Prévision de la consommation électrique par agrégation séquentielle de prédicteurs spécialisés

Pierre Gaillard

EDF – École Normale Supérieure

7 septembre 2011

Plan

I. Introduction et présentation du problème

Introduction

Objectif : Prévision de la consommation électrique à court terme.

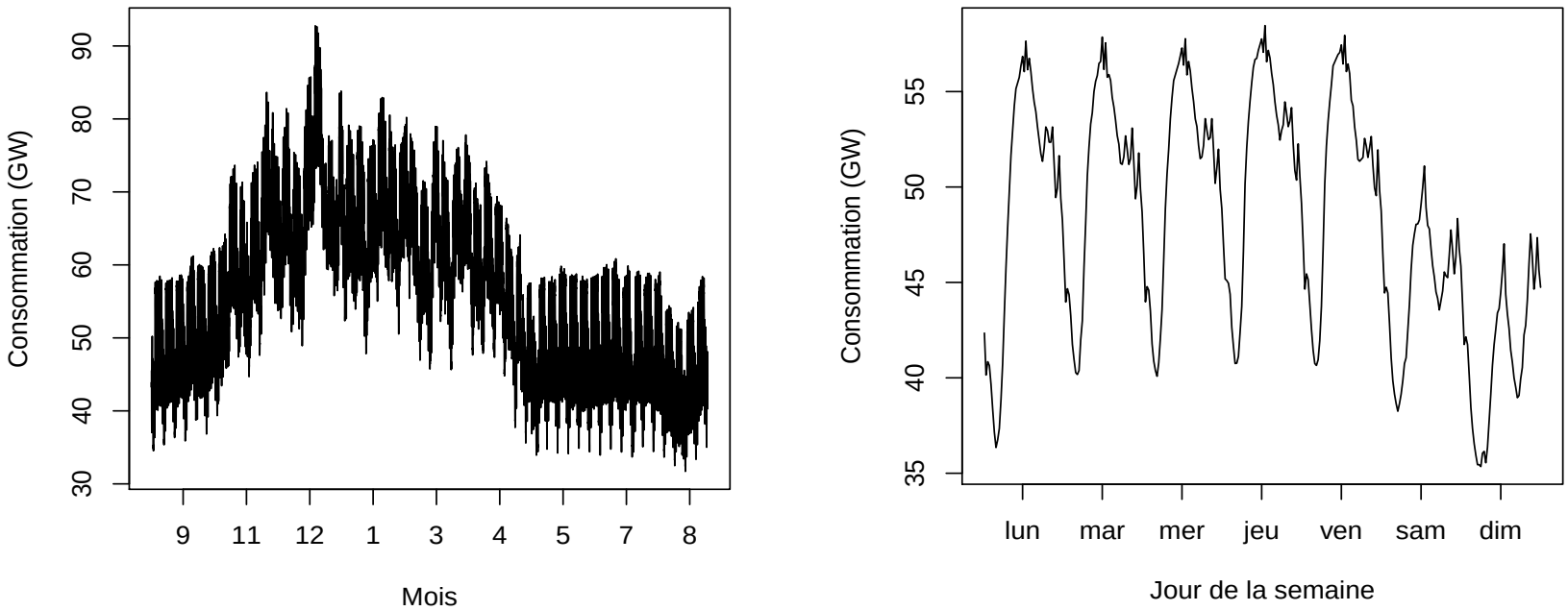


FIGURE: La consommation électrique en France au cours d'une année et d'une semaine.

Qu'est-ce que la consommation électrique ?

- C'est un processus temporel
- Dépend de nombreuses variables contextuelles :
  - météo : température, nébulosité, vent, ...
  - calendaires : type de jour, position dans l'année, ...
  - temporelles : consommation de la veille, ...

Les experts

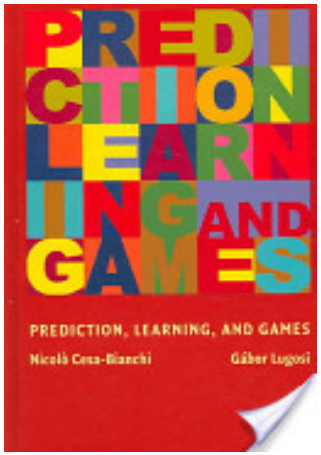
Plusieurs modèles de prévision existent déjà (merci au groupe R39), ils ont permis de créer 24 experts :

- paramétriques : Eventail
- semi-paramétriques : modèle GAM
- non paramétrique : modèle fonctionnel de similarités

Ils sont spécialisés et nous proposent chaque jour des prévisions.

À qui faire confiance ?

II. La théorie des suites individuelles



Introduction aux suites individuelles

- On a une suite arbitraire à prédire : $y_1, y_2, \ldots, y_T$
- On dispose d'un ensemble $E = \{1, \ldots, N\}$ d'experts.
- À chaque instant t,
  - certains experts sont actifs ($E_t \subset E$) et proposent une prévision f_{it}
  - Un algorithme de mélange attribue des pondérations à chaque experts $\mathbf{p}_t = (p_{1t}, \ldots, p_{Nt}) \in \mathbb{R}^N$ et prédit

$$\widehat{y}_t = \sum_{i \in E_t} p_{it} f_{it} \; ;$$

  - y_t est révélée

Estimation de la qualité d'une séquence de prévisions

Perte de la prévision $\widehat{y}_t$: $\ell(\widehat{y}_t, y_t)$.
Perte du vecteur de mélange $\mathbf{p}$ à l'instant t :

$$\ell_t(\mathbf{p}) = \ell\left(\sum_{j \in E_t} p_j f_{jt},\ y_t\right)$$

Objectif : trouver des méthodes de mélange $\mathcal{A}$ qui subissent une faible erreur moyenne,

$$\overline{\mathrm{ERR}}_T(\mathcal{A}) = \frac{1}{T}\sum_{t=1}^{T} \ell_t(\mathbf{p}_t)\,.$$

Quelques fonctions de perte considérées

- La perte carrée est définie pour tout $x, y \in \mathbb{R}_+$ par

$$\ell(x, y) = (x - y)^2 .$$

Dans ce cas, plutôt que l'erreur moyenne $\overline{\mathrm{ERR}}_T(\mathcal{A})$ on utilisera la racine de l'erreur moyenne de la perte carrée

$$\mathrm{RMSE}_T(\mathcal{A}) = \sqrt{\overline{\mathrm{ERR}}_T(\mathcal{A})} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\widehat{y}_t - y)^2} .$$

- La perte absolue est définie pour tout $x, y \in \mathbb{R}_+$ par

$$\ell(x, y) = |x - y| .$$

- Le pourcentage d'erreur absolue est défini pour tout $x, y \in \mathbb{R}_+$ par

$$\ell(x, y) = \frac{|x - y|}{y} .$$

Les performances de référence

Les oracles :

- Meilleur expert fixé ;
- Meilleure combinaison linéaire fixée. Elle correspond à l'utilisation d'un même vecteur de mélange fixé $\mathbf{u} \in \mathbb{R}^N$, renormalisé à chaque instant ;
- Meilleure combinaison convexe fixée ;
- Meilleur expert composé.

Une méthode de mélange :

- Mélange uniforme.

