We could estimate $\theta = (\alpha, \beta)$ by least squares, that is, as

$$(\hat{\alpha}_{\text{ls}}, \hat{\beta}_{\text{ls}}) = \arg\min_{\alpha,\beta} \sum_{i=1}^{N} \left(Y_i - \alpha - \beta^\top X_i\right)^2.$$

Most introductory econometrics texts would focus on the least squares estimator without much discussion. If the model is correct, the least squares estimator has well known attractive properties: it is unbiased, it is the best linear unbiased estimator, it is the maximum likelihood estimator, and so has large sample efficiency properties.

In ML settings the goal may be to make a prediction for the outcome for a new units on the basis of their regressor values. Suppose we are interested in predicting the value of Y_{N+1} for a new unit $N+1$, on the basis of the regressor values for this new unit, X_{N+1}. Suppose we restrict ourselves to linear predictors, so that the prediction is

$$\hat{Y}_{N+1} = \hat{\alpha} + \hat{\beta}^\top X_{N+1},$$

for some estimator $(\hat{\alpha}, \hat{\beta})$. The loss associated with this decision may be the squared error

$$\left(Y_{N+1} - \hat{Y}_{N+1}\right)^2.$$

The question now is to come up with estimators $(\hat{\alpha}, \hat{\beta})$ that have good properties associated with this loss function. This need not be the least squares estimator, and in fact when the dimension of the features exceeds two, we know from decision theory that we can do better, in terms of expected squared error, than the least squares estimator because that is not admissible, that is, there are other estimators that dominate the least squares estimator.

2.2 Terminology

One source of confusion is the use of new terminology in the ML for concepts that have well-established labels in the older literatures. In the context of a regression model the sample used to estimate the parameters is often referred to as the *training* sample. Instead of estimating the model, it is being *trained*. Regressors, covariates, or predictors are referred to as *features*. Regression parameters are often referred to as *weights*. Prediction problems are divided into *supervised learning problems* where we observe both the predictors/features X_i and the outcome Y_i, and *unsupervised learning problems* where we only observe the X_i and try to group them into clusters. Unordered discrete response problems are generally referred to as *classification problems*.

2.3 Validation and Cross-validation

In most discussions on linear regression in econometric textbooks there is little emphasis on model validation. The form of the regression model, be it parametric or nonparametric, and the set of regressors, is assumed to be given from the outside, *e.g.,* economic theory. Given this specification the task of the researcher is to estimate the unknown parameters of this model. Much emphasis is on doing this estimation step efficiently, typically operationalized through definitions of large sample efficiency. If there is discussion of model selection, it is often in the form of testing null hypotheses concerning the validity of a particular model, with the implication that there is a true model that should be selected and used for subsequent tasks.

Consider the regression example in the previous subsection. Let us assume that we are interested in predicting the outcome for a new unit, randomly drawn from the same population as our sample was drawn from. As an alternative to estimating the linear model with an intercept, and a scalar X_i, we could estimate the model with only an intercept. Certainly if $\beta = 0$, that model would lead to better predictions. By the same argument, if the true value of β were close, but not exactly equal, to, zero, we would still do better leaving X_i out of the regression. A simple calculation shows that if $|\beta| > \sigma/\sqrt{\sum_i (X_i - \overline{X})^2}$, we should, and otherwise we should not, include X_i in the regression. But we do not know the value of β or σ. How do we then decide between the two estimators? This is where the cross-validation comes in. Let us split the sample into two parts, a *training* sample and a *test* sample. Using the training sample we calculate the two least squares estimators, $(\hat{\alpha}_L, \hat{\beta}_L)$

for the long regression with X_i included, and $(\hat{\alpha}_S, \hat{\beta}_S = 0)$ for the short regression with X_i not included in the regression function. Now we can assess and compare the performance of the two estimators by calculating for each unit in the test sample the predicted value $\hat{Y}_j$ and averaging the squared deviation from the actual value Y_j. To be specific, suppose that the first N_{train} units are in the training sample, and the last $N_{\text{test}} = N - N_{\text{train}}$ units are in the test sample. Then we calculate the average squared error over the test sample,

$$Q_S = \frac{1}{N_{\text{test}}} \sum_{j=N_{\text{train}}}^{N_{\text{train}}+N_{\text{test}}} \left(Y_j - \hat{\alpha}_S\right)^2,$$

and

$$Q_L = \frac{1}{N_{\text{test}}} \sum_{j=N_{\text{train}}}^{N_{\text{train}}+N_{\text{test}}} \left(Y_j - \hat{\alpha}_L - \hat{\beta}_L X_j\right)^2.$$

Based on this the relative magnitude of Q_L and Q_S we can decide whether to use the long regression or the short regression model.

What is key in this approach is that the test sample in combination with the predictive nature of the question allows us to evaluate effectively the quality of the decisions for the two candidate models. If the test sample is large we will obtain an accurate estimate of the predictive power of the candidate models. There are two components of the problem that are important for this ability. First, the goal is predictive power, rather than estimation of a particular structural or causal parameter. Second, the method uses out-of-sample comparisons, rather than in-sample goodness-of-fit measures. This ensures that we obtain unbiased comparisons of the fit.

2.4 Over-fitting, Regularization, and Tuning Parameters

The ML literature is much more concerned with over-fitting than the standard statistics or econometrics literature. Researchers attempt to select flexible models that fit well, but not so well that out-of-sample prediction is compromised. There is much less emphasis on formal results that particular methods are superior in large samples (asymptotically), instead methods are compared on specific data sets to see "what works well." To avoid overfitting a key concept is that of *regularization*. As Vapnik writes,

> "Regularization theory was one of the first signs of the existence of intelligent inference" (Vapnik [1998], p.)

Consider a setting with a large set of models that differ in their complexity, measured for example as the number of unknown parameters in the model. Instead of directly optimizing an objective function, say minimizing the sum of squared residuals in a least squares regression setting, or maximizing the logarithm of the likelihood function, a term is added to the objective function to penalize the complexity of the model. There are antecedents of this practice in the traditional literature. One is that in likelihood settings researchers sometimes add a term to the logarithm of the likelihood function equal to minus the logarithm of the sample size times the number of free parameters divided by two, leading to the *Bayesian Information Criterion*, or simply the number of free parameters, the *Akaike Information Criterion*. In Bayesian analyses of regression models the use of a prior distribution on the regression parameters, centered at zero, independent accross parameters with a constant prior variance, is a second way of regularizing estimation that has a long tradition. The difference with the modern approaches to regularization is that they are more data driven, with the amount of regularizing determined explicitly directly by the out-of-sample predictive performance.

Consider a linear regression model with K regressors,

$$Y_i|X_i \sim \mathcal{N}\left(\beta^\top X_i, \sigma^2\right).$$

Suppose we also have a prior distribution for the the slope coefficients β_k, with the prior for β_k, $\mathcal{N}(0,\tau^2)$, and independent of $\beta_{k'}$ for any $k \neq k'$. (This may be more plausible if we first normalize the features and outcome to have mean zero and unit variance. We assume this has been done.) Given the value for the variance of the prior distribution, τ^2, the posterior mean for β is the solution to

$$\arg\min_\beta \sum_{i=1}^N \left(Y_i - \beta^\top X_i\right)^2 + \frac{\sigma^2}{\tau^2}\|\beta\|^2,$$

where $\|\beta\| = \sum_{k=1}^K \beta_k^2$. One version of an ML approach to this problem is to estimate β by minimizing

$$\arg\min_\beta \sum_{i=1}^N \left(Y_i - \beta^\top X_i\right)^2 + \lambda\|\beta\|^2.$$

The only difference is in the way the penalty parameter λ is chosen. In a formal Bayesian approach this reflects the prior distribution on the parameters, and it would be chosen *a priori*. In an ML approach λ would be chosen through out-of-sample cross-validation to optimize the predictive performance.

2.5 Sparsity

In many settings in the ML literature the number of features is substantial, both in absolute terms and relative to the number of units in the sample. However, there is often a sense that many of the features are of minor importance, if not completely irrelevant. The problem is that we may not know *ex ante* which of the features matter, and which can be dropped from the analysis without substantially hurting the predictive power.

Hastie et al. [2009, 2015] discuss what they call the *sparsity principle*:

> "Assume that the underlying true signal is sparse and we use an ℓ_1 penalty to try to recover it. If our assumption is correct, we can do a good job in recovering the true signal. ... But if we are wrong—the underlying truth is not sparse in the chosen bases—then the ℓ_1 penalty will not work well. However, in that instance, no method can do well, relative to the Bayes error." (Hastie et al. [2015], page 24).

Exact sparsity is in fact stronger than is necessary, in many cases it is sufficient to have approximate sparsity where most of the explanatory variables have very limited explanatory power, even if not zero, and only a few of the features are of substantial importance (see, for example, Belloni et al. [2014]).

Traditionally in the empirical literature in social sciences researchers limited the number of explanatory variables by hand, rather than choosing them in a data-dependent manner. Allowing the data to play a bigger role in the variable selection process appears a clear improvement, even if the assumption that the underlying process is at least approximately sparse is still a very strong one, and even if inference in the presence of data-dependent model selection can be challenging.

2.6 Computational Issues and Scalability

Compared to the traditional statistics and econometrics literatures the ML literature is much more concerned with computational issues and the ability to implement estimation methods with large data sets. Solutions that may have attractive theoretical properties in terms of statistical efficiency but that do not scale well to large data sets are often discarded in favor of methods that can be implemented easily in very large data sets. This can be seen in the discussion of the relative merits of LASSO versus subset selection in linear regression settings. In a setting with a large number of features that might be included in the analysis,

subset selection methods focus on selecting a subset of the regressors and then estimate the parameters of the regression function by least squares. One can implement subset selection by adding a penalty term to the sum of squared residuals that is proportional to the number of non-zero regression parameters. This may seem a more direct way to choose a subset of features than LASSO, where the sparsity of the estimates is a result of the form of the penalty term. However, LASSO has computational advantages. It can be implemented by adding a penalty term that is proportional to the sum of the absolute values of the parameters. A major attraction of LASSO is that there are effective methods for calculating the LASSO estimates with the number of regressors in the millions. Best subset selection regression, on the other hand, is an NP-hard problem. Until recently it was thought that this was only feasible in settings with the number of regressors in the 30s, although current research (Bertsimas et al. [2016]) suggests it may be feasible with the number of regressors in the 1000s. This has reopened a new, still unresolved, debate on the relative merits of LASSO versus best subset selection (see Hastie et al. [2017]) in settings where both are feasible. There are some indications that in settings with a low signal to noise ratio, as is common in many social science applications, LASSO may have better performance, although there remain many open questions. In many social science applications the scale of the problems is such that best subset selection is also feasible, and the computational issues may be less important than these substantive aspects of the problems.

A key computational optimization tool used in many ML methods is Stochastic Gradient Descent (SGD, Friedman [2002], Bottou [1998, 2012]). It is used in a wide variety of settings, including in optimizing neural networks and estimating models with many latent variables (e.g., Ruiz et al. [2017]). The idea is very simple. Suppose that the goal is to estimate a parameter θ, and the estimation approach entails finding the value $\hat{\theta}$ that minimizes an empirical loss function, where $Q_i(\theta)$ is the loss for observation i, and the overall loss is $\sum_i Q_i(\hat{\theta})$. Classic gradient descent involves an iterative approach, where $\hat{\theta}_k$ is formed from $\hat{\theta}_{k-1}$ as follows:

$$\theta_k = \theta_{k-1} - \eta \frac{1}{N} \sum_i \nabla Q_i(\hat{\theta}),$$

where η is the learning rate.

The challenge with this approach is that evaluating ∇Q_i may be expensive. In stochastic gradient descent, instead of using the whole dataset to evaluate the gradient of the overall loss function, we use the fact that for a single randomly selected observation i, ∇Q_i is an

unbiased (if very noisy) estimate of the overall gradient. The stochastic gradient descent algorithm shuffles the dataset, and then with the shuffled dataset, for $k = 1, ..., N$, let

$$\theta_k = \theta_{k-1} - \eta \nabla Q_k(\hat{\theta}).$$

If convergence is not achieved when $k = N$, reshuffle the dataset and then repeat. The convergence of SGD has been studied, with the conlusion that if the learning rate η decreases at an appropriate rate, under relatively mild assumptions, stochastic gradient descent converges almost surely to a global minimum when the objective function is convex or pseudo-convex, and otherwise converges almost surely to a local minimum. See Bottou [2012] for an overview and practical tips for implementation. The idea of using a noisy, unbiased estimate of the gradient exploits the idea that it is better to take many, many small steps that are noisy but on average in the right direction, than it is to spend equivalent computational cost in very accurately figuring out in what direction to take a single small step.

The idea can be pushed even further in the case where Q_i is itself an expectation. We can consider evaluating ∇Q_i using Monte Carlo integration. But, rather than taking many Monte Carlo draws to get an accurate approximation to the integral, we can instead take a small number of draws, or even a single draw.

2.7 Ensemble Methods and Model Averaging

Another key feature of the machine learning literature is the use of model averaging and ensemble methods (*e.g.,* Dietterich [2000]). In many cases a single model or algorithm does not perform as well as a combination of possibly quite different models, averaged using weights (sometimes called *votes*) obtained by optimizing out-of-sample performance. A striking example is the Netflix Prize Competition (Bennett et al. [2007]), where all the top contenders use combinations of models, often averages of many models (Bell and Koren [2007]). There are two related ideas in the traditional econometrics literature. Obviously Bayesian analysis implicitly averages over the posterior distribution of the parameters. Mixture models are also used to combine different parameter values in a single prediction. However, in both cases this model averaging involves averaging over similar models, typically with the same specification, and only different in terms of parameter values. In the modern literature, and in the top entries in the Netflix competition, the models that are averaged over can be quite different, and the weights are obtained by optimizing out-of-sample predictive power, rather than in-sample fit.

