Learning Time-Varying Forecast Combinations
Antoine Mandel, Amir Sani

To cite this version:

HAL Id: halshs-01317974
https://halshs.archives-ouvertes.fr/halshs-01317974v2
Submitted on 16 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
LEARNING TIME-VARYING FORECAST COMBINATIONS

ANTOINE MANDEL AND AMIR SANI*

Combining forecasts has been demonstrated as a robust solution to noisy data, structural breaks, unstable forecasters and shifting environmental dynamics. In practice, sophisticated combination methods have failed to consistently outperform the mean over multiple horizons, pools of varying forecasters and different endogenous variables. This paper addresses the challenge to “develop methods better geared to the intermittent and evolving nature of predictive relations”, noted in Stock and Watson (2001), by proposing an adaptive nonparametric “meta” approach that provides a time-varying hedge against the performance of the mean for any selected forecast combination approach. This approach arguably solves the so-called “Forecast Combination Puzzle” using a meta-algorithm that adaptively hedges weights between the mean and a specific forecast combination algorithm or pool of forecasters augmented with one or more forecast combination algorithms. Theoretical performance bounds are reported and empirical performance is evaluated on the seven-country macroeconomic output and inflation dataset introduced in Stock and Watson (2001) as well as the Euro-area Survey of Professional Forecasters.

Keywords: Forecast Combinations, Forecast Combination Puzzle, Machine Learning, Econometrics.

1. INTRODUCTION

Macroeconomic forecasts provide crucial inputs to decision-makers addressing monetary and fiscal policy issues. Forecast accuracy depends on a selected model’s power to extract useful and meaningful information from available macroeconomic time series. Unfortunately, reality limits forecasting models to finite data samples, incomplete information sets and changing environmental dynamics that result in estimation error and model misspecification. In particular, macroeconomic time series
are predominantly composed of a limited number of noisy aggregated samples across varying economic conditions within an unstable forecasting environment. These challenges often result in inconsistent and misspecified models.

Introduced by Bates and Granger (1969), forecast combination methods have demonstrated an advantage in addressing noisy data, structural breaks, forecasters with inconsistent performance and changing environmental dynamics (Timmermann, 2006). Accordingly, a large body of research has focused on the theoretical and empirical development of complex forecast combination procedures that aim to fully exploit the information content within the available pool of forecasts (see Timmermann, 2006 for a recent survey). However, in empirical settings, these complex combination procedures generally fail to consistently outperform the simple mean (see e.g. Stock and Watson, 2004, for the case of output and inflation considered in this paper). The theory shows that existing forecast combination methods lack statistical power and subsequently overfit noise in the small number of macroeconomic samples that are generally available for modeling. In contrast, the mean manages this small sample noise consistently, and therefore well. This “Forecast Combination Puzzle” limits the ability to test new forecast combination approaches (especially in real-time settings), without risking underperformance to the mean.

Building on recent advances in online machine learning literature (in particular Sani et al., 2014), the aim of this paper is to propose a solution to the forecast combination puzzle through a meta-algorithm that provides an online hedge to the mean while exploiting the potential superior predictive ability provided by any alternative algorithm. We recast the forecast combination setting to the online (recursive) optimization setting of “Prediction with Expert Advice”, where we propose to learn time-varying “meta-weights” directly from the forecasting performance. Namely, the algorithm $\mathcal{AB}$-Prod, introduced in Sani et al. (2014), provides a meta-structure that combines the weights of a benchmark algorithm $\mathcal{B}$ (the mean in our context) and these of an alternative $\mathcal{A}$ in such a way that performance is never worse than a
precomputed constant to the mean while it learns any superior predictive ability of the alternative. Further, the rate at which this approach “learns” is close to optimal in both “easy” and “worst” case environments. This “meta”-algorithm also comes with theoretical guarantees that make no distribution assumptions on the process generating the target series or the losses and results in performance guaranteed for any stochastic, stationary, non-stationary or shifting environment.

Hence, the proposed algorithm proposes theoretical guarantees tailored to address the “Forecast Combination Puzzle” and provides a robust, data-driven procedure to real-time forecasting without the risks associated to testing new algorithms: it allows decision-makers to use novel forecasters in real-time environments while maintaining an hedge to the mean. We illustrate the empirical performance of this approach by showing it systematically outperform the mean in the seven-country output and inflation dataset introduced in Stock and Watson (2004) as well as in the Euro-area output surveys collected in the survey of professional forecasters (SPF).

The paper proceeds as follows. In Section 2, we briefly review the relevant forecast combination and machine learning literature. In Section 3, we propose a theoretically guaranteed forecast combination approach that “hedges” performance against the mean with synthetic results. In Section 4, we illustrate the workings of these algorithms with synthetic data. In Section 5, we demonstrate the performance of our approach for the forecast of output and inflation in the framework of Stock and Watson (2004) and Euro-area SPF. Section 6 concludes.

2. LITERATURE REVIEW

Real macroeconomic data is observed at an aggregate level and often composed of a small sample of time series observations. Traditional least-squares forecasters often fail to forecast such series due to the limited number of samples, noise and model misspecification (Timmermann 2006; Diebold 1989; Diebold and Pauly 1990; Hendry and Clements 2004). Additionally, macroeconomic models often depend on the configuration of shocks hitting the economy, policy regimes and other
institutional factors. This unstable real-world environment results in inconsistent forecasters.

Forecast combination approaches offer a simple procedure for exploiting the information content of candidate forecasters, while ignoring the need for explicit model selection\(^1\). Theoretical results from the literature demonstrate that gains achieved through combination weights are caused by this forecast model instability, where increasing instability in individual forecasts results in a larger advantage (see e.g. Hendry and Clements (2004); Diebold and Pauly (1987); Clements and Hendry (1998, 1999, 2006); Pesaran and Timmermann (2005); Timmermann (2006); Aiolfi et al. (2010)). More specifically, they provide a robust solution to small sample sizes, noise, regime shifts, model misspecification, diverse information sets, unstable forecasters and provide an efficient way to improve forecasting performance by diversifying over a pool of forecasts (for a survey, see Diebold and Lopez (1996); Newbold and Harvey (2002); Clements et al. (2002); Clemen (1989); Timmermann (2006); Huang and Lee (2010).). Practical successes in the forecast combination literature include output and inflation (Stock and Watson, 2001), interest rates (Guidolin and Timmermann, 2009), money supply (Granziera et al., 2013), monetary policy (Kapetanios et al., 2008), equity premiums (Rapach et al., 2010), commodities (Chen et al., 2008) and realized volatility (Patton and Sheppard, 2009).

Though forecast combination approaches have demonstrated several successes, theoretical results have not resulted in methods that consistently outperform the simple average. Timmermann (2006); Hsiao and Wan (2014) illustrate the specific conditions where the relative gain from the true ex-post optimal weights over an unbiased mean combination are negligible. With regard to out-of-sample performance, Huang and Lee (2010) showed several simple cases where the mean combination approach even outperforms a linear model set to the data generating process.

\(^1\)This work deals with a finite pool of candidate forecasters. Other works address the case of a very large to infinite pools of forecasters. See Elliott et al. (2013, 2015); Uematsu and Tanaka (2015).
This inability to consistently outperform the mean is referred to as the “Forecast Combination Puzzle” and has been explained as the biased weighting of “optimal” weights due to the low predictive content of candidate forecasts (Huang and Lee, 2010). This underperformance to the mean is further explained as the result of finite sample bias, model misspecification, unobserved variables, noise and changes in the underlying process (Stock and Watson, 2001, 2004; Claeskens et al., 2014; Smith and Wallis, 2005, 2009; Clark and McCracken, 2009; Huang and Lee, 2010). Empirical and theoretical results demonstrate no consistent advantage in alternative means (geometric, trimmed, corrected) or the median over different horizons and endogenous variables (Stock and Watson, 2004). One intuition is that these alternatives tend to smooth the forecast density generated by the pool of forecasters in a way that ignores important information. It’s also reasonable to assume that these alternatives bias parts of the forecast density without any consideration for their performance over time.