III. Algorithmes de mélange et extensions

Trois premières familles

- Exponentially weighted average (EWA) : pondère les experts de manière exponentielle en fonction de la performance passée.
- Algorithme specialist : semblable à EWA avec une pondération différente quand les experts ont été désactivés ou activés.
- Algorithme fixed-share : suit dans une première étape la même idée que EWA avant de redistribuer les poids aux experts actifs en s'assurant que chacun a un poids minimal.

Trois premières familles d'algorithmes de mélange

- Les algorithmes précédents sont détaillés dans l'article de M. Devaine, Y. Goude et G. Stoltz (2011).
- Ils se présentent chacun dans deux versions :
  - une basique

$$\overline{\mathrm{ERR}}_T(\mathcal{A}) \leqslant \overline{\mathrm{ERR}}_T(\text{meilleur expert}) + K_1\sqrt{\frac{\ln N}{T}}$$

  - une de type descente gradient

$$\overline{\mathrm{ERR}}_T(\mathcal{A}^{\mathrm{grad}}) \leqslant \overline{\mathrm{ERR}}_T(\text{meilleur combinaison convexe}) + K_2\sqrt{\frac{\ln N}{T}}$$

Exponentially weighted average

Entrée: paramètre d'apprentissage $\eta > 0$
Initialisation: w_1 est le vecteur convexe uniforme, $w_{i1} = 1/N$ pour $i = 1, \ldots, N$

pour les instants t de 1 à T faire
  prédire $\widehat{y}_t = \frac{1}{\sum_{i \in E_t} w_{it}} \sum_{j \in E_t} w_{jt} f_{jt}$
  observer y_t
  pour les experts i de 1 à T mettre à jour

$$w_{it+1} = \begin{cases} w_{it}\, e^{\eta\left(\ell(\widehat{y}_t, y_t) - \ell(f_{it}, y_t)\right)} & \text{si } i \in E_t \\ w_{it} & \text{si } i \notin E_t \end{cases}$$

  fin pour
fin pour

Suivre le leader pénalisé (Ridge)

- Motivé par la définition de l'oracle linéaire.
- A chaque instant t, choisi le vecteur de poids $\mathbf{u}_t \in \mathbb{R}^N$ vérifiant un compromis entre performance et régularité :

$$\mathbf{u}_{t+1} \leftarrow \underset{\mathbf{u}\in\mathbb{R}^d}{\arg\min} \underbrace{\sum_{s=2}^{t} \ell_s\left(\frac{\mathbf{u}}{\tau_s(\mathbf{u})}\right) \left|\tau_s(\mathbf{u})\right|}_{\text{Performance passée}} + \underbrace{\lambda \ \|u\|_2^2}_{\text{Régularité}}$$

où $\tau_t(\mathbf{u}) = \dfrac{\sum_{j\in E_t} u_j}{\sum_{k=1}^{N} u_k}$.

- Borne théorique ?

Les forêts aléatoires comme méthode de mélange stochastique

Méthode de mélange stochastique prenant en compte les variables contextuelles (météo, calendaires, temporelles)

Plus de précision partie 4.

Adaptation opérationnelle

Problème pratique : retour d'information sur la consommation réelle qu'une fois par jour (tous les h instants).

Objectif : Prédire simultanément les consommations des h prochains instants (la prochaine journée).

Méthode : ne faire évoluer dans les poids attribués à chaque expert, que ce qui ne dépend pas de leur perte et de la consommation réelle.

Les bornes théoriques sont conservées !

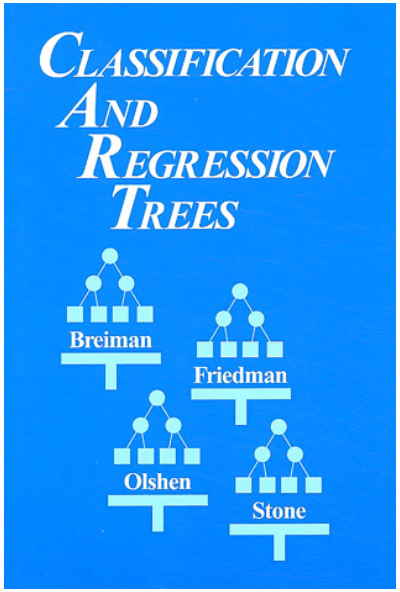
Calibration automatique des paramètres

Objectif : adapter automatiquement et en ligne les paramètres des algorithmes précédents

Méthode : considérer une grille de paramètres potentiels que l'on construit au fur et à mesure :

- `Initialiser` : $\Lambda =$ valeur théorique optimale du paramètre
- À chaque instant t
  - Choisir dans Λ le paramètre donnant pour l'instant la meilleure performance
  - Si il est sur un bord, agrandir Λ de façon exponentielle

IV. Les forêts aléatoires

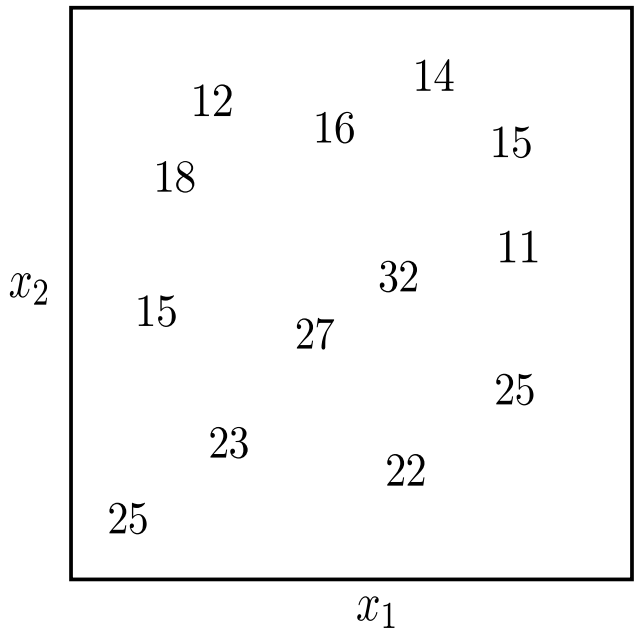


Historique et références

- Méthode introduite par Leo Breiman en 2001.
- Idées plus anciennes : Bagging (1996), arbres de décisions CART (1984)
- Implémentée en `R` avec le paquet `randomForest`
- Un site web utile : `http://www.stat.berkeley.edu/ breiman/RandomForests`
- Preuves de convergences récentes (2006,2008)

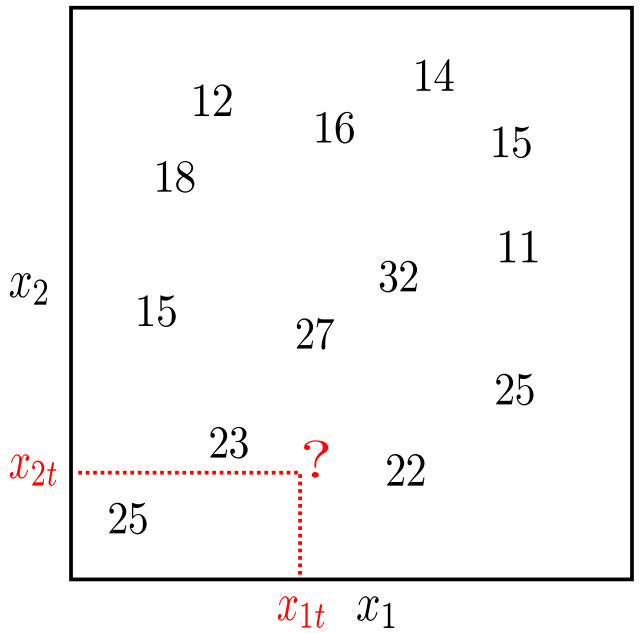
Cadre théorique

- Ensemble d'entrainement $(X_t, Y_t)_{t \in S_0}$ iid
- **Objectif :** expliquer Y_t en fonction des variables contextuelles X_t.



