Making the Most of Supervised Machine Learning in Strategy

Jason Rathje
U.S. Air Force and AF Ventures

Riitta Katila
Stanford University

Philipp Reineke
Stanford University

Draft version under construction

We are thankful for the help and comments from the seminar audiences at the Academy of Management Big Data Conference, Academy of Management Annual Meetings, the 2nd AI and Strategy Consortium, The Organizations and AI LinkedIn Group Talk series, Strategic Management Society Annual Meetings New Research Methods panel, and the Stanford Technology Ventures Program Research Seminar. Tom Byers, Chuck Eesley, Kathy Eisenhardt, Rahul Kapoor, Risto Miikkulainen, and Scott Stern provided invaluable advice. Our research project was supported by the Stanford Technology Ventures Program.

ABSTRACT

Recent advances in machine learning methods have opened up new empirical possibilities for inductive theorizing and for the analysis of unstructured text data in strategy. In contrast, the opportunities to use machine learning in deductive strategy research are less well understood. In this paper, we spotlight supervised machine learning and its application to many problems in strategy that inherently involve prediction. We use a simulation and an analysis of technology invention data to illustrate. A core contribution is to provide guidance to strategy researchers in the use of supervised machine learning.

Keywords:
AI and machine learning, Causal inference, Technological Change and Types of Innovation, Patents and R&D, Government Regulation and Public-Private Partnership

1| INTRODUCTION

Machine learning (ML) offers new empirical tools for strategy researchers, but how do these methods fit with what we know and need? Supervised machine learning – the focus of this paper – centers on producing *predictions* of an outcome variable *y* from *x*. Our contribution is to provide a primer on supervised machine learning methods that can be helpful in strategy problems that involve prediction (logistic regression, causal inference strategies in which the first stage is a prediction, etc.). The key advantage of using machine learning is that it can help reveal detailed patterns that underlie decisions, yet, at the same time, maintain generalizability of the conclusions drawn from the data.

There are two broad categories of strategy research that can benefit. The first category is complex data that includes a prediction task. Here, machine learning can help researchers uncover detailed, and potentially idiosyncratic data patterns, and generalize from them. Examples include predictions of tie formation, firm survival, or predictions from more recently available data, such as whether a satellite picture represents a geographical area with economic activity (Mullainathan and Spiess, 2017). Strategy scholars more and more frequently use these types of complex data.

The second, and related category is causal inference which inherently includes a prediction task. Common examples in strategy scholarship include estimating a causal effect of a treatment which often includes flexibly controlling for many confounders, including through matching methods[1] or instrumental variables regressions, where the first stage is effectively a prediction.[2] So while supervised machine learning is not designed to directly assist parameter estimates in

[1] Propensity score matching uses prediction and is particularly common in strategy research. "Conditioning on the propensity score typically is done by matching on the propensity score, subclassification into strata within which propensity scores are similar, regression adjustment on the propensity score, or weighting by the propensity score."
[2] Another example is post-hoc estimation of heterogenous treatment effects in subsamples (Choudhury et al., 2021).

second-stage regressions,[3] it has a central role in many empirical designs that are currently used in strategy where "first stage" involves a prediction task.[4] While we focus on causal inference applications in the current paper, general ideas apply to both tasks.

Other sister disciplines including economics and biosciences have started to apply machine learning for deductive research (D'Agostino, 1998; McCaffrey, Ridgeway, and Morral, 2004; Setoguchi et al., 2008; Cannas and Arpino, 2018; Goller et al., 2020). We first review how machine learning can help prediction, including the recently introduced machine learning propensity score method that provides a particularly suitable approach to capitalize on machine learning advantages in strategy, as Bettis and Blettner (2020) suggest. We revisit a classic technology strategy matching problem to compare the different approaches and to provide advice for practitioners. We also illustrate the use of machine learning-deduction for strategy using simulation.

2| PREDICTION TASKS IN STRATEGY

In many classic strategy decisions, including whether to invest in a particular project or not, prediction is an essential part of the decision. Similarly, evaluation of consequences of decisions such as which growth strategy (acquisition vs organic), policy (CSR vs not), or technology (renewable vs not) to pick, "treated" observations often differ in many covariates (e.g., age, industry, experience, location, etc.) from those in the counterfactual control group, which makes first-stage prediction a significant component of the policy evaluation. Without the first-stage prediction, evaluation of strategic decisions often suffers from treatment selection bias, potentially causing causal identification to fail and rendering the results from the study invalid. Given the complexity of these real world strategy decisions, covariates that underlie prediction are

[3] Even when machine learning algorithms produce regression coefficients, the estimates are rarely consistent. Further, the tools do not provide consistent standard errors. (Mullainathan and Spiess, 2017: 96).

[4] While other research compares OLS and regression trees (Mullainathan and Spiess, 2017) and OLS/logistic and random forests and regression trees (Choudhury et al 2021), we compare logistic regression and LASSO.

typically not only large in number, but are interdependent with each other, complicating the ability of strategy researchers to make the evaluation. To illustrate, Leiblein et al. (2018) note that in strategic decisions, "decision interdependencies, including higher order interactions… call [traditional] regression approaches into question."

A practical problem that strategy researchers encounter is deciding which variables to include in the prediction equation. "Typically, economic intuition will suggest a set of variables that might be important to control for but will not identify exactly which variables are important or the functional form with which variables should enter the model." (Belloni et al., 2017)"

In particular, to address the selection bias due to non-random samples, strategy researchers have embraced several empirical techniques that use prediction in the first stage. This is important, because accurate prediction in the first stage critically influences the accuracy of the second stage (determining the effectiveness of a treatment; Imbens, 2004). In this way, prediction becomes an inherent task in evaluating how effective a strategic decision (treatment) is. Given the complexity, Bettis and Blettner (2020) observe that "The nature of complexity … strongly suggests the expanded development and use of … appropriate machine learning algorithms." We use ML-propensity score matching in this paper to illustrate.

Example of prediction: Propensity scores. We chose propensity score matching as an example because our meta-analysis of Strategic Management Journal articles pointed to an increase in the use of the method, and identified over 50 papers using the propensity score method in the past few years. (We examined research published in the Strategic Management Journal since 2010 that uses propensity scores to address endogeneity concerns.) As detailed below, our focus was to identify the kinds of strategy problems where propensity scores have been used to enhance

causal inference, the variables and procedures used in the first-stage analyses to calculate the propensity score for the estimation, and the trends in research using propensity scores over time.

Propensity score matching helps us illustrate how machine learning can expand the strategy scholar's method toolbox. It allows the researcher to discover complex structure that was not specified in advance such as (high-level) interactions and thus suggests new theoretical prediction criteria (that need to be interpreted with scholar's insight). At the same time, it lowers concerns regarding over specification of the model to a particular circumstance[5] by using quantifiable criteria to prune. We point to two main ways in which strategy scholars could make the most of supervised machine learning method's advantages by identifying (1) a comprehensive set of relevant matching characteristics, and relationships of characteristics (including functional form, additivity) that contribute to treatment selection bias but not overfit, and (2) measures of how well a particular approach controls for treatment selection bias.

Because much research in strategy centers around the impacts of "treatments" or "policies," propensity scores is a common method. For example, a long tradition of strategy research focuses on growth strategy interventions such as alliances (Asgari, Singh, and Mitchell, 2017), joint ventures (Chang et al., 2013), corporate diversification (Rawley, 2010; Chang et al., 2016), and refocusing moves (de Figueiredo, Feldman and Rawley, 2019) using propensity scores to identify comparable control groups that were not exposed to the intervention. Other research that similarly employs propensity scores to enhance causal inference examines the performance impact of affiliations with high-status actors (Schuler et al., 2017), particular types of CEOs and top executives (Cummings and Knott, 2018; Patel and Cooper, 2014; Chang and Shim, 2015; Mata and Alves, 2018; Cho et al., 2016) or investors (Hasan et al., 2011; DesJardine and Durand, 2020;

[5] That is, overfitting which limits generalizability which can render second-stage estimates invalid.

Oehmichen et al. 2021). More recent work that has used propensity scores examines the implications of business strategy decisions such as adoption of sustainable business practices (Ortiz-de-Mandojana and Bansal, 2016; Durand and Stowoly, 2019). In all of these cases, maintaining a *comparable* control group (that was not treated) is a challenge.

In particular, the motivation to use propensity score methods is that in many strategy questions, the number of observable characteristics of observations (characteristics of firms, employees, etc.) that may bias the sample is relatively high (Vanneste and Gulati, 2020; Asgari, Singh, and Mitchell, 2017).[6] As Dehejia and Wahba (2002) note, with a small number of characteristics (for example, two binary variables), matching is straightforward as one would group units in four cells. In these cases, exact matching can be used.[7] However, when there are many variables, including continuous ones, there are often many cells with missing values and arbitrary cut-off points to create cells that increase the likelihood of bias (Dehejia and Wahba; 2002). In samples with multiple potential confounders, propensity score matching has advantages relative to other matching methods for several reasons: It remains stable even as the number of potential confounders increases, reduces researcher's subjective assessments about "similarity," and increases the number of treated firms that can be matched (and thus reduces the data that need to be thrown away).[8] While matching methods, including propensity score models cannot control for unobservables (and are often combined with other methods such as fixed-effects estimation or using eventually treated as controls in case of staggered treatments), they alleviate selection effects by reducing the observable differences between treated and controls. Altogether, propensity score methods become useful under such circumstances which are typical of strategy data (Leiblein et

[6] Also departures from linearity and non-additivity of observations are common.
[7] Thank you to anonymous reviewers for asking us to clarify these details.
[8] Rosenbaum and Rubin (1983) is a classic reference to the technique, and Guo and Fraser (2010) provide STATA code, and a review of the method for social sciences.

al., 2018; Bettis and Blettner, 2020; Rathje and Katila, 2020). Here, deciding which variables, and which functional forms to include in the first stage equation can critically influence the estimates.

Overall, propensity score methods have become used for a wide variety of strategy research topics where tracking a comparable sample of non-treated units is a frequent problem to solve, and the number of observable covariates is relatively large, and thus provides a useful application to illustrate the use of supervised machine learning.

Our review of the SMJ papers using propensity scores shows that few researchers explain their choice of first stage predictors… thus leaving vulnerable to bias.

