Prévision multi-échelle par agrégation de forêts aléatoires. Application à la consommation électrique.

Benjamin Goehry

NNT : 2019SACLS461

THÈSE DE DOCTORAT

de

L'UNIVERSITÉ PARIS-SACLAY

École doctorale de mathématiques Hadamard (EDMH, ED 574)

Établissement d'inscription : Université Paris-Sud

Laboratoire d'accueil : Laboratoire de mathématiques d'Orsay, UMR 8628 CNRS

Spécialité de doctorat : Mathématiques appliquées

Benjamin GOEHRY

Prévision multi-échelle par agrégation de forêts aléatoires. Application à la consommation électrique.

Date de soutenance : 10 Décembre 2019

Après avis des rapporteurs : AVNER BAR-HEN (CNAM)
JEAN-MICHEL LOUBES (Université Paul-Sabatier)

Jury de soutenance :

AVNER BAR-HEN	(CNAM) Rapporteur
GÉRARD BIAU	(Sorbonne Université) Examinateur
ANNE-LAURE FOUGÈRES	(Université Claude Bernard Lyon 1) Examinatrice
JEAN-MICHEL LOUBES	(Université Paul-Sabatier) Rapporteur
PASCAL MASSART	(Université Paris-Sud) Codirecteur de thèse
ÉRIC MATZNER-LØBER	(Université Rennes 2) Président du jury
JEAN-MICHEL POGGI	(Université Paris-Descartes) Codirecteur de thèse
YANNIG GOUDE	(EDF Saclay) Invité

Remerciements

Avant toutes choses, je tenais à remercier tout particulièrement Pascal Massart et Jean-Michel Poggi pour avoir accepté d'encadrer et de superviser ma thèse. Leur soutien perpétuel et les nombreux conseils qu'ils m'ont prodigués tout au long de ces trois années furent porteurs et indispensables au bon déroulé de mes travaux.

Je remercie également Yannig Goude pour son encadrement et son expertise industrielle sur une thèse qui se voulait initialement être CIFRE mais qui n'en est finalement pas une.

Un grand merci à Avner Ben-Har et Jean-Michel Loubes pour avoir accepté de rapporter ma thèse mais aussi à Gérard Biau, Anne-Laure Fougères et Éric Matzner-Løber pour avoir accepté de faire partie de mon jury de thèse.

Merci également à l'école doctorale, en particulier Stéphane Nonnenmacher et Frédéric Paulin, pour avoir toujours simplifié autant que possible les tâches administratives auxquelles j'ai été confronté. Une mention spéciale pour Mathilde Rousseau pour avoir su régler efficacement mes problèmes informatiques et sauver ma thèse mise en péril par des aléas malchanceux de la vie.

Je remercie aussi les doctorants du laboratoire du midi, à savoir Camille, Corentin, Cyril, Elio, Gabriel, Guillaume L., Guillaume M., Hedi, Hugo, Jean, Jeanne, Louise, Luc, Margaux, Pierre-Louis, Romain, Thomas, Yassine, Yoel avec qui j'ai partagé de fort sympathiques discussions à table ou encore à l'occasion de pauses café.

Je remercie Dr. François pour tous ces moments improductifs où nous avons pu échanger sur des sujets délicats et hautement confidentiels, pour tous ces instants de rigolade mais aussi pour les folles aventures que nous avons été amenés à partager depuis le début de notre odyssée doctorale.

Merci à Antoine, Céline, Juliette, Pierre, Queja et Robert pour m'avoir si chaleureusement accueilli dans votre groupe d'amis. Même s'il m'arrive de louper (de manière totalement involontaire et fortuite) certaines de vos soirées, ce fut toujours un grand plaisir de passer du temps avec vous.

Je profite également de l'occasion pour remercier la team (Discord) Adrien, Rafa, Raphy, Samuel, Toufik et Tran pour m'avoir fait perdre un temps monstrueux sur Internet, passé majoritairement à rire, à se décarcasser devant des parties de jeux interminables et à écouter les histoires personnelles de chacun totalement irréalistes et invraisemblables.

Je remercie également la team Sous-marin composée de Nicolas, Yannick et Fabio avec qui j'ai eu l'occasion de partager une partie de mes études dans la belle ville de Strasbourg et avec lesquels je poursuis maintenant l'histoire autour d'un bon picon, de burgers et parfois de fondues. Mais je n'oublie pas non plus mes amis restés en Alsace dont la liste serait trop longue à énumérer mais ils se reconnaîtront.

Je remercie Élodie pour m'avoir supporté et encouragé pendant cette dernière année, je ne peux te remercier assez pour les nombreuses corrections (syntaxiques et grammaticales, je tiens à préciser) que tu m'as fournies mais surtout pour une dernière année de thèse bien meilleure que je ne pouvais l'espérer à tes côtés.

Enfin, je ne serais jamais arrivé jusqu'ici sans la confiance et le soutien inconditionnels de mes parents, qui m'ont toujours soutenu tout au long de ma vie, m'ont permis de surmonter bien des épreuves et m'ont poussé pour que j'arrive le plus loin possible.

Contents

Liste des figures

Liste des tableaux

Chapter 1

Introduction

Le travail présenté ici a été réalisé dans le cadre d'une thèse au Laboratoire de Mathématiques d'Orsay et d'une collaboration industrielle avec EDF R&D-département OSIRIS. Dans cette introduction, nous exposons le contexte et les motivations pratiques, avant de présenter les contributions sur les sujets directement en lien avec l'aspect énergétique. S'ensuit, dans un second temps, une présentation des forêts aléatoires dans un cadre plus théorique, qui débouche ensuite sur un ensemble de contributions théoriques dans un cadre de dépendance. Vient en fin la dernière contribution, de nature méthodologique, présentant une nouvelle variante de forêts aléatoires pour les séries temporelles.

1.1 Généralités sur les systèmes électriques

La prévision de la consommation électrique pour les fournisseurs d'électricité comme EDF est un enjeu capital. Du fait des problématiques actuelles en matière de stockage de l'électricité et de la forte volatilité de la clientèle, celle-ci s'avère en effet nécessaire pour assurer une bonne gestion des différents moyens de production et optimiser au mieux les réseaux électriques. Afin d'éviter l'accumulation de problèmes financiers et physiques tels que les pannes de courant locales ou encore les blackouts, une bonne prévision de la consommation établie sur différents horizons temporels apparaît comme un élément indispensable. Ceci est d'autant plus vrai avec le développement des énergies renouvelables qui induisent de nouveaux aléas.

La consommation électrique fluctue selon beaucoup de variables. Prenons l'exemple de la consommation d'électricité en France métropolitaine. La variable la plus significative est le cycle temporel qui se divise en trois composantes :

- le cycle annuel. Le climat généralement plus froid en hiver induit une utilisation plus intensive des appareils électriques comme le chauffage ce qui conduit à une consommation en hiver supérieure à celle de l'été. En parallèle, le ralentissement de l'activité économique pendant la période estivale entraîne également une diminution de la consommation. Ce phénomène est représenté en Fig. 1.1a et Fig. 1.1b (source : https://opendata.reseaux-energies.fr/explore/dataset).

- Le cycle hebdomadaire : la consommation est plus élevée les jours ouvrables que les jours fériés ou le week-end, comme illustré en Fig. 1.2b.

- Le cycle journalier qui se caractérise par une consommation évidemment bien plus grande en journée que la nuit. Une représentation de ce phénomène est donnée en Fig. 1.2a.

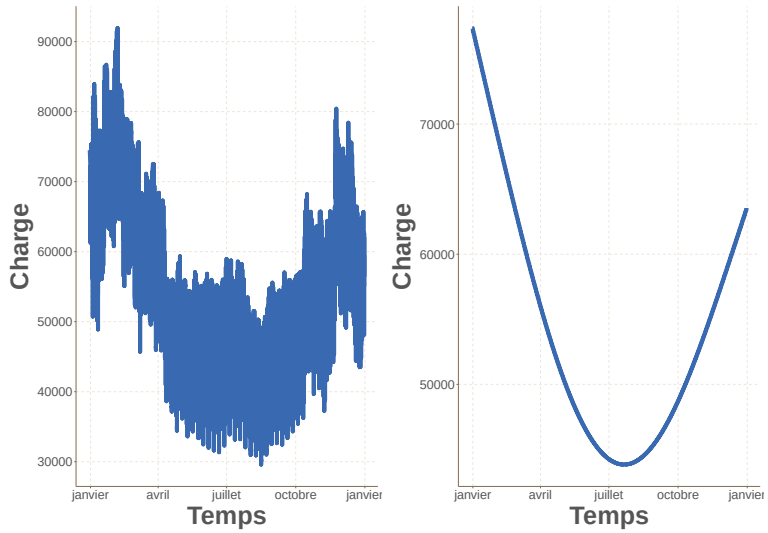

Figure 1.1: Consommation française en 2015 : a brute, et b tendance.

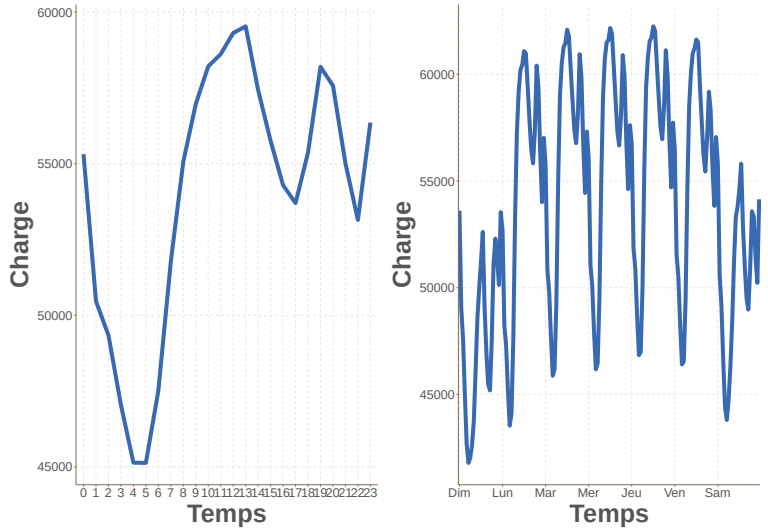

Figure 1.2: Consommation moyenne (française) : a journalière, et b hebdomadaire.

Notons que ces cycles dépendent du jeu de données et du niveau d'agrégation considérés. Au cours de cette thèse je me suis intéressé à d'autres jeux de données, comme celui des consommations d'électricité de résidentiels en Irlande (Commission for Energy Regulation [2011], plus de détails en chapitre 3). Le cycle journalier est alors inversé avec une hausse de la consommation le matin vers 8-9h ainsi qu'un pic de consommation le soir autour de 19h. Le cycle hebdomadaire est également impacté par une hausse de la consommation le week-end, les personnes restant plus généralement chez elles et par un maintien de la consommation qui demeure constante les soirs du week-end par rapport aux autres soirs de la semaine. Ces dernières remarques sont illustrées en Fig. 1.3a et Fig. 1.3b

La température est aussi un des facteurs les plus influents sur la consommation, les clients ayant en effet tendance à allumer le chauffage en période hivernale à cause du froid, comme illustré en Fig. 1.4. Mais d'autres facteurs influent encore la consommation d'électricité comme les jours fériés, les évènements rares ou encore, à une granularité plus fine, la tarification choisie par le client.

Le paysage électrique présente de nouveaux défis tant en termes de production que de

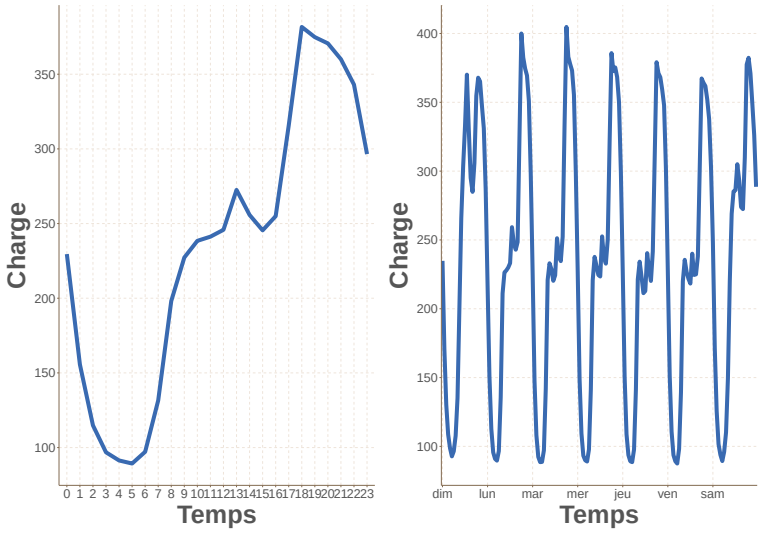

Figure 1.3: Consommation moyenne de consommateurs résidentiels en Irlande : a journalière, et b hebdomadaire.

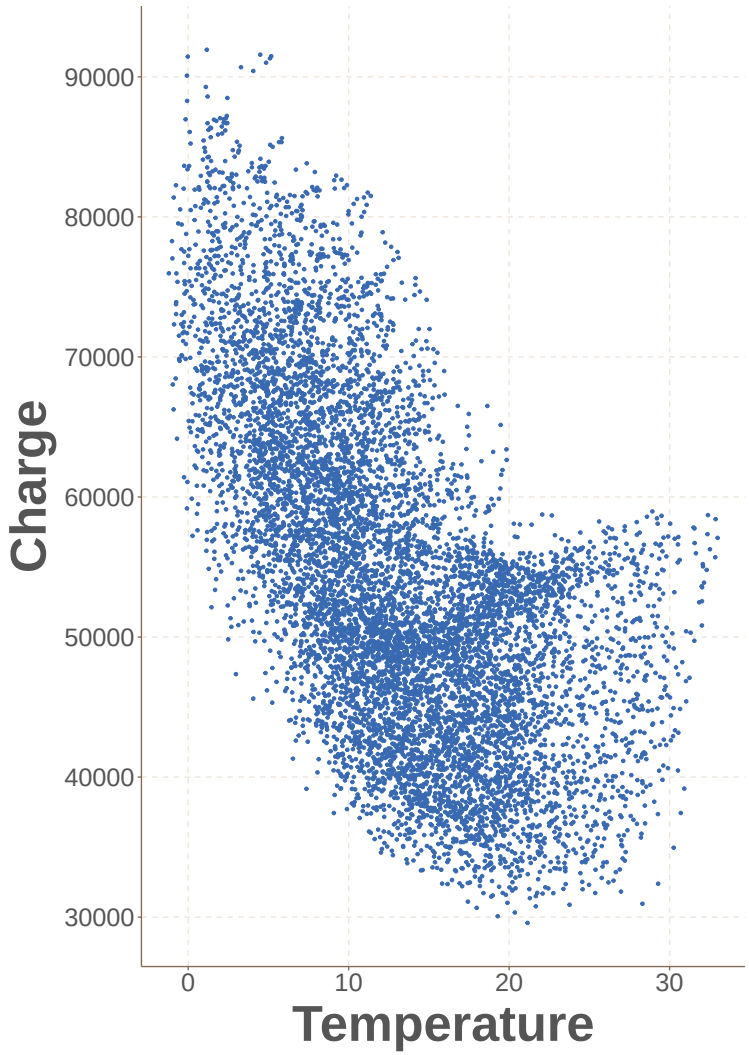

Figure 1.4: Consommation française en fonction de la température moyenne nationale.

consommation d'électricité. L'ouverture des marchés électriques et les habitudes de consommation évoluent. La production plus intermittente induit des incertitudes supplémentaires et la nécessité d'une optimisation locale du marché de l'électricité (voir par exemple Lezama et al. [2018]). En réaction à ces enjeux et au développement des véhicules électriques, des pompes à chaleur ou de l'isolation des bâtiments (voir Fischer et al. [2015]; Kikusato et al. [2019]), des nouvelles méthodes plus robustes et adaptatives sont nécessaires pour mieux répondre à la présence d'incertitudes et de non stationnarités grandissantes.

Une autre source de nouveauté sont les compteurs intelligents (*smart grids* en anglais) qui donnent aux fournisseurs d'énergie une information à une granularité plus fine et plus fréquente. A titre d'exemple, le jeu de données irlandais mentionné précédemment nous fournit une consommation d'électricité pour chaque individu à une période d'une demi-heure. Le développement récent de réseaux électriques intelligents et de nouvelles infrastructures avancées de comptage apportent de nouvelles opportunités et de nouveaux défis pour les services publics. Exploiter l'information des compteurs intelligents pour la prévision est un point clé pour les fournisseurs d'énergie qui doivent faire face à un portefeuille de clients fluctuant dans le temps mais aussi pour les gestionnaires de réseaux qui doivent améliorer la précision des prévisions locales pour faire face au développement de la production d'énergie renouvelable distribuée. Les compteurs intelligents permettent une communication bilatérale entre producteurs, gestionnaires de réseaux et consommateurs. Le système de gestion de l'énergie domestique (*Home Energy Management System* en anglais) et les outils de réponse à la demande font du contrôle de la charge une réalité et encouragent les fournisseurs d'électricité à concevoir de nouvelles offres (voir Shareef et al. [2018]). Les données induites par les compteurs intelligents et leur exploitation pour les entreprises sont énormes. Selon Wang et al. [2017] la Chine va bientôt générer 5 téraoctets par an à partir des données de consommation d'électricité collectées toutes les 15 minutes par des compteurs intelligents. L'exploitation de ces données est cruciale pour améliorer la stratégie commerciale, la politique des prix et pour développer de nouveaux services personnalisés tels que le diagnostic énergétique et les recommandations.

1.1.1 Méthodes de prévision et sources de travail

Les travaux présentés aux chapitres 3 et 4 (et leurs compléments en appendices A, B, C) sont l'aboutissement d'une collaboration industrielle s'étalant de juin 2017 à juin 2019 entre le Laboratoire Mathématiques d'Orsay et EDF. Celle-ci s'inscrit dans la continuité des nombreuses collaborations qui se sont opérées entre ces deux entités. La suite décrit succinctement une partie de cette histoire et les méthodes de prévisions utilisées chez EDF.

Ondelec Dans le cadre d'une série de collaborations de recherche (cf. Misiti et al. [2005, 2006a,b, 2007a,b, 2008, 2009]) entre EDF et l'Université d'Orsay, les travaux effectués ont ouvert la réflexion sur l'intérêt de la désagrégation et de la classification pour un objectif de prévision. L'idée est de décomposer le signal global de telle façon que la somme des prévisions des synchrones par classe améliore de manière significative la prédiction de la synchrone au niveau global. La stratégie est d'optimiser une classification préliminaire des courbes individuelles par rapport à un indice de prévisibilité. La procédure de classification optimisée est pilotée par un indice de prévisibilité croisée. Ces travaux ont donné lieu à la publication Misiti et al. [2010].

KWF Antoniadis et al. [2012, 2014] et ensuite Cugliari et al. [2016] proposent *KWF*, un modèle non paramétrique de prévision pour des séries chronologiques fonctionnelles en

présence de non stationnarités, sans utiliser de variables exogènes et pour des horizons plus ou moins éloignés. Le principe général du modèle de prévision consiste à trouver dans le passé des contextes similaires à la situation présente et prévoir le futur par une combinaison linéaire des passés les plus semblables au présent où la notion de similarité est basée sur les ondelettes. Plusieurs stratégies sont alors mises en œuvre pour prendre en compte les diverses sources d'instationnarités. Plus précisément dans le travail de Antoniadis et al. [2012], nous trouverons les principes de la méthode et la prise en compte des instationnarités classiques ainsi que les premières expériences sur des scénarios de pertes de clients montrant la capacité de prendre en compte rapidement les changements dans Antoniadis et al. [2014].

GAM Une autre source de travail est constituée des techniques plus modernes, en particulier les modèles additifs généralisés (***G**eneralized **A**dditive **M**odel* en anglais), introduits par Hastie and Tibshirani [1986]. Il s'agit cette fois-ci de modèles non paramétriques qui présentent une grande souplesse en expliquant la réponse par une somme d'effets non linéaires des variables explicatives. Des modèles de prévisions de la consommation dans le cadre d'EDF sont développés dans Goude et al. [2013]; Pierrot and Goude [2011] qui offrent d'excellentes performances pour la prévision de la consommation électrique. Ils offrent plus de souplesse et un potentiel d'améliorations par une meilleure adaptation pour les prévisions dans les cas d'instastionnarité ainsi que pour la construction de modèles pour des groupes ayant peu d'individus en permettant d'intégrer les variables exogènes. Ces modèles ont également fait l'objet de la thèse de Thouvenot Thouvenot [2015] qui présente des procédures automatiques de sélection et d'estimation de composantes d'un modèle additif avec des estimateurs en plusieurs étapes. Ces modèles sont actuellement utilisés par les opérateurs ayant en charge la prévision de la consommation d'électricité.

Agrégation d'experts Les méthodes d'agrégation d'experts séquentielle sont des méthodes qui consistent à associer des poids à chaque expert variant au cours du temps avec pour objectif de trouver le meilleur expert ou la meilleur combinaison d'experts selon une perte choisie à l'avance, par exemple la perte quadratique. Pour cela, les poids sont adaptés à chaque pas de temps en fonction des performances passées des experts.

La thèse de Goude Goude [2008] montre que le mélange des prédicteurs est efficace pour la prévision à court terme et présente des résultats théoriques en présence de rupture. Dans le prolongement de cette thèse et réalisée dans le même cadre que Goude, la thèse de Gaillard Gaillard [2015] propose de nouveaux algorithmes qui cherchent à s'adapter automatiquement à la difficulté de la suite. Cette thèse incorpore par ailleurs à ces algorithmes des garanties théoriques sous des hypothèses peu restrictives sur les données et les utilise sur divers jeux de données (par exemple sur la consommation d'électricité ou le prix de l'électricité) entre autres sur les données du portefeuille EDF en utilisant notamment des experts issus de KWF et des modèles additifs.

Modèles de mélange Une dernière source est la thèse académique de Devijver Devijver [2015] dans le cadre des modèles de mélange pour des données fonctionnelles en régression. Elle y propose en particulier deux procédures pour estimer les paramètres de ces modèles et en fait une étude théorique. Elle met également en pratique une des procédures, appelée Lasso-EMV, pour la classification des consommateurs électriques dans le but d'améliorer la prévision sur le jeu de données de résidentiels en Irlande. Dans ce contexte, la classification et la prévision des courbes de charge individuelles s'effectuent alors en une seule étape.

1.2 Prévision ascendante de la consommation d'électricité

Dans le contexte des compteurs intelligents, comme le souligne l'enquête Wang et al. [2018], les données à une granularité fine de ces compteurs offrent de nouvelles possibilités d'améliorer les prévisions à différents niveaux d'agrégats : national, régional, sous-stations du réseau de distribution, une ville, un district, etc. Un premier objectif est l'exploitation des données de charge individuelles pour la prévision à court terme de la charge, qui est l'entrée principale pour la gestion de l'énergie et qui s'avère cruciale pour l'optimisation du réseau. Dans la suite, nous notons les charges individuelles par $(Y_{t,i})_{\substack{t\in\mathbb{Z}\\1\leq i\leq n}}$ et la charge totale par $(Y_t)_{t\in\mathbb{Z}}$ qui est la somme des charges des n individus,

$$Y_t = \sum_{i=1}^{n} Y_{t,i}.$$

Supposons que l'on observe également des facteurs externes qui peuvent dépendre de l'individu i, $(X_{t,i})_{\substack{t\in\mathbb{Z}\\1\leq i\leq n}}$, typiquement les informations météorologiques locales associées à chacun des individus ainsi que l'observation pour un individu i d'un vecteur non temporel I_i qui représente les informations propres à l'individu comme la tarification, le type de chauffage ou son statut social dans le cas d'un résident. Une stratégie répandue pour la prévision de la charge totale Y_t est la prévision ascendante (traduit *bottom-up strategy* en anglais). La stratégie de prévision ascendante consiste dans un premier temps à faire un clustering sur les individus, c'est-à-dire regrouper les individus en faisant en sorte que dans un même groupe les individus se ressemblent, que chaque groupe soit suffisamment prévisible mais que les groupes aient des profils relativement différents vis-à-vis de certaines variables, par exemple des groupes plus thermo-sensibles ou plus réceptifs à la tarification. Comme le montre Sevlian and Rajagopal [2018], prévoir la consommation totale de quelques clients (de 1 à 50) est une tâche difficile pour laquelle il semble plus raisonnable de se concentrer sur des agrégats d'au moins 100 ménages. L'étape suivante consiste à ajuster des modèles de prévision pour la consommation de chaque groupe et à additionner dans un second temps les prévisions obtenues pour chaque groupe afin d'obtenir une prévision de la consommation totale. L'intuition derrière une telle approche est que la population peut être divisée en sous-populations ayant des habitudes de consommations différentes qui nécessitent des modèles différents. Dans Quilumba et al. [2015], le clustering est fait en utilisant la méthode $K-$means sur des variables construites à partir des charges (la consommation moyenne sur certaines périodes pertinentes de la journée, la consommation moyenne par jour sur une semaine, les pics de consommation). Les prévisions pour chacun des groupes sont ensuite faites à partir de réseaux de neurones qui sont ensuite sommées pour avoir une prévision de la charge totale. Un autre exemple, plus récent, est celui de Wang et al. [2018] qui propose de construire un ensemble de prévisions basées sur des réseaux neuronaux dont les groupes sont générés à partir d'un partitionnement hiérarchique sur les profils hebdomadaires individuels. La prévision de la consommation électrique totale se fait ensuite, non pas par une somme comme précédemment, mais en ajustant sur l'ensemble de validation une régression linéaire ajustée sur la consommation totale prenant comme variables explicatives les prévisions obtenues pour les groupes.

Une approche différente du schéma classique "Classification puis prévision" des prévisions ascendantes et celle où la classification et la prévision sont faites en une seule étape. Une approche est développée dans Devijver et al. [2019] proposant une méthodologie où la classification et la prévision sont effectuées en une seule étape à partir d'un problème de sélection de modèles dans le cadre des modèles de mélange pour les données fonctionnelles.

