An interpretable machine learning work ow with an application to economic forecasting

Andreas Joseph

Bank of England

Joint work with Marcus Buckmann (BoE)

Machine Learning and Macroeconomic Analysis
(organised by the Deutsche Bundesbank)

2023 EEA-ESEM Congress
30. August, Barcelona

Disclaimer: The views expressed here are my own and should not be represented as those of the Bank of England (BoE). All errors are ours.

Table of contents

Motivation

The machine learning (ML) setting

Everything here is about supervised learning, i.e. minimising an error

$$\min_{\theta} \; \mathbb{E}_{\Omega}\left[||y - \hat{f}_{\theta}\|_{l}\right] .$$

However, many aspects can be transferred to unsupervised or reinforcement learning (only need some form of model prediction).

Problem: θ not identifying, i.e. degeneracy of parameter sets.

$\Rightarrow$ Black box problem.

Key assumption: Key theorem of statistical learning holds

The problems we consider are learnable. The empirical risk $R_e(y, \hat{f}(x, \theta))$ converges uniformly in probability to the actual risk $R(y, \hat{f}(x, \theta))$, i.e. let $\delta > 0$, then

$$\lim_{m \to \infty} P\left(\sup_{\theta}(R - R_e) > \delta\right) = 0\,. \tag{1}$$

That is, our ML models are error consistent.

See Vapnik (1999), but also Shalev-Shwartz et al. (2010).

NB: Inference on some low dimensional objects still possible when (1) not given (see Chernozhukov et al. (2023)).

Pros & Cons of ML relative to econometric approach

Advantages

- Often higher accuracy
- Lower risk of misspecification
- Return richer information set

Disadvantages

- Higher model complexity ("black box critique")
- Less analytical guarantees, e.g. risk of overfitting
- Often larger data requirement

Well, universal learning from data is great, BUT ...

- ... how to understand what the model is actually doing?
- ... how to relate ML models to an econometric approach, e.g. hypothesis testing?
- ... how to quantify the learning process, e.g. its state of convergence?
- ... how to communicate results for non-linear models?

ML workflow

The ML workflow

1. Comparison of model predictions ("horse race" if accuracy is the goal)

   $\Rightarrow$ Is there gain in using ML, should I continue?

2. Model decomposition into Shapley values

   $\Rightarrow$ Identify important features & uncover learned functional forms

3. Statistical testing: "Shapley regression"

   $\Rightarrow$ Establish confidence & standard *communication*

Recap: Steps for statistical inference

I **Magnitude measurement**: Sample mean, regression coefficients, treatment effects, etc.

II **Hypothesis testing** (certainty assessment): t, F-tests, CLTs, etc.

The linear regression model (LR)

$$(I) : \hat{f}(x_i) = x_i\hat{\beta} = \sum_{k=0}^{n} x_{i,k}\hat{\beta}_k + \hat{\epsilon} \quad \text{with} \quad (II) : \mathcal{H}_0^k : \beta_k = 0 \tag{2}$$

- Workhorse of econometric analysis
- Special: *local* and *global* model inference ($\hat{\beta} = const.$)
- Widely accepted to be interpretable (if not too many regressors)
- Belongs to class of additive local variable attributions

$$\Phi(x_i) \equiv \phi_0 + \sum_{k=1}^{n} \phi_k(x_i) = \hat{f}(x_i) \tag{3}$$

Detour: Shapley values in cooperative game theory

- How much does player A contribute a collective payoff f obtained by a group of n? (Shapley, 1953).
- Observe payoff of the group with and without player A.
- Contribution depends on the other players in the game.
- All possible coalitions S need to be evaluated.

$$\phi_A = \sum_{S \subseteq n \setminus A} \frac{|S|!(|n| - |S| - 1)!}{|n|!}[f(S \cup \{A\} - f(S)] \tag{4}$$

$2^{|n|-1}$ coalitions are evaluated.

Computationally complex!

Shapley values as analogy between game theory and (ML) models

	Cooperative game theory	Machine learning
n	Players	Predictors / variables
$\hat{f}/\hat{y}$	Collective payoff	Predicted value for one observation
S	Coalition of players	Group of predictors in model
Source	Shapley (1953)	Štrumbelj and Kononenko (2010)
		Lundberg and Lee (2017)

Model Shapley decomposition: $\hat{f}(x_i) = \phi_0 + \sum_{k=1}^{n} \phi_k^S(\hat{f}; x_i)$

Why Shapley values? Because they are the only attribution scheme which is *local, linear, exact, respects the null, is consistent (Young, 1985), and allows for interactions* (Agarwal et al., 2019).