Forecasts are likely to have varying performance due to changing predictive content in the data, shifting regimes, the choice of tuning methods and the addition or removal of model parameters. Given these temporal dynamics, the unbiased weights of the mean may not always provide the best performance. In fact, time-varying combination weights have demonstrated great potential versus the mean (See e.g. Novales and de Fruto (1997); Hoogerheide et al. (2010); LeSage and Magura (1992); Granger and Ramanathan (1984); Yang (2004); LeSage and Magura (1992); Sessions and Chatterjee (1989); Sánchez (2008); Sancetta (2010); Timmermann (2006)). Unfortunately, assumptions underlying many of these time-varying approaches are often too restrictive to be applied in realistic empirical settings. In particular, many of these time-varying combination approaches assume a model on the temporal dynamics, a known or stationary covariance structure (Yang, 2004; Sancetta, 2010; Granger and Ramanathan, 1984; Sessions and Chatterjee, 1989; LeSage and Magura, 1992) or normality conditions on the residuals (see Claeskens et al. (2014), resulting in
inconsistent performance in real-time data environments.

The machine learning literature on “prediction with expert advice” (Cesa-Bianchi and Lugosi, 2006) provides a complementary perspective. It measures via the notion of regret, the efficiency with which an algorithm can learn from the data and provides distribution-free theoretical guarantees on the performance of these algorithms. Of particular concern for our work are the contributions of Even-Dar et al. (2008) and Sani et al. (2014) that provide dual theoretical guarantees to a benchmark and to the best forecaster in hindsight. By setting the mean as a benchmark, we can leverage on these results to provide an answer to the forecast combination puzzle. Namely, we can guarantee an online hedge to the mean while exploiting the potential superior predictive ability provided by any alternative algorithm.

3. THEORETICAL RESULTS

The forecast combination problem consists in a setting where a decision-maker has $K$ forecasts of a real variable of interest at his disposal and aims at aggregating the information contained in the pool of forecasts. More precisely, at a sequence of dates $t = h + 1 \cdots T$, the decision-maker has forecasts $(\hat{y}_{1,t|t-h}, \ldots, \hat{y}_{K,t|t-h}) \in \mathbb{R}^K$ available. These $K$ forecasts are then aggregated into a point forecast at time $t$, $\hat{y}_{t|t-h} \in \mathbb{R}$, for horizon $h$. In line with the bulk of the forecast combination literature (Timmermann, 2006) and the “Prediction with Expert Advice” framework of Cesa-Bianchi and Lugosi (2006), we shall restrict attention to convex forecast combinations where the decision-maker chooses decision weights $w_{t|t-h}$ from the decision set $\mathcal{S}$ defined by the $K$-dimensional simplex $\mathcal{S} := \Delta_K := \{ w \in \mathbb{R}^K_+ : \sum_{i=1}^K w_i = 1 \}$ in order to form a combined forecast of the form $\hat{y}_{t|t-h} = \sum_{i=1}^K w_{i,t|t-h} \hat{y}_{i,t|t-h}$. We also follow the forecast combination literature by using the quadratic loss to measure the performance of a forecast $\hat{y}_{i,t|t-h}$ as,

$$ l_{i,t} = (y_t - \hat{y}_{i,t|t-h})^2, $$

(1)
with the forecast combination loss from decision weights \( \mathbf{w} \) as,

\[
    l_{\mathbf{w},t} = \sum_{i=1}^{K} w_i \hat{y}_{i,t|t-h}.
\]

Aggregated over time, these losses yield the mean squared forecast error (MSFE) defined for forecaster \( i \) by,

\[
    \text{MSFE}_i = \frac{1}{T-h+1} \sum_{t=h+1}^{T} l_{i,t},
\]

and respectively, for a forecast combination with (fixed) weights \( \mathbf{w} \),

\[
    \text{MSFE}_\mathbf{w} = \frac{1}{T-h+1} \sum_{t=h+1}^{T} l_{\mathbf{w},t}.
\]

A large share of the forecast combination literature has then focused on determining fixed optimal weights that minimize the mean squared forecast error. However, in practice, theoretically determined optimal weights have failed to consistently outperform the mean, i.e. the forecast combination that assigns fixed uniform \( 1/K \) weights over each of the \( K \) forecasts. This negative result is usually referred to as the “Forecast Combination Puzzle.” Non-stationarity and regime switches are intuitive explanations for the presence of this puzzle. Time-varying weights are a natural approach to handle these issues, yet existing approaches have failed to overcome this puzzle (see the seminal paper by Bates and Granger (1969) and Timmermann (2006) for a survey). In order to shade new light on this problem, we build on recent advances in the online machine learning literature (in particular Sani et al., 2014).

The building blocks of our approach are forecast combination algorithms that provide (time-varying) forecast combination weights at the sequence of dates \( t = 1 \cdots T - h \). More precisely, the information available to the decision-maker at time \( t \) is given by the history of forecasts and realizations:

\[
    \mathcal{H}_t = \{ (y_1, \ldots, y_t), \hat{y}_{1,h+1|t}, \ldots, \hat{y}_{t,h|t-h}, \hat{y}_{1,t-t-h}, \ldots, \hat{y}_{K,t-t-h} \}.
\]

A forecast combination algorithm \( \mathcal{G} \) is then defined as a series of mappings \( (g_t)_{t=1,...,T-h} \), where \( g_t : \mathcal{H}_t \rightarrow \Delta_K \). The mapping \( g_t \) associates to an observation history \( h_t \in \mathcal{H}_t \)
a vector of weights \( g_t(h_t) = (w_{1,t}, \ldots, w_{K,t}) \) in the \( K \)-dimensional simplex \( \Delta_K \) and hence aggregates the pool of forecasts available at time \( t \), \((\tilde{y}_{1,t|t-h}, \ldots, \tilde{y}_{K,t|t-h})\), into a single point forecast \( \hat{y}_{t|t-h} = \sum_{i=1}^{K} w_{i,t} \tilde{y}_{i,t|t-h} \).

Baseline combination methods from the macro-economic forecast combination literature include the mean, trimmed mean and median. Within the online learning literature, a large class of algorithms have the following structure: each forecaster is characterized by a score \( \lambda_{i,t} \) that the decision-maker sequentially updates on the basis of observed losses \( l_{i,t} \) and uses to choose his mixture over forecasts (commonly in the form of probability weights which are assigned over the forecasts). A prominent example of such algorithms is the exponentially weighted average forecaster \textbf{Hedge} (see e.g. Freund and Schapire (1997); Littlestone and Warmuth (1994); Vovk (1990); Cesa-Bianchi and Lugosi (2006)), which exponentially updates the mixture over forecasts according to the gradient of their losses (see the algorithm protocol in Figure 1).

![Algorithm Protocol](https://example.com/algorithm.png)

**Input:** Learning rate \( \eta > 0 \), Experts \( \{1, \ldots, K\} \), Decision set \( \mathcal{S} = \Delta_K \), Rounds \( T \), Losses \( \mathcal{L} = [0,1]^K \).

**Initialize** scores: \( \lambda_{i,1} = \frac{1}{K}, \forall i \).

**For all** \( t = 1, \ldots, T \), **repeat**

1. Simultaneously
   - Environment chooses losses \( l_t \in \mathcal{L} \).
   - Learner chooses decision \( w_t \in \mathcal{S} \), where \( w_{i,t} = \frac{\lambda_{i,t}}{\sum_{i=1}^{K} \lambda_{i,t}}, \forall i \).