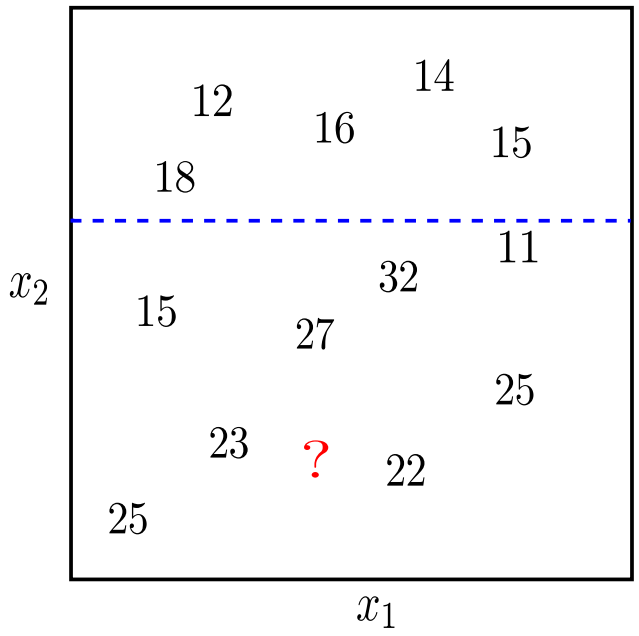
Exemple d'un arbre de décision

- Ensemble d'entrainement $(X_t, Y_t)_{t \in S_0}$ iid
- **Objectif :** expliquer Y_t en fonction des variables contextuelles X_t.



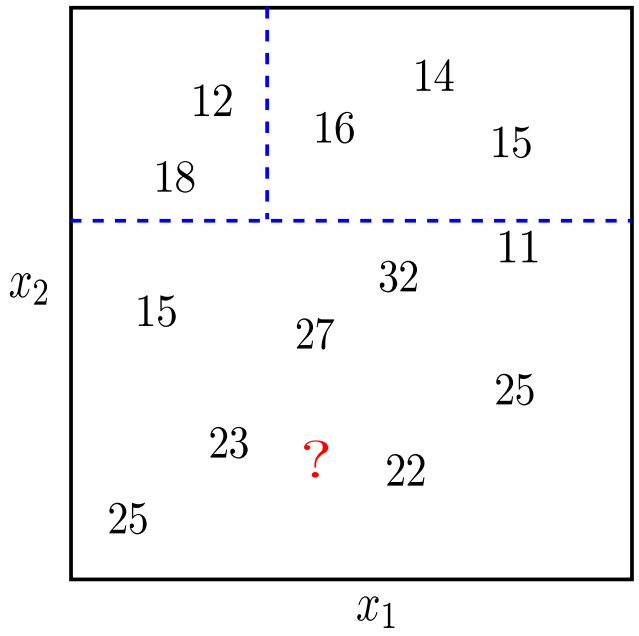
Exemple d'un arbre de décision

- Ensemble d'entrainement $(X_t, Y_t)_{t \in S_0}$ iid
- **Objectif :** expliquer Y_t en fonction des variables contextuelles X_t.



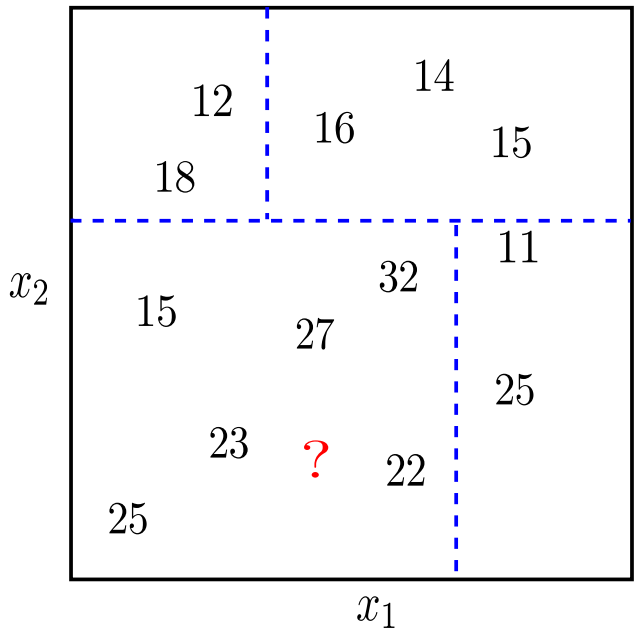
Exemple d'un arbre de décision

- Ensemble d'entrainement $(X_t, Y_t)_{t \in S_0}$ iid
- **Objectif :** expliquer Y_t en fonction des variables contextuelles X_t.



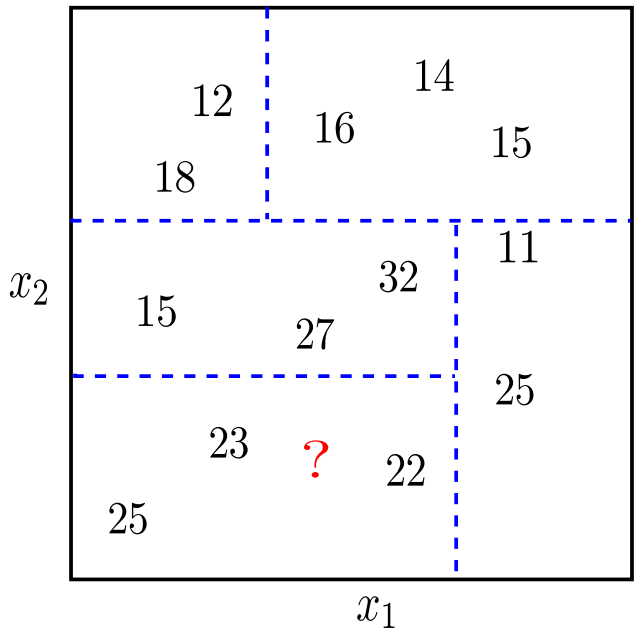
Exemple d'un arbre de décision

- Ensemble d'entrainement $(X_t, Y_t)_{t \in S_0}$ iid
- **Objectif :** expliquer Y_t en fonction des variables contextuelles X_t.