Shapley regression (SR) for statistical inference (Joseph, 2020)

Auxiliary inference analysis on $\hat{f}$ in the space of Shapley values:

$$y_i = \sum_{k=0}^{n} \phi_{ki}^S \hat{\beta}_k^S + \hat{\epsilon}_i \qquad \text{with} \qquad \mathcal{H}_0^k(\Omega) : \beta_k^S \leq 0 \tag{5}$$

Universality: $\hat{f}$ can be any model.

Interpretation: $\hat{\beta}^S$ measures the alignment of model components with the target.

Validity: Eq. 5 relates to generated regressors (Pagan (1984)) imposing minor conditions. Inference generally only valid on test set (standard in ML) and some consideration on convergence rates (cross-fitting helpful).

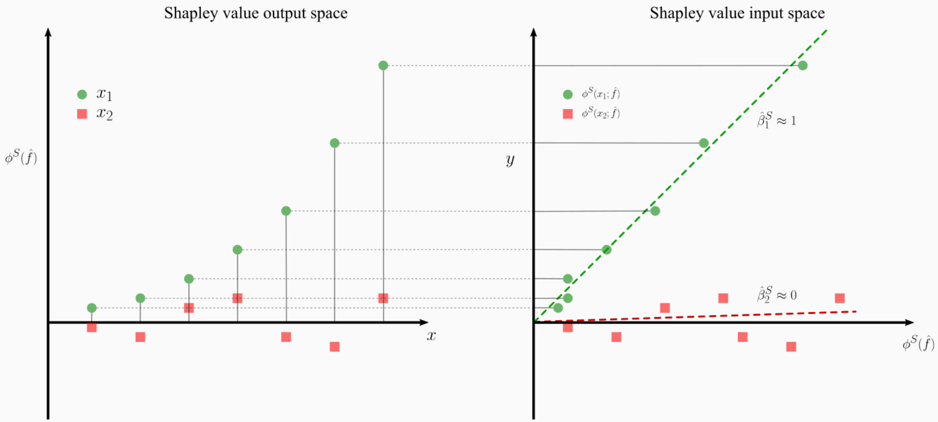
SR interpretation: alignment & learning progress

The true value of each β_k^S is either 1 (**signal**) or 0 (**pure noise**).

If $\mathcal{H}_1^k(\Omega) : \beta_k^S = 1$ is not rejected, we can say that information from variable k has been learned robustly (perfect alignment between y and ϕ_k^S).

Learning asymptotics: β_k^S track learning progress and distinguish between signal from noise.

SR graphical representation



Practical approach to valid inference

Let $\xi_{ml} \leq \frac{1}{2}$ be the convergence rate of our ML model, then we have

$$m_{test} \leq m_{train}^{2\xi_{ml}}. \tag{6}$$

Sample inefficiency can be avoided by cross-fitting with K partitions,

$$K \geq \left\lceil m^{1-2\xi_{ml}} + 1 \right\rceil. \tag{7}$$

Unknown ξ_{ml} can be approximated with a conservative estimate like $\xi_{ml} = \frac{1}{4}$

The resulting sampling uncertainty can be addressed using adjusted variance estimates (Chernozhukov et al. (2018)) over multiple random draws.

However, as K increases with m, the model variation between splits will decrease.

SR communication: Shapley share coefficients (SSC)

Normed summary statistic for the importance of x_k to the model $\hat{f}$ within a region Ω.

$$\Gamma_k^S(\hat{f},\Omega) \equiv \left[sign\left(\hat{\beta}_k\right) \left\langle \frac{|\phi_k^S(\hat{f})|}{\sum_{l=1}^m |\phi_l^S(\hat{f})|} \right\rangle_\Omega \right]^{(*)} \in [-1,1]$$

$$\overset{\hat{f}(x)=x\hat{\beta}}{=} \hat{\beta}_k^{(*)} \cdot \left\langle \frac{|(x_k - \langle x_k \rangle)|}{\sum_{l=1}^m |\hat{\beta}_k(x_l - \langle x_l \rangle)|} \right\rangle_\Omega \tag{8}$$

3 parts: **sign** (alignment of x_k and y), **size** (model fraction attributed to x_k) and **significance level** of $\hat{\beta}_k^S$ against $\mathcal{H}_0^k(\Omega)$.

$\Gamma_k^S(\hat{f},\Omega)$ is proportional to the coefficient of the linear model in the linear regression case (equivalence to SR).

