(Forecasting Inflation With) The Hedged Random Forest

Elliot Beck[1,2], Damian Kozbur[1], and Michael Wolf[1]

[1]University of Zurich

[2]Swiss National Bank

Outline

1. Motivation
2. Generic Methodology
3. Hedging the Random Forest
4. Empirical Application
5. Forecasting Inflation
6. Conclusions

Outline

Motivation

The random forest is one of the most popular and widely employed tools for supervised machine learning.

It can be used for both classification and regression tasks; in this paper, the focus will be on regression only.

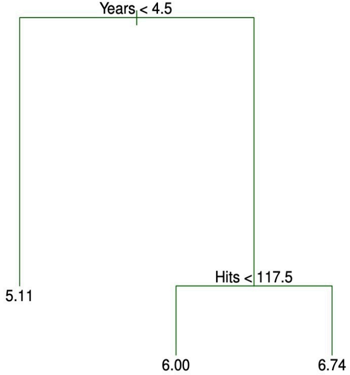
In its standard form, the crux of the random forest is to use an equal-weighted ensemble of (decorrelated) tree-based forecasts.

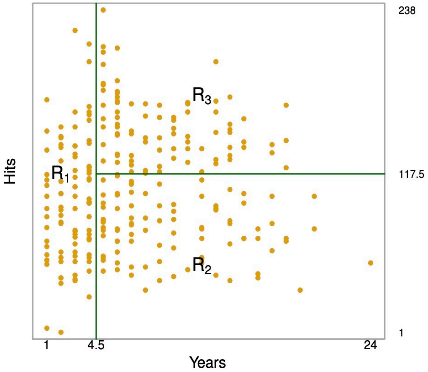
Instead, we suggest a more general weighting scheme that borrows certain ideas from the related problem of financial portfolio selection.

Example of a Regression Tree

Predict log-salary of a MLB baseball player based on experience and hits (in the previous season):

- Left: regression tree
- Right: partition of the sample space
- The predicted value is the average of the corresponding region





Crux of the Random Forest

Motivation:

- Individual trees have low bias but high variance
- If one had many trees, averaging them would reduce variance

Conundrum:

- How to construct different trees from a single dataset?

Solution:

- Create artificial datasets by resampling from the observed data with replacement (bootstrapping)
- Grow a tree on each of these artificial datasets
- For each tree, at each split, only use `mtry` $< d$ of the regressors
- Thus, individual trees are decorrelated and averaging pays off

Note: This is a form of bagging (bootstrap aggregation).

Recent Paper by Amazon Research

International Journal of Forecasting 38 (2022) 1473–1481

Contents lists available at ScienceDirect

International Journal of Forecasting

journal homepage: www.elsevier.com/locate/ijforecast

Forecasting with trees

Tim Januschowski [a,*], Yuyang Wang [a], Kari Torkkola [a], Timo Erkkilä [b], Hilaf Hasson [a], Jan Gasthaus [a]

[a] *Amazon Research, Charlottenstr. 4 10969, Berlin, Germany*
[b] *Veracell Oy, Polttimonkatu 4, Tampere, Finland*

A R T I C L E I N F O

Keywords:
Random forests
Probabilistic forecasting
Gradient Boosted Trees
Global forecasting models
Deep Learning

A B S T R A C T

The prevalence of approaches based on gradient boosted trees among the top contestants in the M5 competition is potentially the most eye-catching result. Tree-based methods out-shone other solutions, in particular deep learning-based solutions. The winners in both tracks of the M5 competition heavily relied on them. This prevalence is even more remarkable given the dominance of other methods in the literature and the M4 competition. This article tries to explain why tree-based methods were so widely used in the M5 competition. We see possibilities for future improvements of tree-based models and then distill some learnings for other approaches, including but not limited to neural networks.

Some Quotes From This Paper

The M5 competition has re-affirmed that tree-based methods and gradient-boosted trees belong to the toolbox of forecasters.

We hope that the M5 competition will spark someresearch into model improvements for tree-based forecasting methods. We believe that meaningful changes in accuracy will come mainly from more sweeping changes to the core models. In this sense, we reject the notion that expert forecasters will become less needed. On the contrary, we believe that a lot of model innovation is still needed.

