JOURNAL OF GEOPHYSICAL RESEARCH, VOL. 114, D05307, doi:10.1029/2008JD009978, 2009

Ozone ensemble forecast with machine learning algorithms

Vivien Mallet,[1,2] Gilles Stoltz,[3,4] and Boris Mauricette[1,2,3]

Received 18 February 2008; revised 12 October 2008; accepted 6 November 2008; published 13 March 2009.

[1] We apply machine learning algorithms to perform sequential aggregation of ozone forecasts. The latter rely on a multimodel ensemble built for ozone forecasting with the modeling system Polyphemus. The ensemble simulations are obtained by changes in the physical parameterizations, the numerical schemes, and the input data to the models. The simulations are carried out for summer 2001 over western Europe in order to forecast ozone daily peaks and ozone hourly concentrations. On the basis of past observations and past model forecasts, the learning algorithms produce a weight for each model. A convex or linear combination of the model forecasts is then formed with these weights. This process is repeated for each round of forecasting and is therefore called sequential aggregation. The aggregated forecasts demonstrate good results; for instance, they always show better performance than the best model in the ensemble and they even compete against the best constant linear combination. In addition, the machine learning algorithms come with theoretical guarantees with respect to their performance, that hold for all possible sequences of observations, even nonstochastic ones. Our study also demonstrates the robustness of the methods. We therefore conclude that these aggregation methods are very relevant for operational forecasts.

Citation: Mallet, V., G. Stoltz, and B. Mauricette (2009), Ozone ensemble forecast with machine learning algorithms, *J. Geophys. Res.*, *114*, D05307, doi:10.1029/2008JD009978.

1. Introduction

[2] The large uncertainties in air quality forecasts have lately been evaluated with ensemble approaches. Besides the propagation of input data uncertainties with Monte Carlo simulations [e.g., *Hanna et al.*, 1998; *Beekmann and Derognat*, 2003], multimodel ensembles have been introduced in order to account for the uncertainties in the chemistry-transport models formulation [*Mallet and Sportisse*, 2006b; *van Loon et al.*, 2007]. In this case, the models are based on different physical parameterizations, different numerical discretizations and different input data. Each model in the ensemble brings information that may be used to improve the forecasts.

[3] A class of methods linearly combines the ensemble members in order to produce a single forecast, hopefully more skillful than any individual model of the ensemble. Henceforth, we refer to these methods as aggregation methods. The simplest example is the ensemble mean, which usually brings limited improvement, if any (depending on the target), compared to the best model in the ensemble [*McKeen et al.*, 2005; *van Loon et al.*, 2007]. Other methods associate weights to the models, using the performance of the models against the observations in the past. *Pagowski et al.* [2005, 2006] applied linear regression methods, which resulted in strong improvements. The weights were computed per observation station, which did not enable the forecast of 2D fields of concentrations. Also dynamic linear regression is usually not a robust method [*West and Harrison*, 1997]. In the work of *Mallet and Sportisse* [2006a], the weights only depend on time (so, not on the location) and were computed with least squares optimization like in the work of *Krishnamurti et al.* [2000]. This led to significant improvements in the forecasts. The method seemed quite robust for ozone forecasts, but it is an empirical method, without any theoretical guarantee.

[4] In this paper, we apply some new methods developed by the machine learning community. Just like the ones discussed above, they perform sequential aggregation on the basis of ensemble simulations and past observations. These machine learning methods come with a strong mathematical background [*Cesa-Bianchi and Lugosi*, 2006]. They provide theoretical bounds on the discrepancy between the performance of some best model combinations and the performance of the aggregated forecast, for any possible sequence of observations. Hence, they can provide a reliable and robust framework for ensemble forecasting. The basis of learning algorithms and more detailed explanations for two algorithms and their variants are addressed in section 2.

[1] INRIA, Paris-Rocquencourt Research Center, Rocquencourt, France.
[2] CEREA, Joint Laboratory ENPC-EDF Research and Development, Université Paris-Est, Marne la Vallée, France.
[3] Département de Mathématiques et Applications, École Normale Supérieure, CNRS, Paris, France.
[4] HEC Paris, CNRS, Jouy-en-Josas, France.

0148-0227/09/2008JD009978$09.00

[5] We rely on the same ensemble of ozone forecasts as given by *Mallet and Sportisse* [2006a]. This ensemble was built with the Polyphemus system [*Mallet et al.*, 2007b] and is made of 48 models run during 4 months in 2001 and over Europe. The experiment setup is briefly explained in section 3.

[6] The learning algorithms are applied to ozone daily peaks and to ozone hourly concentrations. Their results are reviewed in section 4. The analysis addresses issues like the robustness of the forecasts and the ability of the methods to capture extreme events.

[7] Machine learning aims at designing and developing algorithms that can be implemented on computers to make some automatic decisions or predictions. (Here, we are interested in predicting the concentrations of a pollutant.) One way of making good decisions is of course to consider a statistical procedure based on a preliminary estimation step. However, not all machine learning algorithms rely on the estimation of statistical parameters. The sequential aggregation techniques described in this paper are examples of such machine learning algorithms not resorting to estimation.

[8] Quite different machine learning techniques have already been intensively used in atmospheric sciences, and in particular, neural networks [see, e.g., *Lary et al.*, 2004; *Loyola*, 2006]. Neural networks combine in a nonlinear way different input data. However, their design is usually a difficult task in view of all possible combinations involved (number of hidden layers, choice of the input categories). In addition to the choice of the structure of the neural network, weights have also to be associated to each possible input and hidden layer. These weights are chosen by the user as well. In total, one obtains this way a given model, whose performance is difficult to study from a theoretical point of view.

[9] In this paper, we are concerned with the aggregation of several models, some of them being possibly given by different neural networks. We thus work at a metamodel level. It is true that neural networks could be used at this level too, to combine some base models in a nonlinear way. However, we focus below on learning techniques that need only one, or maybe two, user choices, as, e.g., the penalization factor λ of the ridge regression forecaster of section 2.3 or the learning rate η of the exponentiated gradient forecaster of section 2.4. Having very few parameters to set up is the only way for the procedure to be carried out in some automatic manner by a computer.

2. Learning Methods as Ensemble Forecasters

2.1. Principle and Notation

[10] Ensemble forecasts are based on a set of models (a multimodel ensemble) $\mathcal{M} = \{1, \ldots, N\}$. Each model may have its own physical formulation, numerical formulation and input data (see section 3). Its forecasts are compared against measurements from a network of monitoring stations $\mathcal{N} = \{1, \ldots, S\}$. (The indexations of $\mathcal{M}$ and $\mathcal{N}$ are made in an arbitrary order.) At station $s \in \mathcal{N}$ and time index $t \in \{1, \ldots, T\}$ (the indexes of the days or of the hours), the prediction $x^s_{m,t}$ of model $m \in \mathcal{M}$ is compared to the observation y^s_t. In practice, the performance of model m is assessed with a root mean square error including observations from all stations and all time indexes (see section 2.2.2).

[11] The general idea is to combine linearly the predictions of the models to get a more accurate forecast. A weight is associated with each model of the ensemble so as to produce an improved aggregated prediction $\widehat{y}^s_t = \sum_{m=1}^N v_{m,t} x^s_{m,t}$. The weights $v_{m,t}$ may also depend on the station s. In this paper, we essentially focus on weights independent of the stations so that the linear combination should still be valid away from the stations. Otherwise, the ensemble methods would compete with purely statistical models whose results are very satisfactory at a low cost (e.g., Étude et simulation de la qualité de l'air en Île-de-France (ESQUIF), rapport final, 2001).

2.2. Sequential Aggregation of Models

[12] The weights of the combination depend of course on the past predictions of each model and on the past observations. The formalized dependence is called the aggregation rule, and its result (the linear combination) is the aggregated forecaster. It does not attempt to model the evolutions of the concentrations but simply uses the models as black boxes and aims at performing better than the best of them. The aggregation rule does not rely on any stochastic assumption on the evolution of the simulated or observed concentrations. This paper is at a metalevel of prediction: it explains how to get improved performance given an ensemble of models whose preliminary construction is barely dealt with here. One may take any set of models one trusts. Stochastic or statistical modeling might also be used to construct one or several models; and the aggregated rule, since it usually improves on the models, will then benefit automatically from it.

[13] Compared to *Mallet and Sportisse* [2006a], one contribution of this paper lies in the significant improvements of the root mean square errors of the aggregated forecasts. Another key contribution is that the learning algorithms are theoretically grounded: explicit bounds on their practical performance can be exhibited.

2.2.1. Definition of a Sequential Aggregation Rule

[14] At each time index t, a linear sequential aggregation rule produces a weight vector $\mathbf{v}_t = (v_{1,t}, \ldots, v_{N,t}) \in \mathbb{R}^N$ based on the past observations $y^s_1, \ldots, y^s_{t-1}$ (for all $s \in \mathcal{N}$) and the past predictions $x^s_{m,1}, \ldots, x^s_{m,t-1}$ (for all $s \in \mathcal{N}$ and $m \in \mathcal{M}$). The final prediction at t is then obtained by linearly combining the predictions of the models according to the weights given by the components of the vector $\mathbf{v}_t$. More precisely, the aggregated prediction for station s at time index t equals

$$\widehat{y}^s_t = \mathbf{v}_t \cdot \mathbf{x}^s_t = \sum_{m=1}^N v_{m,t} x^s_{m,t}. \tag{1}$$

Convex sequential aggregation rules constrain the weight vector $\mathbf{v}_t$ to indicate a convex combination of N elements, which means that $\sum_{m=1}^N v_{m,t} = 1$ with $v_{m,t} \geq 0$. In this case, we use the notation $\mathbf{v}_t = \mathbf{p}_t$. When the weights are left unconstrained and can be possibly any vector of $\mathbb{R}^N$, they are denoted by $\mathbf{v}_t = \mathbf{u}_t$.

[15] Note that it is always possible to apply the aggregation rule per station. In this case, a weight vector $\mathbf{v}^s_t$ is produced for each station s.

