Empirical Economics (2021) 61:173–199
https://doi.org/10.1007/s00181-020-01864-w

The robustness of forecast combination in unstable environments: a Monte Carlo study of advanced algorithms

Yongchen Zhao[1]

Received: 25 July 2019 / Accepted: 20 March 2020 / Published online: 7 April 2020

Abstract
In this paper, we study the behavior and effectiveness of several recently developed forecast combination algorithms in simulated unstable environments, where the performances of individual forecasters are cross-sectionally heterogeneous and dynamically evolving. Our results clearly reveal how different algorithms respond to structural instabilities of different origin, frequency, and magnitude. Accordingly, we propose an improved forecast combination procedure and demonstrate its effectiveness in a real-time forecast combination exercise using the U.S. Survey of Professional Forecasters.

Keywords Exponential re-weighting · Shrinkage · Estimation error · Performance instability

JEL Classification C53 · C22 · C15

1 Introduction

Researchers and policymakers routinely combine forecasts from different sources. A large number of studies have clearly demonstrated that combining forecasts helps to improve forecast accuracy and hedge against individual forecasters' idiosyncratic errors. Several new and exciting forecast combination algorithms have been proposed recently. However, even today, practitioners still frequently prefer the most

This research is supported by the SAS/IIF Grant to Promote Research on Forecasting. The author thanks three anonymous referees, Kajal Lahiri, Herman Stekler, Tara Sinclair, as well as seminar participants at the George Washington University and conference participants at the 35th International Symposium on Forecasting for their helpful comments. The author is responsible for any and all errors and omissions in the paper.

✉ Yongchen Zhao
yzhao@towson.edu

[1] Department of Economics, Towson University, Towson, MD 21252, USA

basic method of combination, simple averaging, to more sophisticated algorithms. The mediocre performance of many sophisticated algorithms in practice is often attributed to structural instabilities in real-world data.[1] Several studies have since stressed the need for combination algorithms to be robust to instabilities in the data generating process of the target variable and in the performances of individual forecasters.[2] However, existing studies often focus on a specific algorithm's robustness to a specific form of instability.[3] These studies, many of which also propose new combination algorithms or strategies, invariably favor what they propose. While making valuable contributions in their own rights, these studies, limited by space and scope, often give little consideration to alternative algorithms (beyond a few standard benchmarks) and alternative forms of instabilities. There has been no systematic study of the robustness of forecast combination algorithms in unstable environments where the nature and magnitude of instability is known a priori. Theoretical work on this topic is difficult if not impossible: Each combination algorithm is proven optimal under a strict set of assumptions, where deviations often make the math intractable, especially when time-varying parameters, thick-tailed distributions, and non-stationarities have to be considered. Relying solely on real-world data is in no way a better alternative: While one can apply the combination algorithms and document their performances given any particular data set, figuring out why an algorithm performs the way it does is difficult, since real-world data are generated by unknown processes. To systematically study the performance and robustness of different forecast combination algorithms in unstable environments, the only feasible method is simulation.[4]

Our main objective is to document the behaviors and the performances of various forecast combination algorithms in scenarios involving structural instabilities that are commonly encountered in practice. We are primarily interested in a few recently developed algorithms. The first is the set of *aggregate forecast through exponential reweighting* (AFTER) algorithms, proposed in Yang (2004), Wei and Yang (2012), and most recently Cheng and Yang (2015). These algorithms accommodate the squared error loss, the absolute error loss, and a synthetic loss that is a flexible mixture of the first two. The AFTER algorithms are designed to adapt to changes in the performances of individual forecasters and to produce accurate combined forecasts with few outliers. The second is the algorithm proposed in Sancetta (2010). Compared with the AFTER algorithms, it allows for more general loss functions and works under more relaxed assumptions on the properties of individual forecasts. To hedge against structural instabilities, this algorithm features a shrinkage step that reduces the differences in individuals' weights. The results, therefore, are closer to those based on equal weights.

[1] Clemen (1989) reviewed more than 200 studies on forecast combination and observed that simple averaging often outperforms more complicated weighting schemes. In a recent study, Lahiri et al. (2017) documented that when combining the U.S. SPF forecasts using many of the methods also considered in this study, simple averaging remains one of the most effective combination methods. Elliott (2017) provides one explanation of this phenomenon that relates the size of common aggregate shocks to the potential gain from using optimal weights.

[2] See, among others, Stock and Watson (2004), Elliott and Timmermann (2005), Aiolfi and Timmermann (2006), Smith and Wallis (2009), Pesaran et al. (2013), and Tian and Anderson (2014).

[3] For example, Giraitis et al. (2013), Pesaran et al. (2013), Tian and Anderson (2014), and Chevillon (2016).

[4] In as early as 1989, Armstrong (1989) explicitly cited *realistic simulations* as one of three broad directions for future research, along with *meta-analysis* and *rule-based forecasting*.

In addition, we examine the performance of the nonparametric approach proposed in Bürgi and Sinclair (2017), where the authors suggested that only the forecasts with a proven track record of outperforming simple averaging should be combined. This approach eliminates the forecasters whose good performances are due to pure chance. Thus, it has the potential to outperform simple averaging over all the forecasters.

In standard simulation exercises, the pool of candidate forecasts to be combined are usually model-based.[5] Structural instabilities are introduced through misspecifications or breaks in model parameters. When using this approach, the sources of instabilities are usually clear, but the relative performances of the models are often not immediately apparent.[6] Real-world situations tend to be very different: Practitioners usually do not know the precise models behind individual forecasts (nor the reason for a sudden change in their accuracy), especially when working with survey forecasts or judgmental forecasts. However, from historical data, they do observe each individual forecaster's performance and how it changes over time. To mitigate this issue, we use a different approach to simulate unstable environments. Instead of generating candidate forecasts from misspecified models, we directly simulate the forecast *errors* by adding together three components: forecast bias, idiosyncratic error, and unpredictable aggregate shock. This way, we can create unstable environments in which one or more of these components vary cross-sectionally and over time. More importantly, having direct access to each component allows us to precisely control the accuracy of each individual's forecasts.

In the following sections, we conduct seven sets of simulation exercises, in which we simulate the effect of four broad types of instabilities that are likely present in real-world data: The performances of individual forecasters may change due to sudden breaks in their forecast biases or the variances of their forecast errors. Alternatively, the performances could change gradually and continuously. We also consider instabilities due to unpredictable aggregate shocks that affect the performances of all the forecasters. In addition, we let individual forecasters occasionally produce outliers, i.e., forecasts with unusually large errors.

We find that different combination algorithms excel in different kinds of unstable environments: Compared with the rest of the methods we examined in the paper, the one proposed in Sancetta (2010) is more robust to breaks in performances due to idiosyncratic errors; the AFTER algorithms are more robust to unpredictable and sudden aggregate shocks; and the approach in Bürgi and Sinclair (2017) is quicker in adapting to changes in individuals' performances. In addition, our results suggest a clear trade-off between the number of candidate forecasters and the accuracy of the combined forecasts. As the candidate pool grows, the performances of sophisticated algorithms rapidly deteriorate. For example, when combining 30 sets of equally accurate forecasts, the mean squared errors (MSE) of the combined forecasts produced by the AFTER algorithms may be 5–10 times higher than that of simple averaging.

Based on our observations, we believe that in order for the performance-based weighting algorithms to deliver robust performance gains, one should reduce the

[5] For example, simple linear regression models are used in Cheng and Yang (2015). Pesaran and Timmermann (2005), Sancetta (2010), and Chevillon (2016) used ARIMA models.

[6] For example, which model produces more accurate forecasts, one that omits a relevant predictor or one that includes an irrelevant one?

number of candidate forecasters and stabilize their performances before applying the algorithms. Thus, it is advisable to exclude forecasters that have persistent and known poor performances from the candidate pool and to group candidate forecasts with similar performances. The weighting algorithms can then be used to combine the "group consensus forecasts" calculated as simple averages of the forecasts in a group. Using grouped forecasts also help to reduce missing values, which is particularly beneficial when working with surveys such as the U.S. Survey of Professional Forecasters (SPF). In addition, we believe that an effective way to combat frequent regime changes is to limit the amount of historical data used to evaluate individual forecasters' performances. Such a limit could further improve the performance of the AFTER algorithms given their recursive design and the resulting long memory. We demonstrate the effectiveness of the above strategies by combining the forecasts of four important macroeconomic variables reported in the SPF in real time. Comparing our results and those reported in Lahiri et al. (2017), which looked into combining the same SPF variables using the same algorithms, we observe notably higher performance gains from the combination algorithms in our exercise.

The rest of the paper is organized as follows: Sect. 2 introduces the combination methods and algorithms. The setup of our simulation exercises is presented in Sect. 3 and the results in Sect. 4. In Sect. 5, we develop a forecast combination strategy based on the lessons learned from the simulations and use it to combine the SPF forecasts. Concluding remarks are in Sect. 6.

2 Combination methods

The combination methods and algorithms we focus on in the simulation exercises include the s-AFTER algorithm from Yang (2004), the L_{210}-AFTER algorithm from Cheng and Yang (2015),[7] the algorithm proposed in Sancetta (2010) (henceforth SAN), and the algorithm proposed in Bürgi and Sinclair (2017) (henceforth BS). For comparison, we also consider the recent best (RB) method and the method proposed in Bates and Granger (1969) (henceforth BG).

Consider a standard forecast combination exercise: After each release (y_t) of the target variable y becomes available, we first estimate a set of weights $0 \leq \omega_{j,t+h} \leq 1$, $j = 1, 2, \ldots, n$. Then, $\hat{y}_{t+h}$, the combined forecast of y_{t+h}, is calculated as the weighted average of n individual forecasts $\hat{y}_{j,t+h}$, $j = 1, 2, \ldots, n$. Without loss of generality, we set the forecast horizon h to 1. For an individual forecaster j, the most recent forecast error at time period $t + 1$ is $e_{j,t} \equiv y_t - \hat{y}_{j,t}$. Let $\hat{\sigma}^2_{j,t}$ be the estimated variance of j's errors. We start the forecast combination exercise at period t_o. The observations before t_o are used as training data where applicable. We measure the performance of combined forecasts using the mean squared errors (MSE). The MSE

[7] We use the term "algorithm" loosely and interchangeably with "methods," "approaches," and "procedures." Below, they are collectively referred to as the AFTER algorithms or the AFTERs. The L_1-AFTER algorithm and the h-AFTER algorithm from Wei and Yang (2012) also belong to the AFTER family. The L_1-AFTER uses the absolute error loss (or L_1 loss) and the h-AFTER uses the Huber loss. We omit the results on the performances of the latter two AFTER algorithms, since they are similar to that of the s-AFTER and L_{210}-AFTER, respectively.

of a series of combined forecasts changes over time, as more data become available. At each period t, the MSE is calculated over the period t_o to $t-1$. The MSE over the entire evaluation sample is calculated over the period t_o to T, where T is the sample size.