Outline

Setting and Criterion

Setting:

- The goal is to forecast (or predict) a random variable $y \in \mathbb{R}$
- Available is a set of features (or regressors) $x \in \mathbb{R}^d$
- A generic forecast is denoted by $\hat{f}$
- The criterion is the mean-squared error (MSE):

$$\mathrm{MSE}(\hat{f}) := \mathbb{E}[y - \hat{f}(x)]^2$$

Let

- $\mathrm{Bias}(\hat{f}) := \mathbb{E}[y - \hat{f}(x)]$ and
- $\mathrm{Var}(\hat{f}) := \mathbb{V}\mathrm{ar}[y - \hat{f}(x)] = \mathbb{E}[(y - \hat{f}(x)]^2 - \mathrm{Bias}^2(\hat{f})$

MSE Decomposition

$$\mathrm{MSE}(\hat{f}) = \mathrm{Bias}^2(\hat{f}) + \mathrm{Var}(\hat{f})$$

Note: The oracle minimizing the MSE is $\hat{f}_{\mathrm{or}}(x) := \mathbb{E}(y|x)$.

Generic Framework $\supseteq$ Random Forest

Forecast-combination problem:

- Have ensemble of p forecasting methods, denoted by $\{\mathcal{M}_j\}_{j=1}^p$ (which may involve fitted parameters)
- "Hard to beat" is the equal-weighted forecast

$$\hat{f}_{\mathrm{EW}}(x) := \frac{1}{p} \sum_{j=1}^{p} \mathcal{M}_j(x)$$

- Interested in weighted forecasts of the kind

$$\hat{f}_w(x) := \sum_{j=1}^{p} w_j \mathcal{M}_j(x) \quad \text{with} \quad \sum_{j=1}^{p} w_j = 1$$

where negative weights are allowed

The Oracle Solution

Vector of forecast errors $e := (e_1, \dots, e_p)'$, with $e_j := y - \mathcal{M}_j(x)$, has

$$\mu := \mathbb{E}(e) \quad \text{and} \quad \Sigma := \mathbb{V}\text{ar}(e)$$

The MSE of a weighted forecast $\hat{f}_w$ is then given by

$$\text{MSE}(\hat{f}_w) = (w'\mu)^2 + w'\Sigma w$$

The oracle forecast combination is thus the solution to

$$\begin{aligned} &\min_w \; (w'\mu)^2 + w'\Sigma w \\ &\text{s.t.} \quad w'\mathbf{1} = 1 \end{aligned}$$

The Feasible Solution

Based on training data, compute estimators $\hat{\mu}$ and $\hat{\Sigma}$.

A feasible forecast combination is thus the solution to

$$\min_{w} \, (w'\hat{\mu})^2 + w'\hat{\Sigma}w$$
$$\text{s.t.} \quad w'\mathbf{1} = 1 \quad \text{and}$$
$$\|w\|_1 \leq \kappa$$

The final constraint with $\kappa \in [1, \infty]$ is a gross-exposure constraint:

- Motivated by the related problem of financial portfolio selection
- $\kappa = 1$ enforces non-negative weights ("no short-selling")
- $\kappa = \infty$ corresponds to dropping the constraint
- Values $\kappa \in [1.5, 2.5]$ tend to work well in practice

Results in a hedged forecast combination.

Asymptotic Optimality Theory

Finite-Sample Guarantees

Outline

Details

Estimation of μ and Σ:

- Training dataset $\{v_i\}_{i=1}^n$ with $v_i := (y_i, x_i')'$
- We only consider i.i.d. data for now
- Train $p = 500$ trees using the "ranger library" (standard inputs); we use $\texttt{mtry} := \lfloor d/3 \rfloor$ instead of $\texttt{mtry} := \lfloor \sqrt{d} \rfloor$
- Extract the forecasts of each tree $\mathcal{M}_j$ on the entire training set $\Longrightarrow$ residual matrix R of size $n \times p$
- $\hat{\mu}$ is the (column-wise) sample mean of R
- $\hat{\Sigma}$ is the nonlinear shrinkage estimator of Ledoit and Wolf (2022); but can also consider the sample covariance matrix

The default choice for the gross-exposure constraint is $\kappa = 2$.

Possible Concerns

"Using residuals, that is, in-sample errors leads to an underestimation (in magnitude) of μ and Σ."

True, but the feasible solution $\hat{w}$ does not change if the inputs $(\hat{\mu}, \hat{\Sigma})$ are replaced by $(c\hat{\mu}, c^2\hat{\Sigma})$ for any $c > 0 \Longrightarrow$ scale invariance.