2.2.2. Assessment of the Quality of a Sequential Aggregation Rule

[16] Not all stations are active at a given time index t. We denote by $\mathcal{N}_t \subset \mathcal{N}$ the set of active stations on the network $\mathcal{N}$ at time t. These are the stations s that monitored the ozone concentrations y_t^s. When we indicated above that aggregation rules rely on past predictions and past observations, we of course meant at the active stations. We can assess the quality of our strategies only on active stations. This is why the measure of performance, the well-known root mean square error (RMSE), of a rule $\mathcal{A}$ is defined as

$$RMSE(\mathcal{A}) = \sqrt{\frac{1}{\sum_{t=t_0}^{T} |\mathcal{N}_t|} \sum_{t=t_0}^{T} \sum_{s \in \mathcal{N}_t} \left(\mathbf{v}_t \cdot \mathbf{x}_t^s - y_t^s\right)^2} \qquad (2)$$

where t_0 is the first time index when the evaluation starts (hence $1 \leq t_0 < T$), and $|\mathcal{N}_t|$ is the cardinality (i.e., the number of elements) of the set $\mathcal{N}_t$. One may choose $t_0 > 1$ so that the time period $\{1, \ldots, t_0-1\}$ should serve as a short spin-up period for the aggregation rules. In the sequel, a model will be said to be the best for a given set of observations if it has the lowest RMSE.

2.2.3. Reference Performance Measures

[17] We indicate below some challenging performance measures, in terms of RMSE, beyond which it is impossible or difficult to go.

[18] The best performance is the following RMSE that no forecaster, even knowing the observations beforehand, can beat:

$$B_p = \sqrt{\frac{1}{\sum_{t=t_0}^{T} |\mathcal{N}_t|} \sum_{t=t_0}^{T} \min_{\mathbf{u}_t \in \mathbb{R}^N} \sum_{s \in \mathcal{N}_t} \left(\mathbf{u}_t \cdot \mathbf{x}_t^s - y_t^s\right)^2}. \qquad (3)$$

It should be seen as the potential of sequential aggregation.

[19] We introduce three other reference performance measures. The first one picks the best constant linear combination in $\mathbb{R}^N$ over the period $\{t_0, \ldots, T\}$:

$$B_{\mathbb{R}^N} = \min_{\mathbf{u} \in \mathbb{R}^N} \sqrt{\frac{1}{\sum_{t=t_0}^{T} |\mathcal{N}_t|} \sum_{t=t_0}^{T} \sum_{s \in \mathcal{N}_t} \left(\mathbf{u} \cdot \mathbf{x}_t^s - y_t^s\right)^2}. \qquad (4)$$

The second one corresponds to the best constant convex combination. We denote by $\mathcal{X}$ the set of all vectors $\mathbf{p}$ indicating convex combinations over N elements and define

$$B_{\mathcal{X}} = \min_{\mathbf{p} \in \mathcal{X}} \sqrt{\frac{1}{\sum_{t=t_0}^{T} |\mathcal{N}_t|} \sum_{t=t_0}^{T} \sum_{s \in \mathcal{N}_t} \left(\mathbf{p} \cdot \mathbf{x}_t^s - y_t^s\right)^2}. \qquad (5)$$

The third reference performance is that of the best model:

$$B_{\mathcal{M}} = \min_{m=1,\ldots,N} \sqrt{\frac{1}{\sum_{t=t_0}^{T} |\mathcal{N}_t|} \sum_{t=t_0}^{T} \sum_{s \in \mathcal{N}_t} \left(x_{m,t}^s - y_t^s\right)^2}. \qquad (6)$$

[20] Note that by definitions, in view of the inclusions between the sets on which minima are taken, it always holds that $B_p \leq B_{\mathbb{R}^N} \leq B_{\mathcal{X}} \leq B_{\mathcal{M}}$. Comparing to convex or even linear combinations of models is by far more challenging in terms of RMSE than simply competing with respect to the best model. This will be illustrated in section 4, devoted to numerical results.

2.2.4. Minimization of the RMSE via Minimization of the Regret

[21] Given a sequential aggregation rule $\mathcal{A}$, and the weight vectors $\mathbf{v}_t$ it chose at time indexes $t = 1, \ldots, T$, we want to compare its RMSE to one of the reference performance measures. To do so, we define $L_T(\mathcal{A})$ and $L_T(\mathbf{v})$ as the cumulative square errors (over all time indexes, and not only after t_0) of the rule $\mathcal{A}$ and of the constant linear combination $\mathbf{v}$,

$$L_T(\mathcal{A}) = \sum_{t=1}^{T} \sum_{s \in \mathcal{N}_t} \left(\mathbf{v}_t \cdot \mathbf{x}_t^s - y_t^s\right)^2, \qquad (7)$$

$$L_T(\mathbf{v}) = \sum_{t=1}^{T} \sum_{s \in \mathcal{N}_t} \left(\mathbf{v} \cdot \mathbf{x}_t^s - y_t^s\right)^2. \qquad (8)$$

The rules of interest, like the two discussed in sections 2.3 and 2.4, should ensure that the difference

$$R_T(\mathbf{v}) = L_T(\mathcal{A}) - L_T(\mathbf{v}) \qquad (9)$$

is small, i.e., $o(T)$, for all $\mathbf{v} \in W$. The comparison set W is $\mathbb{R}^N$ or $\mathcal{X}$, depending whether the rule is a linear aggregation rule or a convex aggregation rule.

[22] The difference $R_T(\mathbf{v})$ is termed the regret of the rule $\mathcal{A}$. The RMSE of $\mathcal{A}$ may be bounded in terms of the regret as

$$\left(RMSE(\mathcal{A})\right)^2 \leq \inf_{\mathbf{v} \in W} \left\{ \left(RMSE(\mathbf{v})\right)^2 + \frac{1}{\sum_{t=t_0}^{T} |\mathcal{N}_t|} R_T(\mathbf{v}) + \frac{1}{\sum_{t=t_0}^{T} |\mathcal{N}_t|} \sum_{t=1}^{t_0} \left(\mathbf{v} \cdot \mathbf{x}_t^s - y_t^s\right)^2 \right\}. \qquad (10)$$

This infimum is small: in the limit as $T \to \infty$, it equals $B_{\mathbb{R}^N}{}^2$, if $W = \mathbb{R}^N$, or $B_{\mathcal{X}}^2$, if $W = \mathcal{X}$. This is because the regret is sublinear, as is guaranteed by the learning techniques of interest: we refer to the bounds on the regret proposed below in equations (12) and (19) and to the comments that follow them. As a consequence, the averaged regret term $\frac{1}{\sum_{t=t_0}^{T} |\mathcal{N}_t|} R_T(\mathbf{v})$ tends to 0. And so does the third term of the right-hand side of equation (10), which is a constant divided by something of the order of T. Therefore, in total, the RMSE of the aggregation rule $\mathcal{A}$ tends to be at least as small as the RMSE of the best constant linear or convex combination.

[23] In other words, the machine learning algorithms of interest here guarantee that, in the long run, the overall performance of their aggregated forecasts is at least as good

as the performance of the best constant combination. This result holds whatever the sequence of observations and predictions may be, and without any stochastic assumption. This makes the methods of interest very efficient and very robust. We now describe two such methods.

2.3. A First Aggregated Forecaster: Ridge Regression

2.3.1. Statement

[24] The ridge regression forecaster $\mathcal{R}_\lambda$ is presented, for instance, by *Cesa-Bianchi and Lugosi* [2006, section 11.7]. It is parameterized by $\lambda \geq 0$ and chooses $\mathbf{u}_1 = (0, \ldots, 0)$; then, for $t \geq 2$,

$$\mathbf{u}_t = \underset{\mathbf{u}\in\mathbb{R}^N}{\operatorname{argmin}} \left[\lambda \|\mathbf{u}\|_2^2 + \sum_{t'=1}^{t-1} \sum_{s\in\mathcal{N}_{t'}} \left(\mathbf{u}\cdot\mathbf{x}_{t'}^s - y_{t'}^s\right)^2 \right]. \tag{11}$$

The interpretation is as follows. $\mathcal{R}_0$ is also called "follow-the-leader forecaster" (in a least squares regression sense) because it uses, at any time index $t \geq 2$, the linear combination that would have been the best over the past. $\mathcal{R}_\lambda$ is defined similarly, except that its definition includes a penalization factor $\lambda \|\mathbf{u}\|_2^2$ to keep the magnitude of the chosen $\mathbf{u}_t$ small and to have smoother variations from one $\mathbf{u}_t$ to the next $\mathbf{u}_{t+1}$.

2.3.2. Bound on the Regret

[25] An adaptation of the performance bound given by *Cesa-Bianchi and Lugosi* [2006, section 11.7] is presented by *Mallet et al.* [2007a, section 12]. For all $\lambda > 0$ and all $\mathbf{u} \in \mathbb{R}^N$,

$$R_T(\mathbf{u}) \leq \frac{\lambda}{2}\|\mathbf{u}\|_2^2 + SC^2 \sum_{i=1}^{N} \ln\left(1 + \frac{\mu_i}{\lambda}\right) \tag{12}$$

where S is the total number of monitoring stations,

$$C = \max_{t=1,\ldots,T} \max_{s\in\mathcal{N}_t} \left|\mathbf{u}_t\cdot\mathbf{x}_t^s - y_t^s\right| \tag{13}$$

and $\mu_1, \ldots, \mu_N$ are the eigenvalues of the matrix

$$\sum_{t=1}^{T} \sum_{s\in\mathcal{N}_t} \mathbf{x}_t^s \left(\mathbf{x}_t^s\right)^{\mathrm{T}} \left(= A_{T+1} - \lambda I_N \text{ with the notation below}\right). \tag{14}$$

Since each $\mathbf{u}_t$ depends in an awkward way on the parameter λ we cannot propose any theoretical optimal parameter to minimize the upper bound; we simply mention that it is obvious from the bound that λ should not be too large while also bounded away from 0; however, as noted by *Cesa-Bianchi and Lugosi* [2006, section 11.7], the typical order of magnitude of the bound is $O(\ln T)$ as the $\mu_i = O(ST)$.

2.3.3. Implementation

[26] Parameters: penalization factor $\lambda \geq 0$

[27] Initialization: $\mathbf{u}_1 = (0, \ldots, 0)$, $A_1 = \lambda I_N$

[28] For each time index $t = 1, 2, \ldots, T$,

[29] 1. predict with $\mathbf{u}_t$;

[30] 2. compute, with the predictions $\mathbf{x}_t^s$,

$$A_{t+1} = A_t + \sum_{s\in\mathcal{N}_t} \mathbf{x}_t^s \left(\mathbf{x}_t^s\right)^{\mathrm{T}}; \tag{15}$$

[31] 3. get the observations y_t^s, and compute

$$\mathbf{u}_{t+1} = \mathbf{u}_t - A_{t+1}^{-1} \left(\sum_{s\in\mathcal{N}_t} \left(\mathbf{u}_t\cdot\mathbf{x}_t^s - y_t^s\right)\mathbf{x}_t^s \right) \tag{16}$$

(where A_{t+1}^{-1} denotes the pseudo-inverse of A_{t+1} in case A_{t+1} is not invertible).

