Introduction au domaine de recherche

Statistiques séquentielles éventuellement déterministes et applications.

Pierre Gaillard

École Normale Supérieure, Paris, France

Résumé

On s'intéresse ici à la prévision à court terme d'un certain phénomène. Chaque jour, on essaie de prévoir la consommation électrique en France du lendemain de façon la plus précise possible. Celle-ci est complexe et dépend de nombreux paramètres. On ne prétend pas la comprendre. Aussi, dans une première partie, on n'émet aucune hypothèse sur la façon dont elle est générée. Au contraire, on la considère comme une suite déterministe arbitraire. Afin de ne pas partir de rien et de pouvoir estimer la qualité de nos prévisions, on suppose cependant que l'on dispose d'un ensemble de prédicteurs de référence, ou experts, qui nous proposent leurs propres prévisions avant que la prochaine consommation ne soit révélée. On peut alors construire notre prévision, selon ces conseils d'experts, en essayant d'atteindre une performance proche de celle du meilleur d'entre eux.

Paris, le 30 novembre 2011

Table des matières

Je remercie vivement mes deux encadrants de stage de master 2, Yannig Goude et Gilles Stoltz pour m'avoir si bien suivi et fait découvrir ce domaine, que je vous partage à mon tour.

1. Introduction

L'énergie étant difficilement stockable, la prévision de la consommation électrique représente un enjeu important pour une entreprise commme EDF. Elle a donc développé de nombreux modèles de prévision compétitifs. Parallèlement, le paysage électrique français a évolué, avec par exemple la fin du monopole d'EDF, l'augmentation des moyens de production, ou la modification des usages de consommation électrique (climatisation, informatique, évolution économique, etc). Les modèles historiques peuvent alors être remis en cause. À qui doit-on faire confiance ? Peut-on trouver des méthodes qui mélangent de façon intelligente ces différents modèles et qui s'adaptent aux changements ?

Dans une première partie, je présente la théorie des suites individuelles, qui propose des méthodes non stochastiques efficaces pour agréger ces experts. Dans un deuxième temps, j'introduis les forêts aléatoires. On regardera comment les utiliser pour créer des méthodes de mélange stochastiques, qui permettent l'introduction de variables exogènes. Pour finir, je présenterais quelques problèmes encore ouverts et les perspectives de recherche qui s'offrent à nous.

2. Théorie des suites individuelles

Je ne présente ici qu'un très bref aperçu de la théorie, pour plus de détails le lecteur curieux est renvoyé à [CBL06].

2.1. Prévoir à l'aide de conseils d'experts

On considère la prévision séquentielle basée sur des conseils d'experts d'une suite individuelle arbitraire (y_t). On dispose d'un ensemble $E = \{1, \dots, N\}$ d'experts. À chaque instant $t = 1, \dots, T$, chaque expert $i \in E$ donne sa prévision f_{it} dans l'ensemble de sortie convexe $\mathcal{Y}$, typiquement $\mathbb{R}_+$. Un algorithme de mélange $\mathcal{A}$ forme alors un mélange $\boldsymbol{p}_t = (p_{1t}, \dots, p_{Nt}) \in \mathbb{R}^N$ et prévoit

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

La consommation réalisée y_t est alors révélée et l'instant $t+1$ commence.

On restreint souvent la prévision à un vecteur de poids convexe. C'est à dire, $\boldsymbol{p}_t \in \mathcal{X}$ où $\mathcal{X}$ est un sous-ensemble de $\mathbb{R}^N$ tel que pour tout $i \in E$, $p_{it} \geqslant 0$; $\sum_{j \in E} p_{jt} = 1$.