2. Environment reveals losses \( l_t \).

3. Learner suffers loss \( w_t^\top l_t \).

4. Learner updates scores \( \lambda_{i,t+1} \), as \( \lambda_{i,t+1} = \lambda_{i,t} \exp(-\eta l_{i,t}), \forall i \).

**end for**

Figure 1: \textbf{Hedge}
The performance of these algorithms is conventionally measured using the notion of regret that accounts for the learning properties of the algorithm over time. Namely, if one denotes by \( l_G,\tau \) the loss of algorithm \( G \) in period \( \tau \) and by \( L_G,t = \sum_{\tau=1}^t l_G,\tau \) its cumulative loss up to period \( t \), the regret of an algorithm \( G \) with regard to another algorithm \( H \) up to time \( t \) is defined as \( R_{G,t}(H) = L_G,t - L_H,t \). With some abuse of notation, we can also define the regret of an algorithm \( G \) up to time \( t \) against a forecaster \( i \) as \( R_{G,t}(i) \). Note that the regret is generally measured with respect to the best forecaster in hindsight \( i^* := \arg\min_{i \in K} L_i,T \). If the cumulative regret grows at a rate that is less than linear, the algorithm approaches the performance of the best forecaster in hindsight and is said to be “learning”\(^2\) Hedge offers learning properties which are “optimal” in the worst case with respect to the best forecaster in hindsight. Namely, one has:

**Theorem 1** \(\textbf{[Cesa-Bianchi and Lugosi, 2006]}\) For any finite horizon \( T \) and forecasters \( K \), the regret upper bound for Hedge satisfies,

\[
R_{Hedge,T}(i) \leq \frac{T\eta}{8} + \frac{\log K}{\eta},
\]

against any forecaster \( i \), and the following regret bound with optimized learning rate \( \eta = \sqrt{\frac{8\log K}{T}} \),

\[
R_{Hedge,T}(i) \leq \sqrt{\frac{T}{2}} \log K.
\]

Hence, Hedge achieves the worst-case regret \( O(\sqrt{T\log K}) \) to any forecaster \( i \), including the ex-post optimal choice \( i^* \) \(\textbf{[Cesa-Bianchi and Lugosi, 2006]}\). Note that a “worst-case” guarantee holds in all possible realizations of the loss sequence. Also note that according to Theorem 2.2 in \(\textbf{[Cesa-Bianchi and Lugosi, 2006]}\), this bound can not be improved upon by an exponentially weighted average forecaster.

\(^2\)Also note that the regret does not characterize the absolute performance of the algorithm \( G \). A negative regret is possible if \( G \) outperforms the optimal candidate forecaster.
Now, in the context of (macro-economic) forecasting, the best forecaster in hind-
sight might not be the appropriate benchmark. For example, regime switches may
provide an antagonistic realization of the sequence which does not clearly favor
a single forecaster in hindsight. This results in an incentive to hedge against the
risk that the selected forecaster may not be the best choice over time. According
to the forecast combination puzzle, a logical benchmark in this case is the mean
combination forecaster $\mu$.

A naive approach to benchmark against the mean would be to add the mean
forecast combination as an additional forecaster to the pool for Hedge. According
to Theorem 1, this would result in an equivalent $O(\sqrt{T})$ regret guarantee to the
mean and other forecasters that compose the pool. Now, Hedge is optimal when
it comes to provide a uniform upper bound on the regret with respect to all the
forecasters in the pool. Our objective rather is to bound specifically the regret to
the mean (or another benchmark), while maintaining a close to optimal $O(\sqrt{T})$
bound on the regret with respect to the remaining forecasters.

**Input:** Learning rate $\eta \in (0, \frac{1}{2}]$, Experts $\{1, \ldots, K\}$, Decision set $\mathcal{S} = \Delta_K$,
Rounds $T$ and Losses $\mathcal{L} \in [0, 1]^K$.

**Initialize** scores: $\lambda_{i,1} = \frac{1}{K}, \forall i$.

**For all** $t = 1, \ldots, T$, **repeat**

1. Simultaneously
   - Environment chooses losses $l_t \in \mathcal{L}$.
   - Learner chooses decision $w_t \in \mathcal{S}$, where $w_{i,t} = \frac{\lambda_{i,t}}{\sum_{i=1}^{K} \lambda_{i,t}}, \forall i$.

2. Environment reveals losses $l_t$.

3. Learner suffers loss $w_t^\top l_t$.

4. Learner updates scores $\lambda_{t+1}$, as $\lambda_{i,t+1} = \lambda_{i,t}(1 - \eta l_{i,t}), \forall i$.

**end for**

Figure 2: Prod
From an analytical perspective, this requires determining an analytic expression of the regret bound that is specific to a forecaster. Such an analytic expression can be obtained by replacing the exponential weight update, $e^{\eta x}$, of Hedge by its linear approximation, $1 + \eta x$. The resulting algorithm, usually referred to as Prod in the online learning literature, is described in Figure 2. It provides an explicit characterization of the regret with respect to a forecaster as a function of its losses. Namely, one has:

**Theorem 2** (Cesa-Bianchi and Lugosi, 2006; Cesa-Bianchi et al., 2007) For any $T$ and learning rate $\eta \in (0, 1/2]$, Prod satisfies the following second-order regret bound,

$$R_{\text{Prod},T}(i) \leq \eta \sum_{t=1}^{T} l_{i,t}^2 + \frac{\log K}{\eta},$$

$$\leq \eta T + \frac{\log K}{\eta},$$

for any forecaster $i$, and the following regret bound with optimized learning rate $\eta = \sqrt{\frac{\log K}{T}}$,

$$R_{\text{Prod},T}(i) \leq 2\sqrt{T\log K}.$$

**Remark 1** The regret bound in Prod is said to be “second-order” as it is a function of the sum of squared losses.

**Remark 2** Note that the linear approximation in Prod results in an asymmetric update that recovers slower than the exponential equivalent. Analytically, the higher-order terms that are missing from the Prod update (step 4 in Figure 2) imply a lag with regards to the exponential update in Hedge (step 4 in Figure 1). Empirically, the update step in Prod requires additional reinforcement (i.e. further updates in the same direction) to yield an effect equivalent to this of the update step in Hedge.
Input: Rounds $T$, Losses $L \in [0,1]^K$, Decision set $S = \Delta_K$.

Base forecasters $\{1, \ldots, K\}$ and Fixed distribution $D \in S$.

Initialize: Learning rate $\eta = \sqrt{\log K}$, weight $\lambda_0 = 1 - \eta$ on allocation $D$, Base forecaster weights $\mu_i = \frac{K}{i}$ for $i \in \{1, \ldots, K\}$, $w_{i,1} = \mu_i, \forall i \in \{0, \ldots, K\}$.

For all $t = 1, \ldots, T$, repeat

1. Simultaneously
   • Environment chooses $l_t \in L$.
   • Learner chooses decision $w_t \in S$, where
     $$w_{i,t} = \frac{\lambda_{i,t}}{\sum_{i=0}^{K} \lambda_{i,t}},$$
     for $i \in \{0, \ldots, K\}$.