En revanche, dans cette approche, seules les courbes de charge sont considérées sans prise en compte des variables exogènes.

1.2.1 Nos contributions

Dans le chapitre 3, nous proposons une extension du schéma standard de la prévision ascendante Classification-Prévision où nous modifions l'agrégation des prévisions en attribuant à chacune des prévisions un poids variant au cours du temps en se basant sur l'agrégation d'experts séquentielle (voir Cesa-Bianchi and Lugosi [2006] pour une présentation globale). Ce type d'agrégation permet de développer des stratégies qui tirent avantage d'experts, c'est-à-dire dans ce contexte de prévisions, très divers et potentiellement en très grand nombre. Dans les stratégies que nous avons développées, le clustering peut reposer sur différents types de variables, que ce soit sur les profils de consommation obtenues à partir des courbes $(Y_{t,i})_{\substack{t\in\mathbb{Z}\\1\leq i\leq n}}$ ou sur des informations exogènes $(I_i)_{1\leq i\leq n}$ comme par exemple la tarification, la classe sociale, ou bien même une partition aléatoire qui permet aussi de créer une grande diversité d'experts. Considérons une suite croissante de partitions $(\mathcal{A}_j)_{1\leq j\leq K}$ où chaque partition $\mathcal{A}_j$ est composée de κ_j groupes pour $j \in \{1,\dots,K\}$ et notons j le niveau associé à la partition $\mathcal{A}_j$. L'objectif fixé étant de prévoir la consommation totale, nous ajoutons une étape avant l'étape de prévision de renormalisation des consommations pour chacun des groupes. Cette étape consiste à faire en sorte, et ce pour chaque partition, que chaque groupe ait une consommation de même ordre de grandeur que la consommation totale. Elle permet ensuite de mélanger librement les différents experts et d'exploiter entièrement les algorithmes d'agrégation convexe, partie la plus prolifique des méthodes d'agrégation. L'étape de prévision est ensuite faite pour chacun des groupes, et ce pour chaque partition, à partir des consommations précédemment renormalisées. Les prévisions sont obtenues à partir des forêts aléatoires introduites dans Breiman [2001] (et sont l'objet d'un approfondissement dans la suite de cette thèse) offrant un bon compromis entre vitesse de calcul, calibration des paramètres et performances.

Nous avons développé trois stratégies pour prévoir la consommation totale. La première est la prévision à partir d'une seule échelle, c'est-à-dire que la prévision se fait uniquement à partir d'une partition $\mathcal{A}_j$ fixée. Contrairement à cette dernière, les deux autres stratégies sont multi-échelles, c'est-à-dire que les stratégies de prévisions considèrent la suite de partitions jusqu'à un certain niveau L donné, $(\mathcal{A}_j)_{1\leq j\leq L}$.

Nous appelons la première stratégie *Single Scale Weighted Aggregation* (abrégée *SSWA*). Fixons une partition $\mathcal{A}_j$. Cette stratégie est l'agrégation des prévisions obtenues sur les consommations renormalisées des κ_j groupes de la partition fixée. Considérons cette fois une suite de partitions $(\mathcal{A}_j)_{1\leq j\leq L}$ pour expliquer les deux stratégies multi-échelles. La *Multi-Scale Weighted Aggregation* (abrégée *MSWA*) consiste à mélanger en une seule agrégation toutes les prévisions obtenues dans la suite de partitions, c'est-à-dire l'agrégation des $\sum_{j=1}^{L}\kappa_j$ prévisions, contrairement à seulement κ_j prévisions dans SSWA pour une partition $\mathcal{A}_j$ donnée. Cependant les prévisions restent calculées de la même manière que pour la stratégie SSWA. La troisième stratégie est une stratégie en deux étapes que nous appelons *Two Step-Multi-scale Weighted Aggregation* (abrégée *2S-MSWA*). La première étape est de calculer pour chaque partition dans la suite $(\mathcal{A}_j)_{1\leq j\leq L}$ une prévision en utilisant la stratégie SSWA présentée ci-dessus. La deuxième étape consiste alors à agréger les L prévisions obtenues dans la première étape.

Dans l'expérience, effectuée sur les consommations d'électricité de résidentiels en Irlande (Commission for Energy Regulation [2011]), nous utilisons pour les différentes agrégations l'algorithme ML-Poly développé dans Gaillard et al. [2014] qui a déjà fait ses preuves

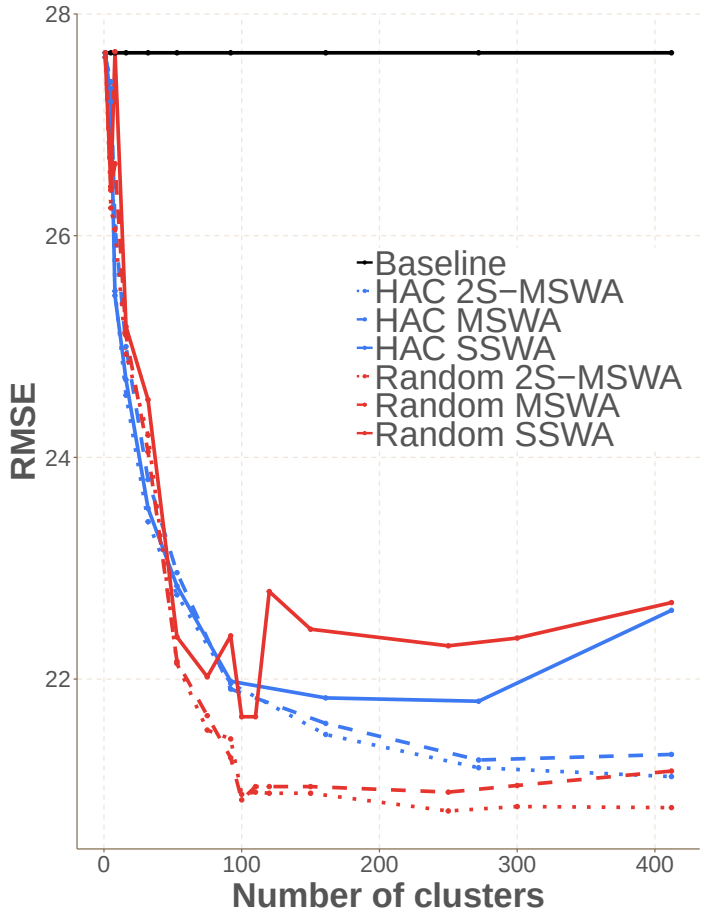

Figure 1.5: Comparaison du RMSE pour les stratégies en fonction du nombre de groupes, clustering hiérarchique *vs* clustering aléatoire.

dans la prévision des prix et de la consommation dans Gaillard and Goude [2015] et Gaillard et al. [2016]. Plusieurs constats sont tirés de cette expérience. Le premier est un gain en performance d'environ 25% sur le RMSE par rapport à un modèle de forêt aléatoire calculé uniquement sur la consommation totale ; les stratégies multi-échelles sont clairement meilleures pour la prévision et ce même en ajoutant des partitions très fines. Nous trouvons en Fig. 1.5 une illustration du gain de performance en fonction du nombre de groupes (pour un nombre de groupes κ_L donné sur le graphique, l'erreur pour une stratégie multi-échelles correspond à la stratégie appliquée sur la suite $(\mathcal{A}_j)_{1\leq j\leq L}$) pour les différentes stratégies. Celles-ci se basent ici, soit sur un clustering hiérarchique à partir de l'information $(I_i)_{1\leq i\leq n}$ (représenté en bleu), soit un clustering aléatoire obtenu (représenté en rouge), pour un nombre de groupes κ_j donné, en attribuant uniformément les n individus dans les κ_j groupes sans remise. Un autre point intéressant (et surprenant) de ce travail est le constat qu'une partition aléatoire des individus semble tout aussi bien, voire mieux, fonctionner pour l'objectif de la prévision de la consommation totale qu'une partition calculée sur la base d'informations sur les individus, que ce soit les courbes de charge ou les informations comme le statut social, le chauffage ou la tarification. Il est cependant intéressant de noter que cette manière d'utiliser les partitions aléatoires s'approche de l'esprit des méthodes d'ensembles, par exemple des forêts aléatoires, qui fonctionnent en générant de la diversité dans les estimateurs en ajoutant de l'aléa, ce qui en fait une approche intéressante à étudier.

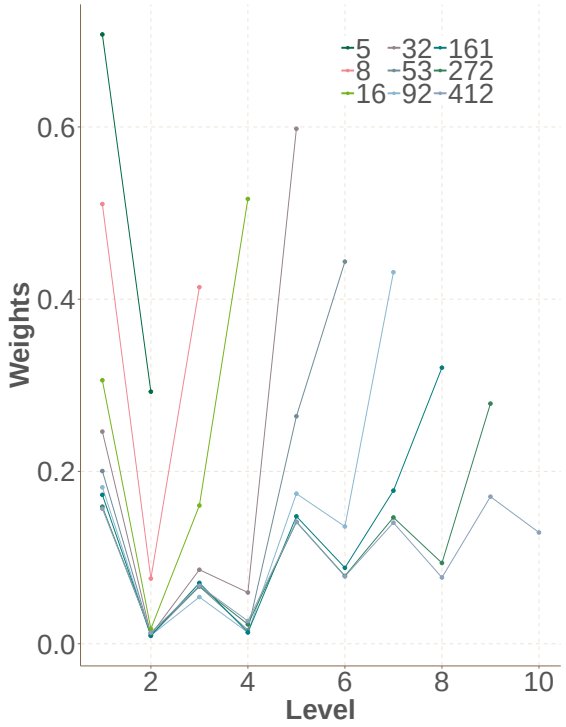

Figure 1.6: Moyenne (sur le temps) des poids en fonction du niveau de la partition pour la stratégie 2S-MSWA Clustering hiérarchique.

Nous avons également étudié les poids des agrégations pour savoir quelles partitions contribuent le plus à la prévision. Cette étude est très difficile au vu du nombre d'experts dans les mélanges, qui peut dépasser le millier. Cependant nous remarquons qu'en moyenne (sur le temps) dans le cas de la stratégie 2S-MSWA, lorsque nous considérons une hiérarchie de partitions basées sur les informations des individus $(I_i)_{1\leq i\leq n}$, plus nous ajoutons des partitions (en plus en plus fines) et plus les poids associés à ces partitions ajoutées semblent être significatifs par rapport aux partitions précédentes avec une structure persistante en forme de V (voir Fig. 1.6). Cette forme est certainement due au fait, étant donné la structure hiérarchique des clusters experts, que la prévision obtenue pour une résolution est proche de la somme des prévisions à l'échelle d'en dessous. Par ailleurs, il semble y avoir y avoir une forme de convergence vers un poids uniforme des poids, c'est-à-dire qu'aucune partition n'est inutile.

Une suite de ce travail était de regarder quelle était l'importance des variables dans les groupes. En effet, l'idée de la prévision ascendante est de décomposer le signal global en groupe ayant des profils différents. Il est donc intéressant de regarder si dans notre étude, les différents groupes ont des comportements différents vis-à-vis de certaines variables comme la température. Une réponse est donnée en appendice A. Tout d'abord, dans le modèle de forêt aléatoire calculé directement pour la prévision de la consommation totale sans étape de clustering, nous observons que les trois variables les plus importantes pour la prévision à un instant t sont la consommation à $t-7$ jours, la consommation à $t-1$ jour et l'instant qui est prévu (l'instant représente la demi-heure de la journée). Parmi les variables de température, la variable de température $t-1$ jour est la plus importante mais bien moins significative pour la prévision que les trois variables précédentes. Lorsque nous regardons ce qu'il se passe pour chaque partition, nous remarquons une tendance inverse de l'importance de la consommation. En effet, pour une partition fixée, nous observons une décroissance mono-

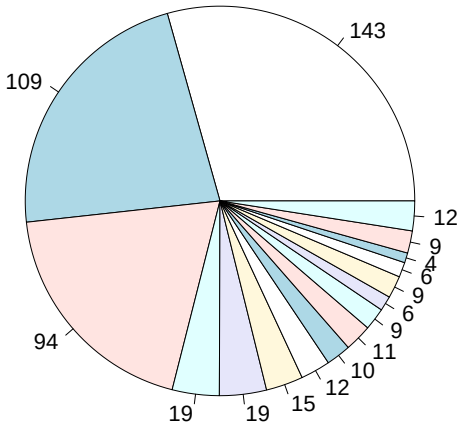

Figure 1.7: Nombre d'individus par groupe pour la partition de 16 groupes obtenu par HAC.

tone de l'importance de la variable de consommation d'il y a une semaine lorsque le cardinal du groupe diminue. Cependant l'importance de la consommation du jour d'avant semble être moins impactée par ce facteur. Nous observons également que les différents lags de température semblent être nettement plus présents que dans le modèle initial sans clustering. Une illustration est donnée pour une partition à 16 groupes avec en Fig. 1.7 une représentation du nombre d'individus par groupe et en Fig. 1.8 et 1.9 les importances des variables normalisées par l'importance maximale dans chaque groupe pour la consommation et la température en fonction du groupe dans la partition. Étant donné que certains groupes sont plus volatiles que d'autres, en particulier lorsque le nombre d'individus est faible, les ordres de grandeur des importances de variables peuvent être très différents et ce même lorsque les consommations sont renormalisées. Tout ceci implique de renormaliser les importances des variables. (Attention, les groupes ne sont pas exactement dans l'ordre décroissant du nombre d'individus).

Dans le cadre du chapitre 3, le but était de prévoir la consommation totale. Dans cette optique, nous avons utilisé différents types d'agrégations. Considérons maintenant uniquement la stratégie à une échelle, la SSWA. Nous utilisons une agrégation des importances de variables de chaque groupe en utilisant les poids obtenus par la stratégie SSWA pour savoir quelles variables contribuent à la prévision de la consommation totale. Il est important de noter que l'importance des variables de chaque groupe est obtenue à partir de modèles conçus pour la prévision de la consommation du groupe et non pas pour la consommation totale. Nous comparons cependant l'importance globale (calculée directement à partir de la consommation totale) et l'importance agrégée, en utilisant les poids optimisés estimés pour la prévision de la consommation totale. En dépit de cette remarque, nous observons finalement en pratique quelque chose de très similaire à ce qu'on obtenait en regardant les importances groupe par groupe. Précisément, plus la partition est fine, moins la variable qui était la plus importante au départ, la consommation passé d'une semaine, est significative.

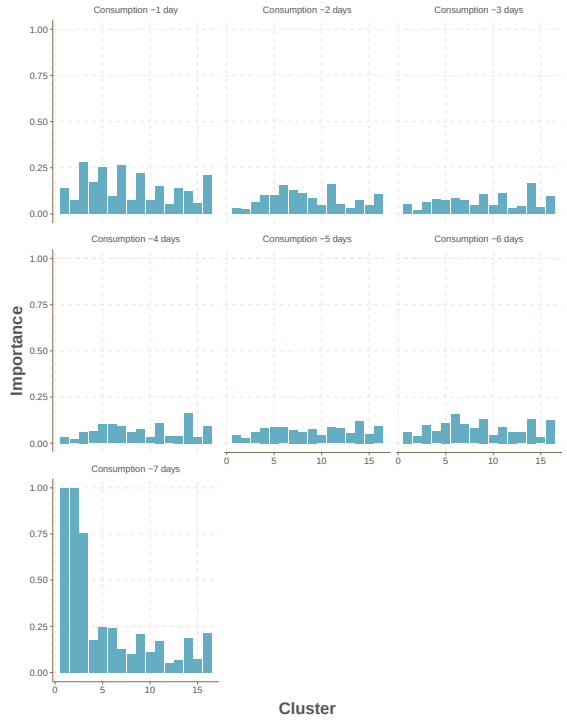

Figure 1.8: Importance des lags de consommation pour une partition à 16 groupes.

Figure 1.9: Importance des lags de température pour une partition à 16 groupes.

Cependant, bien que l'importance des lags de températures croit en allant vers les partitions plus fines, nous retrouvons les variables les plus importantes du modèle de base sans clustering excepté la variable *Instant* qui prend la première place. Ces derniers points sont illustrés en Fig. 1.10 à 1.12.

Dans le schéma de prévision du chapitre 3, deux éléments comptent : le fait de désagréger d'une part et le niveau de désagrégation (combien de groupes et combien de personnes par groupe) d'autre part. Cependant, la classification sur la base de variables exogènes non temporelles (avec ou sans caractéristiques des courbes de charge) s'avère peu compétitive par rapport à un partitionnement au hasard des individus et le choix du nombre de groupes s'avère peu important. Concernant ce dernier, il suffit de prendre un nombre de groupes assez grand à condition que cela soit computationnellement possible. C'est sur cette base que les variantes de l'appendice B se construisent. Contrairement aux stratégies du chapitre 3, nous souhaitons classifier et construire des prévisions simultanément pour ensuite les mélanger afin de prévoir la consommation totale, c'est-à-dire que nous intégrons l'information $(I_i)_{1\leq i\leq n}$ directement dans l'étape de prévision à la place de la partie clustering dans l'espoir d'améliorer la prévision en utilisant l'information non-temporelle qui semblait superficielle jusqu'ici. L'idée centrale de cette méthode repose sur le fait que l'estimateur fasse une classification automatique pendant et pour la prévision à la place d'une classification antérieure basée sur un autre critère.

Fixons une taille de groupe d'individus G et un nombre de groupes B. Une idée pour construire un modèle de prévision, pour un groupe de G individus, est de tirer aléatoirement B groupes en tirant B fois G individus avec remise parmi les n individus. Pour chaque groupe la consommation est sommée et nous avons alors B séries temporelles représentant une consommation d'un groupe de taille G qui est ensuite renormalisée comme dans le chapitre 3. Pour chaque groupe b, nous avons également une synthèse de l'information

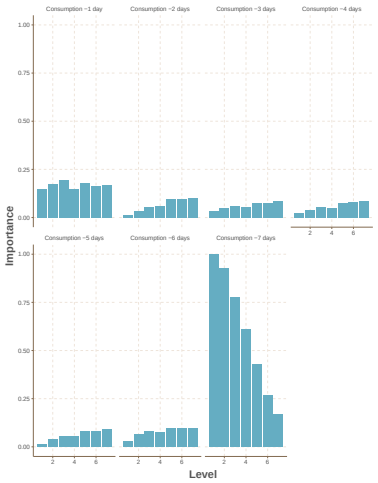

Figure 1.10: Importance des lags de consommation en fonction du niveau de partitionnement.

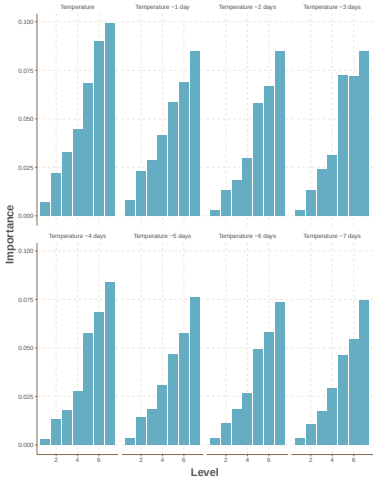

Figure 1.11: Importance des lags de température en fonction du niveau de partitionnement.

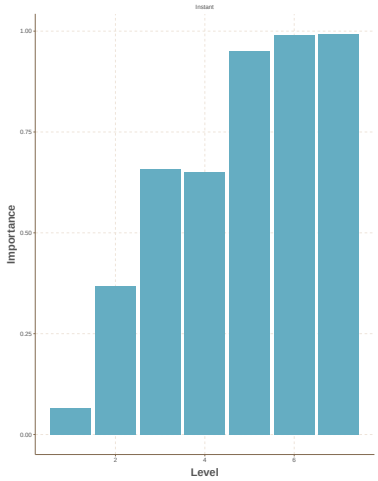

Figure 1.12: Importance de la variable Instant en fonction du niveau de partitionnement.

$\widetilde{I}^{G,b}$ obtenu par l'agrégation de l'information des individus qui sont dans ce groupe. Étant donné que dans notre cas $X_{t,i} = X_t$, nous n'avons pas besoin d'agréger ces données. Les B consommations renormalisées sont alors concaténées avec les données exogènes X_t et les synthèses des informations $\left(\widetilde{I}^{G,b}\right)_{1\leq b\leq B}$. Un estimateur pour la prévision d'un sous-ensemble de la population de taille G est ensuite calculé à partir de la dernière matrice en utilisant une forêt aléatoire. Afin de réduire les temps de calcul et les problèmes de mémoire, nous considérons, pour chaque groupe, un sous-échantillonage temporel au lieu des séries complètes. Cette procédure est ensuite réitérée pour différentes valeurs de G et nous obtenons un ensemble de prévisions pour différents sous-ensembles de la population qui sont ensuite combinées de diverses manières par des agrégations séquentielles.

Les stratégies de ce travail conduisent également à une amélioration de la prévision, comparé à un modèle calculé directement sur la consommation totale, mais ne fournissent pas d'amélioration par rapport aux stratégies du chapitre 3. Cela montre que ces stratégies de prévision en une étape ne permettent pas une intégration de l'information $(I_i)_{1\leq i\leq n}$, pour ce jeu de données en tout cas, pour l'amélioration de la prévision de la consommation totale mais confirment les conclusions obtenues dans le chapitre 3, soit, que la désagrégation

couplée avec une agrégation *experte* sont fondamentales dans l'amélioration de la prévision et qu'il est possible d'améliorer nettement la prévision de la consommation totale en créant des agrégats d'individus de différentes tailles et sans nécessairement avoir d'informations spécifiques aux individus. Nous avons cependant deux nouvelles conclusions et perspectives intéressantes qui émanent de ce travail. Nous remarquons d'abord que l'apprentissage sur une partie des données à partir d'un sous-échantillonnage temporel suffit pour construire nos prévisions ce qui permet un apprentissage plus rapide. Un autre point intéressant est que, contrairement aux stratégies par clustering dans les stratégies du chapitre 3, des modèles de prévision pour différentes échelles fixées en avance sont construits pour prévoir simultanément la consommation totale et la consommation d'un groupe d'individus.

1.3 Cadre statistique et forêts aléatoires

Supposons que nous observons une suite aléatoire stationnaire $(X_t, Y_t)_{t\in\mathbb{Z}} \in \mathcal{X} \times \mathcal{Y}$[1] telle que

$$Y_t = f(X_t) + \epsilon_t$$

où ϵ_t correspond à une erreur, généralement supposée telle que $\mathbb{E}[\epsilon_t | X_t] = 0$. L'objectif est d'estimer la fonction de régression

$$\forall x \in \mathcal{X}, f(x) = \mathbb{E}[Y_t | X_t = x].$$

Nombreuses sont les façons d'estimer f, dans le cadre des séries temporelles, les plus connus étant les modèles ARIMA. Dans le cadre de cette thèse il sera question d'estimer la fonction de régression par une méthode non-paramétrique: les forêts aléatoires.

Nous allons introduire les forêts aléatoires dans le cadre de la régression en supposant que $\mathcal{Y} = \mathbb{R}$ et $\mathcal{X} = \mathbb{R}^p$. Cependant toute la construction peut être étendue au cas de la classification et pour des variables d'entrée catégorielles. Dans le cadre statistique, nous observons uniquement un jeu d'entraînement $\mathcal{D}_n = ((X_1, Y_1), \ldots, (X_n, Y_n)$ utilisé pour la construction de la forêt aléatoire notée $\hat{f}_n$.

Les forêts aléatoires sont une méthode non-paramétrique pour estimer la fonction de régression f. Elles ont été introduites par Breiman en 2001 dans Breiman [2001] et sont depuis une des meilleures méthodes statistiques pour la régression ou la classification Fernández-Delgado et al. [2014]. Cette popularité est due à la performance dans une grande variété d'applications à grande dimension. Elles ont également un avantage en terme computationnel, elles sont facilement parallélisables et ont peu de paramètres à régler. Nous pouvons citer comme applications à succès : la chemo-informatique Svetnik et al. [2003], l'écologie Cutler et al. [2007]; Prasad et al. [2006], la reconnaissance d'objets 3D Shotton et al. [2013] et la prévision de séries temporelles Fischer et al. [2017]; Kane et al. [2014]. Nous référons à Verikas et al. [2011] pour d'autres applications des forêts aléatoires et Biau and Scornet [2016] pour une vue, d'un point de vue théorique, sur les résultats sur celles-ci. Étant utilisées comme référence du fait de sa rapidité d'exécution couplée à de bonnes performances, les forêts aléatoires sont devenues un outil indispensable pour le statisticien.

1.3.1 Construction des forêts aléatoires

Les forêts aléatoires tirent leur origine de deux sources principales, les arbres de régression Breiman et al. [1984] et le bagging, introduit dans Breiman [1996].

[1] Notons qu'ici $(X_t, Y_t)_t$ ne sont pas les mêmes que dans la section précédente.

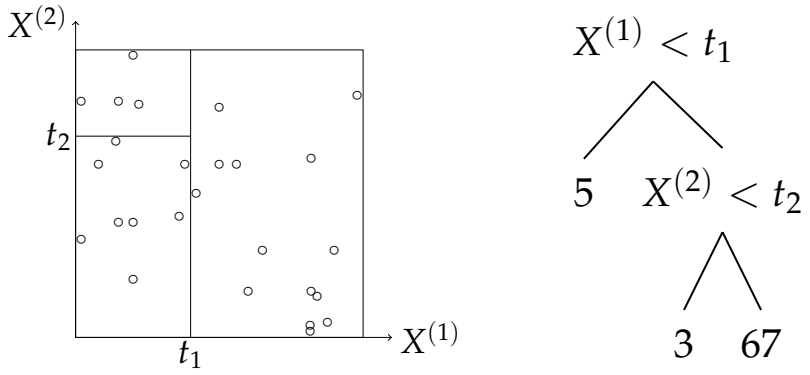

Figure 1.13: Un partitionnement de $[0,1]^2$ et l'arbre binaire associé.