2.2. Estimation de la qualité d'une séquence de prévisions

Pour mesurer la qualité d'une prévision $\widehat{y}_t$ proposée à l'instant t pour l'observation y_t, on considère une fonction de perte $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. Dans la suite on suppose que ℓ est convexe en son premier argument et à valeurs dans un intervalle $[0, B]$ où B est une constante positive. À chaque instant t, le mélange $\boldsymbol{p}_t$ proposé par l'algorithme est alors évalué par la fonction de perte $\ell_t : \mathbb{R}^N \to \mathbb{R}$ définie pour tout $\boldsymbol{p} \in \mathbb{R}^N$ par

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

Notre objectif est de trouver des méthodes de mélange qui subissent une faible erreur cumulée

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

On peut définir le regret cumulé $R_T(\mathcal{A}, \mathcal{O})$ d'un algorithme $\mathcal{A}$ sur l'intervalle de temps $[1, T]$ comme la différence entre les erreurs de prévisions cumulées de $\mathcal{A}$ et de celles d'un oracle [1] $\mathcal{O}$, souvent choisi comme la meilleure stratégie de référence parmi un ensemble fixé. La réécriture

$$\mathrm{ERR}_T(\mathcal{A}) = \underbrace{\mathrm{ERR}_T(\mathcal{O})}_{\text{Erreur d'approximation}} + \underbrace{R_T(\mathcal{A}, \mathcal{O})}_{\text{Erreur d'estimation}} \tag{1}$$

fait apparaître le compromis entre erreur d'approximation et erreur d'estimation que l'on retrouve souvent en sélection de modèle. Le caractère non contrôlé de la suite à prévoir et des prévisions des experts se retrouve dans le premier terme. On cherche à obtenir des bornes sur le regret, uniformes sur l'ensemble des suites d'observations et prévisions d'experts, sous linéaires en T face à différents oracles. Des oracles envisageables sont par exemple le meilleur expert fixé, la meilleure combinaison convexe fixée ou la meilleure combinaison linéaire fixée. Bien sûr, il en existe de nombreux autres suivant le problème considéré ou notre ambition.

2.3. Deux exemples d'algorithmes d'agrégations

Dans cette partie, je présente deux méthodes de mélange étudiées en suites individuelles.

Remarque La fonction de perte étant convexe en son premier argument, on peut dériver des versions gradient de ces deux algorithmes. Celles-ci atteignent des performances de l'ordre de la meilleure combinaison convexe d'experts quand les versions de base ne se comparent qu'au meilleur expert.

Remarque Les algorithmes dépendent tous d'un ou deux paramètres d'apprentissage qui s'assimilent au fameux compromis biais-variance en apprentissage. Leur calibration joue un rôle essentiel en pratique.

2.3.1. Mélange par poids exponentiels

L'algorithme de mélange par poids exponentiels (Algorithme 1) consiste à donner à chaque expert un poids exponentiel en la perte cumulée qu'il a subit jusqu'alors.

1. Par oracle, on désigne des stratégies qui ne peuvent pas être définies séquentiellement, mais seulement rétrospectivement.

Algorithme 1 Algorithme de mélange par poids exponentiels $\mathcal{E}_\eta$.

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

`pour` les instants t `de` 1 `à` T `faire`
  `prévoir` $\widehat{y}_t = \frac{1}{\sum_{i\in E} w_{it}} \sum_{j\in E} w_{jt}\, f_{jt}$
  `observer` y_t
  `pour` les experts i `de` 1 `à` N `mettre à jour`
    $w_{it+1} = w_{it}\, e^{\eta\left(\ell_t\left(\frac{w_t}{\sum_{j\in E_t} w_{jt}}\right) - \ell_t(\delta_i)\right)}$ // voir annotation [a]

a. La version gradient $\mathcal{E}_\eta^{\mathrm{grad}}$ correspond au remplacement de ℓ_t par la pseudo-perte $\widehat{\ell}_t$ definie pour tout $\boldsymbol{u} \in \mathbb{R}^{\mathbb{N}}$ par $\widehat{\ell}_t(\boldsymbol{u}) = \nabla(\boldsymbol{u}_t) \cdot \boldsymbol{u}$ où $\nabla(\boldsymbol{u}_t)$ est un sous-gradient de la fonction de perte ℓ_t..

