A Super-Learning Machine for Predicting Economic Outcomes

Cerulli, Giovanni

IRCrES-CNR, Research Institute on Sustainable Economic Growth, National Research Council of Italy

10 March 2020

A Super–Learning Machine for Predicting Economic Outcomes*

Giovanni Cerulli
CNR-IRCRES
Research Institute on Sustainable Economic Growth
National Research Council of Italy
giovanni.cerulli@ircres.cnr.it

March 12, 2020

Abstract

We present a Super–Learning Machine (SLM) to predict economic outcomes which improves prediction (i) by cross–validated *optimal tuning*, (ii) by comparing/combining results from different learners. Our application to a labor economics dataset shows that different learners may behave differently. However, combining learners into one singleton super–learner proves to preserve good predictive accuracy lowering the variance more than stand-alone approaches.

Keywords: Machine learning, Ensemble methods, Optimal prediction
JEL Classification: C53, C61, C63

*This paper was presented at the HORIZON2020 Program financed project RISIS (European Research Infrastructure for Science, technology and Innovation policy Studies) WEEK, held at ISI Fraunhofer (Karlsruhe, Germany) on January 27th–30th, 2020. I wish to thank all the participants for their useful suggestions.

1 Introduction

The quest for an objective science is in a surge. In economics, some scholars have recently stressed the need for sounder credibility and fairness of empirical research (Angrist and Pischke, 2010). Machine Learning (ML), a relatively new approach to data analysis, may help "taking the con out of econometrics".

Placing itself in the intersection between statistics, computer science, and artificial intelligence, ML main objective is turning information into valuable knowledge by "letting the data speak", limiting model's prior assumptions, and promoting a model-free philosophy. Relying on algorithms and computational techniques, ML targetes Big Data and complexity reduction, although sometimes at expenses of results' interpretability (Varian, 2014).

ML has emerged as a new scientific paradigm within numerous sciences, but its use in economics and econometrics is still lagging behind. One shared belief among economists is that ML is powerful for prediction, whereas less useful for inferential purposes (Athey, 2019). Recently, however, a new econometric literature is trying to bridge ML and causal inference through new ML–adapted methods able to tackle causal inference issues, such as treatment effect estimation with high dimensional data (Belloni, Chernozhukov, Hansen, 2014), heterogenous treatment effect estimation (Athey and Wager, 2017), and optimal policy assignment (Athey and Imbens, 2017).

By focusing on the predictive use of ML, this paper presents a Super–Learning Machine (SLM) to predict economic outcomes, both in regression and classification settings. Concisely, a SLM is an *ensemble* ML toolbox aimed at improving prediction of economic outcomes in two directions: (i) by targeting optimal modeling via cross-validated *optimal tuning*; (ii) by comparing and combining results from a large set of learners instead of relying on one single method as usual done in economics[1].

The illustrative application presented in this paper focuses on classification, but the extension to regression is immediate. We aim at predicting the wage class (categorized as "low", "medium", and "high") of an individual based on her characteristics. We do it by comparing (and then combining via a majority vote ensemble rule) eight different cross-validated learners, stressing the role played not only by larger predictive accuracy, but also by wider accuracy uncertainty.

The structure of the paper is as follows. Section 2 presents the SLM logic and architecture. Section 3 illustrates our application and discusses the results. Section 4 concludes the paper.

2 The SLM architecture

I define a learner L_j as a mapping from the set $[X, \theta, \lambda_j, f_j(\cdot)]$ to an outcome y, where X is the matrix of features, θ a vector of estimation parameters, λ_j a vector of tuning parameters, and $f_j(\cdot)$ an algorithm taking as inputs X, θ, and λ_j. Generally,

[1]The main reference on the statistics of the super–learning prediction can be found in Van der Laan and Rose (2011).

economists use a singleton $f_j(\cdot)$ for modeling and predicting economic outcomes, typically one member of the Generalized Linear Models (GLM) family (linear, probit or multinomial regressions are classical examples). GLM are highly parametric and are not characterized by tuning parameters. Nonparametric models, such as local–kernel, nearest–neighbor, or tree regressions are on the contrary characterized by one or more hyper-parameters λ_j which may be optimally chosen to minimize the so–called *test prediction error*, i.e. the out–of–sample predicting accuracy of the learner.