Full recipe

Application

Forecasting setup

- Target: YoY change in US unemployment on a 1 year horizon
- Predictors: FRED-MD data base, McCracken and Ng (2016); 9 selected variables, lagged target
- Sample period: 1962:M2 - 2019:M11 (no Covid, no stress)
  - validation & training (yearly): Until 1989:M12
  - Testing: 1990:M1–2019:M11 (pseudo real-time), out-of-bag (full)
- Models:
  - *classical ML model:* Artificial neural networks (MLP), random forest, support vector regression (SVR), gradient boosted trees
  - *linear regressions:* OLS, Ridge, Lasso
  - *auto-regressions:* AR(1), AR(p) with $p \leq 12$ by AIC
- Hyper-parameters: (time series) 5-fold cross-validation, every 3 years
- Model-aggregation: Bootstrap aggregation ('bagging' over 100 draws)

Variable selection: Capture different economic channels

Variable	Transformation	Name in Source
Unemployment	changes	UNRATE
3-month treasury bill	changes	TB3MS
Real personal income	log changes	RPI
Consumption	log changes	DPCERA3M086SBEA
Industrial production	log changes	INDPRO
S&P 500	log changes	S&P 500
Business loans	second order log changes	BUSLOANS
CPI	second order log changes	CPIAUCSL
Oil price	second order log changes	OILPRICEx
M2 Money	second order log changes	M2SL

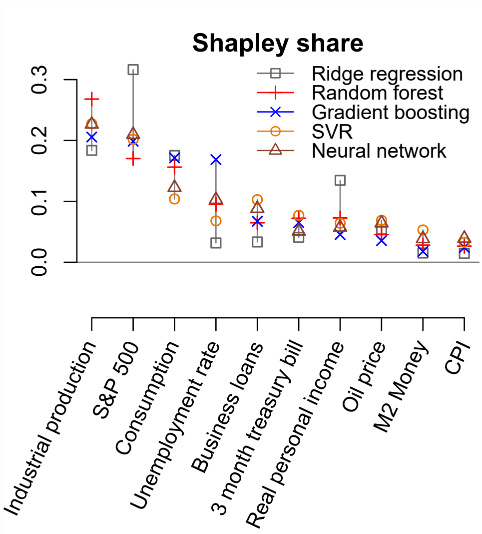
Transformations as suggested in McCracken and Ng (2016), using quarterly changes.

Step 1: Horse race results

Time period	01/1990–11/2019	01/1990–12/1999	01/2000–08/2008	09/2008–11/2019
Gradient boosting	**0.559** -	**0.460** -	**0.466** -	0.718 (0.353)
SVR	0.565 (0.323)	0.470 (0.328)	0.489 (0.219)	**0.709** -
Forest	0.581 (0.018)	0.472 (0.240)	0.471 (0.413)	0.762 (0.005)
Neural network	0.589 (0.009)	0.468 (0.336)	0.503 (0.070)	0.762 (0.001)
AR_1	0.608 (0.063)	0.472 (0.382)	0.503 (0.216)	0.811 (0.064)
AR_{12}	0.626 (0.001)	0.543 (0.011)	0.482 (0.356)	0.810 (0.001)
Lasso regression	0.637 (0.000)	0.498 (0.061)	0.474 (0.378)	0.886 (0.000)
Ridge regression	0.639 (0.000)	0.497 (0.065)	0.481 (0.272)	0.886 (0.000)
OLS regression	0.648 (0.000)	0.516 (0.016)	0.508 (0.053)	0.872 (0.000)

Forecast comparison in the baseline set-up using MAE. P-values in parentheses indicate the statistical significance for (one-sided) DM test. Sources: McCracken and Ng (2016) and authors' calculation.

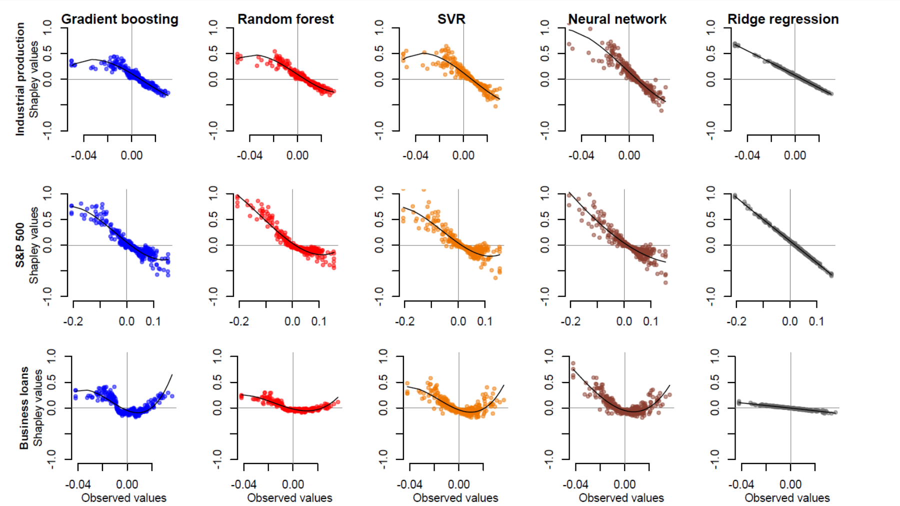
Step 2: Shapley value variable importance



Fraction of absolute feature Shapley values within test period 1990–2019 for all full-information models.

