Information Fusion 103 (2024) 102093

Contents lists available at ScienceDirect

Information Fusion

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

Full length article

Probabilistic study of Induced Ordered Linear Fusion Operators for time series forecasting

Juan Baz [a,*], Mikel Ferrero-Jaurrieta [b], Irene Díaz [c], Susana Montes [a], Gleb Beliakov [d], Humberto Bustince [b]

[a] Department of Statistics and I.O. and Didactic of Mathematics, University of Oviedo, Calle Federico Garcia Lorca 18, Oviedo, 33007, Asturias, Spain
[b] Department of Statistics, Computer Science and Mathematics, Public University of Navarre, Campus Arrosadia, Pamplona, 31006, Navarra, Spain
[c] Department of Computer Science, University of Oviedo, Campus de Viesques, Gijon, 33007, Asturias, Spain
[d] School of Information Technology, Deakin University, 75 Pigdons Rd, Geelong, 3220, Geelong, Australia

ARTICLE INFO

Keywords:
Time series forecasting
Predictions fusion
Induced Ordered Weighted Averaging
Pre-aggregation functions

ABSTRACT

The aggregation of several predictors in time series forecasting has been used intensely in the last decade in order to construct a better resulting model. Some of the most used alternatives are the ones related to the Induced Ordered Weighted Averaging (IOWA), in which the prediction values are ordered using a secondary vector, often related to the accuracy of the prediction model in the last prediction. Although the time series study has been historically a subject related to statistics and stochastic processes, the random behaviour of the aggregation process is typically not considered. In addition, extensions of aggregation functions with a weaker notion of monotonicity, pre-aggregation functions, are appearing as better alternative for some topics such us classification. In this paper, a pre-aggregation extension of the IOWA operator, the Induced Ordered Linear Fusion (IOLF), is defined as a way to aggregate time series model predictions and its behaviour is studied from a probabilistic point of view. The IOLF operator over random vectors is defined, its properties studied and the relation between some averaging aggregation functions established. The expressions of the optimal weights according to statistical criteria are derived. The advantages and consequences of the use of the IOLF operator are studied, and its behaviour is compared to the usual procedures. Numerical results illustrate its performance on a practical example.

1. Introduction

1.1. Motivation and main contributions

In time series forecasting, there exists a huge variety of different prediction methods, each with its benefits and drawbacks. In general, it is not easy to choose the best prediction model and, therefore, ensemble methods are usually used to combine the predictions of different prediction models to obtain a better single prediction [1,2].

One of the most used alternatives in the last decade is the Induced Ordered Weighted Averaging (IOWA) operator, introduced by Yager [3]. The IOWA operator orders the predictions of the models using a secondary vector and then computes a weighted mean. This secondary vector is usually chosen to be the precision of the models in the last time step [4], the main vector itself, resulting in a Ordered Weighted Averaging (OWA) or a increasing vector, resulting in a Weighted Averaging Mean (WAM) [5].

The IOWA operator is an aggregation function, thus it is monotone and fulfil some boundary conditions. In the last years, extensions of aggregation functions that relax these conditions are outperforming classical aggregation functions for some tasks. We refer to the generalization of the Choquet Integral, see [6], and its applicability to classification [7,8] or image analysis [9], or to the use of t-norm based pre-aggregation in Knowledge-Based Systems [10].

In our context, we can follow this idea by generalizing the IOWA operator by considering negative weights when computing the weighted mean of the ordered arguments. This generalization can also be extended to OWA and WAM operators. We devote this paper to follow this idea, studying the properties, semantics and applicability in time series forecasting of the resulting fusion operator. In particular, let us point-out the main original contributions of this paper:

- The Induced Ordered Linear Fusion Operator (IOLF) is defined as a generalization of the IOWA operator by allowing negative weights.
- Its monotonicity, boundary conditions and the semantics of negative weights are studied.

* Corresponding author.
E-mail address: bazjuan@uniovi.es (J. Baz).

https://doi.org/10.1016/j.inffus.2023.102093
Received 24 April 2023; Received in revised form 28 September 2023; Accepted 23 October 2023
Available online 31 October 2023

- A closed expression of the optimal weights with respect to statistical criteria is derived. Results proving important properties of the operator and the derivation of optimal weights are provided, considering a probability approach.
- Several IOLF operators are applied in a illustrative example in time series prediction. Its behaviour is compared with the one of IOWA, OWA and WAM operators.

The probability approach of the paper is justified by the fact that time series and several prediction models, such as the ARIMA, Kalman Filter or Exponential Smoothing [11,12], are defined typically in terms of random variables.

1.2. Related work

We devote this Subsection to make a literature survey on the topic of ensemble methods for time series. The literature review has been carried out through the search engines *Scopus, Web of Science* and *Google Scholar*. The most important used keywords have been *time series, prediction, forecast, ensemble, aggregation, fusion, weighted averaging, ordered weighted averaging, induced weighted averaging, OWA and IOWA*. We have also used forward (snowball) and backward reference searching, considering papers that cite or are cited by relevant works. We have focused mainly on the use of linear aggregation functions, since they are widely used in the literature and also are directly related with our proposal.

The use of ensembles in time series has been investigated deeply in the literature. Some surveys focused on the topic have been published in the last years, for instance, see [1,2,13].

Technically, models based on decision trees such that random forest and boosted decision trees are particular types of ensembles, see [14]. However, the initial prediction models are weak predictors of the same type. This type of ensembles are known as homogeneous ensembles. We are interested in heterogeneous ensembles, which consider strong prediction models based on different techniques that can target different data characteristics.

For instance, the authors in [15] combine machine learning models by using weighted means in an application regarding financial data. The different architectures of the models make the ensembled model to be more flexible and, therefore, to have a better accuracy than the initial models. The use of aggregation functions is extended as ensembles in the literature [1]. For instance, the authors in [16] give a method that construct confidence intervals based on aggregation functions. A comparative study between different alternatives of linear aggregation functions is made in [5]. In the considered datasets, the Ordered Weighted Averaging outperforms the Weighted Arithmetic Mean.

Another linear aggregation function that is used in this regard is the Induced Ordered Weighted Averaging (IOWA), in which the arguments the predictions are ordered by, typically, the precision in the previous times. This technique is quite popular, with applications in safety monitoring [4,17], logistic [18], economy [19–22], energy resources [23–25] or climatology [26,27]. This approach permits to rearrange the prediction models dynamically as time changes. Another alternative of time-dependant ensemble is explored in [28], in which a weighted mean with dynamic weights is proposed.

Other ensemble methods focus in how the initial prediction models are trained. Bagging ensemble, see [1], uses different bootstrap samples to train the prediction models and then applies a weighted mean. Note that bootstrap sampling in time series is not easy due the dependence between observations, thus block bootstrap [29] should be used. In these models, the diversity of the models is not only due to differences in the architecture but also in the training samples.

Not only aggregation functions can be used as ensembles. The predictions of the initial prediction models can be used as the inputs of a machine learning prediction model that gives the final prediction.

Table 1
Table summarizing the ensemble models used and the conclusions or relevant comments of some of the reviewed papers.

Ref	Ensemble model/s	Comments/Conclusions
[4]	WAM, IOWA	IOWA outperforms WAM
[5]	WAM, OWA	OWA outperforms WAM
[15]	WAM	Ensemble outperforms initial models
[16]	Based on the Mean	Focuses on confidence intervals
[17]	IOWA	Ensemble outperforms initial models
[18]	IOWA	Learning the weights improves the results
[19]	IOWA	Ensemble outperforms initial models
[20]	WAM, IOWA	Accuracy-based WAM has better results
[22]	IOWA	Ensemble outperforms initial models
[23]	IOWA	Ensemble outperforms initial models
[24]	IOWA	Ensemble outperforms initial models
[25]	Based on IOWA	Ensembles outperform initial models
[26]	IOWA	Ensemble outperforms initial models
[27]	IOWA	Uses bootstrap
[30]	SVR	SVR ensemble outperforms other models
[31]	Combiner, others	Combiner outperforms other ensembles
[32]	WAM, Median, Mode	Mode outperforms other ensembles

The authors in [30] use, for instance, Support Vector Regression. In [31], a Combiner is proved to have a better behaviour than different weighted-based aggregation functions for medical data. However, these ensemble models lack the explainability of the weighted-based aggregation functions, as pointed out in [32].

We end this section by providing in Table 1 a brief summary of some of the latter papers, focusing on the used ensemble methods, the conclusions of the research and some relevant comments.

1.3. Structure of the paper

The rest of the paper is organized as follows. In Section 2, we will introduce the basic notions of aggregation functions and estimators needed for the development of the rest of the paper. The generalized concept of IOLF is introduced in Section 3 and its main properties, when it is applied over random vectors, are studied. Section 4 is devoted to the study of methods that determine the optimal weights based on classical statistical criteria. A practical example is provided in Section 5, which illustrate the advantages of the proposed method. The conclusions are discussed in Section 6.

2. Preliminaries

In this section we will introduce the general concepts required to develop the rest of the work. In particular, we will show some definitions and results of estimators, aggregation functions and the usual procedure when using IOWA operator in time series forecasting.

2.1. Estimators

In the following we recall the basic concepts about estimators and their properties, using as the main Ref. [33]. We will use indistinctly the term sequence of random variables $X_1, \dots, X_n$ or random vector $\vec{X} = (X_1, \dots, X_n)$ for referring to an ordered finite set of random variables defined in the same probability space. An estimator is a function that maps the sample space of a sequence of identically distributed and independent (iid) random variables to a parameter space. That is, for any observation of the sequence, it returns a value of the parameter (or parameters).

Definition 1. Let $X_1, \dots, X_n$ be a sequence of iid random variables a density function $f_\theta(t)$ that depends on some unknown parameters θ, assuming values in the parametric space Θ. An (point) estimator is a measurable function $T : \mathbf{R}^n \to \Theta$ such that it is not dependant on the value of the unknown parameters.

There are some relevant properties that are desirable for an estimator. We will focus on two of them, the unbiasedness and the efficiency. The unbiasedness says that the expected value of our estimator coincides with the value of the unknown parameter.

Definition 2. Let $X_1, \dots, X_n$ be a sequence of random variables with common density function $f_\theta(t)$ depending on some unknown parameter $\theta \in \Theta$. An estimator T is called unbiased if $E[T] = \theta$ for any $\theta \in \Theta$.

The efficiency between two estimators for a parameter is related to the Mean Squared Error (MSE).

Definition 3. Let $X_1, \dots, X_n$ be a sequence of iid random variables with density function $f_\theta(t)$ depending on the unknown parameter $\theta \in \Theta$ and let T_1, T_2 be two estimators of θ. It is said that T_1 is more efficient than T_2 if $E\left[(T_1-\theta)^2\right] \leq E\left[(T_2-\theta)^2\right]$ for any $\theta \in \Theta$ and it exists $\theta_0 \in \Theta$ such that $E\left[(T_1-\theta_0)^2\right] < E\left[(T_2-\theta_0)^2\right]$.