Figure 1 presents the architecture of the SLM herein proposed. This framework is made of three linked learning processes: (i) the learning over the tuning parameter λ, (ii) the learning over the algorithm $f(\cdot)$, and (iii) the learning over new additional information. The departure is in point 1, from where we set off assuming the availability of a dataset $[X, y]$.

The first learning process aims at selecting the optimal tuning parameter(s) for a given algorithm $f_j(\cdot)$. ML scholars typically do it using K-fold cross–validation, a re-sampling approach estimating the out–of–sample performance of a learner by leaving one group of observations out of the estimation, and then using prediction over these left-out observations to measure predictive accuracy. This procedure is iteratively repeated for each fold, eventually obtaining K test–accuracy (or, equivalently, test error) measures over which taking the average and the standard deviation.

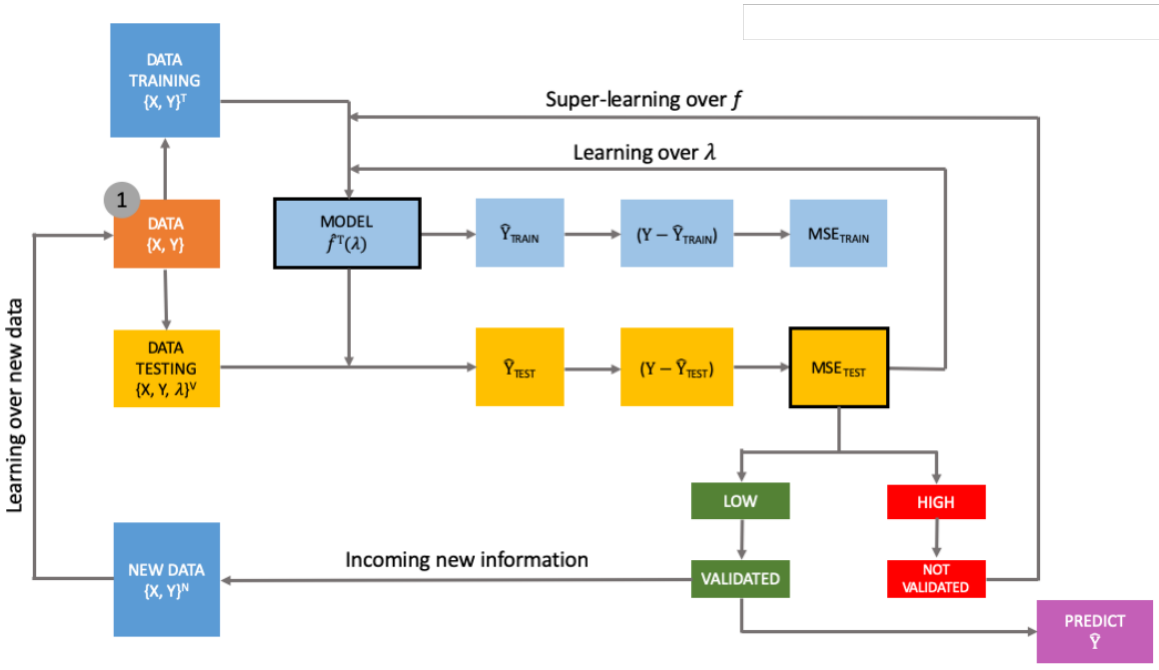

Figure 1: The Super–Learning Machine architecture.

At the optimal λ_j, one can recover the largest possible prediction accuracy for the learner $f_j(\cdot)$. Further prediction improvements can be achieved only by learning from other learners, namely, by exploring other $f_j(\cdot)$, with $j = 1, \ldots, M$ (where M is the number of learners at hand).

It is important to observe that the so–called *training error*, i.e. the in–sample

predictive performance of a learner, is a misleading measure of the actual model fit as plagued by the *overfitting* phenomenon: it may be the case that the training error decreases monotonically with the tuning parameter even if the out–of–sample performance of the learner is worsening. In figure 1, it corresponds to the light blue sequence of boxes leading to the MSE_{TRAIN} which is in fact a dead-end node, as not informative for making correct decisions.

Conversely, the yellow sequence leads to the MSE_{TEST}, which is informative to take correct decisions about the predicting quality of the current learner. At this node, the analyst can compare the current MSE_{TEST} with a benchmark one (possibly, pre–fixed), and conclude whether to predict using the current learner, or explore alternative learners in the hope of increasing predictive performance. If the level of the current prediction error is too high, the SLM would suggest to explore other learners.