3 PREDICTION CHALLENGES AND HOW MACHINE LEARNING CAN HELP

Prediction challenges. Why should strategy researchers look to add supervised machine learning in their toolkit? There are several reasons, including metrics to avoid under and overfitting, and to assess validity.

First, with complex real-world data, researcher intuition used to *pick confounders* is likely to reach its limits. Machine learning helps by pointing to which confounders matter. Machine learning also helps determine precise functional forms. As Mullainathan and Spiess (2017: 101) note, "including all pairwise interactions would be infeasible as it produces more regressors than data points (especially considering that some variables are categorical)." In contrast, supervised machine learning searches for these interactions automatically, and "allows us to let the data explicitly pick effective specifications, and thus allows us to recover more of the variation and construct stronger" instruments and predictions (Mullainathan and Spiess, 2017: 101).

Second, even if scholars have the foresight and patience to hand-curate and test a large number of variables and their interactions, the expansion of dimensionality may lead to *overfitting*.

Overfitting occurs when additional dimensions do not increase the ability for the propensity score model to predict treatment, but rather explain some noisy covariance. The resulting propensity scores will then deviate from the true propensity, which risks making the second stage estimations invalid. Simulations have shown that relying on overfit propensity scores results in inflated standard errors (i.e., lack of precision) (Schuster, Lowe, and Platt, 2016), over-estimated or under-estimated second-stage effects (Cepeda *et al.*, 2003), and even "paradoxical associations" (i.e. significance in the wrong direction) in the second stage (Concato, Feinstein, Peduzzi, Kemper, & Holford, 1996: 1373). In sum, overfitting can severely limit the interpretability of the second stage's estimates of treatment's performance effects, for example.

Third, traditional first-stage models have no objective measurement to assess the severity of the problem. Traditional propensity score models for instance rely on measurements of model fit (pseudo-R^2) to judge confidence. The higher the R^2 metric, the better the propensity score model fits the available data, and scholars have greater confidence that their model is functioning correctly. However, a higher R^2 is not related with predictive performance, only with how well the model fits the currently available data (UCLA, 2012). Therefore, in instances of overfitting, the model fit will either stay the same or increase, giving researchers false confidence.

Recent research in statistics, and our review of the recently-published papers in SMJ suggests that researchers often are limited in ways to objectively select which covariates and their interactions to include in the models involving prediction, leaving scholars unable to validate matching method's performance in controlling for treatment selection bias (see also Holland, 1986). As a result, matching methods often overconfidently rely on a limited set of "potentially" confounding covariates which does not necessarily control for treatment bias, and could even make the results invalid. As a result, some statisticians have called for the extreme measure to limit

research to only those empirical contexts which allow for randomized experiments (Grushka-Cockayne, Jose, and Lictendhal, 2016).

How supervised machine learning can help. To address the concerns, and to enable research in contexts in which randomized experiments are not possible, machine learning methods can possibly assist strategy researchers in prediction. Two core features of *supervised machine learning* - regularization, and cross-validation are particularly relevant.

Regularization allows researchers to quantitatively determine which covariates influence prediction (i.e., are "actually" confounding), by algorithmically removing non-confounding covariates from the regression. Regularization decreases subjectivity in the covariate selection process and prevents overfitting. While traditional methods require ample, subjective justification that some observable covariates are more important than others (e.g., see Stuart and Rubin, 2011 re matching), regularization allows machine learning to determine which covariates are important, minimizing subjectivity, and allowing researcher intuition to complement the data by providing mechanism explanations. Regularization adds penalty terms to the machine learning estimator to penalize complex models in favor of simpler models. The most common forms of regularization applicable for prediction are Lasso (L1), Ridge (L2) and Elastic Net.

C*ross-validation* allows scholars to validate the quality of the prediction by quantitatively testing how well the results generalize. The key idea is to train the prediction on a training set that is separate and distinct from the test set that is used to gauge its accuracy. The idea of separating training from testing is to evaluate how well the model will generalize to yet unseen data. As Varian JEP (2014) notes: "For many years, economists have reported in-sample goodness-of-fit measures using the excuse that we had small datasets. But now that larger datasets have become available, there is no reason not to use separate training and testing sets." In cross-validation,

machine learning again adds a quantitative performance evaluation metric as researcher intuition is augmented with a quantifiable approach.

Cross-validation uses random partitioning of data to training, validation, and test sets (Chernozhukov *et al.*, 2016). 60-20-20 splits between training, validation, and test datasets are common, but these numbers are relative and depend upon the size of the available data. The larger the dataset, the smaller percentages of data are required for validation and test sets. A reasonable general rule of thumb is ensuring that the test data set has at least as many observations as covariates. If there are less, then k-fold cross validation (k times with different splits) is used. [9]

Training data is used strictly for coefficient estimation; Validation data is used to optimize the regularization tuning parameters;[10] Test data is used to measure predictive performance.

- Training data is used strictly for coefficient (θ) estimation. Here the loss function, along with regularization terms, is applied.
- Validation data is used to support regularization to prevent overfitting, i.e. optimize the regularization tuning parameters, λ_1 & λ_2. These parameters are important. As they are tuned larger (i.e., approach one), the regularization terms will over penalize, potentially removing dimensions that are confounding and oversimplifying the model. As they are tuned smaller (i.e., approach zero), they will not remove any dimensions, and the model remains prone to overfitting. Here, cross-validation is used to support parameter tuning. First, a predictive model is generated with the regularization tuning parameters set to zero. Then, model performance is calculated using the validation data. Next, a new model is generated with slightly higher regularization tuning parameters, and the model performance is re-calculated. This occurs iteratively, until the tuning parameters oversimplify the model, weakening model performance. At that point, the tuning parameters are set at the values used to generate the optimum performing model. As a result, the sweet-spot between removing and keeping potentially confounding dimensions has been found.

[9] With smaller sample sizes, modified cross-validation techniques can be applied. For example, techniques such as k-fold cross validation maximize the ability to generate predictive models with small data by bootstrapping training set. For researchers with smaller data sets, this method provides a useful tool to increase model performance (Choudhury et al., 2019).

[10] Reference to a "validation dataset" disappears if the practitioner is choosing to tune model hyperparameters using k-fold cross-validation with the training dataset.

- Test data is used to measure predictive performance. While there are multiple measures of predictive performance (Webb, 2003), Area Under the Curve (AUC) metric is the most commonly used. AUC measures how well predictions can predict true positives, defined as the number of treatments that are correctly predicted while minimizing the prediction of false positives, defined as the number of treatments that are incorrectly predicted (Bradley, 1997). AUC is assessed by calculating out-of-sample performance (Russell and Norvig, 2010), that is, from assessing how well the model predicts the outcome Y_i given some novel input X_i. Therefore, as compared to alternative confidence measurements which measure model fit (such as R^2), AUC measures *predictive* performance. To generate an unbiased measurement of performance, a separate test data set is thus required. The test data set is used once, and *after* the final model has been calculated (i.e., after the training and validation steps). For comparison, traditional matching methods cannot assess performance since the entire data set is used to generate the model.

In the next section, we review how to apply these techniques in a machine learning-matching approach to patent data.

4| EMPIRICAL ANALYSIS

Following Choudhury et al.'s (2019) suggestion that new applications of machine learning in strategy research should start with an interesting research question and an empirical data source, we investigate the relationship between private-public sector collaborations and their performance outcomes – a question that has a long-standing tradition in the strategy field (Trajtenberg, 1997; Pahnke et al., 2015; Bruce, de Figueiredo, and Silverman, 2019; Rathje, 2019; Rathje and Katila, 2020).[11] We examine what machine learning can add to the strategist's toolkit.

For the empirical illustration of supervised machine learning, we compare the "treated" group of private organizations that developed technologies together with the public sector with the

[11] Prior research has for example studied the effect that different public funding mechanisms (e.g., grants, contracts, cooperative agreements) have on technological innovation (Bruce, de Figueiredo, and Silverman, 2019; Jia, Huang, and Zhang, 2018; Rathje, 2019). As it is nearly impossible to run policy "experiments" that randomly treat technologies with public funding, yet public funding is significant for innovation efforts, the setting serves as a particularly appropriate empirical context to illustrate the use of supervised machine learning for matching.

"control" group of organizations which developed without. Empirically, we have a population of patented technologies over a 31-year period from which we can determine which technologies were "treated" with public partners, predict which control observations to include for each treated, and calculate the outcome, i.e., the extent to which a technology output is successful.

We chose the empirical setting because it is a particularly useful context to illustrate when machine learning can provide additional insight to strategy research. Determinants of collaboration decisions are potentially complex and not fully understood in this setting. It is also unclear how prior findings from specific samples generalize to a population of collaborations. Sample size is also reflective of larger samples now analyzed by strategy scholars. [12]

Traditional Approaches

To test the treatment effect without machine learning, we typically use three steps. The first step is to use prior literature, subjective reasoning, and our intuition as researchers to *identify the potentially confounding covariates* that would limit the ability to interpret treatment effects. In the case of public-private R&D collaborations, the norm is to use calendar time and technology, reported in Table 1 (Agrawal, Cockburn, and Rosell, 2009; Belenzon and Schankerman, 2013; Trajtenberg, Henderson, and Jaffe, 1997). In practice, patented technologies are typically matched on age (calendar years that proxy for factors such as variations in macroeconomic and technical climate at the time of patent filing and granting), and patent technology class (e.g., electronics, computers, or manufacturing with different underlying technical requirements and ecosystems).[13]

[12] We analyze the full population of U.S. patents granted between 1982-2012, which includes 3,337,229 patented technologies and 26,174 observed characteristics of the patented technology, many of which are likely to be confounding. Of the patented technologies, 58,082 were publicly funded (treated), and we use supervised machine learning for this first-stage prediction task for matching. As noted above, propensity score matching uses prediction to match observations based on their propensity to be treated (Caliendo and Kopeinig, 2008; Rosenbaum and Rubin, 1983).

[13] Prior research identified technology class, application year, and grant year as important (Pavitt, 1982) because it is reasonable to believe that if public-collaboration patents come from a different population of technology classes than corporate patents, the success of public-collaboration patents may in fact be derived from differences in the

A significant question is whether these 2-3 covariates is enough. For example, will it be enough to take into account broad technology (e.g. information technology) but not technology specialization (e.g. AI)? Using traditional methods, we take it granted that the assumption to use the traditional covariates is "correct" and proceed with the analysis.