Proposition 1. *Pour tout $T \geqslant 1$, le regret de l'algorithme de mélange par poids exponentiels est borné par*

$$\sum_{t=1}^{T} \ell(\widehat{y}_t, y_t) - \min_{i\in E} \sum_{t=1}^{T} \ell(f_{it}, y_t) \leqslant \frac{\ln N}{\eta} + \frac{\eta}{2} B^2 T\,.$$

Le choix optimal du paramètre d'apprentissage η mène à la borne uniforme $B\sqrt{2T \ln N}$ sur le regret.

2.3.2. Fixed-share

On peut se dire que concurrencer le meilleur expert constant ne représente pas un défi suffisamment ambitieux. Rien ne présage un expert d'être le meilleur tout le temps. On peut par exemple imaginer qu'un est meilleur en hiver, un deuxième en été et un autre pendant les vacances. Il peut être alors judicieux d'accepter quelques changements d'experts. L'algorithme de mélange fixed-share (Algorithme 2) est motivé par cette idée. Il tente d'atteindre la performance de la meilleure séquence d'experts qui ne change que rarement d'experts.

On remarque que son vecteur de poids est mis à jour à chaque instant en deux étapes. La première suit l'idée d'EWA, où les experts sont repondérés selon leur performance passée de manière exponentielle. La seconde redistribue les poids en s'assurant que chacun s'attribue un poids minimal ; c'est la clef qui permet d'être compétitif face au meilleur expert composé : aucun n'expert n'est jamais complètement mis de côté et peut toujours rapidement regagner un poids non négligeable.

Proposition 2. *Pour tout $T \geqslant 1$, pour tout entier positif $m < T$, le regret de l'algorithme fixed-share est uniformément borné par*

$$\sum_{t=1}^{T} \ell_t(\boldsymbol{p}_t) - \min_{i_1,\dots,i_T \in S_T^m} \sum_{t=1}^{T} \ell(f_{i_t t}, y_t) \leqslant \frac{m+1}{\eta} \ln N + \frac{1}{\eta} \ln \frac{1}{(\alpha/N)^m\, (1-\alpha)^{n-m-1}} + \frac{\eta}{8} B^2 T\,.$$

où S_T^m est l'ensemble des séquences de T experts admettant m changements d'experts $i_t \neq i_{t+1}$.

3. Vers la construction de méthodes de mélange stochastiques

Dans cette partie, on adopte une approche stochastique. La consommation d'électricité $(Y_t) \in \mathbb{R}^T$ et les conseils d'experts $(F_{jt}) \in \mathbb{R}^{N\times T}$ sont à présent modélisés par un processus

Algorithme 2 Algorithme de mélange fixed-share $\mathcal{F}_{\eta\alpha}$.

`Entrée:` paramètre d'apprentissage $\eta > 0$ et de mélange $0 \leqslant \alpha \leqslant 1$
`Initialisation:` w_1 est la vecteur convexe uniforme, $w_{i1} = 1/N$ pour $i = 1, \ldots, N$

`pour` les instants t `de` 1 `à` T `faire`
  `prévoir` $\widehat{y}_t = \frac{1}{\sum_{i\in E} w_{it}} \sum_{j\in E} w_{jt}\, f_{jt}$
  `observer` y_t
  `pour` les experts i `de` 1 `à` N `mettre à jour`
    $v_{it} = w_{it}\, e^{-\eta \ell_t(\delta_i)}$ // voir annotation [a]
    $w_{it+1} = \dfrac{\alpha}{N} \displaystyle\sum_{i\in E} v_{it} + (1-\alpha)\, v_{jt}$

a. La version gradient $\mathcal{F}^{\text{grad}}_{\eta\alpha}$ correspond au remplacement de ℓ_t par la pseudo-perte $\widehat{\ell}_t$ definie pour tout $\boldsymbol{u} \in \mathbb{R}^{\mathbb{N}}$ par $\widehat{\ell}_t(\boldsymbol{u}) = \nabla(\boldsymbol{u}_t) \cdot \boldsymbol{u}$ où $\nabla(\boldsymbol{u}_t)$ est un sous-gradient de la fonction de perte ℓ_t.