For example, one may have three predictive models, one based on a random forest, leading to predictions $\hat{Y}_i^{\rm RF}$, one based on a neural net, with predictions $\hat{Y}_i^{\rm NN}$, and one based on a linear model estimated by LASSO, leading to $\hat{Y}_i^{\rm LASSO}$. Then, using a test sample, one can choose weights $p^{\rm RF}$, $p^{\rm NN}$, and $p^{\rm LASSO}$, by minimizing the sum of squared residuals in the test sample:

$$(\hat{p}^{\rm RF},\hat{p}^{\rm NN},\hat{p}^{\rm LASSO}) = \arg\min_{p^{\rm RF},p^{\rm NN},p^{\rm LASSO}} \sum_{i=1}^{N^{\rm test}} \left(Y_i - p^{\rm RF}\hat{Y}_i^{\rm RF} - p^{\rm NN}\hat{Y}_i^{\rm NN} - p^{\rm LASSO}\hat{Y}_i^{\rm LASSO}\right)^2,$$

$$\text{subject to } p^{\rm RF}+p^{\rm NN}+p^{\rm LASSO}=1, \quad \text{and } p^{\rm RF},p^{\rm NN},p^{\rm LASSO}\geq 0.$$

One may also estimate weights based on regression of the outcomes in the test sample on the predictors from the different models without imposing that the weights sum to one and are non-negative. Because random forests, neural nets, and lasso have distinct strengths and weaknesses, in terms of how well they deal with the presence of irrelevant features, nonlinearities, and interactions. As a result averaging over these models may lead to out-of-sample predictions that are strictly better than predictions based on a single model.

2.8 Inference

The ML literature has focused heavily on out-of-sample performance as the criterion of interest. This has come at the expense of one of the concerns that the statistics and econometrics literature have traditionally focused on, namely the ability to do inference, *e.g.,* construct confidence intervals that are valid, at least in large samples. Efron and Hastie write:

> "Prediction, perhaps because of its model-free nature, is an area where algorithmic developments have run far ahead of their inferential justification." (Efron and Hastie [2016], p. 209)

Although there has been substantial progress in the development of methods for inference in specific settings (*e.g.*, Wager and Athey [2017]), it remains the case that for many methods it is currently impossible to construct confidence intervals that are valid, even if only asymptotically. A question is whether this ability to construct confidence intervals is as important as the traditional emphasis on it in the econometric literature suggests. For many decision problems it may be that prediction is all that matters and inference is not important. Even in cases where it is possible to do inference, it is important to keep in mind that the requirements that ensure this ability often come at the expense of predictive performance.

One can see this tradeoff in traditional kernel regression, where the bandwidth that optimizes expected squared error balances the tradeoff between the square of the bias and the variance, so that the optimal estimators have an asymptotic bias that invalidates the use of standard confidence intervals. This can be fixed by using a bandwidth that is smaller than the optimal one, so that the asymptotic bias vanishes, but it does so at the expense of increasing the variance. An additional concern is that the traditional basis for inference, assuming that the sample at hand is a random sample from a well-defined population, is not always plausible. In many cases the sample is either a convenience sample, or essentially an entire subpopulation, *e.g.*, all units during a particular period of time, so that a random sampling perspective is not attractive. See, for example, Abadie et al. [2017]. The question is whether this lack of inference should bother us.

Consider a prediction problem where we are interested in the conditional expectation of an outcome Y_i given a set of predictors X_i. Options for estimating this conditional expectation may include a regression tree and a kernel regression estimator. There is much theory for kernel regression, with well-established conditions for consistency and asymptotic normality of the kernel estimator at a particular point x. Let us put aside the questions regarding the applicability of these conditions, such as existence of higher-order derivatives of the conditional expectation. Even in settings where these conditions are satisfied, regression trees may lead to better predictions on average, even if the properties of their sampling distributions are not well understood, and it is unlikely that these estimators will be asymptotically normally distributed around the true value $\mathbb{E}[Y_i|X_i = x]$. In many cases the superior predictive properties may well outweigh the lack of inference.

In other settings inference is clearly of great importance. Again this is often an issue related to whether the focus is on predictive performance versus causal parameters, and the associated question of whether out-of-sample validation is possible. Suppose we are comparing two estimators, A and B, with B based on a more complex model that encompasses the one underlying estimator A. If the focus is on predictive performance, and out-of-sample comparisons show clearly that estimator A has a superior predictive performance compared to estimator B, then the ability to construct confidence intervals many not matter. On the other hand, if we cannot do out-of-sample comparisons, the ability to construct confidence intervals may allow us to compare A and B given their nesting structure.

Of course, as discussed in the introduction, when the goal of the empirical analysis is to estimate a parameter such as an average treatment effect, valid confidence intervals may

be very important to decision-makers who want to understand whether, for example, the finding that a treatment effect is positive is likely to be due to sampling variation.

3 Supervised Learning for Regression Problems

One of the canonical problems in both the ML and econometric literatures is that of estimating the conditional mean of a scalar outcome given a set of of covariates or features. Let Y_i denote the outcome for unit i, and let X_i denote the K-component vector of covariates or features. The conditional expectation is

$$g(x) = \mathbb{E}[Y_i|X_i = x].$$

Compared to the traditional econometric textbook there are some key differences with the ML literature. See also Mullainathan and Spiess [2017] for a discussion. In the settings considered in the ML there may be many covariates, sometimes more than there are observations in the sample. There is no presumption in the ML literature that the conditional distribution of the outcomes given the covariates is Gaussian. The derivatives of the conditional expectation for each of the covariates, which in the linear regression model corresponds to the parameters, are not of intrinsic interest. Instead the focus is on out-of-sample predictions and their accuracy. Furthermore, there is less of a sense that the conditional expectation is monotone in each of the covariates compared to many economic applications. Often there is concern that the conditional expectation may be an extremely non-monotone function with interactions of substantial importance.

The econometric literature on estimating the conditional expectation is also huge. Parametric methods for estimating $g(\cdot)$ often used least squares. Since the work by Bierens [1987], kernel regression methods have become a popular alternative in case more flexibility is required, with subsequently series or sieve methods gaining interest (see Chen [2007] for a survey). These methods have well established large sample properties, allowing for the construction of confidence intervals. Simple non-negative kernel methods are viewed as performing very poorly in settings with high-dimensional covariates, with the difference $\hat{g}(x) - g(x)$ of order $O_p(N^{-1/K})$. This rate can be improved by using higher order kernels and assuming the existence of many derivatives of $g(\cdot)$, but practical experience with high-dimensional covariates has not been satisfactory for these methods, and applications of kernel methods are generally limited to low-dimensional settings.

The differences in performance between some of the traditional methods such as kernel regression and the modern methods such as random forests are particularly pronounced in sparse settings with a large number of more or less irrelevant covariates. Random forests are effective at picking up on the sparsity and ignoring the irrelevant features, even if there are many of them, while the traditional implementations of kernel methods essentially waste degrees of freedom on accounting for these covariates. Although may be possible to adapt kernel methods for the presence of irrelevant covariates, in practice there has been little effort in this direction. A second issue is that the modern methods are particularly good at detecting severe nonlinearities and high-order interactions. The presence of such high-order interactions in some of the success stories of these methods should not blind us to the fact that with many economic data we expect high-order interactions to be of limited importance. If we try to predicting earnings for individuals, we expect the regression function to be monotone in many predictors such as education and prior earnings variables, even for homogenous subgroups. This means that models based on linearizations may do well in such cases relative to other methods, compared to settings where monotonicity is fundamentally less plausible, as, for example, in an image recognition problem. This is also a reason for the superior performance of linear random forests (Friedberg et al. [2018]) relative to standard random forests.

We discuss four specific sets of methods, although there are many more including variations on the basic methods. First methods for the linear case where the class of models considered is linear in the covariates, and the question is about regularization. Next we discuss methods based on partitioning the covariate space using regression trees and random forests. In the third subsection we discuss neural nets, which were the focus on of a small econometrics literature in the 1990s (White [1992], Hornik et al. [1989]), but more recently has become a very prominent literature in ML. Then we discuss boosting as a general principle.

3.1 Regularized Linear Regression: Lasso, Ridge, and Elastic Nets

Suppose we consider approximations to the conditional expectation that have a linear form

$$g(x) = \beta^\top x = \sum_{k=1}^{K} \beta_k x_k,$$

after the covariates and the outcome are demeaned, and the covariates are normalized to have unit variance. The traditional method for estimating the regression function in this

case is least squares, with

$$\hat{\beta}^{\text{ls}} = \arg\min_{\beta} \sum_{i=1}^{N} \left(Y_i - \beta^\top X_i\right)^2 .$$

However, if the number of covariates K is large relative to the number of observations N the least squares estimator $\hat{\beta}_k^{\text{ls}}$ does not even have particularly good repeated sampling properties as an estimator for β_k, let alone good predictive properties. In fact, with $K \geq 3$ the least squares estimator is not even admissible and is dominated by estimators that shrink towards zero. With K very large, possibly even exceeding the sample size N, the least squares estimator has particularly poor properties, even if the conditional distribution of the outcome given the covariates is in fact linear.

Even with K modest in magnitude, the predictive properties of the least squares estimator may be inferior to those of estimators that use some kind of regularization. One common form of regularization is to add a penalty term that shrinks the β_k towards zero, and minimize

$$\arg\min_{\beta} \sum_{i=1}^{N} \left(Y_i - \beta^\top X_i\right)^2 + \lambda \left(\|\beta\|^q\right)^{1/q} .$$

For $q = 1$ this corresponds to LASSO (Tibshirani [1996]). For $q = 2$ this corresponds to ridge regression (Hoerl and Kennard [1970]). As $\lambda \to 0$, the solution penalizes the number of non-zero covariates, leading to best subset regression (Miller [2002], Bertsimas et al. [2016]). In addition there are many hybrid methods and modifications, including elastic nets which combines penalty terms from LASSO and ridge (Zou and Hastie [2005]), the relaxed lasso, which combines least squares estimates from the subset selected by LASSO and the LASSO estimates themselves (Meinshausen [2007]), Least Angle Regression (Efron et al. [2004]), the Dantzig Selector (Candès and Tao [2007]), and many others.

There are a couple of important conceptual differences between these three special cases, subset selection, LASSO, and ridge regression. See for a recent discussion Hastie et al. [2017]. Both best subset and LASSO lead to solutions with a number of the regression coefficients exactly equal to zero, a *sparse* solution. For the ridge estimator on the other hand all the estimated regression coefficients will generally differ from zero. It is not always important to have a sparse solution, and often the variable selection that is implicit in these solutions is over-interpreted. Second, best subset regression is computationally hard (NP-hard), and as a result not feasible in settings with N and p large, although recently progress has been made in this regard (Bertsimas et al. [2016]). LASSO and ridge have a Bayesian interpretation.

Ridge regression gives the posterior mean and mode under a normal model for the conditional distribution of Y_i given X_i, and normal prior distributions for the parameters. LASSO gives the posterior mode given Laplace prior distributions. However, in contrast to formal Bayesian approaches, the coefficient λ on the penalty term is in the modern literature choosen through out-of-sample crossvalidation rather than through the choice of prior distribution.

3.2 Regression Trees and Forests

Regression trees (Breiman et al. [1984]), and their extension random forests (Breiman [2001a]) have become very popular and effective methods for flexible estimating regression functions in settings where out-of-sample predictive power is important, partly because of their ease of use. Given a sample $(X_{i1}, \ldots, X_{iK}, Y_i)$, for $i = 1, \ldots, N$, the idea is to split the sample into subsamples, and estimate the regression function within the subsamples simply as the average outcome. The splits are sequential and based on a single covariate X_{ik} at a time exceeding a threshold c. Starting with the full training sample, consider a split based on feature or covariate k, and threshold c. The sum of in-sample squared errors before the split was

$$Q = \sum_{i=1}^{N} \left(Y_i - \overline{Y}\right)^2, \qquad \text{where } \overline{Y} = \frac{1}{N}\sum_{i=1}^{N} Y_i.$$

After the split based on covariate k and threshold c it is

$$Q(k,c) = \sum_{i:X_{ik}\leq c} \left(Y_i - \overline{Y}_{k,c,l}\right)^2 + \sum_{i:X_{ik}>c} \left(Y_i - \overline{Y}_{k,c,r}\right)^2,$$

where

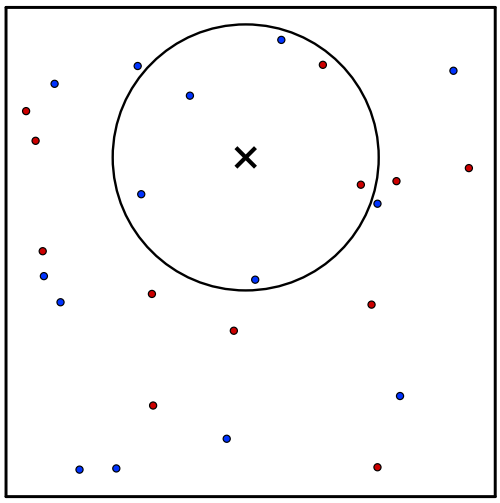
$$\overline{Y}_{k,c,l} = \sum_{i:X_{ik}\leq c} Y_i \Big/ \sum_{i:X_{ik}\leq c} 1, \qquad \text{and } \overline{Y}_{k,c,r} = \sum_{i:X_{ik}>c} Y_i \Big/ \sum_{i:X_{ik}>c} 1,$$

are the average outcomes in the two subsamples. We split the sample using the covariate k and threshold c that minimize the average squared error $Q(k,c)$ over all covariates and thresholds. We then repeat this, now optimizing also over the subsamples or leaves. At each split the average squared error is further reduced (or stays the same). We therefore need some regularization to avoid overfitting by splitting the sample too many times. One approach is to add a penalty term to the sum of squared residuals that is linear in the number of subsamples (the *leaves*). The coefficient on this penalty term is then chosen through cross-validation. In practice, a very deep tree is estimated, and then "pruned" to a more shallow tree using cross-validation to select the optimal tree depth.