Possible Shortfalls of Traditional Approaches. In addition to the standard covariates noted above, another covariate that may be relevant is technology specialization measured by patent examiners. In pioneering work, Alcácer and Gittelman (2006) found that patent examiners are strongly associated with both (1) the specific scientific or technical specialization of the underlying technology, and (2) the number of forward citations a patent receives (i.e. a measure of success). It is reasonable to believe then, that patent examiners, as a significant proxy for the underlying technology specialization, could be correlated with both the treatment (public sector collaboration) and the outcome (such as patent's quality measured by future citations). In combination, patent examiners represent one missing but potentially significant confounding covariate.

Why is it, then, that despite our theoretical understanding of their importance, patent examiners are not included in current matching criteria? A key reason is that current methods do not provide an objective way to determine which confounders to include when the number of confounders increases beyond what can be included in the models. For instance, there are 25,442 unique patent examiners in the patent population that we study (1982-2012), so it would be practically impossible to match exactly on each examiner, or to include additional functional forms. As the number of covariates increases, dimensionality grows dramatically. Even if each

technology class (i.e., some technology classes receive more citations than others). Similarly, if public-sector funding for R&D in the United States is more common during a Democratic administration than a Republican administration, then public-sector patent performance may be influenced by application or grant year (e.g. when public organizations have more additional funding to spend on research or patenting in general).

covariate is binary, thus containing only two discrete dimensions, as the number of covariates grows by P, the number of dimensional combinations grows by 2^P (Rosenbaum and Rubin, 1983). Very quickly, this value grows more massive than the total number of observations. Thus, the increase in observations severely limits the effectiveness of traditional techniques to capture the additional confounders.

In particular, the limitation is that it is often not possible to know which confounders to add. What scholars with traditional methods cannot provide is a numerical evaluation of this step. As we note in more detail below, supervised machine learning offers a quantifiable step in this direction.

Steps of Including Supervised Machine Learning

How to Apply Supervised Machine Learning. We start by including the standard covariate set of technology class, application year, and grant year. Additionally, we then include (for later evaluation) additional covariates which have not been used in matching before, but are also likely to be confounding covariates per prior theory and research in the area, including number of patent inventors, geographic location (state, country, city), originality (i.e., how novel the patent is), and patent examiners (Alcácer and Gittelman, 2006). In total, this results in 27,212 potentially confounding dimensions. Note that exact matching would not be able to handle this task.

(1) Split the data. Machine learning approach starts by splitting the data into subsets – training, validation, and test. As noted above, this step allows us to quantitatively test how well the results generalize. While there are no exact rules of how to split the data, it is preferable to increase training efficiency by maximizing the size of the training set when possible (Jain and Chandrasekaran, 1982; Raudys and Jain, 1991). Given the large sample size of patent data, we

need a relatively small share of observations from which to test our predictions, and therefore chose a 98-1-1 data split (Picard and Berk, 1990; Reitermanová, 2010). This means that 98% of the data was used for the training set, while one percent was used for validation and test sets, respectively.[14]

(2) Select a machine learning model, add regularization, and cross-validate. Regularization and cross-validation are employed in tandem in the next phase. First, we select a supervised learning method for our predictive model, H_θ. Because we use propensity score matching (PSM), we use a *logistic regression model*.[15] Logistic regression is useful because it generates a continuous probability of an observation being treated by regressing the observation's set of covariates (X_i) on its observed treatment (Y_i). As a result, logistic regression model (as opposed to other machine learning techniques, such as a stochastic vector machine (SVM)) allows us to generate continuous estimates of propensity scores.

In this step, we include the regularization terms L1, $\lambda_1 \sum_{i=1}^{n} |\theta_i|$, and L2, $\lambda_2 \sum_{i=1}^{n} \theta_i^2$. These terms remove unconfounding covariates by penalizing their coefficients, θ, if the logistic model begins to overfit. Thus, our likelihood estimation function is: $L \equiv \sum_{i=1}^{n} \big(Y_i - H_\theta(X_i)\big)^2 + \lambda_1 \sum_{i=1}^{n} |\theta_i| + \lambda_2 \sum_{i=1}^{n} \theta_i^2$ where $H_\theta(X_i) = \frac{1}{1+ e^{-(\theta_0 + \theta_1 X_i + \cdots + \theta_N X_N)}}$. With this function, we estimate the covariate coefficients (θ). First, λ_1 & λ_2 are initialized to zero, and the training data is used to generate predictions of θ. Next, the validation set is used to compare model performance. Iteratively, the optimizer first searches the set of θ that minimizes the training error (regularization), and then for the set λ_1 & λ_2 that minimize the validation error (cross-validation)

[14] We conducted preliminary experiments with 80-10-10 or 90-5-5 and found consistent results.

[15] A few other regression models have been suggested and discussed in the statistical literature, regression trees in particular. See Lee et al. (2010) for an interesting comparative analysis of alternative models.

(Zou & Hastie, 2005: 310). Once $\theta, \lambda_1, \& \ \lambda_2$ are determined, supervised learning is complete. In practice, there are a wide variety of available tools and statistical learning packages that make supervised learning relatively straight forward – Scikit-Learn (for python), Caret (for R), and Tensorflow being among the most popular. We used Vowpal Wabbit,[16] which takes advantage of AdaGRAD to minimize the loss function and tune the regularization parameters with built-in cross-validation.

With the final model (i.e., after regularization and cross-validation is complete), we investigate the coefficient weights (sizes of θ) to determine which covariates contributed to treatment selection bias. Those covariates with the largest coefficient weights in magnitude contribute the most to treatment selection bias, while those with smaller weights have less of an effect. As a result, higher weights indicate the need to include the corresponding covariates in the matching model.

In practice, to evaluate the needed covariates, it is useful to plot a covariate-weight graph as a visual tool to investigate coefficient weights. This graph is a bar chart representing the range of dimension weights, grouped by covariates (e.g., in Figure 2 each examiner name is a dimension, examiners are a covariate). As an example, Figure 2 represents the covariate-weight graph for the top five most heavily weighted covariates in our data. Interestingly, patent examiners are the most heavily weighted covariate, followed by technology class and then country. These top three covariates indicate a need to include a *significantly different set of confounding covariates* in estimating treatment effects than the traditional covariate set that has been used. Indeed, 2 of the top 3 covariates are different from the traditional set, and only technology class makes it to the top set.

[16] Vowpal Wabbit is a fast, online learning code by Microsoft Research Group and (previously) Yahoo! Research. https://github.com/JohnLangford/vowpal_wabbit/wiki.

INSERT FIGURE 2 ABOUT HERE

To dive deeper into the examiner covariate and to investigate specific examiners, table 1 represents the coefficient weights of the top three highly weighted patent examiners (both positive and negative) along with three patent examiners who were relatively unimportant and driven to zero by regularization. Weights correspond to the probability of treatment. As an example, patent examiners Zarfas, Ansher, and Douglas have large and negative weights, indicating that they are more heavily assigned to examine technologies in the control group. Patent examiners Goddard, Grarsay, and Ryam have large and positive weights, indicating that they are more heavily assigned to technologies in the treated group. In contrast, examiners Knife, Renner, and Chan likely examine both equally, and so do not help to predict propensity scores. Therefore, regularization drove their weights to zero.

INSERT TABLE 1 ABOUT HERE

(3) Repeat by adding interactions. Next, the goal is to test the model by iteratively adding more covariates. As is typical in machine learning, we repeat the previous steps by first adding all quadratic interactions, and then adding all quadratic and tertiary interactions as potentially confounding covariates. For example, we include year x technology class, year x examiner name, and year x technology class x examiner name, because it is reasonable to assume that in certain years, particular technology classes were more likely to be "treated" with public funding. To protect against overfitting, we evaluate the performance of each model independently.

(4) Evaluate and select the "best" ML-PSM model. Once all the models are calculated, the next step is to identify the one with the best performance. When evaluating performance, it is important to measure both predictive performance using AUC and the model fit using R^2. As noted above, because AUC is the measurement of predictive, it should be the principal

measurement for model selection. R^2, additionally, allows us to interpret the impact of increasing model complexity on fit.[17] As model complexity grows (e.g., adding all interactions), if AUC begins to decrease while R^2 increases, the model is overfit. Selecting the model that maximizes AUC while preventing overfitting is the preferred choice.

To illustrate, we evaluate six different models by comparing changes in AUC (predictive performance) and R^2 (fit): (1) the baseline using propensity score matching with three traditional confounders i.e. technology class, application year, and grant year, (2) machine learning-propensity score matching method using three traditional confounders, (3) machine learning-propensity score matching with the addition of four other potential confounders suggested by prior work (patent inventors, geographic location, originality, patent examiners), (4) machine learning-propensity score matching adding quadratic interactions, but removing regularization, (5) repeating step #4 with regularization, and (6) adding cubic interactions.

INSERT TABLE 2 ABOUT HERE

In table 2, the baseline (model 1) has the lowest predictive performance and fit. For reference, an AUC of 0.514 is little better than a coin flip. Adding machine learning in model 2 increases the model performance by 49%. Adding 4 additional covariates in model 3 increases the predictive performance of the model by an additional 17%, and adding quadratic interactions, but removing regularization in model 4 increases performance by 5%. Adding regularization back in, in model 5 provides some interesting results. Unsurprisingly, by simplifying the model through regularization, model performance using the training data decreases. However, simplifying through regularization *increases* the predictive performance (AUC) in the test data. These differences are absolute and provide evidence that regularization and cross-validation are essential

[17] Since logistic regression models cannot generate accurate R^2, a pseudo-R^2 is applied. Adjusted count[17] is a simple pseudo-R^2 commonly used in logistic regressions, and the one used in our approach (UCLA, 2012).

to protect *against overfitting* and increasing generalizability. Finally, the addition of cubic interactions in model 6 provides strong evidence of overfitting. Overfitting is particularly apparent in the test data, where R^2 increases while AUC decreases. ~~Such a shift is a clear example of limits of elastic net regularization. While elastic net regularization is a useful tool in instances when D > N, it cannot act as a cure-all and will ultimately fail when we increase the number of dimensions such that D >> N.~~ Importantly, machine learning provides a clear metric to indicate that the optimum model is model 5, i.e., the regularized ML-PSM model including quadratic interactions.