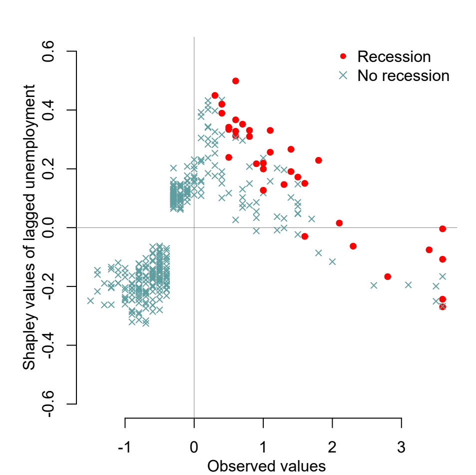
ML models largely agree on feature importances compared to linear Ridge regression.

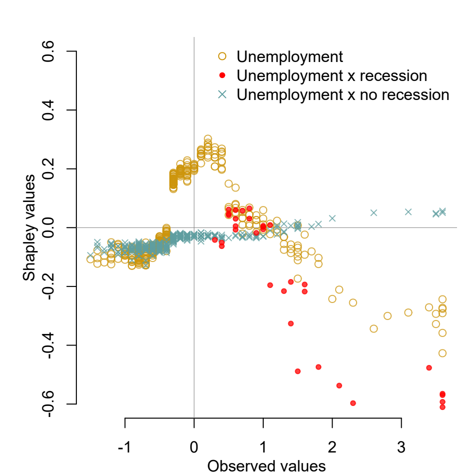
Step 2: Learned functional forms (I): Non-linearities



Lines shows a polynomial fit of Shapley values (dots). Source: Authors' calculations.

Step 2: Learned functional forms (III): Regime learning





Interaction between lagged unemployment and recessions (red) as learned by the boosted tree. LEFT: Baseline model. RIGHT: Unemployment-recession interaction with a recession dummy in the model. Source: Authors' calculations.

Step 3: Statistical inference and communication

	Gradient boosting			Ridge regression		
	β^S	p-value	Γ^S	β^S	p-value	Γ^S
Industrial production	1.132	0.000	-0.217***	2.280	0.000	-0.185***
S&P 500	0.942	0.000	-0.191***	0.907	0.000	-0.317***
Consumption	1.103	0.000	-0.177***	0.966	0.012	-0.173**
Unemployment	1.443	0.000	+0.175***	9.789	0.000	+0.031***
Business loans	3.086	0.000	-0.066***	5.615	0.006	-0.035***
3-month treasury bill	4.273	0.000	-0.062***	-6.816	1.000	-0.042
Personal income	-0.394	0.682	+0.04	-0.658	0.870	+0.138
Oil price	0.298	0.387	-0.035	-2.256	0.973	-0.055
CPI	0.272	0.438	+0.021	-4.294	0.875	+0.014
M2 Money	-8.468	1.000	-0.016	-18.545	0.994	-0.009

Shapley regression of gradient boosting mode (left) and the ridge regression (right) for the forecasting predictions between 1990–2019. Significance levels: *p<0.1; **p<0.05; ***p<0.01. Source: Authors' calculations.

Take-away messages

- We propose an interpretable ML workflow
  1. Model test evaluation ("horse race")
  2. Shapley decomposition of individual predictions
  3. Shapley regression for statistical inference
- Perform macro forecasting exercise of US unemployment
- ML models ...
  - outperform conventional ones (step 1)
  - learn nuanced, meaningful and stable functional forms, which e.g. allow to identify different points in the business cycle (step 2)
  - distinguish signal from noise variables for different settings and qualify the state of convergence of the learning process (step 3)

⇒ Approach opens the door to more ML applications.

Thanks for listening

contact: andreas.joseph@bankofengland.co.uk.

Model inference summary (Joseph (2020))

Assume fitted (i.e. trained) model and Shapley decomposition.

1. Shapley regression [Eq. 5] with appropriate standard errors.
2. Assessment of model bias and component robustness based on $\hat{\beta}^S$ over region Ω:

   Robustness: $\mathcal{H}_0^c : \{\hat{\beta}_c^S = 0|\Omega\}$ rejected and $\mathcal{H}_1^c : \{\hat{\beta}_c^S = 1|\Omega\}$ not rejected for individual components

   Unbiasedness: $\mathcal{H}_1^c : \{\hat{\beta}_c^S = 1|\Omega\}$ not rejected $\forall c \in \{1, \ldots, C\}$, or inclusion condition
3. Calculate Shapley share coefficients (SSC) $\Gamma^S(\hat{f}, \Omega)$ [Eq. 8] and their standard errors

Pros & Cons of ML relative to econometric approach [revisited]