"To estimate μ and Σ one should only use out-of-bag (OOB) residuals."

This could be done to estimate μ but not to estimate Σ,
since nonlinear shrinkage needs a 'full' rectangular matrix R.

Also, scale invariance alleviates this concern.

Outline

Data

Use the 17 numerical benchmark datasets of Grinsztajn et al. (2022).

Accessible on the official openML website `www.openml.org`, which also offers comprehensive metadata and descriptions.

For large datasets, we follow the approach of Grinsztajn et al. (2022) by selecting 10,000 observations at random.

Datasets Used

Name	# Observations	# Features
ailerons	13,750	734
Bike_Sharing_Demand	17,379	12
brazilian_houses	10,692	10
cpu_act	8,192	22
diamonds	53,940	10
elevators	16,599	12
house_16H	22,784	17
house_sales	21,613	18
houses	20,640	9
isolet	7,797	614
medical_charges	163,065	4
MiamiHousing2016	13,932	14
nyc-taxi-green-dec-2016	581,835	19
pol	15,000	49
sulfur	10,081	7
superconduct	21,263	80
wine_quality	6,497	12

Performance Evaluation

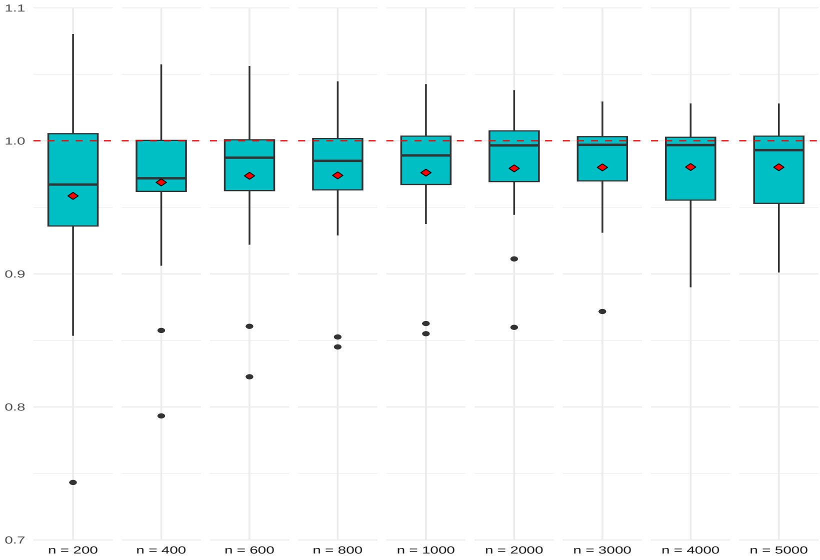
Prescription:

- Partition a dataset into a training set and a test set by selecting n observations at random as the training set
- Any forecasting method then yields a MSE on the test set
- Repeat this process $B = 100$ times
- Obtain the following RMSE ratio:

$$\mathrm{RMSE}_{\mathrm{HRF/RF}} := \frac{\sqrt{\frac{1}{B}\sum_{b=1}^{B} \mathrm{MSE}_{\mathrm{HRF,b}}}}{\sqrt{\frac{1}{B}\sum_{b=1}^{B} \mathrm{MSE}_{\mathrm{RF,b}}}}$$

- Get 17 such ratios for each $n \Longrightarrow$ boxplot
- Use $n \in \{200, 400, 600, 800, 1000, 2000, 3000, 4000, 5000\}$
- Line up the corresponding boxplots horizontally

HRF vs. RF


1.1
1.0
0.9
0.8
0.7
n = 200
n = 400
n = 600
n = 800
n = 1000
n = 2000
n = 3000
n = 4000
n = 5000