(5) Match on propensity scores and plot the results. Once we select the preferred model in step 4, we can evaluate how well the ML-PSM method generated a balanced sample. Using the standard steps of propensity score matching, we first generate propensity scores for each observation (i.e. predicted likelihood of public sector collaboration). Next, we match treatment and control propensity scores one-to-one using a global-optimum matching algorithm. There are a wide variety of matching algorithms, but in cases where the control group is much larger than the treatment group, a global optimum strategy is preferred (Stuart and Rubin, 2008). Lastly, we remove the unmatched sample, leaving us with balanced treatment and control groups.

To evaluate balance, we revisit the balance plots from step one using the matched sample. First, balanced support in propensity score is assessed visually (Figure 3), that is, whether treatment and control groups are balanced across a composite of all covariates (Dehejia and Wahba, 2002). Second, balanced support between the covariates is assessed (Figure 1, right-hand side). Ideally, the distributions should be identical across the treatment and control groups, as shown in Figure 3.

INSERT FIGURE 3 ABOUT HERE

After validating that treatment and control groups are balanced across confounding covariates, we finish executing the ML-PSM method by assessing the treatment effects. Although

assessing the treatment effects is not at the core of our analysis, it is noteworthy that our additional analysis indicates that adding machine learning can also qualitative change the final results of the treatment evaluation: For instance, in splitting the public-sector collaborations in subcategories, the influence of grants on patent success switched from negative (baseline model 1 above) to significantly positive (model 5) (Additional details available from the authors; brief overview in Figure 4), providing further face validity to the importance of considering more advance methods for strategy analysis. As an overview, the list of steps for applying machine learning - propensity score matching is provided in Table 3.

INSERT TABLE 3 ABOUT HERE

5| SIMULATION ANALYSIS

Finally, we conducted simulation analyses to compare different analytical techniques because in simulation we know what the results "should be." We conducted two simulations, both of which draw on repeated sampling from the real U.S. patent data from USPTO. Using simulation based on a large-real world dataset ensures that the underlying data structure (i.e. autocorrelation of covariates, number of factor levels in covariates, correlation between covariates and independent and dependent variables, etc.) reflects a structure that can be found in real life and is not biased by the researchers' choices.

First, we validated the suitability of different techniques for predicting propensity scores that correctly indicate selection into treatment. For this simulation, we drew samples of size 4,000, 32,000, and 256,000 from our patent data, fit models to them, and computed AUC scores of the predictions made by this model vs true treatment status in a validation sample of 20% of the observations.

Second, we used Goller et al.'s (2020) Empirical Monte Carlo (EMC) simulation technique. This method utilizes large datasets that represent the full population under investigation. Our dataset of patent data provides a full account of the patent population between the years 1982 and 2012 and therefore allows us to apply Goller et al.'s (2020) EMC method. Following this method, first a true propensity score is estimated in the full population. Then all treated observations are removed from the sample and a placebo treatment is simulated. Lastly, a sample from the observations is drawn. In this sample we conduct propensity score matching and calculate the average treatment effect using the simulated treatments. Treatment effect should be zero since any remaining treatments are placebos, allowing us to examine the relative performance of the different estimators. The EMC simulation also allows comparing 2nd stage estimation results to those achievable with conventional methods such as Coarsened Exact Matching (CEM). For this analysis we report results on a sample of 256,000 observations.

In both simulations we fit a baseline closed-form logistic regression model that includes technology class, application year, and grant year as confounding variables. We also fit a version of this model that contains quadratic interactions of these variables[18]. We do not include the factorial confounders patent class, patent examiners, and patent examiner location because this would yield a larger number of confounders than there are observations and thereby cause an unregularized closed-form logistic regression model to be over-specified. We fit five machine learning models. One model has the same specification as the logistic regression model. Additional models include combinations of the additional confounders number of patent

[18] For this model we cannot report results of the model run on the full dataset. Running this model on the full data creates such a large regression matrix that the model cannot be computed within the memory and computation time limits of even the large high performance computing cluster used by the authors.

inventors, patent geographic location, patent originality, and patent examiners, λ_1 & λ_2 regularization terms[19], quadratic interactions, and cubic interactions. In the EMC simulation, we also include CEM on a model with all covariates.

INSERT TABLE 2 ABOUT HERE

Simulation results highlight the connection between sample size and flexibility of the regression function and the benefits of using PS-ML methods. Even at small sample sizes of 4,000 observations, a ML model using the full set of confounders yields the best predictive outcomes. However, adding interactions still causes models to overfit – yielding greater predictive success in the training data but not in the test sample. This pattern replicates for sample sizes of 32,00 and 256,000 observations in our data. Finally, at very large sample sizes (using the full data) where overfitting becomes less likely, adding interactions while regularizing further improves predictive power without overfitting. Including λ_1 & λ_2 terms has a positive impact throughout but adding interactions only yields improvements in predictive performance at very large sample sizes.

The EMC simulation results broadly confirm these findings and show that 1st stage propensity score AUC performance carries over to 2nd stage coefficient values[20]. Additionally,

[19] Stochastic gradient descent machine learning models often contain a third hyperparameter, namely the "number of runs" parameter. This parameter specifies how many times the available training dataset is supplied to the machine learning algorithm to iteratively train model parameters. If we have very many observations, as in the full dataset, one run over the data is sufficient and may already lead to over-fitting. However, in a smaller sample, a single run over the data may yield a model that under-fits because the observations were not sufficient for the model to complete the iterative gradient descent process. As such, training the "number of runs" parameter via cross-validation in smaller samples already applies a form of regularization - even without including λ_1 & λ_2 parameters. In the simulation we use the vw-hyperopt library to tune the "number of runs" parameter (jointly with the λ_1 & λ_2 parameters in regularized models) in 200 iterations per simulation run using the Tree-Structured Parzen Estimators (TPE) optimization algorithm.

[20] There are two approaches to evaluating 2nd stage performance propensity score models. One is based on the philosophy that a propensity score that accurately represents assignment to treatment yields unbiased 2nd stage results. A good fit (as shown by high AUC scores) of propensity score and true treatment assignment indicates an good propensity score model by this approach. A second approach is finding a propensity score models that calculates a propensity score that achieves covariate balance - regardless of whether that propensity score mimics the true treatment assignment mechanism (Ho et al., 2007). Both approaches have been found valid, but the latter

the EMC simulation permit performance comparisons of propensity score matching methods with the performance of other matching approaches such as CEM. Our results indicate that in our dataset CEM matching, even when matching on the full set of coefficients, only performs similarly to non-ML propensity score methods using a limited set of confounders. CEM fails to achieve the results of ML-PSM methods run on a larger set of confounders.

In summary our simulations show that in our patent dataset machine learning models that leverage the full set of available covariates succeed at producing propensity score models whose results are more generalizable and precise than those produced by closed form logistic regression - already at samples of size 4,000.

6| DISCUSSION

This paper offers a novel approach to deductive research. We add supervised machine learning to prediction problems in strategy such as logistic regression, and two-stage models such as matching or instrumental variables where the first stage is a prediction. Our review of the traditional matching methods presented compelling evidence that approaches are often incomplete and their evaluation is subjective. Our review of the supervised machine learning methods introduced two key features – regularization, and cross-validation – that can overcome many of these limitations. Regularization quantitatively and algorithmically removes non-confounding covariates, and cross-validation provides a method for measuring matching method performance, thus enabling more robust prediction and causal inference for strategy research.

There are several contributions. We provide a detailed methodology for applying machine learning to propensity score matching and utilized a standard data set – patent data – to explore

approach has produced superior results in some studies (Griffin et al., 2017). However, in our context where some predictors are represented in the models as a sparse matrix of binary indicator variables, with "1" values only occurring once for several of these indicators in subsamples, balance statistics over all covariates may be less meaningful. Thus, we follow the first approach.

how and when machine learning matching should be used. We also show that incrementally adding potentially confounding covariates and assessing treatment effects is a useful way to investigate the impact of machine learning. Rather than throwing in the "kitchen sink" to increase prediction, we show that a sequential, data-driven approach to adding confounding covariates and their complex functional forms increases not only model prediction in the first stage, but also de-biases estimates in the second stage. As a result, scholars have a new tool to explore observational settings.

In conclusion, we argue for a stronger application of machine learning in deductive research. Applying machine learning for causal inference is just starting to surface as an applied economic methodology (Athey *et al.*, 2018; Mullainathan and Spiess, 2017). However, the increasingly sophisticated approaches being developed in econometrics are not necessary for the strategy and organizations field to start applying these concepts. Integrating supervised machine learning concepts into existing practices can help deductive researchers overcome many limitations faced today.

REFERENCES

Agrawal AK, Cockburn IM, Rosell C. 2009. Not Invented Here? Innovation in Company Towns. *NBER Working paper* **15347**.

Alcácer J, Gittelman M. 2006. Patent Citations as a Measure of Knowledge Flows: The Influence of Examiner Citations. *Review of Economics and Statistics* **88**(4): 774–779.

Antonakis J, Brendahan S, Jacquart P, Lalive R. 2010. On making causal claims: A review and recommendations. *The Leadership Quarterly* : 1–93.

Athey S, Imbens GW, Wager S. 2018. *Approximate Residual Balancing: De-Biased Inference of Average Treatment Effects in High Dimensions*.

Ban G, El Karoui N, Lim AEB. 2016. Machine Learning and Portfolio Optimization. *Management Science* **64**(3): 1136–1154.

Belenzon S, Schankerman M. 2013. Spreading the Word: Geography, Policy, and Knowledge Spillovers. *Review of Economics and Statistics* **95**(3): 884–903.

BELLONI, CHERNOZHUKOV, FERNÁNDEZ-VAL, AND HANSEN, 2017.

Bettis RA. 2012. The search for asterisks: Compromised statistical tests and flawed theories. *Strategic Management Journal* **33**(1): 108–113.

Bettis R and Blettner, D. 2020. Strategic reality. Strategic Management Review.

Boyd S, Vandenberghe L. 2004. *Convex Optimization*. *Communication Networking*. Cambridge University Press: Cambridge, UK.

Bradley A. 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition* **30**(7): 1145–1159.