Arbres de régression - CART

Les arbres de régression sont construits par partitionnement récursif de l'espace d'entrée basé sur un certain critère afin d'estimer la fonction de régression f. Un arbre est alors une décomposition par morceaux de l'espace d'entrée. Un arbre binaire peut être associé au partitionnement où chaque noeud de l'arbre correspond à un test binaire. Une illustration d'un partitionnement dans l'espace bidimensionnel et son arbre binaire associé se trouve en Fig. 1.13.

Les arbres de décisions les plus utilisés de nos jours sont les arbres *CART* (pour ***C**lassification **A**nd **R**egression **T**rees*) qui ont été introduits par Breiman et al. [1984]. Nous nous sommes placés dans un cadre de régression mais ces arbres peuvent aussi, comme leur nom l'indique, faire l'objet de prédicteur dans le cadre de classification. Les arbres CART ne sont pas les seuls types d'arbres de décision ayant vu le jour, existent également les arbres *CHAID* introduit dans Kass [1980] ou *C4.5* de Quinlan [2014] par exemple. Dans cette partie, nous nous concentrons sur les arbres CART correspondant aux arbres fondateurs des forêts aléatoires de Breiman.

La racine de l'arbre correspond à tout l'espace d'entrée $\mathcal{X}$ qui contient toutes les observations de l'échantillon $\mathcal{D}_n$. Il s'ensuit le découpage de la racine. Dans le cas où les variables explicatives sont continues, un découpage (appelé *split* en anglais) de la forme suivante est choisi :

$$\left\{X^{(j)} < z\right\} \cup \left\{X^{(j)} \geq z\right\}$$

où $j \in \{1, \dots, p\}$ et $z \in \mathbb{R}$. Cela signifie que les observations avec une valeur plus petite sur la j-ème variable iront dans le noeud fils de gauche, et les autres dans le noeud fils de droite. La méthode CART consiste à trouver le meilleur split, c'est-à-dire le meilleur couple (j,z) qui minimise un certain critère. Dans le cas de régression, ce critère est la variance intra-noeuds suite au split d'un noeud (ou cellule) A en ses deux fils A_L et A_R. Cela se quantifie mathématiquement de la façon suivante. Soit $\mathcal{C}_A$ l'ensemble de toutes les coupes possibles dans la cellule A. Pour chaque couple $(j,z) \in \mathcal{C}_A$, le critère CART empirique est de la forme

$$L_n(j,z) = \frac{1}{N_n(A)} \sum_{i=1, X_i \in A}^{n} (Y_i - \bar{Y}_A)^2 - \frac{1}{N_n(A)} \sum_{i=1, X_i \in A}^{n} (Y_i - \bar{Y}_{A_L} \mathbb{1}_{X_i^{(j)} < z} - \bar{Y}_{A_R} \mathbb{1}_{X_i^{(j)} \geq z})^2, \tag{1.3.1}$$

où $X_i = \left(X_i^{(1)}, \dots, X_i^{(p)}\right)$, $A_L = \left\{x \in A, x^{(j)} < z\right\}$, $A_R = \left\{x \in A, x^{(j)} \geq z\right\}$ et $\bar{Y}_A$ (resp. $\bar{Y}_{A_L}, \bar{Y}_{A_R}$) est la moyenne empirique des Y_i appartenant à la cellule A (resp. A_L, A_R). Le split sera alors effectué sur le couple maximisant $L_n(j,z)$. Cette procédure est ensuite réitérée, de

la même manière, sur chaque nouveau noeud fils jusqu'à ce qu'un critère d'arrêt soit atteint ; un noeud n'est plus découpé lorsque la variance dans ce noeud est nulle. Étant donné que cette condition est très rarement vérifiée dans un noeud avec plus d'une observation, d'autres critères existent comme celui où une cellule doit contenir un nombre d'observations supérieur à un seuil fixé ou alors le nombre de noeuds terminaux doit être inférieur à un certain paramètre fixé.

Une fois l'arbre de décision construit, on associe à chaque noeud "terminal", appelé *feuille* de l'arbre, une valeur constante. Cette valeur dans le cas de régression correspond généralement à la moyenne des Y_i des points qui se trouvent dans cette feuille. Si nous souhaitons prédire un nouveau point x, nous regardons alors dans quelle feuille de l'arbre il tombe, la prédiction $\hat{y}$ sera alors la valeur associée à cette feuille.

Un arbre développé jusqu'au bout (appelé *arbre maximal*), c'est-à-dire que le nombre de feuilles est égal au nombre d'observations, a un biais très faible, mais est doté d'une variance très grande. À l'opposé, un arbre peu profond souffre du problème inverse : une variance faible mais un grand biais. Une manière de résoudre ce problème, hormis les forêts aléatoires, est une deuxième étape généralement indispensable dans les arbres CART : l'élagage. L'élagage consiste à choisir un modèle parmi l'ensemble des sous-arbres élagués de l'arbre maximal. Il s'agit donc de minimiser un critère pénalisé où la pénalité est proportionnelle au nombre de feuilles de l'arbre. Nous ne décrivons pas plus en détail cette étape des arbres CART étant donné que, dans le cadre des forêts aléatoires, cette opération n'est généralement pas réalisée.

Bagging

Le Bagging (qui vient de *Bootstrap Aggregating*) a été introduit par Breiman [1996]. Le principe du bagging est de générer aléatoirement M échantillons $(\mathcal{D}_n(\Theta_1), \ldots, \mathcal{D}_n(\Theta_M))$ de taille α_n où $(\Theta_j)_{1\leq j\leq M}$ sont i.i.d de même loi que Θ, indépendant de $\mathcal{D}_n$. La variable Θ est utilisée en pratique pour rééchantilloner le jeu de données $\mathcal{D}_n$ mais également pour l'introduction d'autres sources d'aléas détaillées par la suite. Une méthode de prédiction (aussi appelée règle de base) $\hat{h}$ est ensuite construite sur chacun des échantillons bootstrap pour avoir une collection de prédicteurs $\left(\hat{h}(\cdot, \mathcal{D}_n(\Theta_1)), \ldots, \hat{h}(\cdot, \mathcal{D}_n(\Theta_M))\right)$ et enfin d'agréger ces prédicteurs. Dans le cas des forêts aléatoires, les méthodes de prédiction seront des arbres de décision mais l'idée est bien plus générale et peut s'appliquer peu importe la règle de base choisie.

La méthode la plus courante pour construire un nouvel échantillon $\mathcal{D}_n(\Theta)$ est de tirer $\alpha_n = n$ observations avec remise dans l'échantillon originel $\mathcal{D}_n$. Une autre façon de faire est de tirer cette fois-ci $\alpha_n < n$ points sans remise parmi les observations de $\mathcal{D}_n$.

Le principe de créer plusieurs échantillons différents et de construire une règle de base sur chacun d'eux permet d'avoir des prédictions différentes et ainsi de les diversifier. Le prédicteur émanant de l'étape d'agrégation sera alors meilleur en termes de performance et de stabilité permettant une réduction de la variance.

Random Forests - Random Input

Nous décrivons maintenant les Random Forests - Random Input (signifiant *forêts aléatoires à variables d'entrées aléatoires* en français et abrégées *RF-RI*) qui ont été introduites dans Breiman [2001] et qui restent à ce jour la variante la plus communément utilisée. L'algorithme est résumé dans algorithm 1. Les forêts étant basées sur le principe du bagging, par construction, chaque prédicteur est construit séparément. Pour expliquer la procédure de la

forêt, il suffit alors d'expliciter la construction d'un arbre de régression.

Pour un échantillon généré $\mathcal{D}_n\left(\Theta_j\right)$ donné, un arbre est construit en utilisant le critère CART rappelé précédemment. Une subtilité des RF-RI est de restreindre à chaque noeud la minimisation du critère sur un sous-ensemble aléatoire de variables m_{try} (un entier entre 1 et p) plutôt que sur les p variables permettant ainsi d'avoir une plus grande diversité dans les prédicteurs par l'introduction d'aléa supplémentaire dans la construction.

Le *jème* arbre aléatoire est défini comme suit

$$\hat{f}_n\left(x;\Theta_j;\mathcal{D}_n\right) = \sum_{i\in\mathcal{D}_n(\Theta_j)} \frac{\mathbb{1}_{X_i\in A_n(x;\Theta_j;\mathcal{D}_n)}Y_i}{N_n\left(x;\Theta_j;\mathcal{D}_n\right)}\mathbb{1}_{E_n(X,\Theta_j)}.$$

où $A_n(x,\Theta_j,\mathcal{D}_n)$ correspond à la cellule contenant le point x,

$$N_n\left(x;\Theta_j;\mathcal{D}_n\right) = \sum_{i=1}^{n}\mathbb{1}_{X_i\in A_n\left(x,\Theta_j,\mathcal{D}_n\right)}$$

et $E_n(x,\Theta_j)$ l'événement défini par $\left\{N_n(x,\Theta_j)\neq 0\right\}$. Cela signifie que la prédiction pour le *jème* arbre aléatoire pour un nouveau point x est la moyenne des Y_i dont les X_i correspondant tombent dans la cellule $A_n\left(x,\Theta\right)$. La prédiction d'une nouvelle observation x de la forêt aléatoire, dans le cas de régression, est alors donnée en prenant la moyenne empirique des prédictions en x de chaque arbre :

$$\hat{f}_{M,n}\left(x;\Theta_1,\ldots,\Theta_M;\mathcal{D}_n\right) = \frac{1}{M}\sum_{j=1}^{M}\hat{f}_n\left(x,\Theta_j,\mathcal{D}_n\right). \tag{1.3.2}$$

Une remarque sur le paramètre m_{try}. Cette idée de prendre un sous-ensemble aléatoire pour découper un noeud fut introduit dans le cadre de problèmes de reconnaissance d'image par Amit and Geman [1997]. Nous retrouvons l'estimateur du bagging décrit précédemment en considérant $m_{try} = p$ en utilisant pour règle de base les arbres CART non élagués. Cependant il est clair que lorsque $m_{try} < p$ les deux variantes sont très différentes par l'introduction d'une perturbation de plus sur le nombre de variables en compétition à un noeud. Le nombre de variables explicatives p pouvant être très grand, prendre $m_{try} << p$ permet alors de réduire significativement le nombre de possibilités et la complexité de calcul à chaque noeud mais peut également améliorer grandement les performances par rapport au bagging classique pour un bon choix de m_{try}, les performances des forêts aléatoires étant basées sur la diversité de ses arbres. Cependant ce dernier paramètre ne peut pas être pris trop petit, en particulier dans les problèmes en grande dimension, où une trop grande perturbation dans les arbres dégraderaient beaucoup les performances de la forêt aléatoire.

Cette version des forêts aléatoires est celle généralement utilisée dans les applications. Cependant, due à la forte dépendance entre les données et la construction, il est très difficile d'en faire une analyse théorique. Des variantes ont ainsi été développées, plus faciles à définir et calculer dans le but d'obtenir des résultats pour une compréhension plus profonde des RF-RI, incorporant certains aspects clés des forêts aléatoires originelles comme la réduction de la variance ou la sélection de variables adaptative. L'autre variante des forêts aléatoires qui fait l'objet d'une étude dans cette thèse sont les forêts aléatoires centrées introduites dans Breiman [2004].

Forêts aléatoires centrées

Une des premières variantes des forêts aléatoires sur lesquelles des garanties théoriques ont vu le jour sont les forêts aléatoires centrées introduites par Breiman [2004]. Les forêts aléatoires centrées font parties de la famille des forêts aléatoires pures (*Purely random forests* en

données : jeu d'entraînement $((X_1, Y_1), \ldots, (X_n, Y_n))$
paramètres : nombre d'arbres M, nombre d'observations par arbre a_n, nombre de variables pré-sélectionnées pour le split m_{try}, nombres de feuilles τ_n
for $j \leftarrow 1$ *to* M **do**
Construire le $j$ème arbre :

- Tirer uniformément $a_n \leq n$ observations sans remplacement.
- Soit $n_{nodes} = 1$.
- **while** $n_{nodes} < \tau_n$ **do**
  - Choisir un noeud sans enfant A, contenant plus d'une observation.
  - Tirer uniformément (sans remplacement), l'ensemble $Mtry \subset \{1, \ldots, p\}$ tel que $|Mtry| = m_{try}$.
  - Sélectionner le meilleur split dans la cellule A en maximisant le critère CART, définie en eq. (1.3.1), sur $Mtry$.
  - Découper la cellule A selon le meilleur split. Soient A_R et A_L les cellules obtenues.
  - $n_{nodes} = n_{nodes} + 1$.

  end

end
prédiction pour une nouvelle observation x : moyenne des M prédictions données par les arbres pour x.

Algorithm 1: Random forest - random input.

anglais) où la construction des arbres se fait indépendamment de l'échantillon d'entraînement $\mathcal{D}_n$.

La première différence par rapport aux RF-RI est qu'il n'y a pas d'étape de ré-échantillonage. Un arbre est ensuite construit de manière récursive de la façon suivante. Une variable est uniformément choisie ou selon une probabilité indépendante des données à chaque noeud et le split est effectué au milieu de la cellule de la variable choisie au préalable. La procédure est ensuite réitérée $k \in \mathbb{N}^\star$ fois. L'algorithme est résumé dans algorithm 2.

données : jeu d'entraînement $((X_1, Y_1), \ldots, (X_n, Y_n))$
paramètre : τ_n
Répéter récursivement $\log_2 \tau_n$ fois :

- À chaque noeud, sélectionner une direction $j \in \{1, \ldots, p\}$ avec probabilité $p_{n,j} \in (0,1)$ tel que $\sum_{j=1}^{p} p_{n,j} = 1$;
- Le split est effectué au milieu de la cellule selon la direction sélectionnée.

Algorithm 2: Forêt aléatoire centrée.

À noter que $\tau_n \geq 2$ est un paramètre déterministe fixé qui peut dépendre de n mais pas de l'échantillon $\mathcal{D}_n$ et que chaque arbre a exactement $2^{\lceil \log_2 \tau_n \rceil} \approx \tau_n$ feuilles.

Les notations introduites précédemment pour les RF-RI restent valables pour les forêts aléatoires centrées.

Garanties théoriques

Les propriétés théoriques sont généralement déduites sur la forêt aléatoire infinie obtenue en prenant la limite de eq. (1.3.2) lorsque le nombre d'arbres M tend vers l'infini. La loi des grands nombres justifie alors d'utiliser

$$\hat{f}_n(X, \mathcal{D}_n) = \mathbb{E}_\Theta \left[\hat{f}_n(X, \Theta, \mathcal{D}_n) \right]$$

à la place de $\hat{f}_{M,n}\left(x; \Theta_1, \ldots, \Theta_M; \mathcal{D}_n\right)$, où $\mathbb{E}_\Theta$ est l'espérance par rapport à Θ conditionnellement à X et $\mathcal{D}_n$. Pour des raisons de lisibilité dans la suite, nous supprimons la dépendance en $\mathcal{D}_n$ et notons $\hat{f}_n(X) := \hat{f}_n(X, \mathcal{D}_n)$.

Les RF-RI ont fait l'objet d'une attention croissante ces dernières années en ce qui concerne l'analyse théorique et nous faisons ici un résumé de certains travaux académiques traitant de ce sujet. Supposons que les observations $(X_i, Y_i)_{1 \leq i \leq n}$ sont indépendantes et identiquement distribuées comme (X, Y). Un lien entre l'erreur des forêts finies et l'erreur des forêts infinies a été établi dans Scornet [2016] et montre que la différence des erreurs peut être arbitrairement proche de zéro à condition que le nombre d'arbres dans la forêt aléatoire finie soit assez grand. Plus précisément,

$$\mathbb{E}\left[\hat{f}_{M,n}(X) - f(X)\right]^2 - \mathbb{E}\left[\hat{f}_n(X) - f(X)\right]^2 \leq \frac{8\left(\|f\|_\infty + \sigma^2\left(1 + 4\log n\right)\right)}{M}$$

lorsque ϵ suit une loi normale centrée de variance finie σ^2 et indépendante de X. Une conséquence de ce résultat est que les forêts aléatoires finies sont consistantes à condition que les forêts aléatoires infinies le soient et que $\frac{\log n}{M} \underset{n \to \infty}{\longrightarrow} 0$. Un résultat de normalité asymptotique est prouvé dans Mentch and Hooker [2016] pour les forêts aléatoires où les points sont sous-échantillonnés sans remplacement et à condition que la taille du sous-échantillon α_n grandit plus lentement que $\sqrt{n}$, c'est-à-dire qu'à condition que $\frac{\alpha_n}{\sqrt{n}} \underset{n \to \infty}{\longrightarrow} 0$, et que le nombre d'arbres M grandit avec n, c'est-à-dire que $\frac{n}{M} \underset{n \to \infty}{\longrightarrow} C$ pour une constante positive C. Cependant, cela

n'implique pas nécessairement que les forêts soient sans biais asymptomatiquement. Ce dernier point a été résolu dans Wager and Athey [2018] qui établit également la consistance de l'infinitesimal jackknife pour estimer la variance de la forêt sous la condition moins restrictive que la taille du sous-échantillon satisfait $\frac{a_n \log n^p}{n} \underset{n\to\infty}{\longrightarrow} 0$. Le travail que nous avons réalisé repose sur le résultat théorique établit dans Scornet et al. [2015] dans lequel la consistance de la version élaguée (c'est-à-dire que la profondeur d'un arbre est contrôlée par un paramètre $\tau_n < a_n$) des RF-RI est établie, i.e. $\mathbb{E}\left[\hat{f}_n(X) - f(X)\right]^2 \to 0$ lorsque $n \to +\infty$, pour les arbres où les points sont sous-échantillonnées sans remplacement et que la fonction de régression f est additive (nous notons cette hypothèse **H3a** dans la suite), c'est-à-dire de la forme

$$f(X) = \sum_{j=1}^{p} f_j\left(X^{(j)}\right) + \epsilon$$

où X est uniforme sur $[0,1]^p$, ϵ suit une loi normale centrée de variance finie $\sigma^2 > 0$ indépendante de X et où chaque f_j est continue. Les forêts aléatoires sont alors consistantes du moment que $a_n \to \infty, \tau_n \to \infty$ et que $\frac{\tau_n (\log(a_n))^9}{a_n} \to 0$. Ce résultat est toujours vrai pour la version non-élaguée des RF-RI (c'est-à-dire, les arbres sont maximaux, $\tau_n = a_n$), au prix d'une conjecture portant sur la faible dépendance des arbres individuels en l'échantillon d'apprentissage, plus difficile à vérifier en général.

Un autre travail repose sur les forêts aléatoires centrées pour lesquelles il est possible d'aller plus loin, en développant des vitesses de convergence. En se plaçant dans le cadre parcimonieux, c'est-à-dire que la fonction de régression f qui est, à la base, une fonction d'un $p-$uplet ne dépend uniquement que d'un sous-ensemble $\mathcal{S}$ (supposé non vide) des p variables. Cela signifie que l'ensemble restant $\left\{x^{(1)}, \ldots, x^{(p)}\right\} \setminus \mathcal{S}$ n'a aucune influence sur la fonction de régression. En posant $X_{\mathcal{S}} = \left(X^{(j)}, j \in \mathcal{S}\right)$, cela se traduit par

$$f(x) = \mathbb{E}\left[Y | X_{\mathcal{S}} = x_{\mathcal{S}}\right].$$

Supposant que les observations $((X_1, Y_1), \ldots, (X_n, Y_n))$ sont i.i.d, Biau [2012] établit que si la probabilité $p_{n,j}$ de split selon la $j$ème direction tend vers $\frac{1}{S}$ (où $S = Card\,(\mathcal{S})$), si la $j$ème variable appartient à l'ensemble $\mathcal{S}$, et que la fonction de régression est Lipschitz, alors

$$\mathbb{E}\left[\hat{f}_n(X) - f(X)\right]^2 = \mathcal{O}\left(n^{\frac{-1}{|\mathcal{S}|(4/3)\log 2 + 1}}\right).$$

Cela montre que la vitesse de convergence des forêts aléatoires vers la fonction de régression f dépend uniquement du nombre de variables $|\mathcal{S}|$ "fortes" et non pas de la dimension de base p ce qui pourrait également expliquer pourquoi les forêts aléatoires fonctionnent très bien dans le cadre de la grande dimension. Cela entraîne que les forêts aléatoires centrées s'adapte au cadre parcimonieux à condition que l'hypothèse que la procédure arrivent à sélectionner les variables informatives soit bien vérifiée. Dans le même papier, Biau propose une procédure afin que les probabilités vérifient en pratique cette hypothèse. Pour cela, le critère CART est calculé, à chaque noeud, sur un sous-ensemble aléatoire des variables d'un deuxième échantillon $\mathcal{D}_n'$. La variable sélectionnée pour le découpage sera alors choisie aléatoirement parmi les variables qui réalisent les meilleurs splits.

Une question survenant dans Biau [2012] est si cette vitesse peut être améliorée. Récemment, Klusowski [2018] répond par la positive et obtient

$$\mathbb{E}\left[\hat{f}_n(X) - f(X)\right]^2 = \mathcal{O}\left(\left(n\sqrt{\log^{S-1} n}\right)^{-a_{\mathcal{S}}}\right)$$

où

$$\alpha_{\mathcal{S}} = \frac{2\log\left(1 - S^{-1}/2\right)}{2\log\left(1 - S^{-1}/2\right) - \log 2}.$$

Ce nouveau résultat entraîne une amélioration dans l'exposant de $\frac{1}{|\mathcal{S}|(4/3)\log 2+1}$ à $\frac{1}{|\mathcal{S}|\log 2+1}$. Par ailleurs, il est également montré que cette nouvelle borne n'est pas améliorable sans autres hypothèses. Le travail présenté dans le chapitre 2 repose cependant sur les travaux de Biau [2012] et non pas sur les plus récents de Klusowski [2018].

1.3.2 Dépendance

Il existe de nombreuses façons de modéliser la dépendance dans les observations. Nous nous sommes intéressés à un type de dépendance appelé la dépendance $\beta-$mélangeante définit de la façon suivante pour un processus $(W_t)_{t\in\mathbb{Z}}$.

Définition 1.3.1 (Processes $\beta-$mélangeant)**.** Soient $\sigma_l = \sigma(W_1^l)$ et $\sigma'_{l+m} = \sigma(W_{l+m}^{\infty})$ les sigma-algèbres des événements générées par les variables aléatoires $W_1^l = (W_1, \dots W_l)$ et $W_{l+m}^{\infty} = (W_{l+m}, W_{l+m+1}, \dots)$. Le coefficient de β-mélange est donné par

$$\beta_m = \sup_{l\geq 1} \mathbb{E}\left[\sup_{B\in\sigma'_{l+m}} |\mathbb{P}(B|\sigma_l) - \mathbb{P}(B)|\right]$$

où l'espérance est prise par rapport à σ_l.

Un processus stochastique est dit absolument régulier, ou β-mélangeant, si

$$\lim_{m\to\infty} \beta_m = 0.$$

Les coefficients $\beta-$mélanges les plus courants sont connus sous le nom de mélange algébrique et exponentiel défini comme suit,

1. Mélange algébrique : $\beta_m = \mathcal{O}(m^{-r_\beta})$ for $r_\beta > 0$.
2. Mélange exponentiel : $\beta_m = \mathcal{O}(\exp(-bm^{k_\beta}))$ for $b, k_\beta > 0$.

La propriété de $\beta-$mélange est intéressante dans le cadre théorique puisque de nombreuses propriétés statistiques sont préservées sous cette condition et sont faciles à manipuler. Une méthode pour manipuler ces processus s'appuie sur un lemme de Yu [1994] qui approxime un processus dépendant par une suite de blocs d'observations indépendants plus un terme dépendant linéairement du coefficient β. Nous référons au chapitre 2 ainsi que Dedecker et al. [2007] et Rio [2013] pour plus d'informations sur les processus dépendants.

1.3.3 Nos contributions

Dans de nombreuses applications, par exemple dans la prédiction des séries temporelles Dudek [2015]; Fischer et al. [2017]; Kane et al. [2014]; Lahouar and Ben Hadj Slama [2015]; Moon et al. [2018], ainsi que les travaux présentés en chapitre 3, appendix A, appendix B, les forêts aléatoires sont utilisées sur les données comme si ces dernières étaient indépendantes. Les résultats dans ces cas appliqués sont les mêmes que dans les autres cas : les forêts aléatoires donnent d'excellentes performances mais la question qui se pose est, cela est-il théoriquement justifié ? La question est, est-ce que la consistance des forêts aléatoires est

conservée lorsque l'hypothèse d'indépendance des observations est supprimée. Les modèles étant jusqu'ici étudiés dans le cadre de données indépendantes, nous répondons à cette question dans le chapitre 2.

Concernant les forêts aléatoires originelles, nous nous sommes uniquement intéressés au cas des arbres élagués, c'est-à-dire $\tau_n < a_n$. Nous nous plaçons dans le cadre où les observations ont une dépendance faible, et où, plus précisément, les observations vérifient l'hypothèse de β-mélange présentée en section 1.3.2, et les hypothèses de stationnarité et d'ergodicité. Sous ces conditions, nous obtenons, tout en gardant les hypothèses du cas i.i.d, que les forêts aléatoires sont consistances dès lors que la dépendance entre les observations n'est pas trop longue, plus précisément quantifié dans le théorème suivant. Soit a_n une suite d'entiers tel que $1 \leq a_n \leq n$.

Théorème 1.3.1. *Si*

- *la séquence* $(X_i, Y_i)_{1\leq i\leq n} \in [0,1]^p \times \mathbb{R}$ *est stationnaire ergodique et* β*-mélangeant;*
- *les erreurs* $(\epsilon_i)_{1\leq i\leq n}$ *sont indépendantes;*
- *l'hypothèse* ***H3a*** *est satisfaite;*
- $\frac{\tau_n \log(a_n)^9 a_n}{a_n} \underset{n\to\infty}{\longrightarrow} 0$;
- $\frac{\log(a_n)^4 \beta_{a_n} a_n}{a_n} \underset{n\to\infty}{\longrightarrow} 0$.