The AFTER algorithms have similar structures but differ in terms of their loss functions: The s-AFTER algorithm uses the squared error loss (or L_2 loss), and the L_{210}-AFTER algorithm uses a synthetic loss function—a mixture of L_2, L_1, and L_0 loss, as discussed below. When combining the U.S. SPF forecasts using s-AFTER, Lahiri et al. (2017) found that the performance of the algorithm is often driven by just a few large errors of the combined forecasts around the target variable's turning points. The latest addition to the AFTER family, L_{210}-AFTER, is designed to specifically address this issue. Since the L_0 loss imposes direct penalty on forecast outliers, the L_{210}-AFTER algorithm is more robust to them. Also, the combined forecasts produced by the algorithm tend to have fewer outliers.

The AFTER algorithms assign weights to individual forecasters recursively. Equal weights are used in the first period, t_o. In the case of s-AFTER, the weights for subsequent periods are calculated as

$$\hat{\omega}_{j,t+1}^{s-\text{AFTER}} = \frac{\hat{\omega}_{j,t}^{s-\text{AFTER}} \hat{\sigma}_{j,t}^{-1} \exp\left(-\frac{e_{j,t}^2}{2\hat{\sigma}_{j,t}^2}\right)}{\sum_{j=1}^{n} \left[\hat{\omega}_{j,t}^{s-\text{AFTER}} \hat{\sigma}_{j,t}^{-1} \exp\left(-\frac{e_{j,t}^2}{2\hat{\sigma}_{j,t}^2}\right)\right]} \text{ for } t \geq t_o + 1. \tag{1}$$

Weights assigned by the L_{210}-AFTER algorithm are given by

$$\hat{\omega}_{j,t+1}^{L_{210}-\text{AFTER}} = \frac{\hat{\omega}_{j,t}^{L_{210}-\text{AFTER}} \hat{\delta}_{j,t}^{-1/2} \exp\left(\frac{-L_{210}(e_{j,t})}{2\hat{\delta}_{j,t}}\right)}{\sum_{j=1}^{n} \left[\hat{\omega}_{j,t}^{L_{210}-\text{AFTER}} \hat{\delta}_{j,t}^{-1/2} \exp\left(\frac{-L_{210}(e_{j,t})}{2\hat{\delta}_{j,t}}\right)\right]} \text{ for } t \geq t_o + 1. \tag{2}$$

Following the suggestions in Cheng and Yang (2015), we set $\hat{\delta}_{j,t} = t^{-1} \sum_{l=1}^{t} L_{210}(y_l - \hat{y}_{j,l})$.[8] The synthetic loss function $L_{210}(\cdot)$, designed for outlier-protective combination, is defined as

$$L_{210}(x) = |x| + \alpha_1 \frac{x^2}{m} + \alpha_2 m \tilde{L}_0(x|\gamma_1 m, \gamma_2 m, r_1 r_2). \tag{3}$$

The parameters $\alpha_1, \alpha_2, \gamma_1, \gamma_2, r_1, r_2$, and m are set according to the suggestions in Cheng and Yang (2015): $a_1 = 1$, $a_2 = 1$, $g_1 = 2$, $g_2 = -2$, $r_1 = 0.75$, $r_2 = 0.75$, $m = 2$. Figure 1, a reproduction of Fig. 1 in Cheng and Yang (2015), shows the $\tilde{L}_0(\cdot)$ loss function, which is defined as

[8] See Remark 3 to Theorem 2 in Cheng and Yang (2015).

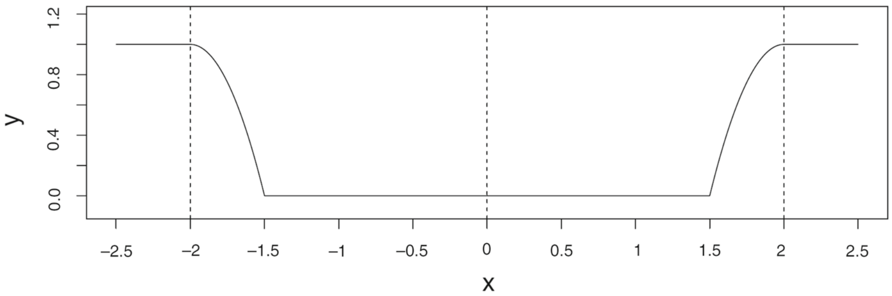

Fig. 1 $\tilde{L}_0(\cdot)$ loss function. This figure, a reproduction of Figure 1 in Cheng and Yang (2015), shows the $\tilde{L}_0(\cdot)$ loss function, which the L$_{210}$-AFTER algorithm uses along with the squared error and the absolute error loss

$$\tilde{L}_0(x) = \begin{cases} 1 & \text{if } x \geq \gamma_1 \text{ or } x \leq \gamma_2 \\ 1 - \frac{(x-\gamma_1)^2}{\gamma_1^2(1-r_1)^2} & \text{if } r_1\gamma_1 \leq x \leq \gamma_1 \\ 1 - \frac{(x-\gamma_2)^2}{\gamma_2^2(1-r_2)^2} & \text{if } \gamma_2 \leq x \leq r_2\gamma_2 \\ 0 & \text{if } \gamma_2 r_2 \leq x \leq \gamma_1 r_1 \end{cases}. \tag{4}$$

The SAN algorithm is implemented according to Algorithm 1 in Sancetta (2010). In each time period, based on the final weights from the previous period ($\omega_{j,t}^{\text{SAN}}$) and the loss incurred by the combined forecast from the previous period, $l_t(\omega_t^{\text{SAN}})$, a preliminary set of weights ($\omega_{j,t+1}^{\text{SAN}'}$) are calculated as

$$\omega_{j,t+1}^{\text{SAN}'} = \frac{\omega_{j,t}^{\text{SAN}} \exp\left[-\eta t^{-\alpha} \nabla_j l_t(\omega_{j,t}^{\text{SAN}})\right]}{\sum_{j=1}^{n} \left\{\omega_{j,t}^{\text{SAN}} \exp\left[-\eta t^{-\alpha} \nabla_j l_t(\omega_{j,t}^{\text{SAN}})\right]\right\}}, \tag{5}$$

where $\nabla l_t(\omega_t^{\text{SAN}})$ is the gradient of the loss function with respect to the previous period weight ω_t^{SAN}, and $\nabla_j l_t(\omega_t^{\text{SAN}})$ is its jth element. $\eta > 0$ and $\alpha \in (0, 1/2]$ are the parameters that control the responsiveness of the weight to changes in a forecaster's performance. We use two sets of values: $\{\alpha = 0.5, \eta = 0.3\}$ (subsequently labeled as SAN1) and $\{\alpha = 0.5, \eta = 0.7\}$ (subsequently labeled as SAN2). With a higher learning rate, SAN2 is more sensitive to new information (and noise). This should make it respond to individual forecasters' performance changes more quickly than SAN1 does. To obtain the set of final weights used in the combination, $\omega_{j,t+1}^{\text{SAN}}$, all of the preliminary weights ($\omega_{j,t+1}^{\text{SAN}'}$) that are lower than a predetermined small threshold γ/n are set to γ/n, and the remaining weights are scaled down such that all the weights sum to unity. We use $\gamma = 0.5$ in our implementations. This additional shrinkage may lead to suboptimal performance in a stable environment, where either one forecaster is clearly better than the rest, or when the optimal weights change only slowly. But shrinkage should help in unstable environments, where the optimal weights often change abruptly. In addition, as shown in Sancetta (2007), shrinking the weights toward equality helps to reduce outliers in combined forecasts.

The BS method identifies a subset of forecasters that frequently outperform the simple average and combines the forecasts of this subset of forecasters using equal weights. More specifically,

$$\omega^{BS}_{j,t+1} = \left\{ \sum_{i=1}^{n} \mathcal{I} \left[\left(\frac{1}{t - t_o + 1} \sum_{s=t_o}^{t} \mathcal{I}(\hat{\sigma}_{j,s} < \hat{\sigma}^{SA}_{j,s}) \right) > p \right] \right\}^{-1}, \tag{6}$$

where $\mathcal{I}(\cdot)$ is the indicator function and $\hat{\sigma}^{SA}$ is the root mean squared error of the simple average forecast. p is the proportion of times one must outperform the simple average in order to be included in the subset. In our exercises we set $p = 0.5$.

The BG method sets $\omega^{BG}_{j,t+1} = \hat{\sigma}^{-2}_{j,t} / \sum_{i=1}^{n} \hat{\sigma}^{-2}_{i,t}$. The recent best (RB) method identifies the best forecaster from the previous period and uses his forecast as the combined forecast, i.e., $\omega^{RB}_{j,t+1} = 1$ if $e^2_{j,t} = \min\{e^2_{1,t}, e^2_{2,t}, \ldots, e^2_{n,t}\}$ and $\omega_{j,t+1} = 0$ otherwise.

Where applicable, we estimate the variance of individual j's forecast errors using a rolling window of size w, i.e., $\hat{\sigma}^2_{j,t} = w^{-1} \sum_{\tau=1}^{w} e^2_{j,t-\tau+1}$. Since we are working in unstable environments, the need for a limited window size naturally arises. The algorithms introduced above use very different strategies when they calculate the performance of individual forecasters using historical data. In one extreme, RB bases this calculation on nothing more than the most recent forecast error. In the other extreme, BG weighs the most recent forecast error the same as all previous errors. For the recursive algorithms such as the AFTERs, information from before $t - w + 1$ is carried over through the use of previous weights. But the impact of older forecast errors on current weights is less than that of more recent errors. Finally, when simple averaging is used to combine forecasts, past performances are not utilized at all. As discussed below, these variations in how historical data are used may be important in driving the algorithms' adjustments to breaks in the performances of individual forecasters.