2. Environment reveals $l_t$.

3. Learner suffers loss $w_t^T l_t$.

4. Learner updates weights $\lambda_{t+1}$, where for $i \in \{1, \ldots, K\}$,
   $$\lambda_{i,t+1} = \lambda_{i,t}(1 - \eta(l_{i,t} - l_{0,t})).$$

end for

Figure 3: $D$-Prod[Even-Dar et al. 2008]

Then, in order to obtain dual regret bounds, one with respect to the benchmark and the other with respect to the remaining forecasters, [Even-Dar et al. 2008] introduce an alternative normalization rule in $D$-Prod for the scores of the different forecasts. Namely, the score of the benchmark is kept fixed while the score of the other forecasters are updated as a function of their relative performance with respect to this of the benchmark. The resulting algorithm $D$-Prod is described in Figure 3. Its regret with respect to the benchmark is minimal because the “benchmark” portion of the algorithm has, by construction, zero regret while the weight on the
other forecasters increase only if they perform better than the benchmark. Namely, one has:

**Theorem 3 (Even-Dar et al. (2008))** For any rounds $T$ and forecasters $K$, $D$–Prod satisfies the following regret bounds,

$$R_{D Prod, T}(i) = \mathcal{O}\left(\sqrt{T \log K} + \sqrt{T \log \log T}\right),$$

for any $x \in S$ and

$$R_{D Prod, T}(i) = R_{D, T}(i) + \mathcal{O}(1).$$

**Remark 3** Hence, $D$-Prod achieves constant regret with respect to the benchmark. As emphasized in Sani (2015), this performance can be achieved because the asymmetry in Prod emphasized in Remark 2 is directed in favor of the benchmark. The use of an “exact” algorithm like Hedge would on the contrary result in equivalent $\mathcal{O}(\sqrt{T})$ regret with respect to the benchmark and other forecasters in the pool.

Sani et al. (2014) generalize this result further to a meta-structure that hedges an algorithm $A$ based on a benchmark algorithm $B$. More precisely, the algorithm $AB$-Prod combines two forecast combination algorithms $A$ (the Alternative) and $B$ (the Benchmark) into a third “hedged” algorithm $C$, where $C$ adapts a weighting over $A$ and $B$ according to the history of their performance (see e.g. Figure 4). This approach allows to consider any algorithm as a benchmark (while $D$-Prod only considered benchmark to a fixed distribution) and moreover allows to refine the bound to Benchmark $B$ by considering a weighted combination of algorithms $A$ and $B$, rather than independent algorithms $K$. Namely, one obtains a fixed constant $2 \log 2$ regret to the benchmark $B$ under any realizations of the loss sequence:

**Theorem 4** (cf. Theorem 1 in Sani et al. (2014)) Let $A$ be any algorithm, $B$ be any benchmark and $D$ be an upper bound on the benchmark losses $L_{B, T}$. Then
Input: Algorithms $\mathcal{A}$ and $\mathcal{B}$, the convex subset of $\mathcal{S} \in \mathbb{R}^2$, Rounds $T$, Losses $\mathcal{L} \in [0,1]^2$ and weight $\lambda_B \in (0,1)$.

Initialize: $\lambda_A = 1 - \lambda_B$, Learning rate $\eta = C\sqrt{\frac{1}{T}} < \frac{1}{2}$, where $C = \sqrt{-\log(1 - \lambda_B)}$.

For all $t = 1, 2, \ldots, T$, repeat

1. Simultaneously
   - Environment chooses $l_t \in \mathcal{L}$.
   - Learner sets $s_t = \frac{\lambda_{A,t}}{\lambda_{A,t} + \lambda_B}$.

2. Environment reveals $l_t$.

3. Learner computes respective losses $l_{A,t}$ and $l_{B,t}$.

4. Learner suffers loss $l_{AB-Prod,t} = s_t l_{A,t} + (1 - s_t) l_{B,t}$.

5. Compute $\delta_t = l_{B,t} - l_{A,t}$.

6. Set $\lambda_{A,t+1} = \lambda_{A,t}(1 + \eta \delta_t)$

end for

Figure 4: $\mathcal{A}\mathcal{B}$-Prod

setting weight $\lambda_B \in (0,1)$, $\lambda_A = 1 - \lambda_B$, Learning rate $\eta = C\sqrt{\frac{1}{T}} < \frac{1}{2}$, where $C = \sqrt{-\log(1 - \lambda_B)}$ simultaneously guarantees,

$$\mathcal{R}_{\mathcal{A}\mathcal{B}-Prod,T}(i) \leq \mathcal{R}_{\mathcal{A},T}(i) + 2C\sqrt{D},$$

for any forecaster $i$ and,

$$\mathcal{R}_{\mathcal{A}\mathcal{B}-Prod,T}(i) \leq \mathcal{R}_{\mathcal{B},T}(i) + 2\log 2,$$

against any assignment of the loss sequence.

The asymmetric update in $Prod$ allows $\mathcal{A}\mathcal{B}$-Prod to adaptively trade-off between $\mathcal{A}$ and $\mathcal{B}$ with an asymmetry that provides the necessary momentum to outperform algorithms that rely on a differencing to determine their weights \footnote{One might consider a similar $\mathcal{A}\mathcal{B}$ structure using $Hedge$ in place of $Prod$ (e.g. $\mathcal{A}\mathcal{B}$-Hedge).} Further, by
LEARNING TIME-VARYING FORECAST COMBINATIONS

initializing the benchmark $B$ with a large starting weight, $AB$-Prod requires that the alternative algorithm $A$ demonstrate an explicit advantage before the $AB$-Prod updates begin to prefer $A$ over $B$. This results in the $O(\sqrt{T})$ regret to any forecaster $i$ and a constant $O(2 \log 2)$ to the performance of the Benchmark $B$.

It follows that setting $B = \mu$ “solves” the “Forecast Combination Puzzle” in the sense that $AB$-Prod then provides constant and distribution-free theoretical guarantees for the relative performance with respect to the mean combination $\mu$ while exploiting any superior predictive ability of an alternative algorithm $A$, which can be chosen arbitrarily by the decision-maker. Namely, one has:

**Theorem 5**  Let $A$ be any algorithm, $B$ be the mean combination $\mu$ and $D$ be an upper bound on the benchmark losses $L_{B,T}$. Then setting weight $\lambda_B \in (0, 1)$, $\lambda_A = 1 - \lambda_B$, Learning rate $\eta = C\sqrt{\frac{1}{T}} < \frac{1}{2}$, where $C = \sqrt{-\log(1 - \lambda_B)}$ simultaneously guarantees,

$$R_{AB\text{-Prod},T}(i) \leq R_{A,T}(i) + 2C\sqrt{D},$$

for any forecaster $i$, and,

$$R_{AB\text{-Prod},T}(i) \leq R_{\mu,T}(i) + 2 \log 2.$$

4. SYNTHETIC RESULTS

In order to demonstrate the ability of $AB$-Prod to effectively and rapidly adapt to the relative performance of the alternative algorithm $A$, while offering the protection of an explicit benchmark $B$, we first construct two synthetic scenarios of loss sequences (according to the method proposed in [de Rooij et al., 2014] and presented in Sani et al. (2014); Sani (2015), which correspond respectively to situations where the alternative and the benchmark (i.e. the mean in our example) perform poorly.

The alternative algorithm we consider in these scenarios is **AdaHedge**, which is but this would result in $O(\sqrt{T})$ regret to both $A$ and $B$. This was also demonstrated in Sani et al. (2014); Sani (2015).
a variant of **Hedge** with an adaptive learning rate $\eta$ (see de Rooij et al., 2014). The two scenarios consist of 1000 loss observations for two forecasters. Losses have values in \{0,1\}, i.e. each expert can be right or wrong. Regret and RMSE results are presented for the mean forecast combination $\mu$, **AdaHedge**, and $\mathcal{A}\mathcal{B}$-Prod with $\mathcal{A}$ set to AdaHedge and $\mathcal{B}$ set to $\mu$ in tables I and II.