For any estimator T of a parameter θ, $E\left[(T-\theta)^2\right]$ is known as the mean squared error and denoted by $MSE(T)$. Notice that if the estimators are unbiased, then the MSE is just the variance. A definition of global efficiency between unbiased estimators can be stated by using the Fréchet–Cramér–Rao lower bound, see [34–36].

We end by giving some classical results for three families of distributions regarding efficient estimation of the mean. For the uniform case, and for some of our results, we need the definition of order statistics.

Definition 4 ([33])**.** Let $X_1, \dots, X_n$ be a sequence of random variables. The function $X_{(k)}$ of $(X_1, \dots, X_n)$ that takes the value kth smaller value in each possible observation $(x_1, \dots, x_n)$ of $(X_1, \dots, X_n)$ is known as the kth order statistic or statistic of order k (of the sequence $X_1, \dots, X_n$).

The set $\{X_{(1)}, \dots, X_{(n)}\}$ is known as the set of the order statistics of $X_1, \dots, X_n$. Of course, the order statistic are random variables [33].

Example 1 ([37,38])**.**

1. If X_i, for any $i \in \{1, \dots, n\}$, has a uniform distribution, the unbiased efficient estimation for its mean is $\frac{X_{(n)}+X_{(1)}}{2}$.
2. If X_i, for any $i \in \{1, \dots, n\}$, has exponential or a Gaussian distribution, the unbiased efficient estimation for its mean is the sample mean.

2.2. Notions of monotonicity and pre-aggregation functions

An aggregation function is typically referred to as a function that summarizes several values by a single number. Formally, given a real interval (bounded or not) I, an aggregation function is defined as a monotone increasing function $A : I^n \to I$ which satisfies that the infimum and the supremum of their image are, respectively, the infimum and the supremum of the aforementioned interval.

Definition 5 ([39])**.** Let I be a (possibly unbounded) interval in the real line $\mathbb{R}$. An aggregation function is a function $A : I^n \to I$ satisfying:

- It is increasing (in each variable).
- The following boundary conditions are fulfilled:

$$\inf_{x \in I^n} A(x) = \inf I, \quad \sup_{x \in I^n} A(x) = \sup I$$

Example 2. Given a vector of weights $\vec{w} \in [0,1]^n$ fulfilling $\sum_{i=1}^n w_i = 1$, the Weighted Averaging Mean (WAM) and the Ordered Weighted Averaging (OWA) are defined as follows:

$$\mathrm{WAM}(\vec{x}; \vec{w}) = \sum_{i=1}^n w_i x_i, \qquad \mathrm{OWA}(\vec{x}; \vec{w}) = \sum_{i=1}^n w_i x_{\sigma(i)}$$

where σ is a permutation such that $x_{\sigma(1)} \geq \cdots \geq x_{\sigma(n)}$. Both are aggregation functions for any interval, since they are monotone and the boundary conditions are fulfilled. Notice that there is a direct relation between the OWA operator and the order statistics, since both involve a ordination of a given sample.

The number of properties defined over aggregation functions is huge, but we are interested in three in particular, the idempotency, ratio scale invariance and additivity properties.

Definition 6 ([39])**.** Let $A : I^n \to I$ be an aggregation function. If for any $s \in \mathbb{I}$ it holds:

$$A(s\vec{1}) = s$$

then A is called an idempotent aggregation function

Definition 7 ([39])**.** Let $A : I^n \to I$ be an aggregation function. If for any $s, r \in \mathbb{R}$ with $r > 0$ it holds:

$$A(r\vec{x} + s\vec{1}) = rA(\vec{x}) + s$$

for any $\vec{x} \in I^n$ such that $r\vec{x} + s\vec{1} \in I^n$, then A is called an interval scale invariant aggregation function.

Definition 8 ([39])**.** Let $A : I^n \to I$ be an aggregation function. If for any $\vec{y} \in \mathbb{I}$ it holds:

$$A(\vec{x} + \vec{y}) = A(\vec{x}) + A(\vec{y})$$

for any $\vec{x} \in I^n$ such that $\vec{x} + \vec{y} \in I^n$, then A is called an additive aggregation function.

We want to remark that additive and interval scale invariant (thus idempotent) aggregation functions are convenient when applied to random variables, since the expectation operator is linear, i.e. if X and Y are two random variables and $r, s, c \in \mathbb{R}$, $E[rX + sY + c] = rE[X] + sE[Y] + c$. Let us now define the Induced Ordered Weighted Averaging, an aggregation function that takes a second vector as argument, which is used to order the first one before a applying a convex linear sum.

Definition 9 ([3])**.** Let $\vec{w} \in \mathbb{R}^n$ be a weight vector such that $w_1, \dots, w_n \geq 0$ and $\sum_{i=1}^n w_i = 1$. Consider the ordination $\pi_{\vec{y}} : \{1, \dots, n\} \to \{1, \dots, n\}$ such that $\pi_{\vec{y}}(\vec{y})_i = y_{(i)}$ and, if there is any draw in $\vec{y}$, replace the associated values of $\vec{x}$ by their average. Then, the Induced Ordered Weighted Averaging (IOWA) has the following expression:

$$\mathrm{IOWA}(\vec{x}, \vec{y}; \vec{w}) = \vec{w}' \pi_{\vec{y}}(\vec{x})$$

This type of aggregation functions contains as particular cases the WAM and the OWA operators. Trivially, this aggregation is additive, interval scale invariant and idempotent. The weight w_1 is associated to the importance of the value $\vec{x}$ on the position of the component with greater value in $\vec{y}$. We end this section by recalling some relaxations in the concept of monotonicity of a function and the concept of pre-aggregation function.

Definition 10 ([40])**.** Let $A : I^n \to I$ be a function and $\vec{r} \in \mathbb{R}^n$. A is said to be directionally monotone with respect to $\vec{r}$ if:

$$A(\vec{x} + c\vec{r}) \geq A(\vec{x}),$$

for any $\vec{x} \in I^n$ and any $c \in \mathbb{R}^+$ such that $\vec{x} + c\vec{r} \in I^n$.

A particular case is the weakly monotonicity, which is directional monotonicity with respect to the vector of ones $\vec{1} \in \mathbb{R}^n$ [41].

Definition 11 ([39])**.** Let I be a (possibly unbounded) interval in the real line $\mathbb{R}$. A pre-aggregation function is a function $P : I^n \to I$ satisfying:

- P is directionally increasing with respect to a vector $\vec{r} \in \mathbb{R}^n$.

- The following boundary conditions are fulfilled:

$$\inf_{x\in I^n} P(x) = \inf I, \quad \sup_{x\in I^n} P(x) = \sup I$$

Any aggregation function is a pre-aggregation function for any positive vector $\vec{r} \in \mathbb{R}^{+n}$. The concepts introduced in Definitions 6–8 also hold for pre-aggregation functions.

2.3. IOWA in time series forecasting: the usual procedure

The IOWA operator, among other applications, is used in the context of time series forecasting for fusing the predictions of several models into a better prediction model. In the literature (see [4,17–27]), the inducing vector of the IOWA operator at the time t is taken as the precision of the prediction, defined as follows:

$$y_{it} = \begin{cases} 1 - \left|\frac{\mu_t - p_{it}}{\mu_t}\right| & \text{if } \left|\frac{\mu_t - p_{it}}{\mu_t}\right| < 1 \\ 0 & \text{if } \left|\frac{\mu_t - p_{it}}{\mu_t}\right| \geq 1 \end{cases} \tag{1}$$

where μ_t is the real value at time t and p_{it} the prediction of the model i at time t.

However, if we want to make a prediction of an unknown value, we cannot compute the inducing vector, since it depends directly on μ. For those cases, it is assumed that the precision is similar to the last precision known. If we just want to predict consecutively the next value of the time series, this approach is similar to the best yesterday's model introduced in [3], where the predictions of today are ordered using the absolute difference between the last prediction and the observed value, for all the prediction models. Notice that in this case, the inducing vector does not depend on the value we want to predict.

Linking this problem to the classical estimation of random samples, this is a more general case, in which for a fixed time, $X_{1,t}, \dots, X_{n,t}$ form a random vector $\vec{X}_t$ with possibly dependant and non-identically distributed components with the same mean μ_t, which is the real value of the time series. We also have an additional random vector $\vec{Y}_t$, that induces the order and it is not dependant on μ_t.

The classical way to optimize the choice of the weights of the IOWA operator is to solve the following optimization problem:

$$\text{Minimize} \sum_{t=1}^{T} \left(\mu_t - \sum_{i=1}^{n} w_i \pi_{\vec{y}_t}(\vec{x}_t)_i \right)^2 \tag{2}$$

$$\text{Subject to} \sum_{i=1}^{n} w_i = 1 \text{ and } w_1, \dots, w_n \geq 0$$

This problem is similar to the one that will be presented in Theorem 20 when using the sample covariance matrix and the sample mean as estimator, but with the difference that in the latter one we eliminate the $w_1, \dots, w_n \geq 0$ condition.

We want to remark that another inducing vectors can be used. In particular, it is also common to consider particular cases of the IOWA operator such as the OWA and the WAM operators [5].

3. Induced Ordered Linear Fusion for random vectors

Our purpose is to apply an extension of the IOWA operator for forecasting models of time series, which are modelled as random variables. In particular, the prediction of several forecasting models are aggregated to obtain a better final prediction. This extension allows the weights to be negative, which implies having a greater feasible region in the optimization problem defined in (2) and a closed expression for its solution (see Theorem 20). Let us start by defining such extension, the Induced Ordered Linear Fusion.

Definition 12. Let $\vec{w} \in \mathbb{R}^n$ be a vector such that $\sum_{i=1}^n w_i = 1$. Consider the ordination $\pi_{\vec{Y}} : \{1, \dots, n\} \to \{1, \dots, n\}$ such that $\pi_{\vec{y}}(\vec{y})_i = y_{(i)}$ and, if there is any draw in $\vec{y}$, replace the associated values of $\vec{x}$ by their average. Then, the Induced Ordered Linear Fusion (IOLF) has the following expression:

$$\text{IOLF}(\vec{x}, \vec{y}; \vec{w}) = \vec{w}' \pi_{\vec{y}}(\vec{x})$$

Notice that, since we have just relaxed one of the conditions over the weights, the IOLF operator generalizes the IOWA operator.

Two particular cases of the IOLF operator are of special interest. If the vectors $\vec{x}$ and $\vec{y}$ are the same, then we will called the IOLF operator an Ordered Linear Fusion (OLF) operator, that will be denoted as $\text{OLF}(\vec{x}; \vec{w})$. It can be seen as an OWA operator with possibly negative weights. Similarly, if $\vec{y}$ is increasing, the resulting IOLF operator will be called Linear Fusion (LF) operator, denoted as $\text{LF}(\vec{x}; \vec{w})$. This operator is an extension of the WAM allowing negative weights.