Alors les RF-RI sont consistantes, i.e.

$$\mathbb{E}\left[\hat{f}_n(X) - f(X)\right]^2 \underset{n\to\infty}{\longrightarrow} 0.$$

Une analyse de ce théorème est donnée en chapitre 2, en particulier en quantifiant de manière précise les contraintes sur le paramètre de profondeur de l'arbre τ_n selon les formes que le coefficient de dépendance puisse prendre, par exemple algébrique ou exponentielle. La conclusion à retenir étant que plus la dépendance entre les observations est forte, moins les arbres doivent être développés, comparé au cas des arbres où les observations seraient indépendantes et uniquement distribuées pour garantir la consistance de la forêt.

La deuxième contribution sur les forêts aléatoires concerne cette fois les forêts aléatoires centrées. La question est similaire à la précédente dans le cas des RF-RI, que se passe-t-il lorsqu'on enlève l'hypothèse d'indépendance des données. Une réponse est apportée dans le même chapitre 2, À noter que ce résultat s'appuie sur le résultat de Biau [2012] et ne possède donc pas l'optimalité proposée par Klusowski [2018]. Nous avons besoin pour cela besoin des deux hypothèses suivantes afin d'établir les résultats :

- **H1b**: la séquence $(X_i, Y_i)_{1\leq i\leq n} \in [0,1]^p \times \mathbb{R}$ est stationnaire β-mélangeant;
- **H2b**: les erreurs $\epsilon_i := Y_i - f(X_i)$ sont indépendantes de variance finie $\sigma^2 > 0$.

Soit a_n une suite d'entiers tel que $1 \leq a_n \leq n$. Sans faire d'hypothèses sur la forme du coefficients de $\beta-$mélange, nous obtenons le résultat suivant.

Théorème 1.3.2. *Supposons les hypothèses d'observations stationnaires* $\beta-$*mélangeants* ***H1b****, des erreurs indépendantes* ***H2b****, X est uniforme sur* $[0,1]^p$ *et que* f^* *est L-Lipschitz sur* $[0,1]^S$*. Supposons que les probabilités de choix des directions sont telles que* $p_{n,j} = \frac{1}{S}\left(1 + \nu_{n,j}\right)$ *pour* $j \in \mathcal{S}$*, alors*

$$\mathbb{E}\left[\hat{f}_n(X) - f(X)\right]^2 \leq \frac{2SL^2 a_n}{\tau_n^{\frac{0,75}{S\log 2}(1+\gamma_n)}} + \mathcal{C}_{1,n}\frac{\tau_n a_n^2}{n} + \mathcal{C}_2\frac{\beta_{a_n} n}{a_n}$$

où

$$C = \frac{576}{\pi}\left(\frac{\pi \log 2}{16}\right)^{S/2p},$$

$$1 + \nu_n = \prod_{j \in \mathcal{S}} \left[\left(1 + \nu_{n,j}\right)^{-1} \left(1 - \frac{\nu_{n,j}}{S-1}\right)^{-1} \right]^{1/2p},$$

$$\mathcal{C}_{1,n} = 4e^{-1} \sup_{x \in [0,1]^p} f^2(x) + C\sigma^2 \left(\frac{S^2}{S-1}\right)^{S/2p} (1 + \nu_n)$$

et

$$\mathcal{C}_2 = SL^2 + \sigma^2 + \sup_{x \in [0,1]^p} f^2(x).$$

Le théorème et les hypothèses sont discutés plus en détails dans le chapitre 2. La conclusion principale de ce travail est, comme dans le cas des RF-RI, que les arbres doivent être moins profonds pour les observations dépendantes que si elles étaient indépendantes et identiquement distribuées. Un point intéressant à souligner néanmoins ; en supposant de plus que le coefficient de $\beta-$mélange est d'une certaine forme, il est possible d'aller plus loin en optimisant le paramètre de profondeur de l'arbre τ_n afin d'obtenir le résultat suivant qui quantifie de manière plus claire les vitesses de convergence selon la forme de la dépendance.

Corollaire 1.3.1. *Si les hypothèses précédentes sont vérifiées, le choix optimal pour τ_n s'écrit*

$$\tau_n \propto \left(\frac{n}{a_n}\right)^{\frac{S \log 2}{0.75 + S \log 2}}.$$

En remplaçant dans la vitesse de convergence précédente, nous obtenons

$$\mathbb{E}\left[\hat{f}_n(X) - f(X)\right]^2 = \mathcal{O}\left(\left(\frac{a_n^{1.5 + S \log 2}}{n^{0.75}}\right)^{\frac{1}{0.75 + S \log 2}}\right) + \mathcal{O}\left(\frac{\beta_{a_n} n}{a_n}\right). \tag{1.3.3}$$

Supposons que r_β et k_β sont connus.

1. *Sous la condition de mélange algébrique; eq. (1.3.3) est minimisé en prenant*

$$a_n \propto n^{\frac{1.5 + S \log 2}{2.25 + 2S \log 2 + r_\beta (0.75 + S \log 2)}}.$$

Le paramètre τ_n est alors de la forme

$$\tau_n \propto n^{\frac{(1 + r_\beta) S \log 2}{2.25 + 2S \log 2 + r_\beta (0.75 + S \log 2)}}$$

et obtenons la vitesse de convergence suivante :

$$\mathbb{E}\left[\hat{f}_n(X) - f(X)\right]^2 = \mathcal{O}\left(n^{\frac{-0.75 r_\beta + 0.75 + S \log 2}{r_\beta (0.75 + S \log 2) + 2.25 + 2S \log 2}}\right).$$

2. *Sous la condition de mélange exponentiel ; prenant*

$$a_n \propto \log n^{\frac{1}{k_\beta}}$$

nous obtenons

$$\tau_n \propto \left(\frac{n}{\log n^{\frac{1}{k_\beta}}} \right)^{\frac{S \log 2}{0.75 + S \log 2}}.$$

En remplaçant dans eq. (1.3.3) *nous obtenons*

$$\mathbb{E}\left[\hat{f}_n(X) - f(X)\right]^2 = \mathcal{O}\left(\left(\frac{\log n^{\frac{1.5 + S \log 2}{k_\beta}}}{n^{0.75}} \right)^{\frac{1}{0.75 + S \log 2}} \right).$$

1.3.4 Vers un nouvel algorithme

Les théorèmes énoncés précédemment du chapitre 2 dans le cas d'observations $\beta-$mélangeants reposent sur une technique de preuve par bloc. Le processus est décomposé en bloc de taille fixe qui sont ensuite approximés par des copies de blocs indépendants en utilisant un lemme de Yu [1994]. Rappelons également que dans les forêts aléatoires, l'étape de bootstrap est celle qui décide de quels points sont choisis pour la construction d'un arbre. Le bootstrap standard Efron [1979] est celui où a_n points sont tirés parmi les n observations avec remise. Ce bootstrap a pour objectif de répliquer la distribution de l'échantillon $\mathcal{D}_n$ et est adapté au cas où les observations sont indépendantes et identiquement distribuées. Cependant lorsque l'hypothèse d'indépendance n'est plus vérifiée, comme dans le cas des séries temporelles, ce bootstrap a pour conséquence de détruire la structure des données et les arbres sont construits comme si les données étaient indépendantes. Un simple exemple de ce phénomène est donné à partir de la consommation représentait en Fig. 1.14 avec en Fig. 1.15a un bootstrap classique et en Fig. 1.15b un *moving block bootstrap*, qui sera rappelé dans le chapitre 4, avec des blocs de longueur de 24 heures. Nous remarquons clairement que la nouvelle série obtenue à partir d'un bootstrap standard ne possède aucune structure contrairement au bootstrap par blocs où nous observons des motifs semblables à la série temporelle originelle. Bien que les forêts aléatoires donnent d'excellents résultats sur des données réelles comme dans Dudek [2015]; Fischer et al. [2017]; Kane et al. [2014]; Lahouar and Ben Hadj Slama [2015]; Moon et al. [2018], nous nous sommes posés la question si nous pouvions garder la structure de dépendance sous-jacente pour la construction des arbres et ainsi permettre une amélioration des performances. Cela a conduit au travail présenté dans le chapitre 4.

L'idée de faire un bootstrap par blocs pour la prévision de série temporelles n'est pas nouvelle. Cordeiro and Neves [2009] utilise un *sieve bootstrap* pour effectuer le bagging avec des modèles de lissage exponentiel. Ils utilisent le lissage exponentiel pour décomposer les données, puis adaptent un modèle autorégressif aux résidus et génèrent de nouveaux résidus à partir de ce processus autorégressif. Enfin, ils ajustent le modèle de lissage exponentiel qui était utilisé pour la décomposition à toutes les séries bootstrap. Un autre travail est de Bergmeir et al. [2016] qui propose une méthode de bagging comme suit. Après avoir appliqué une transformation Box-Cox aux données, la série est décomposée en trois composantes, la tendance, la saisonnalité et les résidus. Les résidus sont ensuite bootstrappés utilisant le *moving block bootstrap*. La tendance et les composantes saisonnières sont ensuite rajoutées et la transformation Box-Cox est inversée. Pour chacune de ces séries chronologiques bootstrappées, un modèle est ensuite choisi parmi plusieurs modèles de lissage exponentiel, en utilisant l'AIC corrigé du biais. Les prévisions ponctuelles sont calculées à l'aide de tous les différents modèles et les prévisions résultantes sont agrégées à l'aide de la médiane.

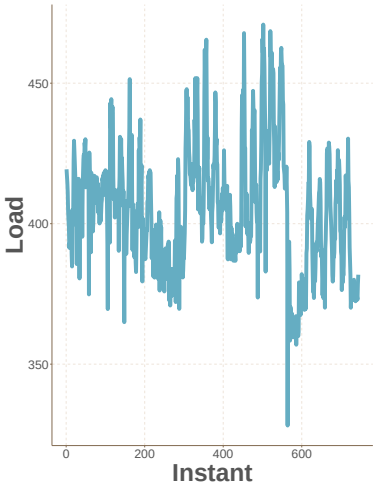

Figure 1.14: Série originelle.

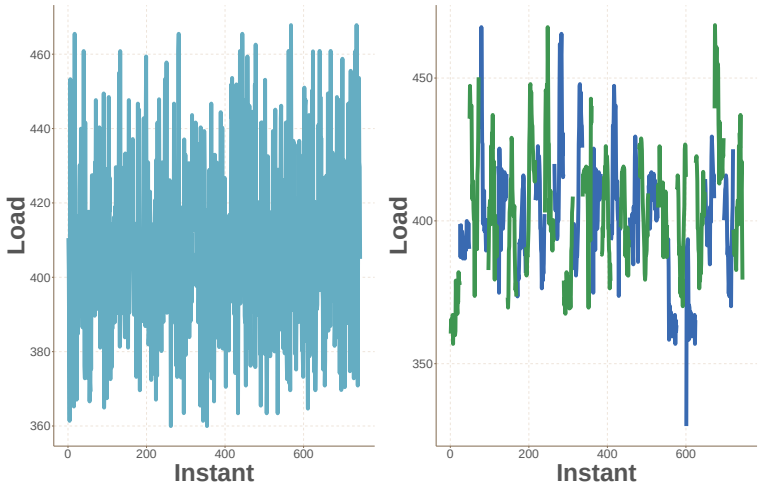

Figure 1.15: Bootstrap : a standard, et b par bloc de taille 24h.

Les travaux cités précédemment de Cordeiro and Neves [2009] et Bergmeir et al. [2016] se basent sur la décomposition préliminaire du signal en certaines composantes dont les résidus sont ensuite bootstrappés en utilisant soit un *sieve bootstrap* ou un *moving block bootstrap*. Dans le travail présenté dans le chapitre 4, cette décomposition du signal n'est pas considérée. Nous proposons de directement remplacer la partie bootstrap standard des forêts aléatoires, première étape pour la construction de chaque arbre, par un bootstrap par blocs qui permet de conserver la structure dépendante des données et ainsi espérer des gains de performances dans le cas des séries temporelles par exemple. Le nouvel algorithme est présenté en algorithm 3 avec en bleu la nouveauté.

Nous avons considéré trois variantes de bootstrap par blocs, dont les constructions sont détaillés dans le chapitre 4, le *circular block bootstrap* Politis and Romano [1992], le *moving block bootstrap* Kunsch [1989]; Liu and Singh [1992], et le *non-overlapping block bootstrap* Carlstein [1986] mais d'autres variantes peuvent être facilement incorporées. Nous avons implémenté une extension du package *ranger* de Wright and Ziegler [2017] incorporant ces nouvelles variantes, nommée *rangerts* et présentée en appendix C. Nous observons des améliorations sur plusieurs jeux de données qui justifient le changement de cette étape clé des forêts aléatoires lorsque les données sont dépendantes. L'exemple considéré dans le chapitre 4 est la prévision de la consommation à une horizon de 24 heures sur un jeu de données issu de

données : jeu d'entrainement $((X_1, Y_1), \ldots, (X_n, Y_n))$
paramètres : $M, \alpha_n, m_{try}, l_n$
for $j \leftarrow 1$ *to* M **do**
Construire le $j$ème arbre :

- Tirer $\alpha_n \leq n$ observations en utilisant un bootstrap par blocs de paramètre l_n.
- Répéter récursivement les étapes suivantes:
  - À chaque noeud A, tirer uniformément (sans remplacement) m_{try} variables.
  - Sélectionner le meilleur split à l'aide du critère CART parmi les variables choisies précédemment.
  - Découper selon le split choisi en deux noeuds A_L et A_R.

end
prédiction pour une nouvelle observation x : moyenne des M prédictions données par les arbres pour x.

Algorithm 3: Forêt aléatoire pour séries temporelles.

Miller and Meggers [2017] dont la charge horaire du mois de janvier 2015 est représentée en Fig. 1.14. Nous retrouvons en Fig. 1.16 un boxplot du RMSE pour les forêts aléatoires standards ainsi que pour chaque variante pour cette objectif. Ces résultats sont obtenus en optimisant, sur un jeu de validation, le paramètre de longueur de blocs l_n. Nous observons un gain moyen allant jusqu'à 11% par rapport aux forêts aléatoires standards uniquement en modifiant la première étape de construction de l'algorithme original.

L'importance des variables peut également être redéfinie pour prendre en compte la structure dépendante. Au lieu de faire des permutations des observations individuelles out-of-bag (les observations qui ne sont pas considérées pour construire un arbre), nous proposons d'uniquement permuter des blocs d'observations de la même taille que durant la construction, tout en gardant la structure interne fixe dans chaque bloc. Cela se calcule immédiatement pour la variante *non-overlapping block bootstrap*. En effet, la construction d'un nouvel échantillon à partir de cette dernière variante repose sur un tirage aléatoire sur des blocs d'observations (et non pas sur les observations individuellement), pré-définis qui ne se chevauchent pas, avec remise. L'out-of-bag est alors composé des blocs qui ne sont pas sélectionnés précédemment et sont de même taille. À la différence de la variante précédemment citée, les deux autres variantes incorporent une autre forme d'aléa engendrant des blocs pouvant se chevaucher. Une modification des observations out-of-bag est alors nécessaire afin de calculer la nouvelle version d'importance des variables. Une comparaison entre l'importance des variables par le calcul de la permutation standard et celle par le calcul de la permutation par bloc sur l'exemple précédent est donné en Figs. 1.17a and 1.17b pour des blocs de 24 heures. La différence la plus significative observée est ici pour la variable *Hour* indiquant l'heure prévue. Étant donné que les blocs sont de taille 24 heures et commencent toujours au même moment (par la construction de la variante *non-overlapping block bootstrap*), la permutation par blocs ne va pas changer l'erreur out-of-bag puisque les blocs sont permutés par des copies identiques pour cette variable. Cela entraîne une importance nulle pour cette dernière par cette procédure.

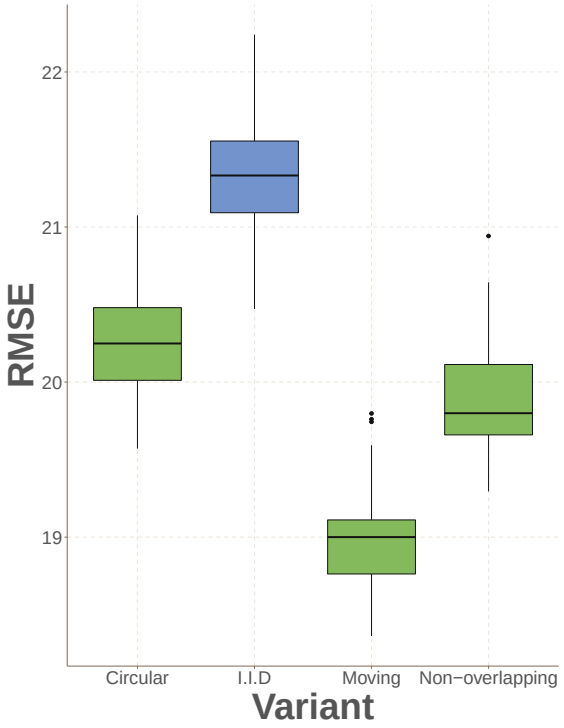

Figure 1.16: Performances des différentes variantes en prenant $m_{try} = 2$ sur 50 tirages.

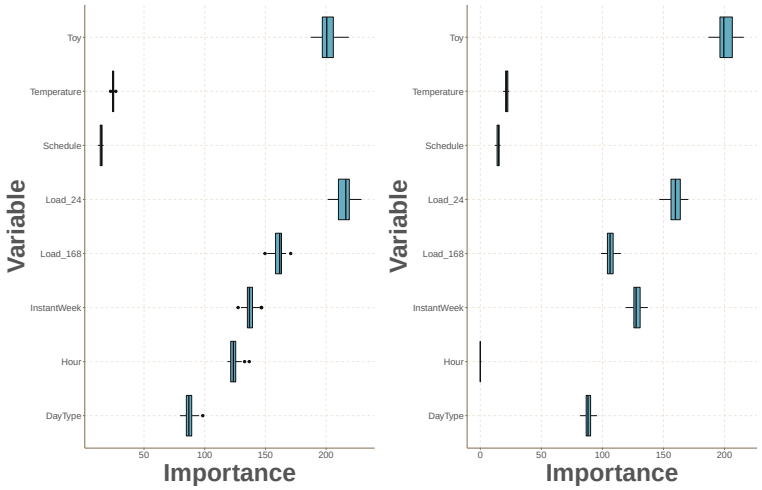

Figure 1.17: Importance des variables basée sur la variante non-overlapping : a permutation standard, et b permutation par blocs de 24h.

1.4 Bibliographie

Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588, 1997. 16

A. Antoniadis, X. Brossat, J. Cugliari, and J.-M. Poggi. Prévision d'un processus à valeurs fonctionnelles en présence de non stationnarités. Application à la consommation d'électricité. *Journal de la Société Française de Statistique*, 153:52 – 78, 2012. 4, 5

A. Antoniadis, X. Brossat, J. Cugliari, and J.-M. Poggi. Une approche fonctionnelle pour la prévision non-paramétrique de la consommation d'électricité. *Journal de la Société Française de Statistique*, 155:202 – 219, 2014. 4, 5

C. Bergmeir, R. J. Hyndman, and J. M. Benítez. Bagging exponential smoothing methods using stl decomposition and box–cox transformation. *International journal of forecasting*, 32:303–312, 2016. 23, 24

G. Biau. Analysis of a random forests model. *Journal of Machine Learning Research*, 13:1063–1095, 2012. 19, 20, 21

G. Biau and E. Scornet. A random forest guided tour. *Test*, 25:197–227, 2016. 13

L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996. 13, 15

L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001. 7, 13, 15

L. Breiman. Consistency for a simple model of random forests. Technical report, 2004. 16

L. Breiman, J. Friedman, C. Stone, and R. Olshen. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984. 13, 14

E. Carlstein. The use of subseries values for estimating the variance of a general statistic from a stationary sequence. *Ann. Statist.*, 14:1171–1179, 1986. 24

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006. ISBN 0521841089. 7

Commission for Energy Regulation. Electricity smart metering customer behaviour trials (cbt) findings report. http://www.cer.ie/docs/000340/cer11080(a)(i).pdf, 2011. 2, 7

C. Cordeiro and M. Neves. Forecasting time series with boot.expos procedure. *REVSTAT-Statistical Journal*, 7:135–149, 2009. 23, 24

J. Cugliari, Y. Goude, and J.-M. Poggi. Disaggregated electricity forecasting using wavelet-based clustering of individual consumers. In *Energy Conference (ENERGYCON), 2016 IEEE International*, pages 1–6. IEEE, 2016. 4

D. R. Cutler, T. C. Edwards, K. H. Beard, A. Cutler, K. T. Hess, J. Gibson, and J. J. Lawler. Random forests for classification in ecology. *Ecology*, 88:2783–2792, 2007. 13

J. Dedecker, P. Doukhan, G. Lang, L. R. J. Rafael, S. Louhichi, and C. Prieur. Weak dependence. In *Weak Dependence: With Examples and Applications*, pages 9–20. Springer, 2007. 20

E. Devijver. *Modèles de mélange pour la régression en grande dimension, application aux données fonctionnelles*. PhD thesis, 2015. Thèse de doctorat dirigée par Massart, P. et Poggi, J.-M. Mathématiques Paris 11 2015. 5

E. Devijver, Y. Goude, and J.-M. Poggi. Clustering electricity consumers using high dimensional regression mixture models. *Applied Stochastic Models in Business and Industry*, pages 1–19, 2019. 6

G. Dudek. Short-term load forecasting using random forests. In *Intelligent Systems'2014*, volume 323, pages 821–828, Cham, 2015. Springer International Publishing. 20, 23

B. Efron. Bootstrap methods: Another look at the jackknife. *Ann. Statist.*, 7:1–26, 1979. 23

M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, 15:3133–3181, 2014. 13

A. Fischer, L. Montuelle, M. Mougeot, and D. Picard. Statistical learning for wind power: a modeling and stability study towards forecasting. *Wind Energy*, 20:2037–2047, 2017. 13, 20, 23

D. Fischer, J. Scherer, A. Flunk, N. Kreifels, K. Byskov-Lindberg, and B. Wille-Haussmann. Impact of hp, chp, pv and evs on households' electric load profiles. In *2015 IEEE Eindhoven PowerTech*, pages 1–6, 2015. 4

P. Gaillard. *Contributions à l'agrégation séquentielle robuste d'experts : Travaux sur l'erreur d'approximation et la prévision en loi. Applications à la prévision pour les marchés de l'énergie.* PhD thesis, 2015. Thèse de doctorat dirigée par Stoltz, G. Mathématiques Paris 11 2015. 5

P. Gaillard and Y. Goude. Forecasting electricity consumption by aggregating experts; how to design a good set of experts. In *Modeling and stochastic learning for forecasting in high dimensions*, volume 217, pages 95–115. Springer, 2015. 8

P. Gaillard, G. Stoltz, and T. van Erven. A second-order bound with excess losses. In *Proceedings of The 27th Conference on Learning Theory*, volume 35 of *Proceedings of Machine Learning Research*, pages 176–196. PMLR, 2014. 7

P. Gaillard, Y. Goude, and R. Nedellec. Additive models and robust aggregation for gefcom2014 probabilistic electric load and electricity price forecasting. *International Journal of forecasting*, 32:1038–1050, 2016. 8

Y. Goude. *Mélange de prédicteurs : application à la prévision de consommation d'électricité*. PhD thesis, 2008. Thèse de doctorat dirigée par Dacunha-Castelle, D. Mathématiques Paris 11 2008. 5

Y. Goude, R. Nedellec, and N. Kong. Local Short and Middle term Electricity Load Forecasting with semi-parametric additive models. *IEEE transactions on smart grid*, 5:440 – 446, 2013. 5

T. Hastie and R. Tibshirani. Generalized additive models. *Statist. Sci.*, 1:297–310, 1986. 5

M. J. Kane, N. Price, M. Scotch, and P. Rabinowitz. Comparison of arima and random forest time series models for prediction of avian influenza h5n1 outbreaks. *BMC bioinformatics*, 15:276, 2014. 13, 20, 23

G. V. Kass. An exploratory technique for investigating large quantities of categorical data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 29:119–127, 1980. 14

H. Kikusato, K. Mori, S. Yoshizawa, Y. Fujimoto, H. Asano, Y. Hayashi, A. Kawashima, S. Inagaki, and T. Suzuki. Electric vehicle charge–discharge management for utilization of photovoltaic by coordination between home and grid energy management systems. *IEEE Transactions on Smart Grid*, 10:3186–3197, 2019. 4

J. M. Klusowski. Complete analysis of a random forest model. *arXiv preprint arXiv:1805.02587*, 2018. 19, 20, 21

H. R. Kunsch. The jackknife and the bootstrap for general stationary observations. *Ann. Statist.*, 17:1217–1241, 1989. 24

A. Lahouar and J. Ben Hadj Slama. Random forests model for one day ahead load forecasting. In *IREC2015 The Sixth International Renewable Energy Congress*, pages 1–6, 2015. 20, 23

F. Lezama, J. Soares, P. Hernandez-Leal, M. Kaisers, T. Pinto, and Z. M. A. do Vale. Local energy markets: Paving the path towards fully transactive energy systems. *IEEE Transactions on Power Systems*, pages 1–1, 2018. 4

R. Y. Liu and K. Singh. Moving blocks jackknife and bootstrap capture weak dependence. In *Exploring the limits of bootstrap (East Lansing, MI, 1990)*, Wiley Ser. Probab. Math. Statist. Probab. Math. Statist., pages 225–248. Wiley, New York, 1992. 24

L. Mentch and G. Hooker. Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *Journal of Machine Learning Research*, 17:1–41, 2016. 18

C. Miller and F. Meggers. The building data genome project: An open, public data set from non-residential building electrical meters. *Energy Procedia*, 122:439 – 444, 2017. 25

M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi. Analyse en ondelettes de la consommation électrique en vue de l'agrégation et de la désagrégation de courbes pour la prévision de la consommation, 2005. 4

M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi. Classification par ondelettes de courbes pour la prévision de la consommation, 2006a. 4

M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi. Ondelec 05-06 : Classification par ondelettes de courbes pour la prévision de la consommation, 2006b. 4