<table>
<thead>
<tr>
<th>Scenario 0</th>
<th>Mean</th>
<th>AdaHedge</th>
<th>AB-Prod(AdaHedge,$\mu$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 0</td>
<td>0.5</td>
<td>13.187577</td>
<td>0.507855</td>
</tr>
<tr>
<td>Scenario 1</td>
<td>248.5</td>
<td>2.250753</td>
<td>89.459579</td>
</tr>
</tbody>
</table>

Table I: Regret

<table>
<thead>
<tr>
<th>Scenario 0</th>
<th>Mean</th>
<th>AdaHedge</th>
<th>AB-Prod(AdaHedge,$\mu$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 0</td>
<td>1.0</td>
<td>1.025401</td>
<td>1.000016</td>
</tr>
<tr>
<td>Scenario 1</td>
<td>1.0</td>
<td>0.507009</td>
<td>0.681601</td>
</tr>
</tbody>
</table>

Table II: Relative MSE

In Scenario 1, the mean combination approach clearly outperforms AdaHedge with regard to its Regret. This is also reflected in its RMSE. Note that the Regret provides a clearer picture of the performance over time. The performance is such that it is impossible to beat the mean combination approach, especially while changing weights over time. This suggests that neither of the forecaster loss sequences shows a substantial advantage. Accordingly, **AdaHedge** is unable to exploit any additional information and underperforms the mean. $\mathcal{A}\mathcal{B}$-Prod quickly recognizes the advantage of the mean combination, paying a slight Regret in “learning”.

In Scenario 2, the mean combination approach fails and **AdaHedge** offers a substantial performance advantage. This performance is such that it is clear one of the forecaster loss sequences dominates the other. $\mathcal{A}\mathcal{B}$-Prod(AdaHedge,$\mu$) recognizes the

---

4The approach generates losses directly for each of the forecasters at each time-step, rather than a prediction for both experts together with a realization.
alternative advantage and shifts weights accordingly to perform almost as well as AdaHedge (see table I).

To further illustrate the hedging capacity of $\mathcal{AB}$-Prod, we perform a second series of experiments with synthetic data in which we generate 1,000,000 sequences of 100 losses for two experts by drawing randomly and independently from binomial distributions (with a parameter that varies with the Monte-Carlo simulation). The distribution of the Spearman correlation between the two loss sequences composing the tuples is reported in Figure 5. The aim of this setting is to investigate the performance of four forecast combination algorithms together with their $\mathcal{AB}$-Prod($\cdot$,\mu) extensions. More precisely, we consider as baseline algorithms AdaHedge, the time-varying forecast combination approach of Bates-Granger (see Model 1 from Timmermann (2006)), the Recent Best forecaster (which chooses the forecaster that performed best last period) and the random forecaster (ie. choosing one of the forecasters uniformly at random). The corresponding extensions are denoted by $\mathcal{AB}$-Prod(AdaHedge,\mu), $\mathcal{AB}$-Prod(Bates-Granger,\mu), $\mathcal{AB}$-Prod(Recent Best ,\mu) and $\mathcal{AB}$-Prod( Random,\mu). In each case, we set $\lambda_S = 0.999$. Results are reported in Figure 6 via the RMSE.
Figure 6: RMSE over synthetic loss sequences for the Random forecaster, AdaHedge, Bates-Granger and the Recent Best algorithms are reported.
For all forecast combination algorithms considered, $\mathcal{AB}$-Prod($\cdot, \mu$) demonstrates its ability to offer a real-time hedge to the mean. In particular, for each baseline algorithm the maximum RMSE is notably greater than 1 while in the $\mathcal{AB}$-Prod instantiation, the maximum RSME is negligibly greater than 1. In the latter case, the worst performance over all the algorithms is 1.0116. It corresponds to the upfront “cost” of using $\mathcal{AB}$-Prod($\cdot, \mu$), which according to Section 3 is bounded by a $2 \log 2$ cost to the Benchmark $B$. Finally, note that the parameters of $\mathcal{AB}$-Prod were set in order to bias heavily the combination towards the mean, which explains that the advantages of the alternative were very partially capture by the algorithm.

5. EMPIRICAL RESULTS

In order to illustrate the empirical value of our approach, we compare the performance of $\mathcal{AB}$-Prod to this of a set of standard forecast combination algorithms from the macro-economic and machine learning literature in two series of experiments. The first aims at forecasting inflation and output using the seven-country forecast combination dataset from Stock and Watson (2001). The second aims at forecasting the growth-rate in the Euro area using data from the survey of professional forecasters. The set of algorithms considered include:

- A set of basic combination methods: the mean forecaster (denoted by $\mu$), trimmed mean forecasters with $\alpha = 0.05$ and $\alpha = 0.10$ and the median forecaster.
- Three benchmark time-varying combination methods: the AdaHedge algorithm, which is a version of Hedge with adaptive learning rate, the Bates Granger time-varying method 1 (BG) introduced in Timmermann (2006) and the Recent Best forecaster, which selects the forecaster with the lowest loss in the last round.
- The ex-post optimal forecaster, which can of course only be determined ex-post but provides a useful benchmark.
- The random forecaster, which selects a single forecaster at random at each
• Three instantiations of the meta-algorithm $AB$-Prod are presented with $\lambda B = 0.999$, and the Benchmark $B$ set to the mean combination approach. The $\lambda$ weights reflect our desire to heavily prefer the Benchmark $\mu$.

- $AB$-Prod(AdaHedge,$\mu$): $A = \text{AdaHedge}$
- $AB$-Prod(Bates-Granger,$\mu$): $A = \text{the Bates-Granger method}$
- $AB$-Prod(Recent Best,$\mu$): $A = \text{the Recent Best forecaster over the previous round}$

REMARK 4  
Macroeconomic data, and most particularly the SPF, often has missing values, resulting in missing losses that negatively bias otherwise well-performing forecasters. The solution we adopt in the following is to impute missing data with the mean, more precisely to lower-bound the performance of the missing forecaster performance by this of the mean.

5.1. Seven country forecast combination dataset

The seven-country forecast combination dataset from Stock and Watson (2001), consists in 43 quarterly time series of macro-economic indicators available for seven different countries: Canada, France, Germany, Italy, Japan, the United Kingdom and United States. The time-series include asset prices, selected measures of real economic activity and money stock from 1959 to 1999. Each of these time-series is then used to produce independent forecasts of inflation and output by estimating an autoregressive model with one exogenous variable (ARX). These forecasts are then combined using our set of candidate algorithms with a burn-in period of 8 quarters. This experiment is then repeated independently for inflation and output for three different forecast horizons, $h = 2, 4$ and 8 quarters.

REMARK 5  
The ARX forecasts are recursively generated for each exogenous variable using the Python Statsmodels library (Seabold and Perktold, 2010). Coefficients are
estimated according to the Akaike information criterion (AIC) over 4 lags, with ARX forecasts generated using a Broyden-Fletcher-Goldfarb-Shanno solver and maximum likelihood estimation on samples up to time \( t \). Failed forecasts due to failed maximum likelihood convergence are replaced with the preceding forecast.