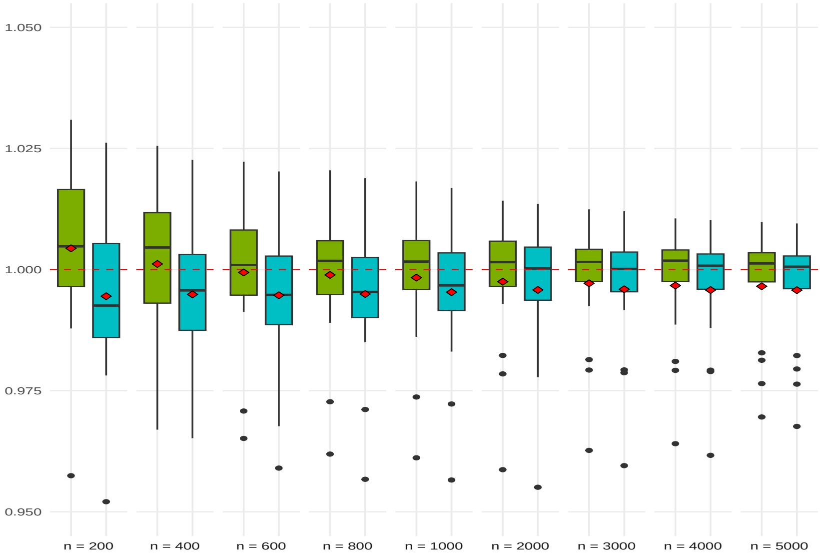
Negative Weights and Shrinkage

Look at

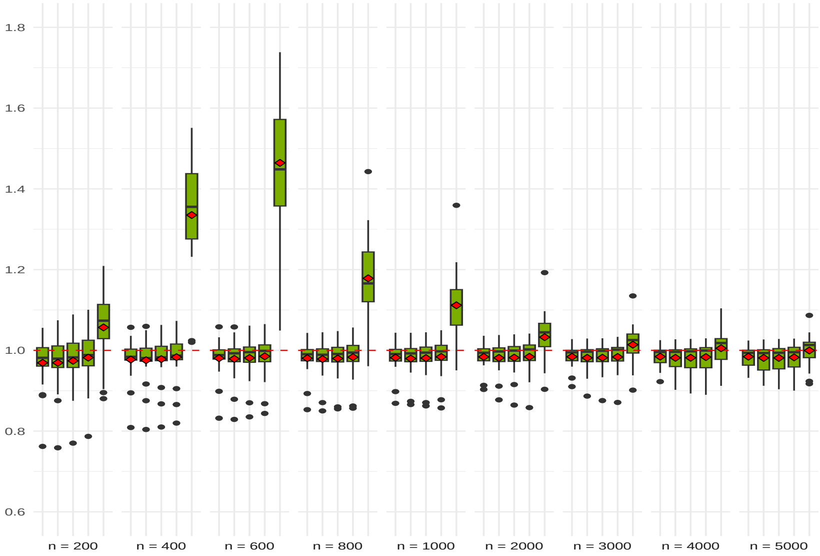
$$\mathrm{RMSE}_{\kappa=2/\kappa=1} := \frac{\sqrt{\frac{1}{B}\sum_{b=1}^{B} \mathrm{MSE}_{\mathrm{HRF,b},\kappa\,=\,2}}}{\sqrt{\frac{1}{B}\sum_{b=1}^{B} \mathrm{MSE}_{\mathrm{HRF,b},\kappa\,=\,1}}}$$

for $\hat{\Sigma} \in \{$Sample Covariance matrix, Nonlinear Shrinkage$\}$.

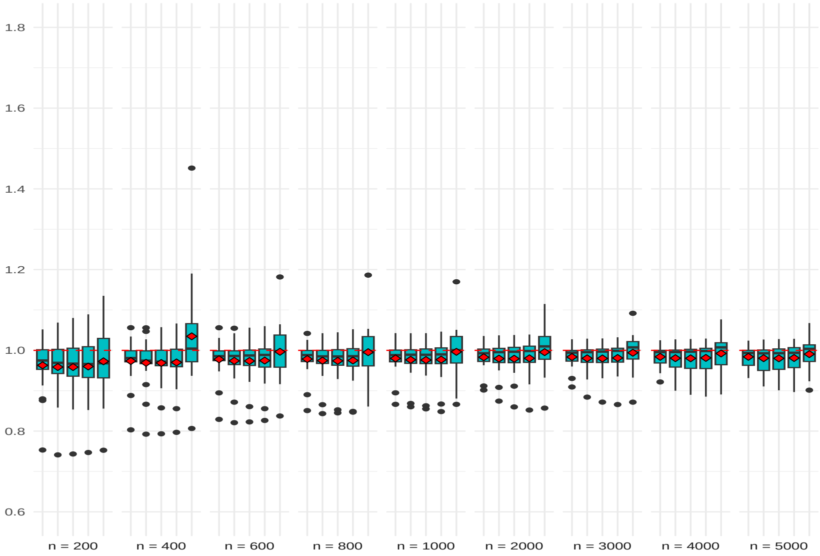
$\kappa = 2$ vs. $\kappa = 1$ for Sample and NL Shrinkage



$\kappa \in \{1, 1.5, 2, 2.5, \infty\}$ for Sample



$\kappa \in \{1, 1.5, 2, 2.5, \infty\}$ for NL Shrinkage



Comparison with Alternative Proposals

Two alternative proposals for a weighted random forest:

- Winham et al. (2013)
- Pham and Olfsson (2020)

Proposed for classification, but can be adapted to regression.