temporel. Ils dépendent de variables contextuelles $(X_t) \in \mathbb{R}^{d\times T}$ qui peuvent être observées à chaque instant avant que la prévision ne soit proposée. On note $(\mathcal{F}_t)$ une filtration du passé définie pour tout $t \geqslant 1$ par

$$\mathcal{F}_t \quad = \quad \sigma\Big(\big\{(X_s, Y_s)\,,\ 1 \leqslant s \leqslant t-1\big\} \cup \big\{X_t\big\}\Big)\,.$$

On admet de plus que la consommation d'électricité Y_t peut s'écrire sous la forme

$$Y_t = \mathbb{E}_t[Y_t] + \varepsilon_t\,,$$

où $\mathbb{E}_t[\,\cdot\,] = \mathbb{E}\,[\,\cdot\,|\mathcal{F}_t]$ correspond à l'espérance conditionnelle sachant le passé $\mathcal{F}_t$ et chaque $\varepsilon_t \sim \mathcal{N}\big(0, \sigma^2\big)$ est un bruit aléatoire Gaussien indépendant de $\mathcal{F}_t$.

On se propose, dans le cadre de ce modèle, d'utiliser les forêts aléatoires pour créer une nouvelle méthode d'agrégation qui prend en compte l'information contextuelle (X_t) avant de proposer un mélange. Les performances des experts peuvent en effet dépendre des variables contextuelles (X_t).

Commençons par présenter rapidement les forêts aléatoires. Nous expliquerons ensuite comment les utiliser pour le mélange de prédicteurs.

3.1. Introduction aux forêts aléatoires

Les forêts aléatoires sont une modification du bagging [2] qui construit une grande collection d'arbres de prévision décorrélés, avant de les moyenner. Les arbres sont construits profondément, ils ont donc tendance à surapprendre les données. Ils ont un biais faible mais une forte variance. Les moyenner permet, s'ils sont non corrélés, de diminuer la variance, sans augmenter le biais.

2. Abbréviation de "bootstrapping and averaging". Méthode qui consiste à construire une collection de prédicteurs faibles en ne sélectionnant qu'une partie de l'ensemble d'entraînement pour chacun d'eux (bootstrapping) avant de les moyenner (averaging).

Références bibliographiques

La méthode a été introduite par Leo Breiman [Bre01] bien que de nombreuses idées soient apparues plus tôt dans la littérature, comme le bagging [Bre96], ou les arbres CART. Le site web `http://www.stat.berkeley.edu/~breiman/RandomForests` donne accès à de la documentation, du code et de nombreux rapport techniques. Une explication détaillée dans le cadre des algorithmes de classification ou de régression est disponible dans le livre [HTF09]. Les preuves de convergence ne sont pas évidentes et sont souvent obtenues pour des modèles simplifiés et éloignés de ce qui est considéré en pratique. Dans le cadre de la régression, [LJ06] a tenté d'expliquer les random forests par leur similitudes avec les K plus proches voisins. Plus récemment, [BDL08] ont prouvé des théorèmes de convergence universelle pour les algorithmes de moyennage, dont les random forests sont un cas particulier.

Le cadre théorique

On dispose d'un ensemble d'entraînement $(X_t, Y_t)_{t\in S_0}$, qui est modélisé par un processus supposé indépendant et identiquement distribué de loi $\mathcal{P}$. Pour $t \geqslant 0$, $X_t = (X_{t1}, \ldots, X_{tM})$ est la réalisation au temps t des M variables contextuelles observables pouvant expliquer la sortie $Y_t \in \mathbb{R}$. Chaque covariable X_{tm}, $1 \leqslant m \leqslant M$, est à valeurs dans un ensemble $\mathcal{X}_m$, qui est soit muni d'un ordre total, comme $\mathbb{R}$ par exemple, soit fini (ensemble de catégories).

L'objectif est, à partir de l'observation des variables contextuelles X, de prévoir la sortie Y d'un nouveau couple (X, Y) tiré selon $\mathcal{P}$, en faisant la plus petite erreur quadratique possible. Les forêts aléatoires proposent une solution efficace à ce problème, présenté en Algorithme 3.