Advantages

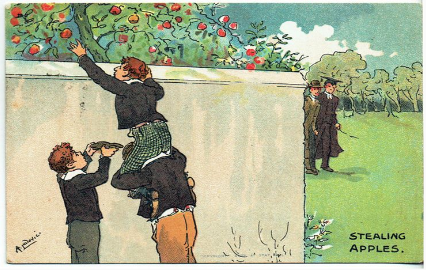
- Often higher accuracy
  Initial motivation (step 1)
- Lower risk of misspecification
  SR distinguishes signal from noise (step 3)
- Return richer information set
  Learned functional forms (step 2)

Disadvantages

- Higher model complexity ("black box critique")
  Learned functional forms (step 2)
- Less analytical guarantees, e.g. risk of overfitting
- Often larger data requirement
  SR tracks learning (step 3)

Intuitive Shapley value example: the Victorian bad boys

- Three siblings (strong [S], tall [T] & smart [M]) set off to nick some apples A (pay-off) from the neighbour's tree
- For each sibling, sum over marginal contribution to coalitions of one and two
- So, the Shapley value of the strong sibling [S] is then:



Source: 6oxgangsavenueedinburgh

$$\phi_S = \frac{1}{6}[A(S) - A(\emptyset)] + \frac{1}{6}[A(T,S) - A(T)] + \frac{1}{6}[A(M,S) - A(M)] + \frac{1}{3}[A(T,M,S) - A(T,M)] \quad (9)$$

Back

Shapley-Taylor expansion (Agarwal et al., 2019)

The discrete set derivative of model $\hat{f}$ at point x_i with respect to the set of variables x' conditioned on active variables x'' with $x'' \subseteq x \setminus x'$ is

$$\delta_{x'} \hat{f}(x_i|x'') \equiv \sum_{x''' \subseteq x'} (-1)^{|x'''|-|x'|} \, \hat{f}(x_i|x''' \cup x''), \tag{10}$$

with x', x'' and $x''' \subseteq \mathcal{C}(x)$. The case $|x'| = 1$ corresponds to (14). Let $h \leq n$,

$$\mathcal{T}_h^S\left(\hat{f}, x_i \,|\, x'\right) = \begin{cases} \delta_{x'} \hat{f}(x_i \,|\, \emptyset) & \text{if} \quad |x'| < h\,, \\ \frac{h}{n} \sum_{x'' \subseteq \mathcal{C}(x) \setminus x'} \frac{\delta_{x'} \hat{f}(x_i \,|\, x'')}{\binom{n-1}{|x''|}} & \text{if} \quad |x'| = h\,. \end{cases} \tag{11}$$

The efficiency statement takes the form

$$\hat{f}(x_i) = \phi_0^S \quad + \sum_{x' \subseteq \mathcal{C}(x),\, |x'| \leq h} \mathcal{T}_h^S\left(\hat{f}, x_i \,|\, x'\right). \tag{12}$$

This allows us to separate single-variable effects and interactions.

SR properties (proofs in Joseph (2020))

- SR identical to LR in case of LR (reassuringly the wheel was not reinvented)
- Inference only strictly valid locally within input region Ω (non-linearity of ML models)
- $\beta^S \in \{0,1\}^m$ only possible true values, corresponding to the "no-signal" ($\mathcal{H}_0^k$) or "signal" ($\mathcal{H}_1^k$) cases, respectively
- SR coefficients $\hat{\beta}^S$ gauge the learning process of $\hat{f}$:
  - $\mathcal{H}_0^k$ rejected: useful information contained x_k
  - And $\mathcal{H}_1^k$ *not* rejected: x_k robustly learned (perfect alignment, asymptotic limit)
  - Generally, $\hat{\beta}_k^S > / < 1$ measure under/over-reliance on x_k, respectively
- SR allow to control for different error structures within ML models.
- SR coefficients $\hat{\beta}^S$ not really useful for communication (no scale information).

Robustness analysis of horse race (part of it)

	Gradient boosting	SVR	Random forest	Neural Network	Ridge regression	AR_1
Prediction horizon h (lag between response and predictors in months)						
1	0.20	0.19	**0.17**	0.18	0.18	**0.17**
3	0.28	0.28	**0.27**	**0.27**	**0.27**	**0.27**
6	0.41	0.41	**0.39**	0.42	0.43	0.41
12 (baseline)	**0.56**	0.57	0.58	0.59	0.64	0.61
24	0.68	0.67	**0.62**	0.69	0.73	0.79
36	0.64	0.63	**0.61**	0.72	0.72	0.80
Training set size (in months)						
60	0.83	0.87	**0.79**	0.84	0.87	0.95
120	0.63	0.67	**0.57**	0.66	0.66	0.71
240	0.58	**0.56**	0.57	0.58	0.61	0.67
360	**0.57**	0.58	0.58	0.60	0.61	0.64
480	**0.56**	0.57	0.57	0.57	0.63	0.62
max (baseline)	**0.56**	0.57	0.58	0.59	0.64	0.61
Transformation span l (in months)						
1	0.57	0.60	**0.55**	0.59	0.64	-
3 (baseline)	**0.56**	0.57	0.58	0.59	0.64	-
6	**0.60**	**0.60**	**0.60**	0.67	0.66	-
9	**0.65**	0.68	0.67	0.70	0.70	-
12	0.68	0.74	0.70	0.71	0.74	**0.61**
Winsorisation at 1% and 99%						
Yes (baseline)	**0.56**	0.57	0.58	0.59	0.64	0.61
No	**0.56**	0.59	0.58	0.60	0.64	0.61