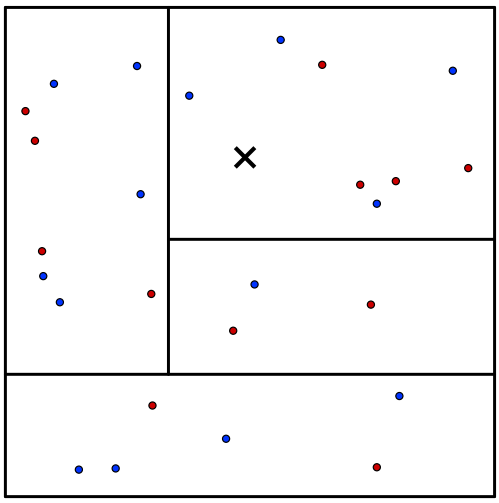
The need for regularization is particularly apparent in the case of regression trees, since a very deep tree with a single observation in each leaf explains the training data perfectly, but obviously will predict poorly in an independent test set.

An advantage of a single tree is that it is very easy to explain and interpret results. Once the tree structure is defined, then the prediction in each leaf is a sample average, and the standard error of that sample average is easy to compute. However, it is not in general true that the sample average of the mean within a leaf is an unbiased estimate of what the mean would be within that same leaf in a new test set. Since the leaves were selected using the data, the leaf sample means in the training data will tend to be more extreme (in the sense of being different from the overall sample mean) than in an independent test set. However, Athey and Imbens [2016] suggest sample splitting as a way to avoid this issue. If a confidence interval for the prediction is desired, then the analyst can simply split the data in half. One half of the data is used to construct a regression tree. Then, the partition implied by this tree is taken to the other half of the data (referrred to as the "estimation sample") where the sample mean within a given leaf should be an unbiased estimate of the true mean value for the leaf. It is important to note that while the estimate will be unbiased for the true mean for a leaf, that is not the same as being unbiased for any particular realization of the covariate.

Although trees are easy to interpret, it is important not to go too far in interpreting the structure of the tree. Standard intuitions from econometrics about "omitted variable bias" can be useful here. In a data set of limited size, the tree will split for some covariates and not others. The fact that a covariate is not selected for splitting does not mean it is not important for prediction. Suppose that a leaf is defined by $X_{1i} < a, X_{2i} > b$. Then, a leaf sample mean in an estimation set is an unbiased estimate of $\mathbb{E}_{X_{-12}}[Y_i | X_{1i} < a, X_{2i} > b]$. The key point is that the distribution of covariates other than X_1 and X_2 also vary across leaves. Thus, one useful way to describe a regression tree, beyond the tree structure, is by the average values of all covariates within each leaf. This makes is more clear that even though leaves are defined by a subset of covariates, leaves differ for all covariates.



Euclidean neighborhood,
for KNN matching.



Tree-based neighborhood.

One way to interpret a tree is that it is an alternative to a "K-nearest-neighbor matching algorithm" (KNN). Within each tree, the prediction for a leaf is simply the sample average outcome within the leaf. Thus, we can think of the leaf as defining the set of nearest neighbors for a given target observation in a leaf, and the estimator from a single regression tree is a matching estimator with non-standard ways of selecting the nearest neighbor to a target point. In particular, the neighborhoods will prioritize some covariates over others in determining which observations qualify as "nearby." The figure illustrates the difference between KNN and a tree-based matching algorithm for the case of two covariates. KNN will create a neighborhood around a target observation based on the Euclidean distance to each point, while tree-based neighborhoods will be rectangles. In addition, a target observation may not be in the center of a rectangle. Thus, a single tree is not the best way to predict outcomes for any given test point x. When a prediction tailored to a specific target observation is desired, generalizations of tree-based methods can be used.

For more targeted estimates of $\mu(x)$, random forests (Breiman [2001a]) build on the regression tree algorithm. The first issue random forests address is that the estimated regression function given a tree is very discontinuous, more than one might like. Random forests induce smoothness by averaging over a large number of trees. These trees differ from each other in two ways. First, each tree is based not on the original sample, but on a bootstrap sample or a subsample. This extension on its own is known as *bagging* (Breiman [1996]) or subsampling; although bagging was recommended by Breiman [1996], it turns out

that subsampling works about as well in most applications, and is more amenable to theoretical analysis (Wager and Athey [2017]). Second, the splits at each stage are not optimized over all possible covariates, but over a random subset of the covariates, changing every split. These two modifications lead to sufficient variation in the trees that the average is relatively smooth and has better predictive power than a single tree.

Random forests have become very popular methods. A key attraction is that they require relatively little tuning and have great performance out-of-the-box compared to more complex methods such as deep learning neural networks. Random forests and regression trees are particularly effective in settings with a large number of features that are not related to the outcome, that is, settings with sparsity. The splits will generally ignore those covariates, and as a result the performance will remain strong even in settings with a large number of features. Indeed, when comparing forests to KNN matching, a reliable way to make forests perform better is to add "noise" covariates that have no predictive power. These will rapidly degrade the performance of KNN, but will not affect random forest nearly as severely [Wager and Athey, 2017].

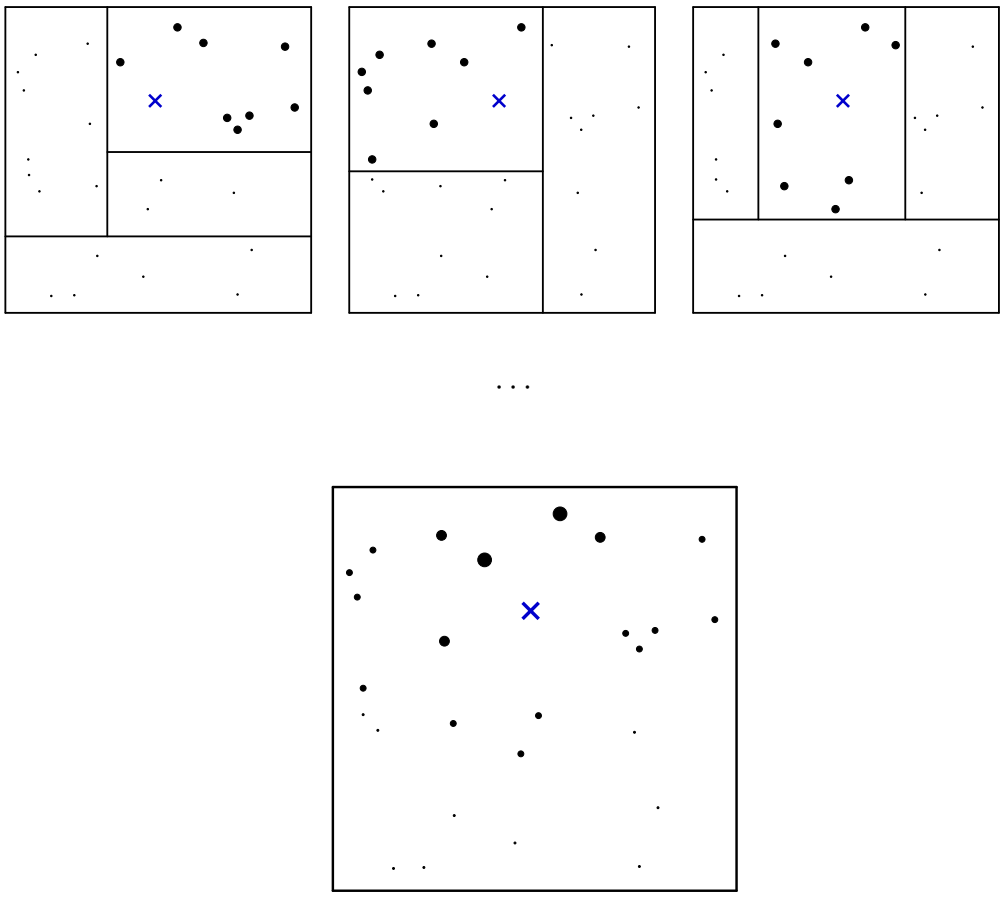
Although the statistical analysis of forests had proved elusive since (Breiman [2001a])'s original work, Wager and Athey [2017] show that a particular variant of random forests can produce estimates $\hat{\mu}(x)$ with an asymptotically normal distribution centered on the true value $\mu(x)$, and further, they provide an estimate of the variance of the estimator so that centered confidence intervals can be constructed. The variant they study uses subsampling rather than bagging; and further, each tree is built using two disjoint subsamples, one used to define the tree, and the second used to estimate sample means for each leaf. This "honest" estimation is crucial fot the asymptotic analysis.

Random forests can be connected to traditional econometric methods in several ways. Returning to the KNN comparison, since each tree is a form of matching estimator, the forest is an average of matching estimators. By averaging over trees, the prediction for each point will be centered on the test point (except near boundaries of the covariate space). However, the forest prioritizes more important covariates for selecting matches in a data-driven way. Another way to interpret random forests (e.g. Athey et al. [2017d]), is that they generate weighting functions analogous to kernel weighting functions. For example, a kernel regression makes a prediction at a point x by averaging nearby points, but weighting closer points more heavily. A random forest, by averaging over many trees, will include nearby points more often than distant points. We can formally derive a weighting function for a

given test point by counting the share of trees where a particular observation is in the same leaf as a test point. Then, random forest predictions can be written as

$$\hat{\mu}_{\text{rf}}(x) = \sum_{i=1}^{n} \alpha_i(x)\, Y_i, \quad \sum_{i=1}^{n} \alpha_i(x) = 1, \quad \alpha_i(x) \geq 0, \tag{3.1}$$

where the weights $\alpha_i(x)$ encode the weight given by the forest to the i-th training example when predicting at x. The difference between typical kernel weighting functions and forest-based weighting functions is that the forest weights are adaptive; if a covariate has little effect, it will not be used in splitting leaves, and thus the weighting function will not be very sensitive to distance along that covariate.



The Kernel Based on Share of Trees in Same Leaf as Test Point X

Recently random forests have been extended to settings where the interest is in causal effects, either average or unit-level causal effects (Wager and Athey [2017]), as well as for estimating paramters in general economic models that can be estimated with maximum likelihood or GMM (Athey et al. [2017d]). In the latter case, the interpretation of the forest

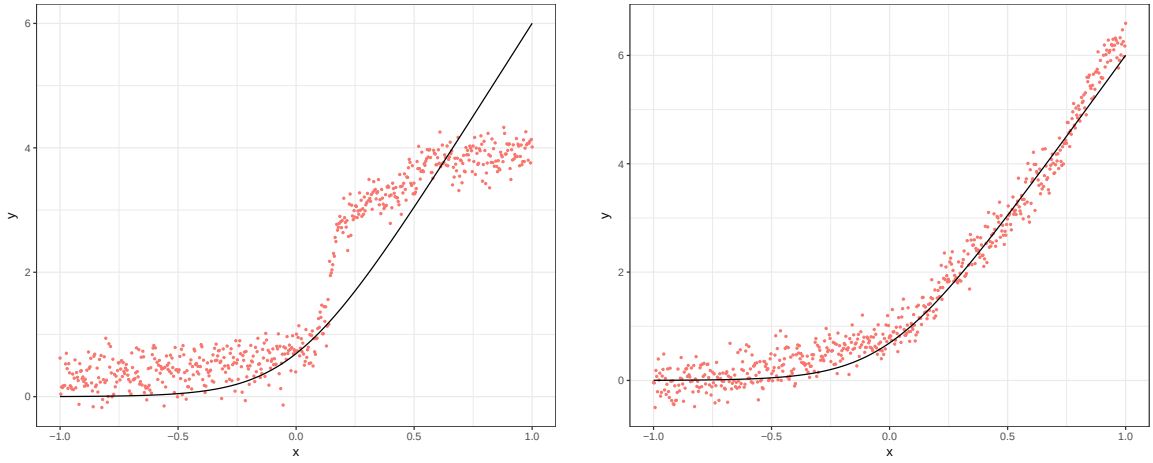
as creating a weighting function is operationalized; the new "generalized random forest" algorithm operates in two steps. First, a forest is constructed, and second, a GMM model is estimated for each test point, where points that are nearby in the sense of frequently occuring in the same leaf as the test point are weighted more heavily in estimation. With an appropriate version of honest estimation, these forests produce parameter estimates with an asymptotically normal distribution. Generalized random forests can be thought of as a generalization of local maximum likelihood, introduced by Tibshirani and Hastie [1987], but where kernel weighting functions are used to weight nearby observations more heavily than observations distant from a particular test point.

A weakness of forests is that they are not very efficient at capturing linear or quadratic effects, or at exploiting smoothness of the underlying data generating process. In addition, near the boundaries of the covariate space, they are likely to have bias, because the leaves of the component trees of the random forest cannot be centered on points near the boundary. Traditional econometrics encounters this boundary bias problem when estimating models using "regression discontinuities," such as geographical boundaries of school districts or test score cutoffs for eligibility for schools or programs (Abadie and Imbens [2011]). The solution proposed in the econometrics literature is to use local linear regression, which is a regression with nearby points weighted more heavily. Suppose that the conditional mean function is increasing as it approaches the boundary. Then the local linear regression corrects for the fact that at a test point near the boundary, most sample points lie in a region with lower conditional mean than the conditional mean at the boundary. Friedberg et al. [2018] extends the generalized random forest framework to local linear forests, which are constructed by running a regression weighted by the weighting function derived from a forest. In their simplest form, local linear forests just take the forest weights $\alpha_i(x)$, and use them for local regression:

$$(\hat{\mu}(x), \hat{\theta}(x)) = \text{argmin}_{\mu,\theta} \left\{ \sum_{i=1}^{n} \alpha_i(x)(Y_i - \mu(x) - (X_i - x)\theta(x))^2 + \lambda||\theta(x)||_2^2 \right\}. \quad (3.2)$$

Performance can be improved by modifying the tree construction to incorporate a regression correction; in essence, splits are optimized for predicting residuals from a local regression. This algorithm performs better than traditional forests in settings where a regression can capture broad patterns in the conditional mean function such as monotonicity or a quadratic structure, and again, asymptotic normality is established. Figure 1, from Friedberg et al. [2018], illustrates how local linear forests can improve on regular random

forests; by fitting local linear regressions with a random-forest estimated kernel, the resulting predictions can match a simple polynomial function even in relatively small data sets. In contrast, a forest tends to have bias, particularly near boundaries, and in small data sets will have more of a step function shape. Although the figure shows the impact in a single dimension, an advantage of the forest over a kernel is that these corrections can occur in multiple dimensions, while still allowing the traditional advantages of a forest in uncovering more complex interactions among covariates.