3.1. Properties and semantic

Negative weights, although are a simple way to extend the IOWA operator, increase substantially the complexity of properties, interpretation and applicability of the resulting function. Firstly, many of the properties related to be an aggregation functions, monotonicity and the boundary conditions, are not longer true, thus it is necessary to work with weaker alternatives of these properties. Secondly, the semantic of negative weights is not intuitive and should be studied. Finally, negative weights allow to capture more complex dependence structures between the aggregated values. Let us discuss all these points one by one.

Starting with the properties, ratio scale invariance, additivity and idempotence remain to hold for the IOLF, since the condition $\sum_{i=1}^n w_i = 1$ is not changed. Therefore, the IOLF operator is weakly monotone. Moreover, following the same procedure as in the case of the OWA operators with negative weights (see Proposition 4.3 in [42]), the IOLF operator with weights $\vec{w} \in \mathbb{R}^n$ is directionally increasing with respect any vector $\vec{r} \in \mathbb{R}^n$ such that:

$$\sum_{i=1}^{n} r_{\pi(i)} w_i > 0$$

for any possible permutation π. Therefore, if we consider $I = \mathbb{R}$, the IOLF operator is a pre-aggregation function.

However, if we consider an interval $I \neq \mathbb{R}$, the negative weights do not allow the IOLF operator to be a pre-aggregation function. For instance, if $I = [0, 1]$ and $\vec{w} = (-0.5, 1.5)$, then:

$$\text{IOLF}((1, 2), (0, 1); (-0.5, 1.5)) = -0.5 \notin [0, 1]$$

Nevertheless, focusing on our main purpose, when predicting values of time series the prediction models typically do not give predictions on a bounded interval but all the real line [11]. For the cases in which we want to require the result to be in a specific interval, we can always consider a function from the real line to the interval. In the previous case, we can transform the predictions into the real line using the inverse of the sigmoid function [43] and apply there the procedure.

However, even considering $\mathbb{R}$ as the interval, having negative weights does not allow the IOLF to be monotone in every component, making it impossible to be an aggregation function. The non-monotonicity may be unintuitive, since if we increase the prediction of one of the prediction models we expect the fused prediction to also increase. However, as we will see later, this proposal could be better from a statistical point of view, even in the case the monotonicity is not preserved. Allowing negative weights expands the feasible region in the problem stated in (2), thus a better optimal solution is expected. We also want to remark that pre-aggregations, even with the loss of monotonicity, are starting to be used in applied problems (see again [7–10]). In Section 5, numerical results will show that the best alternative

for the considered datasets is a Weighted Averaging Mean with negative weights, outperforming the rest of classical alternatives.

We want to acknowledge the difficulties of the semantic of negative weights. For the IOWA operator, we can interpret the value of w_i as the importance of the prediction of the model with the ith best prediction in the previous time. For the OWA and WAM operators, the value of w_i is related to the importance of the ith model or the ith greatest prediction. Although the loss of semantics when using negative weights is evident, we can still interpret the absolute value of w_i as the importance of the prediction of the model with the ith component of the ordered vector, considering also as important models the ones with a bad expected prediction.

From the point of view of probability, the matrix $\Sigma + \vec{\Delta}'\vec{\Delta}$ that will appear in Theorem 20 is the equivalent to the covariance matrix with respect the real prediction of the components of the ordered random vector $\pi_{\vec{Y}}\left(\vec{X}\right)$. The resulting weights of this result will be negative if the sum of the rows of the $\left(\Sigma^{-1} + \vec{\Delta}'\vec{\Delta}\right)^{-1}$ is negative. Since the diagonal elements must be positive (since the matrix is positive semi-definite), negative weights imply negative non-diagonal elements of the latter matrix. In the case of covariance matrices, this negative elements of its inverse are related to positive conditional correlation and its magnitude to the strength of the dependence [44]. A similar interpretation can be made in this case, negative weights are associated with large positive conditional correlations between the variables of the ordered random vector $\pi_{\vec{Y}}\left(\vec{X}\right)$.

This strong positive dependence have consequences in terms of monotonicity. If we just look to the prediction with a negative weight, it is true that the aggregation decreases if the prediction increases. But, since a negative optimal weight implies a positive dependence, we can expect that an increase in the prediction implies an increase in the rest of the predictions, and therefore, an increase in the final aggregation. In particular, we can think on negative weights as a way to fine tuning co-increasing sets of predictions.

3.2. Probability properties of the IOLF operator

In the following, we will assume that the predictions of the different models can be modelled as random variables. Let us starting proving that the IOLF operator applied to random vectors is itself a random variable. We need to prove that $\text{IOLF}(\vec{X},\vec{Y};\vec{w})$ is measurable. We first introduce a lemma to clarify a step in our proof.

Lemma 13 ([45]). *Let (X,F,μ) be a measure space and $f,g: X \to \bar{\mathbb{R}}$ be two measurable functions. Then, $\{x \in X | f(x) \leq g(x)\} \in F$.*

Proposition 14. *Let $\vec{X}$ and $\vec{Y}$ be two random vectors and $\vec{w}$ be a vector of weights. Then $\text{IOLF}(\vec{X},\vec{Y};\vec{w})$ is a random variable.*

Proof. Let us start proving that $\pi_{\vec{Y}}(\vec{X})$ is a random vector. Consider the probability space (Ω,F,P) on which $\vec{X}$ and $\vec{Y}$ are defined. We just need to prove that any component is a random variable (a measurable function) from Ω to $\mathbb{R}$. In particular, we just need to verify that $\pi_{\vec{Y}}(\vec{X})_i^{-1}((\infty,a]) \in F$ for any $a \in \mathbb{R}$ and $i \in \{1,\dots,n\}$ [45].

If we consider the case in which $\pi_{\vec{Y}}(\vec{Y})_i = Y_k$, then $\pi_{\vec{Y}}(\vec{X})_i^{-1}((\infty,a]) = X_k^{-1}((\infty,a])$ and, for being $\vec{X}$ a random vector, this set belongs to F.

This only happens if $Y_{j_1} < \cdots < Y_{j_i} < \cdots < Y_{j_n}$ with $\{j_1,\dots,j_n\} = \{1,\dots,n\}$ (a reordination of the indices {1,...,n}) and $j_i = k$. Consider the following set,

$$B_k^i = \{t \in \Omega \mid Y_{j_1}(t) < \cdots < Y_{j_i=k}(t) < \cdots < Y_{j_n}(t)\},$$

which belongs to F by applying recursively Lemma 13 and the fact that the (finite) intersection of measurable sets is measurable [45]. Then, considering all the possible cases:

$$\pi_{\vec{Y}}(\vec{X})_i^{-1}((\infty,a]) = \cup_{k=1}^n \left(X_k^{-1}((\infty,a]) \cup B_k^i\right)$$

Since it is the (finite) union of measurable sets, then $\pi_{\vec{Y}}(\vec{X})_i^{-1}((\infty,a]) \in F$ for any $i \in \{1,\dots,n\}$ and $\pi_{\vec{Y}}(\vec{X})$ is a random vector [45]. Thus, any linear combination of their components is a random variable, and in particular it holds for $\text{IOLF}(\vec{X},\vec{Y};\vec{w})$. □

In the following we show two of the extreme cases in which the expected value with respect to $\vec{Y}$ of the IOLF operator has a simple expression. Let us first introduce the definition of exchangeability:

Definition 15 ([33]). A random vector $\vec{X}$ is said to be exchangeable if any rearrangement of its components has the same distribution as $\vec{X}$.

Proposition 16. *Let $\vec{X}$ and $\vec{Y}$ be two random vectors and $\vec{w}$ a weight vector. The following statements hold:*

1. *If $\vec{Y}$ and $\vec{X}$ are independent, then:*

$$E_{\vec{Y}}[IOLF(\vec{X},\vec{Y};\vec{w})] = LF(\vec{X};\hat{\vec{w}}) = \sum_{k=1}^n \hat{w}_k X_k,$$

$$\hat{w}_k = \sum_{i=1}^n w_i P(Y_k = Y_{(i)})$$

2. *If $\vec{Y}$ and $\vec{X}$ are independent and $\vec{Y}$ is exchangeable, then*

$$E_{\vec{Y}}[IOLF(\vec{X},\vec{Y};\vec{w})] = \sum_{k=1}^n \frac{1}{n} X_k$$

Proof.

1. Since $\vec{X}$ and $\vec{Y}$ are independent, the distribution of $X_k | Y_k = Y_{(i)}$ is the same as the distribution of X_i for any $i,k \in \{1,\dots,n\}$. Thus, we can compute the density function of $\pi_{\vec{Y}}(\vec{X})$ as follows:

$$f_{\pi_{\vec{Y}}(\vec{X})_i}(t) = \sum_{k=1}^n f_{X_k | Y_k = Y_{(i)}}(t)$$

$$= \sum_{k=1}^n P\left(Y_k = Y_{(i)}\right) f_{X_k}(t)$$

Computing the expected value regarding $\vec{Y}$ leads us to the following expression:

$$E_{\vec{Y}}[\text{IOWA}(\vec{X},\vec{Y};\vec{w})] = E_{\vec{Y}}[\sum_{i=1}^n w_i \pi_{\vec{Y}}(\vec{X})_i]$$

$$= \sum_{i=1}^n w_i E_{\vec{Y}}[\pi_{\vec{Y}}(\vec{X}_i)] = \sum_{i,k=1}^n w_i P\left(Y_k = Y_{(i)}\right) X_k$$

$$= \sum_k \hat{w}_k X_k = \text{WA}(\vec{X};\hat{\vec{w}})$$

2. Notice that if $\vec{Y}$ is exchangeable, then $P\left(Y_k = Y_{(i)}\right) = \frac{1}{n}$ for any $k,i \in \{1,\dots,n\}$. □

Remark 17. Notice that, if we consider only positive weights, the latter result holds for the IOWA operator and the expression $\sum_{k=1}^n \hat{w}_k X_k$ coincides with a Weighted Averaging Mean.

We want to remark that the most interesting cases are the intermediate ones, in which the order induced by $\vec{Y}$ is not the same as the order induced by $\vec{X}$ and neither $\vec{X}$ and $\vec{Y}$ are independent.

The mean vector and the covariance matrix of $\text{IOLF}(\vec{X},\vec{Y};\vec{w})$ in terms of the random vector $\pi_{\vec{Y}}(\vec{X})$, has a straightforward expression based on the properties of the mean and variance of a linear transformation of a random vector.