Bruce JR, de Figueiredo JM, Silverman BS. 2019. Public contracting for private innovation: Government capabilities, decision rights, and performance outcomes. *Strategic Management Journal* **40**(4): 533–555.

Caliendo M, Kopeinig S. 2008. Some practical guidance for the implementation of propensity score matching. *Journal of Econometric Surveys* **22**(1): 31–72.

Camerer CF, Nave G, Smith A. 2019. Dynamic Unstructured Bargaining with Private Information: Theory, Experiment, and Outcome Prediction via Machine Learning. *Management Science* **65**(4): 1867–1890.

Cepeda MS, Boston R, Farrar JT, Strom BL. 2003. Comparison of logistic regression versus propensity score when the number of events is low and there are multiple confounders. *American journal of epidemiology* **158**(3): 280–7.

Chernozhukov V *et al.* 2016. *Double/Debiased Machine Learning for Treatment and Causal Parameters*.

Choudhury P, Allen R, Endres M. 2018. Developing Theory Using Machine Learning Methods. *Harvard Business School Working Paper 19-032*.

Cochran W. 1968. The Effectiveness of Adjustment by Subclassification in Removing Bias in Observational Studies. *International Biometric Society* **24**(2): 295–313.

Concato J, Feinstein AR, Peduzzi P, Kemper E, Holford TR. 1996. A simulation study of the number of events per variable in logistic regression analysis. *Journal of Clinical Epidemiology* **49**(12): 1373–1379.

Congress. 1977. *Federal Grant and Cooperative Agreement Act*.

David PA, Hall BH, Toole AA. 2000. Is Public R&D a Complement or Substitute for Private R&D? A Review of the Econometric Evidence. *Research Policy* **29**(4): 497–529.

de Figueiredo R, Feldman E, Rawley E. 2019. The costs of refocusing: Evidence from hedge

fund closures during the financial crisis. *Strategic Management Journal* **40**(8): 1268–1290.
Dehejia RH, Wahba S. 2002. Propensity Score-Matching Methods for Nonexperimental Causal Studies. *Review of Economics and Statistics* **84**(1): 151–161.
Duchi J, Hazan E, Singer Y. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research* **12**: 2121–2159.
Energy USD of. 2019. U.S. Department of Energy, Phase we Topics.
Funk RJ, Owen-Smith J. 2017. A Dynamic Network Measure of Technological Change. *Management Science* **63**(3): 791–817.
Grushka-Cockayne Y, Jose VRR, Lictendhal KCJ. 2016. Ensembles of Overfit and Overconfident Forecasts. *Management Science* **63**(4): 1110–1130.
Hall BH, Jaffe A, Trajtenberg M. 2005. Market Value and Patent Citations. *RAND Journal of Economics* **36**(1): 16–38.
Ham RM, Mowery DC. 1998. Improving the effectiveness of public–private R&D collaboration: case studies at a US weapons laboratory. *Research policy* **26**(6): 661–675.
Hamilton BH, Nickerson JA. 2003. Correcting for Endogeneity in Strategic Management Research. *Strategic Organization* **1**(1): 51–78.
Handley MA, Lyles C, McCulloch C, Cattamanchi A. 2014. *Selecting and Improving Quasi-Experimental Designs in Effectiveness and Implementation Research*. *Annual Review of Public Health Selecting*.
Hansen PC, O'Leary DP. 2005. The Use of the L-Curve in the Regularization of Discrete Ill-Posed Problems. *SIAM Journal on Scientific Computing* **14**(6): 1487–1503.
Ho DE, Imai K, King G, Stuart EA. 2007. Matching as nonparametric preprocessing for reducing model dependence in parametric causal inference. *Political Analysis* **15**(3): 199–236.
Holland PW. 1986. Statistics and Causal Inference. *Journal of the American Statistical Association* **81**(396): 967.
Howell ST. 2017. Financing Innovation: Evidence from R & D Grants. *American Economic Review* **107**(4): 1136–1164.
Hsu DH. 2006. Venture Capitalists and Cooperative Start-up Commercialization Strategy. *Management Science* **52**(2): 204–219.
Iacus SM, King G, Porro G. 2012. Causal Inference without Balance Checking: Coarsened Exact Matching. *Political Analysis* **20**(1): 1–24.
Imbens GW. 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and statistics* **86**(1): 4–29.
Jaffe A, Trajtenberg M, Henderson R. 1993. Geographic Localization of Knowledge Spillovers as Evidenced by Patent Citations. *The Quarterly Journal of Economics* **108**(3): 577–598.
Jain A, Chandrasekaran B. 1982. Dimensionality and sample size consideration in pattern recognition practice. *Classification, Pattern Recognition and Reduction of Dimensionality. Handbook of Statistics* **2**: 835–856.
Jia N, Huang KG-L, Zhang CM. 2018. Public Governance, Corporate Governance, and Firm Innovation: An Examination of State-Owned Enterprises. *Academy of Management Journal* **62**(1): 1–28.
Kim Y, Steiner P. 2016. Quasi-Experimental Designs for Causal Inference. *Educational Psychology* **51**(4): 395–405.
LABOR U. S. DO. 2010. Veterans ' Employment & Training Service Annual Report to Congress.
Laursen K, Salter AJ. 2014. The paradox of openness: Appropriability, external search and collaboration. *Research Policy*. Elsevier B.V. **43**(5): 867–878.

Li G-C *et al.* 2014. Disambiguation and co-authorship networks of the US Patent Inventor Database. *Research Policy* **2138**: 1–38.
Menon A, Choi J, Tabakovi H. 2018. What You Say Your Strategy Is and Why It Matters: Natural Language Processing of Unstructured Text. *Academy of Management Proceedings* **1**.
Mullainathan S, Spiess J. 2017. Machine Learning: An Applied Econometric Approach. *Journal of Economic Perspectives* **31**(2): 87–106.
Office of Acquisition and Property Management. 2011. Procurement Contracts, Grant and Cooperative Agree. In *Grant Administration*: 1–5.
Pavitt K. 1982. R&D, patenting and innovative activities. A statistical exploration. *Research Policy* **11**(1): 33–51.
Picard RR, Berk KN. 1990. Data Splitting. *The American Statistician* **44**(2): 140–147.
Rathje J 2019. Hybrid conflict and innovation. PhD dissertation. Stanford University.
Rathje J, Katila R. 2020. Enabling technologies and the role of private firms: A machine learning matching analysis. *Strategy Science*, in press.
Raudys S, Jain A. 1991. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on pattern analysis*.
Reitermanová Z. 2010. Data Splitting. *Week of Doctoral Students 2010 -- Proceedings of Contributed Papers* : 31–36.
Robins JM, Rotnitzky A. 2001. Comment on the Bickel and Kwon article, "Inference for semiparametric models: Some questions and an answer". *Statistica Sinica* **11**(4): 920–936.
Rosenbaum PR, Rubin DB. 1983. The Central Role of the Propensity Score in Observational Studies for Causal Effects. *Source: Biometrika Biometrika* **70**(1): 41–55.
Rubin DB. 1973. The Use of Matched Sampling and Regression Adjustment to Remove Bias in Observational Studies. *Biometrics* **29**(1): 185–203.
Russell S, Norvig P. n.d. *Artificial Intelligence A Modern Approach Third Edition*.
Russell S, Norvig P. 2010. *Artificial Intelligence A Modern Approach Third Edition*. *Pearson*.
Schuster T, Lowe WK, Platt RW. 2016. Propensity score model overfitting led to inflated variance of estimated odds ratios. *Journal of Clinical Epidemiology* **80**: 97–106.
Shadish WR, Cook TD, Campbell DT. 2002. Experimental and Quasi-Experimental Designs for Generalized Causal Inference. Wadsworth Cengage Learning.
Stuart EA, Rubin DB. 2008. Best practices in quasi-experimental designs: Matching methods for causal inference. *Best Practices in Quantitative Social Science*.
Thompson P, Fox-Kean M. 2005. American Economic Association Patent Citations and the Geography of Knowledge Spillovers : A Reassessment Author (s): Peter Thompson and Melanie Fox-Kean Source : The American Economic Review , Vol . 95 , No . 1 (Mar ., 2005), pp . 450-460 Published b. *American Economic Review* **95**(1): 450–460.
Trajtenberg M, Henderson R, Jaffe A. 1997. University Versus Corporate Patents: A Window On The Basicness Of Invention. *Economics of Innovation and New Technology* **5**(1): 19–50.
UCLA. 2012. FAQ : What are pseudo R-squareds ? *Institute for Digital Research and Education*. Available at: https://stats.idre.ucla.edu/other/mult-pkg/faq/general/faq-what-are-pseudo-r-squareds/.
Wallsten SJ. 2000. The Effects of Government-Industry R&D Programs on Private R&D: The Case of the Small Business Innovation Research Program. *The RAND Journal of Economics,* **31**(1): 82–100.
Webb AR. 2003. *Statistical pattern recognition*. John Wiley & Sons: New York.
Wilson J. 2014. *Essentials of business research: A guide to doing your research project*. Sage.

Zou H, Hastie T. 2005. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society* **67**(2): 301–320.