In the ML literature, learning over learners is called *meta learning*, and entails an exploration of the out–of–sample performance of alternative algorithms $f_j(\cdot)$ with the goal of identifying one behaving better than the those already explored (Hastie, Tibshirani, and Friedman, 2001). For each new $f_j(\cdot)$, the SLM finds an optimal tuning parameter and a new estimated accuracy (along with its standard deviation). The analyst can either explore the entire bundle of alternatives and finally pick–up the best one, or decide to select the first learner whose accuracy is larger than the benchmark. Either cases are automatically run by the machine.

The third final learning process concerns the availability of new information, via additional data collection. This induces a reiteration of the initial process whose final outcome can lead to choose a different algorithm and tuning parameter(s), depending on the nature of the incoming information.

As final step, one may combine predictions of single optimal learners into one single super–prediction (*ensemble learning*). What is the advantage of this procedure? As an average, this method cannot provide the largest accuracy possible. However, as sums of i.i.d. random variables have smaller variance than the single addends, the benefit consists of a smaller predictive uncertainty (Zhou, 2012; Escanciano et al., 2014).

3 Application

We are interested in predicting the wage class of an individual based on her characteristics[2]. We have data on 500 individuals from the National Longitudinal Survey of Young Women (NLSW) in 1988. Our target dependent variable, `wage_class`, is categorical with three classes of hourly wage: *low*, *medium*, and *high*. As individual features, we consider: `age`: age of the woman; `race`: race of the woman (white, black, other); `married`: married vs. non–married; `never_married`: whether or not never married; `grade`: grade obtained at school final exam; `south`: whether or not the woman comes from the South; `smsa`: whether she lives in SMSA; `c_city`: whether

[2]The SLM has been programmed in Python 3, using the Stata/Python integrated interface available in Stata 16. All codes are available on request.

or not she lives in central city; `collgrad`: whether she is college graduated; `hours`: usual hours worked; `ttl_exp`: total work experience; `tenure`: job tenure in years; `industry`: type of industry; `occupation`: type of occupation.

Learner	*Tuning parameter 1*	*Tuning parameter 2*	*Tuning parameter 3*
Naïve Bayes			
Regularized Multinomial	*Penalization coefficient*		
Nearest-Neighbor	*Number of neighbors*		
Neural Network	*Number of hidden layers*	*Number of neurons*	
Trees	*Number of leaves (or tree-depth)*		
Boosting	*Learning parameter*	*Number of bootstraps*	*Tree-depth*
Random Forest	*Number of features for splitting*	*Number of bootstraps*	*Tree-depth*
Support Vector Machine	*C*	*Gamma*	

Figure 2: Learners and related tuning parameters.

Figure 2 sets out the eight learners considered by our SLM. Except the Naive Bayes, all the other learners present at least one tuning parameter over which the SLM optimizes. These learners are: Boosting, Decision tree, Naive Bayes, Nearest–Neighbor, Neural Network, Random Forest, Regularized Multinomial, and Support Vector Machine.

Figure 3 shows each learner's test and training accuracy as a function of the (main) tuning parameter, and identifies the optimal tuning parameter as the one maximizing test accuracy. Focusing on the Decision Tree, for example, we clearly observe the overfitting produced by increasing the number of leaves (i.e. a monotonic increase of the training accuracy), which is optimal at a value of 7, where the test accuracy is maximized. A similar pattern can be observed for the Nearest–Neighbor, where the optimal numbers of leaves is 7. In this case, the overfitting occurs as a function of $1/K$, with K the number of nearest neighbors.

Figure 4 sets out a summary of the results and provides insights for the choice of the classifier. The learner with the best accuracy is the Regularized Multinomial obtaining an optimal accuracy of 62% which is the out–of–sample probability to correctly classify the wage class of a new out–of–sample woman. The worst classifier in this dataset is the Support Vector Machine with an accuracy of 46%.

Information on the accuracy's standard error is also relevant, as allowing for determining accuracy confidence interval. In this regard, the column with heading "weight" displays the value of $1/\sigma_{L_j}$: the larger the weight, the more precise the estimate of the accuracy. The Neural Network presents the largest weight (33), the worst one being obtained by the Naive Bayes (6.28). Overall, the behavior of the Neural Network in this dataset is quite good, as it performs well both in terms accuracy (59%) and precision.