2.4. A Second Aggregated Forecaster: Exponentiated Gradient

2.4.1. Statement

[32] The basic version of the exponentiated gradient forecaster $\mathcal{E}_\eta$ is presented, for instance, by *Cesa-Bianchi* [1999]. It is parameterized by a learning rate $\eta > 0$ and chooses $\mathbf{p}_1 = (1/N, \ldots, 1/N)$; then, for $t \geq 2$, $\mathbf{p}_t$ is defined as

$$p_{m,t} = \frac{\exp\left(-\eta \sum_{t'=1}^{t-1} \widetilde{\ell}_{m,t'}\right)}{\sum_{j=1}^{N} \exp\left(-\eta \sum_{t'=1}^{t-1} \widetilde{\ell}_{j,t'}\right)} \tag{17}$$

for all $m = 1, \ldots, N$, where

$$\widetilde{\ell}_{m,t'} = \sum_{s\in\mathcal{N}_{t'}} 2\left(\mathbf{p}_{t'}\cdot\mathbf{x}_{t'}^s - y_{t'}^s\right) x_{m,t'}^s. \tag{18}$$

These $\mathbf{p}_t$ all define convex combinations.

2.4.2. Bound on the Regret

[33] An adaptation of the performance bound given by *Cesa-Bianchi* [1999] is presented by *Mallet et al.* [2007a, section 3]. Denoting by L a bound on the $|\widetilde{\ell}_{m,t'}|$ (for all m and t'), the regret of $\mathcal{E}_\eta$ against all convex combinations is uniformly bounded as

$$\sup_{\mathbf{p}\in\mathcal{X}} R_T(\mathbf{p}) \leq \frac{\ln N}{\eta} + \frac{T\eta}{2} L^2 = L\sqrt{2T\ln N} = O\left(\sqrt{T}\right), \tag{19}$$

where the last two equalities hold for the (theoretical) optimal choice $\eta^* = L^{-1}\sqrt{(2\ln N)/T}$.

2.5. Two Variants of the Previous Aggregated Forecasters, Obtained by Windowing or Discounting

2.5.1. Windowing

[34] Windowing relies on the following heuristic. Recent past indicates a trend on the current air quality but far away past is not very informative. Windowing is a way to only account for the most recent past. It formally consists in using the results of at most t_1 past dates to form the prediction. This will be referred to by a superscript $w(t_1)$.

[35] For instance, the windowing version of ridge regression $\mathcal{R}_\lambda^{w(t_1)}$ is the same as $\mathcal{R}_\lambda$ for times $t \leq t_1 + 1$, but for $t > t_1 + 1$ it uses $\mathbf{u}_t$ defined by

$$\mathbf{u}_t = \underset{\mathbf{u}\in\mathbb{R}^N}{\operatorname{argmin}} \left[\lambda \|\mathbf{u}\|_2^2 + \sum_{t'=t-t_1}^{t-1} \sum_{s\in\mathcal{N}_t'} \left(\mathbf{u}\cdot\mathbf{x}_{t'}^s - y_{t'}^s\right)^2 \right]; \tag{20}$$

it differs from (11) only by the starting index of the first summation [see *Mallet et al.*, 2007a, section 14].

[36] The windowing version of exponentiated gradient $\mathcal{E}_\eta^{w(t_1)}$ predicts as $\mathcal{E}_\eta$ for times $t \leq t_1 + 1$, but for $t > t_1 + 1$ it uses $\mathbf{p}_t$ defined by

$$p_{m,t} = \frac{\exp\left(-\eta \sum_{t'=t-t_1}^{t-1} \widetilde{\ell}_{m,t'}\right)}{\sum_{j=1}^{N} \exp\left(-\eta \sum_{t'=t-t_1}^{t-1} \widetilde{\ell}_{j,t'}\right)} \tag{21}$$

for all $m = 1, \ldots, N$, where $\widetilde{\ell}_{m,t'}$ is defined in equation (18). Again, the differences to (17) concern the indexes of the summations [see *Mallet et al.*, 2007a, section 5]. A drawback of this technique is that no nontrivial theoretical bound can be exhibited, because no sublinear bound on the regret can hold.

2.5.2. Discounting

[37] Discounting relies on the same heuristic. For the farthest away past to have a limited influence, the discrepancies with the observations are weighted according to their closeness to the present (the closer, the higher weight). The weights $\overline{\beta} = (\beta_t)_{t \geq 1}$ are given by a decreasing sequence of positive numbers, which will be referred to by a superscript $\overline{\beta}$.

[38] For instance, the $\overline{\beta}$-discounted version of ridge regression $\mathcal{R}_\lambda^{\overline{\beta}}$ chooses $\mathbf{u}_1 = (0, \ldots, 0)$ and for $t \geq 2$,

$$\mathbf{u}_t = \underset{\mathbf{u} \in \mathbb{R}^N}{\operatorname{argmin}} \left[\lambda \|\mathbf{u}\|_2^2 + \sum_{t'=1}^{t-1} (1 + \beta_{t-t'}) \sum_{s \in \mathcal{N}_{t'}} \left(\mathbf{u} \cdot \mathbf{x}_{t'}^s - y_{t'}^s\right)^2 \right] \tag{22}$$

[see *Mallet et al.*, 2007a, section 13]. In our experiments, we took $\beta_{t-t'} = 100/(t-t')^2$ for ridge regression.

[39] The $\overline{\beta}$-discounted version of exponentiated gradient $\mathcal{E}_\eta^{\overline{\beta}}$ chooses $\mathbf{p}_1 = (1/N, \ldots, 1/N)$ and for $t \geq 2$, $\mathbf{p}_t$ is defined as

$$p_{m,t} = \frac{\exp\left(-(\eta/\sqrt{t}) \sum_{t'=1}^{t-1} (1 + \beta_{t-t'}) \widetilde{\ell}_{m,t'}\right)}{\sum_{j=1}^{N} \exp\left(-(\eta/\sqrt{t}) \sum_{t'=1}^{t-1} (1 + \beta_{t-t'}) \widetilde{\ell}_{j,t'}\right)} \tag{23}$$

for all $m = 1, \ldots, N$, where $\widetilde{\ell}_{m,t'}$ is defined in equation (18). See *Mallet et al.* [2007a, section 6], which provides a theoretical performance bound for this forecaster under a suitable choice of $\overline{\beta}$. In our experiments, we took $\beta_{t-t'} = 1/(t-t')^2$ for the exponentiated gradient forecaster.

2.6. Remarks

2.6.1. Addition of Aggregated Forecasts in the Ensemble

[40] All forecasters presented in this paper (with the exception of their windowing versions) are competitive with respect to all possible constant convex or linear combinations of N base forecasting models. The parameter N enters merely in a logarithmic way in most of the performance bounds, so that, from a theoretical viewpoint, some more base forecasting models can be added at a negligible theoretical price, say N' of them.

[41] For instance, the learning methods do not make use of any stochastic assumption. But if one suspects that a stochastic modeling would be useful, then a model exploiting this can be added to the ensemble. Since it usually improves on each of the individual models, the aggregated forecaster, will then benefit from this modeling.

[42] The additional N' models can also be given by methods that are supported by the intuition and for which no precise theoretical guarantee can be exhibited. Running the learning algorithms then leads to an aggregated forecaster that exploits the intuitions that yielded the N' new models. At the same time, the theoretical performance guarantees still hold. They even improve since all convex or linear combinations over the first N models are combinations over the $N + N'$ models.

[43] To illustrate this, we tested at some point the addition of the aggregated forecasters $\mathcal{R}_0^{w(10)}$, $\mathcal{R}_0^{w(20)}$ and $\mathcal{R}_0^{w(30)}$ to the base 48 models. This forms the 51-model ensemble. The description of the three additional forecasters follows from a combination of sections 2.3 and 2.5.1: they predict with the weights of the best constant linear combination, in the least squares sense and over the learning window made of the 10, 20 or 30 previous dates. These three aggregated forecasters were already introduced by *Mallet and Sportisse* [2006a] under the notation ELS^{d} (which stands for ensemble least squares method per date). In the work of *Mallet et al.* [2007a], they are denoted by ELS^{10}, ELS^{20} and ELS^{30}.

2.6.2. Per Station and Hourly Predictions

[44] Any learning algorithm may be applied for predictions per station. In this case, each station s has its own sequence of weights $\mathbf{v}_t^s$ computed with the algorithm. The algorithm is independently applied to each station (on each singleton network $\mathcal{N} = \{s\}$). The sole difference is that when a station is unavailable on a given day, the learning step is skipped and the last available weights at the station are used for the next day. Some results are shown in Appendix A.

[45] In case of hourly forecasts, an algorithm is independently applied to each hour of the day. For instance, the algorithm predicts the weights for the forecast at 15:00 UT only on the basis of the observations and the predictions at 15:00 UT in the previous days. This strategy is expected to be more efficient because one uses information from forecasts in comparable situations (e.g., with respect to the emissions or to the height of the planetary boundary layer). It already exhibited better performance in the work of *Mallet and Sportisse* [2006a].

3. Experiment Setup

3.1. Ensemble Design

[46] We define a numerical model for ozone forecasting by (1) its physical formulation, (2) its numerical discretization, and, for convenience, (3) its input data.

[47] Eulerian chemistry-transport models all solve the same given reactive-transport equation, but they include different physical parameterizations to estimate the coefficients of the equation. Many alternative parameterizations are available to compute a coefficient or a set of related coefficients in the equation. For instance, a given model includes a limited number of chemical species whose reactions are described by a chemical mechanism. Several

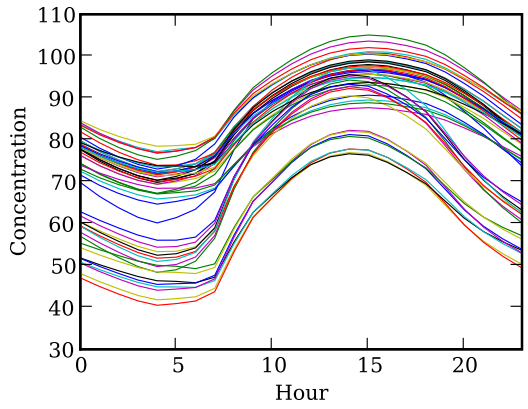

Figure 1. Ozone daily profiles of the 48 models. The concentrations are in μg m^{-3} and are averaged over Europe (at ground level) and over the 4 months of the simulation. The ensemble shows a wide spread, even on these strongly averaged hourly concentrations.

chemical mechanisms were developed for ozone [*Gery et al.*, 1989; *Carter*, 1990; *Stockwell et al.*, 1990, 1997] and incorporated in chemistry-transport models.