FIGURE 1. BALANCE PLOTS, PRE-MATCH VS POST-MATCH

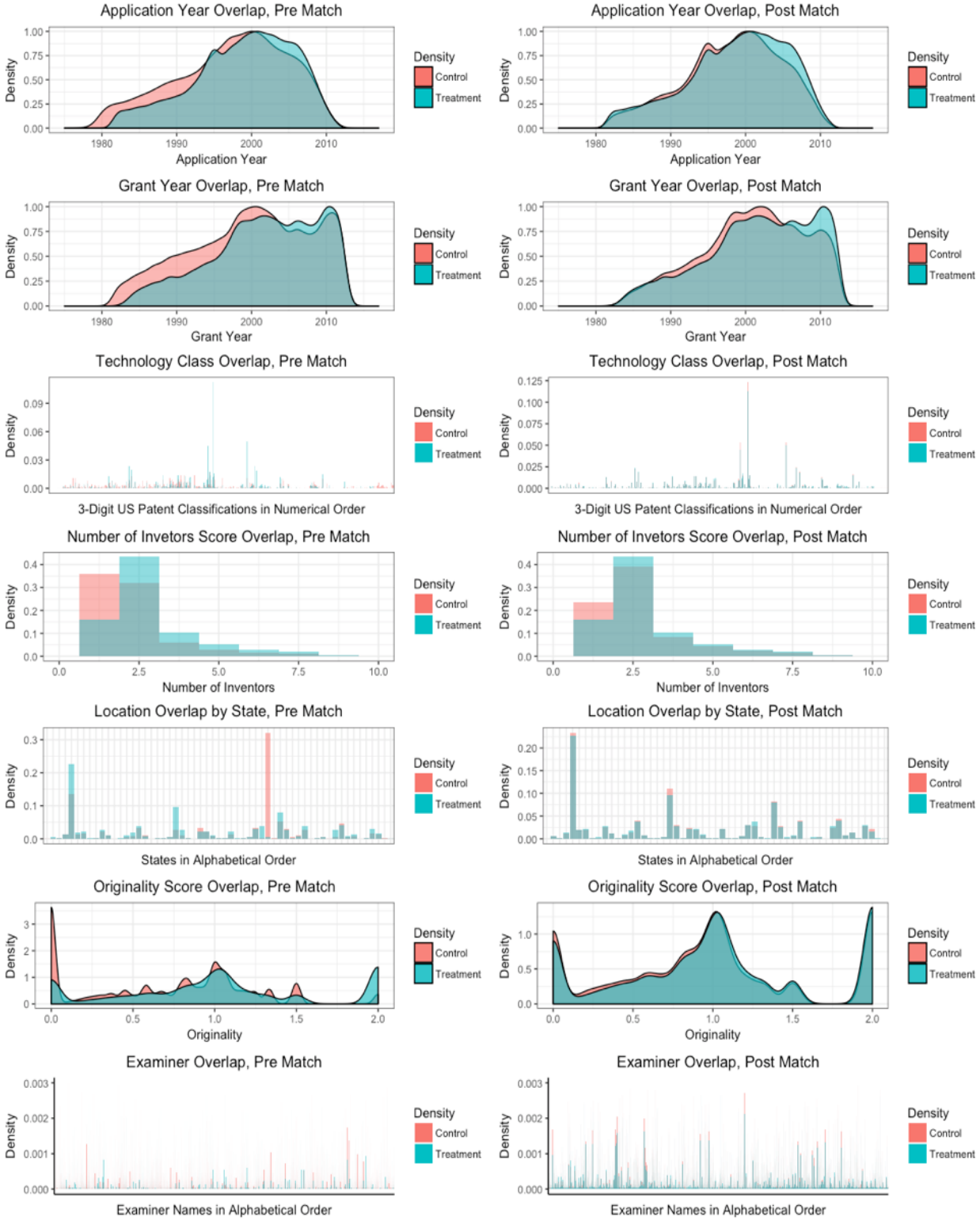
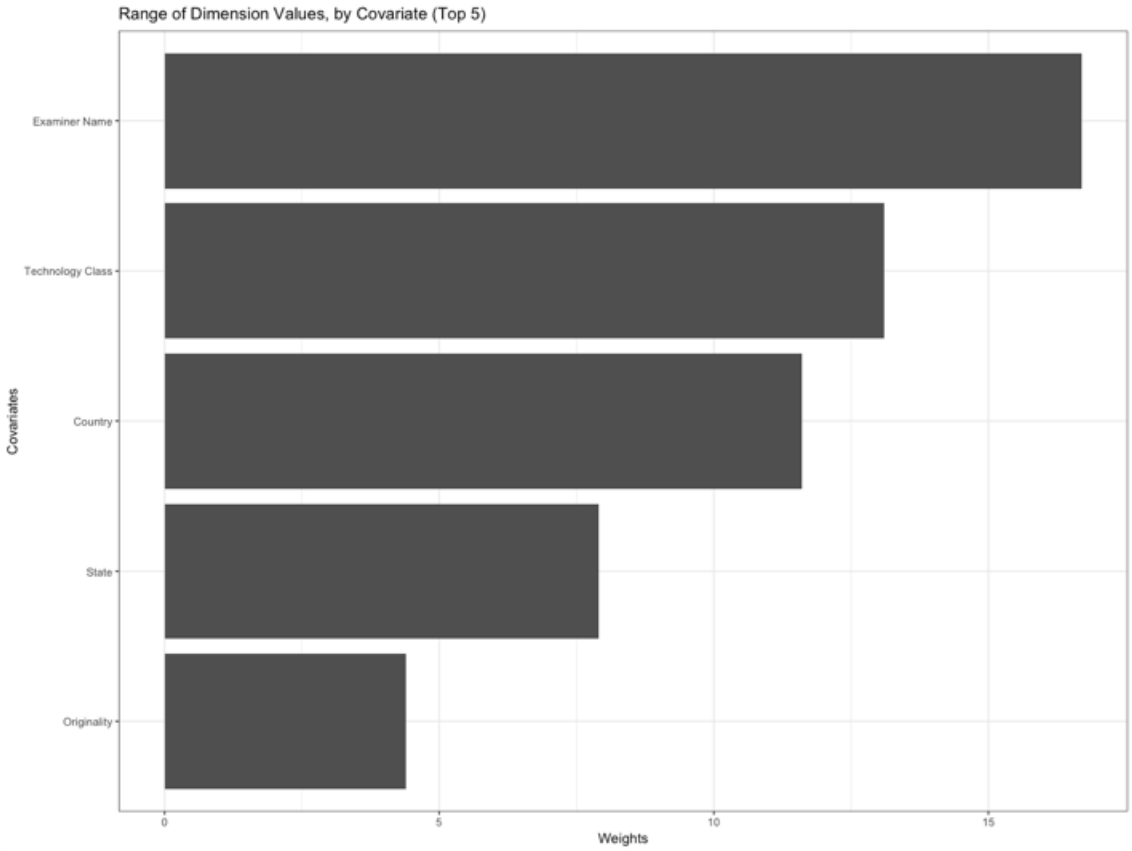


FIGURE 2. COVARIATE-WEIGHT GRAPH



Covariate-weight graph representing the range of covariate weights, per the top 5 covariates. As compared to traditional covariates used in patent matching, we find patent examiners to be particularly good predictors of public-funding.