Random forest Local linear forest

Figure 1: Predictions from random forests and local linear forests on 600 test points. Training and test data were simulated from $Y_i = \log\left(1 + e^{6X_{i1}}\right) + \epsilon_i, \epsilon_i \sim \mathcal{N}(0,\, 20)$ with X having dimension $d = 20$ (19 covariates are irrelevant) and errors $\epsilon \sim N(0, 20)$. Forests were trained on n = 600 training points using the R package GRF, and tuned via cross-validation. Here the true conditional mean signal $\mu(x)$ is in black, and predictions are shown in red.

3.3 Deep Learning and Neural Nets

Neural networks and related deep learning methods are another general and flexible approach to estimating regression functions. They have been found to be very succesful in complex settings, with extremely large number of features. However, in practice these methods require a substantial amount of tuning in order to work well for a given application, relative to methods such as random forests. Neural networks were studied in the econometric literature in the 1990s, but did not catch on at the time (see White [1992], Hornik et al. [1989], White [1992]).

Let us consider a simple example. Given K covariates or features X_{ik}, we model K_1

latent/unobserved variables Z_{ik} (*hidden nodes*) that are linear in the original covariates:

$$Z_{ik}^{(1)} = \sum_{j=1}^{K} \beta_{kj}^{(1)} X_{ij}, \quad \text{for } k = 1, \ldots, K_1.$$

We then model the outcome as a linear function of some simple known nonlinear transformation of these hidden nodes plus noise:

$$Y_i = \sum_{k=1}^{K_1} \beta_k^{(2)} g\left(Z_{ik}^{(1)}\right) + \varepsilon_i,$$

where the transformation is a simple function, *e.g.,* a sigmoid $g(a) = (1 + \exp(-a))^{-1}$, or a rectified linear $g(a) = a\mathbf{1}_{a>0}$. This is a neural network with a single hidden layer with K_1 hidden nodes. The transformation $g(\cdot)$ introduces nonlinearities in the model. Even with this single layer, with multiple nodes one can approximate arbitrarily well a rich set of smooth functions.

It may be tempting to fit this into a standard framework and interpret this model simply as a complex, but fully parametric, specification for the potentially nonlinear conditional expectation of Y_i given X_i:

$$\mathbb{E}[Y_i | X_i = x] = \sum_{k'=1}^{K_1} \beta_{k'}^{(2)} g\left(\sum_{k=1}^{K} \beta_{k'k}^{(1)} X_{ik}\right).$$

Given this interpretation, we can estimate this using nonlinear least squares. We could then derive the properties of the least squares estimators, and functions thereof, under standard regularity conditions. However, this interpretation of a neural net as a standard nonlinear model would be missing the point, for four reasons. First, it is likely that the asymptotic distributions for the parameter estimates would be poor approximations to the actual sampling distributions. Second, the estimators for the parameters would be poorly behaved, with likely substantial collinearity without careful regularization. Third, and more important, these properties are not of intrinsic interest. We are interested in the properties of the predictions from these specifications, and these can be quite attractive even if the properties of the paramater estimates are not. Fourth, we can make these models much more flexible, and at the same time, make the properties of the corresponding least squares estimators of the parameters substantially less attractive, by adding layers to the neural network. A second layer of hidden nodes would have representations that are linear in the same transformation $g(\cdot)$ of linear combinations of the first layer of hidden nodes:

$$Z_{ik}^{(2)} = \sum_{j=1}^{K_1} \beta_{kj}^{(2)} g\left(Z_{ij}^{(1)}\right), \quad \text{for } k = 1, \ldots, K_2,$$

with the outcome now a function of the second layer of hidden nodes,

$$Y_i = \sum_{k=1}^{K_2} \beta_k^{(3)} g\left(Z_{ik}^{(2)}\right) + \varepsilon_i.$$

The depth of the network substantially increases the flexibility in practice, even if with a single layer and many nodes we can already approximate a very rich set of functions. In applications researchers have used models with many layers, *e.g.*, ten or more, and millions of parameters:

> "We observe that shallow models [models with few layers] in this context overfit at around 20 millions parameters while deep ones can benefit from having over 60 million. This suggests that using a deep model expresses a useful preference over the space of functions the model can learn." LeCun et al. [2015])

In cases with multiple hidden layers and many hidden nodes one needs to carefully regularize the parameter estimation, possibly through a penalty term that is proportional to the sum of the squared coefficients in the linear parts of the model. The architecture of the networks is also important. It is possible, as in the specification above, to have the hidden nodes at a particular layer be a linear function of all the hidden nodes of the previous layer, or restrict them to a subset based on substantive considerations (*e.g.,* proximity of covariates in some metric, such as location of pixels in a picture). Such *convolutional* networks have been very succesful, but require even more careful tuning (Krizhevsky et al. [2012]).

Estimation of the parameters of the network is based on approximately minimizing the sum of the squared residuals plus a penalty term that depends on the complexity of the model. This minimization problem is challenging, especially in settings with multiple hidden layers. The algorithms of choice use the *back-propagation* algorithm and variations thereon (Rumelhart et al. [1986]) to calculate the exact unit-level terms in the objective function and the associated derivatives, exploiting the hierarchical structure of the layers, and the fact that each parameter enters only in a single layer. The algorithms then use stochastic gradient descent (Friedman [2002], Bottou [1998, 2012]), described in detail above, as a computationally efficient method for finding the approximate optimum.

3.4 Boosting

Boosting is a general purpose technique to improve the performance of simpler methods. See Schapire and Freund [2012] for a detailed discussion. Let us say we are interested

in prediction of an outcome given a substantial number of features. Suppose we have a very simple algorithm for prediction, a *simple base learner*. For example, we could have a regression tree with three leaves, that is, a regression tree based on a two splits, where we estimate the regression function as the average outcome in the corresponding leaf. That is in itself not a very attractive predictor in terms of predictive performance because it at most uses two of the many features. Boosting improves this in the following way. Take for all units in the training sample the residual from the prediction based on the simple model, $Y_i - \hat{Y}_i^{(1)}$. Now we apply the same method (say the two split regression tree) with the residuals as the outcome of interest (and with the same set of original features). Given this new tree we can calculate the new residual, $Y_i - \hat{Y}_i^2$, where $\hat{Y}_i^{(2)}$ combines the prediction from the first and second steps. We can then repeat this step, using this new residual as the outcome and again construct a two split regression tree. We can do this many times, and get a prediction based on re-estimating the basic model many times on the updated residuals.

If we do this using as the basic estimator a regression tree with L splits, it turns out that the resulting predictor can approximate any regression function that can be written as the sum of functions of L of the original features at a time. So, with $L = 1$, we can approximate any function that is additive in the features, and with $L = 2$ we can approximate any function that is additive in functions that allow for general second order effects.

Boosting can be applied using base estimator other than regression trees, e.g., neural nets.

4 Supervised Learning for Classification Problems

Classification problems are the focus of the other main branch of the supervised learning literature. The problem is, given a set of observations on a vector of features X_i, and a label Y_i (an unordered discrete outcome), the goal is a function that assigns new units, on the basis of their features, to one of the labels. This is very closely related to discrete choice analysis in econometrics, where researchers specify models that ultimately lead to a probability, conditional on the covariates/features, that the outcome takes on a particular value. Given such an probability, estimated or true, it is of course straightforward to predict a unique label, namely the one with the highest probability. There are differences between the two approaches. An important one is that in the classification literature the focus is often solely on the classification, the choice of a single label. One can classify given a probability

for each label, but one does not need such a probability to do the classification. Many of the classification methods do not, in fact, first estimate a probability for each label, and so are not directly relevant in settings where such a probability is required. A practical difference is that the classification literature has often focused on settings where ultimately the covariates allow one to assign the label with almost complete certainty, as opposed to settings where the even the best methods have high error rates.

The classic example is that of digit recognition. Based on a picture, coded as a set of say 16 or 256 black and white pixels, the challenge is to classify the image as corresponding to one of the ten digits from 0 to 9. Here ML methods have been spectacularly successful. Support Vector Machines (SVMs Cortes and Vapnik [1995]) greatly outperformed other methods in the nineties, and more recently deep convolutional neural networks (Krizhevsky et al. [2012]) have improved error rates even further.

4.1 Classification Trees and Forests

Trees and random forests are easily modified from a focus on estimation of regression functions to classification tasks. See Breiman et al. [1984] for a general discussion. Again we start by splitting the sample into two leaves, based on a single covariate exceeding or not a threshold. We optimize the split over the choice of covariate and the threshold. The difference between the regression case and the classification case is in the objective function that measures the improvement from a particular split. In classification problems this is called the *impurity* function. It measures, as a function of the shares of units in a given leaf with a particular label, how *impure* that particular leaf is. If there are only two labels, we could simply assign the labels the numbers zero and one, interpret the problem as one of estimating the conditional mean and use the average squared residual as the impurity function. That does not generalize naturally to the multi-label case. Instead a more common impurity function, as a function of the K shares $p_1, \ldots, p_K$ is the Gini impurity,

$$I(p_1, \ldots, p_K) = -\sum_{k=1}^{K} p_k \ln(p_k).$$

This impurity function is minimized if the leaf is pure and all units in that leaf have the same label, and is maximized if the shares are all equal to $1/K$. The regularization typically works again through a penalty term on the number of leaves. The same extension from a single tree to a random forest that was discussed for the regression case works for the classification case.

4.2 Support Vector Machines and Kernels

SVMs are another flexible set of methods for classification analyses (see Scholkopf and Smola [2001] for a textbook discussion). They can also extended to regression settings, but are more naturally discussed in a classification context. Suppose we have a set with N observations on a K-dimensional vector of features X_i and a binary label $Y_i \in \{-1, 1\}$. Given a K-vector ω and a constant b, define the hyperplane $x \in \mathbb{R}$ such that $\omega^\top x + b = 0$. We can think of this hyperplane defining a binary classifier $\text{sgn}(\omega^\top X_i + b)$, with units i with $\omega^\top x + b > 0$ classified as 1 and units with $\omega^\top x + b < 0$ classified as -1. Now consider for each hyperplane the number of classification errors in the sample. If we are very fortunate there would be some hyperplanes with no classification errors. In that case there are typically many such hyperplanes, and we choose the one that maximizes the distance to the closest units. There will typically be a small set of units that have the same distance to the hyperplan (the same *margin*). These are called the *support vectors*.

We can write this as an optimization problem as

$$\min_{\omega,b} \|\omega\|^2, \qquad \text{subject to } Y_i(\omega^\top X_i + b) \geq 1, \quad \text{for all } i = 1, \ldots, N.$$

with classifier

$$\text{sgn}(\hat{\omega}^\top X_i + \hat{b}).$$

Note that if there is a hyperplane with no classification errors, a standard logit model would not have a maximum likelihood estimator: the argmax of the likelihood function would diverge.

We can also write this problem in terms of the Lagrangian,

$$\min_{\alpha,\omega,b} \left\{ \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{N} \alpha_i (Y_i(\omega^\top X_i + b) - 1) \right\}, \qquad \text{subject to } 0 \leq \alpha_i,$$

which, after concentrating out the weights ω, is equivalent to

$$\max_{\alpha} \left\{ \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j Y_i Y_i X_i^\top X_j \right\}, \qquad \text{subject to } 0 \leq \alpha_i, \quad \sum_{i=1}^{N} \alpha_i Y_i = 0,$$

with classifier

$$f(x) = \text{sgn}\left(b + \sum_{i=1}^{N} Y_i \alpha_i X_i^\top x \right),$$

where b solves $\alpha_i(Y_i(X_i^\top\omega + b) - 1) = 0$.

In practice, of course, we are typically in a situation where there exists no hyperplane that corresponds to no classification errors. In that case there is no solution as the α_i diverge for some i. We can modify the classifier by adding the constraint that the $\alpha_i \leq C$. Scholkopf and Smola [2001] recommend $C = 10N$.

This is still a linear problem, differing from a logistic regression only in terms of the loss function. Units far away from the hyperplane do not affect the estimator as much in the SVM approach as they do in a logistic regression, leading to more robust estimates. However, the real power from the SVM approach is in the nonlinear case. We can think of that in terms of constructing a number of functions of the original covariates, $\phi(X_i)$, and then finding the optimal hyperplane in the transformed feature space. However, because the features enter only through the inner product $X_i^\top X_j$, it is possible to skip the step of specifying the transformations $\phi(\cdot)$, and instead directly write the classifier in terms of a kernel $K(x, z)$,

$$\max_\alpha \sum_{i=1}^N \left\{ \alpha_i - \frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N \alpha_i\alpha_j Y_i Y_i K(X_i, X_j) \right\}, \quad \text{subject to } 0 \leq \alpha_i \leq C, \ \sum_{i=1}^N \alpha_i Y_i = 0,$$

with classifier

$$f(x) = \text{sgn}\left(\sum_{i=1}^N Y_i\alpha_i K(X_i, x) + b \right).$$

Common choices for the kernel are $k(x, z) = \exp(-(x-z)^\top(x-z)/h)$, or $k(x, z) = \tanh(\kappa(x-z)^\top(x - z) + \Theta)$. The parameters of the kernel, capturing the amount of smoothing, are typically chosen through crossvalidation.

5 Unsupervised Learning

A second major topic in the ML literature is unsupervised learning. Here the goal i,s given a set of observations on features X_i, to partition the feature space into a number of subspaces. The result may be used to to create new features, based on subspace membership. For example, we may wish to use the partioning to estimate parsimonious models within each of the subspaces. This is an unusual problem, in the sense that there is no natural benchmark to assess whether a particular solution is a good one relative to some other one.

A key method is the k-means algorithm (Hartigan and Wong [1979], Alpaydin [2009]). Consider the case where we wish to partition the feature space into K subspaces or clusters.

We wish to choose centroids $b_1, \ldots, b_K$, and then assign units to the cluster based on their proximity to the centroids. The basic algorithm works as follows. Given a set of centroids, assign each unit to the cluster that minimizes the distance between the unit and the centroid of the cluster:

$$C_i = \arg \min_{c \in \{1,\ldots,K} \|X_i - b_c\|^2 .$$

Then update the centroids:

$$b_c = \sum_{i:C_i=c} X_i \Big/ \sum_{i:C_i=c} 1.$$

Repeatedly interate between the two steps.