[48] Similarly, several numerical schemes were proposed, e.g., to handle the strong concentration gradients in the vicinity of emission sources, or to deal with the stiffness of gas phase chemistry [see, e.g., *Hundsdorfer and Verwer*, 2003]. Other numerical issues are addressed in the models, such as the distribution of the vertical layers, the space steps, and the time steps. Consequently, every model has its own numerical formulation.

[49] In addition, an air quality simulation involves a wide set of input data: land data, chemical data, meteorological fields, and emissions. There are alternative databases, and the data is often uncertain, with uncertainties often ranging from 30% to 100% [e.g., *Hanna et al.*, 1998]. Hence the modeler is doomed to make choices at that stage as well. Thereafter, for convenience, we consider that the selected input data (to a model) is actually part of the model. Thus a model is also defined by a set of physical data.

[50] In this paper, our ensemble encompasses most sources of uncertainties. We rely on models built with different physical parameterizations, numerical discretizations, and data sets. The same ensemble as given by *Mallet and Sportisse* [2006a] is used; see this paper for further details. A similar ensemble was deeply analyzed by *Mallet and Sportisse* [2006b]. The models are generated within the air quality modeling system Polyphemus [*Mallet et al.*, 2007b] which is flexible enough to build models that behave in drastically different manners; see Figure 1.

3.2. Simulations Over Europe

[51] The ensemble simulations cover 4 months, essentially in summer 2001, over western Europe. In this paragraph, we provide the main common characteristics of all simulations. The domain is [10.25°W, 22.25°E] $\times$ [40.25°N, 56.75°N]. The horizontal resolution is 0.5°. The meteorological fields are provided by the ECMWF (12-hour forecast cycles starting from analyzed fields). Raw emissions are retrieved in EMEP database and chemically processed according to *Middleton et al.* [1990]. Biogenic emissions are computed as in the work of *Simpson et al.* [1999]. The boundary conditions are computed by Mozart 2 [*Horowitz et al.*, 2003].

[52] Among the changes in the models, those related to the vertical diffusion parameterization [*Louis*, 1979; *Troen and Mahrt*, 1986] and to the chemical mechanism (RADM 2 [*Stockwell et al.*, 1990] and RACM [*Stockwell et al.*, 1997]) have the most prominent impacts on the ozone concentrations.

[53] The resulting ensemble contains 48 models or members. It shows a wide spread (Figure 1).

3.3. Individual Performance Measures of the Ensemble Members

[54] Although the ensemble shows a wide spread, all models bring useful information in the ensemble. For instance, at a given date, the ozone peaks forecast over Europe may vary strongly from one model to another, and many models turn out to be the best in some region. This is shown in Figure 2.

[55] The models are evaluated over the last 96 days of the simulation. The first 30 days are excluded in the evaluation because they are considered as a learning period for the ensemble algorithms. With the notation of section 2.2.2, the first forecast date with the ensemble methods is at $t_0 = 31$.

[56] As in the work of *Mallet and Sportisse* [2006a], three networks are used:

[57] 1. Network 1 is composed of 241 urban and regional stations, primarily in France and Germany (116 and 81 stations respectively). It provides about 619 000 hourly concentrations and 27 500 peaks.

[58] 2. Network 2 includes 85 EMEP stations (regional stations distributed over Europe), with about 240 000 hourly observations and 10 400 peaks.

[59] 3. Network 3 includes 356 urban and regional stations in France. It provides 997 000 hourly measurements

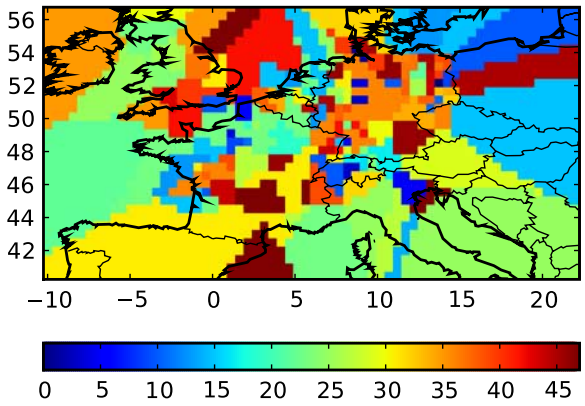

Figure 2. Map of best model indexes. In each cell of the domain, the color shows which model (marked with its index, in [0, 47]) gives the best ozone peak forecast on 7 May 2001 at the closest station to the cell center. Note that the most outer regions have almost no monitoring stations: the best model can barely be identified there. Despite all, there are enough stations to produce a highly fragmented map in which a significant number of models deliver the best forecast in some region. In addition, this map strongly changes from one day to the other.

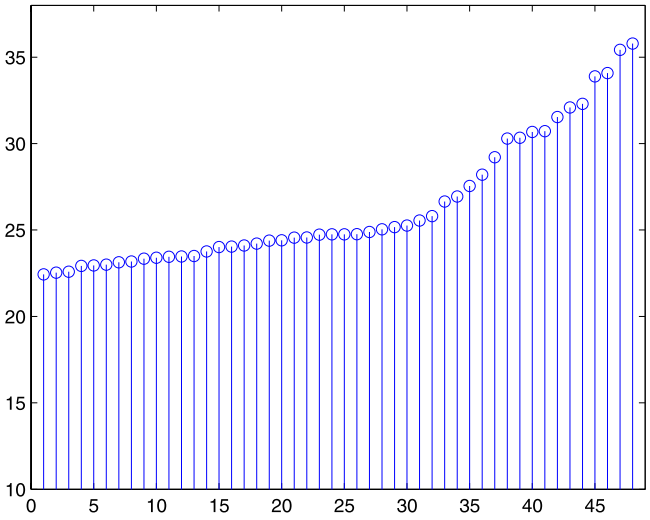

Figure 3. Root mean square errors (RMSE, in μg m^{-3}) for the 48 members in the ensemble. The error is computed with all ozone peak observations of network 1 during the last 96 days of the simulations. Here the models are sorted according to their RMSE.

and 42 000 peaks. Note that it includes most of the French stations of network 1.

[60] The study is chiefly carried out with network 1 (including calibration of the parameters). The performance of the learning methods on the two other networks is as good as on network 1, even though all parameters are determined from the optimization on network 1. See section 4.4.1 for more details on that and a summary of the results for networks 2 and 3.

[61] The performance of the individual models is first measured with the root mean square error (RMSE) over network 1 and is presented in Figure 3. The best model (see equation (6)) has a RMSE of $B_{\mathcal{M}} = 22.43$ μg m^{-3}. This is the reference performance to beat with the ensemble methods.

4. Results

[62] For convenience, the unit of the RMSEs (μg m^{-3}) is omitted in this section. Unless mentioned, the RMSEs are computed with the observations of network 1.

[63] As introduced in section 2.6.1, we consider two ensembles: the 48-model ensemble and a 51-model ensemble including $\mathcal{R}_0^{w(10)}$, $\mathcal{R}_0^{w(20)}$ and $\mathcal{R}_0^{w(30)}$ as additional models. These three aggregated models are added because of their good performance (20.78, 20.27, and 20.18, respectively) and because of their natural formulation, derived from a least squares minimization. In sections 4.2 and 4.3, all results are for daily peaks and on network 1.

Table 1. Reference Performance Measures of Section 2.2.3 Over the Last 96 Simulated Days[a]

Network	B_p	$B_{\mathbb{R}^N}$	$B_{\mathcal{X}}$	$B_{\mathcal{M}}$
		Daily Peaks		
Network 1	11.99	19.24	21.45	22.43
Network 2	8.47	18.16	20.91	21.90
Network 3	12.46	20.26	22.60	23.87
		Hourly Concentrations		
Network 1	14.88	22.80	24.81	26.68
Network 2	12.03	23.52	24.51	25.98
Network 3	15.32	23.19	26.28	28.45

[a]RMSE, μg m^{-3}.

Table 2. Performance (RMSE) of the Aggregated Forecasters for Ozone Daily Peaks Over the Last 96 Simulated Days on Network 1[a]

$B_{\mathbb{R}^N}$	$B_{\mathcal{X}}$	$B_{\mathcal{M}}$	$\mathcal{R}_0^{w(10)}$	$\mathcal{R}_0^{w(20)}$	$\mathcal{R}_0^{w(30)}$
19.24	21.45	22.43	20.78	20.27	20.18
$\mathcal{R}_{100}$	$\mathcal{E}_{2\times10^{-5}}$	$\mathcal{R}_{100}^{w(45)}$	$\mathcal{E}_{2\times10^{-5}}^{w(83)}$	$\mathcal{R}_{1000}^{\overline{\beta}}$	$\mathcal{E}_{1.2\times10^{-4}}^{\overline{\beta}}$
20.77	21.47	20.03	21.37	**19.45**	21.31

[a]The ensemble is made of the 48 base forecasting models.

4.1. Reference Performance Measures

[64] In order to assess the possible improvements of ensemble forecasts, we provide the reference performance measures of section 2.2.3 in Table 1. The most desirable performance to reach on network 1 and for daily peaks is $B_p \simeq 12$. It is certainly impossible to fulfill this objective because of the erratic time evolution of the associated weights [*Mallet and Sportisse*, 2006a], but this demonstrates the potential of combining the models forecasts.

[65] A more reasonable objective is to compete against the best constant linear combination, since several aggregated forecasters are guaranteed to have similar performance in the long run. The reference performance for daily peaks and on network 1 is as small as 19.24, which is a strong improvement compared to the best-model performance, $B_{\mathcal{M}} = 22.43$.

[66] More generally, Table 1 shows that the reference RMSEs of convex combinations are roughly 5% smaller (10% for hourly concentrations) than those of the best models, and that an additional 10% improvement is obtained by considering all linear combinations in $\mathbb{R}^N$.