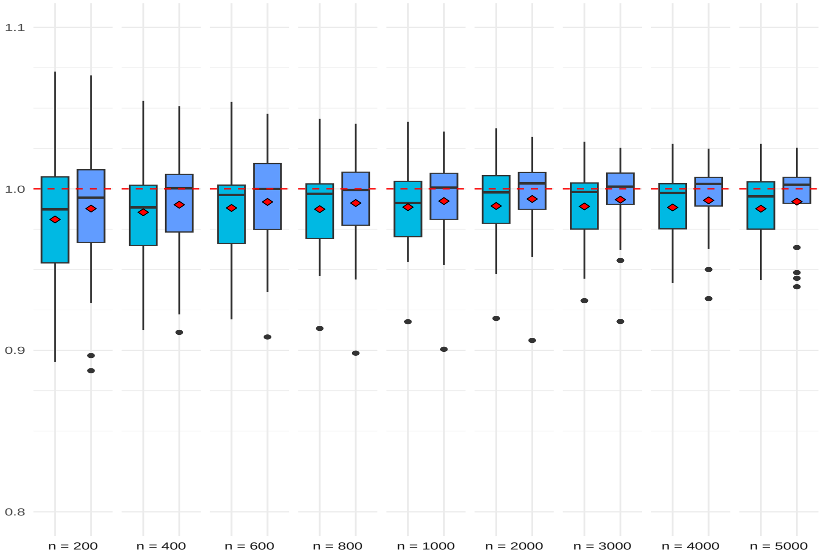
Common characteristics:

- Only use out-of-bag (OOB) residuals
- Rank performances of individual trees
- Neglect "covariances"
- All weights are strictly positive

Look at

$$\mathrm{RMSE}_{\mathrm{HRF/AP}} := \frac{\sqrt{\frac{1}{B}\sum_{b=1}^{B}\mathrm{MSE}_{\mathrm{HRF,b}}}}{\sqrt{\frac{1}{B}\sum_{b=1}^{B}\mathrm{MSE}_{\mathrm{AP,b}}}}$$

Comparison with Alternative Proposals



Comparison with Alternative Proposals

Chen et al. (2024, JMLR) propose an “optimal” weighted random forest, where the weights are also non-negative.

Key feature:

- Method uses shallow trees, unlike most of the RF literature
- Needed to prove asymptotic optimality theory

Problem:

- Method cannot be used on many datasets,
  since numerical optimizer fails to converge reliably
- Hence, cannot include it in our comparisons above

Solution:

- Include HRF in the comparisons of Chen et al. (2024, JMLR)
- This amounts to giving them the “home field advantage”
- Call their method OWRF

Comparison with Alternative Proposals

Chen et al. (2024, JMLR) use 12 datasets, selected somehow.

Details:

- Partition each dataset into training (70%) and test (30%) set
- Repeate this process $B = 1,000$ times, then average results
- Two performance measures: MSE and MAE

Difference in our analysis:

- RF, WRF, and CRF use same the hyperparameters as HRF
- That is, they use deep trees as well
- This is how applied researchers would use the methods

MSE Results

Dataset	RF	HRF	OWRF	WRF	CRF
BH	11.66	10.26	12.01	11.00	11.09
servo	1.49	0.60	0.77	0.70	1.05
CCS	32.47	25.39	39.35	29.00	29.45
ASN	13.27	5.43	12.30	8.96	10.31
CCPP	11.56	11.14	14.85	11.31	11.22
CST	18.74	12.10	15.57	13.74	15.69
EE	2.33	1.88	2.60	2.09	2.21
PT	0.26	0.12	0.61	0.16	0.19
QSAR	1.25	1.28	1.34	1.23	1.24
SM	2.46	0.88	2.25	1.30	1.87
YH	17.73	1.14	2.54	1.68	5.53
tecator	4.19	2.63	3.28	2.65	3.09