M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi. Optimisation, pilotée par la prévisibilité, de partitions pour la prévision par désagrégation de la courbe de charge, 2007a. 4

M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi. Analyse de l'apport des ondelettes pour la prévision des séries chronologiques dans le contexte électrique, 2007b. 4

M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi. Prévision par désagrégation : variantes, interprétation et effet de la taille des données, 2008. 4

M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi. Stratégie divisive pour la prévision par désagrégation - parallélisation et effet de la taille des données, 2009. 4

M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi. Optimized Clusters for Disaggregated Electricity Load Forecasting. *REVSTAT – Statistical Journal*, 8:105 – 124, 2010. 4

J. Moon, Y. Kim, M. Son, and E. Hwang. Hybrid short-term load forecasting scheme using random forest and multilayer perceptron. *Energies*, 11:3283, 2018. 20, 23

A. Pierrot and Y. Goude. Short-term electricity load forecasting with generalized additive models. *Proceedings of ISAP power*, 2011, 2011. 5

D. N. Politis and J. P. Romano. A circular block-resampling procedure for stationary data. In *Exploring the limits of bootstrap (East Lansing, MI, 1990)*, Wiley Ser. Probab. Math. Statist. Probab. Math. Statist., pages 263–270. Wiley, New York, 1992. 24

A. M. Prasad, L. R. Iverson, and A. Liaw. Newer classification and regression tree techniques: bagging and random forests for ecological prediction. *Ecosystems*, 9:181–199, 2006. 13

F. L. Quilumba, W.-J. Lee, H. Huang, D. Y. Wang, and R. L. Szabados. Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities. *IEEE Trans. Smart Grid*, 6:911–918, 2015. 6

J. R. Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014. 14

E. Rio. Inequalities and limit theorems for weakly dependent sequences. Lecture: cel-00867106, 2013. 20

E. Scornet. On the asymptotics of random forests. *Journal of Multivariate Analysis*, 146:72–83, 2016. Special Issue on Statistical Models and Methods for High or Infinite Dimensional Spaces. 18

E. Scornet, G. Biau, and J.-P. Vert. Consistency of random forests. *The Annals of Statistics*, 43: 1716–1741, 2015. 19

R. Sevlian and R. Rajagopal. A scaling law for short term load forecasting on varying levels of aggregation. *International Journal of Electrical Power & Energy Systems*, 98:350 – 361, 2018. 6

H. Shareef, M. S. Ahmed, A. Mohamed, and E. A. Hassan. Review on home energy management system considering demand responses, smart technologies, and intelligent controllers. *IEEE Access*, 6:24498–24509, 2018. 4

J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56:116–124, 2013. 13

V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston. Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43:1947–1958, 2003. 13

V. Thouvenot. *Estimation et sélection pour les modèles additifs et application à la prévision de la consommation électrique*. PhD thesis, 2015. Thèse de doctorat dirigée par Poggi, J.-M. et Antoniadis A. Mathématiques appliquées Paris Saclay 2015. 5

A. Verikas, A. Gelzinis, and M. Bacauskiene. Mining data with random forests: A survey and results of new tests. *Pattern recognition*, 44:330–349, 2011. 13

S. Wager and S. Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113:1228–1242, 2018. 19

Y. Wang, Q. Chen, C. Kang, Q. Xia, and M. Luo. Sparse and redundant representation-based smart meter data compression and pattern extraction. *IEEE Transactions on Power Systems*, 32:2142–2151, 2017. 4

Y. Wang, Q. Chen, T. Hong, and C. Kang. Review of smart meter data analytics: Applications, methodologies, and challenges. *IEEE Transactions on Smart Grid*, 10:1–1, 2018. 6

M. Wright and A. Ziegler. ranger: A fast implementation of random forests for high dimensional data in c++ and r. *Journal of Statistical Software, Articles*, 77:1–17, 2017. 24

B. Yu. Rates of convergence for empirical processes of stationary mixing sequences. *The Annals of Probability*, 22:94–116, 1994. 20, 23

Chapter 2

Random forests for time-dependent processes: theoretical results

Abstract

Random forests were introduced by Breiman in 2001. We study theoretical aspects of both original Breiman's random forests and a simplified version, the centred random forests. Under the independent and identically distributed hypothesis, Scornet, Biau and Vert proved the consistency of Breiman's random forest, while Biau studied the simplified version and obtained a rate of convergence in the sparse case. However, the i.i.d hypothesis is generally not satisfied for example when dealing with time series. We extend the previous results to the case where observations are weakly dependent, more precisely when the sequences are stationary $\beta-$mixing.

Sommaire

2.1 Introduction

Random forests were introduced in 2001 by Breiman in Breiman [2001] and are since then extremely successful as a regression and classification method. The popularity comes from the wide range of applications in which they are used and the accuracy they offer in high-dimensional problems. They are also easy to implement, can be easily parallelizable and require only few tuning parameters. We can cite as successful applications: chemo-informatics Svetnik et al. [2003], ecology Cutler et al. [2007]; Prasad et al. [2006], 3D object recognition Shotton et al. [2013] and time series prediction Fischer et al. [2017]; Kane et al. [2014].

Let a stationary random sequence $(X_t, Y_t)_{t\in\mathbb{Z}} \in \mathbb{R}^p \times \mathbb{R}$ be such that

$$Y_t = f(X_t) + \epsilon_t \tag{2.1.1}$$

and the error ϵ_t is such that $\mathbb{E}[\epsilon_t | X_t] = 0$. The purpose of random forests is to estimate the regression function

$$\forall x \in \mathbb{R}^p, f(x) = \mathbb{E}[Y_t \mid X_t = x].$$

In the statistical context we only observe a training sample $\mathcal{D}_n = ((X_1, Y_1), \ldots, (X_n, Y_n))$ used to build the random forest estimator denoted by $\hat{f}_n$.

Random forests can be related to two main sources, regression trees Breiman et al. [1984] and bagging Breiman [1996]. Regression trees are constructed by a recursive partitioning of the input space based on some criterion, dependent or independent of the data (we detail precisely two in the following), to estimate the regression function f. At each step of the tree construction, a split is selected (a variable and a location on the variable) based on the evaluation of the criterion among all the admissible splits based on all the variables. The cell is cut in two on the selected split and the previous step is reiterated on the new cells. A tree is then a piecewise constant decomposition of the input space. We can associated to the input space partitioning a binary tree where each node corresponds to a test matching how the input space was cut. An illustration is given in Fig. 2.1 of a partitioning in the two-dimensional space and its associated binary tree. The principle of bagging (short form of *bootstrap aggregating*) is to create M randomly generated training sets by randomly sampling α_n observations with or without replacement from the set $\mathcal{D}_n$ and to construct on each set a predictor. Once the predictors are constructed, the bagging prediction for a new observation x is an aggregation, generally the empirical mean, of the predictions given by the M predictors for the point x. This procedure aims to improve stability and accuracy of the base predictor. In the context of random forests, the predictors are regression trees.

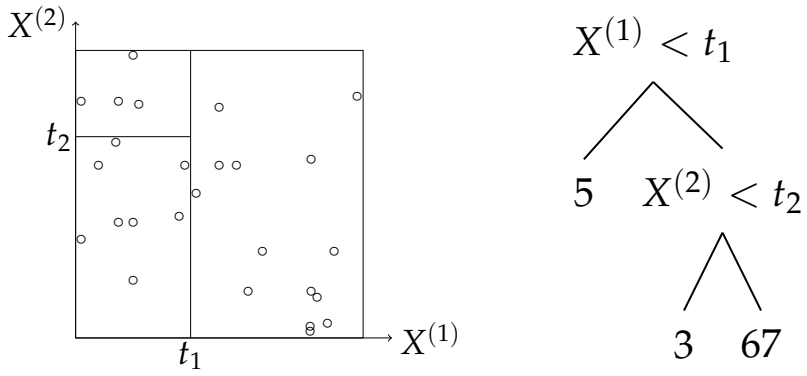

Figure 2.1: A partitioning of $[0,1]^2$ and the associated binary tree.

We study two variants of random forests, the random forest-random input and the centred forest. By construction of the bagging, each predictor is computed in the same way. In order to explain the different procedures we then have to explicit the construction of one predictor. Let us begin with the variant which remains to this day the most commonly used and referred to as the original Breiman's random forest, the random forest-random input (RF-RI). For a given generated training set of a_n points, a tree is computed using the CART Breiman et al. [1984] criterion: at each node of the tree the best split is selected by minimising the intra-node variance. This criterion is detailed in section 2.2.1. A subtlety of the RF-RI is to restrict at each node the minimisation of the criterion on a random subset of m_{try} variables rather than on the p variables and thereby increase the diversity of the predictors by adding randomness in the construction. This is then recursively repeated until a stopping criterion is met, typically when the number of nodes reached a given number or when the number of observations in each node is below a given threshold.

The RF-RI have received increasing attention in recent years regarding theoretical analysis and we can cite for example the works described in Mentch and Hooker [2016]; Scornet [2016]; Scornet et al. [2015]; Wager and Athey [2018]. Since notations are only set later on for ease of readability, we decide to develop these results in the section 2.2.1. With the exception of the result in Scornet et al. [2015] on which the present work relies on and doesn't require additional notations. Assuming that the observations $(X_i, Y_i)_{1\leq i\leq n}$ are independent and identically distributed as (X, Y), they establish the consistency of the pruned version (that is, the depth of the trees is controlled by a parameter) of the RF-RI, i.e. that $\mathbb{E}\left[\hat{f}_n(X) - f(X)\right]^2 \to 0$ as $n \to +\infty$, for trees where points are selected without replacement and the regression function is an additive model. Under an additional assumption, yet hard to verify in general, they also established consistency of the unprunned version (that is, the depth of a tree is not controlled) which is almost the algorithm commonly used in practice.

The second variant of random forests we study belongs to the so-called *purely random forests*'s family. The RF-RI is based on the CART criterion which is heavily data-dependent, while the purely random forests are based on criteria which are independent of the data. The variant we consider is called *centred forest* which was introduced in Breiman [2004]. The first difference with the RF-RI is that there is no re-sampling step, meaning that the set used to computed the trees is $\mathcal{D}_n$. A tree is then recursively constructed as follows. At each node, a coordinate is chosen uniformly or according to some probability independent of the data and the split is performed in the middle of the cell along the selected coordinate. This kind of variants has been preferred for statistical analysis since they are easier to define, provide non-asymptotic risk bound giving insight in the choice of the parameters of the forest but also capture some attractive features of the original random forest as the variance reduction by randomisation and adaptive variable selection. Under the hypothesis that $(X_i, Y_i)_{1\leq i\leq n}$ are i.i.d, Biau [2012] established that if the splits concentrate on the relevant variables then the procedure adapts to sparsity by giving a rate of convergence which depends on the number of strong features. We refer to Biau and Scornet [2016] for a complete theoretical survey on random forests.

The aforementioned theoretical results are established under the condition that the observations are independent and identically distributed. However, in applications, it is very common to have dependent data instead of independent one such as in time series and random forests are proven to perform well on these kind of observations. We may cite as an example of successful applications of random forests in time series Dudek [2015]; Fischer et al. [2017]; Kane et al. [2014]; Lahouar and Ben Hadj Slama [2015]. In this regard, many algorithms were studied in the case of weakly dependent observations, and in particular,

when dealing with $\beta-$mixing sequences. The $\beta-$mixing provide some kind of measure of how the dependence between observations decreases as the distance between them increases. It is usually difficult to estimate the mixing rates in practice. However, $\beta-$mixing sequences can be theoretically well-studied and estimated for various classes of random processes as Gaussian or Markov processes. We refer to Dedecker et al. [2007] and Rio [2013] for more details about dependent processes. The general problem of one-step ahead predicting of time series was considered in Meir [2000] when the time series satisfies $\beta-$mixing and stationary condition, establishing consistency and rates of convergence for a certain class of functions which complexity and memory are determined by the data and minimising the structural risk. Consistency and a rate of convergence are also established for the boosting algorithm in Lozano et al. [2014] when the observations are stationary $\beta-$mixing. Their rate of convergence has an additional term, we also find in our analysis, which can be viewed as a penalty when considering $\beta-$mixing sequences instead of independent observations, $\mathcal{O}\left(n^{1-a(r_\beta+1)}\right)$ with $a \in [0,1)$ and where r_β measures the dependence of the mixing sequence we precise later on.

The paper is organised as follows: we first formalise the models studied and then set the statistical framework together with the notion of $\beta-$mixing sequences. We then state our contribution, including the extension of the aforementioned results to the case where observations are weakly dependent, namely the consistency of the RF-RI when trees are not fully grown and the rate of convergence of centred random forests. The proofs are postponed to the appendices for ease of readability.

2.2 Models

In this section, we formalise the previous mentioned models, namely the RF-RI and the centred random forest.

Recall that a random forest (either RF-RI or simpler models) is a collection of M random trees, computed in the same way, and the trees are constructed from a recursive partitioning of the input space $\mathcal{X}$ to which a binary tree can be associated matching how the input space was cut. We denote for the *jth* random tree, the predicted value at the point x, $\hat{f}_n\left(x;\Theta_j;\mathcal{D}_n\right)$ where $(\Theta_1,\ldots,\Theta_M)$ are independent and identically distributed as Θ and independent of $\mathcal{D}_n$. The random variable Θ is defined later on depending on the variant. The *jth* random tree is defined as follows

$$\hat{f}_n\left(x;\Theta_j;\mathcal{D}_n\right) = \sum_{i\in\mathcal{D}_n(\Theta_j)} \frac{\mathbb{1}_{X_i\in A_n(x;\Theta_j;\mathcal{D}_n)}Y_i}{N_n\left(x;\Theta_j;\mathcal{D}_n\right)}\mathbb{1}_{E_n(X,\Theta_j)}$$

where $\mathcal{D}_n\left(\Theta_j\right)$ is the data set which can be dependent on the random variable Θ_j for example if re-sampling or sub-sampling is used to construct the *jth* tree. The cell containing the point x is denoted $A_n(x,\Theta_j,\mathcal{D}_n)$,

$$N_n\left(x;\Theta_j;\mathcal{D}_n\right) = \sum_{i=1}^{n}\mathbb{1}_{X_i\in A_n\left(x,\Theta_j,\mathcal{D}_n\right)} = \#\left\{i\in\{1,\ldots,n\}, X_i\in A_n(x,\Theta_j,\mathcal{D}_n)\right\}$$

and $E_n(x,\Theta_j)$ the event defined by $\left\{N_n(x,\Theta_j)\neq 0\right\}$. This means that each random tree outputs for a new point x the average value over all Y_i for which the corresponding X_i fall into the cell $A_n\left(x,\Theta\right)$ of the random partition.

In the regression case, we aggregate the predictions by taking the average in the following way to get the random forest estimator

$$\hat{f}_{M,n}\left(x;\Theta_1,\ldots,\Theta_M;\mathcal{D}_n\right)=\frac{1}{M}\sum_{j=1}^{M}\hat{f}_n\left(x,\Theta_j,\mathcal{D}_n\right). \tag{2.2.1}$$

Since M can be chosen as large as possible in practice, we study the properties of the infinite random forest estimate which is obtained as the limit of eq. (2.2.1) when the number of trees M grows to infinity. The law of large numbers then justifies using

$$\hat{f}_n(X,\mathcal{D}_n)=\mathbb{E}_\Theta\left[\hat{f}_n(X,\Theta,\mathcal{D}_n)\right]$$

instead of $\hat{f}_{M,n}\left(x;\Theta_1,\ldots,\Theta_M;\mathcal{D}_n\right)$, where $\mathbb{E}_\Theta$ denotes expectation with respect to Θ conditionally to X and $\mathcal{D}_n$. In the following, to ease legibility we omit the dependency on $\mathcal{D}_n$ and denote simply $\hat{f}_n(X):=\hat{f}_n(X,\mathcal{D}_n)$.

2.2.1 Random forest - random input

We begin by recalling the variant of random forest which is the most commonly used in practice, the random forest-random input. We denote:

- $\alpha_n\in\{1,\ldots,n\}$ the number of sampled data points in each tree;
- $m_{try}\in\{1,\ldots,p\}$ the preselected number of variables for splitting;
- $\tau_n\in\{1,\ldots,\alpha_n\}$ the number of leaves in each tree.

Here we consider the stopping criterion where the number of leaves must not exceed the given parameter τ_n. The random forest is then computed as detailed in algorithm 4.

The CART criterion is defined as follows. Let $\mathcal{C}_A$ be the set of all possible cuts in the cell A. For any $(j,z)\in\mathcal{C}_A$, the CART-split criterion takes the form

$$L_n(j,z)=\frac{1}{N_n(A)}\sum_{i=1,X_i\in A}^{n}(Y_i-\bar{Y}_A)^2-\frac{1}{N_n(A)}\sum_{i=1,X_i\in A}^{n}(Y_i-\bar{Y}_{A_L}\mathbb{1}_{X_i^{(j)}<z}-\bar{Y}_{A_R}\mathbb{1}_{X_i^{(j)}\geq z})^2, \tag{2.2.2}$$

with $X_i=\left(X_i^{(1)},\ldots,X_i^{(p)}\right)$, $A_L=\left\{x\in A,x^{(j)}<z\right\}$, $A_R=\left\{x\in A,x^{(j)}\geq z\right\}$ and $\bar{Y}_A$ (resp. $\bar{Y}_{A_L},\bar{Y}_{A_R}$) is the average of the Y_i's belonging to A (resp. A_L,A_R).

Let us suppose that the observations $(X_i,Y_i)_{1\leq i\leq n}$ are independent and identically distributed as (X,Y). A link between the error of the finite and infinite forest is established in Scornet [2016] and shows that the error of the finite forest can be made arbitrary close to the infinite one providing the number of trees is large enough,

$$\mathbb{E}\left[\hat{f}_{M,n}(X)-f(X)\right]^2-\mathbb{E}\left[\hat{f}_n(X)-f(X)\right]^2\leq\frac{8\left(\|f\|_\infty+\sigma^2\left(1+4\log n\right)\right)}{M}$$

when ϵ is a centred Gaussian noise with finite variance $\sigma^2>0$ and independent of X. Another consequence of this result is that as soon as infinite random forests are consistent then the finite random forests are consistent provided that $\frac{\log n}{M}\underset{n\to\infty}{\longrightarrow}0$. Asymptotic normality of random forests based on subsampling was proven in Mentch and Hooker [2016] when the subsample size α_n grows slower than $\sqrt{n}$, i.e. that $\frac{\alpha_n}{\sqrt{n}}\underset{n\to\infty}{\longrightarrow}0$ and that the number of trees M

input: Training set $((X_1, Y_1), \ldots, (X_n, Y_n))$
parameters: number of trees M, number of observations per tree a_n, size of preselected variables for spitting m_{try}, number of leaves τ_n
for $j \leftarrow 1$ *to* M **do**
Construct the jth tree:

- Draw uniformly $a_n \leq n$ points without replacement.
- Set $n_{nodes} = 1$.
- **while** $n_{nodes} < \tau_n$ **do**
  - Choose a childless node A, containing more than one observation.
  - Select uniformly (without replacement), the set $Mtry \subset \{1, \ldots, p\}$ such that $|Mtry| = m_{try}$.
  - Choose the best split in the cell A maximising the CART criterion, defined in eq. (2.2.2), on $Mtry$.
  - Cut the cell A according to the best split. Let A_R and A_L be the cells we obtain.
  - $n_{nodes} = n_{nodes} + 1$.

end

end
output for a new observation x: mean of the M predictions given by the trees for x.

Algorithm 4: Random forest - random input.

varies with n, i.e. that $\frac{n}{M} \underset{n\to\infty}{\longrightarrow} C$ for some constant $C > 0$. However, this does not necessarily imply that random forests are asymptotically unbiased. This gap was filled in Wager and Athey [2018] and also established that the infinitesimal jackknife consistently estimates the forest variance under the less restrictive condition that the subsample size grows such that $\frac{a_n \log n^p}{n} \underset{n\to\infty}{\longrightarrow} 0$.

2.2.2 Centred forest

We now recall the construction of the centred random forest introduced in Breiman [2004], detailed in algorithm 5.

Data:$((X_1, Y_1), \ldots, (X_n, Y_n))$
Parameters: τ_n
Repeat recursively $\log_2 \tau_n$ times:

- At each node, select a coordinate $j \in \{1, \ldots, p\}$ with probability $p_{n,j} \in (0, 1)$ where $\sum_{j=1}^{p} p_{n,j} = 1$;
- The split is performed at the centre of the cell along the selected variable.

Algorithm 5: Centred random forest.

We note that $\tau_n \geq 2$ is a fixed deterministic parameter which may depend on n but not on $\mathcal{D}_n$ and that each tree has exactly $2^{\lceil \log_2 \tau_n \rceil} \approx \tau_n$ nodes. However, there is no re-sampling step in the centred random forest algorithm and so $\mathcal{D}_n(\Theta) = ((X_1, Y_1), \ldots, (X_n, Y_n))$.

2.3 Statistical framework

Let us denote $(W_t)_{t\in\mathbb{Z}} := (X_t, Y_t)_{t\in\mathbb{Z}}$ where (X_t, Y_t) are defined in eq. (2.1.1). The first assumption throughout this paper is that the random sequence $(W_t)_{t\in\mathbb{Z}}$ is stationary. More precisely, we assume that $(W_t)_{t\in\mathbb{Z}}$ is a strongly stationary process as defined in definition 2.3.1.

Definition 2.3.1. The process $(W_t)_{t\in\mathbb{Z}}$ is said to be (strongly) stationary if $\forall k \in \mathbb{N}, \forall (t_1, \dots, t_k) \in \mathbb{Z}^k$ and for all $\tau \in \mathbb{Z}$,

$$(W_{t_1+\tau}, \dots W_{t_k+\tau}) = (W_{t_1}, \dots W_{t_k})$$

in distribution.

In order to prove the consistency of the RF-RI we also need to assume that $(W_t)_{t\in\mathbb{Z}}$ is an ergodic process as defined in definition 2.3.2.

Definition 2.3.2. The process $(W_t)_{t\in\mathbb{Z}}$ is said to be (mean-)ergodic if

$$\frac{1}{T}\int_0^T W_t \, \mathrm{d}t \xrightarrow[T\to\infty]{\mathrm{L}^2} \mathbb{E}(W_t).$$

Let $(C_n)_n$ be a positive sequence and define the truncated operator T_{C_n} by

$$T_{C_n} u = \begin{cases} u & \text{when } |u| \le C_n \\ C_n & \text{when } |u| > C_n. \end{cases}$$

and the set

$$T_{C_n}\mathcal{G}_n = \{T_{C_n} g, g \in \mathcal{G}_n\}$$

where $\mathcal{G}_n = \mathcal{G}(\mathcal{D}_n)$ denotes a class of functions $g : \mathcal{X} \to \mathcal{Y}$.

The consistency proof in Scornet et al. [2015] relies on the general consistency theorem found in Györfi et al. [2006]. In order to extend the consistency result to the dependent case, we use the extension of the general consistency theorem to the stationary ergodic setting as stated in proposition 2.3.1. We postpone the proof in section 2.7.3 for ease of readability.

Proposition 2.3.1. *Let $(W_t)_{t\in\mathbb{Z}}$ be a stationary ergodic process and $\mathcal{D}_n$ a data set. Let $\mathcal{G}_n = \mathcal{G}(\mathcal{D}_n)$ be a class of functions $g : \mathcal{X} \to \mathcal{Y}$, $(C_n)_n$ a positive sequence, f the regression function in eq. (2.1.1) and $\hat{f}_n$ an estimator which minimises the empirical L^2 risk on $\mathcal{G}_n$. If*

$$\lim_{n\to\infty} C_n = \infty, \tag{2.3.1a}$$

$$\lim_{n\to\infty} \mathbb{E}\left\{ \inf_{g\in\mathcal{G}_n, \|g\|_\infty \le C_n} \int |g(x) - f(x)|^2 \, \mu(\mathrm{d}x) \right\} = 0, \tag{2.3.1b}$$

$$\lim_{n\to\infty} \mathbb{E}\left\{ \sup_{g\in T_{C_n}\mathcal{G}_n} \left| \frac{1}{n}\sum_{i=1}^n |g(X_i) - Y_{i,L}|^2 - \mathbb{E}\left[g(X) - Y_L\right]^2 \right| \right\} = 0 \ \forall L > 0 \tag{2.3.1c}$$

then

$$\lim_{n\to\infty} \mathbb{E}\left\{ \int |\hat{f}_n(x) - f(x)|^2 \mu(\mathrm{d}x) \right\} = 0.$$

We recall the notion of weak dependence, more precisely the $\beta-$mixing case in which we establish the results.

Definition 2.3.3 (β-mixing process)**.** Let $\sigma_l = \sigma(W_1^l)$ and $\sigma'_{l+m} = \sigma(W_{l+m}^{\infty})$ be the sigma-algebras of events generated by the random variables $W_1^l = (W_1, \dots W_l)$ and $W_{l+m}^{\infty} = (W_{l+m}, W_{l+m+1}, \dots)$. The β-mixing coefficient is given by

$$\beta_m = \sup_{l \geq 1} \mathbb{E}\left[\sup_{B \in \sigma'_{l+m}} |\mathbb{P}(B|\sigma_l) - \mathbb{P}(B)|\right]$$

where the expectation is taken with respect to σ_l.

A stochastic process is said to be absolutely regular, or β-mixed, if

$$\lim_{m \to \infty} \beta_m = 0.$$

The most common $\beta-$mixing coefficients are known as the algebraic and exponential mixing defined as follows,

1. Algebraic mixing: $\beta_m = \mathcal{O}(m^{-r_\beta})$ for $r_\beta > 0$.
2. Exponential mixing: $\beta_m = \mathcal{O}(\exp(-bm^{k_\beta}))$ for $b, k_\beta > 0$.

The exponential mixing hypothesis is stronger than algebraic mixing. The values r_β and k_β are called the mixing exponents and the i.i.d process can be recovered by taking either the limit $r_\beta \to +\infty$ for the algebraic mixing or $k_\beta \to +\infty$ for the exponential mixing.