4.2. Performance of Aggregated Forecasters

[67] The performance of several aggregated forecasters is shown in Table 2. Note that the parameters of the algorithms were tuned (offline) on network 1 to get the best performance [*Mallet et al.*, 2007a]. Section 4.4.1 will show that these tunings are robust and seem to be valid for a wide range of situations. On the basis of our experience, the methods are not too sensitive to the parameters, and optimal parameters (e.g., those proposed here) can be applied to different networks, different periods of time, different target concentrations and even different ensembles. Note also that the theoretical guarantees hold for any value of the parameters (e.g., any penalization factor $\lambda > 0$, in the ridge regression algorithm).

[68] Table 2 and the results of many other learning algorithms [*Mallet et al.*, 2007a] indicate that ridge regression-type aggregated forecasters show the best performance. Several algorithms get results less than or close to the best constant convex combination ($B_{\mathcal{X}} = 21.45$), but none except ridge regression may compete against the best constant linear combination ($B_{\mathbb{R}^N} = 19.24$). The use of the aggregated forecaster $\mathcal{R}_{1000}^{\overline{\beta}}$ results in a RMSE of 19.45 and thus, the theoretical guarantee of performing similarly, in the long

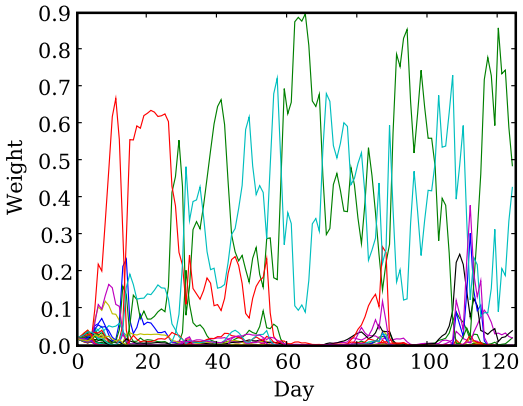

Figure 4. Weights associated by $\mathcal{E}_{2\times10^{-5}}$ to the 48 models against the time indexes (i.e., days). Because of the exponentiated-gradient formulation, the weights are positive and sum up to 1. Several models contribute to the linear combination, and the weight may vary quickly, even at the end of the time period.

run, to the best constant linear combination is essentially achieved.

[69] It is noteworthy that ridge regression is of a similar nature as the least squares methods. However, the algorithm benefits from the penalization factor driven by $\lambda > 0$; taking $\lambda = 0$ leads to worse performance. Including $\mathcal{R}_0^{w(10)}$, $\mathcal{R}_0^{w(20)}$, and $\mathcal{R}_0^{w(30)}$ in the ensemble leads to strong improvements in the performance of several algorithms; see the technical report [*Mallet et al.*, 2007a] for further details. This shows that the aggregated forecasters are generally driven by the best performance in the ensemble. Adding aggregated predictions to the ensemble, as was suggested by section 2.6.1, is consequently a meaningful and efficient strategy. Nevertheless, this does not apply to ridge regression with discount which has a slightly better RMSE with the base 48-model ensemble. In this case, its RMSE is about 3 μg m^{-3} lower than the best model, which is a strong improvement: in the work of *Mallet and Sportisse* [2006a], the decrease of the RMSE was about 2 μg m^{-3}, and without theoretical bounds on the method performance.

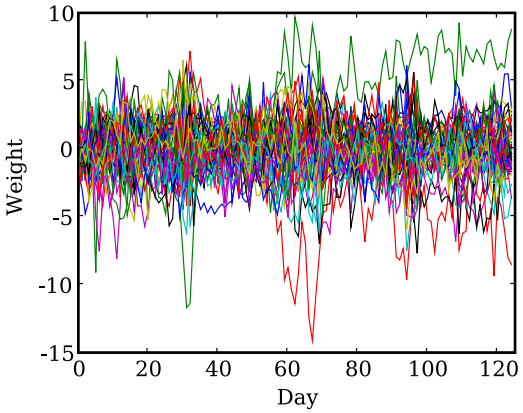

Figure 5. Weights associated by $\mathcal{R}_{1000}^{\overline{\beta}}$ to the 48 models against the time indexes (i.e., days). The forecaster attributes significant weights to a large set of models.

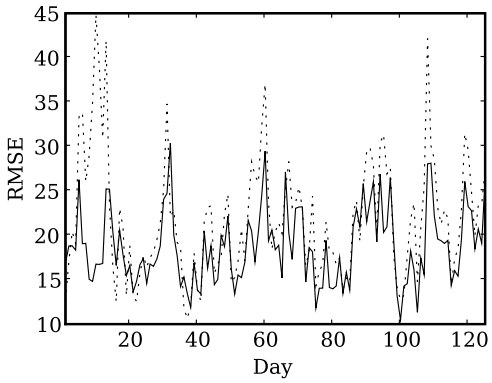

Figure 6. RMSE of $\mathcal{R}_{1000}^{\overline{\beta}}$ (continuous line) and of the best model (dotted line), against the day (or time index).

[70] A drawback of ridge regression might be its lack of constraints on the weights (and thus, the lack of interpretability of the obtained weights). In contrast, several algorithms produce convex weight vectors. Such weights may be applied far from the stations more safely than the unconstrained weights. The combined predictions will always fall in the envelop of the ensemble predictions, which avoids unrealistic concentrations. For instance, with the 48-model ensemble, the exponentiated-gradient aggregated forecaster performs as well as the best constant convex combination (21.47, while $B_{\mathcal{X}} = 21.45$) which is of course better than the best model ($B_{\mathcal{M}} = 22.43$). The weights computed by $\mathcal{E}_{2\times10^{-5}}$ are shown in Figure 4.

[71] In section 4.3, we focus on ridge regression with discount. For further analysis of all results, please refer to Appendix A or to *Mallet et al.* [2007a].

4.3. Ridge Regression With Discount

4.3.1. Further Results

[72] The weights computed by ridge regression with discount are shown in Figure 5. A key feature is the high number of contributing models: significant weights are associated to many models. This is a rather clear difference with most of the other aggregated forecasters (refer to *Mallet et al.* [2007a] in order to consult the time-evolution of the weights of all algorithms).

[73] The correlation (computed with all observations at once, in the last 96 days) between observations and simulated data is improved: the best model (with respect to the RMSE) has a correlation of 0.78 while the aggregated forecaster $\mathcal{R}_{1000}^{\overline{\beta}}$ predicts with a correlation of 0.84. The bias factor, defined as

$$\frac{1}{\sum_{t=t_0}^{T} |\mathcal{N}_t|} \sum_{t=t_0}^{T} \sum_{s\in\mathcal{N}_t} \frac{\mathbf{v}_t \cdot \mathbf{x}_t^s}{y_t^s}, \tag{24}$$

is 1.06 for the best model and 1.03 for $\mathcal{R}_{1000}^{\overline{\beta}}$.

4.3.2. Robustness

[74] The theory brings guarantees on the performance in the long run. This means that the ridge regression algorithm (with or without discount) is meant to be robust, which is an important feature in day-to-day operational forecasts. In addition to global performance, one concern is the perfor-

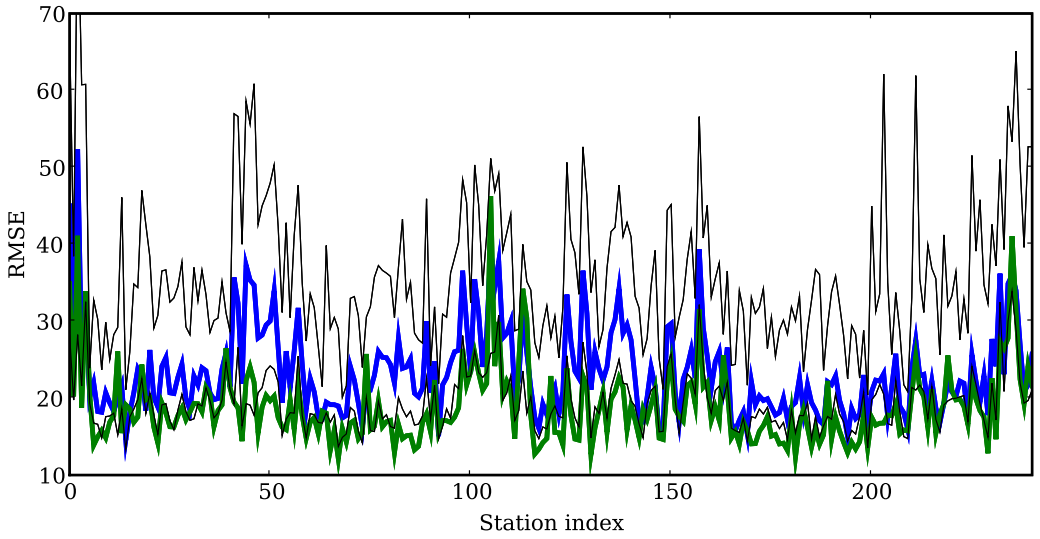

Figure 7. RMSE against the station index (for 241 stations). In green, $\overline{\mathcal{R}}^{\beta}_{1000}$; in blue, the best model (over all stations); in black, the best model and the worst model for the station.

mance at each single station and on each single day. We need robust aggregation methods that produce reasonable forecasts at all stations and on all days. We must avoid any method which would have a good overall performance but which would show disastrous performance at some stations or on some days.

[75] In practice, we noticed that the aggregated forecasters behave very well not only on average but also at most stations and on most days; $\overline{\mathcal{R}}^{\beta}_{1000}$ even predicts better than the (overall) best model for 73% of all observations. In Figure 6, the aggregated forecaster $\overline{\mathcal{R}}^{\beta}_{1000}$ shows better performance than this best model for most days (83% in the last 96 days). There is no day when it has an unreasonably high RMSE compared to the best model. Interestingly enough, the converse is not true: for some days, the RMSE of the best model is much higher than the one of $\overline{\mathcal{R}}^{\beta}_{1000}$. In Figure 7, $\overline{\mathcal{R}}^{\beta}_{1000}$ performs better than the (overall) best model at 223 stations (93% of the 241 stations). It performs better than the best model per station (with respect to the RMSE at the individual station) at 167 stations (70% of the stations). Its performance is always better than the worst model per station: a nontrivial fact, since the weights for the aggregation are unconstrained, as is illustrated, e.g., by Figure 5. One can clearly conclude from these tests that this learning method is robust.