Chaque nœud interne d'un arbre des forêts aléatoires divise l'ensemble des données en deux. Un point essentiel lors de la construction des arbres est de déterminer la bonne partition qui regroupe au mieux les données selon la variable à expliquer Y_t. L'intuition est de minimiser un critère de variances au seins de chaque sous ensemble. Des précisions sur la manière de le faire efficacement sont disponibles dans [Gai11].

3.2. Les forêts aléatoires comme méthode de mélange stochastique

L'idée est de considérer les variables contextuelles (X_t) disponibles au début de chaque instant et dont les valeurs sont un bon indicateur de performance des différents experts.

À l'instant t, on note $\widehat{Y}_t$ la prévision de cette méthode de mélange et par $\widehat{L}_t = \ell(\widehat{Y}_t, Y_t)$ et $L_{it} = \ell(F_{it}, Y_t)$ les pertes respectivement subies par elle et par l'expert i. On suppose que toutes les pertes L_{it} peuvent être modélisées par

$$L_{it} = \mathbb{E}_t[L_{it}] + \varepsilon'_{it}\,,$$

où $\varepsilon'_{it} \sim \mathcal{N}(0, \sigma_i'^2)$ est un bruit Gaussien indépendant du passé (i.e., de $\mathcal{F}_t$).

À l'instant t, on obtient l'estimation $\widehat{\ell}_{it}$ de l'espérance conditionnelle de la perte subie par l'expert i. Cette estimation peut être obtenue à l'aide de n'importe quelle méthode de régression, comme les forêts aléatoires. On admet qu'elle peut se décomposer

$$\widehat{\ell}_{it} = \mathbb{E}_t[L_{it}] + \varepsilon_{it}\,,$$

où $\varepsilon_{it} \sim \mathcal{N}\big(0, \sigma_t^2\big)$ est un bruit Gaussien indépendant du passé $\mathcal{F}_t$ et des autres bruits $(\varepsilon_{jt})_{j\neq i}$ et (ε'_{jt}).

En utilisant cette estimation de la perte de chaque expert dans l'algorithme de mélange par poids exponentiels, au lieu de leur perte cumulée, on en déduit l'algorithme de mélange stochastique présenté en Algorithme 4.

Algorithme 3 Régression par Random Forest

`Entrées :` $(X_t, Y_t)_{t\in S_0}$, ensemble d'entraînement ; $\boldsymbol{x} \in \mathcal{C}$, covariables pour la valeur à prévoir ; K, nombre d'arbres dans la forêt ; m, nombre de covariables sélectionnées à chaque coupe ; n_{feuille}, nombre de covariables X_t de l'ensemble d'entraînement maximal par feuille.

`Pour` k `de` 1 `à` K, construire l'arbre T_k ainsi :

1. Choisir un ensemble d'entraînement pour cet arbre (bootstrapping) :
   $S_k \leftarrow$ Tirer N fois avec remise et uniformément dans S_0
2. `Initialiser :` $T_k \leftarrow$ racine contenant tout l'ensemble bootstrap S_0
3. `Tant que` T_k contient une feuille ayant plus de n_{feuille} données `faire`
   a. $M' \leftarrow$ choisir uniformément m parmi les M covariables
   b. Choisir la meilleure variable et la meilleure coupe parmi les m covariables
   $$(j^\star, c^\star) = \underset{j\in M',\, c\in C_j}{\arg\min} \; \min_{a\in\mathbb{R}^2} \sum_{l=1}^{2} \sum_{X_i\in c_l} (Y_i - a_l)^2 \,,$$
   où C_j est l'ensemble des façons de couper les données présentes en deux ensembles ordonnés si un ordre est disponible pour j.
   c. Transformer la feuille en un nœud avec deux feuilles. Associer à chaque feuille respectivement les ensembles de variables contextuelles $c_1^\star$ et $c_2^\star$ et les données d'entraînement correspondantes.
4. Faire descendre $\boldsymbol{x}$ dans l'arbre jusqu'à une feuille d'ensemble de covariable associé $\mathcal{F}_k(\boldsymbol{x})$ et prévoir
   $$h_k(\boldsymbol{x}) \leftarrow \tfrac{1}{|\mathcal{F}_k(\boldsymbol{x})|} \textstyle\sum_{t:X_t\in\mathcal{F}_k(\boldsymbol{x})} Y_t$$