The $\beta-$mixing property is appealing in the theoretical setting since many statistical properties are preserved under this condition and are easy to manipulate. One method to manipulate $\beta-$mixing sequences is by using a lemma established in Yu [1994], recalled in lemma 2.7.1. Using this lemma, the dependent process is approximated with independent blocks of observations plus some linear function in β.

2.4 Results on the RF-RI

We recalled the studied models and the notion of weak dependence. We need the following hypotheses to establish the consistency of the RF-RI when the observations are weakly dependent:

- **H1a:** the data set $\mathcal{D}_n = ((X_1, Y_1), \dots, (X_n, Y_n))$ is composed of stationary ergodic $\beta-$mixing $(X_i, Y_i) \in [0,1]^p \times \mathbb{R}$;
- **H2a:** the errors $(\epsilon_i)_{1 \leq i \leq n}$ are independent;
- **H3a:** the response Y follows the additive model

$$Y = \underbrace{\sum_{j=1}^{p} f_j(X^{(j)})}_{f(X)} + \epsilon$$

  where $X = \left(X^{(1)}, \dots, X^{(p)}\right)$ is uniformly distributed over $[0,1]^p$, ϵ is an independent centred Gaussian noise with finite variance $\sigma^2 > 0$ and each component f_j is continuous.

We can now state the result of consistency of random forests when the observations are weakly dependent under the regime $\tau_n < a_n$ (i.e. the trees are not fully grown). Let us denote by a_n a sequence of integers such that $1 \leq a_n \leq n$.

Theorem 2.4.1. *Assume the hypothesis of stationary ergodic $\beta-$mixing data **H1a**. If*

- *the independent errors hypothesis **H2a** is satisfied;*
- *the additive model hypothesis **H3a** is satisfied;*
- $\frac{\tau_n \log(\alpha_n)^9 a_n}{\alpha_n} \underset{n\to\infty}{\longrightarrow} 0$;
- $\frac{\log(\alpha_n)^4 \beta_{a_n} \alpha_n}{a_n} \underset{n\to\infty}{\longrightarrow} 0$.

Then RF-RI are consistent, i.e.

$$\mathbb{E}\left[\hat{f}_n(X) - f(X)\right]^2 \underset{n\to\infty}{\longrightarrow} 0.$$

Let us first verify if we recover the result in the independent case. If the observations $(X_i, Y_i)_{1\leq i\leq n}$ are independent, $\beta_m = 0$ for all $m \geq 0$. We then get exactly the same hypotheses and result as in Scornet et al. [2015] by setting a_n equal to 1.

The hypotheses **H2a** and **H3a** are the same as in Scornet et al. [2015]. The condition $\frac{\tau_n \log(\alpha_n)^9 a_n}{\alpha_n} \longrightarrow 0$ as n tends to infinity is also highly similar to the last hypothesis in their theorem and recover it by setting a_n equal to 1. The last one is simply saying that the dependence between the data must not be too long in order to have consistency of the forest. Let us see how the dependence influences the number of leaves parameter τ_n. Let us suppose, in the following analysis, that r_β (or k_β in the exponential mixing case) is known. Let us consider the algebraic mixing case and suppose that $a_n = \alpha_n^a$ with $\frac{1}{1+r_\beta} < a < 1$. The last condition is then verified and the greatest value of τ_n must verify the following in order to obtain consistency:

$$\frac{\tau_n \log(\alpha_n)^9}{\alpha_n^{\frac{r_\beta}{1+r_\beta}}} \underset{n\to\infty}{\longrightarrow} 0.$$

In the exponential mixing case, suppose that $a_n = \frac{c}{b}\log(\alpha_n)^{\frac{1}{k_\beta}}$ with $c > 1$. The last condition is then equal to $\frac{\log(\alpha_n)^{4-\frac{1}{k_\beta}}}{\alpha_n^{c-1}}$ which tends to 0 as n tends to infinity. The penultimate condition can then be rewritten, implying that τ_n cannot be greater than the following condition is true,

$$\frac{\tau_n \log(\alpha_n)^{9+\frac{1}{k_\beta}}}{\alpha_n} \underset{n\to\infty}{\longrightarrow} 0.$$

This analysis leads to the following conclusion. The nature of the hypothesis appears in the choice of the parameter τ_n, influenced by r_β (or k_β) : the stronger the dependence between the observations, meaning that r_β (or k_β) is small, the shallower the trees need to be compared to the trees constructed based on i.i.d observations, in order to guarantee convergence.

2.5 Results on centred random forest

We analyse now the convergence rates of the centred random forest model when the observations are stationary β-mixing. The space $[0,1]^p$ is equipped with the standard Euclidean metric. We analyse the centred random forest in a sparse framework; this arises from the fact that in many applications the true dimension is always smaller than p. We assume that

the regression function only depends on a nonempty subset $\mathcal{S}$ of the p features. We use the letter S to denote the cardinal of $\mathcal{S}$. Based on this assumption we have

$$f(X) = \mathbb{E}\left[Y|X_{\mathcal{S}}\right]$$

where $X_{\mathcal{S}} = \left\{X^{(i)}, i \in \mathcal{S}\right\}$. Let us introduce $f^* : [0,1]^S \to \mathbb{R}$ that is the section of f corresponding to $\mathcal{S}$. We then have

$$f(X) = f^*\left(X_{\mathcal{S}}\right).$$

We also need the following hypotheses to establish the results:

- **H1b:** the data set $\mathcal{D}_n = \left(\left(X_1, Y_1\right), \ldots, \left(X_n, Y_n\right)\right)$ is composed of stationary $\beta-$mixing $(X_i, Y_i) \in [0,1]^p \times \mathbb{R}$;
- **H2b:** the errors $\epsilon_i := Y_i - f(X_i)$ are independent of finite variance $\sigma^2 > 0$.

2.5.1 Convergence rates

We first decompose $\mathbb{E}\left[\hat{f}_n(X) - f(X)\right]^2$ with the variance/bias decomposition:

$$\mathbb{E}\left[\hat{f}_n(X) - f(X)\right]^2 = \underbrace{\mathbb{E}\left[\hat{f}_n(X) - \tilde{f}_n(X)\right]^2}_{\text{Variance}} + \underbrace{\mathbb{E}\left[\tilde{f}_n(X) - f(X)\right]^2}_{\text{Bias}}$$

where

$$\tilde{f}_n(X) = \sum_{i=1}^{n} \mathbb{E}_\Theta\left[W_{n,i}(X,\Theta)\right] f(X_i).$$

We assume throughout that the coordinate-sampling probabilities are such that $p_{n,j} = \frac{1}{S}\left(1 + \nu_{n,j}\right)$ for $j \in \mathcal{S}$ and $p_{n,j} = \nu_{n,j}$ otherwise where each $\nu_{n,j}$ tends to 0 as n tends to infinity.

The first result concerns the variance term and the second the bias term. Let us denote by a_n a sequence of integers such that $1 \leq a_n \leq n$.

Proposition 2.5.1. *Assume the hypotheses of stationary $\beta-$mixing data **H1b**, independent errors **H2b** and that X is uniformly distributed on $[0,1]^p$ .Assuming that the coordinate-sampling probabilities are such that $p_{n,j} = \frac{1}{S}\left(1 + \nu_{n,j}\right)$ for $j \in \mathcal{S}$, then*

$$\mathbb{E}\left[\hat{f}_n(X) - \tilde{f}_n(X)\right]^2 \leq C\sigma^2 \left(\frac{S^2}{S-1}\right)^{S/2p} (1+\nu_n) \frac{\tau_n a_n^2}{n(\log \tau_n)^{S/2p}} + \sigma^2 \frac{\beta_{a_n} n}{a_n}$$

where

$$C = \frac{576}{\pi}\left(\frac{\pi \log 2}{16}\right)^{S/2p}$$

and

$$1 + \nu_n = \prod_{j \in \mathcal{S}} \left[\left(1 + \nu_{n,j}\right)^{-1} \left(1 - \frac{\nu_{n,j}}{S-1}\right)^{-1}\right]^{1/2p}.$$

As noted in Biau [2012], if $p_{lower} < p_{n,j} < p_{upper}$ for some constants $p_{lower}, p_{upper} \in (0,1)$ we have

$$1 + \nu_n \leq \left(\frac{S-1}{S^2 p_{lower}\left(1 - p_{upper}\right)}\right)^{\frac{S}{2p}}.$$

Proposition 2.5.2. *Assume the hypotheses of stationary $\beta-$mixing data **H1b**, X is uniformly distributed on $[0,1]^p$ and f^* is L-Lipschitz on $[0,1]^S$. Assuming that the coordinate-sampling probabilities are such that $p_{n,j} = \frac{1}{S}\left(1+\nu_{n,j}\right)$ for $j \in \mathcal{S}$, then*

$$\mathbb{E}\left[\tilde{f}_n(X) - f(X)\right]^2 \leq \frac{2SL^2 a_n}{\tau_n^{\frac{0,75}{S\log 2}(1+\gamma_n)}} + \exp\left(-\frac{\mu_n}{2\tau_n}\right) \sup_{x\in[0,1]^p} f^2(x) + \frac{\beta_{a_n} n}{a_n}\left[SL^2 + \sup_{x\in[0,1]^p} f^2(x)\right]$$

where $\gamma_n = \min_j \nu_{n,j}$.

The bias in the weakly dependent case only depends on the true dimension and not p which confirms the intuition and the result in the independent case as noted in Biau [2012]. However, we should keep in mind, whether in the dependent or independent setting, that the result relies on the assumption that the splits concentrate on the relevant variables.

Using the inequality $z\exp(-nz) \leq \frac{1}{en}$ for $z \in (0,1]$ and combining both previous convergence rates we get the following result.

Theorem 2.5.1. *Assume the hypotheses of stationary $\beta-$mixing data **H1b**, independent errors **H2b**, X is uniformly distributed on $[0,1]^p$ and f^* is L-Lipschitz on $[0,1]^S$. Assuming that the coordinate-sampling probabilities are such that $p_{n,j} = \frac{1}{S}\left(1+\nu_{n,j}\right)$ for $j \in \mathcal{S}$, then*

$$\mathbb{E}\left[\hat{f}_n(X) - f(X)\right]^2 \leq \frac{2SL^2 a_n}{\tau_n^{\frac{0,75}{S\log 2}(1+\gamma_n)}} + \mathcal{C}_{1,n}\frac{\tau_n a_n^2}{n} + \mathcal{C}_2\frac{\beta_{a_n} n}{a_n}$$

with

$$\mathcal{C}_{1,n} = 4e^{-1}\sup_{x\in[0,1]^p} f^2(x) + C\sigma^2\left(\frac{S^2}{S-1}\right)^{S/2p}(1+\nu_n),$$

$$\mathcal{C}_2 = SL^2 + \sigma^2 + \sup_{x\in[0,1]^p} f^2(x).$$

The independent errors hypothesis **H2b** is generally not true when $(X_i, Y_i)_{1\leq i\leq n}$ are $\beta-$mixing but it is assumed in some theoretical models as in the autoregressive model. The hypothesis $X \sim \mathcal{U}(0,1)^p$ is only a convenience and can be easily extended to the case where X admits a Lebesgue density which is lower and upper bounded.

We also recover the convergence rate in the independent setting given in Biau [2012] up to a constant factor. Let us suppose that we are in the independent case hence $\beta_m = 0$ for all $m \geq 0$. Setting a_n equal to 1 and plugging into propositions 2.5.1 and 2.5.2, we get exactly the same upper bound for the variance as in Biau [2012] back. However, regarding the bias term, we get a term with $\exp\left(-\frac{n}{4\tau_n}\right)$ instead of $\exp\left(-\frac{n}{2\tau_n}\right)$ which is due to a necessary pre-processing needed in order to work with $\beta-$mixing sequences.

Under the hypothesis of algebraic mixing and thus exponential mixing, the term depending on β is converging to 0 when n tends to infinity. The last term shows the price we must pay when dealing with $\beta-$mixing sequences instead of independent observations. More precisely, under algebraic mixing the penalty is of the form $\mathcal{O}\left(n^{1-a(r_\beta+1)}\right)$ with $a \in [0,1)$ which is the same penalty as in the convergence rate of boosting established in Lozano et al. [2014]. The following corollary precises, under algebraic and exponential mixing conditions, the choices of τ_n with the associated upper bound on the rate of consistency.

Corollary 2.5.1. *If the previous hypotheses are verified, we can compute the optimal choice of* τ_n*,*

$$\tau_n \propto \left(\frac{n}{a_n}\right)^{\frac{S\log 2}{0.75+S\log 2}}.$$

Plugging into the convergence rate, we get

$$\mathbb{E}\left[\hat{f}_n(X) - f(X)\right]^2 = \mathcal{O}\left(\left(\frac{a_n^{1.5+S\log 2}}{n^{0.75}}\right)^{\frac{1}{0.75+S\log 2}}\right) + \mathcal{O}\left(\frac{\beta_{a_n} n}{a_n}\right). \tag{2.5.1}$$

Suppose that r_β *and* k_β *are known.*

1. *Under algebraic mixing condition; eq. (2.5.1) is minimised taking*

$$a_n \propto n^{\frac{1.5+S\log 2}{2.25+2S\log 2+r_\beta(0.75+S\log 2)}}.$$

This implies that the parameter τ_n *is of the form*

$$\tau_n \propto n^{\frac{(1+r_\beta)S\log 2}{2.25+2S\log 2+r_\beta(0.75+S\log 2)}}$$

and achieves the following convergence rate:

$$\mathbb{E}\left[\hat{f}_n(X) - f(X)\right]^2 = \mathcal{O}\left(n^{\frac{-0.75r_\beta+0.75+S\log 2}{r_\beta(0.75+S\log 2)+2.25+2S\log 2}}\right).$$

2. *Under exponential mixing; taking*

$$a_n \propto \log n^{\frac{1}{k_\beta}}$$

gives

$$\tau_n \propto \left(\frac{n}{\log n^{\frac{1}{k_\beta}}}\right)^{\frac{S\log 2}{0.75+S\log 2}}.$$

Plugging into eq. (2.5.1) we get

$$\mathbb{E}\left[\hat{f}_n(X) - f(X)\right]^2 = \mathcal{O}\left(\left(\frac{\log n^{\frac{1.5+S\log 2}{k_\beta}}}{n^{0.75}}\right)^{\frac{1}{0.75+S\log 2}}\right).$$

The form of the convergence rate under algebraic mixing condition implies that in order to have consistency, we need the couple (r_β, S) to satisfy the inequality $0.75 + S\log 2 < 0.75 r_\beta$. It also implies that this result only treats the case where $r_\beta \geq 1.41$. We note that we recover the same optimal parameter and convergence rate as in Biau [2012] by letting r_β go to infinity. Under exponential mixing condition, the chosen τ_n is, up to a logarithmic factor in the denominator, the optimal parameter in the i.i.d case and gives the same convergence rate up to a logarithmic term depending on the inverse of k_β.

The previous analysis leads to the following conclusion. The choice of the parameter τ_n is determined by the nature of the hypothesis; the stronger the dependence between the observations, meaning that r_β (or k_β) is small, the shallower the trees need to be compared to the trees constructed based on i.i.d observations, in order to guarantee convergence.

2.6 Conclusion

The results for either the random forest-random input or the centred forest lead to the same conclusion: the more the dependence between the observations is long, the shallower the trees need to be compared to the trees constructed based on independent and identically distributed observations.

Theses results may also lead to new variants of random forests. The proofs of the results are based on a decomposition in blocks of the random process and the blocks are close to being independent. An analogy can be drawn between this decomposition and the so-called *block bootstraps* commonly used in time series estimation. Instead of considering the observations one by one, the algorithm is fed with blocks of observations and lead to better estimations. It could be interesting to modify the random forest algorithm in the same way to get a random forest adapted to time series.

2.7 Proofs

The proofs are based on the construction and lemma given in Yu [1994], also recalled below, but we note that a similar coupling lemma is proved in Berbee [1979].

We divide the sequence $(W_i)_{1\leq i\leq n}$ into $2\mu_n$ blocks each of size a_n. We assume that $n = 2\mu_n a_n$ and so consider that there is no remaining terms. We then define for $1 \leq i \leq \mu_n$,

$$\begin{aligned} H_j &= \{i : 2(j-1)a_n + 1 \leq i \leq (2j-1)a_n\} \\ T_j &= \{i : (2j-1)a_n + 1 \leq i \leq 2ja_n\}. \end{aligned}$$

and we denote

$$\begin{aligned} W^{(j)} &= \{W_i, i \in H_j\} \\ W'^{(j)} &= \{W_i, i \in T_j\}. \end{aligned}$$

We then denote the sequence of H-blocks $W_{a_n} = \left(W^{(j)}\right)_{1\leq j\leq \mu_n}$. We construct a sequence of independently distributed blocks $\Xi_{a_n} = \left(\Xi^{(j)}\right)_{1\leq j\leq \mu_n}$ where $\Xi^{(j)} = \{\xi_i, i \in H_j\}$ and such that for all $j \in \{1, \ldots, n\}$,

$$W^{(j)} \overset{(d)}{=} \Xi^{(j)}.$$

We construct in the same way a sequence of T-blocks. An illustration of this construction is given in Fig. 2.2.

Lemma 2.7.1 (Yu [1994]). *Let the distributions of W_{a_n} and Ξ_{a_n} be $\mathcal{Q}$ and $\tilde{\mathcal{Q}}$ respectively. Then for any measurable function u on $\mathbb{R}^{a_n\mu_n}$ with bound m,*

$$|\mathbb{E}_{\mathcal{Q}} u(W_{a_n}) - \mathbb{E}_{\tilde{\mathcal{Q}}} u(\Xi_{a_n})| \leq m\mu_n\beta_{a_n}.$$

2.7.1 Proof of consistency of the RF-RI

The computation of the approximation error is the same as in Scornet et al. [2015] since it does not require the independence of $(X_i, Y_i)_{1\leq i\leq n}$ but only stationarity and that the errors $(\epsilon_i)_{1\leq i\leq n}$ are independent.

The partition obtained with the random variable Θ and the data set $\mathcal{D}_n$ is denoted by $\mathcal{P}_n(\mathcal{D}_n, \Theta)$. We let

$$\Pi_n(\Theta) = \left\{\mathcal{P}\left((x_1, y_1), \ldots, (x_n, y_n), \Theta\right), (x_i, y_i) \in [0,1]^p \times [0,1]\right\}$$

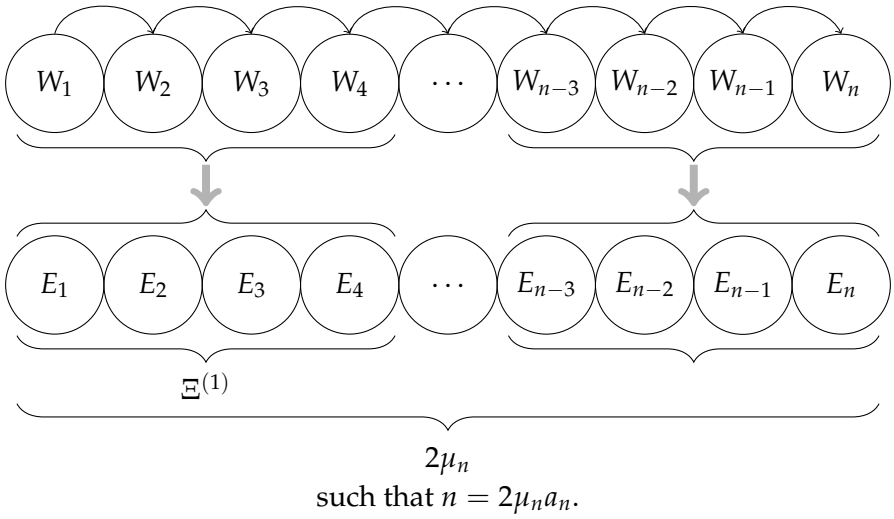

Figure 2.2: Construction of the new independent sequence Ξ.

be the family of all achievable partitions with random parameter Θ. We let

$$M\left(\Pi_n(\Theta)\right) = \max\left\{\text{Card}(\mathcal{P}, \mathcal{P} \in \Pi_n(\Theta)\right\}$$

be the maximal number of terminal nodes among all partitions in $\Pi_n(\Theta)$.

Given a set $z_1^n = \{z_1, \ldots, z_n\} \subset [0,1]^p$, $\Gamma_n\left(z_1^n, \Pi_n(\Theta)\right)$ denotes the number of distinct partitions of z_1^n induced by elements of $\Pi_n(\Theta)$, that is, the number of different partitions $\{z_1^n \cap A, A \in \mathcal{P}\}$ of z_1^n, for $\mathcal{P} \in \Pi_n(\Theta)$. Consequently, the partitioning number $\Gamma_n\left(\Pi_n(\Theta)\right)$ is defined by

$$\Gamma_n\left(\Pi_n(\Theta)\right) = \max\left\{\Gamma\left(z_1^n, \Pi_n(\Theta)\right), z_1, \ldots, z_n \in [0,1]^p\right\}.$$

Let $\mathcal{G}_n(\Theta)$ be the set of all functions $g : [0,1]^p \to \mathbb{R}$ piecewise constant on each cell of the partition $\mathcal{P}_n(\Theta)$. We define as in Scornet et al. [2015], $C_n = \|f\|_\infty + \sigma\sqrt{2\log\left(\alpha_n\right)^2}$, hence eq. (2.3.1a) is verified.

Regarding the estimation error, it is very similar to the computation done in Scornet et al. [2015] but we need to use a result established in Meir [2000] to introduce the $\beta-$mixing coefficient.

Theorem 2.7.1. *Let $(W_t)_{t\in\mathbb{Z}}$ be a β-mixing stationary stochastic process, with $|Y_i| \leq A_n$ and let $\mathcal{G}_n$ be a class of functions $g : \mathbb{R}^p \to \mathbb{R}$. Then, for any $d \geq 2$,*

$$\begin{aligned}
&\mathbb{P}\left(\sup_{\substack{g\in\mathcal{G}_n(\Theta)\\ \|g\|\leq A_n}} \left|\frac{1}{n}\sum_{j=1}^{n}\left|Y_j - g(X_j)\right|^d - \mathbb{E}\left[Y - g(X)\right]^d\right| > \epsilon\right) \\
&\leq 8\mathbb{E}\mathcal{N}\left(\frac{\epsilon}{32d(2A_n)^{d-1}}, \mathcal{G}_n(\Theta), l_{1,n}\right)\exp\left(-\frac{\mu_n\epsilon^2}{128(2A_n)^{2d}}\right) + 2\mu_n\beta_{a_n}
\end{aligned}$$

where $\mathcal{N}\left(\nu, \mathcal{G}(\Theta), l_{1,n}\right)$ is the ν-covering number of $\mathcal{G}_n(\Theta)$ w.r.t $l_{1,n} := \frac{1}{n}\sum_{i=1}^{n}|f(X_i) - g(X_i)|$.