<table>
<thead>
<tr>
<th></th>
<th>Average RMSE</th>
<th>Min RMSE</th>
<th>Max RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaHedge</td>
<td>1.006743</td>
<td>0.727263</td>
<td>1.424006</td>
</tr>
<tr>
<td>Recent Best</td>
<td>1.288761</td>
<td>0.395634</td>
<td>18.447350</td>
</tr>
<tr>
<td>Bates-Granger</td>
<td>1.026792</td>
<td>0.726393</td>
<td>1.247406</td>
</tr>
<tr>
<td>Median</td>
<td>0.975889</td>
<td>0.723440</td>
<td>1.102208</td>
</tr>
<tr>
<td>Trimmed Mean (alpha=0.05)</td>
<td>0.957247</td>
<td>0.719920</td>
<td>1.024449</td>
</tr>
<tr>
<td>Trimmed Mean (alpha=0.10)</td>
<td>1.663990</td>
<td>0.659524</td>
<td>3.803069</td>
</tr>
<tr>
<td>AB-Prod (AdaHedge,( \mu ))</td>
<td>0.952805</td>
<td>0.718049</td>
<td>0.999840</td>
</tr>
<tr>
<td>AB-Prod (Bates-Granger,( \mu ))</td>
<td>0.952807</td>
<td>0.718049</td>
<td>0.999842</td>
</tr>
<tr>
<td>AB-Prod (Recent Best,( \mu ))</td>
<td>0.952835</td>
<td>0.718046</td>
<td>0.999843</td>
</tr>
<tr>
<td>Random Forecaster</td>
<td>1.051770</td>
<td>0.735312</td>
<td>1.353242</td>
</tr>
<tr>
<td>Ex-Post Optimal</td>
<td>0.798261</td>
<td>0.557397</td>
<td>0.974834</td>
</tr>
</tbody>
</table>

Table III: Average, Minimum and Maximum Ratio to the mean of the Mean Square Forecast Error over GDP, CPI, Horizons and Countries

Table III provides a summary of the main results (whose details are in the appendix). The experiment clearly illustrate the performance advantage and adaptive hedging capabilities of the AB-Prod meta structure. The three AB-Prod algorithms outperform the mean for every possible combination of indicator, country and horizon, i.e the maximal RMSE ratio is less than 1. In terms of average performance, they outperform all but the ex-post optimal forecaster, which can only be determined ex-post. Their average performance is also better that this of other time-varying combination algorithms (AdaHedge, Recent Best and Bates-Granger). Moreover, these algorithms do not systematically guarantee better performance than the mean.
A detailed analysis of the results presented in the appendix shows that AB-Prod outperforms AdaHedge, Recent Best and Bates-Granger almost systematically. This suggests that AB-Prod manages, thanks to its meta-structure, to catch faster regime switches in the data and hence alternates between the alternative algorithm when it has a comparative advantage to the mean, while rapidly falling back to the safety of the mean when the alternative algorithm is unable to exploit further available information.

5.2. Survey of professional forecasters

The Euro-area Survey of Professional Forecasters has been conducted by the European Central Bank at a quarterly frequency since the inception of the European Monetary Union (see Bowles et al., 2007, 2010; Garcia, 2003, for a detailed description). There are around 75 survey participants, who are experts affiliated with financial and non-financial European institutions. The average number of respondents per survey is 59. Each participant\(^5\) receives a survey of growth expectations for 1 and 2 year rolling horizons\(^6\), with one week to reply. Survey results are published the following month. Further, the target forecast changes depending on the specific criterion by which the GDP is measured. These experts are asked to provide point forecasts for GDP and inflation at different horizons (we focus on the 1-year rolling GDP forecast horizon). Their answers provide a time-series of forecasts that are natural inputs for a forecast combination approach.

Remark 6 The SPF suffers from a large number of missing values with less than 60 respondents in average. We have considered two approaches to overcome this issue: the reduction to a balanced panel, as is common in the SPF literature (see e.g. Elliott\(^7\)).

\(^{5}\)Note that the specific expert at each institution is not necessarily the same in each survey and the data has many missing values.

\(^{6}\)Note that the rolling horizons are set one and two years ahead of the latest period for which the variable in question is observed when the survey is conducted and not one or two years ahead of the survey date.
and Timmermann (2005); Aiolfi et al. (2010); Genre et al. (2013)), and imputation of the missing values through the mean of available forecasts at the specific time step. Both approaches give similar results. Due to these gaps and the frequency of mean imputations, forecaster performance is expected to be close to the mean of existing forecasts. $\mathcal{AB}$ instantiations are set to $\lambda_B = 0.999$.

<table>
<thead>
<tr>
<th></th>
<th>Balanced</th>
<th>Imputed</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaHedge</td>
<td>0.969944</td>
<td>0.969086</td>
</tr>
<tr>
<td>Recent Best</td>
<td>0.810353</td>
<td>0.808181</td>
</tr>
<tr>
<td>Bates-Granger</td>
<td>1.021676</td>
<td>1.010879</td>
</tr>
<tr>
<td>Median</td>
<td>0.995093</td>
<td>0.995307</td>
</tr>
<tr>
<td>Trimmed Mean(alpha=0.05)</td>
<td>0.997080</td>
<td>0.996030</td>
</tr>
<tr>
<td>Trimmed Mean(alpha=0.10)</td>
<td>0.985286</td>
<td>1.006134</td>
</tr>
<tr>
<td>AB-Prod(AdaHedge,$\mu$)</td>
<td>0.995106</td>
<td>0.995319</td>
</tr>
<tr>
<td>AB-Prod(Bates-Granger,$\mu$)</td>
<td>0.995108</td>
<td>0.995316</td>
</tr>
<tr>
<td>AB-Prod(RB,Median)</td>
<td>0.995831</td>
<td>0.996091</td>
</tr>
<tr>
<td>Random Forecaster</td>
<td>0.969570</td>
<td>0.963621</td>
</tr>
<tr>
<td>Ex-Post Optimal</td>
<td>0.862678</td>
<td>0.860968</td>
</tr>
</tbody>
</table>

Table IV: SPF Data: Imputed and Balanced

Table IV reports the performance of the forecast combination algorithms in this setting. The three $\mathcal{AB}$-Prod algorithms are close to or outperform the mean combination forecast. The cost of protection is apparent in the gap between the $\mathcal{AB}$ instantiations and non-$\mathcal{AB}$ forms of the algorithms. This is most clear in the gap in performance between the $\mathcal{AB}$ and non-$\mathcal{AB}$ forms of the Recent Best algorithm. Given the lack of structure in the data resulting from gaps, changing forecasters and inconsistent forecast histories per forecaster, the problem of “learning” is expected. In each case, the $\mathcal{AB}$ instantiations learn to prefer the safety and consistency of the
mean. If a larger margin of error was acceptable, one might consider reducing the weight $\lambda_B$.

6. DISCUSSION AND CONCLUSIONS

The paper presented a novel meta-hedging approach to adaptively combining candidate forecasts over time. More specifically, an algorithm was proposed based on several modifications to the state-of-the-art $AB$-Prod algorithm from the online machine learning literature that provides an intuitive imputation strategy over an augmented pool and explicit protection against the forecast combination puzzle. In the latter, the proposed algorithm actively and adaptively “hedges” performance to the Benchmark, while providing dual distribution-free theoretical regret guarantees that the performance will never be worse by a fixed constant against the benchmark with additional dual guarantees against a pool of forecasters augmented by the man and any other forecast combination algorithm. In addition to providing outstanding performance, the proposed methods provide a simple, consistent and theoretically guaranteed procedure for hedging against the so-called Forecast Combination Puzzle, while also giving access to state-of-the-art tools for combining forecasts.

7. ACKNOWLEDGMENTS

We gratefully acknowledge the support of H2020-FET proactive project DOLFINS and the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research. We also thank Gilles Stoltz for his useful comments.
REFERENCES


de Rooij, S., van Erven, T., Grünwald, P. D., and Koolen, W. M. (2014). Follow the leader if you can, hedge if you must. Accepted to the Journal of Machine Learning Research.