Proposition 18. *Consider two random vectors $\vec{X},\vec{Y}$ and a vector of weights $\vec{w}$. Then $IOLF(\vec{X},\vec{Y};\vec{w})$ satisfies:*

$$E\left[IOLF(\vec{X},\vec{Y};\vec{w})\right] = \vec{w}' E\left[\pi_{\vec{Y}}(\vec{X})\right]$$

$$Var\left[IOLF(\vec{X},\vec{Y};\vec{w})\right] = \vec{w}' Var\left[\pi_{\vec{Y}}(\vec{X})\right] \vec{w}$$

4. Mean estimation using IOLF operator

In this section we study the applicability of the IOLF aggregation over random variables in the estimation of a common mean. We want to remark that we will associate the random variables with the prediction of several time series forecasting models and the common mean will be identified as the real value of the time series.

In this direction, we consider a random vector $\vec{X}$ for which the mean of every component equals the same value μ. Thus, we can decompose $\vec{X} = \mu\vec{1} + \vec{Z}$, being $\vec{Z}$ a noise random vector for which the mean of every component equals 0 and μ a parameter that we want to estimate. We also consider in the following that $\vec{Y}$ has not a dependence with μ, because in these cases we may be tempted to use $\vec{Y}$ in order to estimate it. This is a reasonable assumption, as we have discussed in Section 2.3. Finally, we suppose that the matrix $Var\left[\pi_{\vec{Y}}(\vec{X})\right]$ is invertible. No additional considerations on $\vec{X}$ and $\vec{Y}$ are made, the involved random variables can be dependant and may have different distribution. This is a very flexible scheme in which the prediction models can have a quite different behaviour and can be related. In the simplest case, they may be a collection of $2n$ independent and identically distributed random variables.

When considering an observation of the random vectors $\vec{X}$ and $\vec{Y}$, how we can use $\text{IOLF}(\vec{X},\vec{Y};\vec{w})$ to estimate μ? In the following we will denote $E\left[\pi_{\vec{Y}}(\vec{X})\right]-\mu\vec{1}$ as $\vec{\Delta}$, i.e. the drift with respect to the global mean, and $Var\left[\pi_{\vec{Y}}(\vec{X})\right]$ as Σ. With this change of notation and noticing that the elements of $\vec{w}$ sum 1, the expressions of Proposition 18 have the following form:

$$E\left[\text{IOLF}(\vec{X},\vec{Y};\vec{w})\right] = \mu + \vec{\Delta}'\vec{w}$$
$$Var\left[\text{IOLF}(\vec{X},\vec{Y};\vec{w})\right] = \vec{w}'\Sigma\vec{w}$$

Notice that, since $\sum_{i=1}^{n}\pi_{\vec{Y}}(\vec{X}) = \sum_{i=1}^{n}X_i$, then it holds that $\sum_{i=1}^{n}\Delta_i = 0$. We also want to remark that $\text{IOLF}(\vec{X},\vec{Y},\vec{w})$ is an unbiased estimator for μ if and only if $\vec{\Delta}$ and $\vec{w}$ are orthogonal. In the following result we explore two elementary situations where this property holds.

Proposition 19. *Let $\vec{X}$ be a random vector with all the components having the same mean μ. Let $\vec{Y}$ be a random vector and let $\vec{w}$ be a vector of weights.*

1. *If $\vec{Y}$ and $\vec{X}$ are independent and $\vec{Y}$ is exchangeable, then IOLF $(\vec{X},\vec{Y};\vec{w})$ is an unbiased estimator for μ.*
2. *If the components of $\vec{X}$ are symmetric iid random variables, $\vec{X} = \vec{Y}$ and $\vec{w}$ satisfy $w_k = w_{n-k+1}$ for any $k \in \{1,\dots,n-1\}$, then $IOLF(\vec{X},\vec{Y};\vec{w})$ is an unbiased estimator for μ*

Proof.

1. The first statement is a direct consequence of 2 in Proposition 16, since in this case $\vec{\Delta} = \vec{0}$
2. Since $\vec{X} = \vec{Y}$ and the components of $\vec{X}$ are iid random variables, the IOLF $\left(\vec{X},\vec{Y}\right)$ operator is a linear combination of the order statistics of $\vec{X}$. Moreover, since the distribution is symmetric, we have that $\Delta_k = \Delta_{n-k+1}$ if $k \neq \frac{n+1}{2}$. If n is odd and $k = \frac{n+1}{2}$, then $\Delta_k = 0$. Thus, since $w_k = w_{n-k+1}$ for any $k \in \{1,\dots,n-1\}$, $\vec{w}'\vec{\Delta} = 0$ □

In general, we want to sacrifice the unbiased property of our estimator to reduce the Mean Squared Error (MSE). Since we are allowing negative weights, we can achieve a closed expression of the optimal weights using Lagrange Multipliers.

Theorem 20. *Consider a random vector $\vec{X}$ with the same mean μ for all its components and a random vector $\vec{Y}$. Then, the vector of weights $\vec{w}$ (verifying that $\sum_{i=1}^{n}w_i = 1$) which minimize $E\left[\left(\mu - IOLF(\vec{X},\vec{Y};\vec{w})\right)^2\right]$ is:*

$$\vec{w} = \frac{\left(\Sigma + \vec{\Delta}\vec{\Delta}'\right)^{-1}\vec{1}}{\vec{1}'\left(\Sigma + \vec{\Delta}\vec{\Delta}'\right)^{-1}\vec{1}} \tag{3}$$

Proof. We express $E\left[\left(\mu - \text{IOLF}(\vec{X},\vec{Y};\vec{w})\right)^2\right]$ as $\vec{w}'\Sigma\vec{w} + \left(\vec{w}'\vec{\Delta}\right)^2$. Then, we consider the following optimization problem:

Minimize $\vec{w}'\Sigma\vec{w} + \left(\vec{w}'\vec{\Delta}\right)^2$

Subject to $\vec{1}'\vec{w} = 1$

Noticing that $\left(\vec{w}'\vec{\Delta}\right)^2 = \vec{w}'\vec{\Delta}\vec{\Delta}'\vec{w}$ and using Lagrange multipliers, the expression is the following:

$$\vec{w}'\left(\Sigma + \vec{\Delta}\vec{\Delta}'\right)\vec{w} - \lambda\left(\vec{1}'\vec{w} - 1\right)$$

Deriving by $\vec{w}$ and equalling to 0:

$$2\left(\Sigma + \vec{\Delta}\vec{\Delta}'\right)\vec{w} - \lambda\vec{1} = 0,\ \vec{w} = \frac{\lambda}{2}\left(\Sigma + \vec{\Delta}\vec{\Delta}'\right)^{-1}\vec{1}$$

Then, substituting in the restriction:

$$\frac{\lambda}{2}\vec{1}'\left(\Sigma + \vec{\Delta}\vec{\Delta}'\right)^{-1}\vec{1} = 1, \qquad \frac{\lambda}{2} = \frac{1}{\vec{1}'\left(\Sigma + \vec{\Delta}\vec{\Delta}'\right)^{-1}\vec{1}}$$

$$\vec{w} = \frac{\left(\Sigma + \vec{\Delta}\vec{\Delta}'\right)^{-1}\vec{1}}{\vec{1}'\left(\Sigma + \vec{\Delta}\vec{\Delta}'\right)^{-1}\vec{1}} \quad \square$$

The latter formula can be used also when fitting OLF and LF operators. Notice that if we use the sample covariance matrix and the sample mean in order to estimate Σ and $\vec{\Delta}$, the here-presented problem is equivalent to Eq. (2), but without the requirement that the weights must be positive. In addition, this result gives a closed expression of the optimal weights, which is useful both for proving probabilistic properties and for facilitating calculations.

Corollary 21. *Consider a random vector $\vec{X}$ with the mean of all its components equal to μ and a random vector $\vec{Y}$. Then, for all possible values of $\vec{w}$, the minimum value of $E\left[\left(\mu - IOLF(\vec{X},\vec{Y};\vec{w})\right)^2\right]$ is:*

$$E\left[\left(\mu - IOLF(\vec{X},\vec{Y};\vec{w})\right)^2\right] = \frac{1}{\vec{1}'\left(\Sigma + \vec{\Delta}\vec{\Delta}'\right)^{-1}\vec{1}}$$

We end this section establishing cases where the weights computed in Theorem 20 lead to an unbiased estimator.

Proposition 22. *Consider a random vector $\vec{X}$ with the same mean μ for all its components and a random vector $\vec{Y}$. If one of the following conditions are fulfilled:*

1. *$\vec{\Delta} = \vec{0}$*
2. *$\vec{X} = \vec{Y}$ and the components of $\vec{X}$ are independent and symmetric,*

then the vector of weights $\vec{w}$ (verifying that $\sum_{i=1}^{n}w_i = 1$) that minimizes $E\left[\left(\mu - IOLF(\vec{X},\vec{Y};\vec{w})\right)^2\right]$ makes $IOLF(\vec{X},\vec{Y};\vec{w})$ be an unbiased estimator of μ.

Proof. We recall that $\vec{w}'\vec{\Delta} = 0$ is a sufficient condition for the IOLF operator to be an unbiased estimator of μ. If $\vec{\Delta} = \vec{0}$, then $\vec{w}'\vec{\Delta} = 0$ regardless of the expression of $\vec{w}$.

For the other case, without lose of generality, suppose that $\mu = 0$. The components of $\pi_{\vec{Y}}\left(\vec{X}\right)$ are the order statistics of $\vec{X}$, sorted from

the greatest to the lowest. Since the distribution is symmetric, then the distribution of $X_{(i)}$ is the same as the one of $-X_{(n+1-i)}$.

As a consequence, Σ is persymmetric ($\Sigma_{i,j} = \Sigma_{n+1-i,n+1-j}$ [46]), since $E[X_{(i)}, X_{(j)}] = E[X_{(n+1-i)}, X_{(n+1-j)}]$ for any $i, j \in \{1, \dots, n\}$. Thus, since the inverse of a persymmetric matrix is persymmetric [46], Σ^{-1} is also persymmetric. Then, the weights that minimize the variance, which are $\vec{w} = \frac{\Sigma^{-1}\vec{1}}{\vec{1}\Sigma^{-1}\vec{1}}$ holds that $w_i = w_{n+1-i}$ for any $i \in \{1, \dots, n\}$. Also, we have that $\Delta_i = -\Delta_{n+1-i}$ for any $i \in \{1, \dots, n\}$. Then, $\vec{w}'\Delta = 0$. □

In summary, the IOLF operator can be used for estimation of a common mean μ of the components of a random vector. If $\vec{Y}$ and $\vec{X}$ are independent and $\vec{Y}$ is exchangeable, or if $\vec{X}$ are symmetric iid random variables, $\vec{X} = \vec{Y}$ and $\vec{w}$ satisfy $w_k = w_{n-k+1}$ for any $k \in \{1, \dots, n-1\}$, IOLF$(\vec{X}, \vec{Y}; \vec{w})$ is an unbiased estimator of μ. The weights that minimize the Mean Squared Error can be derived as the closed expression of Eq. (3) using the drift vector $\vec{\Delta}$ and the covariance matrix Σ of the ordered random vector. This optimal weights makes IOLF to be an unbiased estimator if the drift is null or $\vec{X} = \vec{Y}$ and the components of $\vec{X}$ are independent and symmetric.