MAE for forecasting US unemployment one year out. Source: Author's calculations.

Experts vs 'robots' (I)

We (experts) hand-picked inputs, BUT should we not let data and algorithms speak freely?

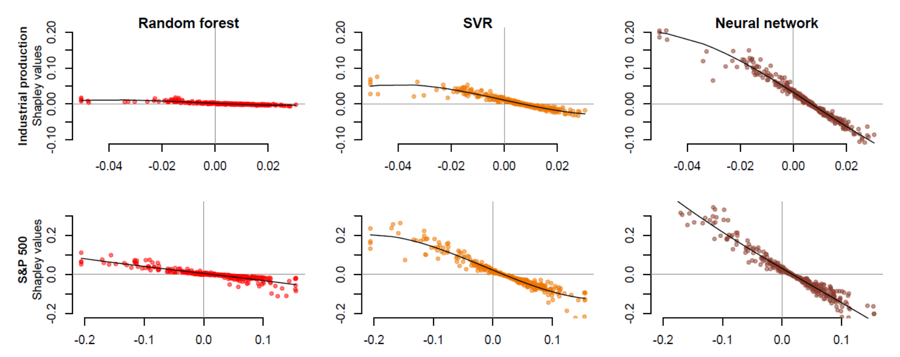
	Key features	All features	PCA_1	PCA_2	PCA_3	PCA_5	PCA_7
Gradient boosting	0.56	0.58	0.67	0.53	**0.52**	0.54	0.57
SVR	0.57	0.57	0.61	**0.52**	**0.52**	0.55	0.59
Random forest	0.58	0.55	0.62	**0.52**	0.53	0.55	0.61
Neural network	0.59	0.57	0.69	**0.52**	0.53	0.55	0.55
Lasso	0.64	0.63	0.65	0.56	**0.54**	0.56	0.59
Ridge	0.64	0.58	0.65	0.56	**0.54**	0.56	0.58
OLS	0.65	0.80	0.65	0.56	**0.54**	0.56	0.59

Comparison of the forecasting performance (MAE) when using different input data. Source: Authors' calculations.

Yes, to some extent.

Experts vs 'robots' (II)

BUT black box problem returns: No consistent signal anymore.

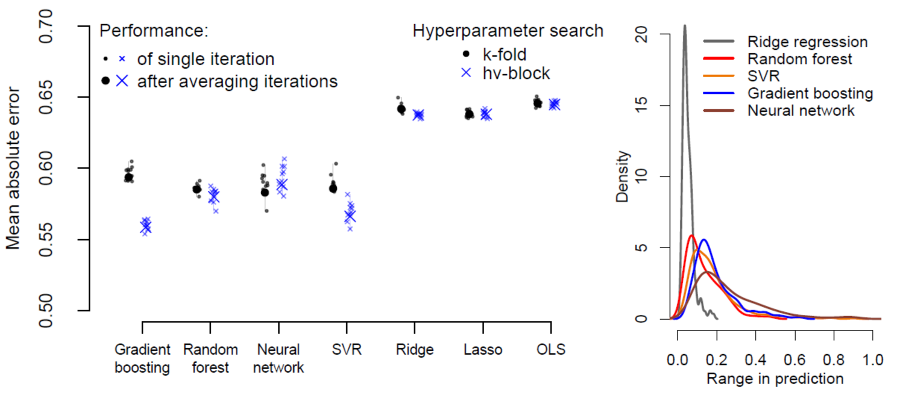


Learned functional forms. Lines shows a polynomial fit of Shapley values (dots). Source: Authors' calculations.

⇒ Combination of experts and robots best (complements).

