Sequential classification of ozone peaks with ensemble methods

Karim DRIFI, Vivien MALLET, Gilles STOLTZ

INRIA Rocquencourt – École normale supérieure de Paris – École normale supérieure de Cachan

Outline

Mathematical framework

Machine learning : Predicting individual sequences with expert advice.

- Predict a sequence of observation $y_1, y_2, \ldots$.
- At each time step t we have M expert advice $\mathbf{x}_t = (x_{1,t}, x_{2,t}, \ldots x_{M,t})$.
- Make a prediction $\hat{y}_t$ based on past results of each expert
- Build sequentially a weight vector $\mathbf{w}_t$ and combine expert advice :

$$\hat{y}_t = \mathbf{w}_t \cdot \mathbf{x}_t = \sum_{i=1}^{M} w_{i,t} x_{i,t} \tag{1}$$

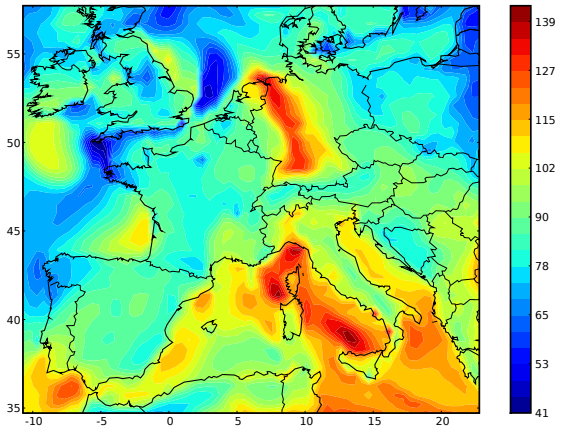
- Reference book : Prediction, Learning, and Games (Nicolò Cesa-Bianchi and Gábor Lugosi).
- No stochastic assumption.

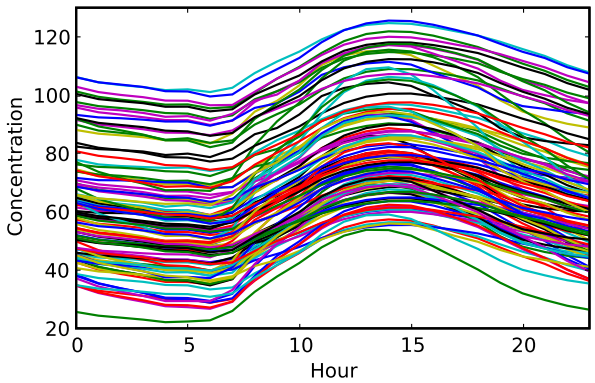
Air quality

System of Partial differential equation :

$$\forall i \qquad \frac{\partial c_i}{\partial t} = \underbrace{-\mathrm{div}(\mathbf{V}c_i)}_{\text{advection}} + \underbrace{\mathrm{div}\left(\rho \mathbf{K} \nabla \frac{c_i}{\rho}\right)}_{\text{diffusion}} + \underbrace{\chi_i(\mathbf{c})}_{\text{chemistry}} + S_i - P_i \quad (2)$$

- Solving the PDE system using Polyphemus software.
- Many possible parameterizations.
- We use as experts members of an ensemble of 107 simulations.





Aim

We focus on a particular task which is to predict whether or not the ozone concentration will exceed a given threshold.

High ozone concentration causes health damages.
Above some threshold the governement is compelled to inform population.

Notations

- Time steps $t = 1, 2, \ldots, T$
- Locations index $s \in= \{1, \ldots, S\}$
- Simulations/Experts index $m = 1, \ldots, M$
- Threshold $\Upsilon = 150\,\mu\mathrm{g/m}^3$
- Simulated/observed/predicted concentrations $x_{m,t}^s$, y_t^s et $\hat{y}_t^s$
- Binary versions of data (± 1) : $c_{m,t}^s = \mathrm{sgn}(x_{m,t}^s - \Upsilon)$, $d_t^s = \mathrm{sgn}(y_t^s - \Upsilon)$ et $\hat{d}_t^s = \mathrm{sgn}(\hat{y}_t^s - \Upsilon)$

Criterion

		Observed	
		yes	no
Forecast	yes	hit	false alarm
	no	miss	correct negative

Classical evaluation criterion :

$$\text{Classification rate} = \frac{\text{hit} + \text{correct neg.}}{\text{hit} + \text{correct neg.} + \text{miss} + \text{false al.}} \qquad (3)$$

Few exceedances are observed so that a forecaster predicting no exceedances get a classification rate of 95.0% ! ! !
We need to use another criterion to evaluate performance of a forecaster.