MAE Results

Dataset	RF	HRF	OWRF	WRF	CRF
BH	2.25	2.16	2.36	2.20	2.21
servo	0.87	0.45	0.54	0.49	0.71
CCS	4.21	3.57	4.82	3.93	3.98
ASN	2.94	1.75	2.76	2.35	2.55
CCPP	2.47	2.40	2.92	2.44	2.42
CST	3.26	2.62	3.00	2.80	3.00
EE	0.97	0.83	1.05	0.90	0.94
PT	0.34	0.23	0.57	0.26	0.29
QSAR	0.82	0.83	0.85	0.81	0.81
SM	1.20	0.68	1.13	0.84	1.03
YH	2.61	0.54	0.97	0.70	1.47
tecator	1.45	1.09	1.31	1.18	1.27

Outline

The Challenge

The RF is (one of) the leading ML method for forecasting inflation; for example, see Medeiros et al. (2021, JBES).

No success so far in beating the equal-weighted ensemble.

Can the HRF do it?

The Setting

We largely follow the empirical study of Medeiros et al. (2021, JBES):

- Have monthly data going back to 01/1960
- Use rolling-window approach with a training set of 30 years
- Use a total of 464 features (which include lagged variables)
- (Pseudo) out-of-sample forecasting period: 01/1990–03/2025

Differences:

- Define inflation as yearly 'return' on price index:

$$\pi_t := \frac{P_t}{P_{t-12}} - 1$$

  which is the convention of central banks and policy makers
- Winsorize training data instead of using dummies for outliers

Price indexes:

- Four from the U.S.
- Two from Switzerland

Price Indexes

U.S.:

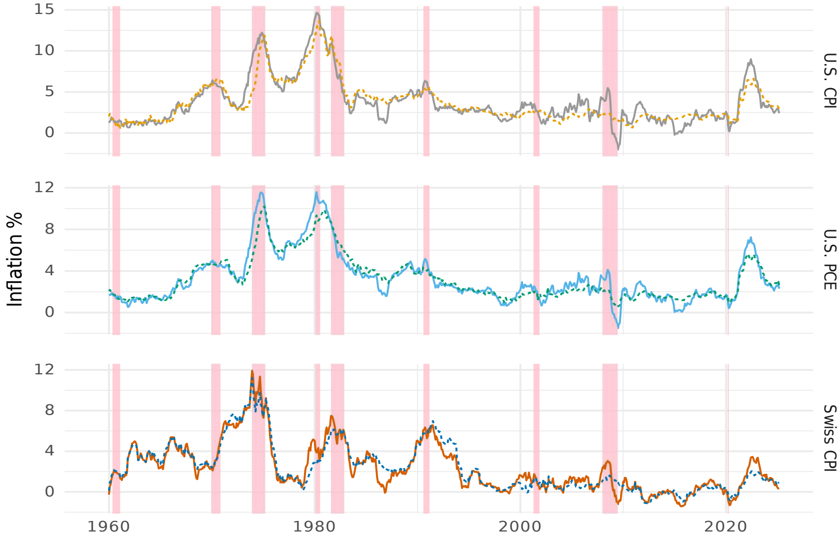
- CPIAUCSL: All Items in U.S. City Average
- CPILFESL: Ditto Less Food and Energy
- PCEPI: Personal Consump. Expenditure: Chain-type Price Index
- PCEPILFE: Ditto Less Food and Energy

Switzerland:

- CHECPIALLMINMEI: Total
- CHECPICORMINMEI: All Items Non-Food Non-Energy

Plots of Inflation Measures

Headline measures (solid lines) and core measures (dashed lines):



Estimation of μ and Σ

It is crucial to take the time-series nature of the data into account.

Start with EWMA estimation on the residual matrix R (exponentially-weighted moving average estimation):

- Decay parameter λ determines how fast the weights decline
- RiskMetrics recommends $\lambda = 0.06$ for daily data
- Since we have monthly data, we use $\lambda = 0.15$

Then use linear shrinkage:

- Shrink the estimator of μ to the common mean
- Shrink the estimator of Σ to the CVCC matrix (constant-variance-constant-covariance matrix)

Two Forecasting Approaches

One-shot approach:

- Build a single model for desired horizon h

Path-average approach:

- Forecast monthly changes for each intermediate horizon
- Then aggregate these forecasts

Consistent with the literature, path-average approach works better.