3 Simulation setup

Let the actual value be the sum of a predictable component s_t and an unpredictable component, i.e., common aggregate shock, c_t:

$$y_t = s_t + c_t, \;\; c_t \sim \mathcal{N}(0, 1). \tag{7}$$

We decompose individual j's forecast of y_t into three components:

$$\hat{y}_{j,t} = s_t + b_{j,t} + \varepsilon_{j,t}, \;\; \varepsilon_{j,t} \sim \mathcal{N}(0, \sigma^2_{j,t}). \tag{8}$$

s_t enters the forecast since it is by definition predictable. $b_{j,t}$ represents potential bias in the forecast. $\varepsilon_{j,t}$ is the remaining part of the forecast, which is constrained to have zero mean so as not to cause any more bias. It becomes the only idiosyncratic part of the forecast error when we assume unbiasedness. $\sigma^2_{j,t}$ is the variance of $\varepsilon_{j,t}$. The forecast error, therefore, depends on the unpredictable common aggregate shock c_t, the forecast bias $b_{j,t}$, and the remainder error $\varepsilon_{j,t}$:

$$e_{j,t} \equiv y_t - \hat{y}_{j,t} = c_t - b_{j,t} - \varepsilon_{j,t}. \tag{9}$$

The intuition behind this decomposition is similar to that discussed in Davies et al. (2011), where c_t is affected by aggregate uncertainty and $b_{j,t}$ is the source of disagreement. Note that since s_t is not a part of the forecast error, no assumption needs to be made about its properties.

In all subsequent exercises, $n \in \{5, 30\}$, $t_o = 61$, $T = 300$, and $w \in \{24, 60\}$. These parameter choices allow us to recreate many familiar situations. For example, when $n = 5$, we work as if we are to combine the forecasts from a small set of models. When $n = 30$, the setup is similar to what used when combining survey forecasts. A short window of $w = 24$ periods is often the preferred choice in highly unstable environments, while $w = 60$ is suitable for more stable environments.

For each set of values of $\{n, t_o, T, w\}$, the simulation is carried out as follows:

1. Draw $\{b_{j,t}, \sigma^2_{j,t}\}$ $\forall j$ according to the specification of the simulation exercise. Details on the setup of each exercise are presented below.
2. For each t, draw c_t, draw $\varepsilon_{j,t}$ $\forall j$ given $\sigma^2_{j,t}$, and calculate $\hat{\sigma}^2_{j,t}$.
3. Apply the combination algorithms presented in the previous section to generate combined forecasts. Record the MSE of the combined forecasts produced by each algorithm.
4. Repeat Step 2 to Step 3 200 times. For each algorithm, obtain the average MSE across these repetitions. These MSEs are conditional on the specific draws of $b_{j,t}$ and $\sigma^2_{j,t}$.
5. Repeat Step 1 to Step 4 1000 times. For each algorithm, obtain the average MSE across these repetitions. These MSEs are no longer conditional on specific draws of $b_{j,t}$ and $\sigma^2_{j,t}$.

When presenting simulation results, we report relative MSEs. The relative MSE of an algorithm is its MSE from the last step divided by that of simple averaging. A relative MSE bigger than one means that the algorithm produces combined forecasts that are less accurate than the simple averages of individual forecasts. In the next section, we report the results of seven sets of simulation exercises. In each exercise, we consider a different data generating process for $\{b_{j,t}, \sigma^2_{j,t}\}$. Other than in Exercise 1, where we consider a stable environment, we set up four scenarios within each exercise, varying the magnitude of the impact of instabilities on forecasters' performances. When we simulate forecast biases or outliers, for convenience, we only generate positive values. There is no need to specifically allow for negative biases or forecast errors, since the loss functions used by all of the algorithms are symmetric.

Note that while we attempt to set up the simulation exercises in ways that are relevant to combination exercises in practice, we do not try to replicate the features of any particular data set. When the objective is to identify the most suitable algorithm for a given data set, one can simply use it rather than trying to simulate its features. We choose the parameters in each simulation exercise so that the impact of instabilities on the performances of different algorithms is easily observable, provided that our choices do not result in simulated data sets that are almost impossible to encounter in real life.[9] In addition, even though we consider several combination algorithms and report their relative MSEs, we do not intend to run a horse race and proclaim the "best," neither do we attempt to improve upon the standard algorithms by experimenting with small changes to their implementations. All of the algorithms considered here have been used in the literature and shown to have the potential to outperform simple averaging. Instead of fixating on how an algorithm compares to this benchmark, we emphasize its performances (and how they differ) in different scenarios. Our hope is to better understand the respective strengths of the algorithms. This knowledge would help us identify a more effective forecast combination procedure, i.e., more robust in unstable environments, than blindly applying any one particular algorithm.

4 Simulation results

4.1 Combining unbiased and homoscedastic forecasts in a stable environment

In exercise 1, we look at the performances of the combination algorithms in a stable environment, in which all the forecasts are unbiased and homoscedastic. In this environment, the optimal weights are equal weights. Here, $b_{j,t} = 0$ and $\sigma^2_{j,t} = \sigma^2$ $\forall j, t$. While holding the variance of aggregate shocks fixed at 1, we consider a set of forecast error variances that are progressively larger: $\sigma^2 \in \{0.2, 0.4, 0.6, \ldots, 20\}$—that is, we consider environments where the target variable becomes more and more difficult to predict. Whenever we have to estimate individual-specific weights, there are estimation errors. These errors may cause the combined forecasts to be less accurate than a simple average of individual forecasts. This exercise allows us to clearly reveal this cost.

Figure 2 plots the relative MSEs against σ^2, showing how quickly the cost of estimation increases as the target variable becomes harder to forecast, i.e., everyone's forecasts are less accurate. Since BS does not estimate weights like, e.g., the AFTERs, its performance does not deteriorate as σ^2 increases. The RB method in this environment amounts to randomly choosing a forecaster each period, since no one is systematically better. It is thus reassuring to see that RB does not outperform any other method. The relative MSEs of s-AFTER and L_{210}-AFTER increase at a much higher rate than those of BG, SAN1, and SAN2. When the number of forecasts to be combined is 30 instead of 5, the relative MSEs of s-AFTER and L_{210}-AFTER become almost three times bigger. To the contrary, the relative MSE of BG stays roughly the same, and the relative MSEs of SAN1 and SAN2 decrease slightly. We do not see any significant difference in the results when we increase the window size from 24 to 60.

As we observe here, even though the environment is stable, estimating a large number of weights may become so costly that the resulting combined forecasts are much less accurate than what simple averaging offers. For all of the algorithms, this

[9] In fact, we carefully compared the parameters of the simulation exercises with relevant characteristics of the SPF data used in Sect. 5. The boundaries of the distributions of b and σ used in the simulations are largely consistent with their counterparts in the SPF data. Detailed results omitted but available from the author.

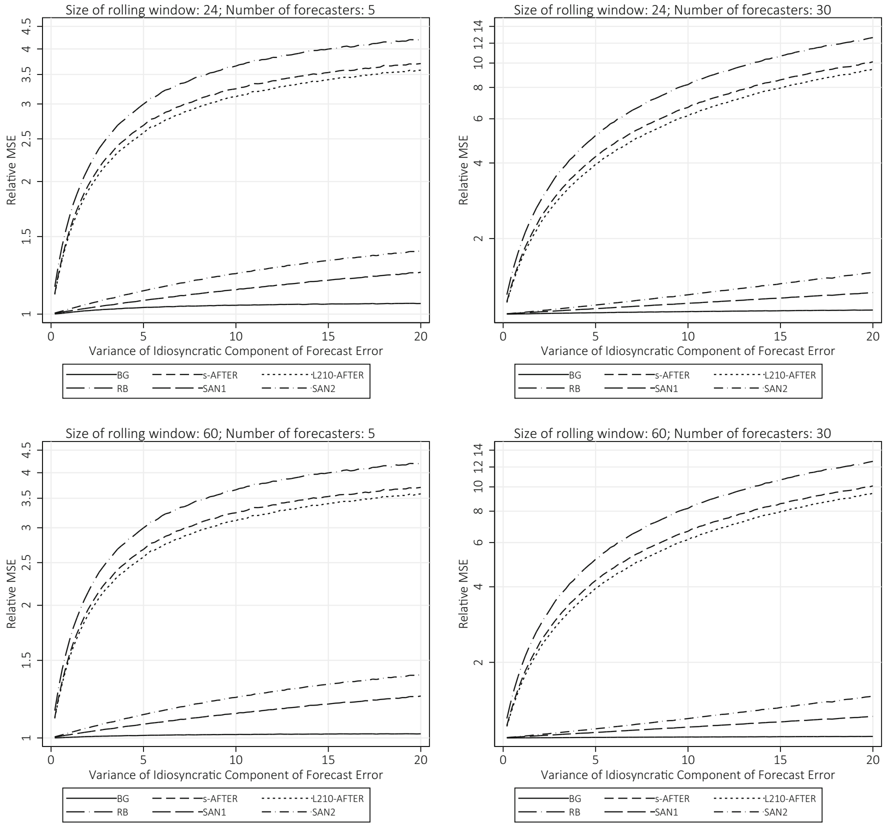

Fig. 2 Exercise 1 Results. This figure shows the relative MSEs of various combination algorithms when combining unbiased and homoscedastic forecasts that are increasingly inaccurate

cost increases as the target variable becomes more difficult to forecast.[10] For the AFTER algorithms, the cost also increases significantly as the number of forecasters increases.

4.2 Combining biased forecasts in a stable environment

In exercise 2, we combine biased, but homoscedastic, forecasts in a stable environment. For each forecaster, the amount of bias and the variance of the forecast errors remain constant over time: $b_{j,t} = b_j \ \forall t$ and $\sigma^2_{j,t} = \sigma^2 \ \forall j, t$. The optimal weights should be such that the forecaster with the smallest amount of bias receives a weight of one, while all other forecasters receive zero weight. In this situation, the performance-based weighting algorithms should perform well.

[10] Although in both cases, simple averaging is found to be difficult to improve upon, our result is not the same as those reported in Elliott (2017). We consider uncorrelated forecasts with large errors, whereas Elliott (2017) considered highly correlated forecasts due to the errors having a large common component.