Finally, the row with heading "Overall" sets out the performance of the meta–learner, taking the form of a *majority vote* classifier. It would reach an accuracy of 56% with 95% confidence interval ranging between 51% and 60%, smaller than

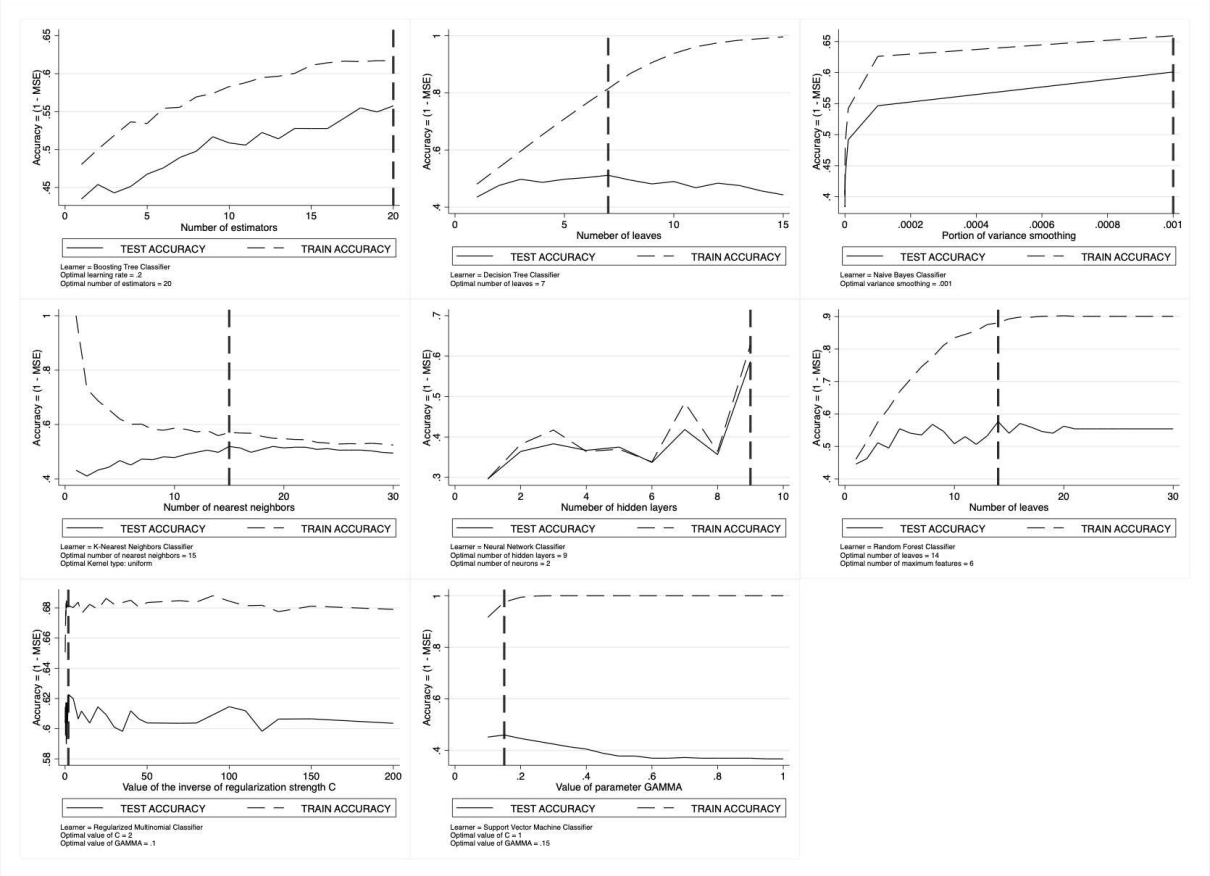

Figure 3: Learners' test and training accuracy as a function of the main tuning parameter.

the one provided by the best learner (as seen, the Neural Network). This result was expected as ensemble learners are purposely built for reducing estimation uncertainty, generally at negligible expenses in terms of accuracy reduction.

4 Conclusion

As an ensemble ML toolbox, our SLM improves prediction in two directions: (i) by model's *optimal tuning*; (ii) by comparing and combining results from a many learners. Our economic application shows that different learners have different performance, both in terms of accuracy and variability. Combining learners into a singleton super–learner preserves reasonable accuracy by allowing lower variance than stand-alone approaches.