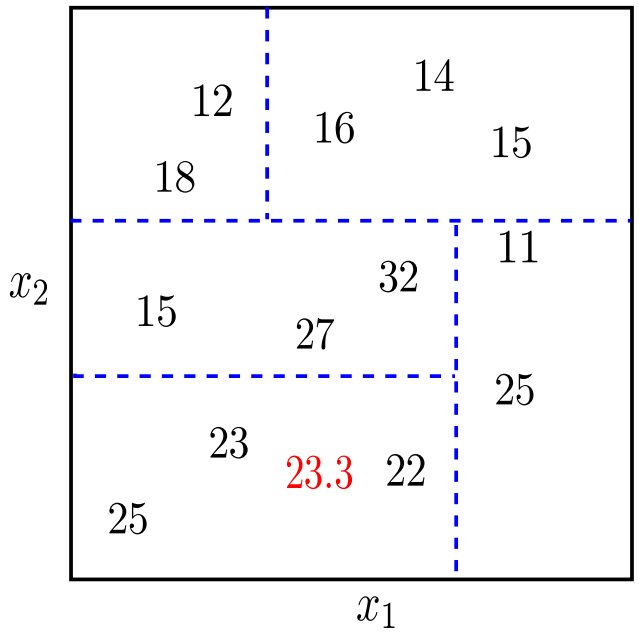
Exemple d'un arbre de décision

- Ensemble d'entrainement $(X_t, Y_t)_{t \in S_0}$ iid
- **Objectif :** expliquer Y_t en fonction des variables contextuelles X_t.



Exemple d'un arbre de décision

- Ensemble d'entrainement $(X_t, Y_t)_{t \in S_0}$ iid
- **Objectif :** expliquer Y_t en fonction des variables contextuelles X_t.

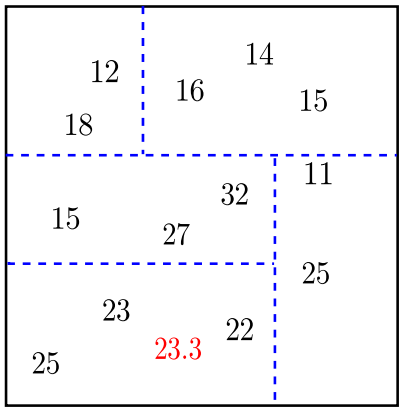


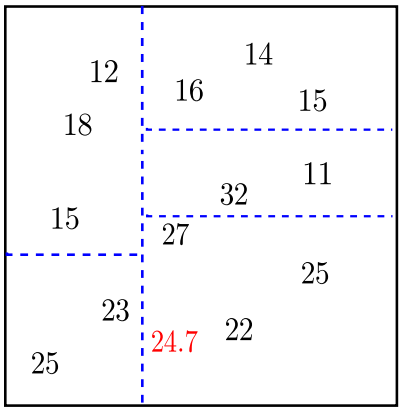
Forêts aléatoires

Les forêts aléatoires consistent à faire tourner en parallèle un grand nombre (~ 400) d'arbres de décisions construits aléatoirement, avant de les moyenner.

Si les arbres sont « décorrélés », cela permet de réduire la variance des prévisions.

$$\bar{\sigma}^2 = \rho\sigma^2 + \frac{1-\rho}{K}\sigma^2 .$$





prédit $\frac{24.7+23.3}{2} = 24$

Comment utiliser les forêts aléatoires pour du mélange ?

À chaque instant t,

- Utiliser l'information passée comme ensemble d'entrainement :
  - la consommation passée $Y_1, \ldots, Y_{t-1}$
  - les variables contextuelles $X_1, \ldots, X_{t-1}$
  - les prévisions des experts f_{is} pour $1 \leqslant i \leqslant N$ et $1 \leqslant s \leqslant t-1$
- Construire les forêts aléatoires
- Observer les variables contextuelles X_t (météo, calendaires, temporelles)
- Estimer pour chaque expert j la perte qu'il va subir $\widehat{\ell}_{jt}$ par les forêts aléatoires
- Proposer un vecteur de mélange

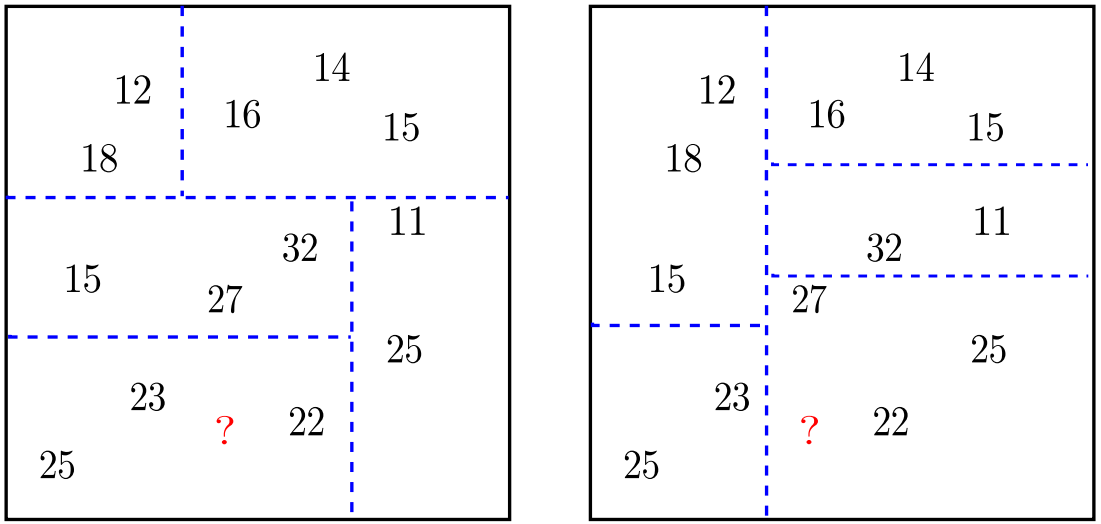
Avantages et propriétés de l'algorithme

- Avantage : crée un mélange prenant en compte l'information contextuelle
- Performance théorique se rapproche de celle du meilleur expert
- Inconvénients :
  - Long temps de calcul
  - S'adapte difficilement à de nouvelle situations

$\rightarrow$ améliorations grâce à la notion de proximité

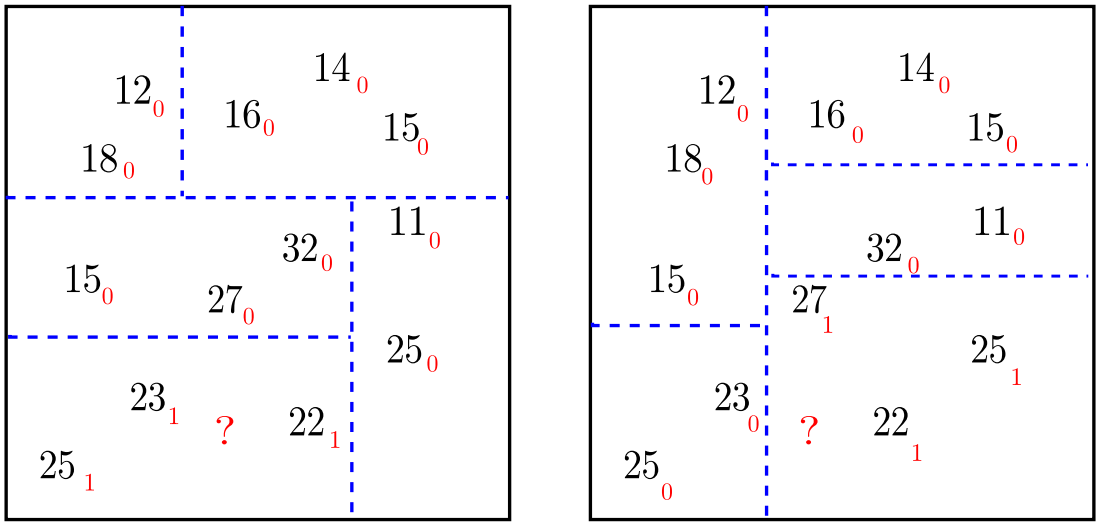
Une notion importante : la proximité

Intuition : Tomber souvent dans les même feuilles des arbres $\rightarrow$ expliquer la sortie Y de façon similaire.