`Renvoyer :` $h(\boldsymbol{x}) = \frac{1}{K}\sum_{k=1}^{K} h_k(\boldsymbol{x})$

Algorithme 4 Mélange stochastique par forêts aléatoires $\mathcal{T}_\eta$

`Entrée:` suite de paramètre d'apprentissage (η_t), possiblement constante

`pour` les instants t `de` 1 `à` T `faire`

  `Construire` la forêt aléatoire avec les données $(X_s, Y_s)_{s\leqslant t-1}$ observées jusqu'à présent
  `pour` l'expert j `de` 1 `à` N `faire`
    `observer` F_{jt}
    `obtenir` $\widehat{\ell}_{jt}$ // estimée par la forêt construite ci-dessus
    $w_{jt} \leftarrow e^{-\eta_t \widehat{\ell}_{jt}}$ // mettre à jour
  `fin pour`
  `prévoir` $\widehat{y}_t \leftarrow \dfrac{1}{\sum_{i\in E_t} w_{it}} \displaystyle\sum_{j\in E_t} w_{jt} F_{jt}$
  `observer` y_t
`fin pour`

Proposition 3. *Sous les hypothèses précédentes, le regret de l'algorithme de mélange par les forêts aléatoires décrit ci-dessus, calculé avec la suite de paramètres d'apprentissage (η_t), peut être borné par*

$$\sum_{t=1}^{T} \mathbb{E}_t\Big[\widehat{L}_t\Big] - \min_{1\leqslant j\leqslant N} \mathbb{E}_t[L_{jt}] \leqslant N^2 \left(\sum_{t=1}^{T} \sigma_t + \frac{1}{N\eta_t}\right).$$

La meilleure valeur théorique est à la vue de cette borne $\eta_t = +\infty$. Cela revient à donné un poids de 1 à l'expert ayant la plus faible estimation de perte $\widehat{\ell}_{jt}$ et 0 à tous les autres experts. La borne devient alors

$$\sum_{t=1}^{T} \mathbb{E}_t\Big[\widehat{L}_t\Big] - \min_{1\leqslant j\leqslant N} \mathbb{E}_t[L_{jt}] \leqslant N^2 \sum_{t=1}^{T} \sigma_t$$

(et peut être facilement améliorée en une borne linéaire en N). Cependant, pour $\eta_t \simeq (1/N)\sum_{t=1}^{T}\sigma_t$, on ne perd qu'un facteur constant en T par rapport à cette borne théorique optimale alors que la performance en pratique est améliorée d'environ 10%.

4. Perspectives et problèmes ouverts

Je présente ici quelques problèmes non résolus pour l'instant, qui pourraient être développés par la suite en stage ou en thèse. Toute idée est bienvenue !

4.1. Extension et amélioration des algorithmes de mélange pour les experts spécialisés

En pratique, tous les experts ne proposent pas forcément des prévisions à chaque instant. Certains sont spécialisés et sont performants dans des périodes connues à l'avance de manière déterministe. Ils ne proposent alors leurs prédictions qu'au cours de ces périodes. À chaque instant t, seul un sous-ensemble $E_t \subset E$ des experts est actif et donne une prévision. La théorie des suite individuelles s'étend en grande partie à ce cadre (voir [DGS11]).