HRF Relative To RF

Panel (a): RMSE ratios

	Forecast horizon					
Target	1	3	6	9	12	Mean
CPIAUCSL	0.993	0.971	0.955	0.963	0.963	0.966
CPILFESL	0.983	0.955	0.934	0.926	0.924	0.940
PCEPI	0.995	0.964	0.946	0.952	0.959	0.958
PCEPILFE	0.972	0.958	0.953	0.938	0.937	0.950
CHECPIALL	0.990	0.981	0.966	0.961	0.950	0.968
CHECPICOR	0.958	0.932	0.913	0.906	0.901	0.918

Panel (b): MAE ratios

	Forecast horizon					
Target	1	3	6	9	12	Mean
CPIAUCSL	0.972	0.953	0.946	0.954	0.960	0.955
CPILFESL	0.969	0.920	0.906	0.903	0.904	0.914
PCEPI	0.979	0.949	0.933	0.941	0.952	0.946
PCEPILFE	0.974	0.947	0.935	0.922	0.927	0.937
CHECPIALL	0.990	0.971	0.964	0.941	0.933	0.958
CHECPICOR	0.953	0.939	0.902	0.901	0.900	0.916

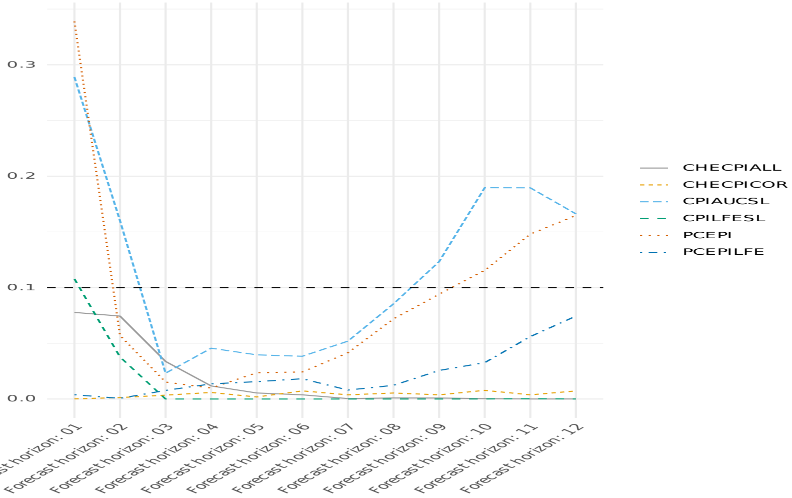
Addressing Statistical Significance

Is the outperformance of HRF over RF statistically significant?

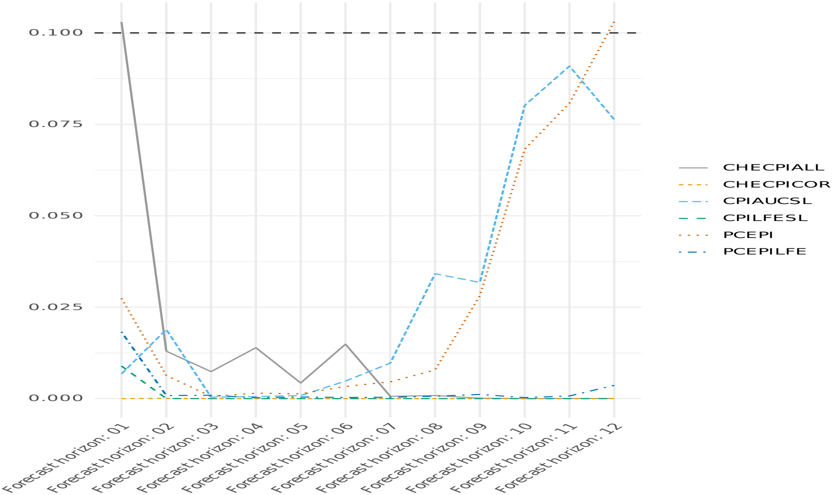
To this end we carry out a DM test for equal forecasting accuracy:

- Get a p-value for each scenario of (series, horizon, measure)
- Our p-values are (unashamedly) one-sided
- We use proper HAC standard errors and do not pretend that forecast errors are at most $(h-1)$-dependent

p-Values for Squared Errors



p-Values for Absolute Errors



Performance Over Time

For any target variable and any point in time s, compute
Cumulative Sum of Squared Error Differences

$$\text{CSSED}(s) := \sum_{t=1}^{s} \left(\hat{e}_{t,\text{HRF}}^2 - \hat{e}_{t,\text{RF}}^2 \right)$$