Une notion importante : la proximité

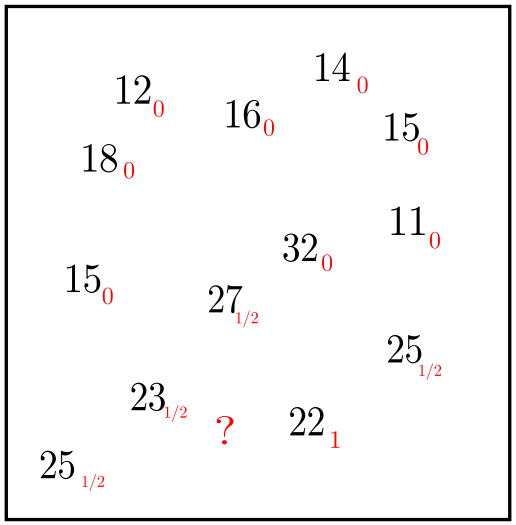
Intuition : Tomber souvent dans les même feuilles des arbres $\rightarrow$ expliquer la sortie Y de façon similaire.



$$\text{prox}(X_t, X_s) = \frac{1}{K} \sum_{k=1}^{K} \mathbb{1} \left\{ \begin{array}{c} X_t \text{ et } X_s \text{ tombent dans la} \\ \text{même feuille dans l'arbre } k \end{array} \right\}.$$

Une notion importante : la proximité

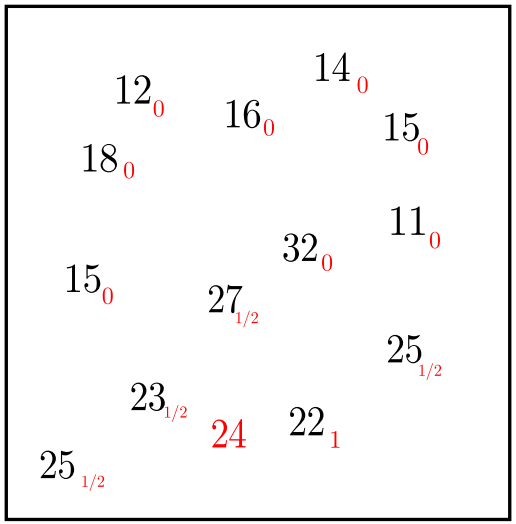
Intuition : Tomber souvent dans les même feuilles des arbres $\rightarrow$ expliquer la sortie Y de façon similaire.



$$\text{prox}(X_t, X_s) = \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}\left\{ \begin{array}{c} X_t \text{ et } X_s \text{ tombent dans la} \\ \text{même feuille dans l'arbre } k \end{array} \right\}.$$

Une notion importante : la proximité

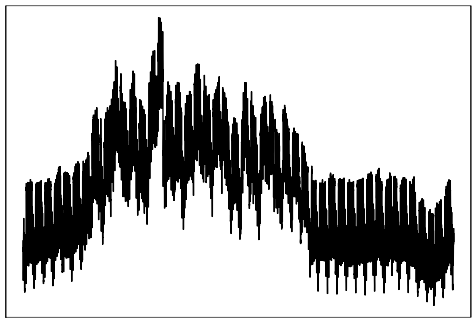
Intuition : Tomber souvent dans les même feuilles des arbres $\rightarrow$ expliquer la sortie Y de façon similaire.



On prédit ensuite par exemple par

$$\arg\min_{a\in\mathbb{R}} \sum_{X_s\in E_t} \text{prox}(X_t, X_s)\,(Y_s - a)^2 .$$

V. Expériences dans le cadre de la consommation électrique



Le jeu de données

Nombre de jours D	320
Intervalles de temps	30 minutes
Nombre d'instants T	15 360 $(= 320 \times 48)$
Nombre d'experts N	24 $(= 15 + 8 + 1)$
Unité	GW
Médiane des y_t	56.33
Borne B sur les y_t	92.76

TABLE: Quelques caractéristiques des observations y_t (consommations demi-horaire) du jeu de données considéré.

Les experts

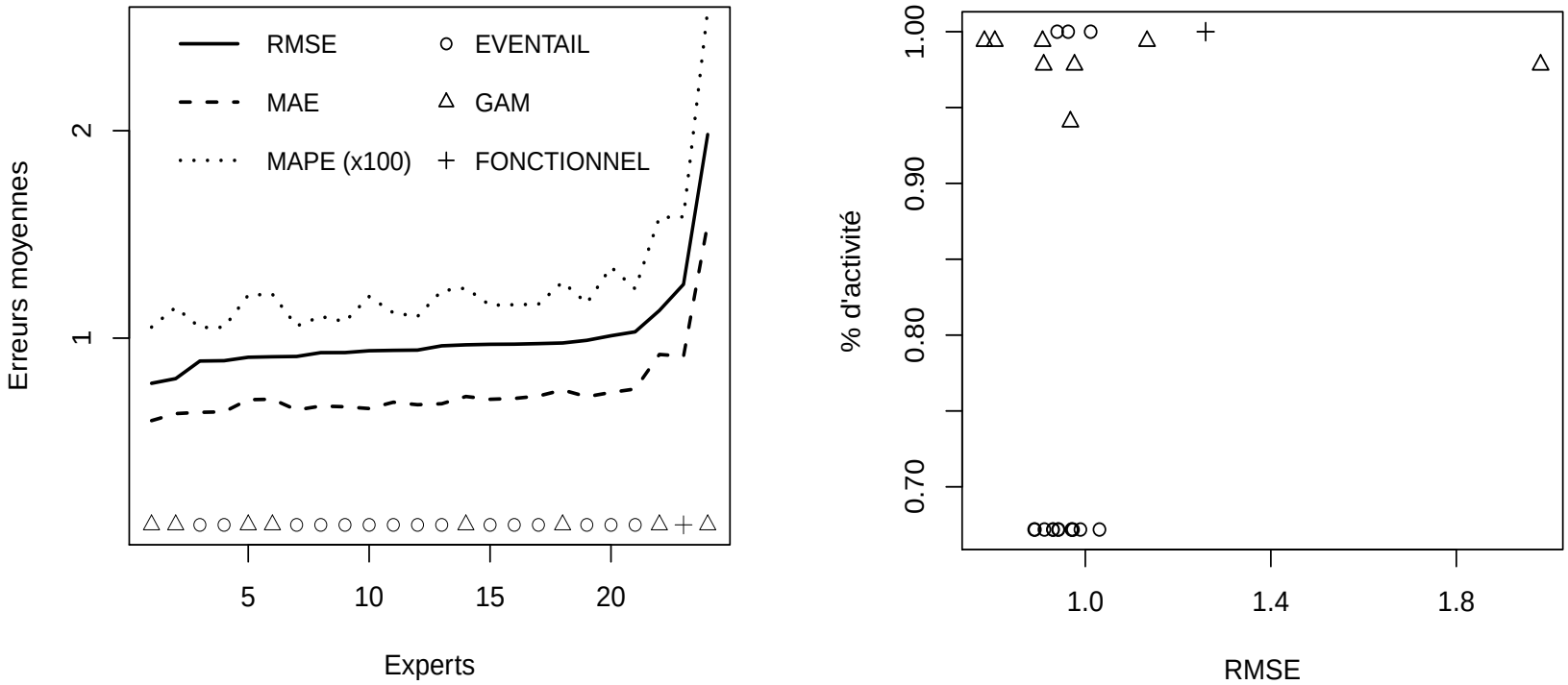


FIGURE: [Figure gauche] Erreurs moyennes et RMSEs des experts (axe-y) triés selon leur RMSEs (axe-x). [Figure droite] Frequences d'activité des experts (axe-y) selon leur RMSEs (axe-x).