Table 1 Relative MSEs: Exercise 2

Algorithm	Exercise and Number of Forecasters (n)							
	Exercise 2.1		Exercise 2.2		Exercise 2.3		Exercise 2.4	
	$n = 5$	$n = 30$	$n = 5$	$n = 30$	$n = 5$	$n = 30$	$n = 5$	$n = 30$
Windows size: 24								
BG	0.991	0.977	1.022	0.998	0.964	0.949	0.994	0.974
BS	1.169	1.244	1.155	1.380	0.984	0.835	1.240	1.354
L_{210}-AFTER	1.348	1.402	2.212	2.931	0.968	0.898	1.504	1.592
RB	1.574	1.788	2.579	3.844	1.192	1.229	1.787	2.141
s-AFTER	1.366	1.440	2.288	3.124	0.974	0.913	1.535	1.663
Windows size: 60								
BG	0.986	0.976	1.005	0.993	0.963	0.949	0.987	0.972
BS	1.118	1.227	1.026	1.022	0.943	0.799	1.169	1.363
L_{210}-AFTER	1.347	1.400	2.209	2.923	0.967	0.897	1.503	1.590
RB	1.574	1.788	2.579	3.844	1.192	1.229	1.787	2.141
s-AFTER	1.365	1.438	2.286	3.119	0.974	0.912	1.534	1.661
Windows size: Not applicable								
SAN1	0.978	0.930	1.049	1.000	0.922	0.858	1.010	0.940
SAN2	0.993	0.944	1.093	1.034	0.938	0.875	1.042	0.976

This table shows the relative MSEs of various combination algorithms when combinaing biased but homoskedastic forecasts. In exercise 2.1, both the bias and the forecast error variance are small. In exercise 2.2, the bias remains small but the variance is large. In exercise 2.3, the bias is large and the variance is small. In exercise 2.4, both the bias and the variance are large

We consider four scenarios with varying magnitudes of bias and forecast error variance. Exercise 2.1 features small biases and a small variance: b_j is uniformly distributed over the interval (0, 1) and $\sigma^2 = 1$. In exercise 2.2, the distribution of the bias term b_j is the same as in 2.1, but the variance is four times bigger: $\sigma^2 = 4$. In exercise 2.3, we look at large biases coupled with a small variance: $b_j \sim \mathcal{U}(1, 2)$ and $\sigma^2 = 1$. Both terms are large in exercise 2.4, with $b_j \sim \mathcal{U}(1, 2)$ and $\sigma^2 = 4$.

Note that the boundaries of the distributions should be interpreted relative to the variance of the forecast errors and the unpredictable component of the actual values. While in real-world data sets like the SPF, distributions of biases or variances are often bell-shaped, we use the uniform distribution in the simulation exercises here and below. This is to make sure that we have a variety of values even with only five forecasters. Using the uniform distribution also helps to create sufficiently large differences among the forecasters so that there is a non-trivial potential for improvement through combination.

Table 1 shows the results from this exercise. As expected, the performance-based weighting algorithms perform well.[11] Comparing exercise 2.1 with 2.2 and comparing 2.3 with 2.4, we see that the combined forecasts become less accurate as forecast errors

[11] The fact that many of the relative MSEs are larger than 1 is not a concern. We can easily lower these relative MSEs by changing the parameters of the data generating process (in this exercise, increasing the biases). The same applies to subsequent simulation exercises.

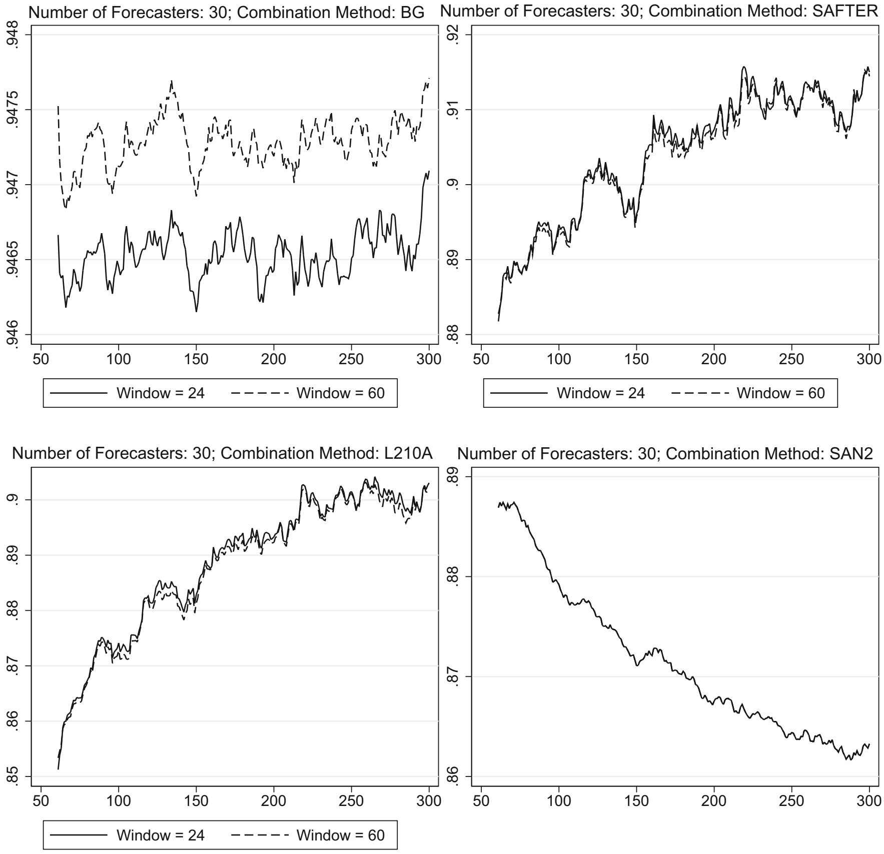

Fig. 3 Moving averages of relative MSEs: Exercise 2.3. This figure shows the 12-period moving average of relative MSEs of various combination algorithms when combining biased but homoscedastic forecasts, where the bias is large relative to the error variance

become more variable, making estimating individuals' performances more difficult. Consistent with our observations in exercise 1, this effect of increasing forecast error variances is particularly pronounced for the AFTERs but very subtle for both SAN1 and SAN2. Comparing exercise 2.1 with 2.3 and comparing 2.2 with 2.4, we see that the larger the biases, the more accurate the combined forecasts. In particular, as shown in exercise 2.3, when combining forecasts with large biases, algorithms such as the AFTERs may perform better despite having to combine a larger number of forecasts.

Figure 3 shows the 12-period moving averages of the relative MSEs of BG, s-AFTER, L_{210}-AFTER, and SAN2 when combining 30 forecasts in exercise 2.3. We can clearly see how the performances of these algorithms change over time, as historical data accumulate. In the case of BG, the performance of combined forecasts barely changes (note the scale of the vertical axis). However, this is not the case for the other three algorithms. For the two AFTER algorithms, the combined forecasts become less accurate as more data become available, although the deterioration is only 3% to 5%.

For SAN, the opposite is observed. But still, we are looking at a difference less than 3%. These observations suggest that one may benefit more from the AFTER algorithms when applying them to shorter samples. Alternatively, one may "reset" the algorithm periodically by discarding historical data that are too old (or simply reset the weights to equal). The performances of all the algorithms eventually stabilize after about 250 periods (which is a long time even when the data are monthly). But in practice, there may not be nearly as many periods before a break in the forecasters' performances that change their optimal weights. In the exercises below, we start to build structural instabilities into the simulations.

4.3 Combining biased forecasts with breaks in performances

In exercise 3, we destabilize the environment in exercise 2 with the simplest form of structural break—a one-time mean shift. More specifically, for each forecaster, we make a one-time change to the bias halfway in the sample. Let $b_{j,t} = b_{j,1}$ for $t \leq 181$ and $b_{j,t} = b_{j,181}$ for $t \geq 181$; $\sigma^2_{j,t} = \sigma^2 \ \forall j, t$. We consider the same set of forecast bias/error variance combinations as used previously: In exercise 3.1, both the biases and the variance are small: $b_{j,1}, b_{j,181} \sim \mathcal{U}(0, 1)$ and $\sigma^2 = 1$. In exercise 3.2, the bias remains small but the variance is large: $b_{j,1}, b_{j,181} \sim \mathcal{U}(0, 1)$ and $\sigma^2 = 4$. In exercise 3.3, we consider large biases and a small variance: $b_{j,1}, b_{j,181} \sim \mathcal{U}(1, 2)$ and $\sigma^2 = 1$. Finally, in exercise 3.4, both the biases and the variance are large: $b_{j,1}, b_{j,181} \sim \mathcal{U}(1, 2)$ and $\sigma^2 = 4$. Note that the one-time break in forecast bias amounts to simply redraw the bias term.[12] The distribution of the bias term does not change. In particular, it is possible for the pre- and post-break levels of forecast bias to be very similar.

Table 2 presents the results. As expected, for all the algorithms, even with a one-time break in individual forecasters' performances, combined forecasts are more accurate when the forecast error variances are small and/or the biases are large. Compared with the previous exercise, we see minimal deterioration in the performance of the combined forecasts.

However, we should not conclude that all the algorithms are robust to the kind of breaks considered here. A closer look at how the algorithms' performances change over time provides additional insights: Fig. 4 shows the 12-period moving averages of the relative MSEs of BG, L_{210}-AFTER, SAN1, and SAN2 with $n = 30$ in exercise 3.3. From the figure, we can clearly see the effect of the break. The relative MSEs of the combined forecasts increase immediately after the break hits, before they gradually decline as more post-break data become available. With over 30% increase in relative MSE, L_{210}-AFTER suffers the biggest loss from the break. The performances of SAN1 and SAN2 deteriorated by 13% and 9%, respectively. The difference between SAN1 and SAN2 is as expected. With a higher learning rate, SAN2 should adapt to the post-break environment more quickly than SAN1 does, at the cost of having slightly worse

[12] Here, we consider a sudden change in forecast performances, which, in reality, could be due to a change in the underlying forecasting model or a personnel change in the forecast institution. The situation where individuals' performances steadily improve/decline is considered in exercise 5 below.

Table 2 Relative MSEs: Exercise 3

Algorithm	Exercise and number of forecasters (n)							
	Exercise 3.1		Exercise 3.2		Exercise 3.3		Exercise 3.4	
	$n = 5$	$n = 30$	$n = 5$	$n = 30$	$n = 5$	$n = 30$	$n = 5$	$n = 30$
Windows size: 24								
BG	0.992	0.978	1.022	0.999	0.966	0.952	0.995	0.976
BS	1.170	1.249	1.155	1.380	0.991	0.846	1.241	1.362
L_{210}-AFTER	1.398	1.457	2.233	2.975	1.059	0.986	1.567	1.682
RB	1.573	1.788	2.578	3.842	1.192	1.230	1.786	2.141
s-AFTER	1.417	1.494	2.310	3.170	1.065	1.002	1.600	1.754
Windows size: 60								
BG	0.988	0.979	1.006	0.994	0.968	0.956	0.989	0.976
BS	1.118	1.237	1.027	1.022	0.966	0.825	1.169	1.381
L_{210}-AFTER	1.397	1.453	2.231	2.967	1.058	0.984	1.566	1.679
RB	1.573	1.788	2.578	3.842	1.192	1.230	1.786	2.141
s-AFTER	1.415	1.492	2.309	3.165	1.064	0.999	1.599	1.751
Windows size: Not applicable								
SAN1	0.987	0.954	1.050	1.007	0.934	0.885	1.012	0.949
SAN2	0.995	0.954	1.093	1.036	0.942	0.886	1.042	0.979

This table shows the relative MSEs of various combination algorithms when combining biased but homoscedastic forecasts. There is a one-time random change in the magnitude of the bias in period 181. In exercise 3.1, both the bias and the forecast error variance are small. In exercise 3.2, the bias remains small, but the variance is large. In exercise 3.3, the bias is large and the variance is small. In exercise 3.4, both the bias and the variance are large

performance in stable environments. Among all the algorithms we implemented, BG is the least impacted, with only 4% increase in relative MSE.