5. An illustrative example for time series forecasting

We devote this section to describe a practical example that illustrates the benefits of the use of the IOLF operator.

Time series forecasting is a task used to obtain estimates of future values of different measurements in the real world. Models used for predicting time series values are commonly regressors, i.e. any variable in a regression model that is used to predict a response variable. If we consider a set of regression models, they may have several problems: the first is that the models are weak when taken individually (there are models that obtain admissible errors only in some cases) and the second is that the regressors have similar effects. In order to try to avoid these two problems and obtain the most optimal model, combining them becomes a fundamental task.

5.1. Description of the experimental procedure

In this example, we fuse the forecasts using different prediction models based on aggregation and pre-aggregation functions. Three of the alternatives are the WAM, the OWA and the IOWA operator considering as the inducing vector the precision in the previous time step, as explained in Section 2.3. We use these alternatives since they appear in the recent literature as the prominent examples of aggregation functions used as ensembles in time series forecasting, see Section 1.2. They are also closely related with our proposal, thus we should compare the results with them.

The other three alternatives are the LF, the OLF and the IOLF operator considering the same inducing vector as in the case of the IOWA. Notice that we can see the three first cases as IOWA operators with different inducing vectors and the three second cases as the same models but allowing negative weights, being all of them particular cases of the IOLF operator introduced in Definition 12. In this example we will use 7 different prediction models.

Taking into account the notation of Section 2.3, p_{it} is the prediction of model i at time t, and thus, we obtain the vector $\vec{p} = (p_{1t}, \dots, p_{7t})$ of the predictions of the 7 models at time t. On the other hand, since we do not know the real value of the predictions at time t but we do know them at time $t-1$, we use these predictions to obtain the induction vector of the IOWA and IOLF operator. In this sense, we use Eq. (1) to obtain the vector $\vec{y} = (y_{1,t-1}, \dots, y_{7,t-1})$. In the case of OWA and OLF operators, we order the predictions from the greatest to the smallest for each time.

Therefore, the fused predicted values (for time t) are the ones obtained in the following way:

$$\hat{p}_{IOWA} = \text{IOWA}(\vec{p}; \vec{y}; \vec{w}) = \vec{w}'\pi_{\vec{y}}(\vec{p}), \qquad \hat{p}_{IOLF} = \text{IOLF}(\vec{p}, \vec{y}; \vec{w}),$$

$$\hat{p}_{OWA} = \text{OWA}(\vec{p}; \vec{w}) = \sum_{i=1}^{7} w_i p_{\sigma(i)_t t}, \qquad \hat{p}_{OLF} = \text{OLF}(\vec{p}; \vec{w}),$$

$$\hat{p}_{WAM} = \text{WAM}(\vec{p}; \vec{w}) = \sum_{i=1}^{7} w_i p_{it}, \qquad \hat{p}_{LF} = \text{LF}(\vec{p}; \vec{w}).$$

where σ_t is a permutation such that $p_{\sigma(1)_t t} \geq \cdots \geq p_{\sigma(7)_t t}$.

The optimal weights for IOWA, OWA and WAM operators are computed numerically by solving the problem stated in (2). In the case of the IOLF, OLF and LF operators, we can compute the optimal weights directly by applying Theorem 20. We want to remark that, in some cases, inverting the matrix $\Sigma + \vec{\Delta}\vec{\Delta}'$ can be not easy if the dimension is too high or is ill-conditioned. In this cases, we can always solve the optimization problem numerically, as it is done for the IOWA, OWA and WAW operator.

In this example, we use the latter six alternatives to combine different data forecasting model outputs, such as temperature and humidity. The time series of the data set [47] are composed of almost 20 000 observations, which are measured every 10 min for about 4.5 months. The house temperature and humidity conditions are monitored with a ZigBee wireless sensor network. Therefore, the time series measure the temperature (T) and humidity (RH) in different areas.

These time series have several characteristics in common. Firstly, there exists a strong seasonal component with a period of one day. On the other hand, there not exist a weekly or monthly seasonal component. Secondly, a moderate amount of outliers appear in the data. Thirdly, the values of the time series seem to have a bell-shaped distribution. For more information in this regard, we refer the reader to [47].

For each of these eighteen time series, the data was divided in the first 70% of the days, the training sample, and the remaining 30%, the test sample. Over the training samples, seven different forecasting models were fitted. The used models are the following:

- RF Random Forest [48]. This regression method builds a set of decision trees in the training process. It returns the average prediction of the individual trees. In this case, the number of trees (estimators) to be used is set to 1000.
- GB Gradient Boosting [49]. This method uses decision trees, which when weak are boosted by the gradient. A gradient boosting model is built in stages by optimizing based on a cost function. The number of stages is set to 1000. The learning rate of the model is set to 0.1 and the loss function used is the Friedman Mean Squared Error [50].
- ARI ARIMA [51]. Autoregressive integrated moving average (ARIMA) model is a statistical model that uses variations and regressions of statistical data in order to find patterns for prediction into the future. It is a model that tries to identify coefficients and number of regressions to be used and since it is a dynamic model, predictions are not based on independent variables but on past data. In this case, the parameters are according to the Akaike's Information Criterium.
- KNN Regression based on K-Nearest Neighbors [52]. We set $k = 3$. This classical method is based on the entry of k nearest examples in the data set. The predicted value is assigned to the average of the values of the k nearest neighbours. The function used to measure the distance between examples has been the Euclidean distance.
- BR Bagging regressor [53]. Bagging is a general variance reduction method based on the use of bootstrap (a technique for estimating variances), together with a decision tree. For regression trees, many trees are grown (without pruning) and the mean of the predictions is calculated. An additional advantage of bagging is that it allows estimating the prediction error directly, without the need to use a test sample or to apply cross-validation. The number of trees used in this model is fixed to 1000.

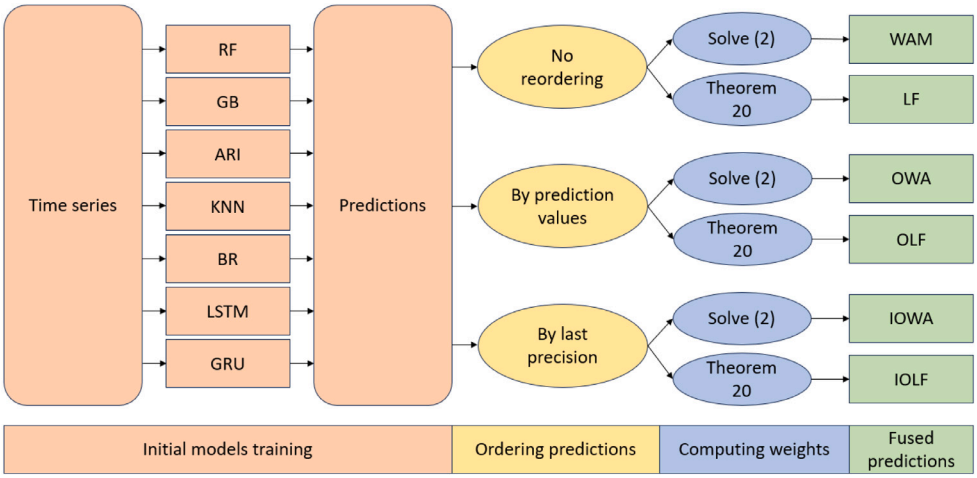

Fig. 1. Diagram showing the followed steps to obtain the fused prediction of each considered ensemble.

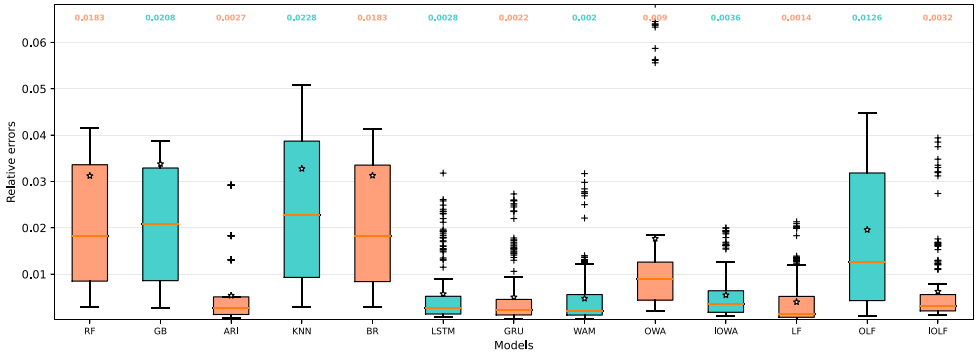

Fig. 2. Boxplot of the relative errors of the different models and their aggregations. The median the errors for each model and aggregation is at the top of the graph.

Finally, models 6 and 7 are models based on recurrent neural networks (RNN). Specifically, we use models based on Long Short-Term Memories (LSTM) [54] and Gated Recurrent Unit (GRU) [55] neurons. These models, introduced in 1997 and 2014 respectively, were designed to avoid the training problems of classical RNNs with input data having long dependencies. To avoid these problems, these neural networks use multiplicative units called gates, which control the flow of information being fed into the network. On the other hand, these models allow information to be stored in short- and long-term memories for use at future points in time.

LSTM LSTM-based model [54]. The first RNN-based model consists of a single LSTM layer with a hidden size of 64 elements and a dense layer. The loss function used in this model is the Mean Squared Error, and the optimization method is the Adam algorithm with a learning rate of 0.01. 1200 epochs have been run. At the input of the LSTMs the data are normalized to the range (−1,1) by min–max, and denormalised at the output.

GRU GRU-based model [55]. The second RNN-based model is composed of a single GRU layer with a hidden size of 128 elements and a dense layer. Dropout of 0.4 has been used. The loss function used in this model is the Mean Squared Error, and the optimization method is the Adam algorithm with a learning rate of 0.01. 1200 epochs have been run. The input data are normalized to the range (−1,1) by min–max, and denormalised at the output.

Notice that the latter models belong to different families and its behaviour will be, in general, very different. We can summarize our procedure in 4 steps, as it can be seen in Fig. 1:

1. Train the prediction models,
2. Reorder the prediction vectors for each time,
3. Compute the optimal weights,
4. Obtain the fused predictions.

5.2. Results

The performance of the initial models and the fusion operators are compared using the test sample. Since some of the forecasting models involve random initialization or dropout, we have repeated this procedure 10 times for each of the time series. The result for each time series and prediction model and fusion operator in the test sample can be found in Table 3. For the Mean Squared Error, in 4 of the time series the best performance was made by an initial prediction model, in other 4 one of the classical aggregation functions was the best option and in the remaining 10 time series, one of the here-proposed fusion operators with negative weights had the lowest MSE.