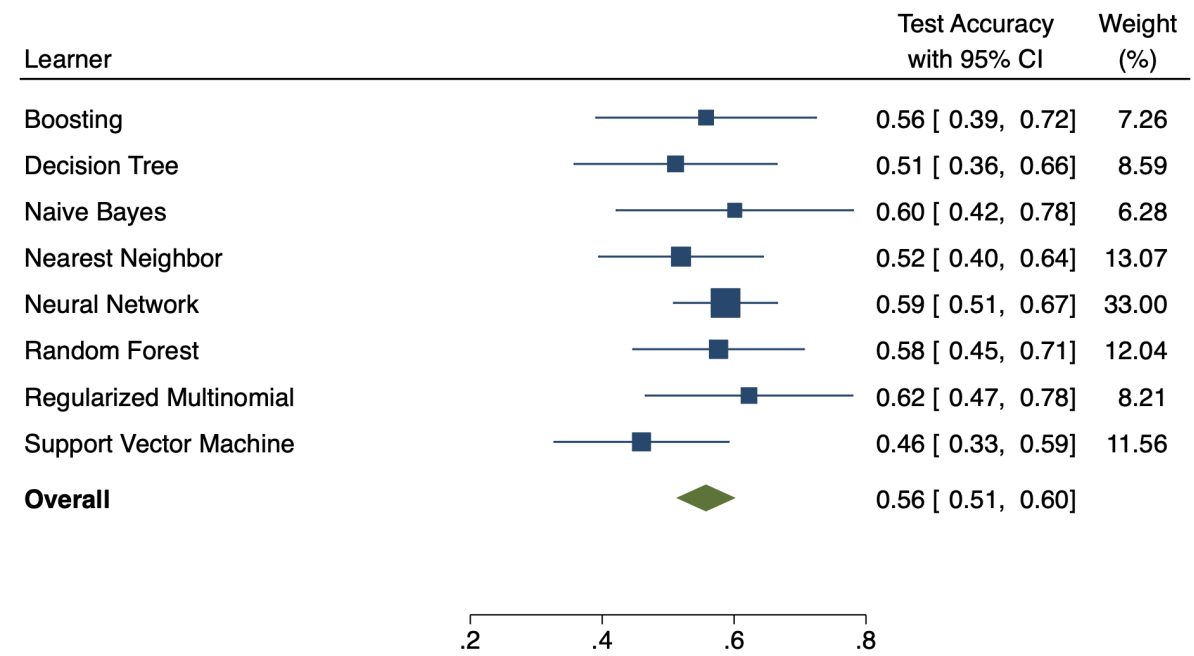

Figure 4: Predictive accuracy meta analysis.

References

[1] Angrist J.D., Pischke J.S. 2010. The Credibility Revolution in Empirical Economics: How Better Research Design Is Taking the Con out of Econometrics. *Journal of Economic Perspectives*, 24(2): 3-30.

[2] Varian H.R. 2014. Big Data: New Tricks for Econometrics. *Journal of Economic Perspectives*, 28(2): 3-28.

[3] Athey S. 2019. "The Impact of Machine Learning on Economics", Chapter in NBER book: A. Agrawal, J. Gans, and A. Goldfarb (Eds.), *The Economics of Artificial Intelligence: An Agenda*. University of Chicago Press (pp. 507-547).

[4] Belloni A., Chernozhukov V., Hansen C. 2014. High-Dimensional Methods and Inference on Structural and Treatment Effects. *Journal of Economic Perspectives*, 28(2): 29-50.

[5] Athey S. and Imbens G.W. 2017. The state of applied econometrics: Causality and policy evaluation. *Journal of Economic Perspectives*, 31(2):3-32.

[6] Athey S. and Wager S. 2017. "Efficient policy estimation". arXiv preprint arXiv:1702.02896. URL: https://arxiv.org/abs/1702.02896.

[7] Hastie T., Tibshirani R., and Friedman J. 2001. *The elements of Statistical Learning - Data Mining, Inference, and Prediction*. Berlin: Springer-Verlag.

[8] Zhou Z.H. 2012. *Ensemble Methods: Foundations and Algorithms*. CRC Press.

[9] Van der Laan M.J. and Rose S. 2011. *Targeted learning: causal inference for observational and experimental data*. Springer.

[10] Escanciano J.C., Jacho-Chvez D., and Lewbel A. 2014. Uniform Convergence of Weighted Sums of Non- and Semi-parametric Residuals for Estimation and Testing. *Journal of Econometrics*, 178: 426-443.