A relevant criterion

Empirical consideration led us to use three criteria to evaluate forecasters :

$$\text{Hit rate} = \frac{\text{hit}}{\text{hit} + \text{miss}} \tag{4}$$

$$\text{Correct negative ratio} = \frac{\text{correct neg.}}{\text{correct neg.} + \text{false al.}} \tag{5}$$

$$\text{Threat score} = \frac{\text{hit}}{\text{hit} + \text{miss} + \text{false al.}} \tag{6}$$

The most relevant is the Threat score.

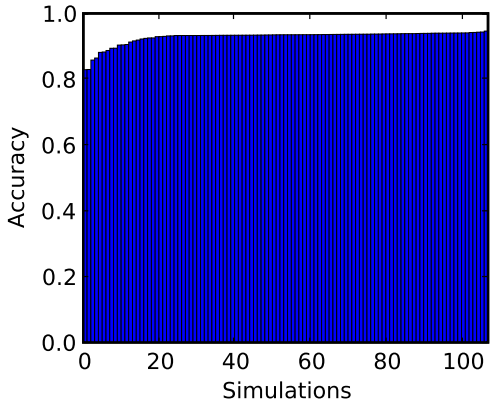


FIGURE: Classification rate

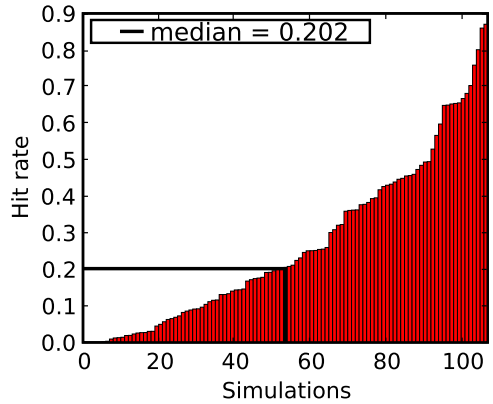


FIGURE: Hit rate

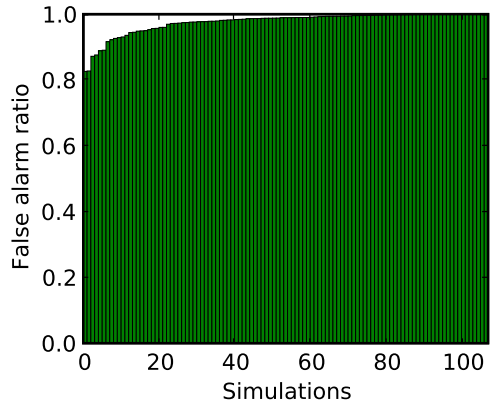


FIGURE: Correct negative ratio

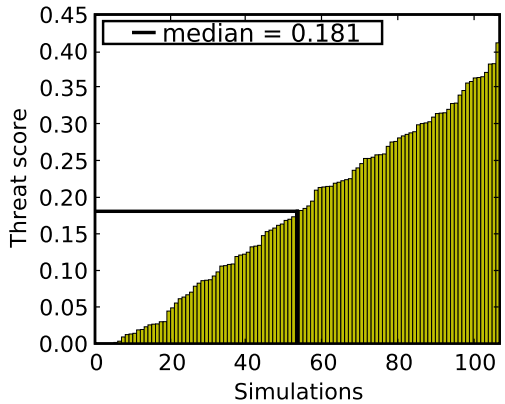


FIGURE: Threat score

Different aggregating methods

- Concentration aggregation : $x_{m,t}^s$

$$\hat{d}_t^s = \text{sgn}\left[\sum_m \left(w_{m,t} x_{m,t}^s\right) - \Upsilon\right] \tag{7}$$

- Discrepancies between concentration and threshold : $x_{m,t}^s - \Upsilon$

$$\hat{d}_t^s = \text{sgn}\left[\sum_m w_{m,t}\left(x_{m,t}^s - \Upsilon\right)\right] \tag{8}$$

- Binary version of data : $\text{sgn}(x_{m,t}^s - \Upsilon)$

$$\hat{d}_t^s = \text{sgn}\left[\sum_m w_{m,t}\text{sgn}\left(x_{m,t}^s - \Upsilon\right)\right] \tag{9}$$

Oracle

Some oracles have been computed in order to get reference scores to compete with.

- Draw many random vectors, aggregate simulations (independant of the time) and keep the one which obtains the best threat score.
- Vectors can either be on the simplex or hypercube.
- Aggregation of simulation has been performed on discrepancies to the threshold and on binary data.

The best threat score has been obtained aggregating discrepancies to threshold without simplex constraint.