A natural response to frequent structural breaks is to use a shorter window of historical data. While this strategy is clearly helpful when the BG method is used, it does little to speed up the adaptation of L_{210}-AFTER to the post-break environment. In fact, by the end of our sample period, the performances of BG, SAN1, and SAN2 have reached their pre-break level, while the relative MSE of L_{210}-AFTER remains 10% worse. Again, the results suggest the need for periodically resetting the AFTER algorithms in unstable environments. More generally, a more aggressive algorithm may work better during unstable periods, but it may also be over-sensitive in stable environments.

4.4 Combining heteroskedastic forecasts with breaks in performances

In the previous two exercises, individual forecasts are assumed to be biased. Even though this makes it convenient for us to vary individuals' performances, having a significant amount of bias in all the forecasts is somewhat unrealistic. Therefore, from this exercise onward, the assumption of unbiasedness will be maintained.

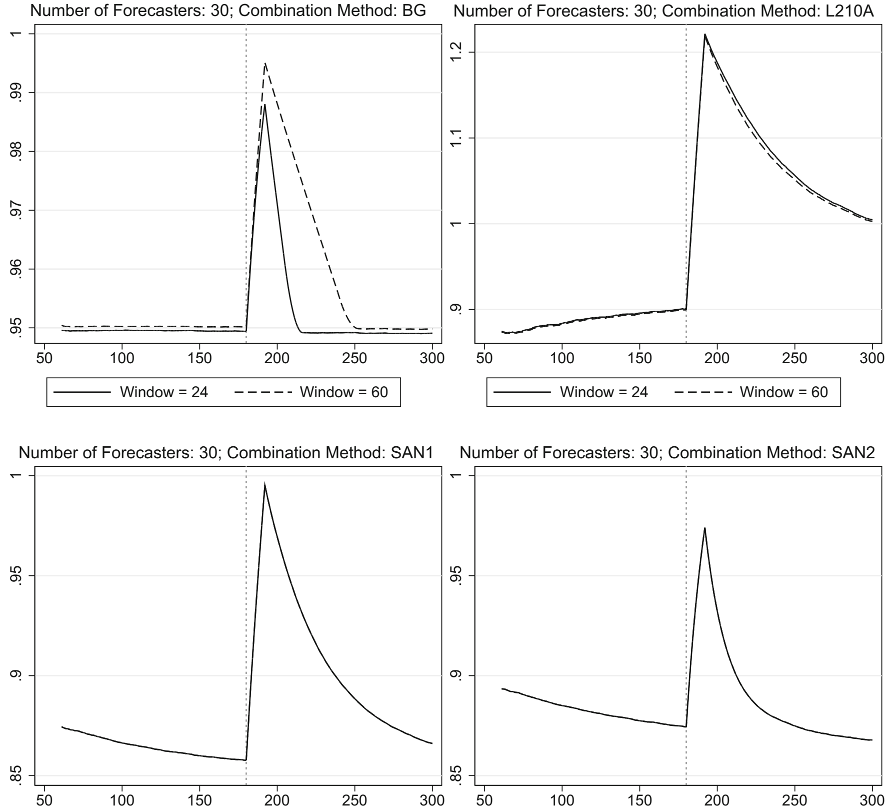

Fig. 4 Moving averages of relative MSEs: Exercise 3.3. This figure shows the 12-period moving average of relative MSEs of various combination algorithms when combining biased but homoscedastic forecasts. There is a one-time random change in the magnitude of the bias in period 181

In exercise 4, we work with unbiased but heteroskedastic forecasts. In this exercise, individual forecasters' performances depend solely on the variance of the idiosyncratic component of their forecast errors. Forecasts with a smaller variance are preferable and should receive higher weights. Instabilities are introduced through breaks in individuals' forecast performances. More specifically, in this exercise, $b_{j,t} = 0 \ \forall j, t$, and $\sigma^2_{j,t} = \sigma^2_{j,r}$ if $\delta_{r-1} < t \leq \delta_r$, where $r = 1, 2, \ldots, R$ indexes regimes, and δ_r is the time when regime r ends. $\delta_0 = 0$, $\delta_R = T$. In exercise 4.1 and 4.2, we introduce only one break: $R = 2$, $\delta_1 = 180$. The difference between exercise 4.1 and 4.2 lies in the variability of individuals' forecast performances: In exercise 4.1, the performances are less variable, $\sigma^2_{j,r} \sim \mathcal{U}(0.1, 2.5)$, while in exercise 4.2, we allow much more variability with $\sigma^2_{j,r} \sim \mathcal{U}(0.1, 6.5)$. There are three breaks in exercises 4.3 and 4.4, where we have $R = 4$ and $\delta_1 = 90$, $\delta_2 = 150$, $\delta_3 = 210$). The variability of individuals' performances is low in exercise 4.3, where $\sigma^2_{j,r} \sim \mathcal{U}(0.1, 2.5)$, and high in exercise 4.4, where $\sigma^2_{j,r} \sim \mathcal{U}(0.1, 6.5)$.

Table 3 Relative MSEs: Exercise 4

Algorithm	Exercise and number of forecasters (n)							
	Exercise 4.1		Exercise 4.2		Exercise 4.3		Exercise 4.4	
	$n = 5$	$n = 30$	$n = 5$	$n = 30$	$n = 5$	$n = 30$	$n = 5$	$n = 30$
Windows size: 24								
BG	0.971	0.993	0.912	0.971	0.976	0.994	0.930	0.976
BS	1.121	1.251	1.086	1.275	1.139	1.280	1.124	1.344
L_{210}-AFTER	1.406	1.373	1.766	1.891	1.548	1.620	2.068	2.532
RB	1.762	2.175	2.348	3.542	1.768	2.179	2.361	3.552
s-AFTER	1.412	1.377	1.776	1.884	1.561	1.638	2.088	2.557
Windows size: 60								
BG	0.968	0.992	0.911	0.971	0.981	0.995	0.950	0.984
BS	1.075	1.179	1.048	1.160	1.119	1.250	1.137	1.277
L_{210}-AFTER	1.403	1.365	1.761	1.852	1.546	1.615	2.060	2.502
RB	1.762	2.175	2.348	3.542	1.768	2.179	2.361	3.552
s-AFTER	1.409	1.367	1.769	1.836	1.558	1.631	2.079	2.520
Windows size: Not applicable								
SAN1	0.983	1.017	0.942	1.038	0.998	1.015	0.968	1.035
SAN2	0.993	1.030	0.986	1.086	1.003	1.026	0.999	1.080

This table shows the relative MSEs of various combination algorithms when combining unbiased but heteroskedastic forecasts. In exercises 4.1 and 4.2, there is a one-time random change in each individual's forecast error variance in period 181. In exercises 4.3 and 4.4, there are three changes in periods 91, 151, and 211. In exercises 4.1 and 4.3, individuals' performances are less variable. In exercises 4.2 and 4.4, the performances are more variable

In Table 3, we report the results of this exercise. Comparing exercises 4.1 with 4.3 and comparing exercise 4.2 with 4.4, we see that having three breaks instead of one only makes the performances of the combination algorithms slightly worse. Changing the variability of individuals' performances, on the other hand, has a much bigger effect on the accuracy of the combined forecasts. Regardless of whether there are 5 or 30 forecasters, BG and BS tend to perform better when individuals' performances have higher levels of variability, as in exercises 4.2 and 4.4. But at the same time, RB and the two AFTER algorithms perform worse in cases with high levels variability. Interestingly, the number of forecasters affects the performance of SAN1 and SAN2 more than the level of variability. With fewer forecasters, they actually perform slightly better in the higher variability cases.[13] This is likely due to how the SAN method shrinks the differences between individuals' weights. When there are 5 forecasters, with $\gamma = 0.5$, the smallest weight is 0.1, while it is only 0.017 when there are 30 forecasters. In unstable environments, the shrinkage step effectively helps SAN to avoid producing combined forecasts that are significantly worse than the mean of the individual forecasts.

[13] Note that in terms of MSEs (not relative MSEs), both SAN1 and SAN2 perform worse in the higher variability cases as expected.

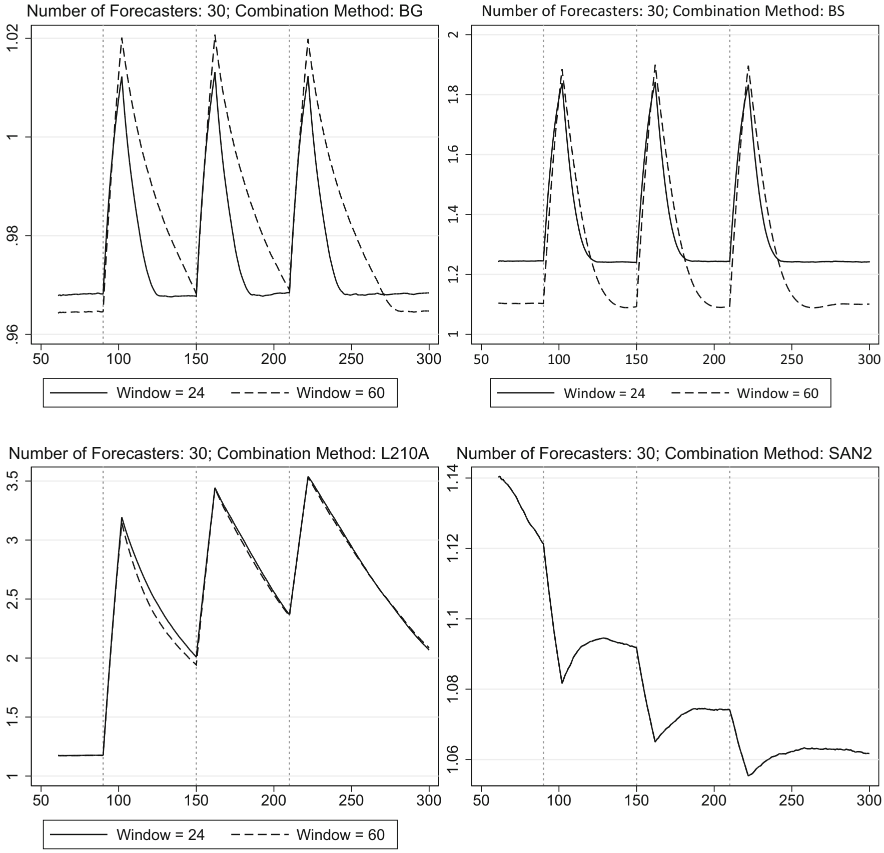

Fig. 5 Moving averages of relative MSEs: Exercise 4.4. This figure shows the 12-period moving average of relative MSEs of various combination algorithms when combining unbiased but heteroskedastic forecasts. There are three random changes in each individual's forecast error variance in periods 91, 151, and 211

Next, we look more closely at exercise 4.4, where individuals' performances are highly variable and they change three times during the sample period. Figure 5 shows the performances of BG, BS, L_{210}-AFTER, and SAN2. Much like what we saw in the previous exercise, BG adapts to new regimes quickly after breaks, especially when the window size is small. BS adapts to new regimes even more quickly than BG does, regardless of the window size. The slow adaptation and long memory of the AFTER algorithms clearly affect their performances: The impact of breaks accumulates over time, making combined forecasts increasingly unreliable. In situations with more frequent breaks and shorter regimes, the AFTERs may suffer even heavier losses. This behavior is the opposite of that of SAN2, where shrinkage helps to reduce the impact of breaks, so much so that the combined forecasts become more accurate after each of the three breaks (but still slightly worse than simple averages of individual forecasts).