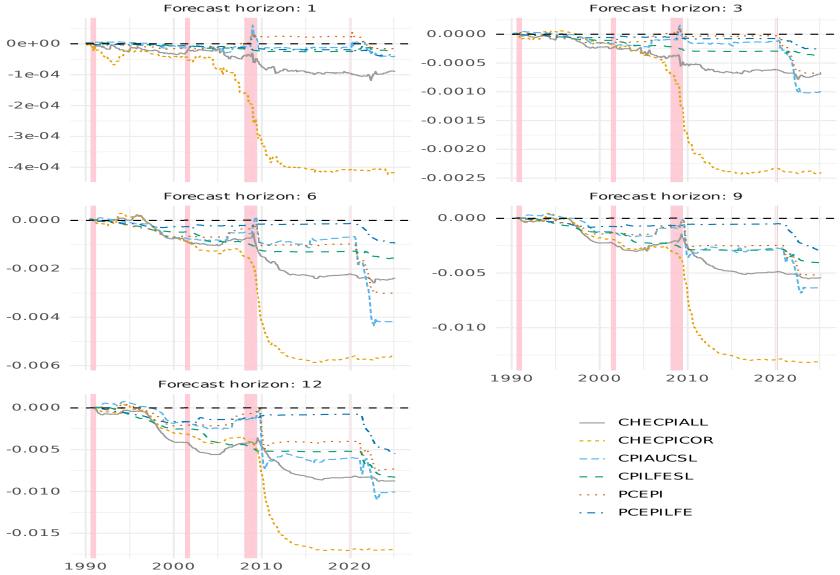
and Cumulative Sum of Absolute Error Differences

$$\text{CSAED}(s) := \sum_{t=1}^{s} \left(|\hat{e}_{t,\text{HRF}}| - |\hat{e}_{t,\text{RF}}| \right)$$

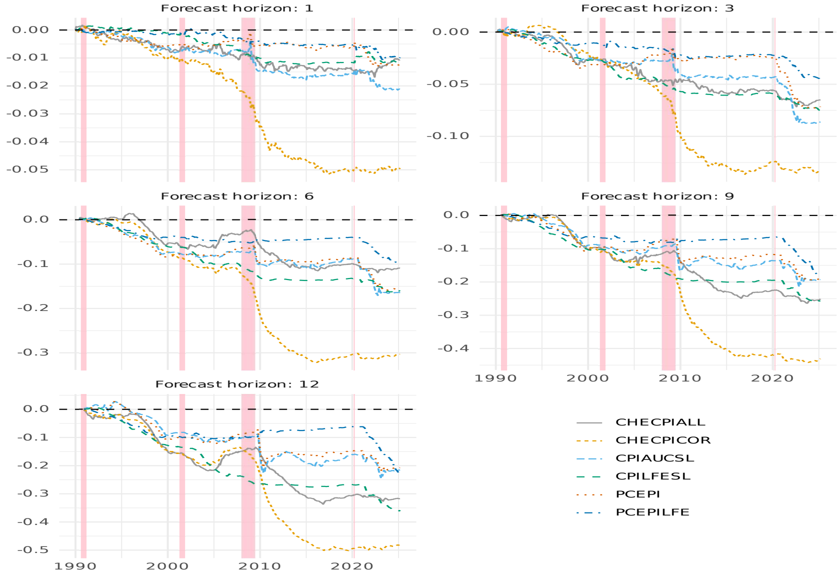
Then plot these series over time.

We would like to see trajectories that (mainly) lie below zero.

CSSED Trajectories



CSAED Trajectories



Outline

Conclusions

We have proposed a generic methodology to estimate optimal forecast combinations under the MSE criterion.

The methodology allows for negative weights but provides a hedge by imposing an upper bound on the sum of the absolute weights.

An important application is the random forest (tree-based forecasts).

The hedged random forest (HRF) outperforms the standard RF and three weighted competitors on 17 real-life benchmark datasets.

The hedged random forest (HRF) outperforms the standard RF for the task of forecasting inflation.

Future research:

- Causal HRF
- Quantile HRF
- HRF for classification

References