4.3.3. Do Statistics Miss Extreme Events?

[76] A usual criticism against statistical methods is that they may be efficient on average but they miss the most important events. In daily air quality forecasts, the most important events to forecast are the highest pollution peaks.

[77] The methods we apply in this paper are not subject to the same limitations as purely statistical methods because they rely on the physical models. The weights are computed with learning methods, but the forecasts are still carried out by the models. In addition, the methods are responsive enough to allow quick variations of the weights (see Figures 4 and 5).

[78] To check that an aggregated model still has good skills in predicting extreme events, we count the number of improved forecasts (with respect to the best model) for the highest observed concentrations. We apply this to the discounted version of ridge regression, $\overline{\mathcal{R}}^{\beta}_{1000}$, whose predictions are denoted $\widehat{y}^s_t$. If the predictions of the overall best model are $x^s_{m,t}$ (and y^s_t are the corresponding observations), then we compute the score

$$F_{[o_i,o_{i+1}]} = \frac{|\{(t,s) \text{ such that } |\widehat{y}^s_t - y^s_t| < |x^s_{m,t} - y^s_t| \text{ and } y^s_t \in [o_i, o_{i+1}]\}|}{|\{(t,s) \text{ such that } y^s_t \in [o_i, o_{i+1}]\}|} \tag{25}$$

for a set of concentration intervals $[o_i, o_{i+1}]$ that cover all observed daily peaks. A perfect score is $F_{[\ldots]} = 1$ (all predictions of $\overline{\mathcal{R}}^{\beta}_{1000}$ better than the best model) and the worst score is $F_{[\ldots]} = 0$. Figure 8 plots this score. In this Figure 8, the number of observations considered to draw each bar varies of course with the bar; for instance, the

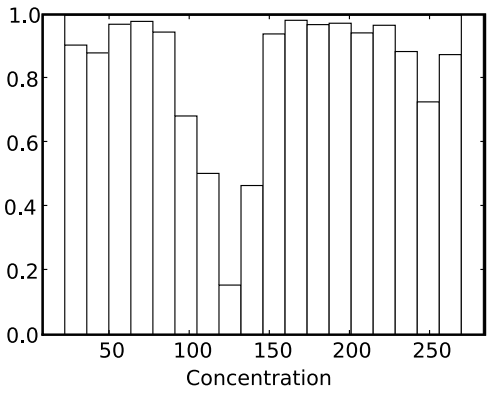

Figure 8. Frequency of improved forecast against the observed concentration, for $\overline{\mathcal{R}}^{\beta}_{1000}$. The frequency $F_{[o_i,o_{i+1}]}$ is computed as in equation (25). This illustrates that the discounted version of ridge regression is efficient on extreme events. (Remember from section 4.3.2 that $\overline{\mathcal{R}}^{\beta}_{1000}$ predicts better than the best model for 73% of all observations, that is, $F_{[0,\infty)} = 0.73$.)

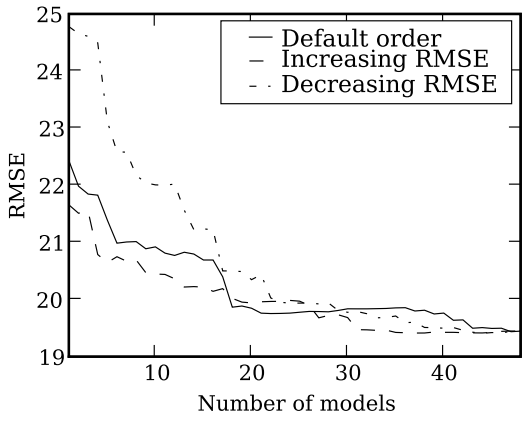

Figure 9. RMSE of $\mathcal{R}^{\overline{\beta}}_{1000}$ against the number of models in the ensemble. Three sets of ensembles with an increasing number of models are built. They follow the default order [*Mallet and Sportisse*, 2006a], the increasing RMSE order, and the decreasing RMSE order.

leftmost and rightmost bars are for extreme events, hence with few observations. Starting from the rightmost bar and moving to the left, the number of observations per bar is 4, 8, 11, 26, 60, and so on, and the histogram means that first, the four highest observations are better predicted with learning than with the best model; second, that seven of the eight following observations are better predicted; and so on. Similar results are obtained with the other learning methods. The conclusion is that the extreme events are well caught by our approach.

4.3.4. Influence of the Number of Models

[79] The 48 models are sorted as by *Mallet and Sportisse* [2006a]: herein referred to as the default order. Note this order has no link with the performance of the models. 48 ensembles are then built: the first ensemble only includes the first model, the second ensemble includes the first two models, the third ensemble includes the first three models, and so on. $\mathcal{R}^{\overline{\beta}}_{1000}$ is applied to each ensemble in order to test the influence of the number of models on the performance.

[80] Since this experiment depends on the inclusion order, two additional orders are tested: (1) from the best model (lowest RMSE) to the worst model (highest RMSE), and (2) from the worst model to the best model. The results are shown in Figure 9.

[81] The performance of $\mathcal{R}^{\overline{\beta}}_{1000}$ increases with the number of models. However, the improvement brought by one model rapidly decreases as the ensemble size increases. There is no obvious relation between the performance increase and the individual performance of the models.

[82] Note that $\mathcal{R}^{\overline{\beta}}_{1000}$ applied to the first model alone shows a lower RMSE than this first model (see Table 3), which indicates that this aggregated forecaster enjoys a property of automatic bias correction. Here, there is only one weight, associated with the single model, and it can indeed be interpreted as multiplicative bias correction factor.

4.4. Other Results

4.4.1. Application to Networks 2 and 3

[83] All results shown in the previous sections were presented for network 1. The two other networks are used for validation: in our study process, the algorithms were intensively analyzed and even tuned on network 1, and they were subsequently applied to networks 2 and 3 without specific optimization. This means that we directly use for them the parameters optimal for network 1. (The precise study of *Mallet et al.* [2007a] shows, by the way, that these offline optimal parameters for network 1 are usually larger than those prescribed by theory, leading to faster learning rates and a greater ability for the weight vectors to change when needed, as is illustrated, for instance, on Figure 4.)

[84] The methods achieve good performance on the two other networks; the results of Table 4 show in particular that the reference performance measures $B_{\mathcal{X}}$ and $B_{\mathbb{R}^N}$ are almost achieved again. All in all, this shows that the methods can be successfully applied to new data sets, demonstrating some kind of robustness.

[85] We do not provide satisfactory automatic tuning rules for the parameters yet. Theoretical optimal tuning rules for exponentiated gradient were proposed in the machine learning literature [*Mallet et al.*, 2007a, section 4] but they lead to lower performance. However, one may note that most of the proposed aggregation rules only depend on one or two learning parameters, to be set by the user. The results, though sensitive to tuning, are always better than the ones of the best model, even with the worst possible choices of the parameters. This is contrast to statistical methods, which require several parameters depending on underlying distributions to be estimated on data, or to other machine learning techniques, like neural networks, where the user has to choose the structure of the network and the values of several weights.

4.4.2. Results on Hourly Forecasts

[86] The methods are also used for hourly ozone forecasts; see section 2.6.2. Just like in the previous section, only the leading aggregated forecasters for network 1 and daily peaks are applied. Their parameters were tuned on network 1 and for daily peaks, but they perform well for hourly concentrations and on all networks. The scores are summarized in Table 5.

[87] The aggregated forecasters are more efficient for hourly concentrations than for the daily peaks: the improvements with respect to the best model are stronger (decrease of the RMSE over 4.5 μg m^{-3} on average). The reference performance measure $B_{\mathbb{R}^N}$ is beaten by $\mathcal{R}^{\overline{\beta}}_{1000}$ on the three networks. Since the physical models predict better the daily peaks than the hourly concentrations, one might state that the machine learning methods compensate for stronger physical limitations in the hourly predictions case.

5. Conclusion

[88] On the basis of ensemble simulations generated with the Polyphemus modeling system, machine learning algorithms prove to be efficient in forecasting ozone concen-

Table 3. Performance (RMSE) of $\mathcal{R}^{\overline{\beta}}_{1000}$ Applied to a Single Model[a]

Order	First Model RMSE	$\mathcal{R}^{\overline{\beta}}_{1000}$ RMSE
Default order	24.01	22.43
Increasing RMSE	22.43	21.66
Decreasing RMSE	35.79	24.78

[a]The first models of the ensembles defined in section 4.3.4.

Table 4. RMSE of the Leading Methods on the Three Networks[a]

	$\mathcal{E}_{2\times10^{-5}}$	$\mathcal{R}_{100}$	$\mathcal{R}_{1000}^{\overline{\beta}}$	$\mathcal{R}_{100}^{w(45)}$	$\mathcal{R}_0{}^{w(10)}$	$\mathcal{R}_0{}^{w(20)}$	$\mathcal{R}_0{}^{w(30)}$	$B_{\mathcal{M}}$
				48-Model Ensemble				
Network 1	21.47	20.77	**19.45**	20.03	20.78	20.27	20.18	22.43
Network 2	21.05	19.12	**18.12**	18.91	19.50	19.08	18.93	21.90
Network 3	24.12	21.92	20.88	21.10	21.95	21.31	21.20	23.87
	$\mathcal{E}_{3\times10^{-5}}$	$\mathcal{R}_{10^6}$	$\mathcal{R}_{1000}^{\overline{\beta}}$	$\mathcal{R}_{1000}^{w(45)}$				
				51-Model Ensemble				
Network 1	19.77	19.92	19.62	19.83				
Network 2	18.65	18.96	18.26	18.62				
Network 3	21.55	20.74	**20.70**	20.82				

[a]With the 48 base models only (upper part), with $\mathcal{R}_0^{w(10)}$, $\mathcal{R}_0^{w(20)}$, and $\mathcal{R}_0^{w(30)}$ in addition to the 48 base models (lower part).

trations by sequential aggregation of models. This conclusion applies to 1-day forecasts of ozone hourly concentrations and daily peaks. In short, the results improve the previous work [*Mallet and Sportisse*, 2006a] with respect to the performance and the mathematical framework. In this study, the learning algorithms always turn out to be better than the best model in the ensemble and they even compete with the best (constant and unconstrained) linear combination of models. Meanwhile the algorithms come with theoretical bounds that guarantee high-quality forecasts in the long run. In addition, our analysis clearly shows the robustness of the approach: good performance on new observation sets, no unreasonably high RMSE per station or per date compared to the best model, improvements almost in all regions, and for a majority of dates. It is noteworthy that the learning methods behave very well on the extreme events. Because of all these desirable properties, the methods are very relevant for operational forecasting.