Discounted Ridge and Lasso

First we tested sequential Ridge (eq. 10) and Lasso (eq. 11) regression. These are computed this way :

$$\mathbf{u}_t = \operatorname{argmin}_{\mathbf{u}\in\mathbb{R}^N} \left[\lambda \|\mathbf{u}\|_2^2 + \sum_{t'=1}^{t-1} \beta_{t-t'} \sum_{s\in\mathbf{N}} \left(\mathbf{u}\cdot\mathbf{x}_{t'}^s - y_{t'}^s\right)^2 \right] \quad (10)$$

$$\mathbf{u}_t \in \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^N} \left[\lambda \|\mathbf{u}\|_1 + \sum_{t'=1}^{t-1} \beta_{t-t'} \sum_{s\in\mathbf{N}} \left(\mathbf{u}\cdot\mathbf{x}_{t'}^s - y_{t'}^s\right)^2 \right] \quad (11)$$

Using a time discount : $\beta_t = 1 + \alpha/t^\beta$.

« Hinge loss » and conservative forecasters

« Hinge loss » :

$$\ell_\gamma(p,y) = (\gamma - py)_+, \quad \mathbf{L}_{\gamma,T} = \sum_{t=1}^{T}\sum_{s}\ell_\gamma \tag{12}$$

For a given potential function ϕ :

Parameter : Learning rate $\lambda > 0$

Initialization : $\mathbf{w}_0 = \nabla\Phi(0)$.

At each time step $t = 1, 2, \ldots, T$

(1) Set $\widehat{p}_t = \mathbf{w}_{t-1} \cdot \mathbf{x}_t$, and predict $\widehat{y}_t = \mathrm{sgn}(\widehat{p}_t)$;

(2) Weights update

$$\mathbf{w}_t = \nabla\Phi\left[\nabla\Phi^*(\mathbf{w}_{t-1}) + \lambda\sum_{s\in\mathbf{N}}\left(1_{y_t^s \neq \widehat{y}_t} y_t^s \mathbf{x}_t^s\right)\right].$$

p-Perceptron, generalization bound

Using $\phi_p(\mathbf{u}) = \frac{1}{2}\|\mathbf{u}\|_p^2 = \frac{1}{2}\left(\sum_{m=1}^{M} |u_m|^p\right)^{\frac{2}{p}}$ the forecaster is called **p-Perceptron** and the generalization bound is :

$$\sum_{t=1}^{T}\sum_{s} 1_{\widehat{y_t}^s \neq y_t^s} \leqslant \frac{\mathbf{L}_{\gamma,T}(\mathbf{u})}{\gamma} + \|\mathbf{u}\|_q^2 \frac{p-1}{\gamma^2}\sum_{s} X_p^{s2} + \sqrt{(p-1)\frac{\|\mathbf{u}\|_q^2}{\gamma^2}\frac{\mathbf{L}_{\gamma,T}(\mathbf{u})}{\gamma}\sum_{s} X_p^{s2}} \quad (13)$$

with $q = \frac{p}{p-1}$ and $X_p^s = \max_t \|\mathbf{x}_t^s\|_p$. This inequality holds for all $\mathbf{u}$ and γ.

p-Perceptron, generalization bound

In a more readable form :

$$\frac{1}{T}\left[\sum_{t=1}^{T}\sum_{s}1_{\widehat{y_t}^s\neq y_t^s}-\frac{\mathbf{L}_{\gamma,T}(\mathbf{u})}{\gamma}\right] \leqslant \frac{K_1}{T}\sqrt{\frac{\mathbf{L}_{\gamma,T}(\mathbf{u})}{\gamma}}+\frac{K_2}{T}$$
$$= O(\frac{1}{\sqrt{T}}) \qquad (14)$$

which is a classical generalization bound of machine learning algorithms (K_1 and K_2 functions of $(\mathbf{u},p,X_p^s)$ and $\frac{\mathbf{L}_{\gamma,T}(\mathbf{u})}{\gamma} \quad \alpha \quad T$).

Winnow, generalization bound

Using $\phi(\mathbf{u}) = \sum_i e^{u_i}$ the forecaster is called **Winnow** and the generalization bound is :

$$\sum_{t=1}^{T} \sum_{s} 1_{\widehat{y_t}^s \neq y_t^s} \leqslant \frac{1}{1-\varepsilon} \frac{\mathbf{L}_{\gamma,T}(\mathbf{u})}{\gamma} + \frac{\sum_s {X_\infty^s}^2}{\gamma^2} \frac{\ln M}{2\varepsilon(1-\varepsilon)} \tag{15}$$

With $X_\infty^s = \max_t \|\mathbf{x}_t^s\|_\infty$.