Choosing the number of clusters K is difficult because there is no direct cross-validation method to assess the performance of one value versus the other.

There are a large number of alternative unsupervised methods, including topic models, which we discuss further below in the section about text. Unsupervised variants of neural nets are particularly popular for images and videos.

6 Machine Learning and Causal Inference

An important difference between much of the econometrics literature and the machine learning literature is that the econometrics literature is often focused on questions beyond simple prediction. In many, arguably most, cases, researchers are interested in average treatment effects or other causal or structural parameters (see Abadie and Cattaneo [2018] and Imbens and Wooldridge [2009] for surveys). Covariates that are of limited importance for prediction may still play an important role in estimating such structural parameters.

6.1 Average Treatment Effects

A canonical problem is that of estimating average treatment effects under unconfoundedness (Rosenbaum and Rubin [1983], Imbens and Rubin [2015]). Given data on an outcome Y_i, a binary treatment W_i, and a vector of covariates or features X_i, the estimand, the average treatment effect (ATE) is defined as $\tau = \mathbb{E}[Y_i(1) - Y_i(0)]$, where $Y_i(w)$ is the potential outcome unit i would have experienced if their treatment assignment had been w. Under the unconfoundedness assumption, which ensures that potential outcomes are independent of the treatment assignment conditional on covariates ($W_i \perp\!\!\!\perp (Y_i(0), Y_i(1))|X_i$), the ATE is

identified. The ATE can be written in an number of different ways: (i) as the covariate-adjusted difference between the two treatment groups, (ii) as a weighted average of the outcomes, and (iii) in terms of the influence function.

$$\begin{aligned}\tau &= \mathbb{E}\left[\mathbb{E}\left[Y_i \,|W_i=1,X_i\right]-\mathbb{E}\left[Y_i \,|W_i=0,X_i\right]\right] \\ &= \mathbb{E}\left[\frac{Y_iW_i}{\mathbb{E}[W_i|X_i]}-\frac{Y_i(1-W_i)}{1-\mathbb{E}[W_i|X_i]}\right] \\ &= \mathbb{E}\left[\frac{(Y_i-\mathbb{E}[Y_i|W_i=1,X_i])W_i}{\mathbb{E}[W_i|X_i]}-\frac{(Y_i-\mathbb{E}[Y_i|W_i=0,X_i])(1-W_i)}{1-\mathbb{E}[W_i|X_i]}\right] \\ &\quad +\mathbb{E}\left[\mathbb{E}\left[Y_i \,|W_i=1,X_i\right]-\mathbb{E}\left[Y_i \,|W_i=0,X_i\right]\right].\end{aligned} \tag{6.3}$$

One can estimate the average treatment effect by using the first representation by estimating $\mathbb{E}[Y_i|W_i=w,X_i]$, using the second representation by estimating $\mathbb{E}[W_i|X_i]$, or using the third representation and estimating both. Given a particular choice of representation to mimic, there is the question of the appropriate estimator for the particular conditional expectations that enter into that representation. For example, if we wish to use the first representation, and want to consider linear models, it may seem natural to use LASSO or subset selection. However, as illustrated in Belloni et al. [2014], such a strategy could have very poor properties. The set of features that is optimal for inclusion when the objective is estimating $\mathbb{E}[Y_i|W_i,X_i]$ is not necessarily optimal for estimating τ. The reason is that omitting from the regression covariates that are highly correlated with the treatment W_i can introduce substantial biases even if their correlation with the outcome is only modest. Thus, optimizing model selection for predicting outcomes is not the best approach. Belloni et al. [2014] propose using a covariate selection method that selects both covariates that are predictive of the outcome and covariates that are predictive of the treatment, and show that this improves the properties of the corresponding estimator for τ.

More recent methods focus on combinations of estimating $\mathbb{E}[Y_i|W_i=w,X_i]$ and $\mathbb{E}[W_i|X_i]$ flexibly and combining them in doubly robust methods (Robins and Rotnitzky [1995], Chernozhukov et al. [2016a,b]), and methods that combine estimating $\mathbb{E}[Y_i|W_i=w,X_i]$ with covariate balancing (Athey et al. [2016a]). Covariate balancing is inspired by another common approach in ML, which is frame data analysis as an optimization problem. Here, instead of trying to estimate a primitive object, the propensity score ($\mathbb{E}[W_i|X_i]$), the optimization procedure directly optimizes weights for the observations that lead to the same mean values of covariates in the treatment and control group. This approach allows for efficient estimation of average treatment effects even when the propensity score is too complex to estimate well. Since traditional propensity score weighting entails dividing by the estimated propen-

sity score, instability in propensity score estimation can lead to variance in average treatment effect estimates. Further, in an environment with many potential confounders, estimating the propensity score using regularization may lead to the omission of weak confounders that still contribute to bias. Directly optimizing for balancing weights can be more effective in environments with many weak confounders.

The case of estimating average treatment effects under unconfoundedness is an example of a more general theme from econometrics; typically, economists prioritize consistent estimates of causal effects above predictive power (see Athey [2017, 2018] for further elaboration of this point). In instrumental variables models, it is common that goodness of fit falls by an order of magnitude between an ordinary least squares regression and the second stage of a two-stage least squares model. However, the instrumental variables estimate of causal effects can be used to answer questions of economic interest, and so the loss of predictive power is considered the price that must be paid for estimating the object of interest.

6.2 Orthogonalization and Cross-Fitting

A theme that has emerged across multiple distinct applications of machine learning to parameter estimation is that both practical performance and theoretical guarantees can be improved by using two simple tricks, both involving "nuisance parameters" that are estimated using machine learning. These can be illustrated through the lens of ATE estimation. Building from the third representation in (6.3), we can define the influence function of each observation as follows:

$$\Gamma_i = \mu(1, X_i) - \mu(0, X_i) + \frac{W_i}{e(X_i)}(Y_i - \mu(1, X_i) + \frac{1 - W_i}{1 - e(X_i)}(Y_i - \mu(0, X_i)).$$

An estimate of the ATE can be constructed by first constructing estimates $\hat{\mu}(w, x)$ and $\hat{e}(x)$, and plugging those in to get an estimate $\hat{\Gamma}_i$ for each observation. Then, the sample average of $\hat{\Gamma}_i$ becomes an estimator for the ATE. This approach is analyzed in Chernozhukov et al. [2017] for the average treatment effect case, and in Bickel et al. [1998] and Van der Vaart [2000] for the general semiparametric case. A key result is that an estimator based on this approach is efficient if the estimators are reasonably accurate in the following sense:

$$\mathbb{E}[(\hat{\mu}(w, X_i) - \mu(w, X_i))^2]^{\frac{1}{2}} \mathbb{E}[(\hat{e}(X_i) - e(X_i))^2]^{\frac{1}{2}} = o_P\left(\frac{1}{\sqrt{n}}\right).$$

For example, each nuisance component ($\hat{\mu}$ and $\hat{e}$) could converge at rate $\frac{1}{n^{1/4}}$, an order

of magnitude slower than the ATE estimate. This works because Γ_i makes use of orthogonalization; by construction, errors in estimating the nuisance components are orthogonal to errors in Γ_i. This idea is more general, and has been exploited in a series of papers, with theoretical analysis in Chernozhukov et al. [2018a,c], and other applications including Athey et al. [2017d] for estimating heterogeneous effects in models with unconfoundedness or those that make use of instrumental variables.

A second idea, also exploited in the same series of papers, is that performance can be improved using techniques such as sample splitting, cross-fitting, out-of-bag prediction, and leave-one-out estimation. All of these techniques have the same final goal: nuisance parameters estimated to construct the influence function $\hat{\Gamma}_i$ for observation i (for the ATE case, $\hat{\mu}(w, X_i)$ and $\hat{e}(X_i)$) should be estimated without using outcome data about observation i. When random forests are used to estimate the nuisance parameters, this is straightforward, since "out of bag" predictions (standard in random forest statistical packages) provide the predictions obtained using trees that were constructed without using observation i. When other types of ML models are used to estimate the nuisance parameters, "cross-fitting" or sample splitting advocates splitting the data into folds and estimating the nuisance parameters separately on all data except a left-out fold, and then predicting the nuisance paramters in the left-out fold. When there are as many folds as observations, this is known as leave-one-out estimation.

Although these two observations are helpful in traditional "small data" applications, when ML is used to estimate nuisance parameters (because, e.g., there are many covariates), the observations become much more important. Overfitting is more of a concern, and in particular, a single observation i can have a strong effect on the predictions made for covariates X_i when the model is very flexible. Cross-fitting can solve this problem. Second, we should expect that with many covariates relative to the number of observations, accurate estimation of nuisance parameters is harder to achieve. Thus, orthogonalization makes estimation more robust to these errors.

6.3 Heterogenous Treatment Effects and Estimating Optimal Policy Functions

Another place where machine learning can be very useful is in uncovering treatment effect heterogeneity, where we focus on heterogeneity with respect to observable covariates. Examples of questions include, which individuals benefit most from a treatment? For which

individuals is the treatment effect positive? How do treatment effects change with covariates? Understanding treatment effect heterogeneity can be useful for basic scientific understanding, or for estimating optimal policy assignments; see Athey and Imbens [2017] for further discussion.

Continuing with the potential outcome notation from the last subsection, we define the conditional average treatment effect (CATE) as $\tau(x) = E[Y_i(1) - Y_i(0)|X_i = x] = E[\tau_i|X_i = x]$, where $\tau_i = Y_i(1) - Y_i(0)$ is the treatment effect for individual i. The CATE is identified under the unconfoundedness assumption introduced in the last subsection. Note that τ_i can not be observed for any unit; this "fundamental problem of causal inference" (Holland [1986]) is the source of an apparent difference between estimating heterogeneous treatment effects and predicting outcomes, which are typically observed for each unit.

We focus here on three types of questions: (i) learning a low-dimensional representation of treatment effect heterogeneity, and conducting hypothesis tests about this heterogeneity;(ii) learning a flexible (nonparametric) estimate of $\tau(x)$; and (iii) estimating an optimal policy allocating units to either treatment or control on the basis of covariates x.

An important issue in adapting machine learning methods to focus on causal parameters relates to the criterion function used in model selection. Predictive models typically use a mean squared error (MSE) criterion, $\frac{1}{N}\sum_i (Y_i - \hat{\mu}(X_i))^2$ to evaluate performance. Although the MSE in a held-out test set is a noisy estimate to the population expectation of the MSE in an independent set, the sample average MSE is a good approximation that does not rely on further assumptions (beyond independence of observations), and the standard error of the squared errors in the test set accurately captures the uncertainty in the estimate. In contrast, consider the problem of estimating the CATE in observational studies. It would be natural to use as a criterion function the mean squared error of treatment effects, $\frac{1}{N}\sum_i (\tau_i - \hat{\tau}(X_i))^2$, where $\hat{\tau}(x)$ is the estimate of the CATE. However, this criterion is infeasible, since we can never observe unit-level causal effects. Further, there is no simple, model-free unbiased estimate of this criterion in observational studies. For this reason, comparing models, and as a result developing regularization strategies, is a much bigger challenge in settings where we are interested in structural or causal parameters than in settings where we are interested in predictive performance.

These difficulties in finding effective cross-validation strategies are not always unsurmountable, but they lead to a need to carefully adapt and modify basic regularization methods to address the questions of interest. Athey and Imbens [2016] proposes several different

possible criterion to use for optimizing splits as well as for cross-validation. A first insight is that when conducting model selection, it is only necessary to compare models. The τ_i^2 term (which would be hard to estimate) cancels out when comparing two estimators, say $\hat{\tau}'(x)$ and $\hat{\tau}''(x)$. The remaining terms are linear in τ_i, and the expected value of τ_i can be estimated. If we let $e(x)$ be the propensity score, so that $e(x) = Pr(W_i = w|X_i = x)$, and we define

$$Y_i^* = W_i \frac{Y_i}{e(X_i)} - (1 - W_i) \frac{Y_i}{1 - e(X_i)},$$

then $\mathbb{E}[Y_i^*] = \mathbb{E}[\tau_i]$. When the propensity score is unknown, it must be estimated, which implies that a criterion based on an estimate of the mean squared error of the CATE will depend on modeling choices.

Athey and Imbens [2016] build on this insight and propose several different estimators for the MSE of the CATE. They develop a method, which they call "causal tree," for learning a low-dimensional representation of treatment effect heterogeneity, where the method is interpretable and provides reliable confidence intervals for the parameters it estimates. The paper builds on regression tree methods, creating a partition of the covariate space and then estimating treatment effects in each element of the partition. Unlike regression trees optimized for prediction, the splitting rule optimizes for finding splits associated with treatment effect heterogeneity. In addition, the method makes use of sample splitting; half the data is used to estimate the tree structure, and the other half (the "estimation sample") is used to estimate treatment effects in each leaf. The tree is pruned using cross-validation, just as in standard regression trees, but where the criterion for evaluating the performance of the tree in held-out data is based on treatment effect heterogeneity rather than predictive accuracy.

Some advantages of the causal tree method are similar to advantages of regression trees. They are easy to explain; in the case of a randomized experiment, the estimate in each leaf is simply the sample average treatment effect. A disadvantage is that the tree structure is somewhat arbitrary; there may be many partitions of the data that exhibit treatment effect heterogeneity, and taking a slightly different subsample of the data might lead to a different estimated partition. The approach of estimating simple models in the leaves of shallow trees can be applied to other types of models; see [Zeileis et al., 2008] for an early version of this idea, although that paper did not provide theoretical guarantees or confidence intervals.