In order to compare globally the performance of the different alternatives, the boxplot associated with the relative error of all the executions, 10 for each of the 18 time series, can be found in Fig. 2. The relative error has been computed by dividing the square of the MSE by the mean value of the corresponding time series.

Qualitatively, the best initial prediction models seem to be ARIMA, LSTM and GRU. Between the fusion operators, the WAM and LF seem to be the better options, followed by the IOWA and IOLF. In order to have a quantitative comparison, we also performed pairwise Mann–Whitney U rank tests or Wilcoxon tests (see [33]), considering the 180 paired samples of Mean Squared Errors. The p-values can be found in Table 2, in which the alternative hypothesis has been chosen as the row model having a smaller MSE than the column model.

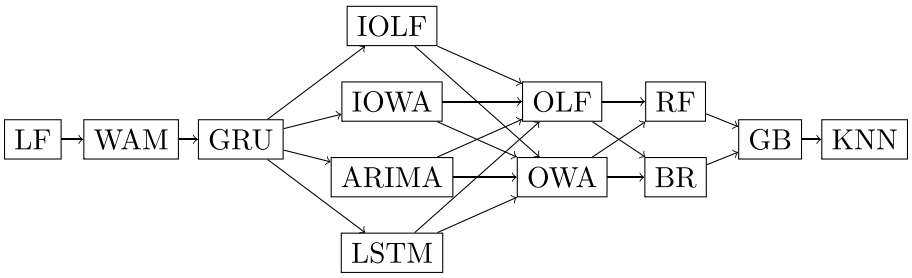

Fig. 3. Prediction models and fusion operators ordered from left to right according to having a significantly smaller MSE according to the Wilcoxon test.

Table 2
P-values associated with the Wilcoxon test with null hypothesis: the row model has a bigger Mean Squared Error than the column model.

	RF	GB	ARIMA	KNN	BR	LSTM	GRU	WAM	OWA	IOWA	LF	OLF	IOLF
RF	–	0.017	1	0	0.536	1	1	1	1	1	1	1	1
GB	0.983	–	1	0	0.983	1	1	1	1	1	1	1	1
ARIMA	0	0	–	0	0	0.752	1	1	0	0.744	1	0	0.388
KNN	1	1	1	–	1	1	1	1	1	1	1	1	1
BR	0.464	0.017	1	0	–	1	1	1	1	1	1	1	1
LSTM	0	0	0.248	0	0	–	0.997	1	0	0.431	1	0	0.086
GRU	0	0	0	0	0	0	–	0.997	0	0	1	0	0
WAM	0	0	0	0	0	0	0.003	–	0	0	1	0	0
OWA	0	0	1	0	0	1	1	1	–	1	1	0.36	1
IOWA	0	0	0.257	0	0	0.569	1	1	0	–	1	0	0.888
LF	0	0	0	0	0	0	0	0	0	0	–	0	0
OLF	0	0	1	0	0	1	1	1	0.64	1	1	–	1
IOLF	0	0	0.612	0	0	0.914	1	1	0	0.112	1	0	–

Table 3
Mean and Standard Deviation (SD) of Mean Squared Error and Mean Absolute Error for the prediction models and their aggregations in the considered time series. The lowest error of each row is in bold.