[89] This study comes with several applicable algorithms for air pollution. Although they rely on well-known methods from machine learning, they were adapted, primarily to account for the presence of multiple monitoring stations and for past observations to have a relevance that decreases with time.

[90] Next developments may concentrate on the spatial validity of the aggregation. Furthermore, the sequential selection of (small subsets of) models in the ensemble may also help improve the results or at least decrease the computational load. The performance of the sequential aggregation should be compared with classical data assimilation (optimal interpolation, Kalman filtering, variational assimilation). For instance, while classical data assimilation often faces a limited impact in time, the learning methods could be more efficient, e.g., for 2-day forecasts.

Appendix A

[91] We provided in the main body of this paper a brief overview of the empirical analysis of two families of machine learning forecasting methods. Further details on the analysis, as well as a discussion of other methods, follow. All of them are included in the technical report [*Mallet et al.*, 2007a], which is available at http://www.dma.ens.fr/edition/publis/2007/resu0708.html.

A1. Per Station Predictions

[92] Any learning algorithm may be applied for predictions per station. In this case, each station s has its own sequence of weights $\mathbf{v}_t^s$ computed with the algorithm. The algorithm is independently applied to each station (on each singleton network $\mathcal{N} = \{s\}$). The sole difference is that when a station is unavailable on a given day, the learning step is skipped and the last available weights at that station are used for the next day. Also, the additional forecasts $\mathcal{R}_0^{w(10)}$, $\mathcal{R}_0^{w(20)}$, and $\mathcal{R}_0^{w(30)}$ cannot be computed and therefore added to the ensemble for per station predictions: the least squares problems associated with $\mathcal{R}_0^{w(10)}$, $\mathcal{R}_0^{w(20)}$ and $\mathcal{R}_0^{w(30)}$ are underdetermined.

[93] We kept the same learning parameters as those used in the main body of the paper. The performance is still computed with the global RMSE defined by (2). It is shown in Table A1 for daily peaks and in Table A2 for hourly concentrations, and should be compared to, respectively, Tables 4 and 5. Except for network 1 and daily predictions, the RMSEs usually improve when the aggregation rules are run station by station.

[94] We do not provide further details on the predictions per station as they are not the main objective of this paper; they are essentially provided for reference.

A2. Further Learning Methods

[95] We report briefly in this section the references and results of other learning methods. We only describe in detail the gradient descent forecasters because they showed promising results on networks and settings not reported here.

A2.1. Gradient Descent Forecasters

[96] The simplest version of gradient descent has already been implemented and studied by *Mallet and Sportisse* [2006a]. It is parameterized by $\eta > 0$ and is referred to

Table 5. RMSE of Leading Methods on the Three Networks for Hourly Forecasts

	$B_{\mathcal{M}}$	$\mathcal{E}_{2\times10^{-5}}$	$\mathcal{R}_{100}$	$\mathcal{R}_{1000}^{\overline{\beta}}$	$\mathcal{R}_{100}^{w(45)}$
Network 1	26.68	24.31	22.69	**22.02**	22.32
Network 2	25.98	24.67	23.53	**22.82**	23.29
Network 3	28.45	26.01	22.82	**22.27**	22.54

Table A1. RMSE of Leading Methods on the Three Networks for Daily Peaks and per Station Predictions

	$\mathcal{E}_{2\times10^{-5}}$	$\mathcal{R}_{100}$	$\mathcal{R}_{1000}^{\overline{\beta}}$	$\mathcal{R}_{100}^{w(45)}$
Network 1	23.26	19.72	19.73	**19.59**
Network 2	22.50	17.44	**17.43**	17.83
Network 3	24.42	19.73	19.72	**19.57**

Table A2. RMSE of Leading Methods for Hourly Forecasts and per Station Predictions

	$\mathcal{E}_{2\times10^{-5}}$	$\mathcal{R}_{100}$	$\mathcal{R}^{\overline{\beta}}_{1000}$	$\mathcal{R}^{w(45)}_{100}$
Network 1	23.16	**19.91**	19.98	20.04
Network 2	23.34	18.41	**18.37**	18.58
Network 3	24.43	**20.12**	20.30	20.20

below as $\mathcal{G}_\eta$. It starts with an arbitrary weight vector $\mathbf{u}_1$, we take for simplicity and for efficiency $\mathbf{u}_1 = (1/N, \ldots, 1/N)$. Then, for $t \geq 1$, the update is

$$\mathbf{u}_{t+1} = \mathbf{u}_t - \eta\widetilde{\ell}_t = \mathbf{u}_t - \eta\sum_{s\in\mathcal{N}_t} 2\left(\mathbf{u}_t\cdot\mathbf{x}_t^s - y_t^s\right)\mathbf{x}_t^s. \tag{A1}$$

This forecaster is competitive with respect to $B_{\mathbb{R}^N}$, its regret being bounded as follows. For all $\eta > 0$ and all $\mathbf{u} \in \mathbb{R}^N$,

$$R_T(\mathbf{u}) \leq \frac{\|\mathbf{u}-\mathbf{u}_1\|_2^2}{2\eta} + 2\eta N(SCB)^2 T = O\left(\sqrt{T}\right) \tag{A2}$$

with the notation C of section 2.3, the bound $B \geq |x_{t,m}^s|$ for all m, t, and s, and for a choice $\eta = O(1/\sqrt{T})$. See *Mallet et al.* [2007a, section 7] and *Cesa-Bianchi* [1999] for more details.

[97] The previous gradient descent forecaster is unconstrained. A variation with an additional projection step (e.g., on the set $\mathcal{X}$ of all convex combinations) was considered by *Zinkevich* [2003], leading to the forecaster referred to below as $\mathcal{Z}_\eta$ (with parameter $\eta > 0$). For $t \geq 1$, the update is

$$\mathbf{p}_{t+1} = P_{\mathcal{X}}\left(\mathbf{p}_t - \eta\widetilde{\ell}_t\right) \tag{A3}$$

$$= P_{\mathcal{X}}\left(\mathbf{p}_t - \eta\sum_{s\in\mathcal{N}_t} 2\left(\mathbf{p}_t\cdot\mathbf{x}_t^s - y_t^s\right)\mathbf{x}_t^s\right) \tag{A4}$$

where $P_{\mathcal{X}}$ denotes the projection onto the set $\mathcal{X}$ of convex combinations. This forecaster is competitive with respect to $B_{\mathcal{X}}$ since its regret is uniformly bounded as

$$\sup_{\mathbf{p}\in\mathcal{X}} R_T(\mathbf{p}) \leq \frac{1}{\eta} + 2\eta N(SCB)^2 T = O\left(\sqrt{T}\right) \tag{A5}$$

for a choice $\eta = O(1/\sqrt{T})$. See *Mallet et al.* [2007a, section 10] and *Zinkevich* [2003] for more details.

A2.2. Overview of Ten Other Forecasters

[98] We do not describe them in detail and we only present Table A3 to summarize all previously introduced forecasters and the new ones. We indicate also the reference performance to which each aggregated forecaster is guaranteed to be close, as well as the bound on the regret to be substituted into (10) to control the discrepancy between the reference performance and the performance of the aggregated forecaster. The reader interested in the mathematical details may take a look at our technical report [*Mallet et al.*, 2007a] to have more information. The column "Section" refers to a section of this report. Only the algorithms of the first half of the table were described above.

A2.3. Automatic Bias Correction

[99] This trick, introduced by *Kivinen and Warmuth* [1997], transforms an aggregated forecaster $\mathcal{A}$ proposing convex combinations $\mathbf{p}_t$ into an aggregated forecaster $\mathcal{A}'$ proposing linear combinations $\mathbf{u}_t$. More precisely, $\mathcal{A}'$ proposes predictions $\mathbf{u}_1, \mathbf{u}_2, \ldots$ in the ℓ_1 ball of a given radius $U > 0$ centered at the origin, which we denote by $\mathrm{B}_{\|\cdot\|_1}(0, U) \subset \mathbb{R}^N$. For the converted algorithm, we have a bound on the regret with respect to all elements of $\mathrm{B}_{\|\cdot\|_1}(0, U)$. The interest of this trick is to correct biases in an automatic way since the weights $\mathbf{u}_t$ do not necessarily sum up to 1. (For example, if all models propose predictions that are larger than the actual observation, the aggregated forecaster has a chance not to predict too high a value.)

[100] Formally, the conversion algorithm is as follows. We take $U > 0$, denote $\mathbf{z}_t^s = (U\mathbf{x}_t^s, -U\mathbf{x}_t^s)$, and let the original

Table A3. Overview of the Considered Aggregated Forecasters[a]