For some purposes, it is desirable to have a fully non-parametric estimate of $\tau(x)$. For example, if a treatment decision must be made for a particular individual with covariates x,

a regression tree may give a biased estimate for that individual given that the individual may not be in the center of the leaf, and that the leaf may contain other units that are distant in covariate space. In the traditional econometrics literature, non-parametric estimation could be accomplished through kernel estimation or matching techniques. However, even though they work well in theory, they do not work well in practice with many covariates. Wager and Athey [2017] introduces "causal forests." Essentially, a causal forest is the average of a large number of causal trees, where trees differ from one another due to subsampling. Similar to prediction forests, a causal forest can be thought of as a version of a nearest neighbor matching method, but one where there is a data-driven approach to determine which dimensions of the covariate space are important to match on. The paper establishes asymptotic normality of the estimator (so long as tree estimation is "honest," making use of sample splitting for each tree) and provides an estimator for the variance of estimates so that confidence intervals can be constructed.

A challenge with forests is that it is difficult to describe the output, since the estimated CATE function $\hat{\tau}(x)$ may be quite complex. However, in some cases one might wish to test simpler hypotheses, such as the hypothesis that the top 10% of individuals ranked by their CATE have a different average CATE than the rest of the population. Chernozhukov et al. [2018b] provides methods for testing this type of hypothesis.

As described above in our presentation of regression forests, Athey et al. [2016b] extended the framework of causal forests to analyze nonparametric parameter heterogeneity in models where the parameter of interest can be estimated by maximum likelihood or GMM. As an application, the paper highlights the case of instrumental variables. Friedberg et al. [2018] extends local linear regression forests to the problem of heterogeneous treatment effects, so that regularity in the function $\tau(x)$ can be better exploited.

An alternative approach to estimating parameter heterogeneity in instrumental variables models was proposed by Hartford et al. [2016], who use an approach based on neural nets, though distributional theory is not available for that estimator.

Other possible approaches to estimating conditional average treatment effects can be used when the structure of the heterogeneity is assumed to take a simple form. Targeted maximum likelihood [van der Laan and Rubin, 2006] is one approach to this, while Imai et al. [2013] proposed using LASSO to uncover heterogeneous treatment effects. Künzel et al. [2017] proposes an ML approach using "meta-learners." Another popular alternative that takes a Bayesian approach is Bayesian Additive Regression Trees (BART), developed

by Chipman et al. [2010] and applied to causal inference by Hill [2011], Green and Kern [2012].

A main motivation for understanding treatment effect heterogeneity is that the CATE can be used to define policy assignment functions, that is, functions that map from the observable covariates of individuals to policy assignments. A simple way to define a policy is to estimate $\hat{\tau}(x)$ and to assign the treatment to all individuals with positive values of $\hat{\tau}(x)$, where the estimate should be augmented with any costs of being in the treatment or control group. Hirano and Porter [2009] shows that this is optimal under some conditions. A concern with this approach, depending on the method used to estimate $\hat{\tau}(x)$, is that the policy may be very complex and is not guaranteed to be smooth.

Kitagawa and Tetenov [2015] focus on estimating the optimal policy from a class of potential policies of limited complexity in an observational study with known propensity scores. The goal is to select a policy function to minimize the loss from failing to use the (infeasible) ideal policy, referred to as the "regret" of the policy. Athey and Wager [2017] also studies policies with limited complexity and accomodates other constraints, such as budget constraints on the treatment, and proposes an algorthm for estimating optimal policies. The paper provides bounds on the performance of its algorithm for the case where the data come from an observational study under confoundedness and the propensity score is unknown. The paper also extends the analysis to settings that do not satisfy unconfoundedness, for example, to settings where there is an instrumental variable.

For the case of unconfoundedness, the policy estimation procedure recommended by Athey and Wager [2017] can be written as follows, where Π is the set of functions $\pi : \mathbb{X} \to 0, 1$, and $\hat{\Gamma}_i$ is defined above and makes use of cross-fitting as well as orthogonalization:

$$\max_{\pi \in \Pi} \sum_i (2\pi(X_i) - 1) \cdot \hat{\Gamma}_i \tag{6.4}$$

The topic of optimal policy estimation has received some attention in the ML literature, focusing on data from observational studies with unconfoundedness, including Strehl et al. [2010], Dudik et al. [2011], Li et al. [2012], Dudik et al. [2014], Li et al. [2014], Swaminathan and Joachims [2015], Jiang and Li [2016], Thomas and Brunskill [2016], Kallus [2017]. Athey and Wager [2017] shows how bringing in insights from semi-parametric efficiency theory enables tighter bounds on performance than the ML literature, thus narrowing down substantially the set of algorithms that might achieve the regret bound.

One insight that comes out of the ML approach to this problem is that the optimization problem 6.4 can be reframed as a classification problem and thus solved with off-the-shelf classification tools. See Athey and Wager [2017] for details.

7 Experimental Design, Reinforcement Learning, and Multi-Armed Bandits

ML methods have recently made substantial contributions to experimental design, with multiarmed bandits becoming more popular especially in online experiments. Thompson sampling (Scott [2010], Thompson [1933]) can be viewed as a simple example of reinforcement learning (Sutton et al. [1998]) where successful assignment decisions are rewarded by sending more units to the corresponding treatment arm.

7.1 A/B Testing versus Multi-Armed Bandits

Traditionally much experimentation is done by assigning a predetermined number of units to each of a number of treatment arms. Often there would be just two treatment arms. After the outcomes are measured the average effect of the treatment would be estimated using the difference in average outcomes by treatment arm. This is a potentially very inefficient way of experimentation. Suppose what we are interested in is primarly finding a treatment that is good among the set of treatments considered, rather than in testing hypotheses on the full set of treatments. Moreover, suppose that we measure the outcomes quickly after the treatments have been assigned, and suppose the units arrive sequentially. After outcomes for half the units have been observed, we may have a pretty good idea which of the treatments are still candidates for the optimal treatment. Exposing more units to treatments that are no longer competitive does not serve any purpose: it does not help us distinguish between the remaining candidate optimal treatments, and it exposes those units to inferior treatments.

Multi-armed bandit approaches (Scott [2010], Thompson [1933]) attempt to improve over this static design. In the extreme case, the assignment for each unit depends on all the information learned up to that point. Given that information, and given a parametric model for the outcomes for each treatment, and a prior for the parameters of these models, we can estimate the probability of each treatment being the optimal one. Thompson sampling suggests assigning the next unit to each treatment with probability equal to the probability that that particular treatment is the optimal one. This means that the probability of assign-

ment to a treatment arm for which we are confident that it is inferior to some of the other treatments is low, and eventually all new units will be assigned to the optimal treatment with probability close to one.

To provide some more intuition, consider a case with K treatments where the outcome is binary, so the model is a binomial distribution with treatment-arm-specific success probability p_k, $k = 1, \ldots, K$. If the prior distribution for all probabilities is uniform, the posterior distribution for the success probability of arm k, given M_k succeses in N_k trials, is Beta with parameters $M_k + 1$ and $N_k - M_k + 1$. Given that it is simple to approximate by simulation the probability that treatment arm k is the optimal one (the one with the highest success probability), $\text{pr}(p_k = \max_{m=1}^{K} p_m)$.

We can simplify the calculations by updating the assignment probabilities only after seeing a number of new observations. That is, we re-evaluate the assignment probabilities after a batch of new observations has come in, all based on the same assignment probabilities. From this perspective we can view a standard A/B experiment as one where the batch is the full set of observations. From that perspective it is clear that to, at least occasionally, update the assignment probabilities to avoid sending units to inferior treatments, is a superior strategy.

An alternative approach is to use the Upper Confidence Bounds (UCB, Lai and Robbins [1985]) approach. In that case we construct a 100(1-p)% confidence interval for the population average outcome μ_k for each treatment arm. We then collect the upper bounds of these confidence intervals for each treatment arm and assign the next unit to the treatment arm with the highest value for the upper confidence bounds. As we get more and more data, we let one minus the level of the confidence intervals p go to zero slowly. With UCB methods we need to be more careful with if we wish to update assignments only after batches of units have come in. If two treatment arms have very similar UCBs, assigning a large number of units to the one that is slightly superior may not be satisfactory: here Thompson sampling would assign similar numbers of units to both those treatment arms. More generally, the stochastic nature of the assignment under Thompson sampling, compared to the deterministic assignment in the UCB approach, has conceptual advantages for inference.

7.2 Contextual Bandits

The most important extension of multiarmed bandits is to settings where we observe features of the units that can be used in the assignment mechanism. If treatment effects are het-

erogeneous, and that heterogeneity is associated with observed characteristics of the units, there may be substantial gains from assigning units to different treatments based on these characteristics. See Dimakopoulou et al. [2017].

A simple way to incorporate covariates would be to build a parametric model for the expected outcomes in each treatment arm (the *reward function*, estimate that given the current data and infer from there the probability that a particular arm is optimal for a new unit conditional on the characteristics of that unit. This is conceptually a straightforward way to incorporate characteristics, but it has some drawbacks. The main concern is that such methods may implicitly rely substantially on the model being correct. It may be the case that the data for one treatment arm come in with a particular distribution of the characteristics, but it gets used to predict outcomes for units with very different characteristics. See Bastani and Bayati [2015] for some discussion. A risk is that if the algorithm estimates a simple linear model mapping characteristics to outcomes, then the algorithm may estimate a great deal of certainty about outcomes for an arm in a region of characteristic space where the arm has never been observed. This can lead the algorithm to never experiment with the arm in that region, allowing for the possibility that the algorithm never corrects its mistake and learns the true optimal policy.

As a result one should be careful in building a flexible model relating the characteristics to the outcomes. Dimakopoulou et al. [2017] highlight the benefits of using random forests as a way to avoid making functional form assumptions.

Beyond this issue, a number of novel considerations that arise in contextual bandits. Because the assignment rules as a function of the features changes as more units arrive, and tend to assign more units to a given arm in regions of the covariate space where it has performed well in the past, particular care has to be taken to eliminate biases in the estimation of the reward function. Thus, although there is formal randomization, the issues concerning robust estimation of conditional average causal effects in observational studies become relevant here. One solution, motivated by the literature on causal inference, is to use propensity score weighting of outcome models. Dimakopoulou et al. [2017] studies bounds on the performance of contextual bandits under propensity weighting, and also demonstrates on a number of real-world datasets that propensity weighting improves performance.

Another insight is that it can be useful to make use of simple assignment rules, particularly in early stages of bandits, because complex assignment rules can lead to confounding later. In particular, if a covariate is related to outcomes and is used in assignment, then

later estimation much control for this covariate to eliminate bias. For this reason, LASSO, which selects a sparse model, can perform better than Ridge, which places weights on more covariates, when estimating an outcome model that will be used to determine the assignment of units in subsequent batches. Finally, flexible outcome models can be important in certain settings; random forests can be a good alternative in those cases.

8 Matrix Completion and Recommender Systems

So far the methods we have discussed are primarily for settings where we observe information on a number of units in the form of a single outcome and a set of covariates or features, what is known in the econometrics literature as a cross-section setting. There are also many interesting new methods for settings that resemble what are in the econometric literature referred to as longitudinal or panel data settings. Here we discuss a canonical version of that problem, and then consider some specific methods.

8.1 The Netflix Problem

The Netflix Prize Competition was set up in 2006 (Bennett et al. [2007]), and asked researchers to use a training data set to develop an algorithm that improved on the Netflix algorithm for recommending movies by providing predictions for movie ratings. Researchers were given a training data set that contained movie and individual characteristics, as well as movie ratings, and were asked to predict ratings for movie/individual pairs for which they were not given the ratings. Because of the magnitude of the prize, $1,000,000, this competition and the associated problem generated a lot of attention, and the development of new methods for this type of setting accelarated substantially as a result.

The winning solutions, and those that were competitive with the winners, had some key features. One is that they relied heavily on model averaging. Second, many of the models included matrix factorization and nearest neighbor methods.

Although this may appear at first as a problem that is very distinct from the type of problem studied in econometrics, one can cast many econometric panel data in a similar form. In settings where researchers are interested in causal effects of a binary treatment, one can think of the realized data as consisting of two incomplete matrices, one for the outcomes given the treatment, and one for the outcomes given the control treatment. Hence the problem of estimating the average treatment effects can be cast as a matrix completion

problem. Suppose we observe outcomes on N units, over T time periods, with the outcome for unit i at time period t denoted by Y_{it}, and a binary treatment, denoted by W_{it}, with

$$\mathbf{Y} = \begin{pmatrix} Y_{11} & Y_{12} & Y_{13} & \ldots & Y_{1T} \\ Y_{21} & Y_{22} & Y_{23} & \ldots & Y_{2T} \\ Y_{31} & Y_{32} & Y_{33} & \ldots & Y_{3T} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Y_{N1} & Y_{N2} & Y_{N3} & \ldots & Y_{NT} \end{pmatrix} \qquad \text{(realized outcome)}.$$

$$\mathbf{W} = \begin{pmatrix} 1 & 1 & 0 & \ldots & 1 \\ 0 & 0 & 1 & \ldots & 0 \\ 1 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 1 & \ldots & 0 \end{pmatrix} \qquad \text{(binary treatment)}.$$

We can think of there being two matrices with potential outcomes,

$$\mathbf{Y}(0) = \begin{pmatrix} ? & ? & Y_{13} & \ldots & ? \\ Y_{21} & Y_{22} & ? & \ldots & Y_{2T} \\ ? & Y_{32} & ? & \ldots & Y_{3T} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ ? & Y_{N2} & ? & \ldots & Y_{NT} \end{pmatrix} \qquad \text{(potential control outcome)},$$

and

$$\mathbf{Y}(1) = \begin{pmatrix} Y_{11} & Y_{12} & ? & \ldots & Y_{1T} \\ ? & ? & Y_{23} & \ldots & ? \\ Y_{31} & ? & Y_{33} & \ldots & ? \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Y_{N1} & ? & Y_{N3} & \ldots & ? \end{pmatrix} \qquad \text{(potential treated outcome)}.$$

Now the problem of estimating causal effects becomes one of imputing missing values in a matrix.

The ML literature has developed effective methods for matrix completion in settings with both N and T large, and a large fraction of missing data. We discuss some of these methods in the next section, as well as their relation to the econometrics literature.