Prediction & performance robustness

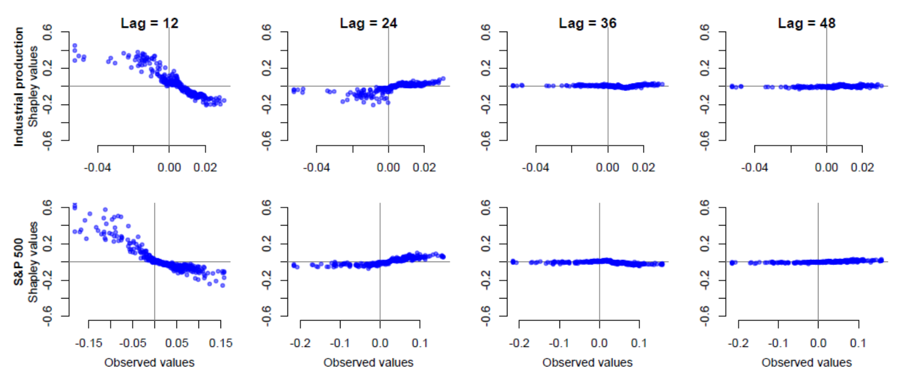
ML models better overall, BUT show more variability.



Left: model performance by choice. Right: Predictive ranges. Source: Authors' calculations.

The importance of lagged variables

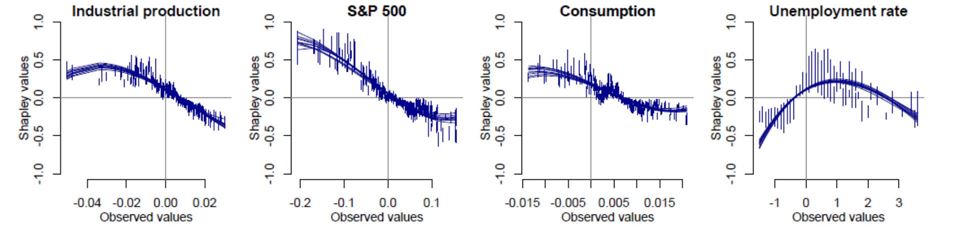
Higher-order lags make sub-leading contributions.



Shapley values for gradient boosting for different lag-length of inputs. Source: Authors' calculations.

Shapley values by random seed

Learned functional forms robust, BUT attributions in the tails can be unstable.



Shapley values for gradient boosting with different random seeds. Source: Authors' calculations.

Numerical calculation of Shapley component for a math. model

The Shapley value of a feature is the weighted sum of marginal contributions to all possible coalitions of other features (players):

$$\phi_k^S(\hat{f}, x_i) = \sum_{S \subseteq \mathcal{C} \setminus \{k\}} \frac{|S|!(n-|S|-1)!}{n!} \Big(\hat{f}(x_i|S \cup \{k\}) - \hat{f}(x_i|S) \Big) \qquad (13)$$

$$= \sum_{S \subseteq \mathcal{C} \setminus \{k\}} \omega_S \Big(\mathbb{E}_b[\hat{f}(x_i)|S \cup \{k\}] - \mathbb{E}_b[\hat{f}(x_i)|S] \Big) \qquad (14)$$

$$\text{with} \qquad \mathbb{E}_b[\hat{f}(x_i)|S] \equiv \int \hat{f}(x_i)\, \mathrm{d}b(\bar{S}) = \frac{1}{|b|} \sum_b \hat{f}(x_i|\bar{S}) \qquad (15)$$

"Excluded" features are integrated out over background b, which is an informative dataset determining ϕ_0. E.g. training dataset or sample of untreated population.

There are some challenges (and solutions) to the calculation of (1)–(3).

Challenges in calculating model Shapley values

- Computational complexity: Generally intractable for large feature sets ($n!$ in 1)

  ⇒ *Solutions:*

  - Coalition sampling
  - Feature grouping: important and 'others'
  - Model specific algorithms (e.g. Lundberg et al. (2018))

- Feature dependence: Equation 14 assumes independence

  ⇒ *Solutions:*

  - Use exact method for trees and compare
  - Calculate higher-order terms of Shapley-Taylor index (Agarwal et al., 2019) and compare relative magnitudes

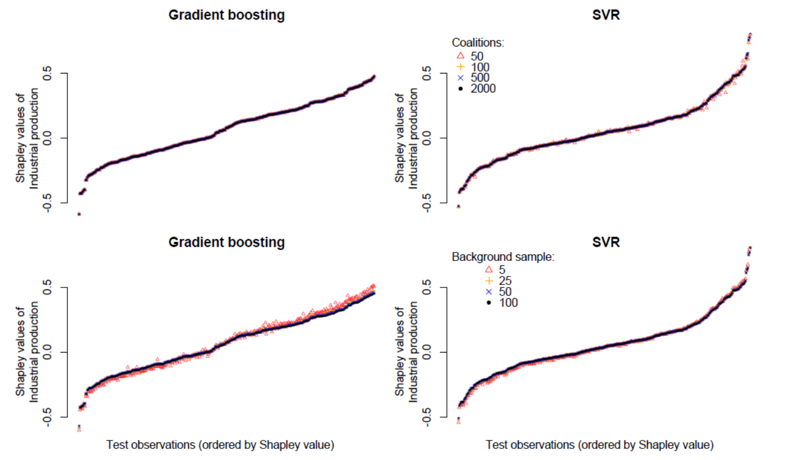
- Expectation consistency: Integration in (15) can break consistency

  ⇒ *Solutions:* When comparing models, their background values ϕ_0 need to coincide (or close). Mostly the case in practical applications. See Joseph (2020).