L'algorithme de mélange par poids exponentiels (Algorithme 1) s'adapte par exemple facilement en conservant $w_{it+1} = w_{it}$ si l'expert $i \notin E_t$, et en ne faisant la moyenne pondérée que sur les experts actifs pour proposer la prévision $\widehat{y}_t$. Son regret face à un expert i est alors définit et majoré par

$$R_T(\mathcal{E}_\eta, i) = \sum_{t=1}^{T} \big(\ell(\widehat{y}_t, y_t) - \ell(f_{it}, y_t)\big)\mathbb{1}_{\{i\in E_t\}} \leqslant \frac{\ln N}{\eta} + \frac{\eta}{2}B^2T\,.$$

La valeur optimale de η mène à la majoration par $B\sqrt{2T\ln N}$. [BM07] ont montré comment l'améliorer en $B\sqrt{2\ln N\sum_{t=1}^{T}\mathbb{1}_{\{i\in E_t\}}}$ en considérant des pertes pénalisées dans l'algorithme $\mathcal{E}_\eta$ et un paramètre d'apprentissage η_i dépendant de l'expert. L'algorithme fixed-share s'adapte également, mais de façon moins directe cependant.

Quelques lacunes perdurent toutefois en tout ce qui concerne les mélanges non convexes. La régression Ridge, qui propose des mélanges linéaires, n'a par exemple pas encore été adaptée à ce cadre, alors qu'elle est la méthode la plus performante sur de nombreux jeux de données. Aucun algorithme ne réussit pour l'instant avec des experts spécialisés à atteindre la performance de la meilleure combinaison linéaire. D'ailleurs, qu'est-ce que la meilleure combinaison linéaire ? La réponse à cette simple question n'est déjà pas évidente avec les activations et les désactivations

des experts. Une façon de la définir serait de généraliser la définition de la perte cumulée ERR_T pour tout $\boldsymbol{u} \in \mathbb{R}^N$ par

$$\text{ERR}_T(\boldsymbol{u}) = \frac{T}{\sum_{t=1}^T \left|\tau_t(\boldsymbol{u})\right|} \sum_{t=1}^T \ell_t\left(\frac{\boldsymbol{u}}{\tau_t(\boldsymbol{u})}\right) \left|\tau_t(\boldsymbol{u})\right|,$$

où $\tau_t(\boldsymbol{u}) = \sum_{j\in E_t} u_j / \sum_{k=1}^N u_k$; puis de choisir le vecteur linéaire $\boldsymbol{u}^\star$ la minimisant. Cette définition a l'avantage de généraliser à la fois celle de la meilleure combinaison convexe mais aussi le cas non spécialisé. D'autres définitions sont cependant envisageables. Une adaptation de la régression ridge se déduit de cette définition. Ses résultats pratiques sont correctes mais elle ne dispose encore d'aucune borne théorique. Peut-on y arriver ?

4.2. Fixed-share pour s'approcher de nouveaux oracles

L'algorithme fixed-share (Algorithme 2) dépend d'un paramètre d'apprentissage η, ainsi que d'un paramètre de mélange α. On gagne beaucoup en liberté à ne pas les prendre constants mais dépendants des experts ou de l'instant t.

À partir d'un algorithme dérivé de fixed-share, [BW03] arrivent par exemple à concurrencer la meilleure séquence d'experts qui n'utilise qu'un petit sous ensemble d'experts. On peut y arriver en considérant fixed-share avec un paramètre de mélange α_{it} variant dans le temps et selon les experts. Dans de nombreux problèmes similaires, considérer des nouveaux oracles permet de réduire l'erreur d'approximation dans (1). Cependant, les algorithmes actuels ne sont pas adaptés pour ces oracles et l'erreur d'estimation s'en retrouve détériorée. Utiliser différents α_{it} dans l'algorithme fixed-share lui donne beaucoup de liberté et permet peut-être de l'adapter spécialement à de nouveaux oracles, comme par exemple à la meilleure séquence d'experts ne retournant jamais deux fois vers un expert déjà utiliser.

4.3. Le mélange stochastique et l'introduction de variables exogènes

Je n'ai présenté ici que brièvement les forêts aléatoires et j'ai proposé une façon de les utiliser pour le mélange dans le cadre d'un modèle très simple. Cependant, cela n'a pas du tout été exploré pour l'instant. Un travail conséquent est à prévoir pour étendre les algorithmes de mélanges à des algorithmes stochastiques. Quelles hypothèses sont judicieuses ? Quelles bornes peut-on prouver ? Peut-on les utiliser pour créer de nouvelles méthodes de mélanges non stochastiques ? Des méthodes comme Adaboost, ou les forêts aléatoires nous donnent en effet déjà plus ou moins l'impression de faire du mélange, peut-on s'en inspirer ?