LEARNING TIME-VARYING FORECAST COMBINATIONS


Figure 7: Relative MSFE, Horizon = 2
Figure 8: Relative MSFE, Horizon = 4
Figure 9: Relative MSFE, Horizon = 8
<table>
<thead>
<tr>
<th></th>
<th>China</th>
<th>France</th>
<th>Germany</th>
<th>Italy</th>
<th>Japan</th>
<th>UK</th>
<th>US</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaHedge</td>
<td>1.027692</td>
<td>0.917716</td>
<td>0.955901</td>
<td>1.089603</td>
<td>1.07820</td>
<td>1.043915</td>
<td>1.079786</td>
</tr>
<tr>
<td>Recent Best</td>
<td>0.887602</td>
<td>0.785163</td>
<td>0.813654</td>
<td>0.778690</td>
<td>0.77533</td>
<td>0.919000</td>
<td>1.515963</td>
</tr>
<tr>
<td>Bates-Granger</td>
<td>1.028942</td>
<td>0.926949</td>
<td>1.006539</td>
<td>1.109468</td>
<td>1.007305</td>
<td>1.023833</td>
<td>1.084799</td>
</tr>
<tr>
<td>Median</td>
<td>0.967980</td>
<td>0.850313</td>
<td>0.979326</td>
<td>1.003207</td>
<td>0.971819</td>
<td>0.974722</td>
<td>1.009402</td>
</tr>
<tr>
<td>Trimmed Mean(alpha=0.05)</td>
<td>0.977863</td>
<td>0.850787</td>
<td>0.979695</td>
<td>0.979157</td>
<td>0.918648</td>
<td>0.974839</td>
<td>0.986960</td>
</tr>
<tr>
<td>Trimmed Mean(alpha=0.10)</td>
<td>1.759095</td>
<td>2.099013</td>
<td>1.337272</td>
<td>2.456080</td>
<td>1.962812</td>
<td>1.796089</td>
<td>2.445601</td>
</tr>
<tr>
<td>AB-Prod(AdaHedge, µ)</td>
<td>0.97713</td>
<td>0.854275</td>
<td>0.975541</td>
<td>0.975925</td>
<td>0.918648</td>
<td>0.974839</td>
<td>0.986961</td>
</tr>
<tr>
<td>AB-Prod(Bates-Granger, µ)</td>
<td>0.97713</td>
<td>0.854276</td>
<td>0.975546</td>
<td>0.975922</td>
<td>0.918648</td>
<td>0.974839</td>
<td>0.986961</td>
</tr>
<tr>
<td>AB-Prod(Recent Best, µ)</td>
<td>0.977699</td>
<td>0.854261</td>
<td>0.975526</td>
<td>0.975892</td>
<td>0.918625</td>
<td>0.974829</td>
<td>0.987004</td>
</tr>
<tr>
<td>Random Forecaster</td>
<td>1.023256</td>
<td>1.165608</td>
<td>1.032905</td>
<td>1.064479</td>
<td>1.193638</td>
<td>0.974331</td>
<td>1.034971</td>
</tr>
<tr>
<td>Ex-Post Optimal</td>
<td>0.903505</td>
<td>0.795443</td>
<td>0.788056</td>
<td>0.885872</td>
<td>0.774105</td>
<td>0.926796</td>
<td>0.965566</td>
</tr>
</tbody>
</table>

Table V: CPI, Horizon = 2, Relative MSFE Results

<table>
<thead>
<tr>
<th></th>
<th>China</th>
<th>France</th>
<th>Germany</th>
<th>Italy</th>
<th>Japan</th>
<th>UK</th>
<th>US</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaHedge</td>
<td>0.942731</td>
<td>1.008187</td>
<td>1.424006</td>
<td>1.084366</td>
<td>0.829337</td>
<td>1.013956</td>
<td>0.953742</td>
</tr>
<tr>
<td>Recent Best</td>
<td>0.898264</td>
<td>1.051995</td>
<td>1.587237</td>
<td>0.692809</td>
<td>0.663120</td>
<td>1.116999</td>
<td>0.931004</td>
</tr>
<tr>
<td>Bates-Granger</td>
<td>0.935480</td>
<td>1.002353</td>
<td>1.247406</td>
<td>1.092291</td>
<td>1.068171</td>
<td>1.002029</td>
<td>0.953713</td>
</tr>
<tr>
<td>Median</td>
<td>0.922007</td>
<td>1.005610</td>
<td>1.101678</td>
<td>1.089201</td>
<td>1.078190</td>
<td>0.962154</td>
<td>0.923202</td>
</tr>
<tr>
<td>Trimmed Mean(alpha=0.05)</td>
<td>0.915359</td>
<td>0.985940</td>
<td>1.018093</td>
<td>1.099618</td>
<td>1.024449</td>
<td>0.962236</td>
<td>0.919850</td>
</tr>
<tr>
<td>Trimmed Mean(alpha=0.10)</td>
<td>1.102148</td>
<td>1.435502</td>
<td>1.716145</td>
<td>1.705908</td>
<td>0.659524</td>
<td>1.430694</td>
<td>1.267201</td>
</tr>
<tr>
<td>AB-Prod(AdaHedge, µ)</td>
<td>0.912228</td>
<td>0.975225</td>
<td>0.997529</td>
<td>0.997485</td>
<td>0.999730</td>
<td>0.966550</td>
<td>0.916113</td>
</tr>
<tr>
<td>AB-Prod(Bates-Granger, µ)</td>
<td>0.912227</td>
<td>0.975224</td>
<td>0.997513</td>
<td>0.997486</td>
<td>0.999755</td>
<td>0.966549</td>
<td>0.916113</td>
</tr>
<tr>
<td>AB-Prod(Recent Best, µ)</td>
<td>0.912223</td>
<td>0.975229</td>
<td>0.997543</td>
<td>0.997445</td>
<td>0.999709</td>
<td>0.966560</td>
<td>0.916111</td>
</tr>
<tr>
<td>Random Forecaster</td>
<td>0.955054</td>
<td>1.001357</td>
<td>1.306822</td>
<td>1.072049</td>
<td>0.968677</td>
<td>1.001748</td>
<td>0.955778</td>
</tr>
<tr>
<td>Ex-Post Optimal</td>
<td>0.865022</td>
<td>0.850251</td>
<td>0.764778</td>
<td>0.559406</td>
<td>0.557397</td>
<td>0.887923</td>
<td>0.793327</td>
</tr>
</tbody>
</table>