Section	Notation	Name	Reference	Theo. bound	Parameters	RMSE
12	$\mathcal{R}_\lambda$	Ridge regression	$\mathbb{R}^N$	$O(\ln T)$	$\mathcal{R}_{100}$	20.77
3	$\mathcal{E}_\eta$	Exponentiated gradient	$\mathcal{X}$	$O(\sqrt{T})$	$\mathcal{E}_{2\times10^{-5}}$	21.47
14	$\mathcal{R}_\lambda^{w(t_1)}$	Windowing ridge regression	$\mathbb{R}^N$	-	$\mathcal{R}^{w(45)}_{100}$	20.03
5	$\mathcal{E}_\eta^{w(t_1)}$	Windowing exponentiated gradient	$\mathcal{X}$	-	$\mathcal{E}^{w(83)}_{2\times10^{-5}}$	21.37
13	$\mathcal{R}_\lambda^{\overline{\beta}}$	Discounted ridge regression	$\mathbb{R}^N$	$o(T)$	$\mathcal{R}^{\overline{\beta}}_{1000}$	19.45
6	$\mathcal{E}_\eta^{\overline{\beta}}$	Discounted exponentiated gradient	$\mathcal{X}$	$o(T)$	$\mathcal{E}^{\overline{\beta}}_{1.2\times10^{-4}}$	21.31
10	$\mathcal{Z}_\eta$	Projected gradient descent	$\mathcal{X}$	$O(\sqrt{T})$	$\mathcal{Z}_{10^{-6}}$	21.28
7	$\mathcal{G}_\eta$	Gradient descent	$\mathbb{R}^N$	$O(\sqrt{T})$	$\mathcal{G}_{4.5\times10^{-9}}$	21.56
17	$\mathcal{F}'_{\alpha,\eta}$	Gradient fixed-share	$\mathcal{X}$	$O(\sqrt{T})$	$\mathcal{F}'_{0.02,\ 2.5\times10^{-5}}$	21.35
4	$\mathcal{E}_{a,b}$	Adaptive exponentiated gradient	$\mathcal{X}$	$O(\sqrt{T})$	$\mathcal{E}_{100,\ 1}$	21.48
9	$\mathcal{P}'_p$	Gradient polynomially weighted average	$\mathcal{X}$	$O(\sqrt{T})$	$\mathcal{P}'_{14}$	21.49
19	$\mathcal{O}_\beta$	Online Newton step	$\mathcal{X}$	$O(\ln T)$	$\mathcal{O}_{6\times10^{-7}}$	21.51
16	$\mathcal{F}_{\alpha,\eta}$	Fixed-share	$\mathcal{M}$	$O(\sqrt{T})$	$\mathcal{F}_{0.15,\ 5\times10^{-5}}$	21.89
1	$\mathcal{A}_\eta$	Exponentially weighted average	$\mathcal{M}$	$O(1)$	$\mathcal{A}_{3\times10^{-6}}$	22.46
15	$\mathcal{R}_\lambda^{\mathrm{vaw}}$	Nonlinear ridge regression	$\mathbb{R}^N$	$O(\ln T)$	$\mathcal{R}^{\mathrm{vaw}}_{106}$	22.90
2	$\mathcal{M}_\eta$	Mixture	$\mathcal{X}$	$O(\ln T)$	$\mathcal{M}_{10^{-3}}$	23.1
8	$\mathcal{P}_p$	Polynomially weighted average	$\mathcal{M}$	$O(\sqrt{T})$	$\mathcal{P}_{1.2}$	23.20
11	Π_η	Prod	$\mathcal{X}$	$O(\sqrt{T})$	$\Pi_{5.5\times10^{-7}}$	23.33

[a]We indicate their reference performance measures ($B_{\mathbb{R}^N}$, $B_{\mathcal{X}}$, or $B_{\mathcal{M}}$; see section 2.2.3), their theoretical regret bounds (if available), the section of the technical report where more information can be found, the parameters we used, and the results obtained for prediction of daily peaks on network 1.

aggregated forecaster $\mathcal{A}$ be fed with the $\mathbf{z}_t^s$ (it therefore behaves as if there were $2N$ models); it outputs predictions $\mathbf{q}_1$, $\mathbf{q}_2$, ... in the set of all convex combinations over $2N$ elements. The "extended" weights $\mathbf{u}_t$ of $\mathcal{A}'$ are defined as follows. For all $j = 1, \ldots, N$,

$$u_{j,t} = U\left(q_{j,t} - q_{j+N,t}\right); \tag{A6}$$

these $\mathbf{u}_t$ satisfy that for all stations s,

$$\mathbf{u}_t \cdot \mathbf{x}_t^s = \mathbf{q}_t \cdot \mathbf{z}_t^s. \tag{A7}$$

[101] Now, it is easy to see that all $\mathbf{u} \in \mathrm{B}_{\|\cdot\|_1}(0, U)$ may be represented in the following sense by a convex combination $\mathbf{q}$ of $2N$ elements,

$$\mathbf{q} \cdot \mathbf{z}_t^s = \mathbf{u} \cdot \mathbf{x}_t^s \tag{A8}$$

for all t and s. Equations (A7) and (A8) thus show that the regret of $\mathcal{A}'$ against the elements of $\mathrm{B}_{\|\cdot\|_1}(0, U)$ is smaller than about U times the regret of $\mathcal{A}$ against all convex combinations in $\mathcal{X}$.

[102] See *Mallet et al.* [2007a, section 20] for more details and more precise bounds. The experiments reported there show that $U = 0.99$ is sometimes an interesting value for some aggregated forecasters $\mathcal{A}$, in accordance to the automatic bias correction alluded at above. However, this procedure seldom performs better than the aggregated forecaster alone. We mention it here for the sake of completeness and we plan to further study this ability of automatic bias correction. We recall that another occurrence of an ability of automatic bias correction can be found at the end of section 4.3.4.

[103] **Acknowledgments.** The authors acknowledge the support of the French Agence Nationale de la Recherche (ANR), under grant ATLAS (JCJC06_137446) "From Applications to Theory in Learning and Adaptive Statistics". The first author is in the INRIA-ENPC joint project team CLIME, and in the ENPC-EDF R&D joint laboratory CEREA.

References

Beekmann, M., and C. Derognat (2003), Monte Carlo uncertainty analysis of a regional-scale transport chemistry model constrained by measurements from the atmospheric pollution over the Paris area (ESQUIF) campaign, *J. Geophys. Res.*, *108*(D17), 8559, doi:10.1029/2003JD003391.

Carter, W. P. L. (1990), A detailed mechanism for the gas-phase atmospheric reactions of organic compounds, *Atmos. Env., Part A*, *24*, 481–518.

Cesa-Bianchi, N. (1999), Analysis of two gradient-based algorithms for on-line regression, *J. Comput. Syst. Sci.*, *59*(3), 392–411.

Cesa-Bianchi, N., and G. Lugosi (2006), *Prediction, Learning, and Games*, 394 pp., Cambridge Univ. Press, New York.

Gery, M. W., G. Z. Whitten, J. P. Killus, and M. C. Dodge (1989), A photochemical kinetics mechanism for urban and regional scale computer modeling, *J. Geophys. Res.*, *94*, 12,925–12,956.

Hanna, S. R., J. C. Chang, and M. E. Fernau (1998), Monte Carlo estimates of uncertainties in predictions by a photochemical grid model (UAM-IV) due to uncertainties in input variables, *Atmos. Env.*, *32*, 3619–3628.

Horowitz, L. W., et al. (2003), A global simulation of tropospheric ozone and related tracers: Description and evaluation of MOZART, version 2, *J. Geophys. Res.*, *108*(D24), 4784, doi:10.1029/2002JD002853.

Hundsdorfer, W., and J. G. Verwer (2003), *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, *Ser. Comput. Math.*, vol. 33, 471 pp., Springer, New York.

Kivinen, J., and M. Warmuth (1997), Exponentiated gradient versus gradient descent for linear predictors, *Inf. Comput.*, *132*(1), 1–63.

Krishnamurti, T. N., C. M. Kishtawal, Z. Zhang, D. B. T. LaRow, and E. Williford (2000), Multimodel ensemble forecasts for weather and seasonal climate, *J. Clim.*, *13*, 4196–4216.

Lary, D. J., M. D. Müller, and H. Y. Mussa (2004), Using neural networks to describe tracer correlations, *Atmos. Chem. Phys.*, *4*, 143–146.

Louis, J.-F. (1979), A parametric model of vertical eddy fluxes in the atmosphere, *Boundary Layer Meteorol.*, *17*, 187–202.

Loyola, D. (2006), Applications of neural network methods to the processing of Earth observation satellite data, *Neural Networks*, *19*, 168–177.

Mallet, V., and B. Sportisse (2006a), Ensemble-based air quality forecasts: A multimodel approach applied to ozone, *J. Geophys. Res.*, *111*, D18302, doi:10.1029/2005JD006675.

Mallet, V., and B. Sportisse (2006b), Uncertainty in a chemistry-transport model due to physical parameterizations and numerical approximations: An ensemble approach applied to ozone modeling, *J. Geophys. Res.*, *111*, D01302, doi:10.1029/2005JD006149.

Mallet, V., B. Mauricette, and G. Stoltz (2007a), Description of sequential aggregation methods and their performances for ozone ensemble forecasting, *Tech. Rep. DMA-07-08*, École normale supérieure, Paris. (Available at http://www.dma.ens.fr/edition/publis/2007/resu0708.html)

Mallet, V., et al. (2007b), Technical note: The air quality modeling system Polyphemus, *Atmos. Chem. Phys.*, *7*, 5479–5487.

McKeen, S., et al. (2005), Assessment of an ensemble of seven real-time ozone forecasts over eastern North America during the summer of 2004, *J. Geophys. Res.*, *110*, D21307, doi:10.1029/2005JD005858.

Middleton, P., W. R. Stockwell, and W. P. L. Carter (1990), Aggregation and analysis of volatile organic compound emissions for regional modeling, *Atmos. Env., Part A*, *24*, 1107–1133.

Pagowski, M., et al. (2005), A simple method to improve ensemble-based ozone forecasts, *Geophys. Res. Lett.*, *32*, L07814, doi:10.1029/2004GL022305.

Pagowski, M., et al. (2006), Application of dynamic linear regression to improve the skill of ensemble-based deterministic ozone forecasts, *Atmos. Env.*, *40*, 3240–3250.

Simpson, D., et al. (1999), Inventorying emissions from nature in Europe, *J. Geophys. Res.*, *104*, 8113–8152.

Stockwell, W. R., P. Middleton, J. S. Chang, and X. Tang (1990), The second generation regional acid deposition model chemical mechanism for regional air quality modeling, *J. Geophys. Res.*, *95*, 16,343–16,367.

Stockwell, W. R., F. Kirchner, M. Kuhn, and S. Seefeld (1997), A new mechanism for regional atmospheric chemistry modeling, *J. Geophys. Res.*, *102*, 25,847–25,879.

Troen, I., and L. Mahrt (1986), A simple model of the atmospheric boundary layer: Sensitivity to surface evaporation, *Boundary Layer Meteorol.*, *37*, 129–148.

van Loon, M., et al. (2007), Evaluation of long-term ozone simulations from seven regional air quality models and their ensemble, *Atmos. Env.*, *41*, 2083–2097.

West, M., and J. Harrison (1997), *Bayesian Forecasting and Dynamic Models*, 2nd ed., 680 pp., Springer, New York.

Zinkevich, M. (2003), Online convex programming and generalized infinitesimal gradient ascent, in *Proceedings of the Twentieth International Conference on Machine Learning (ICML '03)*, edited by T. Fawcett and N. Mishra, pp. 928–936, AAAI Press, Menlo Park, Calif.

V. Mallet, INRIA, CLIME, BP 105, F-78153 Le Chesnay CEDEX, France. (vivien.mallet@inria.fr)

B. Mauricette and G. Stoltz, Département de Mathématiques et Applications, École Normale Supérieure, 45 rue d'Ulm, F-75005 Paris CEDEX, France. (gilles.stoltz@ens.fr)