8.2 Matrix Completion Methods for Panel Data

The matrix completion literature has focused on using low rank representations for the complete data matrix. Let us consider the case without covariates, that is, no characteristics of the units or time periods. Let $\mathbf{L}$ be the complete data matrix, and $\mathbf{Y}$ the observed data

matrix. The observed values are assumed to be equal to the corresponding values of the complete data matrix, possibly with error:

$$Y_{it} = \begin{cases} L_{it} + \varepsilon_{it} & \text{if } W_{it} = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Using the singular value decomposition, $\mathbf{L} = \mathbf{USV}^\top$, where $\mathbf{U}$ is an $N \times N$ matrix, $\mathbf{V}$ is a $T \times T$ matrix, and $\mathbf{S}$ is a rank R $N \times T$ matrix, with the only non-zero elements on the diagonal (the singular values). We are not interested in estimating the matrices $\mathbf{U}$ and $\mathbf{V}$, only in the product $\mathbf{USV}^\top$, and possibly in the singular values. Obviously some regularization is required, and an effective one is to use the nuclear norm $\|\cdot\|_*$, which is proportional to the sum of the singular values. Building on the ML literature (Candès and Recht [2009], Mazumder et al. [2010]), Athey et al. [2017a] focus on estimating $\mathbf{L}$ by minimizing

$$\min_{\mathbf{L}} \left\{ \sum_{(i,t)\in\mathcal{O}} (Y_{it} - L_{it})^2 + \lambda \|\mathbf{L}\|_* \right\},$$

where λ is a penalty parameter chosen through cross-validation. Using the nuclear norm here, rather than the rank of the matrix $\mathbf{L}$ is important for computational reasons. Using the Frobenius norm, equal to the sum of the squares of the singular values would not work because it is equal to the sum of the squared values of the matrix, and thus would lead to imputing all missing values as zeros. For the nuclear norm case there are effective algorithms that can deal with large N and large T. See Candès and Recht [2009], Mazumder et al. [2010].

8.3 The Econometrics Literature on Panel Data

The econometrics literature has studied these problems from a number of different perspectives. The panel data literature traditionally focused on fixed effect methods, and has generalized those to models with multiple latent factors (Bai and Ng [2017, 2002], Bai [2003]) that are essentially the same as the low rank factorizations in the ML literature. The difference is that in the econometrics literature there has been more focus on actually estimating the factors, and using normalizations that allow for their identification. Typically it is assumed that there is a fixed number of factors.

The synthetic control literature studied similar settings, but focused on the case with only missing values for a single row of the matrix $\mathbf{Y}$. Abadie et al. [2010, 2015] propose imputing these using a weighted average of the outcomes for other units in the same period. Doudchenko and Imbens [2016] show that the Abadie et al. [2015] methods can be viewed

as regression the outcomes for the last row on outcomes for the other units, and using the regression estimates from that to impute the missing values, in what Athey et al. [2017a] call the vertical regression. This contrasts with a horizontal regression, common in the program evaluation literature where outcomes in the last period are regressed on outcomes in earlier periods, and those estimates are used to impute the missing values. In contrast to both the horizontal and vertical regression approaches the matrix completion approach in principle attempts to exploit both stable patterns over time and stable patterns between units in imputing the missing values, and also can deal directly with more complex missing data patterns.

8.4 Demand Estimation in Panel Data

A large literature in economics and marketing focuses on estimating consumer preferences using data about their choices. A typical paper analyzes the discrete choice of a consumer who selects a single product from a set of prespecified imperfect substitutes, e.g. laundry detergent, personal computers, or cars (see, e.g., Keane et al. [2013] for a review). The literature typically focuses on one product category at the time, and typically models choices among a small number of products. Often this literature focuses on estimating cross-price elasticities, so that counterfactuals about firm mergers or price changes can be analyzed. Although it is common to incorporate individual-specific preferences for observable characteristics, such as prices and other product characteristics, there are typically a small number of latent variables in the models. A standard set-up starts with consumer i's utility for product j at time t, where

$$U_{ijt} = \mu_{ij} - \phi_i p_{jt} + \epsilon_{ijt},$$

where ϵ_{ijt} has an extreme value distribution and is independently and identically distributed across consumers, products, and time. μ_{ij} is consumer i's mean utility for product j, ϕ_i is consumer i's price sensitivity, and p_{jt} is the price of product j at time t. If the consumer selects the item with highest utility, then

$$Pr(Y_{ijt} = j) = \frac{\exp^{U_{ijt}}}{\sum_{j'} \exp^{U_{ij't}}}$$

From the machine learning perspective, a panel dataset with consumer choices might be studied using techniques from matrix completion, as described above. The model would

draw inferences from products that had similar purchase patterns across consumers, as well as consumers who had similar purchase patterns across products. However, such models would typically not be well-suited to analyze the extent to which two products are substitutes, or to analyze counterfactuals.

For example, Jacobs et al. [2014] propose using a related latent factorization approach in order to flexibly model consumer heterogeneity in the context of online shopping with a large assortment of products. They use data from medium sized online retailer. They consider 3,226 products, and aggregate up to the category x brand level to reduce to 440 "products." They do not model responses to price changes or substitution between similar products; instead, in the spirit of the machine learning literature, they evaluate performance in terms of predicting which new products a customer will buy.

In contrast to this "off-the-shelf" application of machine learning to product choice, a recent literature has emerged that attempts to combine machine learning methods with insights from the economics literature on consumer choice, typically in panel data settings. A theme of this literature is that models that take advantage of some of the structure of the problem will outperform models that will not. For example, the functional form implied by the consumer choice model from economics places a lot of structure on how products within a category interact with one another. An increase in the price of one product affects other products in a particular way, implied by the functional form. To the extent that the restrictions implied by the functional form are good approximations to reality, they can greatly improve the efficiency of estimation. Incorporating the functional forms that have been established to be effective across decades of economic research can improve performance.

On the other hand, economic models have typically failed to incorporate all of the information that is available in a panel dataset, the type of information that matrix completion methods typically exploit. In addition, computational issues have prevented economists from studying consumer choices across multiple product categories, even though in practice data about a consumer's purchases in one category is informative about the consumer's purchases in other categories; and further, the data can also reveal which products tend to have similar purchase patterns. Thus, the best-performing models from this new hybrid literature tend to exploit techniques from the matrix completion literature, and in particular, matrix factorization.

To see how matrix factorization can augment a standard consumer choice model, we can write the utility of consumer i for product j at time t as

$$U_{ijt} = \beta_i' \theta_j - \rho_i' \alpha_j p_{jt} + \epsilon_{ijt},$$

where β_i, θ_j, ρ_i, and α_j are each vectors of latent variables. The vector θ_j, for example, can be interpreted as a vector of latent product characteristics for product j, while β_i represents consumer i's latent preferences for those characteristics. The basic functional form for choice probabilities is unchanged, except that the utilities are now functions of the latent characteristics.

Such models had not been studied in the machine learning literature until recently, in part because the functional form for choice probabilities, which is nonlinear in a large number of latent parameters, makes computation challenging. In contrast, traditional machine learning models might treat all products as independently chosen (e.g. Gopalan et al. [2015]), making computation much easier. Ruiz et al. [2017] applies state-of-the-art computational techniques from machine learning (in particular, stochastic gradient descent and variational inference) together with a number of approximations in order to make the method scalable to thousands of consumers making choices over thousands of items in dozens or hundreds of shopping trips per consumer. Ruiz et al. [2017] does not make use of any data about the categories of products; it attempts to learn from the data (which incorporates substantial price variation) which products are substitutes or complements. In contrast, Athey et al. [2017b] incorporates information about product categories and imposes the assumption that consumers buy only one product per category on a given trip; the paper also introduces a nested logit structure, which allows utilities to be correlated across products within a category, thus better accounting for consumers' choices about whether to purchase a category at all.

A closely related approach is taken in Wan et al. [2017]. They use a latent factorization approach that incorporates price variation. They model consumer choice as a three stage process: (i) Choose whether to buy the category, (ii) Choose which item in category, and (iii) Choose number of the item to purchase. The paper uses customer loyalty transaction data from two different datasets.

In all of these approaches, using the utility maximization approach from economics makes it possible to perform traditional analyses such as analyzing the impact of price changes on consumer welfare.

A complementary approach to one based on latent product characteristics is the work by Semenova et al. [2018], who considers observational high-dimensional product attributes

(e.g., text descriptions and images) rather than latent features.

9 Text Analysis

There is a large machine learning literature on analyzing text. It is beyond the scope of this paper to fully describe this literature; Gentzkow et al. [2017] provides an excellent recent review. Here, we provide a high-level overview.

To start, we consider a dataset consisting of documents $i = 1, .., N$. Each document contains a set of words. One way to represent the data is as a $N \times T$ matrix, denoted C, where T is the number of words in the language, where each element of the matrix is an indicator for whether word t appears in document i. Richer reprsentations might let T be the number of bigrams, where a bigram is a pair of words that appear adjacent to one another in the document.

There are two types of exercises we can do with this type of data. One is unsupervised learning, and the other is supervised. For the unsupervised case, the goal would be to find a lower-rank representation of the matrix C. Given that a low-rank matrix can be well approximated by a factor structure, as discussed above, this is equivalent to finding a set of k latent characteristics of documents (denoted β) and a set of latent weights on these topics, denoted θ, such that the probability that word t appears in document i is a function of $\theta_i'\beta_j$). This view of the problem basically turns the problem into a matrix completion problem; we would say that a particular representation performs well if we hold out a test set of randomly selected elements of C, and the model predicts well those held-out elements. All of the methods described above for matrix completion can be applied here.

One implementation of these ideas are referred to as "topic models"; seeBlei and Lafferty [2009] for a review. These models specify a particular generative model of the data. In the model, there are a number of topics, which are latent variables. Each topic is associated with a distribution of words. An article is characterized by weights on each topic. The goal of a topic model is to estimate the latent topics, the distribution over words for each topic, and the weights for each article. A popular model that does this is known as the Latent Dirichlet Allocation model.

More recently, more complex models of language have emerged, following the theme that, although simple machine learning models perform quite well, incorporating problem-specific structure is often helpful and is typically incorporated in state-of-the art machine learning in

popular application areas. Broadly, these are known as "word embedding methods." These attempt to capture latent semantic structure in language; see [Mnih and Hinton, 2007, Mnih and Teh, 2012, Mikolov et al., 2013a,b,c, Mnih and Kavukcuoglu, 2013, Pennington et al., 2014, Levy and Goldberg, 2014, Vilnis and McCallum, 2015, Arora et al., 2016, Barkan, 2016, Bamler and Mandt, 2017]. Consider the neural probabilistic language model of Bengio et al. [2003, 2006]. That model specifies a joint probability of sequences of words, parameterized by a vector representation of the vocabulary. Vector representations of words (also known as "distributed representations") can incorporate ideas about word usage and meaning [Harris, 1954, Firth, 1957, Bengio et al., 2003, Mikolov et al., 2013b].

Another class of models uses supervised learning methods. These methods are used when there is a specific characteristic the researcher would like to learn from text. Examples might include favorability of a review, political polarization of text spoken by legislators, or whether a tweet about a company is positive or negative. Then, the outcome variable is a label that contains the characteristic of interest. A simple supervised learning model takes the data matrix C, views each document i as a unit of observation, and treats the columns of C (each corresponding to indicators for whether a particular word is in a document) as the covariates in the regression. Since T is usually much greater than N, it is important to use ML methods that allow for regularization. Sometimes, other types of dimension reduction techniques are used in advance of applying a supervised learning method (e.g. unsupervised topic modeling).

Another approach is to think of a generative model, where we think of the words in the document as a vector of outcomes, and where the characteristics of interest about the document determine the distribution of words, as in the topic model literature. An example of this approach is the supervised topic model, where information about the observed characteristics in a training dataset are incorporated in the estimation of the generative model. The estimated model can then be used to predict those characteristics in a test dataset of unlabelled documents. See Blei and Lafferty [2009] for more details.

10 Conclusion

There is a fast growing machine learning literature that has much to offer empirical researchers in economics. In this review we describe some of the methods we view as most useful for economists, and that we view as important to include in the core graduate econo-

metrics sequences. Being familar with these methods will allow researchers to do more sophisticated empirical work, and to communicate more effectively with researchers in other fields.

References

A Abadie and MD Cattaneo. Econometric methods for program evaluation. Annual Review of Economics, 18, 2018.

Alberto Abadie and Guido W Imbens. Bias-corrected matching estimators for average treatment effects. Journal of Business & Economic Statistics, 29(1):1–11, 2011.

Alberto Abadie, Alexis Diamond, and Jens Hainmueller. Synthetic control methods for comparative case studies: Estimating the effect of California's tobacco control program. Journal of the American Statistical Association, 105(490):493–505, 2010.

Alberto Abadie, Alexis Diamond, and Jens Hainmueller. Comparative politics and the synthetic control method. American Journal of Political Science, pages 495–510, 2015.

Alberto Abadie, Susan Athey, Guido W Imbens, and Jeffrey M Wooldridge. Sampling-based vs. design-based uncertainty in regression analysis. arXiv preprint arXiv:1706.01778, 2017.

Ethem Alpaydin. Introduction to machine learning. MIT press, 2009.

S. Arora, Y. Li, Y. Liang, and T. Ma. RAND-WALK: A latent variable model approach to word embeddings. Transactions of the Association for Computational Linguistics, 4, 2016.

Susan Athey. Beyond prediction: Using big data for policy problems. Science, 355(6324): 483–485, 2017.

Susan Athey. The impact of machine learning on economics. The Economics of Artificial Intelligence, 2018.

Susan Athey and Guido Imbens. Recursive partitioning for heterogeneous causal effects. Proceedings of the National Academy of Sciences, 113(27):7353–7360, 2016.

Susan Athey and Guido W Imbens. The state of applied econometrics: Causality and policy evaluation. The Journal of Economic Perspectives, 31(2):3–32, 2017.

Susan Athey and Stefan Wager. Efficient policy estimation. arXiv preprint arXiv:1702.02896, 2017. URL https://arxiv.org/abs/1702.02896.

Susan Athey, Guido Imbens, and Stefan Wager. Efficient inference of average treatment effects in high dimensions via approximate residual balancing. arXiv preprint arXiv:1604.07125, 2016a.

Susan Athey, Julie Tibshirani, and Stefan Wager. Generalized random forests. arXiv preprint arXiv:1610.01271, 2016b.