Exercise 4 also illustrates the robustness of simple averaging. In none of the four setups is simple averaging the optimal combination method. However, as we can see from Table 3, the only algorithm that delivers better performance across exercises 4.1 to 4.4 is BG. Both SAN1 and SAN2, as well as BS, behave similarly to simple averaging. The AFTERs and RB perform notably worse.

4.5 Combining heteroskedastic forecasts with continuously changing performances

So far, we have only simulated instabilities in the form of discrete changes in forecast performances. In this exercise, we shift our focus to unstable environments in which the accuracy of individuals' forecasts is continuously changing. This type of environments is arguably more realistic, since it is unlikely that a forecaster's performance stays exactly the same over a number of years.

We continue to maintain the assumption of unbiasedness and set $b_{j,t} = 0 \; \forall j, t$. For performance-based weighting algorithms to work, a forecaster's performance in the future must at least in part depend on its past. When the performances change continuously, we cannot allow the changes to be completely random. Let

$$\sigma_{j,t}^2 = \sigma_{j,1}^2 + \frac{\sigma_{j,T}^2 - \sigma_{j,1}^2}{T-1} \cdot (t-1), \tag{10}$$

where $\sigma_{j,1}^2 \sim \mathcal{U}(p_1, q_1)$ and $\sigma_{j,T}^2 \sim \mathcal{U}(p_T, q_T)$. p_1, q_1, p_T, and q_T are parameters to be specified below. This way, for each forecaster, while his performances in the first period and the last period are randomly chosen, the way his performance changes over time is predictable and the speed of the change is constant.

In exercise 5.1, changes happen to all the forecasters: $p_1 = p_T = 0.1$ and $q_1 = q_T = 6.5$. In exercise 5.2, with probability 0.5, $p_1 = p_T = 0.1, q_1 = q_T = 3.5$. with probability 0.5, $p_1 = p_T = 3.5, q_1 = q_T = 6.5$. In this setting, with equal probability, the performance of a forecaster can either be good or poor. While everyone's performance gradually changes, nobody leaves his group: A good forecaster will always be a good forecaster, and a poor forecaster will always remain so. Obviously, the optimal weighting scheme should place no weight on the poor forecasters, while the weights of the good forecasters change over time according to changes in their performances. In exercise 5.3, with probability 0.5, $p_1 = 0.1, q_1 = 3.5, p_T = 3.5, q_T = 6.5$. with probability 0.5, $p_1 = 3.5, q_1 = 6.5, p_T = 0.1, q_T = 3.5$. This setting is similar to the that of exercise 5.2, except that, instead of remaining in the same group, everyone slowly moves into the other group. Eventually, the group of good forecasters become the group of poor forecasters, vice versa. In the process, the relative performances within a group may change as well. In exercise 5.4, with probability 0.5, $p_1 = 0.1, q_1 = 6.5$ and $\sigma_{j,T}^2 = \sigma_{j,1}^2$. With probability 0.5, $p_1 = p_T = 0.1$ and $q_1 = q_T = 6.5$. In this setting, the performances of half of the forecasters change continuously over time, while the remaining forecasters' performances remain stable.

We report the results of this exercise in Table 4. As expected, the more widespread the breaks, the worse the performances of combination algorithms. In exercise 5.1, everyone's performance changes. In exercise 5.4, only half the forecasters' performances change over time. Comparing their results, we can see that almost all the

Table 4 Relative MSEs: Exercise 5

Algorithm	Exercise and number of forecasters (n)							
	Exercise 5.1		Exercise 5.2		Exercise 5.3		Exercise 5.4	
	$n = 5$	$n = 30$	$n = 5$	$n = 30$	$n = 5$	$n = 30$	$n = 5$	$n = 30$
Windows size: 24								
BG	0.957	0.985	1.006	1.002	1.011	1.004	0.931	0.977
BS	1.133	1.348	1.137	1.295	1.151	1.351	1.102	1.294
L$_{210}$-AFTER	1.722	1.704	2.053	2.557	2.108	2.691	1.510	1.441
RB	2.453	3.665	2.557	3.841	2.588	3.910	2.393	3.603
s-AFTER	1.738	1.718	2.098	2.661	2.146	2.779	1.522	1.450
Windows size: 60								
BG	0.944	0.981	0.990	0.997	0.995	0.998	0.919	0.973
BS	1.054	1.103	1.041	1.087	1.039	1.043	1.029	1.101
L$_{210}$-AFTER	1.721	1.700	2.051	2.553	2.106	2.686	1.509	1.439
RB	2.453	3.665	2.557	3.841	2.588	3.910	2.393	3.603
s-AFTER	1.736	1.715	2.096	2.658	2.144	2.776	1.521	1.448
Windows size: Not applicable								
SAN1	0.977	1.036	1.031	1.036	1.035	1.036	0.951	1.038
SAN2	1.027	1.074	1.075	1.064	1.078	1.062	1.003	1.082

This table shows the relative MSEs of various combination algorithms when combining unbiased but heteroskedastic forecasts, where the forecast error variances change continuously. In exercise 5.1, this change is random and it happens to all the forecasters. In exercises 5.2 and 5.3, initially, half the forecasters are in the "good" group and the other half are in the "poor" group. In exercise 5.2, despite the random performance changes, forecasters always stay in the same group. In exercise 5.3, the random performance changes eventually result in every forecaster switching to the other group. In exercise 5.4, the performances of half the forecasters change randomly, while those of the other half remain constant

algorithms perform better in exercise 5.4, regardless of how many forecasters are in the pool. A comparison between exercise 5.2 and 5.3 leads to similar observations. However, SAN1 and SAN2 are the exceptions, most likely due to the shrinkage step as discussed previously.[14]

4.6 Combining heteroskedastic forecasts subject to unexpected aggregate shocks

Next, we examine the effect of unexpectedly large aggregate shocks. Unlike breaks in individuals' performances, aggregate shocks are the same to all the individuals. They do not affect an individual's performance, but they do make estimating the performance more difficult. We continue to work with unbiased but heteroskedastic forecasts. Apart from aggregate shocks, there is no other instabilities or breaks in performances. The optimal weighting scheme should place all the weights on the forecasts with the lowest variance. Specifically, we set $b_{j,t} = 0 \; \forall j, t$ and $\sigma^2_{j,t} = \sigma^2_j \sim \mathcal{U}(0.1, 6.5)$. In all the previous exercises, c_t is simply a standard normal random variable, see equation (7).

[14] Again, when we examine the MSEs instead of the relative MSEs, both SAN1 and SAN2 behave as expected.

Table 5 Relative MSEs: Exercise 6

Algorithm	Exercise and number of forecasters (n)							
	Exercise 6.1		Exercise 6.2		Exercise 6.3		Exercise 6.4	
	$n = 5$	$n = 30$	$n = 5$	$n = 30$	$n = 5$	$n = 30$	$n = 5$	$n = 30$
Windows size: 24								
BG	0.938	0.982	0.974	0.994	0.952	0.987	0.986	0.997
BS	1.070	1.185	1.071	1.137	1.059	1.146	1.050	1.090
L$_{210}$-AFTER	1.217	1.109	1.121	1.052	1.180	1.085	1.082	1.033
RB	2.028	2.715	1.609	1.910	1.857	2.357	1.420	1.602
s-AFTER	1.225	1.113	1.128	1.059	1.187	1.090	1.088	1.041
Windows size: 60								
BG	0.929	0.979	0.969	0.992	0.945	0.985	0.983	0.996
BS	1.015	1.078	1.025	1.064	1.013	1.062	1.017	1.042
L$_{210}$-AFTER	1.217	1.108	1.121	1.051	1.180	1.084	1.082	1.033
RB	2.028	2.715	1.609	1.910	1.857	2.357	1.420	1.602
s-AFTER	1.225	1.113	1.127	1.059	1.187	1.090	1.088	1.040
Windows size: Not applicable								
SAN1	0.956	1.040	0.995	1.033	0.970	1.037	1.007	1.028
SAN2	1.008	1.078	1.034	1.058	1.014	1.068	1.032	1.045

This table shows the relative MSEs of various combination algorithms when combining unbiased and heteroskedastic forecasts, where forecasters face unexpected aggregate shocks. In exercise 6.1, infrequent and small aggregate shocks are considered. In exercises 6.2, aggregate shocks are frequent but small. In exercise 6.3, aggregate shocks are infrequent but large. In exercise 6.4, there are frequent large aggregate shocks

In this exercise, to allow for unexpectedly large aggregate shocks, we let c_t follow a mixture distribution: With probability $1 - p$, $c_t \sim \mathcal{N}(0, 1)$. With probability p, $c_t \sim \mathcal{U}(2.5, q)$. The parameter p controls the frequency of large aggregate shocks, while q determines how large they can be. In exercise 6.1, infrequent and small shocks are considered, where $p = 0.05, q = 4.5$. Exercise 6.2 considers frequent small shocks with $p = 0.2, q = 4.5$. Exercise 6.3 considers infrequent but large shocks, where $p = 0.05, q = 6.5$. An environment which large shocks hit frequently is considered in exercise 6.4 with $p = 0.2, q = 6.5$.