Performances de références

Stratégies et oracles de référence	RMSE (MW)
Meilleur expert	782 ± 10
Meilleure combinaison convexe	658 ± 9
Meilleure combinaison linéaire	625 ± 7
Mélange uniforme	724 ± 11
Meilleur expert composé $m = 13$	629
$m = 50$	534
$m = T - 1 = 15\,359$	223

Performances des algorithmes de mélange pour des valeurs fixées des paramètres

Algorithme de mélange	RMSE (MW)	Gains (MW)
$\mathcal{E}_\eta$	718 ± 12	64
$\mathcal{S}_\eta$	691 ± 10	91
$\mathcal{F}_{\eta\alpha}$	632 ± 11	150
$\mathcal{E}_\eta^{\text{grad}}$	629 ± 8	29
$\mathcal{S}_\eta^{\text{grad}}$	631 ± 9	27
$\mathcal{F}_{\eta\alpha}^{\text{grad}}$	599 ± 9	59
$\mathcal{R}_\lambda$	650 ± 9	-25
$\mathcal{T}_\eta$	676 ± 13	106

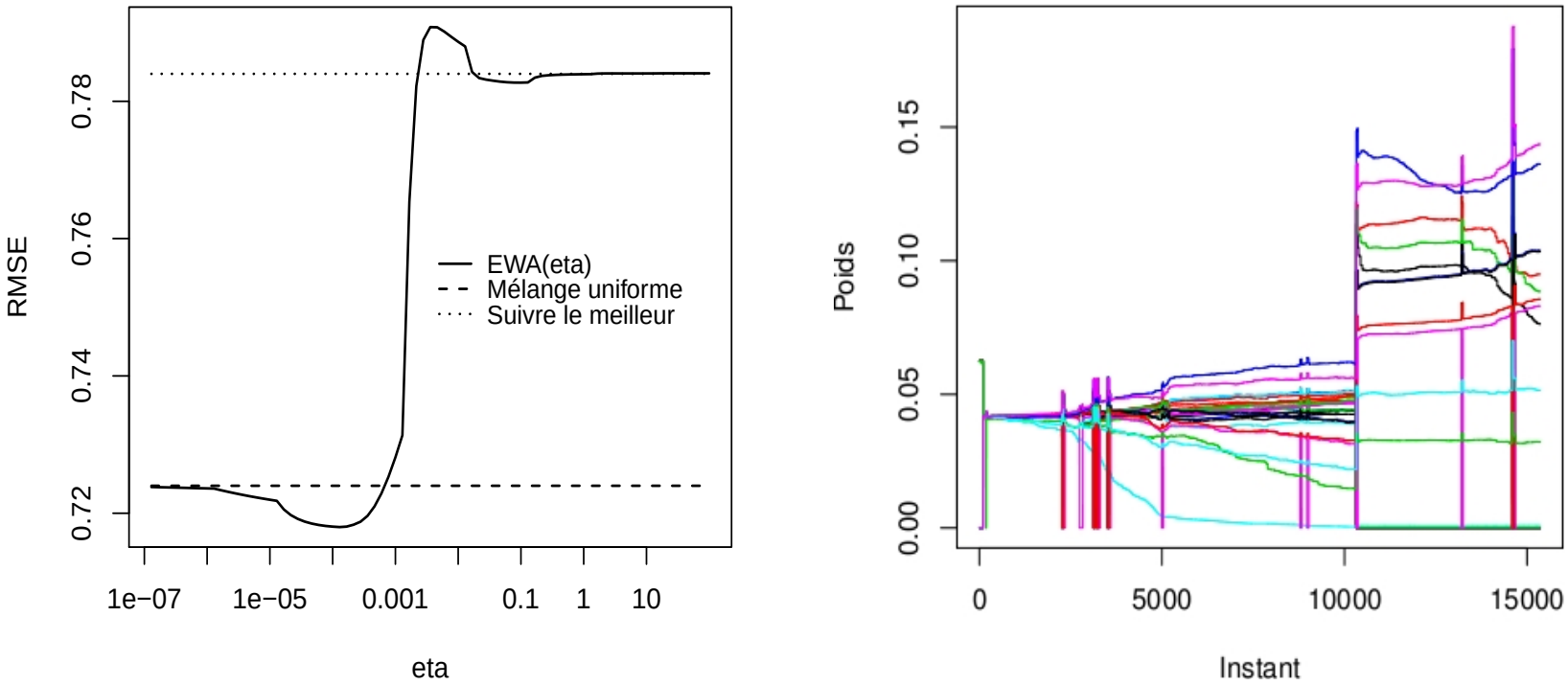


FIGURE: [Figure gauche] Performance de l'algorithme $\mathcal{E}_\eta$ en fonction de la valeur de son paramètre d'apprentissage η. [Figure droite] Évolution des poids accordés à chaque expert par l'algorithme $\mathcal{E}_\eta$.