DS	Mean Squared Error												
	RF	GB	ARI	KNN	BR	LSTM	GRU	WAM	OWA	IOWA	LF	OLF	IOLF
T1	0.256 ± 0.00105	0.276 ± 0.00264	0.0004 ± 0.0	0.271 ± 3e−05	0.255 ± 0.000708	0.000849 ± 0.0013	0.000323 ± 0.000146	0.00121 ± 0.00058	0.0566 ± 0.00657	0.022 ± 0.0182	**0.000236 ± 0.000117**	0.0936 ± 0.00638	0.0214 ± 0.0178
RH1	0.188 ± 0.000976	0.219 ± 0.00141	0.043 ± 0.0	0.436 ± 4e−06	0.188 ± 0.00144	0.0419 ± 0.00442	0.0417 ± 0.00403	0.0506 ± 0.00517	0.132 ± 0.00371	0.0499 ± 0.00246	0.0499 ± 0.00992	0.0942 ± 0.00227	**0.0439 ± 0.00156**
T2	0.602 ± 0.00105	0.618 ± 4.8e−05	0.0086 ± 0.0	0.619 ± 3e−06	0.602 ± 0.00102	**0.00638 ± 0.00131**	0.00808 ± 0.00116	0.0305 ± 0.0108	0.0505 ± 0.0106	0.0101 ± 0.00384	0.0111 ± 0.00313	0.599 ± 0.0205	0.00698 ± 0.00179
RH2	0.765 ± 0.00422	0.953 ± 0.00548	0.0425 ± 0.0	1.0 ± 3e−06	0.766 ± 0.00296	0.215 ± 0.478	0.0153 ± 0.00284	0.011 ± 0.00335	0.384 ± 0.0622	0.0248 ± 0.0102	**0.00802 ± 0.00271**	0.267 ± 0.0256	0.0205 ± 0.00779
T3	0.0794 ± 0.000328	0.0833 ± 0.000625	0.0013 ± 0.0	0.116 ± 4e−06	0.0794 ± 0.00023	0.00183 ± 0.00379	0.00104 ± 0.000435	**0.000725 ± 0.000641**	0.0163 ± 0.00303	0.000918 ± 0.000588	0.000959 ± 0.000821	0.0513 ± 0.0057	0.000976 ± 0.00031
RH3	0.0516 ± 0.000106	0.0452 ± 0.000304	0.0122 ± 0.0	0.0628 ± 8e−06	0.0517 ± 0.000234	0.0043 ± 0.00186	0.00409 ± 0.000824	0.00315 ± 0.00142	0.0291 ± 0.00103	0.00454 ± 0.000801	**0.0029 ± 0.00127**	0.0129 ± 0.00087	0.00473 ± 0.000703
T4	0.49 ± 0.00287	0.471 ± 5.7e−05	**0.000737 ± 0.0**	0.478 ± 9e−06	0.489 ± 0.00249	0.00223 ± 0.00516	0.014 ± 0.0389	0.00229 ± 0.00237	0.03 ± 0.0107	0.00111 ± 0.00113	0.0011 ± 0.00145	0.426 ± 0.0461	0.00454 ± 0.00274
RH4	0.0312 ± 0.000112	0.0321 ± 0.000118	0.00777 ± 0.0	0.0392 ± 4e−06	0.0312 ± 0.000136	0.0457 ± 0.115	0.00147 ± 0.000334	0.000661 ± 0.000542	0.0166 ± 0.000877	0.00325 ± 0.000673	**0.000589 ± 0.000495**	0.00761 ± 0.000287	0.00321 ± 0.00065
T5	0.655 ± 0.00335	0.424 ± 0.00541	**0.000769 ± 0.0**	0.949 ± 1.2e−05	0.649 ± 0.00455	0.00311 ± 0.00145	0.00437 ± 0.00133	0.00641 ± 0.00386	0.00709 ± 0.00368	0.0115 ± 0.00564	0.00636 ± 0.00792	0.497 ± 0.0394	0.00575 ± 0.00403
RH5	16.0 ± 0.61	45.8 ± 0.116	0.448 ± 0.0	6.72 ± 2e−06	16.6 ± 0.805	0.747 ± 0.208	0.672 ± 0.0923	**0.444 ± 0.00743**	9.83 ± 1.31	0.682 ± 0.0361	0.45 ± 0.0209	4.16 ± 0.405	0.705 ± 0.0494
T6	2.32 ± 0.00771	2.26 ± 0.00909	0.0534 ± 0.0	2.47 ± 4e−06	2.32 ± 0.00536	0.0335 ± 0.00838	0.0377 ± 0.00374	0.0428 ± 0.0161	1.03 ± 0.11	**0.0217 ± 0.00424**	0.0239 ± 0.00486	0.498 ± 0.0628	0.0728 ± 0.0148
RH6	0.661 ± 0.00127	0.914 ± 0.00604	0.997 ± 0.0	1.4 ± 0.0	0.661 ± 0.00241	0.914 ± 0.214	0.773 ± 0.497	0.437 ± 0.119	0.505 ± 0.024	**0.407 ± 0.0473**	0.426 ± 0.109	0.476 ± 0.027	0.439 ± 0.0525
T7	0.268 ± 0.000603	0.277 ± 3e−06	0.000196 ± 0.0	0.258 ± 7e−06	0.268 ± 0.000523	0.000479 ± 0.000293	0.000127 ± 0.000121	0.000312 ± 0.000448	0.0547 ± 0.00951	0.00873 ± 0.00755	**0.00012 ± 9.4e−05**	0.106 ± 0.00787	0.00815 ± 0.0076
RH7	0.0105 ± 2.7e−05	0.00915 ± 2.6e−05	0.00854 ± 0.0	0.0107 ± 3e−06	0.0105 ± 4.5e−05	0.0529 ± 0.145	0.00299 ± 0.000512	0.00156 ± 0.00103	0.00566 ± 0.000405	0.00312 ± 0.000576	**0.0014 ± 0.000954**	0.00357 ± 0.00123	0.00318 ± 0.000594
T8	0.105 ± 0.000305	0.101 ± 0.000242	0.000523 ± 0.0	0.114 ± 3.3e−05	0.106 ± 0.000294	0.00152 ± 0.00162	0.000142 ± 3.9e−05	0.000232 ± 0.000183	0.0339 ± 0.00693	0.00717 ± 0.00516	**9.7e−05 ± 9.4e−05**	0.0368 ± 0.00649	0.00696 ± 0.00516
RH8	0.0219 ± 2.6e−05	0.0208 ± 1.8e−05	0.024 ± 0.0	0.0221 ± 1e−06	0.0219 ± 2.3e−05	0.013 ± 0.00191	0.0163 ± 0.00796	0.0108 ± 0.00144	0.0157 ± 0.000496	0.012 ± 0.00107	**0.0101 ± 0.00198**	0.0137 ± 0.000537	0.0126 ± 0.00114
T9	0.321 ± 0.00647	0.233 ± 0.00135	**8.5e−05 ± 0.0**	0.905 ± 4e−06	0.32 ± 0.00729	0.0126 ± 0.0125	0.00268 ± 0.00213	0.00125 ± 0.000633	0.0585 ± 0.00866	0.0148 ± 0.0039	0.000238 ± 0.000155	0.185 ± 0.0235	0.0123 ± 0.0038
RH9	0.123 ± 0.000821	0.128 ± 0.00106	0.0142 ± 0.0	0.15 ± 6e−06	0.123 ± 0.000681	0.0144 ± 0.0088	0.0224 ± 0.0438	0.00432 ± 0.00198	0.0714 ± 0.00762	0.00864 ± 0.00476	**0.00333 ± 0.00196**	0.0327 ± 0.00253	0.00844 ± 0.00417
DS	Mean Absolute Error												
	RF	GB	ARI	KNN	BR	LSTM	GRU	WAM	OWA	IOWA	LF	OLF	IOLF
T1	0.258 ± 0.000673	0.268 ± 0.00158	0.0149 ± 0.0	0.27 ± 0.00043	0.257 ± 0.000436	0.0179 ± 0.0132	0.0134 ± 0.00332	0.021 ± 0.00491	0.125 ± 0.00609	0.0755 ± 0.0251	**0.0107 ± 0.00318**	0.157 ± 0.00535	0.0746 ± 0.0251
RH1	0.183 ± 0.000372	0.183 ± 0.000341	0.0981 ± 0.0	0.261 ± 8.2e−05	0.183 ± 0.000459	0.0998 ± 0.00418	0.103 ± 0.0169	0.107 ± 0.00577	0.135 ± 0.00169	0.102 ± 0.00246	0.113 ± 0.0105	0.137 ± 0.00116	**0.0953 ± 0.002**
T2	0.291 ± 0.000362	0.296 ± 4.8e−05	0.0526 ± 0.0	0.308 ± 3.5e−05	0.292 ± 0.000369	**0.0466 ± 0.00426**	0.0561 ± 0.00253	0.0872 ± 0.0158	0.108 ± 0.00949	0.0592 ± 0.00962	0.0644 ± 0.00795	0.293 ± 0.00417	0.0532 ± 0.00459
RH2	0.308 ± 0.00102	0.333 ± 0.00114	0.11 ± 0.0	0.361 ± 1.5e−05	0.309 ± 0.000542	0.219 ± 0.262	0.0741 ± 0.00823	0.0539 ± 0.00698	0.217 ± 0.0154	0.0791 ± 0.0118	**0.0503 ± 0.0088**	0.186 ± 0.0149	0.0743 ± 0.011
T3	0.092 ± 0.000447	0.0909 ± 0.000501	0.02 ± 0.0	0.133 ± 0.000106	0.092 ± 0.000261	0.0237 ± 0.0253	0.0205 ± 0.00457	**0.0154 ± 0.0083**	0.0454 ± 0.00268	0.0165 ± 0.00344	0.0189 ± 0.00953	0.0806 ± 0.00483	0.0185 ± 0.00252
RH3	0.0802 ± 6.6e−05	0.0715 ± 0.000108	0.059 ± 0.0	0.0909 ± 0.000186	0.0802 ± 0.000127	0.0418 ± 0.0122	0.0358 ± 0.00294	**0.034 ± 0.00791**	0.0563 ± 0.000983	0.0374 ± 0.00265	0.0344 ± 0.00826	0.0513 ± 0.00149	0.0383 ± 0.00239
T4	0.362 ± 0.00139	0.348 ± 5.8e−05	**0.0161 ± 0.0**	0.361 ± 0.000124	0.362 ± 0.00122	0.0271 ± 0.0329	0.057 ± 0.0981	0.0306 ± 0.0151	0.0964 ± 0.0161	0.0215 ± 0.008	0.0219 ± 0.0162	0.339 ± 0.0209	0.0401 ± 0.00878
RH4	0.0743 ± 8.9e−05	0.0682 ± 7.8e−05	0.0496 ± 0.0	0.0836 ± 8.8e−05	0.0743 ± 9.1e−05	0.105 ± 0.17	0.024 ± 0.00257	0.0153 ± 0.00559	0.0486 ± 0.00183	0.0321 ± 0.00359	**0.015 ± 0.00543**	0.0385 ± 0.00554	0.0322 ± 0.00362
T5	0.449 ± 0.0013	0.349 ± 0.00269	**0.011 ± 0.0**	0.565 ± 0.000229	0.447 ± 0.00173	0.0363 ± 0.00812	0.0449 ± 0.00841	0.049 ± 0.0152	0.0505 ± 0.0122	0.0619 ± 0.0154	0.0376 ± 0.0313	0.39 ± 0.0175	0.0388 ± 0.017
RH5	0.903 ± 0.0108	1.3 ± 0.00165	0.145 ± 0.0	0.703 ± 1.9e−05	0.913 ± 0.0144	0.414 ± 0.141	0.377 ± 0.0605	**0.168 ± 0.00951**	0.692 ± 0.0275	0.239 ± 0.00674	0.186 ± 0.0271	0.55 ± 0.0227	0.243 ± 0.0144
T6	0.597 ± 0.00101	0.595 ± 0.00127	0.163 ± 0.0	0.639 ± 2.2e−05	0.597 ± 0.000716	0.127 ± 0.0209	0.135 ± 0.00689	0.138 ± 0.038	0.423 ± 0.0164	**0.107 ± 0.014**	0.11 ± 0.019	0.315 ± 0.023	0.165 ± 0.0152
RH6	0.532 ± 0.000844	0.575 ± 0.00125	0.547 ± 0.0	0.681 ± 0.0	0.532 ± 0.000737	0.556 ± 0.0685	0.576 ± 0.283	0.429 ± 0.0602	0.471 ± 0.00971	**0.38 ± 0.0277**	0.421 ± 0.0699	0.466 ± 0.0177	0.381 ± 0.0296
T7	0.248 ± 0.000338	0.25 ± 3.8e−05	0.00891 ± 0.0	0.252 ± 0.000154	0.248 ± 0.000402	0.015 ± 0.00459	0.00728 ± 0.00331	0.00901 ± 0.00418	0.112 ± 0.00904	0.0441 ± 0.0168	**0.00658 ± 0.00263**	0.158 ± 0.00626	0.0422 ± 0.0175
RH7	0.0719 ± 6.5e−05	0.0687 ± 4.3e−05	0.0611 ± 0.0	0.0743 ± 8e−06	0.0719 ± 8.6e−05	0.109 ± 0.195	0.0374 ± 0.00326	0.0268 ± 0.0106	0.0534 ± 0.00194	0.0382 ± 0.00405	**0.0255 ± 0.0101**	0.0434 ± 0.00915	0.0381 ± 0.00409
T8	0.127 ± 0.000235	0.12 ± 0.000179	0.0175 ± 0.0	0.137 ± 0.000624	0.128 ± 0.000221	0.0265 ± 0.0149	0.00837 ± 0.000861	0.00857 ± 0.00217	0.0738 ± 0.00614	0.0353 ± 0.00947	**0.00606 ± 0.00252**	0.075 ± 0.0063	0.0348 ± 0.00965
RH8	0.111 ± 7.6e−05	0.108 ± 3.3e−05	0.109 ± 0.0	0.112 ± 1e−05	0.111 ± 4.7e−05	0.0811 ± 0.0063	0.0929 ± 0.0284	0.0762 ± 0.00561	0.0933 ± 0.00153	0.08 ± 0.00401	**0.0739 ± 0.00822**	0.0897 ± 0.00163	0.0808 ± 0.00424
T9	0.363 ± 0.00389	0.284 ± 0.00109	**0.00529 ± 0.0**	0.635 ± 0.000016	0.362 ± 0.00384	0.0796 ± 0.0426	0.0382 ± 0.0173	0.0236 ± 0.00592	0.151 ± 0.0106	0.0789 ± 0.0116	0.0102 ± 0.0039	0.279 ± 0.0185	0.071 ± 0.0119
RH9	0.135 ± 0.00044	0.134 ± 0.000408	0.08 ± 0.0	0.153 ± 2.9e−05	0.135 ± 0.000319	0.0737 ± 0.0242	0.0919 ± 0.0921	0.0394 ± 0.0123	0.106 ± 0.00427	0.0551 ± 0.0083	**0.036 ± 0.0125**	0.0767 ± 0.00957	0.0546 ± 0.00797

As it can be seen, is statistically accepted that the Linear Fusion operator is better than the rest of alternatives. In addition, the *p*-value between the IOLF and IOWA operator is 0.112, which is small but not enough to be able to draw a conclusion. Nothing can be said about the comparison between the OWA and OLF operator. In order to ease the visualization of the hierarchy between the alternatives, the 13 alternatives are ordered from right to left, being the ones of the left significantly better than the others (see Fig. 3).

Notice that the GRU model is the best between all the initial prediction models and is also better than OWA, IOWA, OLF and IOLF. This can be explained that, since these fusion operators exchange the predictions

positions for each time, it is hard for them to give importance to a particular model.

We want to remark that the benefits of the here-presented methods are related to the appearance of negative weights in the optimization process. This is the case of the Linear Fusion in the latter example, which outperforms the rest of alternatives. However, there are databases and elections of forecasting models in which the negative weights do not appear or are not relevant. In these cases, the behaviour allowing or not negative weights should be similar. This can be see also in this example for the cases of OWA and OLF operators.

We end this section by remarking that the IOLF and derived operators can be used in general time series forecasting, they are not restricted to any field of application or particular type of time series.

6. Conclusions

The use of Induced Ordered Linear Fusion (IOLF) operator, Ordered Linear Fusion (OLF) and Linear Fusion (LF) operators as a way to aggregate prediction models for time series forecasting has been proposed. Firstly, the IOLF operator over random vectors has been defined and some equivalences to another averaging aggregation functions has been determined, in addition to the expressions of its first and second moments. The monotonicity and semantics of negative weights have been studied.

Secondly, the use of the IOLF operator as an estimator for a common mean parameter of the components of the aggregated random vector has been explored. Firstly, the measurability of the IOLF operator has been proved and its mean and variance derived. Some conditions on which the IOLF is an unbiased estimator for this parameter are given. The expressions for the optimal weights that make the IOLF having the lowest MSE or are provided. These results were extended also to the OLF and LF operators.

The benefits of the proposal with respect to the classical aggregation functions have been also discussed. In particular, an illustrative example focused in time series prediction has been presented. Different prediction models and their aggregation have been compared, concluding that the Linear Fusion is better than the rest of alternatives. In particular, it has been statistically accepted that LF improves the performance of WAM thanks to the fact that it relaxes the positivity condition of the weights in the considered data sets.

This probabilistic point of view also allows for future improvement of this type of methods. When using the sample mean and sample covariance matrix as estimators in order to compute the optimal weights, the here-proposed method is equivalent to the traditional methods used in the literature, but allowing for negative weights. However, other estimators can be considered. For instance, some statistical procedures such as robust estimation [56] or online estimation [57] can be considered and easily incorporated into the model. We leave these possibilities as a future line of research.