Table VI: RGDP, Horizon = 2, Relative MSFE Results

<table>
<thead>
<tr>
<th></th>
<th>China</th>
<th>France</th>
<th>Germany</th>
<th>Italy</th>
<th>Japan</th>
<th>UK</th>
<th>US</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaHedge</td>
<td>0.987378</td>
<td>1.148972</td>
<td>0.962878</td>
<td>1.188571</td>
<td>0.915973</td>
<td>0.981786</td>
<td>1.148653</td>
</tr>
<tr>
<td>Recent Best</td>
<td>0.690108</td>
<td>0.725732</td>
<td>0.831245</td>
<td>0.651087</td>
<td>0.234325</td>
<td>0.719829</td>
<td>0.791346</td>
</tr>
<tr>
<td>Bates-Granger</td>
<td>1.048280</td>
<td>0.997643</td>
<td>0.983153</td>
<td>1.047659</td>
<td>1.053514</td>
<td>1.008424</td>
<td>1.126311</td>
</tr>
<tr>
<td>Median</td>
<td>0.989002</td>
<td>0.945450</td>
<td>0.931294</td>
<td>0.956771</td>
<td>0.985518</td>
<td>0.975252</td>
<td>1.020092</td>
</tr>
<tr>
<td>Trimmed Mean(alpha=0.05)</td>
<td>0.995129</td>
<td>0.932900</td>
<td>0.946988</td>
<td>0.938937</td>
<td>0.915762</td>
<td>0.971114</td>
<td>1.005232</td>
</tr>
<tr>
<td>Trimmed Mean(alpha=0.10)</td>
<td>1.914956</td>
<td>1.726528</td>
<td>1.623579</td>
<td>2.029534</td>
<td>3.803069</td>
<td>1.351599</td>
<td>2.776898</td>
</tr>
<tr>
<td>AB-Prod(AdaHedge, µ)</td>
<td>0.990434</td>
<td>0.933375</td>
<td>0.946516</td>
<td>0.936971</td>
<td>0.894786</td>
<td>0.964997</td>
<td>0.999670</td>
</tr>
<tr>
<td>AB-Prod(Bates-Granger, µ)</td>
<td>0.990440</td>
<td>0.933360</td>
<td>0.946518</td>
<td>0.936957</td>
<td>0.894800</td>
<td>0.964999</td>
<td>0.999698</td>
</tr>
<tr>
<td>AB-Prod(Recent Best, µ)</td>
<td>0.990404</td>
<td>0.933333</td>
<td>0.946503</td>
<td>0.936917</td>
<td>0.894787</td>
<td>0.964970</td>
<td>0.999634</td>
</tr>
<tr>
<td>Random Forecaster</td>
<td>1.222143</td>
<td>0.910372</td>
<td>0.928015</td>
<td>1.211898</td>
<td>1.188929</td>
<td>1.015007</td>
<td>1.220862</td>
</tr>
<tr>
<td>Ex-Post Optimal</td>
<td>0.877543</td>
<td>0.835150</td>
<td>0.837161</td>
<td>0.753022</td>
<td>0.591105</td>
<td>0.762880</td>
<td>0.932993</td>
</tr>
</tbody>
</table>

Table VII: CPI, Horizon = 4, Relative MSFE Results
## Table VIII: RGDP, Horizon = 4, Relative MSFE Results

<table>
<thead>
<tr>
<th></th>
<th>China</th>
<th>France</th>
<th>Germany</th>
<th>Italy</th>
<th>Japan</th>
<th>UK</th>
<th>US</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaHedge</td>
<td>0.799612</td>
<td>0.895833</td>
<td>0.919928</td>
<td>0.980815</td>
<td>0.956147</td>
<td>1.021483</td>
<td>0.896565</td>
</tr>
<tr>
<td>Recent Best</td>
<td>0.395634</td>
<td>0.568066</td>
<td>0.721221</td>
<td>0.621968</td>
<td>0.615868</td>
<td>0.879640</td>
<td>0.661627</td>
</tr>
<tr>
<td>Bates-Granger</td>
<td>1.044667</td>
<td>1.025635</td>
<td>1.007630</td>
<td>1.064509</td>
<td>1.102054</td>
<td>1.039898</td>
<td>1.076220</td>
</tr>
<tr>
<td>Median</td>
<td>1.006999</td>
<td>0.980125</td>
<td>0.967675</td>
<td>0.982552</td>
<td>1.008216</td>
<td>0.999707</td>
<td>1.033027</td>
</tr>
<tr>
<td>Trimated Mean</td>
<td>1.002034</td>
<td>0.968371</td>
<td>0.983280</td>
<td>0.976519</td>
<td>0.975715</td>
<td>0.997962</td>
<td>1.000458</td>
</tr>
<tr>
<td>AB-Prod(AdaHedge, $\mu$)</td>
<td>0.990934</td>
<td>0.964579</td>
<td>0.967675</td>
<td>0.982552</td>
<td>1.008216</td>
<td>0.999707</td>
<td>1.033027</td>
</tr>
<tr>
<td>AB-Prod(Bates-Granger, $\mu$)</td>
<td>0.990959</td>
<td>0.964592</td>
<td>0.980977</td>
<td>0.976519</td>
<td>0.975715</td>
<td>0.997962</td>
<td>1.000458</td>
</tr>
<tr>
<td>AB-Prod(Recent Best, $\mu$)</td>
<td>0.990894</td>
<td>0.964546</td>
<td>0.989494</td>
<td>0.973782</td>
<td>0.966159</td>
<td>0.994597</td>
<td>0.997467</td>
</tr>
<tr>
<td>Random Forecaster</td>
<td>0.955566</td>
<td>0.949688</td>
<td>1.007614</td>
<td>1.353242</td>
<td>1.057964</td>
<td>0.931288</td>
<td>0.979254</td>
</tr>
<tr>
<td>Ex-Post Optimal</td>
<td>0.821905</td>
<td>0.755434</td>
<td>0.887838</td>
<td>0.616751</td>
<td>0.663060</td>
<td>0.836903</td>
<td>0.806403</td>
</tr>
</tbody>
</table>

## Table IX: CPI, Horizon = 8, Relative MSFE Results

<table>
<thead>
<tr>
<th></th>
<th>China</th>
<th>France</th>
<th>Germany</th>
<th>Italy</th>
<th>Japan</th>
<th>UK</th>
<th>US</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaHedge</td>
<td>0.940745</td>
<td>0.727263</td>
<td>1.151296</td>
<td>1.264264</td>
<td>1.027481</td>
<td>1.001673</td>
<td>0.950453</td>
</tr>
<tr>
<td>Recent Best</td>
<td>0.946763</td>
<td>0.704556</td>
<td>1.037182</td>
<td>1.088804</td>
<td>1.053447</td>
<td>1.106216</td>
<td>1.068228</td>
</tr>
<tr>
<td>Bates-Granger</td>
<td>0.964636</td>
<td>0.726393</td>
<td>1.179088</td>
<td>1.128691</td>
<td>1.048846</td>
<td>1.099904</td>
<td>1.003223</td>
</tr>
<tr>
<td>Median</td>
<td>0.961519</td>
<td>0.723440</td>
<td>0.991709</td>
<td>1.090088</td>
<td>1.050152</td>
<td>0.956663</td>
<td>0.964825</td>
</tr>
<tr>
<td>Trimated Mean</td>
<td>0.949375</td>
<td>0.719920</td>
<td>0.995852</td>
<td>0.999659</td>
<td>1.008941</td>
<td>0.964316</td>
<td>0.966100</td>
</tr>
<tr>
<td>Trimated Mean</td>
<td>0.939316</td>
<td>0.800743</td>
<td>1.591848</td>
<td>2.573644</td>
<td>0.783951</td>
<td>2.112149</td>
<td>2.243635</td>
</tr>
<tr>
<td>AB-Prod(AdaHedge, $\mu$)</td>
<td>0.945688</td>
<td>0.718049</td>
<td>0.999464</td>
<td>0.990341</td>
<td>0.998400</td>
<td>0.972429</td>
<td>0.970134</td>
</tr>
<tr>
<td>AB-Prod(Bates-Granger, $\mu$)</td>
<td>0.945690</td>
<td>0.718049</td>
<td>0.999467</td>
<td>0.990327</td>
<td>0.998422</td>
<td>0.972430</td>
<td>0.970139</td>
</tr>
<tr>
<td>AB-Prod(Recent Best, $\mu$)</td>
<td>0.945689</td>
<td>0.718049</td>
<td>0.999453</td>
<td>0.990323</td>
<td>0.998433</td>
<td>0.972440</td>
<td>0.970145</td>
</tr>
<tr>
<td>Random Forecaster</td>
<td>0.967656</td>
<td>0.735312</td>
<td>0.976700</td>
<td>1.249795</td>
<td>1.010907</td>
<td>1.014356</td>
<td>0.976907</td>
</tr>
<tr>
<td>Ex-Post Optimal</td>
<td>0.873424</td>
<td>0.668203</td>
<td>0.974834</td>
<td>0.638040</td>
<td>0.658253</td>
<td>0.936141</td>
<td>0.903260</td>
</tr>
</tbody>
</table>

## Table X: RGDP, Horizon = 8, Relative MSFE Results