FIGURE 3. MACHINE LEARNING- PROPENSITY SCORE MATCHING CALCULATED PROPENSITY SCORE BALANCE PLOTS, PRE & POST-MATCH

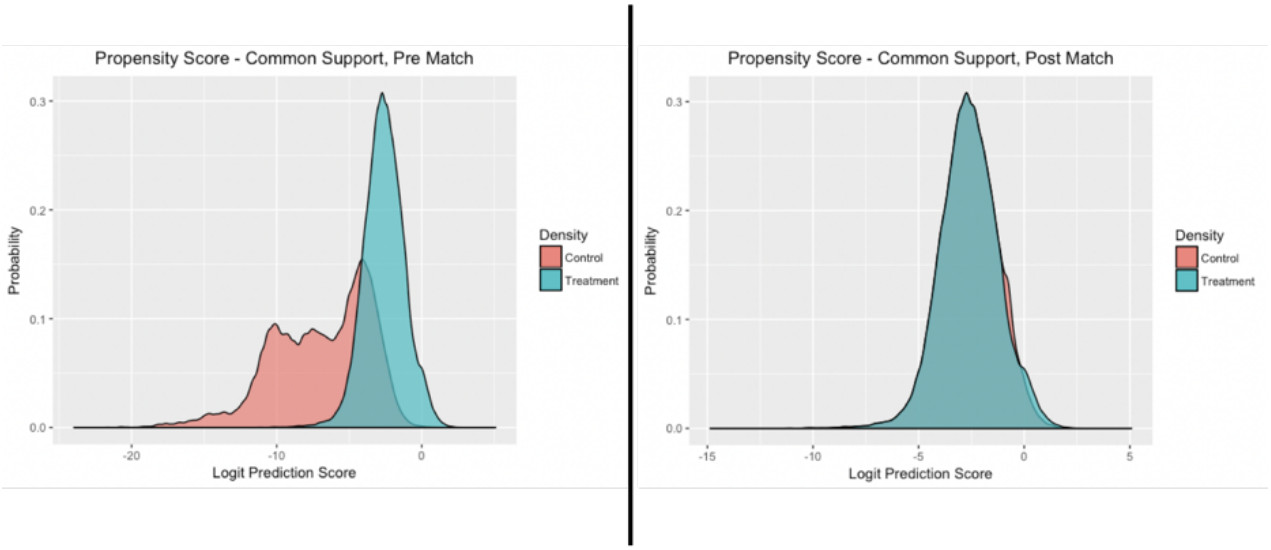


FIGURE 4. SECOND STAGE COEFFICIENT PLOT OF OLS PREDICTING DISRUPTION ACROSS 3 DIFFERENT FIRST STAGE INFERENCE STRATEGIES

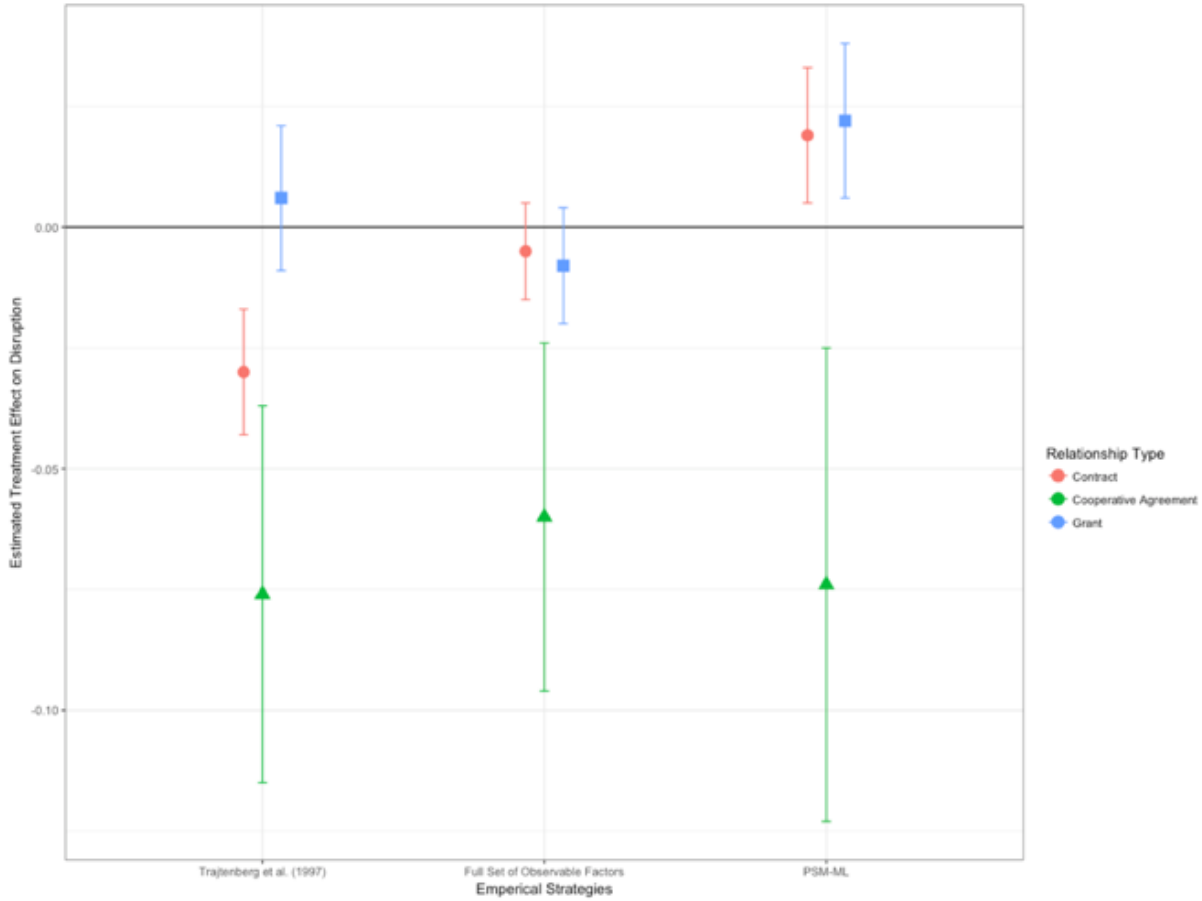


FIGURE 5. PROPENSITY SCORE BALANCE ACROSS THREE TREATMENTS

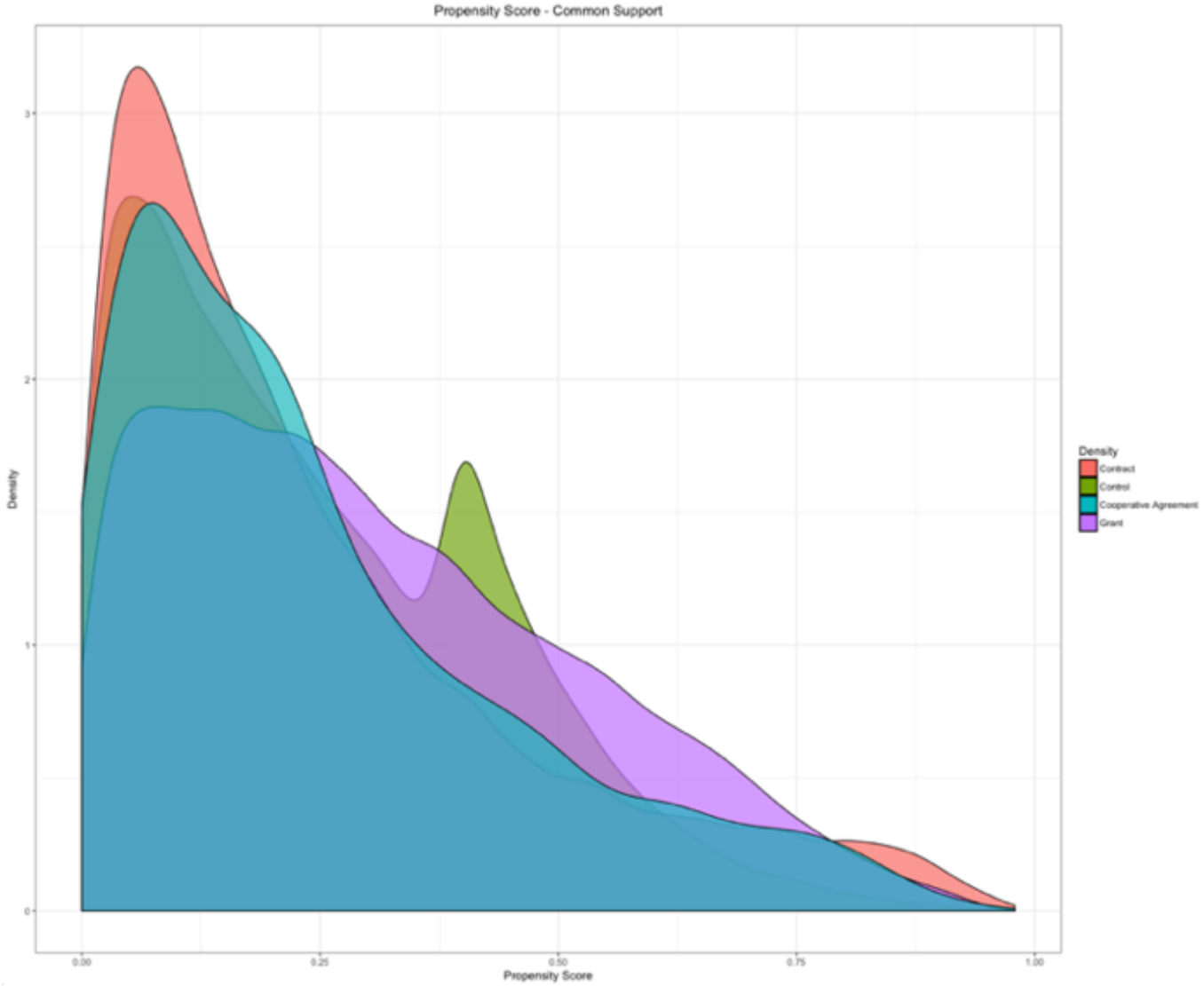


TABLE 1. EXAMINER NAME WEIGHTS. *Nine examiners who either were more likely to examine corporate-only patents (A-C), publicly-funded patents (G-we) or neither (D-F)*

Examiner ID	Examiner Name	Covariate-Dimension Weight
A	Zarfas, S	-13.11
B	Ansher, B	-12.89
C	Douglas, P	-12.71
D	Knife, M	0
E	Renner, A	0
F	Chan, F	0
G	Goddard, B	6.3
H	Grarsay, T	6.45
we	Ryam, P	6.49

TABLE 2. PREDICTIVE PERFORMANCE (AUC) VS. FIT (R^2) PREDICTING PUBLIC FUNDING

First Stage Methods		Training Data		Test Data	
Number	*Model*	*AUC*	*R-squared*	*AUC*	*R-squared*
1	*Trajtenberg et al. (1997) subsampling with covariates technology class, application year, grant year*	0.514	0.000	0.514	0.000
2	*Propensity Score Matching-Machine Learning with Trajtenberg et al. (1997) covariate set (technology class, application year, grant year)*	0.768	0.000	0.768	0.000
3	*PSM-ML with additional covariates (patent inventors, geographic location, originality, patent examiners)*	0.896	0.003	0.886	0.003
4	*PSM-ML with covariates from #3 + Quadratic Interactions, removing Regularization*	0.939	0.058	0.894	0.038
5	*PSM-ML with covariates from #3 + Quadratic Interactions*	0.912	0.058	***0.896***	0.038
6	*PSM-ML with covariates from #5 + Cubic Interactions*	0.955	0.084	0.881	0.048

TABLE 3. PROPENSITY SCORE MATCHING-MACHINE LEARNING ROAD MAP

Step	Activities
1	*Select covariates*
2	*Plot covariate distributions*
3	*Build a propensity score model*
3.1	***Split the data into training, validation, and test sets***
3.2	***Select a propensity score matching-machine learning model (i.e., logistic regression)***
3.3	***Train & validate model using elastic-net regularization***
3.4	***Evaluate model performance on test data***
3.5	***Repeat steps 3.2-3.4 with additional covariates/interactions***
3.6	***Select the best propensity score model***
4	*Use propensity score model to generate propensity scores*
5	*Match treatment and control observations which have the same scores*
6	*Plot covariate distributions, post-match*
7	*Run second-stage regression*

The unique machine learning approach is bolded in steps 3.1-3.6. The other, un-bolded steps represent the conventional propensity score matching approach

TABLE 3. EMC SIMULATION RESULTS

Values in this table indicate 2nd stage absolute difference of the predicted treatment effect from the true score of 0. Lower scores indicate a less biased estimate.

Sample size: 4,000			
Model type			
Analytic solution - unregularized without high dimensional factor variables and with quadratic interactions		0.0803	
Coarsened exact matching		0.247	
	Number of passes		
	1 (underfit)	*100 (overfit)*	*hyperopt optimization of passes (max 50)*
Stochastic gradient descent - unregularized without high dimensional factor variables and with quadratic interactions	0.0831	0.0809	**0.0753**
Stochastic gradient descent - unregularized with high dimensional factor variables and with quadratic interactions	0.0927	0.0806	0.0827
Stochastic gradient descent - regularized with high dimensional factor variables and with quadratic interactions	0.0874	0.0817	0.0759
Sample size: 32,000			
Model type			
Analytic solution - unregularized without high dimensional factor variables and with quadratic interactions		0.0288	
Coarsened exact matching		0.0489	
	Number of passes		
	1	*100*	*hyperopt optimization of passes (max 50)*
Stochastic gradient descent - unregularized without high dimensional factor variables and with quadratic interactions	0.0292	0.0316	0.0300
Stochastic gradient descent - unregularized with high dimensional factor variables and with quadratic interactions	0.0307	0.0312	0.0288
Stochastic gradient descent - regularized with high dimensional factor variables and with quadratic interactions	0.0282	**0.0262**	0.0325
Sample size: 256,000			
Model type			
Analytic solution - unregularized without high dimensional factor variables and with quadratic interactions		0.0176	
Coarsened exact matching		0.0188	
	Number of passes		
	1	*100*	*hyperopt optimization of passes (max 50)*
Stochastic gradient descent - unregularized without high dimensional factor variables and with quadratic interactions	0.0146	0.0140	0.0151
Stochastic gradient descent - unregularized with high dimensional factor variables and with quadratic interactions	0.0099	0.0184	0.0094
Stochastic gradient descent - regularized with high dimensional factor variables and with quadratic interactions	**0.0092**	0.0151	0.0113

best score; worst score

APPENDIX

Causal Inference in Strategy. While much strategy research puts the spotlight on parameter estimation in linear regressions (estimates of parameters β that underlie the relationship between y and x) – a task that

machine learning algorithms are not built for (Mullainathan and Spiess, 2017) – we highlight a useful application of machine learning to prediction in strategy.