CRediT authorship contribution statement

Juan Baz: Conceptualization, Methodology, Investigation, Writing – original draft. **Mikel Ferrero-Jaurrieta:** Methodology, Software, Validation, Writing – original draft. **Irene Díaz:** Conceptualization, Investigation, Writing – review & editing, Supervision. **Susana Montes:** Conceptualization, Investigation, Writing – review & editing, Supervision. **Gleb Beliakov:** Writing – review & editing, Supervision. **Humberto Bustince:** Conceptualization, Investigation, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Funding details

J. Baz is partially supported by Programa Severo Ochoa of Principality of Asturias (BP21042). H. Bustince and M. Ferrero-Jaurrieta are supported by Agencia Estatal de Investigación (PID2019-108392GB-I00, AEI/10.13039/501100011033). J. Baz, S. Montes and I. Díaz are supported by the Ministry of Science and Innovation (PDI2022-139886NB-l00). The work of G. Beliakov was supported by the Australian Research Council Discovery project DP210100227.

References

[1] H. Allende, C. Valle, Ensemble methods for time series forecasting, in: Claudio Moraga: A Passion for Multi-Valued Logic and Soft Computing, 2017, pp. 217–232.
[2] J.D. Wichard, M. Ogorzalek, Time series prediction with ensemble models, in: 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541), Vol. 2, IEEE, 2004, pp. 1625–1630.
[3] R.R. Yager, Induced aggregation operators, Fuzzy Sets and Systems 137 (1) (2003) 59–69.
[4] Y. Bin, Y. Hai-Bo, G. Zhen-Wei, A combination forecasting model based on iowa operator for dam safety monitoring, in: 2013 Fifth International Conference on Measuring Technology and Mechatronics Automation, IEEE, 2013, pp. 5–8.
[5] D. Leite, I. Škrjanc, Ensemble of evolving optimal granular experts, owa aggregation, and time series prediction, Inf. Sci. 504 (2019) 95–112.
[6] G.P. Dimuro, J. Fernández, B. Bedregal, R. Mesiar, J.A. Sanz, G. Lucca, H. Bustince, The state-of-art of the generalizations of the choquet integral: from aggregation and pre-aggregation to ordered directionally monotone functions, Inf. Fusion 57 (2020) 27–43.
[7] G. Lucca, J.A. Sanz, H. Bustince, G.P. Dimuro, V. Gomes, R.C.C. Madureira, P. Melo-Pinto, Applying aggregation and pre-aggregation functions in the classification of grape berries, in: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2018, pp. 1–6.
[8] G. Lucca, J. Sanz, G.P. Dimuro, B. Bedregal, H. Bustince, Analyzing the behavior of aggregation and pre-aggregation functions in fuzzy rule-based classification systems with data complexity measures, in: Advances in Fuzzy Logic and Technology 2017, Springer, Berlin, Germany, 2017, pp. 443–455.
[9] C. Dias, J. Bueno, E. Borges, G. Lucca, H. Santos, G. Dimuro, H. Bustince, P. Drews, S. Botelho, E. Palmeira, Simulating the behaviour of choquet-like (pre) aggregation functions for image resizing in the pooling layer of deep learning networks, in: International Fuzzy Systems Association World Congress, Springer, 2019, pp. 224–236.
[10] P. Su, T. Chen, H. Mao, J. Xie, Y. Zhao, J. Liu, On the application of preaggregation functions to fuzzy pattern tree, in: 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2019, pp. 1–6.
[11] J.K. Lindsey, Statistical Analysis of Stochastic Processes in Time, Cambridge University Press, Cambridge, England, 2004.
[12] R.E. Kalman, A new approach to linear filtering and prediction problems, J. Basic Eng. 82 (1960) 35–45.
[13] O. Sagi, L. Rokach, Ensemble learning: A survey, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 8 (4) (2018) 1249.
[14] M. Muzammal, R. Talat, A.H. Sodhro, S. Pirbhulal, A multi-sensor data fusion enabled ensemble approach for medical data from body sensor networks, Inf. Fusion 53 (2020) 155–164.
[15] K. He, Q. Yang, L. Ji, J. Pan, Y. Zou, Financial time series forecasting with the deep learning ensemble model, Mathematics 11 (4) (2023) 1054.
[16] V. Jensen, F.M. Bianchi, S.N. Anfinsen, Ensemble conformalized quantile regression for probabilistic time series forecasting, IEEE Trans. Neural Netw. Learn. Syst. (2022).
[17] Y. Sun, N. Geng, Forecasting civil aviation incident rate in china using a combined prediction model, J. Adv. Transp. 2021 (2021).
[18] D. Liang, L.F. Ma, Y.F. Hu, S. Wu, Combination forecasting method based on iowa operator and application, in: Advanced Materials Research, Vol. 945, Trans Tech Publ, 2014, pp. 2515–2518.
[19] B. Li, J. Ding, Z. Yin, K. Li, X. Zhao, L. Zhang, Optimized neural network combined model based on the induced ordered weighted averaging operator for vegetable price forecasting, Expert Syst. Appl. 168 (2021) 114232.
[20] M. Guojian, B. Zhong, Z. Xiangbo, et al., Combination forecasting model of equipment spare parts demand based on iowa operator, Ordnance Ind. Autom. 32 (1) (2013) 8–11.
[21] G. Jiang, Y. Liu, Research on collaborative forecasting model based on cpfr, in: Software Engineering and Knowledge Engineering: Theory and Practice, Springer, New York, 2012, pp. 523–529.

[22] J. Zhu, P. Wu, H. Chen, J. Liu, L. Zhou, Carbon price forecasting with variational mode decomposition and optimal combined model, Physica A 519 (2019) 140–158.
[23] F. Li, J. Qian, Q. Yan, X. Yang, J. Zhao, K. Qu, Q. Song, A combination forecasting model based on iowa operator for pv generation, in: International Conference on Renewable Power Generation (RPG 2015), IET, 2015, pp. 1–4.
[24] Y. Sun, K. Li, Y. Yan, X. Wei, C. Zhang, Combination load forecasting method for cchp system based on iowa operator, in: 2017 Chinese Automation Congress (CAC), IEEE, 2017, pp. 4193–4197.
[25] G. Liu, F. Xiao, C.-T. Lin, Z. Cao, A fuzzy interval time-series energy and financial forecasting model using network-based multiple time-frequency spaces and the induced-ordered weighted averaging aggregation operation, IEEE Trans. Fuzzy Syst. 28 (11) (2020) 2677–2690.
[26] G. Li, W. Chang, H. Yang, A novel combined prediction model for monthly mean precipitation with error correction strategy, IEEE Access 8 (2020) 141432–141445.
[27] J. Zhu, P. Wu, H. Chen, L. Zhou, Z. Tao, A hybrid forecasting approach to air quality time series based on endpoint condition and combined forecasting model, Int. J. Environ. Res. Public Health 15 (9) (2018) 1941.
[28] V. Cerqueira, L. Torgo, M. Oliveira, B. Pfahringer, Dynamic and heterogeneous ensembles for time series forecasting, in: 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2017, pp. 242–251.
[29] S.N. Lahiri, Theoretical comparisons of block bootstrap methods, Ann. Statist. 27 (1999) 386–404.
[30] X. Qiu, P.N. Suganthan, G.A. Amaratunga, Fusion of multiple indicators with ensemble incremental learning techniques for stock price forecasting, J. Bank. Financ. Technol. 3 (2019) 33–42.
[31] F. Piccialli, F. Giampaolo, E. Prezioso, D. Camacho, G. Acampora, Artificial intelligence and healthcare: Forecasting of medical bookings through multi-source time-series fusion, Inf. Fusion 74 (2021) 1–16.
[32] D. Ruta, B. Gabrys, Neural network ensembles for time series prediction, in: 2007 International Joint Conference on Neural Networks, IEEE, 2007, pp. 1204–1209.
[33] V.K. Rohatgi, A.M.E. Saleh, An Introduction to Probability and Statistics, John Wiley & Sons, New Jersey, USA, 2015.
[34] M. Fréchet, Sur l'extension de certaines évaluations statistiques au cas de petits échantillons, Rev. l'Inst. Int. Statist. 11 (1943) 182–205.
[35] H. Cramér, A contribution to the theory of statistical estimation, Scand. Actuar. J. 10 (1) (1946) 85–94.
[36] C. Radhakrishna Rao, Information and accuracy attainable in the estimation of statistical parameters, Bull. Calcutta Math. Soc. 37 (3) (1945) 81–91.
[37] E. Lloyd, Least-squares estimation of location and scale parameters using order statistics, Biometrika 39 (1/2) (1952) 88–95.
[38] A.E. Sarhan, Estimation of the mean and standard deviation by order statistics, Ann. Math. Stat. 26 (1954) 317–328.
[39] M. Grabisch, J.-L. Marichal, R. Mesiar, E. Pap, Aggregation Functions, Vol. 118, Cambridge University Press, Cambridge, England, 2009.
[40] G. Beliakov, T. Calvo, T. Wilkin, Three types of monotonicity of averaging functions, Knowl.-Based Syst. 72 (2014) 114–122.
[41] T. Wilkin, G. Beliakov, Weakly monotonic averaging functions, Int. J. Intell. Syst. 30 (2) (2015) 144–169.
[42] G. Beliakov, J. Špirková, T. Calvo, On the extended set of weights of the owa functions, Int. J. Gen. Syst. 49 (4) (2020) 355–373.
[43] C.M. Bishop, N.M. Nasrabadi, Pattern Recognition and Machine Learning, Springer, Berlin, Germany, 2006.
[44] K.V. Mardia, J.T. Kent, J.M. Bibby, Multivariate Analysis, Academic Press INC, San Diego, 1979.
[45] V.I. Bogachev, Measure Theory, Springer Science & Business Media, Berlin, Germany, 2007.
[46] G.H. Golub, C.F. Van Loan, Matrix Computations, in: Johns Hopkins studies in the mathematical sciences, Johns Hopkins University Press, Baltimore, USA, 1996.
[47] L. Candanedo, Appliances energy prediction, UCI Mach. Learn. Repos. (2017) http://dx.doi.org/10.24432/C5VC8G.
[48] T.K. Ho, Random decision forests, in: Proceedings of 3rd International Conference on Document Analysis and Recognition, Vol. 1, IEEE, 1995, pp. 278–282.
[49] J.H. Friedman, Stochastic gradient boosting, Comput. Statist. Data Anal. 38 (4) (2002) 367–378.
[50] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Statist. 29 (2001) 1189–1232.
[51] G.E. Box, G.M. Jenkins, G.C. Reinsel, G.M. Ljung, Time Series Analysis: Forecasting and Control, John Wiley & Sons, New Jersey, USA, 2015.
[52] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27.
[53] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140.
[54] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[55] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder–decoder for statistical machine translation, 2014, arXiv:1406.1078.
[56] A. Farcomeni, L. Greco, Robust Methods for Data Reduction, CRC Press, 2016.
[57] W. Zhu, X. Chen, W.B. Wu, Online covariance matrix estimation in stochastic gradient descent, J. Amer. Statist. Assoc. (2021) 1–12.