La méthode que j'ai proposée utilise en fait les forêts aléatoires comme une boite noire. On pourrait les remplacer par n'importe quelle méthode de régression. Présentons rapidement une autre idée pour faire du mélange avec les forêts aléatoires à l'aide de variables exogènes X_t, cette fois ci spécifique à celles-ci.

Les forêts aléatoires induisent une notion de *proximité* entre les entrées X_t. Intuitivement, si deux ensembles de covariables X_{t_1} et X_{t_2} tombent souvent dans les mêmes feuilles des arbres, on peut supposer qu'elles expliquent la sortie Y de façon similaire. Plus formellement, après que les K arbres de la forêt ont été construits, on fait descendre toutes les données au niveau des feuilles et on définit la proximité entre deux observations t_1 et t_1 par

$$\text{prox}(X_{t_1}, X_{t_2}) = \frac{1}{K} \sum_{k=1}^K \mathbb{1}_{\{X_{t_1} \text{ et } X_{t_2} \text{ tombent dans la même feuille dans l'arbre } k\}}.$$

On peut alors utiliser cette notion de proximité pour choisir le vecteur de mélange $\boldsymbol{u}_t$ à l'instant t par

$$\boldsymbol{u}_t \quad \in \quad \underset{\boldsymbol{u}\in\mathbb{R}^N}{\arg\min} \sum_{s=1}^{t-1} \mathrm{prox}(X_t, X_s)\,(Y_s - \boldsymbol{u}\cdot F_t)^2 \ ,$$

où $F_t = (F_{1t}, \ldots, F_{Nt})^T \in \mathbb{R}^N$ correspond aux prévisions des experts. Toute la théorie correspondante reste à découvrir.

4.4. La gestion des incertitudes

Plutôt que prévoir uniquement des réels $\widehat{y}_t$ pour la consommation électrique y_t, il est nécessaire en pratique de prévoir des lois ou des intervalles de confiances. Dans cette optique, on distingue alors deux idées suivant les experts à notre disposition.

Si les experts nous fournissent déjà une mesure d'incertitude et proposent des densités de probabilités plutôt que des réels $f_{it} \in \mathbb{R}$, il est naturel d'essayer de combiner ces lois.

Si les experts ne fournissent que des réels f_{it}, on peut tout de même tenter de tirer des incertitudes de leurs prévisions. Cela part d'une intuition pratique de la part des opérateurs ayant en charge la prévision d'électricité. Lorsque les opérateurs reçoivent plusieurs prévisions de consommation, si celles-ci sont significativement différentes entre elles, cela signifie pour eux un risque accru pour la prévision. En revanche, si ces différentes prévisions concordent, le risque semble plus faible. Cette idée relativement naïve pourrait être formalisée et exploitée dans un mélange.

Références

[BDL08] G. Biau, L. Devroye, and G. Lugosi. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9 :2015–2033, 2008.

[BM07] A. Blum and Y. Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8 :1307–1324, 2007.

[Bre96] L. Breiman. Bagging predictors. *Machine Learning*, 24(2) :123–140, 1996.

[Bre01] L. Breiman. Random forests. *Machine Learning*, 45(1) :5–32, 2001.

[BW03] O. Bousquet and M. K. Warmuth. Tracking a small set of experts by mixing past posteriors. *The journal of Machine Learning Research*, 2003.

[CBL06] N. Cesa-Blanchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[DGS11] M. Devaine, Y. Goude, and G. Stoltz. Forecasting the electricity consumption by aggregating specialized experts. 2011.

[Gai11] P. Gaillard. Prévision de la consommation électrique par agrégation séquentielle de prédicteurs spécialisés. *Rapport de stage de M2*, 2011.

[HTF09] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009.

[LJ06] Y. Lin and Y. Jeon. Random forests and adaptive nearest neighbors. *Journal of American Statistical Association*, 101 :578–590, 2006.