Demonstrating causation is a continuous challenge for empirical strategy research. A typical challenge is when one group is exposed to a well-defined treatment (e.g. type of acquisition) but unlike in randomized experiments, no experimental design is in place to track a comparable control group that is not exposed to the treatment. Such systematic differences between treated and comparison units can generate selection bias, which occurs when an observation's covariate characteristics influence both the probability that the observation is treated and the outcome of that treatment, i.e., the characteristics "confound" treatment (Rubin, 1990).

Experiments randomly assign treatments to observations, ensuring that there are no systematic differences between treatment and control groups (Shadish, Cook, and Campbell, 2002). While the premier inference technique for controlling for differences across covariates is thus randomized experiments, such experiments are often unavailable to strategy scholars who study real business organizations and industries. Strategy is often interested in settings in which randomized experiments are unavailable, either because they are proprietary, prohibitively costly, or even unethical. And because strategic decisions to engage in an activity (or not) are inherently complex and interdependent (Leiblein et al., 2018), and typically decided by firms (not government policy that could involve randomization), finding comparable treated and control units is especially challenging for strategy scholars. As a result, strategy research has turned to a wide range of quasi-experimental designs (e.g., matching, instrumental variables) to help control for selection.

Both matching (propensity score matching in particular) and instrumental variables include a two-stage procedure. The first stage is the prediction step, and the result of predictions (treatment and control observations with similar propensity scores, or fitted values of x) then enter the second stage.[21] In this way, machine learning can be used in the first stage prediction without concerns about interpreting parameter estimates in the first stage (Mullainathan and Spiess, 2017). Although both instrumental variables and matching share this same principle, we use matching methods to illustrate the use of supervised machine learning in this paper.[22]

---

[21] Matching happens in two stages. In the first stage, researchers identify the observable characteristics which may influence selection to treatment, i.e., "confounding covariates." Matched subsamples of treatment and control observations are then identified such that they are "balanced" with respect to the confounding covariates, i.e., the covariate distributions are nearly (i.e., statistically) identical across the treatment and control samples. This stage is critical. If confounding covariates are not properly accounted for, matching will fail. In the second stage, scholars measure the treatment effect using the "balanced sample." By using only the balanced sample, thus mimicking as closely as possible the gold standard of randomized experiment, researchers have greater confidence that their treatment effect is unconfounded, i.e., the measured treatment effect is due to the treatment and not due to the confounding covariates (Imbens, 2004).

[22] While many prediction algorithms exist (including random forests and regression trees (Choudhury et al 2021), we focus on logistic regression because our intent is to expand existing methods that are commonly used in strategy, and

Matching approaches. The purpose of matching is to control for selection bias by generating a matched set of treatment and control observations that are similar across a number of observable covariates and their interactions, with the objective to minimize the differences between the two groups. Matching methods, therefore, provide a useful method to select the appropriate observations to include in the sample to measure treatment effects.

Matching methods that identify a "balanced sample" and then run the regression are attractive for strategy researchers for two reasons. First, when the covariate distributions of observations are vastly different, regressions on all observations rely on *extrapolation* to measure treatment effects. As a result, regressions report treatment effects that apply to observations which, statistically, are not able to be treated, rendering them inferentially invalid. Matching methods restrict the testable domain to only those observations which can be treated, and therefore, the treatment effect calculated with matching methods centers on the effect that is of particular interest to strategy scholars. Second, matching methods do not use the outcome variable, and therefore, "preclude the selection of a particular design to yield a desired result" (Stuart and Rubin, 2011: 157). In other words, they provide some additional protection against practices such as *p-hacking* (Bettis, 2012). In combination, matching methods provide a good alternative for many strategy problems.

Propensity score matching has seen a particular increase in interest in the strategy field in the last decade. The motivation for using propensity score methods is that in many strategy questions, observable characteristics of observations (e.g. features of firms or their employees) that may bias the sample are high in number, and interdependent with one another (Rosenbaum and Rubin, 1983). Propensity scores, as opposed to exact matching, can more efficiently handle increased dimensionality (i.e. many determinants) by regressing available covariates into a single propensity score. Specifically, propensity score of a unit to be treated – i.e., $\boxed{Pr(X)}$ – is estimated with a binary, maximum likelihood regression technique (e.g., probit, logit) that regresses the treatment binary variable (i.e., "1" if treated, "0" if not) on the set of potentially confounding covariates. It then matches observations which have similar propensity scores.

Matching methods: Notation. The goal of deductive methods is to be able to derive consistent and unbiased estimates of the average treatment effect (ATE) (Wilson, 2014).[23] In its simplest form, the ATE (referred to as "treatment effect" in this paper) is the difference between the expected value of an outcome given a treatment ($E_X[E[Y_i|\, Z_i = 1; X]]$) and the expected value of an outcome given a control

because logistic regression can handle interactions and nonlinearity unlike many of the other supervised machine learning methods (https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7724478).

[23] For a full explanation, see Kim and Steiner (2016). We use their notation.

($E_X[E[Y_i|\, Z_i = 0; X]]$). For the i^{th} observation, Y_i denotes the outcome, Z_i denotes whether or not the outcome was sampled from treatment group, and X denotes the full set of covariates.

$ATE = E_X[E[Y_i|\, Z_i = 1]] - E_X[E[Y_i|\, Z_i = 0; X]]$ To argue that the difference between expected values will derive an unbiased treatment effect, scholars must assume that treatment assignment is conditionally independent (i.e., exogenous) in the observed covariates, $\boldsymbol{X}$ (Rosenbaum and Rubin, 1983).[24]

Notationally, this is represented as $\{Y(1), Y(0)\} \perp Z|\boldsymbol{X}$, where the set of observed treatment and control outcomes, $\{Y(1), Y(0)\}$, is independent of the treatment, Z, given the set of observed covariates, $\boldsymbol{X}$**.** It is important to note that while X denotes all covariates, $\boldsymbol{X}$ denotes the observed covariates (i.e., those included in the study).

Matching methods are explicitly used to generate the matched sample $\{Y(1), Y(0)\}$. The intent is to develop a matched sample such that the observed covariates, $\boldsymbol{X}$**,** are similar in distribution between treatment and control groups. By matching treatment and control observations which have similar covariate distributions, we can statistically argue that the treatment is conditionally independent of the outcome given the set of observed covariates, $\{Y(0), Y(1)\} \perp Z|\boldsymbol{X}$ (Kim and Steiner, 2016).

The key, then, for matching methods, is that $\boldsymbol{X}$ must account for the complete set of confounding covariates (Stuart and Rubin, 2008). If exogeneity within the matched set is assumed, then scholars are also assuming that the unobserved covariates are not influencing the treatment effect (i.e., they are "unconfounding" covariates). As a result, any method that employs matching for unbiased estimation is assuming exogeneity external to the matched sample. For reference, there is no way to quantitatively prove exogeneity, although we do outline one method in Section 5 that helps to provide some additional quantitative evidence.

In its simplest form, supervised learning estimates a predictive model's coefficient weights with a maximum likelihood estimator: $L \equiv \sum_i^n (Y_i - H_\theta(X_i))^2$ where $H_\theta(X_i)$ is the predictive model, Y_i is the outcome variable, X_i is the set of observed covariates, and θ is the set of coefficient weights. To build a model, the optimizing solver iteratively updates estimates of θ to drive likelihood (L) to zero. For reference, this "base" likelihood estimator is equivalent to traditional econometric approaches (Rosenbaum and Rubin, 1983). The difference is the regularization term. This term prevents overfitting by penalizing the estimator if the estimator tries to weight particular coefficients too heavily.

The two most common forms of regularization applicable for propensity scores are Lasso (L1) and Ridge (L2). L1 regularization penalizes the estimator by driving the weights of dimension coefficients which do not contribute to selection to zero. Using L1 regularization, overfitting is controlled for by

[24] This is also referred to as the "strong ignorability of treatment assignment" assumption (Stuart and Rubin, 2008: 158)

effectively deleting those observational factors which are *not confounding* (i.e., not predictive). The L1 penalized estimator takes the form of: $L \equiv \sum_{i=1}^{n}\left(Y_i - H_\theta(X_i)\right)^2 + \lambda_1 \sum_{i=1}^{n} |\theta_i|$ The L2 regularization normalizes coefficient weights to ensure that no singular dimension dominates the model. Thus, L2 regularization forces coefficient weights to be small, but keeps all information in the model. While it therefore minimizes the impact of any single dimension, it remains prone to including non-confounding dimensions. The L2 penalized estimator takes the form $L \equiv \sum_{i=1}^{n}\left(Y_i - H_\theta(X_i)\right)^2 + \lambda_2 \sum_{i=1}^{n} \theta_i^2$ Finally, the combination of both L1 and L2, i.e., *elastic net regularization*, is particularly useful for matching (Athey, Imbens, and Wager, 2018; Zou and Hastie, 2005) computed as $L \equiv \sum_{i=1}^{n}\left(Y_i - H_\theta(X_i)\right)^2 + \lambda_1 \sum_{i=1}^{n}|\theta_i| + \lambda_2 \sum_{i=1}^{n} \theta_i^2$ Elastic net regularization not only handles overfitting but also decreases curse of dimensionality problems by algorithmically removing covariates and, therefore, intelligently limiting the number of dimensions. As the number of dimensions approaches, or surpasses, the number of observations, L1 regularization begins randomly deleting dimensions with equivalent covariance (Zou and Hastie, 2005). While this certainly minimizes overfitting, it is also likely to remove confounding dimensions, violating the exogeneity assumption. Elastic-net regularization combines L1 and L2 such that L2 can re-weight dimensions before L1 sets dimension coefficient weights to zero, minimizing the likelihood of equivalent covariance. As a second stage, then, L1 can more accurately remove non-confounding dimensions. In combination, elastic net regularization quantitatively and algorithmically determines which covariates are confounding covariates and thus helps avoid overfitting and curse of dimensionality problems.]]]