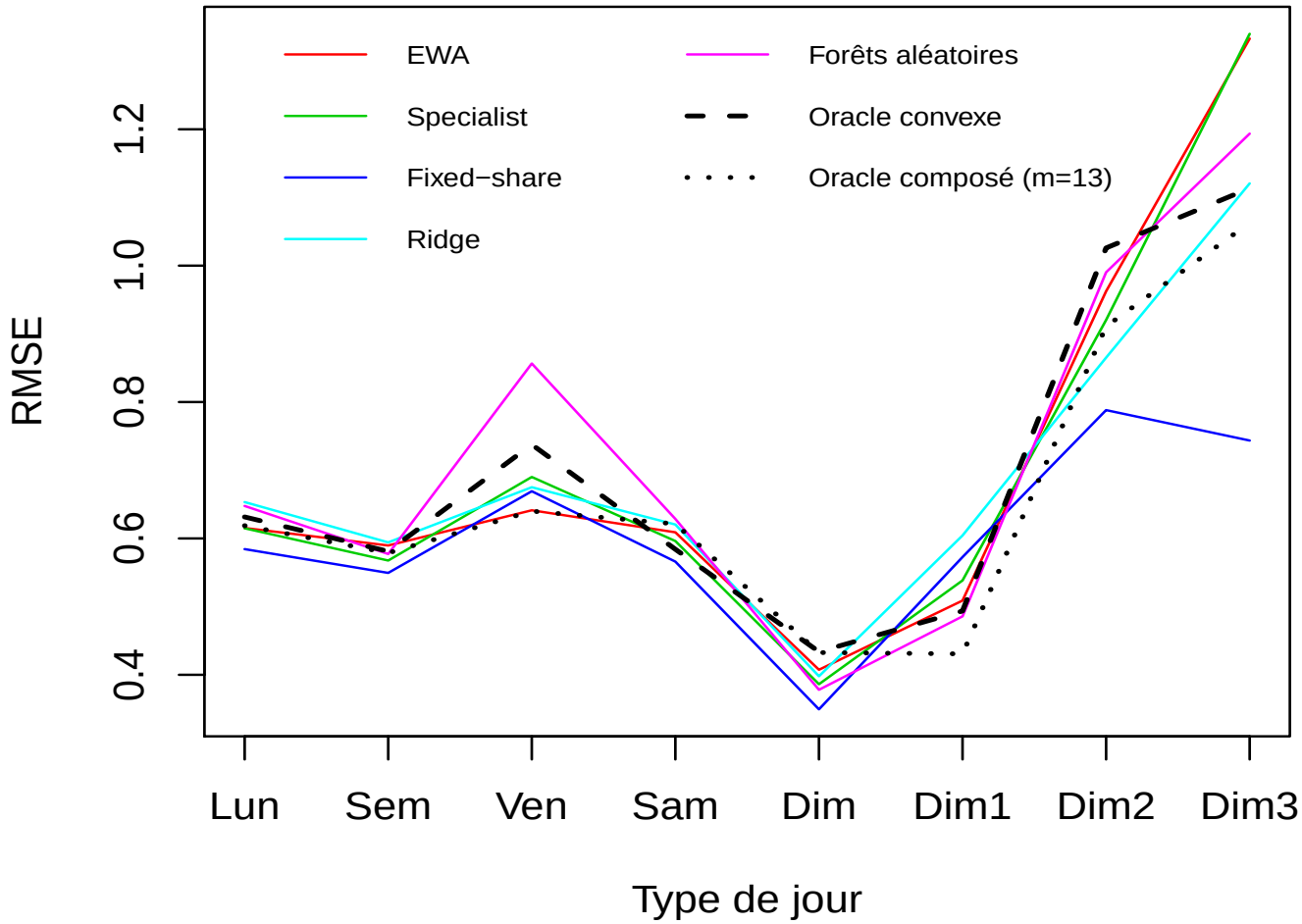


FIGURE: Erreur moyenne des différents algorithmes et de deux oracles selon le type de jour.

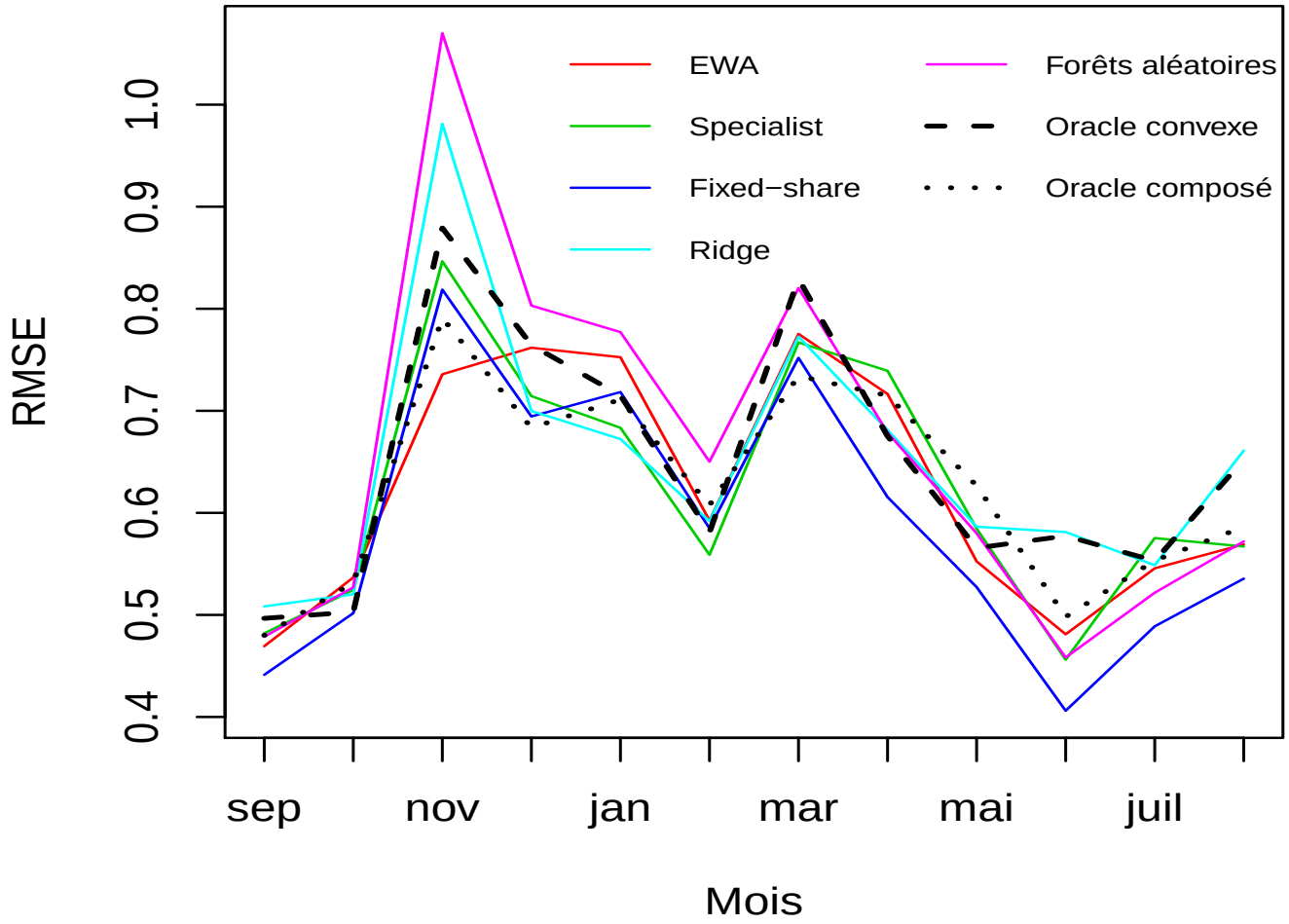


FIGURE: Erreur moyenne des différents algorithmes et de deux oracles en fonction du mois de l'année.