Table 5 shows the performances of the combination algorithms. Aggregate shocks make it harder to differentiate the good forecasters from the poor ones. As a result, we expect to see combined forecasts with higher accuracy in environments with smaller and less frequent shocks. This turns out to be the case for most of the algorithms including BG, SAN1, and SAN2, where the best performance is observed in exercise 6.1. However, BS, RB, and the two AFTER algorithms perform the best in exercise 6.4. Despite these observations, it is worth noting that the BG method remains the only one that consistently outperforms simple averaging, although just by a few percentage points. While this exercise does not precisely simulate the business cycle, having unexpectedly large aggregate shocks is similar to forecasters missing a business cycle turning point. Therefore, the results here may also shed some light on the behavior of these algorithms when the target variable is highly cyclical.

4.7 Combining heteroskedastic forecasts with outliers

Finally, in exercise 7, we look at the algorithms' robustness to forecast outliers. Each period, there is a small chance for a forecaster to produce an outlier, i.e., an unusually large forecast error. Except when influenced by outliers, forecasts are all unbiased with $b_{j,t} = 0\ \forall j, t$ and have stable performance $\sigma^2_{j,t} = \sigma^2_j\ \forall t$. For each forecaster, with probability 0.5, $\sigma^2_j \sim \mathcal{U}(0.1, 3.5)$. Otherwise, he has poor performance, with $\sigma^2_j \sim \mathcal{U}(3.5, 6.5)$. Forecast outliers are introduced by letting $\varepsilon_{j,t}$ follow a mixture distribution. With probability $(1 - p_j)$, $\varepsilon_{j,t} \sim \mathcal{N}(0, \sigma^2_{j,t})$. With probability p_j, $\varepsilon_{j,t} \sim \mathcal{U}(2\sigma_{j,t}, 4\sigma_{j,t})$. In this setup, outliers have random, but significantly larger than usual, errors. p_j controls the frequency of their occurrences. In exercise 7.1, everyone produces outliers with a low probability $p_j = 0.05\ \forall j$. In exercise 7.2, $p_j = 0.05$ if $\sigma^2_j \sim \mathcal{U}(0.1, 3.5)$, otherwise, $p_j = 0$. This means that only the good forecasters occasionally produce outliers. In exercise 7.3, the opposite case is considered, where only the poor forecasters may produce outliers: $p_j = 0.05$ if $\sigma^2_j \sim \mathcal{U}(3.5, 6.5)$, otherwise, $p_j = 0$. In the last setting, exercise 7.4, a random set of 20% of the forecasters produce outliers, i.e., with probability 0.2, $p_j = 0.05$, and with probability 0.8, $p_j = 0$.

The results from this exercise are presented in Table 6. We first compare exercise 7.1 with 7.4. The combination algorithms are expected to perform better when there are fewer outliers. This turns out to be true for all except BG, which performs slightly better in exercise 7.1. But for the other algorithms, the performance differences are small. Comparing exercise 7.2 with 7.3, we see that, relative to weighting everyone equally, the performance-based algorithms take a bigger performance hit when the good forecasters produce outliers. A natural remedy for outliers is a long window. The effect of an outlier on the MSE of a forecaster becomes more diluted as more historical data are used to compute the MSE. This is true for BG and BS. However, a longer window does not help the AFTER algorithms. This is most likely because of the algorithms' already long memories. Finally, a comparison of the two AFTER algorithms shows that the s-AFTER is uniformly worse (though only slightly), confirming the effectiveness of the outlier-protective design of the L_{210}-AFTER.

4.8 Lessons and remarks

After reviewing all seven sets of simulation exercises, we believe that, first, given the high cost of estimating individual-specific weights, we should keep the number of forecasters small. This can be achieved by excluding infrequent survey participants and forecasters with consistent and obvious poor performances and by grouping the forecasters and combining group consensuses instead of individual forecasts. Second, using a short window to calculate individuals' performances helps little when using the AFTER algorithms. Instead, given the recursive structure of the algorithms, we may reset individuals' weights to be equal periodically or after an apparent structural

Table 6 Relative MSEs: Exercise 7

Algorithm	Exercise and number of forecasters (n)							
	Exercise 7.1		Exercise 7.2		Exercise 7.3		Exercise 7.4	
	$n = 5$	$n = 30$	$n = 5$	$n = 30$	$n = 5$	$n = 30$	$n = 5$	$n = 30$
Windows size: 24								
BG	0.891	0.921	0.953	0.982	0.801	0.898	0.890	0.958
BS	1.099	1.258	1.110	1.302	1.048	1.203	1.078	1.245
L_{210}-AFTER	1.413	1.186	1.617	1.368	1.095	1.068	1.308	1.180
RB	2.777	4.901	2.622	4.190	2.623	4.562	2.502	3.966
s-AFTER	1.429	1.194	1.634	1.378	1.103	1.071	1.315	1.183
Windows size: 60								
BG	0.856	0.908	0.930	0.974	0.776	0.890	0.872	0.953
BS	1.013	1.101	1.029	1.121	0.976	1.078	1.008	1.098
L_{210}-AFTER	1.413	1.185	1.616	1.368	1.095	1.067	1.307	1.179
RB	2.777	4.901	2.622	4.190	2.623	4.562	2.502	3.966
s-AFTER	1.430	1.194	1.634	1.378	1.103	1.071	1.315	1.183
Windows size: Not applicable								
SAN1	0.930	1.052	0.971	1.048	0.849	1.037	0.915	1.045
SAN2	1.024	1.173	1.033	1.099	0.950	1.176	0.985	1.114

This table shows the relative MSEs of various combination algorithms when combining unbiased and heteroskedastic forecasts, where there may be outliers, i.e., unusually large forecast errors. In exercise 7.1, everyone produces outliers with a low probability. In exercises 7.2, only the good forecasters occasionally produce outliers. In exercise 7.3, only the poor forecasters occasionally produce outliers. In exercise 7.4, a random set of 20% of the forecasters produce outliers, regardless of whether they are good or poor

break. Third, we may hedge against unexpected aggregate shocks and forecast outliers using procedures similar to the shrinkage step in SAN. This can be as simple as, e.g., combining the combined forecast produced by s-AFTER with the simple average of individual forecasts, where the latter receives a small but non-trivial weight. Finally, we should keep in mind that, in a continuously changing environment, it may be very difficult for performance-based combination algorithms to be effective, given the amount of data they need to properly estimate individuals' performances. Therefore, if frequent structural breaks are expected or the amount of historical data is extremely limited, using the mean of individual forecasts may be a prudent option.

5 Example: combining the SPF forecasts

The lessons learned from the simulation exercises should help researchers and policy makers develop forecast combination procedures and workflows that are more robust to various types of instabilities. To illustrate the usefulness of our suggestions in the previous section, we revisit the exercise in Lahiri et al. (2017), where the authors employed some of the algorithms examined in this paper when combining the forecasts reported in the U.S. Survey of Professional Forecasters, and they found that the more sophisticated algorithms only provided some limited improvements over simple averaging.

The SPF is a well-respected quarterly survey that collects forecasts made by professional forecasters. The survey was initially conducted by the American Statistical Association (ASA) and the National Bureau of Economic Research (NBER). Starting from 1990, the survey was taken over by the Federal Reserve Bank of Philadelphia. Consistent with the setup of the exercise in Lahiri et al. (2017), we use the forecasts of four different target variables: the real GDP growth rate (RGDP), the CPI inflation rate (CPI), the GDP deflator inflation rate (PGDP), and the unemployment rate (UNEMP). For each of the four variables, we separately combine the forecasts up to four quarters ahead ($h = 1, 2, 3, 4$).

In order to maximize potential performance gains, we try to improve the Lahiri et al. (2017) procedure in three straightforward ways: First, when calculating the MSEs of each individual forecaster, we use a rolling window of five years. Second, when using the AFTER algorithms, we periodically reset the weights to equal. To be consistent with the size of the rolling window, the reset happens every five years. Third, before applying the weighting algorithms, we put the forecasters into five groups based on their past performances measured by the MSE. Within a group, all the forecasters receive equal weights. These group means are then combined using a weighted average, with the weights assigned by the combination algorithms.[15] This way, we avoid the cost of estimating many weights. In addition, the group means have more stable performances than those of individual forecasters, see Aiolfi and Timmermann (2006). As we discuss below, these improvements help us obtain much more accurate combined forecasts.

Partly due to the change in the survey administrator, there is a large amount of missing values in the survey. As shown in Lahiri et al. (2017), one cannot compare the results of different combination algorithms when they are applied to unbalanced panels, because all the algorithms implicitly impute the missing values and different algorithms do so differently. Since we intend to compare the performances of different combination algorithms, we must have balanced panels of forecasts. This is achieved using the same procedure as in Lahiri et al. (2017). First, instead of using the entire time series, we consider two subsamples separately. The first subsample covers the period from 1968:IV to 1990:IV. The second subsample starts from 2000:I and ends at 2019:IV. The forecasts of CPI inflation rate before 1981 are not available. So, for CPI, we only work with subsample 2. Second, we discard the forecasters with excessively large amounts of missing forecasts.[16] Specifically, in each quarter, we only combine the forecasts of those who have reported at least 10 forecasts during the most recent 20 quarters. Imposing this restriction leaves us, on average, some 20 forecasters in the first subsample and 30 in the second, depending on the specific target variable and horizon. As the last step in data preparation, we impute the remaining missing values ourselves. This imputation method is based on Genre et al. (2013) and is used with good results in Lahiri et al. (2017). Specifically, a missing forecast $f_{j,t}$ is imputed as

[15] The group means are treated as "individual forecasts" for all purposes. For example, RB takes the best group mean and SA takes the average of the group means.

[16] As reported in Capistrán and Timmermann (2009), Genre et al. (2013), as well as Lahiri et al. (2017), a forecaster's performance and the frequency of his participation appear unrelated.

$f_{j,t} = \bar{f}_t + \hat{\beta}_j[\sum_{s=1}^{4}(f_{j,t-s} - \bar{f}_{t-s})/4]$, where $\hat{\beta}_j$ is the OLS estimate. This way, a missing forecast is imputed with an adjusted average for that time period. The average is that of the non-missing forecasts for that same period. The adjustment is forecaster specific and is based on the forecaster's usual amount of deviation from the average in the past year, $\sum_{s=1}^{4}(f_{j,t-s} - \bar{f}_{t-s})/4$.

In real time, we implement the combination algorithms considered in the previous section using these balanced panels of forecasts. Specifically, we place ourselves at the end of each quarter trying to combine the forecasts reported in the current quarter's survey, which target the next four quarters. Due to their publication lags, we only have the actual values up to the previous quarter when we calculate the weights. These actual values are the first releases/vintages of the target variables. And they are also used when we evaluate the performance of the combined forecasts. To avoid the possibility of excessive data mining and to keep the results in this exercise comparable with those in the previous section, all the algorithms are implemented using the same set of parameters as in the simulation exercises. We put the forecasters into five groups. As already mentioned, we use a window of 20 quarters for all the target variables and subsamples. So we discard the first 20 quarters in each subsample before calculating the MSE of the combined forecasts. We carry out this exercise separately for each target variable, horizon, and subsample.