Susan Athey, Mohsen Bayati, Nikolay Doudchenko, Guido Imbens, and Khashayar Khosravi. Matrix completion methods for causal panel data models. arXiv preprint arXiv:1710.10251, 2017a.

Susan Athey, David Blei, Rob Donnelly, and Francisco Ruiz. Counterfactual inference for consumer choice across many product categories. 2017b.

Susan Athey, Markus M Mobius, and Jenő Pál. The impact of aggregators on internet news consumption. 2017c.

Susan Athey, Julie Tibshirani, and Stefan Wager. Generalized random forests. arXiv preprint arXiv:1610.01271, 2017d. URL https://arxiv.org/abs/1610.01271.

Jushan Bai. Inferential theory for factor models of large dimensions. Econometrica, 71(1): 135–171, 2003.

Jushan Bai and Serena Ng. Determining the number of factors in approximate factor models. Econometrica, 70(1):191–221, 2002.

Jushan Bai and Serena Ng. Principal components and regularized estimation of factor models. arXiv preprint arXiv:1708.08137, 2017.

R. Bamler and S. Mandt. Dynamic word embeddings via skip-gram filtering. In International Conference in Machine Learning, 2017.

O. Barkan. Bayesian neural word embedding. arXiv preprint arXiv:1603.06571, 2016.

Hamsa Bastani and Mohsen Bayati. Online decision-making with high-dimensional covariates. Technical report, 2015.

Robert M Bell and Yehuda Koren. Lessons from the netflix prize challenge. Acm Sigkdd Explorations Newsletter, 9(2):75–79, 2007.

Alexandre Belloni, Victor Chernozhukov, and Christian Hansen. High-dimensional methods and inference on structural and treatment effects. Journal of Economic Perspectives, 28 (2):29–50, 2014.

Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. Journal of Machine Learning Research, 3:1137–1155, 2003.

Y. Bengio, H. Schwenk, J.-S. Senécal, F. Morin, and J.-L. Gauvain. Neural probabilistic language models. In Innovations in Machine Learning. Springer, 2006.

James Bennett, Stan Lanning, et al. The netflix prize. In Proceedings of KDD cup and workshop, volume 2007, page 35. New York, NY, USA, 2007.

Dimitris Bertsimas, Angela King, Rahul Mazumder, et al. Best subset selection via a modern optimization lens. The Annals of Statistics, 44(2):813–852, 2016.

Peter Bickel, Chris Klaassen, Yakov Ritov, and Jon Wellner. Efficient and adaptive estimation for semiparametric models. 1998.

Herman J Bierens. Kernel estimators of regression functions. In Advances in econometrics: Fifth world congress, volume 1, pages 99–144, 1987.

David M Blei and John D Lafferty. Topic models. In Text Mining, pages 101–124. Chapman and Hall/CRC, 2009.

Léon Bottou. Online learning and stochastic approximations. On-line learning in neural networks, 17(9):142, 1998.

Léon Bottou. Stochastic gradient descent tricks. In Neural networks: Tricks of the trade, pages 421–436. Springer, 2012.

Leo Breiman. Bagging predictors. Machine Learning, 24(2):123–140, 1996.

Leo Breiman. Random forests. Machine Learning, 45(1):5–32, 2001a.

Leo Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). Statistical Science, 16(3):199–231, 2001b.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. Classification and Regression Trees. CRC press, 1984.

Emmanuel Candès and Terence Tao. The dantzig selector: Statistical estimation when p is much larger than n. The Annals of Statistics, pages 2313–2351, 2007.

Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. Foundations of Computational mathematics, 9(6):717, 2009.

Gary Chamberlain. Econometrics and decision theory. Journal of Econometrics, 95(2): 255–283, 2000.

Xiaohong Chen. Large sample sieve estimation of semi-nonparametric models. Handbook of econometrics, 6:5549–5632, 2007.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney K Newey, et al. Double machine learning for treatment and causal parameters. Technical report, Centre for Microdata Methods and Practice, Institute for Fiscal Studies, 2016a.

Victor Chernozhukov, Juan Carlos Escanciano, Hidehiko Ichimura, and Whitney K Newey. Locally robust semiparametric estimation. arXiv preprint arXiv:1608.00033, 2016b.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, and Whitney Newey. Double/debiased/neyman machine learning of treatment effects. American Economic Review, 107(5):261–65, 2017.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. The Econometrics Journal, 21(1):C1–C68, 2018a.

Victor Chernozhukov, Mert Demirer, Esther Duflo, and Ivan Fernandez-Val. Generic machine learning inference on heterogenous treatment effects in randomized experiments. Technical report, National Bureau of Economic Research, 2018b.

Victor Chernozhukov, Whitney Newey, and James Robins. Double/de-biased machine learning using regularized riesz representers. arXiv preprint arXiv:1802.08667, 2018c.

Hugh A Chipman, Edward I George, Robert E McCulloch, et al. Bart: Bayesian additive regression trees. The Annals of Applied Statistics, 4(1):266–298, 2010.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. Machine learning, 20(3): 273–297, 1995.

Thomas G Dietterich. Ensemble methods in machine learning. In International workshop on multiple classifier systems, pages 1–15. Springer, 2000.

M. Dimakopoulou, S. Athey, and G. Imbens. Estimation considerations in contextual bandits. arXiv, 2017.

Nikolay Doudchenko and Guido W Imbens. Balancing, regression, difference-in-differences and synthetic control methods: A synthesis. Technical report, National Bureau of Economic Research, 2016.

M. Dudik, J. Langford, and L. Li. Doubly robust policy evaluation and learning. International Conference on Machine Learning, 2011.

M. Dudik, D. Erhan, J. Langford, and L. Li. Doubly robust policy evaluation and optimization. Statistical Science, 2014.

Bradley Efron and Trevor Hastie. Computer Age Statistical Inference, volume 5. Cambridge University Press, 2016.

Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. The Annals of statistics, 32(2):407–499, 2004.

J. R. Firth. A synopsis of linguistic theory 1930-1955. In Studies in Linguistic Analysis (special volume of the Philological Society), volume 1952–1959, 1957.

Rina Friedberg, Julie Tibshirani, Susan Athey, and Stefan Wager. Local linear forests. arXiv preprint arXiv:1807.11408, 2018.

Jerome H Friedman. Stochastic gradient boosting. Computational Statistics & Data Analysis, 38(4):367–378, 2002.

Matthew Gentzkow, Bryan T Kelly, and Matt Taddy. Text as data. Technical report, National Bureau of Economic Research, 2017.

P. Gopalan, J. Hofman, and D. M. Blei. Scalable recommendation with hierarchical Poisson factorization. In Uncertainty in Artificial Intelligence, 2015.

Donald P Green and Holger L Kern. Modeling heterogeneous treatment effects in survey experiments with bayesian additive regression trees. Public opinion quarterly, 76(3):491–511, 2012.

Z. S. Harris. Distributional structure. Word, 10(2–3):146–162, 1954.

Jason Hartford, Greg Lewis, and Matt Taddy. Counterfactual Prediction with Deep Instrumental Variables Networks. 2016. URL `https://arxiv.org/pdf/1612.09596.pdf`.

John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1):100–108, 1979.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The Elements of Statistical Learning. New York: Springer, 2009.

Trevor Hastie, Robert Tibshirani, and Martin Wainwright. Statistical Learning with Sparsity: The Lasso and Generalizations. CRC Press, 2015.

Trevor Hastie, Robert Tibshirani, and Ryan J Tibshirani. Extended comparisons of best subset selection, forward stepwise selection, and the lasso. arXiv preprint arXiv:1707.08692, 2017.

Jennifer L Hill. Bayesian nonparametric modeling for causal inference. Journal of Computational and Graphical Statistics, 20(1):217–240, 2011.

Keisuke Hirano and Jack R Porter. Asymptotics for statistical treatment rules. Econometrica, 77(5):1683–1701, 2009.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1):55–67, 1970.

Paul W Holland. Statistics and causal inference. Journal of the American statistical Association, 81(396):945–960, 1986.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. Neural networks, 2(5):359–366, 1989.

Kosuke Imai, Marc Ratkovic, et al. Estimating treatment effect heterogeneity in randomized program evaluation. The Annals of Applied Statistics, 7(1):443–470, 2013.

Guido Imbens and Jeffrey Wooldridge. Recent developments in the econometrics of program evaluation. Journal of Economic Literature, 47(1):5–86, 2009.

Guido W Imbens and Donald B Rubin. Causal Inference in Statistics, Social, and Biomedical Sciences. Cambridge University Press, 2015.

Bruno Jacobs, Bas Donkers, and Dennis Fok. Product Recommendations Based on Latent Purchase Motivations. 2014. URL `http://www.ssrn.com/abstract=2443455`.

N. Jiang and L. Li. Doubly robust off-policy value evaluation for reinforcement learning. International Conference on Machine Learning, 2016.

N. Kallus. Balanced policy evaluation and learning. arXiv, 2017.

Michael P Keane et al. Panel data discrete choice models of consumer demand. Prepared for The Oxford Handbooks: Panel Data, 2013.

Toru Kitagawa and Aleksey Tetenov. Who should be treated? Empirical welfare maximization methods for treatment choice. Technical report, Centre for Microdata Methods and Practice, Institute for Fiscal Studies, 2015.

Steven W Knox. Machine learning: a concise introduction, volume 285. John Wiley & Sons, 2018.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

Sören Künzel, Jasjeet Sekhon, Peter Bickel, and Bin Yu. Meta-learners for estimating heterogeneous treatment effects using machine learning. arXiv preprint arXiv:1706.03461, 2017.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. Advances in applied mathematics, 6(1):4–22, 1985.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. nature, 521(7553):436, 2015.

O. Levy and Y. Goldberg. Neural word embedding as implicit matrix factorization. In Advances in Neural Information Processing Systems, 2014.

L. Li, W. Chu, J. Langford, T. Moon, and X. Wang. An unbiased offline evaluation of contextual bandit algorithms with generalized linear models. Journal of Machine Learning Research Workshop and Conference Proceedings, 2012.

L. Li, S. Chen, J. Kleban, and A. Gupta. Counterfactual estimation and optimization of click metrics for search engines. CoRR, 2014.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. Journal of machine learning research, 11(Aug): 2287–2322, 2010.

Nicolai Meinshausen. Relaxed lasso. Computational Statistics & Data Analysis, 52(1): 374–393, 2007.

T. Mikolov, K. Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space. International Conference on Learning Representations, 2013a.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems, 2013b.

T. Mikolov, W.-T. au Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2013c.

Alan Miller. Subset selection in regression. Chapman and Hall/CRC, 2002.

A. Mnih and G. E. Hinton. Three new graphical models for statistical language modelling. In International Conference on Machine Learning, 2007.

A. Mnih and K. Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In Advances in Neural Information Processing Systems, 2013.

A. Mnih and Y. W. Teh. A fast and simple algorithm for training neural probabilistic language models. In International Conference on Machine Learning, 2012.

Sendhil Mullainathan and Jann Spiess. Machine learning: an applied econometric approach. Journal of Economic Perspectives, 31(2):87–106, 2017.

J. Pennington, R. Socher, and C. D. Manning. GloVe: Global vectors for word representation. In Conference on Empirical Methods on Natural Language Processing, 2014.

James Robins and Andrea Rotnitzky. Semiparametric efficiency in multivariate regression models with missing data. Journal of the American Statistical Association, 90(1):122–129, 1995.

Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. Biometrika, 70(1):41–55, 1983.

Francisco JR Ruiz, Susan Athey, and David M Blei. Shopper: A probabilistic model of consumer choice with substitutes and complements. arXiv preprint arXiv:1711.03560, 2017.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. nature, 323(6088):533, 1986.

Robert E Schapire and Yoav Freund. Boosting: Foundations and algorithms. MIT press, 2012.

Bernhard Scholkopf and Alexander J Smola. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2001.

Steven L Scott. A modern bayesian look at the multi-armed bandit. Applied Stochastic Models in Business and Industry, 26(6):639–658, 2010.

V. Semenova, M. Goldman, V. Chernozhukov, and M. Taddy. Orthogonal ML for demand estimation: High dimensional causal inference in dynamic panels. arXiv:1712.09988, 2018.

A. Strehl, J. Langford, L. Li, and S. Kakade. Learning from logged implicit exploration data. Conference on Neural Information Processing Systems, 2010.

Richard S Sutton, Andrew G Barto, et al. Reinforcement learning: An introduction. MIT press, 1998.

A. Swaminathan and T. Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. Journal of Machine Learning Research, 2015.

P. Thomas and E. Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. International Conference on Machine Learning, 2016.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika, 25(3/4):285–294, 1933.

Robert Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological), pages 267–288, 1996.

Robert Tibshirani and Trevor Hastie. Local likelihood estimation. Journal of the American Statistical Association, 82(398):559–567, 1987.

Mark J van der Laan and Daniel Rubin. Targeted maximum likelihood learning. The International Journal of Biostatistics, 2(1), 2006.

Aad W Van der Vaart. Asymptotic Statistics. Cambridge University Press, 2000.

Vladimir Naumovich Vapnik. Statistical learning theory, volume 1. Wiley New York, 1998.

Hal R Varian. Big data: New tricks for econometrics. Journal of Economic Perspectives, 28 (2):3–28, 2014.

L. Vilnis and A. McCallum. Word representations via Gaussian embedding. In International Conference on Learning Representations, 2015.

Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. Journal of the American Statistical Association, (just-accepted), 2017.

M. Wan, D. Wang, M. Goldman, M. Taddy, J. Rao, J. Liu, D. Lymberopoulos, and J. McAuley. Modeling consumer preferences and price sensitivities from large-scale grocery shopping transaction logs. In International World Wide Web Conference, 2017.

Halbert White. Artificial neural networks: approximation and learning theory. Blackwell Publishers, Inc., 1992.

Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining. Knowledge and information systems, 14(1):1–37, 2008.

Achim Zeileis, Torsten Hothorn, and Kurt Hornik. Model-based recursive partitioning. Journal of Computational and Graphical Statistics, 17(2):492–514, 2008.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2):301–320, 2005.