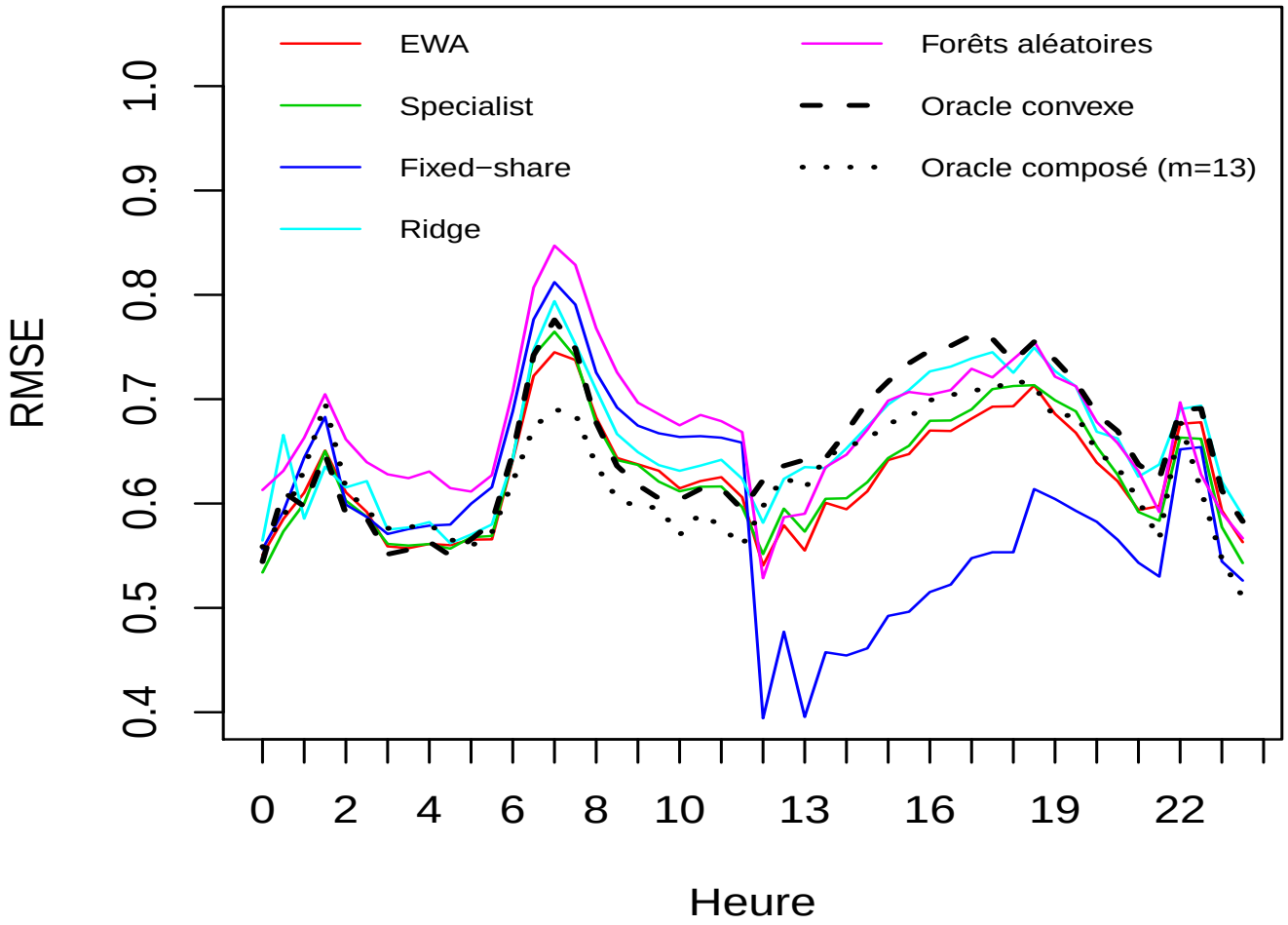
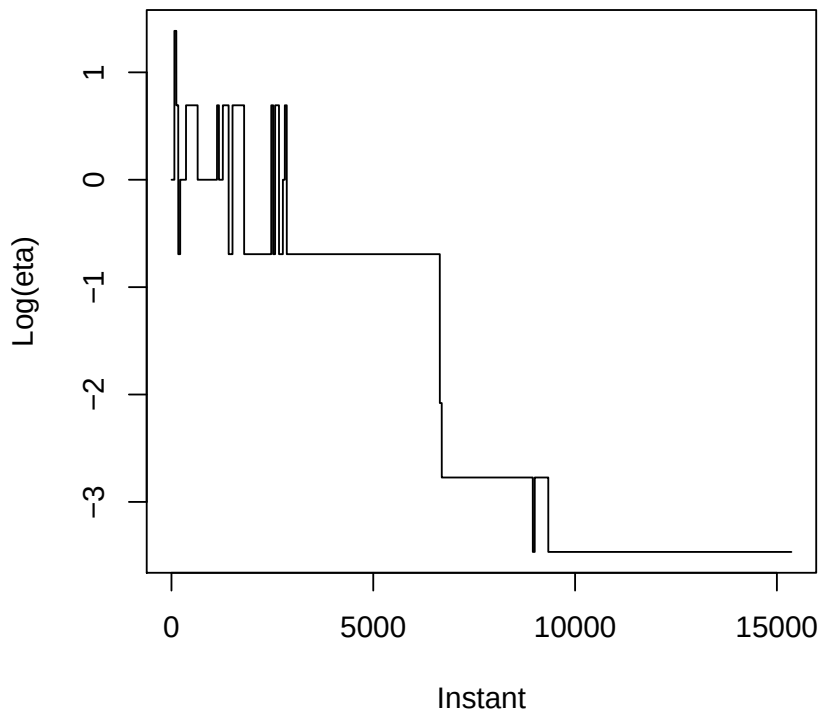


FIGURE: Erreur moyenne des différents algorithmes et de deux oracles en fonction de l'heure de la journée.

Performances des algorithmes adaptatifs

Algorithme de mélange	RMSE (MW)
$\mathcal{E}_\eta$	724 ± 11
$\mathcal{E}_\eta^{\text{grad}}$	637 ± 9
$\mathcal{S}_\eta$	715 ± 12
$\mathcal{S}_\eta^{\text{grad}}$	635 ± 9
$\mathcal{F}_{\eta\alpha}$	639 ± 11
$\mathcal{F}_{\eta\alpha}^{\text{grad}}$	623 ± 11



Conclusion

Durant ce stage, j'aurais :

- continué le travail de M. Devaine sur les algorithme EWA, specialist et fixed-share (autres fonctions de pertes, dispersion, adaptation des paramètres,...)
- définit un nouvel oracle linéaire $\rightarrow$ nouvelle version de l'algorithme Ridge
- comparé avec une méthode stochastique, les forêts aléatoires
- utilisé et adapté les forêts aléatoires pour créer de nouveaux experts

Il reste encore de nombreuses pistes à étudier !

Questions ?

Annexes

Construction de nouveaux experts par les forêts aléatoires

- Ensemble d'entrainement : (X_t, Y_t) entre 2002 et 2007
- **Objectif :** estimer la consommation Y_t en fonction des variables contextuelles X_t (météo, calendaires, temporelles).

Problème : prédicteur forêts aléatoires de base (RF_0) trop générique (décembre 2007) et supposent une stationnarité $\rightarrow$ améliorations nécessaires.

Améliorations

- Ensemble d'entrainement en ligne : RF_1
- Modèle linéaire : RF_2, prédire $\widehat{Y}_t = f(X_t)$ où

$$f \in \arg\min_{g \in \mathcal{F}} \sum_{s \in S_t} \mathrm{prox}(t,s) \left(Y_s - g(X_s)\right)^2 .$$

- Correction du biais : RF_3, prédire $\widehat{Y}'_t = \widehat{\theta}_t \widehat{Y}_t$ où

$$\widehat{\theta}_t = \arg\min_{\theta \in \mathbb{R}} \sum_{s \in S_t} \left(Y_s - \theta \widehat{Y}_s\right)^2 = \frac{\sum_{s \in S_t} X_s Y_s}{\sum_{s \in S_t} X_s^2}$$

Performances

RMSE $\times 10^3$	366 jours	320 jours
RF_0	1353	1297
RF_1	1204	1133
RF_2	1161	1074
RF_3	1130	1035