Winnow, generalization bound

Considering that $\frac{1}{1-\varepsilon} \approx (1+\varepsilon)$, We minimize the bound with $\varepsilon \quad \alpha \quad \frac{1}{\sqrt{T}}$. Thus we get :

$$\frac{1}{T}\left[\sum_{t=1}^{T}\sum_{s} 1_{\widehat{y_t}^s \neq y_t^s} - \frac{\mathbf{L}_{\gamma,T}(\mathbf{u})}{\gamma}\right] \leqslant O(\frac{1}{\sqrt{T}}) \tag{16}$$

Truth selection

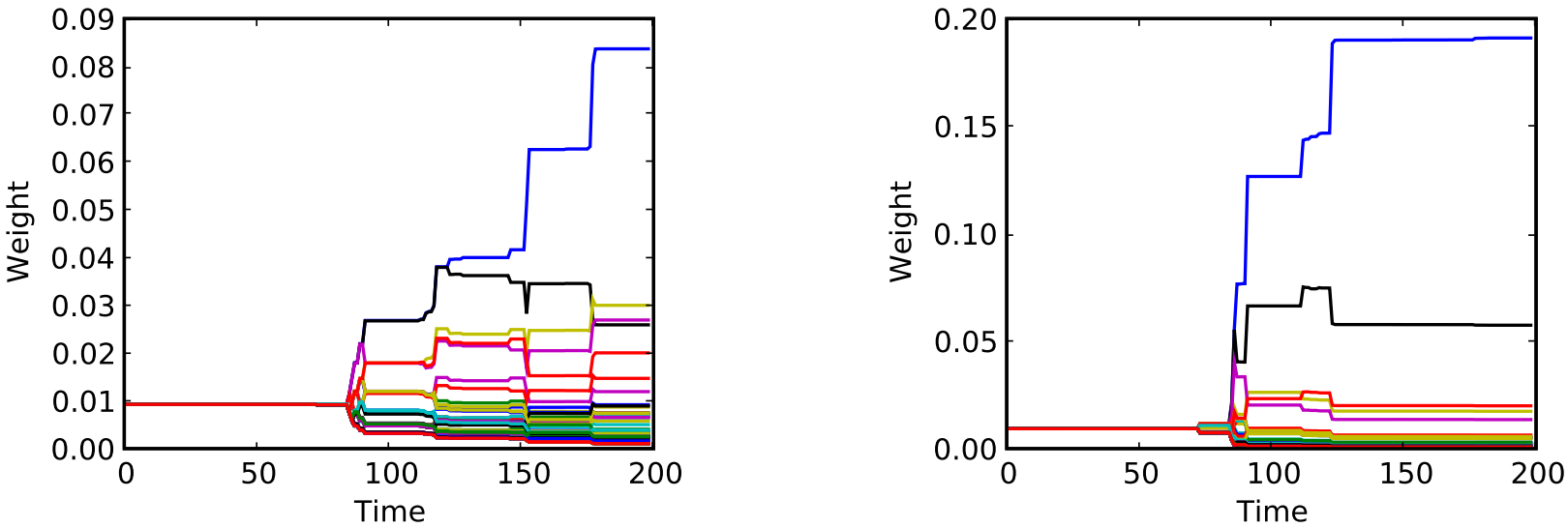


FIGURE: Weights evolution — Winnow

Results, 107 simulations ensemble

	Hit rate	Correct neg. ratio	Threat score
Ref. model	9.1	99.6	8.7
Best model	68.1	95.3	41.1
Oracle	64.5	96.0	41.4
Ridge	55.8	97.8	42.6
Lasso	55.6	97.9	43.2
p-Perceptron	54.3	95.9	34.5
Winnow	52.7	97.5	39.1

INERIS ensemble — 7 simulations

Model	Hit rate	Correct neg. ratio	Threat score
1	70.7	95.0	28.22
2	72.7	94.1	25.95
3	73.0	94.2	26.41
4	73.6	93.4	24.44
5	6.0	94.1	5.76
6	25.1	99.9	19.94
7	27.6	99.1	17.51
Ridge	29.0	98.1	25.3
Lasso	28.1	99.5	24.7
Winnow	53.1	97.6	30.1

TABLE: Results INERIS

Conclusions

- A difficult task...
- Results are good considering the expert performance but not enough to provide reliable prediction.

107 simulations ensemble :

- Discounted Lasso get the best performance
- Lasso forecast beat each models

INERIS ensemble, Winnow gets the best performance.

Possible improvements

- Predict a probability of exceedance and compare to various values
- Exponentiated gradient (EG), Exponentiated Weighted Average (EWA)...
- Allow to use other loss functions, distinguish error cases

EWA use a convex loss function $\ell : X \longrightarrow [0, 1]$ and update weights according to :

$$w_{m,t} = \frac{w_{m,t-1} e^{-\eta \ell(x_{m,t}, y_t)}}{\sum_{j=1}^{M} w_{j,t-1} e^{-\eta \ell(x_{j,t}, y_t)}} \tag{17}$$

Generalization bound :

$$\sum_{t=1}^{T} \ell\left(\hat{p}_t, y_t\right) - \min_m \left[\sum_{t=1}^{T} \ell\left(x_{m,t}, y_t\right)\right] \leqslant \frac{lnM}{\eta} + \frac{T\eta}{2} \tag{18}$$