Back

Shapley value computation choices

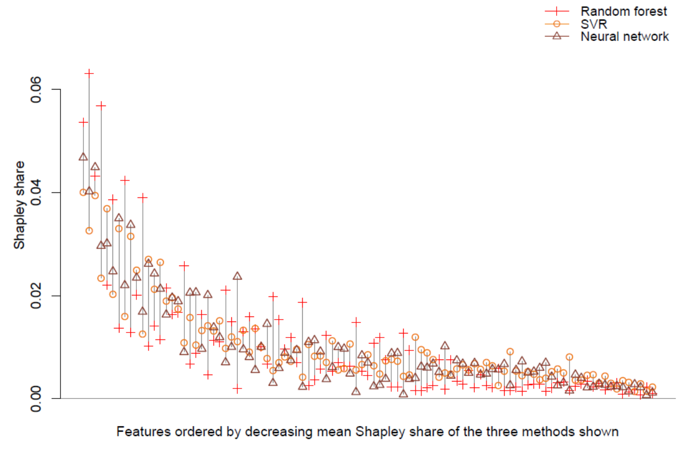
Using KernelShap (Lundberg and Lee, 2017) mostly precise for fast approximations.



Shapley values for gradient boosting with different computational choices. Source: Authors' calculations.

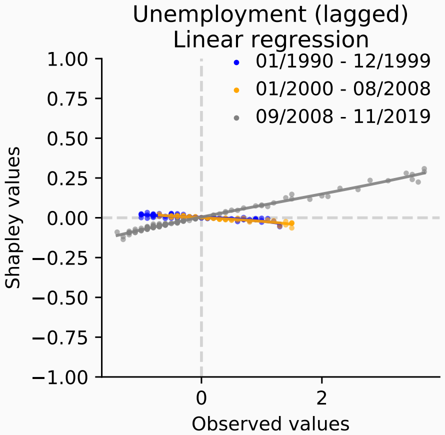
Shapley value range in high-dim setting

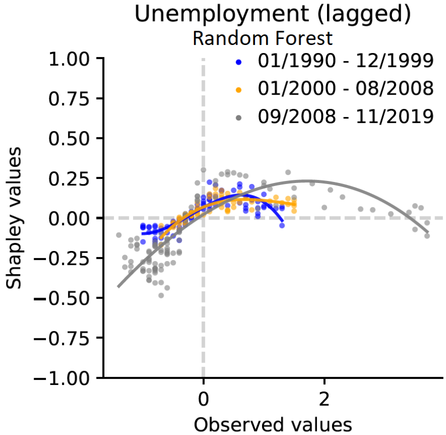
Using 'too many' features can create attribution instability.



Shapley share ranges for different models. Source: Authors' calculations.

Learned functional forms: Stability





Lines shows a polynomial fit of Shapley values (dots). Shapley values are computed on the out-of-bag predictions (look-ahead bias, but no model drift). Extreme values, below 2.5% and above 97.5% quantile, are excluded.

References i

Agarwal, A., Dhamdhere, K., and Sundararajan, M. (2019). A new interaction index inspired by the taylor series. *arXiv e-prints*, 1902.05622.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68.

Chernozhukov, V., Demirer, M., Duflo, E., and Fernandez-Val, I. (2023). Generic machine learning inference on heterogenous treatment effects in randomized experiments. *NBER Working Paper Series*, (24678).

Joseph, A. (2020). Parametric inference with universal function approximators. *arXiv preprint arXiv:1903.04209*.

Lundberg, S., Erion, G., and Lee, S. (2018). Consistent individualized feature attribution for tree ensembles. *ArXiv e-prints*, 1802.03888.

References ii

Lundberg, S. and Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*, pages 4765–4774.

McCracken, M. W. and Ng, S. (2016). FRED-MD: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, 34(4):574–589.

Pagan, A. (1984). Econometric issues in the analysis of regressions with generated regressors. *International Economic Review*, 25(1):221–47.

Shalev-Shwartz, S., Shamir, O., Srebro, N., and Sridharan, K. (2010). Learnability, stability and uniform convergence. *J. Mach. Learn. Res.*, 11:2635–2670.

Shapley, L. (1953). A value for n-person games. *Contributions to the Theory of Games*, 2:307–317.

Štrumbelj, E. and Kononenko, I. (2010). An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11:1–18.

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999.

References iii

Young, P. (1985). Monotonic solutions of cooperative games. *International Journal of Game Theory*, 14:65–72.