Similar to the practice in the previous section, in Table 7, we report the results as relative MSEs. A relative MSE smaller than 1 means that the combination algorithm performs better than simple averaging. In such cases, we shade the table cell. In addition, we test all the cases using the one-sided modified Diebold-Mariano test (Harvey et al. 1997) at 10% level. The statistically significant numbers are set in bold. As Table 7 shows, the performance-based weighting algorithms often outperform simple averaging. For all four target variables, significant performance gains can be obtained from combining the one-quarter-ahead forecasts. For RGDP and UNEMP, moderate improvements over simple averaging are observed even at higher horizons. For all the algorithms, there are cases when they outperform simple averaging. In several cases, the combined forecasts produced by the AFTER algorithms are much more accurate than the benchmark.

There is a clear contrast between these results and those reported in Lahiri et al. (2017), where the performance-based algorithms deliver very modest improvements in most cases. Here, we are able to achieve statistically significant and practically meaningful performance gains with an improved procedure due to the lessons learned from the simulation exercises. In practice, when the parameter values of the combination algorithms can be more appropriately selected, one may obtain even better results.

As robustness checks, we repeated this real-time combination exercise using an alternative window size of 40 quarters and the participation requirements of 8, 12, 16, and 20 quarters. The longer window size results in slightly lower performance gains, while a more stringent participation requirement tends to slightly boost the performance gains. The precise amount varies by target variable and horizon. But overall, a window size of 20 and a participation requirement of 10 provides the best balance for this data set. Our conclusions stay the same regardless. We also carried out our combination exercise using individual forecasts directly, i.e., without grouping. On average over all the target variables, subsamples, and horizons, the MSE of the combined

Table 7 Combining SPF forecasts: relative MSEs and DM test results (with grouping)

Algorithm	Subsample 1: 1968:IV to 1990:IV				Subsample 2: 2000:I to 2018:II			
	h=1	h=2	h=3	h=4	h=1	h=2	h=3	h=4
Target variable: CPI								
BG					**0.856**	1.004	1.005	1.001
BS					**0.701**	1.033	1.044	1.001
L_{210}-AFTER					**0.581**	1.030	1.052	1.015
RB	CPI data available				*0.889*	1.056	1.077	1.123
s-AFTER	for subsample 2 only				**0.579**	1.037	1.059	1.011
SAN1					**0.745**	1.019	1.033	1.018
SAN2					**0.727**	1.028	1.035	1.020
Target variable: PGDP								
BG	**0.944**	1.001	**0.990**	1.008	*0.989*	1.007	1.001	1.006
BS	**0.882**	1.048	1.106	1.027	*0.994*	1.087	1.057	1.133
L_{210}-AFTER	*0.911*	1.112	*0.983*	1.055	*0.926*	1.050	1.025	1.062
RB	*0.997*	1.036	*0.987*	1.023	1.071	1.169	1.070	**0.897**
s-AFTER	*0.899*	1.130	*0.999*	1.037	*0.934*	1.039	1.045	1.105
SAN1	**0.922**	1.005	1.003	1.040	*0.985*	1.012	1.004	1.008
SAN2	**0.916**	1.001	1.000	1.015	*0.966*	1.027	1.012	1.032
Target variable: RGDP								
BG	**0.961**	1.004	*0.993*	1.014	*0.995*	*0.999*	*0.999*	*0.996*
BS	**0.869**	1.169	1.005	1.002	1.011	1.021	1.021	*0.965*
L_{210}-AFTER	**0.849**	1.089	*0.967*	1.165	*0.981*	1.014	*0.999*	*0.973*
RB	1.245	0.933	1.101	1.188	*0.984*	1.038	1.017	1.081
s-AFTER	**0.866**	1.105	0.966	1.136	*0.982*	*0.979*	1.000	*0.969*
SAN1	**0.947**	1.033	0.995	1.083	*0.986*	*0.984*	1.000	*0.983*
SAN2	0.995	1.001	1.007	1.008	*0.983*	*0.989*	*0.999*	*0.986*
Target variable: UNEMP								
BG	**0.960**	1.002	**0.980**	*0.985*	**0.939**	**0.972**	**0.976**	**0.987**
BS	0.898	**0.920**	**0.926**	*0.992*	**0.907**	**0.918**	**0.948**	*0.999*
L_{210}-AFTER	**0.834**	1.021	**0.917**	*0.931*	**0.913**	**0.933**	**0.953**	1.002
RB	1.260	1.079	1.053	1.157	*0.983*	*0.891*	*0.971*	1.052
s-AFTER	**0.811**	1.031	**0.924**	*0.897*	*0.950*	*0.941*	**0.951**	1.031
SAN1	**0.996**	1.001	**0.982**	**0.985**	**0.997**	**0.994**	**0.993**	*0.994*
SAN2	**0.991**	1.004	**0.964**	**0.966**	**0.993**	**0.987**	**0.986**	**0.986**

This table shows the relative MSEs of various combination algorithms when combining the SPF forecasts, relative to the MSE of simple averaging. Relative MSEs lower than 1 are italic. Bold numbers are statistically significantly lower than 1 according to the one-sided DM test at the 10% level. Individual forecasters are grouped according to their MSEs. The combination algorithms, including simple averaging, are used to combine the group means

forecasts obtained without grouping is 9.1% higher. In 96% of the cases, combining grouped forecasts results in a lower MSE than combining individual forecasts without grouping.

6 Concluding remarks

In this study, we examined the performances of several recently developed forecast combination algorithms in unstable environments in seven simulation exercises. The first exercise revealed the cost of estimating weights for individual forecasters as the number of forecasters increases. In the second exercise, we documented the algorithms' performances when combining biased forecasts. A one-time break in forecast bias was introduced in the third exercise, which allowed us to observe closely the performances of the combined forecasts immediately after the break. We considered multiple breaks and heteroskedastic forecasts in the fourth exercise. Next, we studied cases in which individual forecasters' performances were changing continuously. The role of unexpectedly large aggregate shocks and the effect of forecaster-specific outliers were considered in the remaining two simulation exercises, respectively. Each of the seven exercises was carried out using four different sets of parameter values, which allowed us to perform comprehensive and in-depth analysis of the performances of the combination algorithms.

The simulation exercises led to several observations. First, the accuracy of the combined forecasts deteriorates rapidly as the number of forecasters increases. This is so even when individual forecasters' performances stay constant over time, i.e., there is no structural instability. The second observation is that, in many cases, the length of estimation window has little effect on the performance of the algorithms, especially the AFTERs. Given their recursive structure, the AFTERs may benefit more from resetting the recursion than limiting the window size used to estimate individual forecasters' performances. We also observed that a higher level of heteroskedasticity in the forecasts helps the algorithms to differentiate the good forecasters from the poor ones. But at the same time, it makes estimating individuals' weights more costly. Therefore, a higher level of heteroskedasticity may be good for less aggressive algorithms such as BG, but not for the AFTERs, which are more sensitive to small changes in forecasters' performances. In addition, our results suggested that it takes a long time for the performance of an algorithm to stabilize after a break in forecasters' performances. In environments with frequent breaks, the performances of algorithms such as the AFTERs may never reach the optimal level. In such cases, simple averaging may be a better alternative. Based on these results, we proposed a combination procedure that reduces the number of weights that must be estimated and stabilizes the performances of candidate forecasts by pooling the forecasts from individuals with similar performances. We demonstrated the effectiveness of this procedure in a real-time forecast combination exercise using forecasts from the U.S. Survey of Professional Forecasters.

References

Aiolfi M, Timmermann A (2006) Persistence in forecasting performance and conditional combination strategies. J Econom 135:31–53

Armstrong JS (1989) Combining forecasts: the end of the beginning or the beginning of the end? Spe Sect: Time Ser Monit 5:585–588

Bates JM, Granger CWJ (1969) The combination of forecasts. Oper Res Q 20:451–468

Bürgi C, Sinclair TM (2017) A nonparametric approach to identifying a subset of forecasters that outperforms the simple average. Empir Econ 53:101–115

Capistrán C, Timmermann A (2009) Forecast combination with entry and exit of experts. J Econ Stat 27:428–440

Cheng G, Yang Y (2015) Forecast combination with outlier protection. Int J Forecast 31:223–237

Chevillon G (2016) Multistep forecasting in the presence of location shifts. Int J Forecast 32:121–137

Clemen RT (1989) Combining forecasts: a review and annotated bibliography. Int J Forecast 5:559–583

Davies A, Lahiri K, Sheng X (2011) Analyzing three-dimensional panel data of forecasts. The Oxford handbook of economic forecasting, pp 473–495

Elliott G (2017) Forecast combination when outcomes are difficult to predict. Empir Econ 53:7–20

Elliott G, Timmermann A (2005) Optimal forecast combination under regime switching*. Int Econ Rev 46:1081–1102

Genre V, Kenny G, Meyler A, Timmermann A (2013) Combining expert forecasts: Can anything beat the simple average? Int J Forecast 29:108–121

Giraitis L, Kapetanios G, Price S (2013) Adaptive forecasting in the presence of recent and ongoing structural change. J Econ 177:153–170

Harvey D, Leybourne S, Newbold P (1997) Testing the equality of prediction mean squared errors. Int J Forecast 13:281–291

Lahiri K, Peng H, Zhao Y (2017) On-line learning and forecast combination in unbalanced panels. Econ Rev 36:257–288

Pesaran MH, Pick A, Pranovich M (2013) Optimal forecasts in the presence of structural breaks. J Econ 177:134–152

Pesaran MH, Timmermann A (2005) Small sample properties of forecasts from autoregressive models under structural breaks. J Econ 129:183–217

Sancetta A (2007) Online forecast combinations of distributions: worst case bounds. J Econ 141:621–651

Sancetta A (2010) Recursive forecast combination for dependent heterogeneous data. Econ Theory 26:598–631

Smith J, Wallis KF (2009) A simple explanation of the forecast combination puzzle. Oxf Bull Econ Stat 71:331–355

Stock JH, Watson MW (2004) Combination forecasts of output growth in a seven-country data set. J Forecast 23:405–430

Tian J, Anderson HM (2014) Forecast combinations under structural break uncertainty. Int J Forecast 30:161–175

Wei X, Yang Y (2012) Robust forecast combination. J Econ 166:224–236

Yang Y (2004) Combining forecasting procedures: some theoretical results. Econ Theory 20:176–222

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.