Using theorem 2.7.1 we get,

$$\mathbb{P}\left(\sup_{\substack{g \in \mathcal{G}_n(\Theta) \\ \|g\| \leq C_n}} \left| \frac{1}{\alpha_n} \sum_{i=1}^{\alpha_n} |g(X_i) - Y_{i,L}|^2 - E|g(X) - Y_L|^2 \right| > \epsilon \right)$$
$$\leq 8\mathbb{E}\mathcal{N}\left(\frac{\epsilon}{128C_n}, \mathcal{G}_n(\Theta), l_{1,n} \right) \exp\left(-\frac{\mu_n \epsilon^2}{128(2C_n)^4} \right) + 2\mu_n \beta_{a_n}$$

where $\alpha_n = 2\mu_{\alpha_n} a_{\alpha_n}$. For simplicity's sake, we denote $\mu_n = \mu_{\alpha_n}$ and $a_n = a_{\alpha_n}$.
Let us compute $\mathbb{E}\mathcal{N}\left(\frac{\epsilon}{128C_N}, \mathcal{G}_n(\Theta), l_{1,n} \right)$ (cf. Györfi et al. [2006]),

$$\begin{aligned}
\mathcal{N}\left(\frac{\epsilon}{128C_n}, \mathcal{G}_n(\Theta), l_{1,n} \right) &\leq \Gamma_n(\Pi_n(\Theta)) \left[3\left(\frac{3e(2C_n)}{\frac{\epsilon}{128C_n}} \right)^2 \right]^{M(\Pi_n(\Theta))} \\
&\leq \Gamma_n(\Pi_n(\Theta)) \left[3\left(\frac{768eC_n^2}{\epsilon} \right)^2 \right]^{M(\Pi_n(\Theta))} \\
&\leq \Gamma_n(\Pi_n(\Theta)) \left[\frac{1331eC_n^2}{\epsilon} \right]^{2M(\Pi_n(\Theta))}.
\end{aligned}$$

Hence

$$\mathbb{E}\mathcal{N}\left(\frac{\epsilon}{128C_n}, \mathcal{G}_n(\Theta), l_{1,n} \right) \leq \Gamma_n(\Pi_n(\Theta)) \left[\frac{1331eC_n^2}{\epsilon} \right]^{2M(\Pi_n(\Theta))}.$$

Going back to the probability computation

$$\mathbb{P}\left(\sup_{\substack{g \in \mathcal{G}_n(\Theta) \\ \|g\| \leq C_n}} \left| \frac{1}{\alpha_n} \sum_{i=1}^{\alpha_n} |g(X_i) - Y_i|^2 - E|g(X) - Y|^2 \right| > \epsilon \right)$$
$$\leq 2\mu_n \beta_{a_n} + 8\exp\left(-\frac{\mu_n \epsilon^2}{2048C_n^4} \right) \exp\left(2M(\Pi_n(\Theta)) \log\left(\frac{1331eC_n^2}{\epsilon} \right) \right) \exp\left(\log(\Gamma_n(\Pi_n(\Theta))) \right).$$

Since $M(\Pi_n(\Theta)) \leq \tau_n$ and $\Gamma_n(\Pi_n(\Theta)) \leq (d\alpha_n)^{\tau_n}$,

$$\begin{aligned}
&2\mu_n \beta_{a_n} + 8\exp\left(-\frac{\mu_N \epsilon^2}{2048C_N^4} \right) \exp\left(2M(\Pi_N(\Theta)) \log\left(\frac{1331eC_n^2}{\epsilon} \right) \right) \exp\left(\log(\Gamma_n(\Pi_n(\Theta))) \right) \\
&\leq 2\mu_n \beta_{a_n} + 8\exp\left(-\frac{\mu_n \epsilon^2}{2048C_n^4} + 2\tau_n \log\left(\frac{1331eC_n^2}{\epsilon} \right) + \tau_n \log(d\alpha_n) \right) \\
&\leq 2\mu_n \beta_{a_n} + 8\exp\left(-\frac{\mu_n}{C_n^4} \left[\frac{\epsilon^2}{2048} - \frac{2\tau_n C_n^4}{\mu_n} \log\left(\frac{1331eC_n^2}{\epsilon} \right) - \frac{\tau_n C_n^4}{\mu_n} \log(d\alpha_n) \right] \right).
\end{aligned}$$

For n large enough,

$$\mathbb{P}\left(\sup_{\substack{g \in \mathcal{G}_n(\Theta) \\ \|g\| \leq C_n}} \left| \frac{1}{\alpha_n} \sum_{i=1}^{\alpha_n} |g(X_i) - Y_i|^2 - E|g(X) - Y|^2 \right| > \epsilon \right) \leq 2\mu_n \beta_{a_n} + 8\exp\left(-\frac{\mu_n}{C_n^4} \eta_{\epsilon,n} \right)$$

with

$$\eta_{\epsilon,n} = \frac{\epsilon^2}{2048} - \frac{8\sigma^4\tau_n \log(\alpha_n)^8 \log\left(\frac{2662e\sigma^2\log(\alpha_n)^4}{\epsilon}\right)}{\mu_n} - \frac{4\sigma^4\tau_n \log(\alpha_n)^8 \log(d\alpha_n)}{\mu_n}$$
$$\leq \frac{\epsilon^2}{2048} - \frac{8\sigma^4\tau_n \log(\alpha_n)^8 \log\left(\frac{2662e\sigma^2\log(\alpha_n)^4}{\epsilon}\right)}{\mu_n} - \frac{4\sigma^4\tau_n \log(d\alpha_n)^9}{\mu_n}.$$

We can now show that eq. (2.3.1c) holds:

$$\lim_{n\to\infty} \mathbb{E}\left\{\sup_{g\in T_{\beta_n}\mathcal{G}_n}\left|\frac{1}{n}\sum_{i=1}^{n}|g(X_i)-Y_{i,L}|^2 - \mathbb{E}\left[g(X)-Y_L\right]^2\right|\right\} = 0\,\forall L>0.$$

We denote

$$I = \sup_{g\in T_{\beta_n}\mathcal{G}_n}\left|\frac{1}{n}\sum_{i=1}^{n}|g(X_i)-Y_{i,L}|^2 - \mathbb{E}\left[g(X)-Y_L\right]^2\right|.$$

and observe that

$$I \leq 2(C_n+L)^2.$$

Thus for n large enough

$$\begin{aligned}\mathbb{E}\{I\} &\leq \mathbb{E}\{I\mathbb{1}_{I>\epsilon} + I\mathbb{1}_{I\leq\epsilon}\}\\ &\leq \epsilon + 2(C_n+L)^2\left(2\mu_n\beta_{a_n} + 8\exp\left(-\frac{\mu_n}{C_n^4}\eta_{\epsilon,n}\right)\right)\\ &= \epsilon + 16(C_n+L)^2\exp\left(-\frac{\mu_n}{C_n^4}\eta_{\epsilon,n}\right) + 4(C_n+L)^2\mu_n\beta_{a_n}.\end{aligned}$$

Hence with the $\beta-$mixing condition

$$\lim_{n\to\infty} \mathbb{E}\left\{\sup_{g\in T_{\beta_n}\mathcal{G}_n}\left|\frac{1}{n}\sum_{i=1}^{n}|g(X_i)-Y_{i,L}|^2 - \mathbb{E}\left[g(X)-Y_L\right]^2\right|\right\} = 0\,\forall L>0.$$

Thus, according to proposition 2.3.1,

$$\lim_{n\to\infty}\mathbb{E}\left(T_{\beta_n}\hat{f}_n(X,\Theta) - f(X)\right)^2 = 0.$$

We only need to check if the non-truncated random forest estimate is consistent, this step is identical to Scornet et al. [2015].

□

2.7.2 Proofs for centred forests

Proof of the variance rate, proposition 2.5.1. We follow the proof given in Biau [2012]. Since the training sample is not independent, we cannot get the same lines and results but the *main* ideas are, associated with lemma 2.7.1, the same.

Remember that the random forest estimator is written

$$\hat{f}_n(X,\mathcal{D}_n) = \mathbb{E}_\Theta\left[\hat{f}_n(X,\Theta,\mathcal{D}_n)\right]$$

with

$$\hat{f}_n(X, \Theta, \mathcal{D}_n) = \sum_{i=1}^{n} W_{n,i}(X, \Theta) Y_i$$

where

$$W_{n,i}(X, \Theta) = \frac{\mathbb{1}_{X_i \in A_n(X,\Theta)}}{\sum_{k=1}^{n} \mathbb{1}_{X_k \in A_n(X,\Theta)}} \mathbb{1}_{E_n(X,\Theta)} \quad \forall i \in \{1, \dots n\}.$$

Thus, omitting the dependence in $\mathcal{D}_n$, the random forest estimator can be written

$$\hat{f}_n(X) = \sum_{i=1}^{n} \mathbb{E}_\Theta \left[W_{n,i}(X, \Theta) \right] Y_i.$$

We define also $\tilde{f}_n(X)$

$$\tilde{f}_n(X) = \sum_{i=1}^{n} \mathbb{E}_\Theta \left[W_{n,i}(X, \Theta) \right] f(X_i).$$

We can now begin the computation,

$$\begin{aligned}
\mathbb{E}\left[\hat{f}_n(X) - \tilde{f}_n(X) \right]^2 &= \mathbb{E}\left[\sum_{i=1}^{n} \mathbb{E}_\Theta \left[W_{n,i}(X, \Theta) \right] (Y_i - f(X_i)) \right]^2 \\
&= \mathbb{E}\left[\sum_{i=1}^{n} \mathbb{E}_\Theta^2 \left[W_{n,i}(X, \Theta) \right] (Y_i - f(X_i))^2 \right] \\
&\quad + \mathbb{E}\left[\sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \mathbb{E}_\Theta \left[W_{n,i}(X, \Theta) W_{n,j}(X, \Theta) \right] \epsilon_i \epsilon_j \right]. \qquad (2.7.1)
\end{aligned}$$

The second term of eq. (2.7.1) is equal to zero since the errors $(\epsilon_i)_{1 \leq n}$ are independent by hypothesis **H2b**.

We next analyse the first term. We can upper-bound

$$\mathbb{E}\left[\sum_{i=1}^{n} \mathbb{E}_\Theta^2 \left[W_{n,i}(X, \Theta) \right] \epsilon_i^2 \right] \leq \sigma^2 \mathbb{E}\left[\sum_{i=1}^{n} \mathbb{E}_\Theta^2 \left[W_{n,i}(X, \Theta) \right] \right]$$

by hypothesis on the variance of the errors **H2b**.

The next step is to analyse the expectation of $W_{n,i}$. Since the data is not independent we cannot do exactly the same as in Biau [2012]. We need to rewrite the sum over n, decompose it in blocks and then use lemma 2.7.1. We can then use a similar argument as Biau [2012] which is, by introducing another random variable, to reveal a random binomial variable in the denominator. Let us first decompose the previous term

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{n} \mathbb{E}_\Theta^2 \left[W_{n,i}(X, \Theta) \right] \right] &= \mathbb{E}\left[\sum_{j=1}^{\mu_n} \sum_{i \in H_j} \mathbb{E}_\Theta^2 \left[W_{n,i}(X, \Theta) \right] \right] + \mathbb{E}\left[\sum_{j=1}^{\mu_n} \sum_{i \in T_j} \mathbb{E}_\Theta^2 \left[W_{n,i}(X, \Theta) \right] \right] \\
&= \mathbb{E}\left[u(X_{a_n}^H) \right] + \mathbb{E}\left[u(X_{a_n}^T) \right]
\end{aligned}$$

where

$$u(X_{a_n}^B) = \sum_{j=1}^{\mu_n} \sum_{i \in B_j} \mathbb{E}_\Theta^2 \left[W_{n,i}(X, \Theta) \right]$$

for $B = H$ or T. We easily observe that $u \leq 1$ by definition of $W_{n,i}$.

Let us begin with the first part of the right hand:

$$\mathbb{E}\left[\sum_{j=1}^{\mu_n}\sum_{i\in H_j}\mathbb{E}_\Theta^2\left[W_{n,i}(X,\Theta)\right]\right] \leq \mathbb{E}\left[\sum_{j=1}^{\mu_n}\sum_{i\in H_j}\mathbb{E}_\Theta^2\left[\widetilde{W}_{n,i}(X,\Theta)\right]\right] + \mu_n\beta_{a_n}$$

with

$$\widetilde{W}_{n,i}(X,\Theta) = \frac{\mathbb{1}_{\xi_i\in\widetilde{A}_n(X,\Theta)}}{\sum_{k=1}^n \mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta)}}\mathbb{1}_{\widetilde{E}_n(X,\Theta)}$$

and

$$\widetilde{E}_n(X,\Theta) = \left\{\sum_{i=1}^n \mathbb{1}_{\xi_i\in\widetilde{A}_n(X,\Theta)} \neq 0\right\}$$

where $\widetilde{A}_n(x,\Theta)$ denotes a copy of the cell $A_n(x,\Theta)$ composed of the $(\xi_i)_{1\leq i\leq n}$.

We introduce Θ' independent of Θ but with same distribution,

$$\begin{aligned}
\mathbb{E}\left[\sum_{j=1}^{\mu_n}\sum_{i\in H_j}\mathbb{E}_\Theta^2\left[\widetilde{W}_{n,i}(X,\Theta)\right]\right] &= \sum_{j=1}^{\mu_n}\mathbb{E}\left[\sum_{i\in H_j}\mathbb{E}_\Theta\left[\widetilde{W}_{n,i}(X,\Theta)\right]\mathbb{E}_{\Theta'}\left[\widetilde{W}_{n,i}(X,\Theta')\right]\right] \\
&= \sum_{j=1}^{\mu_n}\mathbb{E}\left[\sum_{i\in H_j}\mathbb{E}_{\Theta,\Theta'}\left[\widetilde{W}_{n,i}(X,\Theta)\widetilde{W}_{n,i}(X,\Theta')\right]\right] \\
&= \sum_{j=1}^{\mu_n}\mathbb{E}_{X,\Theta,\Theta'}\left[\sum_{i\in H_j}\frac{\mathbb{1}_{\xi_i\in\widetilde{A}_n(X,\Theta)\cap\widetilde{A}_n(X,\Theta')}}{\left(\sum_{k=1}^n \mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta)}\right)\left(\sum_{k=1}^n \mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta')}\right)}\mathbb{1}_{\widetilde{E}_n(X,\Theta)}\mathbb{1}_{\widetilde{E}_n(X,\Theta')}\right].
\end{aligned}$$

For a fixed j,

$$\begin{aligned}
&\mathbb{E}_{X,\Theta,\Theta'}\left[\sum_{i\in H_j}\frac{\mathbb{1}_{\xi_i\in\widetilde{A}_n(X,\Theta)\cap\widetilde{A}_n(X,\Theta')}}{\left(\sum_{k=1}^n \mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta)}\right)\left(\sum_{k=1}^n \mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta')}\right)}\mathbb{1}_{\widetilde{E}_n(X,\Theta)}\mathbb{1}_{\widetilde{E}_n(X,\Theta')}\right] \\
&\leq \mathbb{E}_{X,\Theta,\Theta'}\left[\sum_{i\in H_j}\mathbb{1}_{\xi_i\in\widetilde{A}_n(X,\Theta)\cap\widetilde{A}_n(X,\Theta')}\right. \\
&\qquad\left.\times\,\mathbb{E}\left[\frac{1}{\left(1+\sum_{k=1,\xi_k\notin H_j}^n \mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta)}\right)\left(1+\sum_{k=1,\xi_k\notin H_j}^n \mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta')}\right)}\middle| X,\xi_i,\Theta,\Theta'\right]\right].
\end{aligned}$$

By independence of the blocks we can remove the conditioning to ξ_i,

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{\left(1+\sum_{k=1,\xi_k\notin H_j}^n \mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta)}\right)\left(1+\sum_{k=1,\xi_k\notin H_j}^n \mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta')}\right)}\middle| X,\xi_i,\Theta,\Theta'\right] \\
&= \mathbb{E}\left[\frac{1}{\left(1+\sum_{k=1,\xi_k\notin H_j}^n \mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta)}\right)\left(1+\sum_{k=1,\xi_k\notin H_j}^n \mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta')}\right)}\middle| X,\Theta,\Theta'\right].
\end{aligned}$$

Using Cauchy-Schwarz's inequality, for a fixed j,

$$\mathbb{E}\left[\frac{1}{\left(1+\sum_{k=1,\xi_k\notin H_j}^{n}\mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta)}\right)\left(1+\sum_{k=1,\xi_k\notin H_j}^{n}\mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta')}\right)}\middle|X,\Theta,\Theta'\right]$$
$$\leq \mathbb{E}^{1/2}\left[\frac{1}{\left(1+\sum_{k=1,\xi_k\notin H_j}^{n}\mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta)}\right)^2}\middle|X,\Theta\right]\times\mathbb{E}^{1/2}\left[\frac{1}{\left(1+\sum_{k=1,\xi_k\notin H_j}^{n}\mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta')}\right)^2}\middle|X,\Theta'\right].$$

Using the following fact (cf. Györfi et al. [2006]) that

$$\mathbb{E}\left[\frac{1}{1+Bin(N,p)^2}\right]\leq\frac{3}{(N+1)(N+2)p^2}.$$

and since each blocks are independent

$$\mathbb{E}^{1/2}\left[\frac{1}{\left(1+\sum_{k=1,\xi_k\notin H_j}^{n}\mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta)}\right)^2}\middle|X,\Theta\right]\leq\mathbb{E}^{1/2}\left[\frac{1}{1+\left(\sum_{\tilde{j}=1}^{2\mu_n-1}\mathbb{1}_{\xi_{\tilde{j}}\in\widetilde{A}_n(X,\Theta)}\right)^2}\middle|X,\Theta\right]$$

where $\tilde{j}$ denotes one component of each block $(H_j)_{1\leq j\leq\mu_n}$ and $(T_j)_{1\leq j\leq\mu_n}$. By independence of the blocks we get

$$\sum_{\tilde{j}=1}^{2\mu_n-1}\mathbb{1}_{\xi_{\tilde{j}\in\widetilde{A}_n(X,\Theta)}}\sim Bin\left(2\mu_n-1,\mathbb{P}(X\in A_n(X,\Theta)|X,\Theta)\right).$$

Since we suppose that the law is uniform on $[0,1]^p$ and by the construction of the tree we get

$$\mathbb{P}\left(X\in A_n(X,\Theta)|X,\Theta\right)=2^{-\lceil\log_2\tau_n\rceil}.$$

The same is done for the conditional expectation with respect to (X,Θ'). Thus

$$\mathbb{E}_{X,\Theta,\Theta'}\left[\sum_{i\in H_j}\mathbb{1}_{\xi_i\in\widetilde{A}_n(X,\Theta)\cap\widetilde{A}_n(X,\Theta')}\mathbb{E}\left[\frac{1}{\left(1+\sum_{k=1,\xi_k\notin H_j}^{n}\mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta)}\right)\left(1+\sum_{k=1,\xi_k\notin H_j}^{n}\mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta')}\right)}\middle|X,\xi_i,\Theta,\Theta'\right]\right]$$
$$\leq\frac{3\times2^{2\lceil\log_2\tau_n\rceil}}{4\mu_n^2}\mathbb{E}_{X,\Theta,\Theta'}\left[\sum_{i\in H_j}\mathbb{1}_{\xi_i\in\widetilde{A}_n(X,\Theta)\cap\widetilde{A}_n(X,\Theta')}\right]$$
$$\leq\frac{12\tau_n^2}{4\mu_n^2}\mathbb{E}_{X,\Theta,\Theta'}\left[\sum_{i\in H_j}\mathbb{1}_{\xi_i\in\widetilde{A}_n(X,\Theta)\cap\widetilde{A}_n(X,\Theta')}\right]$$
$$\leq\frac{3\tau_n^2}{\mu_n^2}a_n\mathbb{P}\left(\xi_1\in\widetilde{A}_n\left(X,\Theta\right)\cap\widetilde{A}_n\left(X,\Theta'\right)\right).$$

The last inequality using the fact that even though dependent, they have the same distribution.

The rest is the same as in Biau [2012]. After the computations over H, we get

$$\mathbb{E}\left[\sum_{j=1}^{\mu_n}\sum_{i\in H_j}\mathbb{E}_\Theta^2\left[\widetilde{W}_{n,i}(X,\Theta)\right]\right]\leq\widetilde{C}\left(\frac{S^2}{S-1}\right)^{S/2p}(1+\nu_n)\frac{\tau_n a_n\mu_n}{\mu_n^2(\log\tau_n)^{S/2p}}$$

with

$$\widetilde{C} = \frac{144}{\pi}\left(\frac{\pi \log 2}{16}\right)^{S/2p}$$

and

$$1+\nu_n = \prod_{j\in\mathcal{S}}\left[\left(1+\nu_{n,j}\right)^{-1}\left(1-\frac{\nu_{n,j}}{S-1}\right)^{-1}\right]^{1/2p}.$$

We do the same over T.

Combining both analyses we have

$$\mathbb{E}\left[\hat{f}_n(X)-\tilde{f}_n(X)\right]^2 \leq 2\widetilde{C}\sigma^2\left(\frac{S^2}{S-1}\right)^{S/2p}(1+\nu_n)\frac{\tau_n a_n}{\mu_n(\log\tau_n)^{S/2p}} + 2\sigma^2\beta_{a_n}\mu_n.$$

By construction of the blocs $\mu_n = \frac{n}{2a_n}$, plugging in the previous expression we have

$$\mathbb{E}\left[\hat{f}_n(X)-\tilde{f}_n(X)\right]^2 \leq C\sigma^2\left(\frac{S^2}{S-1}\right)^{S/2p}(1+\nu_n)\frac{\tau_n a_n^2}{n(\log\tau_n)^{S/2p}} + \frac{\sigma^2\beta_{a_n}n}{a_n}$$

with

$$C = \frac{576}{\pi}\left(\frac{\pi \log 2}{16}\right)^{S/2p}$$

and

$$1+\nu_n = \prod_{j\in\mathcal{S}}\left[\left(1+\nu_{n,j}\right)^{-1}\left(1-\frac{\nu_{n,j}}{S-1}\right)^{-1}\right]^{1/2p}.$$

□

Proof of the bias term, proposition 2.5.2. The start of the proof is the same as in Biau [2012] since it does not use the hypothesis of independence between the observations:

$$\begin{aligned}
\mathbb{E}\left[\tilde{f}_n(X)-f(X)\right]^2 &\leq \mathbb{E}\left[\sum_{i=1}^n W_{n,i}(X,\Theta)\left(f(X_i)-f(X)\right)\right]^2 + \sup_{x\in[0,1]^d} f^2(x)\mathbb{P}\left(E_n^c(X,\Theta)\right)\\
&\leq \mathbb{E}\left[\sum_{i=1}^n W_{n,i}(X,\Theta)\left(f^*(X_{i,\mathcal{S}})-f^*(X_{\mathcal{S}})\right)^2\right] + \sup_{x\in[0,1]^d} f^2(x)\mathbb{P}\left(E_n^c(X,\Theta)\right) \text{ (cf. Biau [2012])}\\
&\leq L^2\mathbb{E}\left[\sum_{i=1}^n W_{n,i}(X,\Theta)\|X_i-X\|_{\mathcal{S}}^2\right] + \sup_{x\in[0,1]^d} f^2(x)\mathbb{P}\left(E_n^c(X,\Theta)\right)
\end{aligned}$$

where we get the last inequality using the hypothesis that f^* is $L-$Lipschitz. To go further in the analysis we have to use lemma 2.7.1 to get independent variables. We proceed similarly to the first proof,

$$\mathbb{E}\left[\sum_{i=1}^n W_{n,i}(X,\Theta)\|X_i-X\|_{\mathcal{S}}^2\right] = \mathbb{E}\left[v\left(X_{a_n}^H\right)\right] + \mathbb{E}\left[v\left(X_{a_n}^T\right)\right]$$

with

$$v\left(X_{a_n}^B\right) = \sum_{j=1}^{\mu_n}\sum_{i\in H_j} W_{n,i}(X,\Theta)\|X_i-X\|_{\mathcal{S}}^2$$

for $B = H$ or T. We observe that

$$v \leq \sup_{(x,y)\in[0,1]^{\mathcal{S}}\times[0,1]^{\mathcal{S}}} \|x-y\|_{\mathcal{S}}^2 \leq S.$$

Thus, using lemma 2.7.1,

$$\mathbb{E}\left[v\left(X_{a_n}^H\right)\right] \leq \mathbb{E}\left[\sum_{j=1}^{\mu_n}\sum_{i\in H_j}\widetilde{W}_{n,i}(X,\Theta)\|\xi_i - X\|_{\mathcal{S}}^2\right] + S\mu_n\beta_{a_n}.$$

We do the same over T.

We need to do a similar operation to compute the probability $\mathbb{P}\left(E_n^c(X,\Theta)\right)$. We recall that $E_n := \left\{\sum_{i=1}^n \mathbb{1}_{X_i\in A_n(X,\Theta)} \neq 0\right\}$:

$$\begin{aligned}\mathbb{P}\left(E_n^c(X,\Theta)\right) &= \mathbb{E}\left[\mathbb{1}_{\sum_{i=1}^n \mathbb{1}_{X_i\in A_n(X,\Theta)}=0}\right]\\ &= \mathbb{E}\left[\mathbb{1}_{X_1\notin A_n(X,\Theta)}\cdots\mathbb{1}_{X_n\notin A_n(X,\Theta)}\right]\\ &\leq E\left[w(X_{a_n}^H)\right]\end{aligned}$$

where

$$w(X_{a_n}^H) = \prod_{j=1}^{\mu_n}\prod_{i\in H_j}\mathbb{1}_{X_i\notin A_n(X,\Theta)} \Rightarrow w \leq 1.$$

Using lemma 2.7.1,

$$\mathbb{E}\left[w(X_{a_n}^H)\right] \leq \mathbb{P}\left[\forall\, 1\leq j\leq\mu_n, \forall i\in H_j, \xi_i\notin\widetilde{A}_n(X,\Theta)\right] + \mu_n\beta_{a_n}.$$

We get

$$\begin{aligned}\mathbb{E}\left[\tilde{f}_n(X) - f(X)\right]^2 \leq L^2\mathbb{E}&\left[\sum_{i=1}^n \widetilde{W}_{n,i}(X,\Theta)\|\xi_i - X\|_{\mathcal{S}}^2\right] \qquad (2.7.2)\\ &+ \sup_{x\in[0,1]^p} f^2(x)\,\mathbb{P}\left[\forall\, 1\leq j\leq\mu_n, \forall i\in H_j, \xi_i\notin\widetilde{A}_n(X,\Theta)\right]\\ &+ \mu_n\beta_{a_n}\left[2SL^2 + \sup_{x\in[0,1]^p} f^2(x)\right]. \qquad (2.7.3)\end{aligned}$$

We first analyse the term $\mathbb{P}\left[\forall\, 1\leq j\leq\mu_n, \forall i\in H_j, \xi_i\notin\widetilde{A}_n(X,\Theta)\right]$,

$$\mathbb{P}\left[\forall\, 1\leq j\leq\mu_n, \forall i\in H_j, \xi_i\notin\widetilde{A}_n(X,\Theta)\right] \leq \mathbb{P}\left[\forall\, 1\leq j\leq\mu_n,\ \text{pick } \tilde{i}\in H_j, \xi_{\tilde{i}}\notin\widetilde{A}_n(X,\Theta)\right]$$

where $\tilde{i}$ is an arbitrary index chosen in $\{1,\ldots,a_n\}$. Since the the blocks are independent, the terms in the probability are independent. Furthermore, they have the same distribution. Thus

$$\begin{aligned}\mathbb{P}\left[\forall 1\leq j\leq\mu_n, \forall i\in H_j, \xi_i\notin\widetilde{A}_n(X,\Theta)\right] &\leq \mathbb{P}^{\mu_n}\left[\xi_1\notin\widetilde{A}_n(X,\Theta)\right]\\ &= \left(1 - 2^{-\lceil\log_2\tau_n\rceil}\right)^{\mu_n} \text{ (by construction of the tree)}\\ &\leq \exp\left(-\frac{\mu_n}{2\tau_n}\right).\end{aligned}$$

Plugging in eq. (2.7.3) we have

$$\mathbb{E}\left[\tilde{f}_n(X) - f(X)\right]^2 \leq L^2\mathbb{E}\left[\sum_{i=1}^{n} \widetilde{W}_{n,i}(X,\Theta)\|\xi_i - X\|_{\mathcal{S}}^2\right] + \exp\left(-\frac{\mu_n}{2\tau_n}\right) \sup_{x\in[0,1]^d} f^2(x)$$
$$+ \mu_n\beta_{a_n}\left[2SL^2 + \sup_{x\in[0,1]^d} f^2(x)\right].$$

Let us analyse the first term:

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{n} \widetilde{W}_{n,i}(X,\Theta)\|\xi_i - X\|_{\mathcal{S}}^2\right] &= \sum_{j=1}^{\mu_n}\mathbb{E}\left[\sum_{i\in H_j}\frac{\mathbb{1}_{\xi_i\in\widetilde{A}_n(X,\Theta)}}{\widetilde{N}_n(X,\Theta)}\mathbb{1}_{\widetilde{E}_n(X,\Theta)}\|\xi_i - X\|_{\mathcal{S}}^2\right] \\
&+ \sum_{j=1}^{\mu_n}\mathbb{E}\left[\sum_{i\in T_j}\frac{\mathbb{1}_{\xi_i\in\widetilde{A}_n(X,\Theta)}}{\widetilde{N}_n(X,\Theta)}\mathbb{1}_{\widetilde{E}_n(X,\Theta)}\|\xi_i - X\|_{\mathcal{S}}^2\right] \\
&\leq \sum_{j=1}^{\mu_n}\mathbb{E}\left[\sum_{i\in H_j}\mathbb{1}_{\xi_i\in\widetilde{A}_n(X,\Theta)}\|\xi_i - X\|_{\mathcal{S}}^2\,\mathbb{E}\left[\frac{1}{\left(1+\sum_{k=1,\xi_k\notin H_j}^{n}\mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta)}\right)}\middle| X,\xi_i,\Theta\right]\right] \\
&+ \sum_{j=1}^{\mu_n}\mathbb{E}\left[\sum_{i\in T_j}\mathbb{1}_{\xi_i\in\widetilde{A}_n(X,\Theta)}\|\xi_i - X\|_{\mathcal{S}}^2\,\mathbb{E}\left[\frac{1}{\left(1+\sum_{k=1,\xi_k\notin T_j}^{n}\mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta)}\right)}\middle| X,\xi_i,\Theta\right]\right].
\end{aligned}$$

For a fixed j

$$\mathbb{E}\left[\frac{1}{\left(1+\sum_{k=1,\xi_k\notin H_j}^{n}\mathbb{1}_{\xi_k\in\widetilde{A}_n(X,\Theta)}\right)}\middle| X,\xi_i,\Theta\right] \leq \mathbb{E}\left[\frac{1}{\left(1+\sum_{\tilde{k}=1}^{2\mu_n-1}\mathbb{1}_{\xi_{\tilde{k}}\in\widetilde{A}_n(X,\Theta)}\right)}\middle| X,\Theta\right]$$

where $\tilde{k}$ denotes one component of each block $(H_j)_{1\leq j\leq\mu_n}$ and $(T_j)_{1\leq j\leq\mu_n}$. By independence of the blocks we have

$$\sum_{\tilde{k}=1}^{2\mu_n-1}\mathbb{1}_{\xi_{\tilde{k}}\in\widetilde{A}_n(X,\Theta)} \sim Bin(2\mu_n - 1, 2^{-\lceil\log_2\tau_n\rceil})$$

using the same argument as in the proof "convergence rate for the variance". The following inequality (cf. Györfi et al. [2006]),

$$\mathbb{E}\left[\frac{1}{1+Bin(N,p)}\right] \leq \frac{1}{(N+1)p},$$

gives

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{n} \widetilde{W}_{n,i}(X,\Theta)\|\xi_i - X\|_{\mathcal{S}}^2\right] &\leq \sum_{j=1}^{\mu_n}\mathbb{E}\left[\sum_{i\in H_j}\mathbb{1}_{\xi_i\in\widetilde{A}_n(X,\Theta)}\|\xi_i - X\|_{\mathcal{S}}^2\frac{2^{\lceil\log_2\tau_n\rceil}}{2\mu_n}\right] \\
&+ \sum_{j=1}^{\mu_n}\mathbb{E}\left[\sum_{i\in T_j}\mathbb{1}_{\xi_i\in\widetilde{A}_n(X,\Theta)}\|\xi_i - X\|_{\mathcal{S}}^2\frac{2^{\lceil\log_2\tau_n\rceil}}{2\mu_n}\right] \\
&\leq \tau_n\mathbb{E}\left[\sum_{i\in H_1}\mathbb{1}_{\xi_i\in\widetilde{A}_n(X,\Theta)}\|\xi_i - X\|_{\mathcal{S}}^2\right] + \tau_n\mathbb{E}\left[\sum_{i\in T_1}\mathbb{1}_{\xi_i\in\widetilde{A}_n(X,\Theta)}\|\xi_i - X\|_{\mathcal{S}}^2\right] \\
&\leq 2a_n\tau_n\mathbb{E}\left[\mathbb{1}_{\xi_1\in\widetilde{A}_n(X,\Theta)}\|\xi_1 - X\|_{\mathcal{S}}^2\right] \text{ by stationarity.}
\end{aligned}$$

The rest is the same as in Biau [2012]. We get

$$\mathbb{E}\left[\sum_{i=1}^{n}\widetilde{W}_{n,i}(X,\Theta)\|\xi_i - X\|_S^2\right] \leq \frac{2a_n S}{\tau_n^{\frac{0.75}{S\log 2}(1+\gamma_n)}}$$

with $\gamma_n = \min_j \nu_{n,j}$. We conclude

$$E\left[\tilde{f}_n(X) - f(X)\right]^2 \leq \frac{2SL^2 a_n}{\tau_n^{\frac{0.75}{S\log 2}(1+\gamma_n)}} + \exp\left(-\frac{\mu_n}{2\tau_n}\right)\sup_{x\in[0,1]^d} f^2(x) + \mu_n\beta_{a_n}\left[2SL^2 + \sup_{x\in[0,1]^d} f^2(x)\right].$$

Replacing using that $\mu_n = \frac{n}{2a_n}$ we have

$$E\left[\tilde{f}_n(X) - f(X)\right]^2 \leq \frac{2SL^2 a_n}{\tau_n^{\frac{0.75}{S\log 2}(1+\gamma_n)}} + \exp\left(-\frac{n}{4a_n\tau_n}\right)\sup_{x\in[0,1]^d} f^2(x) + \frac{\beta_{a_n} n\left[SL^2 + \sup_{x\in[0,1]^d} f^2(x)\right]}{a_n}.$$

□

2.7.3 Tool to establish consistency in stationary ergodic case

Following the definition of the truncated operator T we denote

$$W_L = T_L W$$

and

$$W_{i,L} = T_L W_i$$

for $W = X$ or Y.

We first introduce the general consistency theorem as known from Györfi et al. [2006] and used in Scornet et al. [2015]. From now on μ denotes the distribution of X.

Theorem 2.7.2. *Let $\mathcal{D}_n = \{(X_1, Y_1), \ldots (X_n, Y_n)\}$ i.i.d. Let $\mathcal{G}_n = \mathcal{G}(\mathcal{D}_n)$ be a class of functions $g : \mathcal{X} \to \mathcal{Y}$,the estimator $\hat{f}_n$ which minimises the empirical L^2 risk on $\mathcal{G}_n$ and f the regression function. If*

$$\lim_{n\to\infty} C_n = \infty,$$

$$\lim_{n\to\infty}\mathbb{E}\left\{\inf_{g\in\mathcal{G}_n,\|g\|_\infty\leq C_n}\int |g(x) - f(x)|^2\,\mu(\mathrm{d}x)\right\} = 0,$$

$$\lim_{n\to\infty}\mathbb{E}\left\{\sup_{g\in T_{C_n}\mathcal{G}_n}\left|\frac{1}{n}\sum_{i=1}^{n}|g(X_i) - Y_{i,L}|^2 - \mathbb{E}\left[g(X) - Y_L\right]^2\right|\right\} = 0\ \forall L > 0$$

then

$$\lim_{n\to\infty}\mathbb{E}\left\{\int |\hat{f}_n(x) - f(x)|^2\mu(\mathrm{d}x)\right\} = 0.$$

We extend this theorem to dependent process. The only assumption we actually need is that the stochastic process is stationary and ergodic.

Proposition 2.7.1. *Let $(X_t, Y_t)_{t\in\mathbb{Z}}$ be a stationary ergodic process and a data set $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$. Let $\mathcal{G}_n = \mathcal{G}(\mathcal{D}_n)$ be a class of functions $g : \mathcal{X} \to \mathcal{Y}$, the estimator $\hat{f}_n$ which minimises the empirical L^2 risk on $\mathcal{G}_n$ and f the regression function. Under eqs. (2.3.1a) to (2.3.1c),*

$$\lim_{n\to\infty} \mathbb{E}\left\{\int |\hat{f}_n(x) - f(x)|^2 \mu(\,\mathrm{d}x)\right\} = 0.$$

Proof. To prove this result, we follow the same line as in Györfi et al. [2006]. Instead of using the large of law numbers for i.i.d variables we use the law of large numbers for stationary ergodic processes.

We write

$$\begin{aligned}
\int_{\mathbb{R}^p} |\hat{f}_n(x) - f(x)|^2 \mu(\,\mathrm{d}x) &= \mathbb{E}\left[|\hat{f}_n(X) - Y|^2 \middle| \mathcal{D}_n\right] - \mathbb{E}|f(X) - Y|^2 \\
&= \left(\left(\mathbb{E}\left[|\hat{f}_n(X) - Y|^2 \middle| \mathcal{D}_n\right]\right)^{1/2} - \left(\mathbb{E}|f(X) - Y|^2\right)^{1/2}\right) \\
&\times \left(\left(\mathbb{E}\left[|\hat{f}_n(X) - Y|^2 \middle| \mathcal{D}_n\right]\right)^{1/2} + \left(\mathbb{E}|f(X) - Y|^2\right)^{1/2}\right) \\
&= \left(\left(\mathbb{E}\left[|\hat{f}_n(X) - Y|^2 \middle| \mathcal{D}_n\right]\right)^{1/2} - \left(\mathbb{E}|f(X) - Y|^2\right)^{1/2}\right)^2 \\
&+ 2\left(\mathbb{E}|f(X) - Y|^2\right)^{1/2} \left(\left(\mathbb{E}\left[|\hat{f}_n(X) - Y|^2 \middle| \mathcal{D}_n\right]\right)^{1/2} - \left(\mathbb{E}|f(X) - Y|^2\right)^{1/2}\right).
\end{aligned}$$

It suffices to show

$$\mathbb{E}\left(\left(\mathbb{E}\left[|\hat{f}_n(X) - Y|^2 \middle| \mathcal{D}_n\right]\right)^{1/2} - \left(\mathbb{E}|f(X) - Y|^2\right)^{1/2}\right)^2 \underset{n\to\infty}{\longrightarrow} 0.$$

We rewrite this term,

$$\begin{aligned}
&\mathbb{E}\left(\left(\mathbb{E}\left[|\hat{f}_n(X) - Y|^2 \middle| \mathcal{D}_n\right]\right)^{1/2} - \left(\mathbb{E}|f(X) - Y|^2\right)^{1/2}\right)^2 \\
&\leq 2\mathbb{E}\left(\left(\mathbb{E}\left[|\hat{f}_n(X) - Y|^2 \middle| \mathcal{D}_n\right]\right)^{1/2} - \inf_{g\in\mathcal{G}_n, \|g\|\leq\beta_n} \left(\mathbb{E}|g(X) - Y|^2\right)^{1/2}\right) \\
&+ 2\mathbb{E}\left(\inf_{g\in\mathcal{G}_n, \|g\|\leq\beta_n} \left(\mathbb{E}|g(X) - Y|^2\right)^{1/2} - \left(\mathbb{E}|f(X) - Y|^2\right)^{1/2}\right).
\end{aligned}$$

The last term can be bounded,

$$\begin{aligned}
&2\mathbb{E}\left(\inf_{g\in\mathcal{G}_n, \|g\|\leq\beta_n} \left(\mathbb{E}|g(X) - Y|^2\right)^{1/2} - \left(\mathbb{E}|f(X) - Y|^2\right)^{1/2}\right) \\
&\leq 2\mathbb{E}\left(\inf_{g\in\mathcal{G}_n, \|g\|\leq\beta_n} \left(\mathbb{E}|g(X) - f(X)|^2\right)^{1/2}\right)^2 \\
&\leq 2\mathbb{E}\left(\inf_{g\in\mathcal{G}_n, \|g\|\leq\beta_n} \mathbb{E}|g(X) - f(X)|^2\right) \underset{n\to\infty}{\longrightarrow} 0 \text{ by eq. (2.3.1b).}
\end{aligned}$$

It remains to show that

$$2\mathbb{E}\left(\left(\mathbb{E}\left[|\hat{f}_n(X) - Y|^2 \middle| \mathcal{D}_n\right]\right)^{1/2} - \inf_{g\in\mathcal{G}_n, \|g\|\leq\beta_n} \left(\mathbb{E}|g(X) - Y|^2\right)^{1/2}\right) \underset{n\to\infty}{\longrightarrow} 0.$$

We can lower bound this term by

$$-\mathbb{E}\left[\inf_{g\in\mathcal{G}_n,\|g\|\leq\beta_n}\left(\int_{\mathbb{R}^p}|g(x)-f(x)|^2\mu(\,\mathrm{d}x)\right)^{1/2}\right]^2$$

and upper bound it by

$$\mathbb{E}\left(2\left(\mathbb{E}|Y-Y_L|^2\right)^{1/2}+2\left(\frac{1}{n}\sum_{j=1}^{n}|Y_i-Y_{i,L}|^2\right)^{1/2}\right.$$
$$\left.+2\sup_{g\in T_{\beta_n}\mathcal{G}_n}\left|\left(\frac{1}{n}\sum_{j=1}^{n}|g(X_i)-Y_{i,L}|^2\right)^{1/2}-\left(\mathbb{E}|g(X)-Y|^2\right)^{1/2}\right|\right)^2. \tag{2.7.5}$$

Using the inequality: $(a+b+c)^2\leq 3a^2+3b^2+3c^2\ \forall(a,b,c)\in\mathbb{R}^3$ and $(\sqrt{a}-\sqrt{b})^2\leq|a-b|$ we have

$$\begin{aligned}
\textit{eq. (2.7.5)} &\leq \mathbb{E}\left[\inf_{G\in\mathcal{G}_n,\|g\|\leq\beta_n}\int_{\mathbb{R}^p}|g(x)-f(x)|^2\mu(\,\mathrm{d}x)\right]\\
&\quad+6\mathbb{E}\left[\sup_{g\in T_{\beta_n}\mathcal{G}_n}\left|\frac{1}{n}\sum_{j=1}^{n}|g(X_i)-Y_{i,L}|^2-\mathbb{E}|g(X)-Y|^2\right|\right]\\
&\quad+6\mathbb{E}|Y-Y_L|^2+6\mathbb{E}\left(\frac{1}{n}\sum_{j=1}^{n}|Y_i-Y_{i,L}|^2\right)\\
&\quad\underset{n\to\infty}{\longrightarrow}12\mathbb{E}|Y-Y_L|^2.
\end{aligned}$$

The last line using eqs. (2.3.1b) and (2.3.1c) and the strong law for stationary ergodic process.
We get the result letting $L\to\infty$.

□

2.8 Bibliography

H. C. P. Berbee. Random walks with stationary increments and renewal theory. *MC Tracts*, 112:1–223, 1979. 45

G. Biau. Analysis of a random forests model. *Journal of Machine Learning Research*, 13:1063–1095, 2012. 35, 42, 43, 44, 48, 49, 51, 52, 55

G. Biau and E. Scornet. A random forest guided tour. *TEST*, 25:197–227, 2016. 35

L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996. 34

L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001. 34

L. Breiman. Consistency for a simple model of random forests. Technical report, 2004. 35, 38

L. Breiman, J. Friedman, C. Stone, and R. Olshen. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984. 34, 35

D. R. Cutler, T. C. Edwards, K. H. Beard, A. Cutler, K. T. Hess, J. Gibson, and J. J. Lawler. Random forests for classification in ecology. *Ecology*, 88:2783–2792, 2007. 34

J. Dedecker, P. Doukhan, G. Lang, L. R. J. Rafael, S. Louhichi, and C. Prieur. Weak dependence. In *Weak Dependence: With Examples and Applications*, pages 9–20. Springer, 2007. 36

G. Dudek. Short-term load forecasting using random forests. In *Intelligent Systems'2014*, pages 821–828, Cham, 2015. Springer International Publishing. 35

A. Fischer, L. Montuelle, M. Mougeot, and D. Picard. Statistical learning for wind power: A modeling and stability study towards forecasting. *Wind Energy*, 20:2037–2047, 2017. 34, 35

L. Györfi, M. Kohler, A. Krzyzak, and H. Walk. *A distribution-free theory of nonparametric regression*. Springer Science & Business Media, 2006. 39, 47, 51, 54, 55, 56

M. J. Kane, N. Price, M. Scotch, and P. Rabinowitz. Comparison of arima and random forest time series models for prediction of avian influenza h5n1 outbreaks. *BMC bioinformatics*, 15:276, 2014. 34, 35

A. Lahouar and J. Ben Hadj Slama. Random forests model for one day ahead load forecasting. In *IREC2015 The Sixth International Renewable Energy Congress*, pages 1–6, 2015. 35

A. C. Lozano, S. R. Kulkarni, and R. E. Schapire. Convergence and consistency of regularized boosting with weakly dependent observations. *IEEE Transactions on Information Theory*, 60: 651–660, 2014. 36, 43

R. Meir. Nonparametric time series prediction through adaptive model selection. *Machine learning*, 39:5–34, 2000. 36, 46

L. Mentch and G. Hooker. Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *Journal of Machine Learning Research*, 17:1–41, 2016. 35, 37

A. M. Prasad, L. R. Iverson, and A. Liaw. Newer classification and regression tree techniques: bagging and random forests for ecological prediction. *Ecosystems*, 9:181–199, 2006. 34

E. Rio. Inequalities and limit theorems for weakly dependent sequences. Lecture: cel-00867106, 2013. 36

E. Scornet. On the asymptotics of random forests. *Journal of Multivariate Analysis*, 146:72–83, 2016. Special Issue on Statistical Models and Methods for High or Infinite Dimensional Spaces. 35, 37

E. Scornet, G. Biau, and J.-P. Vert. Consistency of random forests. *The Annals of Statistics*, 43: 1716–1741, 2015. 35, 39, 41, 45, 46, 48, 55

J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56:116–124, 2013. 34

V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston. Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43:1947–1958, 2003. 34

S. Wager and S. Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113:1228–1242, 2018. 35, 38

B. Yu. Rates of convergence for empirical processes of stationary mixing sequences. *The Annals of Probability*, 22:94–116, 1994. 40, 45

Chapter 3

Aggregation of Multi-scale Experts for Bottom-up Load Forecasting

In collaboration with Yannig GOUDE, *Pascal* MASSART *and Jean-Michel* POGGI

Abstract

The development of smart grid and new advanced metering infrastructures induces new opportunities and challenges for utilities. Exploiting smart meters information for forecasting stands as a key point for energy providers who have to deal with time varying portfolio of customers as well as grid managers who needs to improve accuracy of local forecasts to face with distributed renewable energy generation development. We propose a new machine learning approach to forecast the system load of a group of customers exploiting individual load measurements in real time and/or exogenous information like weather and survey data. Our approach consists in building experts using random forests trained on some subsets of customers then normalise their predictions and aggregate them with a convex expert aggregation algorithm to forecast the system load. We propose new aggregation methods and compare two strategies for building subsets of customers: 1) hierarchical clustering based on survey data and/or load features and 2) random clustering strategy. These approaches are evaluated on a real data set of residential Irish customers load at a half hourly resolution. We show that our approaches achieve a significant gain in short term load forecasting accuracy of around 25 percent of RMSE.

Sommaire

Nomenclature

$d_{\text{gower}}(i,l)$ Gower's distance between individuals i and l

$D_{i,l}$ Dissimilarity between individuals i and l

$l_t(x)$ Loss for of a forecast x at instant t

e Expert index

h Forecasting horizon

i,l Individual indices

j Level of the partition

k Information index

q,q' Cluster indices

s,t Time/Instant indices

$\mathcal{A}_j$ jth partition

$\mathcal{A}_{j,q}$ qth cluster of partition $\mathcal{A}_j$

α_T Number of observations selected for a given tree

$\eta_{e,t}$ Learning rate of the eth expert at instant t

$\hat{p}_t^{\mathcal{A}_j}$ Aggregation weight associated to the forecast of the partition $\mathcal{A}_j$ at instant t

$\hat{p}_t^{\mathcal{A}_{j,q}}$ Aggregation weight associated to the forecast of cluster $\mathcal{A}_{j,q}$ at instant t

$\hat{Z}_t^{\mathcal{A}_{j,q}}$ System load forecast for the cluster $\mathcal{A}_{j,q}$ at instant t

$\hat{p}_{e,t}$ Aggregation weight of the eth expert at instant t

$\hat{Y}_t^e$ Load forecast provided by the expert e at time t

$\hat{Y}_{t,\text{ 2S-MSWA}}^{\mathcal{A}_{1:L}}$ System load forecast given by 2S-MSWA for the sequence $(\mathcal{A}_j)_{1\leq j\leq L}$ at instant t

$\hat{Y}_{t,\text{ MSWA}}^{\mathcal{A}_{1:L}}$ System load forecast given by MSWA for the sequence $(\mathcal{A}_j)_{1\leq j\leq L}$ at instant t

$\hat{Y}_t^{\mathcal{A}_j}$ System load forecast for partition $\mathcal{A}_j$ given by SSWA at instant t

κ_j	Number of clusters in the partition $\mathcal{A}_j$
$c_{\mathcal{A}_{j,q}}$	Renormalising constant associated to the cluster $\mathcal{A}_{j,q}$
I_i	*ith* individual information
$I_{i,k}$	*k*th information variable of individual i
L	Number of partitions considered
M	Number of trees of a random forest
m_{try}	Number of variables selected at each node for the construction of a given tree
N	Number of experts
n	Number of individuals
R_t	Regret term at instant t
$R_{e,t}$	Regret term for the eth expert at instant t
T	Length of the time series
t_X	Maximum lag in days of the information X
t_Y	Maximum lag in days for the load Y
U_t	Input space for the model
$X_t^{\mathcal{A}_{j,q}}$	Exogenous information associated to the cluster $\mathcal{A}_{j,q}$ at instant t
$X_{t,i}$	*ith* individual temporal information at instant t
Y_t^*	Oracle at instant t
$Y_t^{\mathcal{A}_{j,q}}$	Load associated to the cluster $\mathcal{A}_{j,q}$ at instant t
$Y_{t,i}$	Load for individual i at instant t
$Z_t^{\mathcal{A}_{j,q}}$	Renormalised load associated to the cluster $\mathcal{A}_{j,q}$ at instant t

3.1 Introduction

Electricity landscape is facing new challenges both in term of electricity production and consumption. The more intermittent generation induces additional uncertainties and a need for a local optimisation of the electricity market (see e.g. Lezama et al. [2018]). Consumption habits are changing as well as with the development of electric vehicles, heat pumps or building insulation (see Fischer et al. [2015]; Kikusato et al. [2019]).

Recent development of smart grid, in particular new advanced metering infrastructure (AMI) generates new opportunities for utilities. AMI allows communications between producers, network managers and consumers in a bidirectional way.

Home Energy Management System (HEMS) and demand response tools make load control a reality and encourage electricity providers to design new offers (see Shareef et al. [2018]). Data induces by AMI and their exploitation for business are huge. According to Wang et al. [2017] China will soon generate 5 terabytes each year from the electricity consumption data collected every 15 minutes by smart meters. Exploiting these data is crucial for improving commercial strategy, pricing policy and to develop new personalised services like energy diagnostics and recommendations.

We focus on exploiting individual load data for short term load forecasting which is the main entry for energy management and crucial for the optimisation of the grid. As stated in the survey Wang et al. [2018a], smart meter fine-grained data offer new opportunities to improve forecasts at aggregate levels: national, regional, substations of the distribution grid, a town, a district, etc.

In the so-called bottom-up forecasting setting we suppose to have access, for an individual i among n (typically i is a household and n is the total number of customers in the portfolio of an electricity provider), to its load at time t denoted $Y_{t,i}$ and our objective is to forecast $Y_{t+h} = \sum_{i=1}^{n} Y_{t+h,i}$ the system load of a portfolio at horizon h (we consider the case $h = 48$ half-hours). Smart meter data are generally coming with side-information like household characteristics or tariff option; we associate to each individual i a vector I_i representing this information. Furthermore we have access to an exogenous time-dependent vector $X_{t,i}$ e.g meteorological information from a station close to the individual i. As shown in Sevlian and Rajagopal [2018], forecasting the load of a few number of customers (from 1 to 50) is a hard task and it sounds more reasonable to focus on aggregates of a least 100 households. Here, we consider 500 residential customers from Ireland (see Commission for Energy Regulation [2011]).

Bottom-up forecasting strategy consists in clustering the individuals, fitting forecasting models in each cluster and summing forecasts to predict the total. The intuition behind such an approach is that the population could be divided in subpopulations with different consumption habits which necessitate different models. In the literature on bottom-up forecasting customer grouping is achieved with clustering methods and final forecast is obtained with a simple aggregation of group forecasts. In Quilumba et al. [2015a], clustering is done with a k-means algorithm applied to specific features of load curves (mean consumption for 5 well chosen periods of day, mean consumption per day of a week and peak position into the year) and aggregation is the sum of group forecasts derived from deep learning models. They obtain a gain of 11% in forecast accuracy on the Irish data set and New-York smart meters. Wang et al. [2018b] propose an ensemble of forecasts based on a hierarchical clustering on individual average weekly profiles and a deep learning model for forecasting in each cluster. Forecasts are then aggregated using linear regression. We propose here an extension of these approaches in which the aggregation is conducted with time varying weights derived from robust online aggregation of experts methods (see Cesa-Bianchi and Lugosi [2006] for an introduction and overview). The objective of these methods is to propose a mathematical framework that allows to develop strategies and associated algorithms to take advantage of very different and potentially high number of forecasts. We propose here to generate a variety of experts from clustering methods based either on load profiles or exogenous information. The total load of each group of customers is then used to compute experts using random forests (see Breiman [2001]) which are a good com-

promise between fast automatic calibration and performances (see Dudek [2015], Lahouar and Ben Hadj Slama [2015], Moon et al. [2018] for recent applications of random forests to short term load forecasting). Exploiting the robustness of online aggregation strategies we also propose to generate our experts on clusters coming from random or hierarchical clustering, and show that the associated expert family achieves good performance because of the diversity of the experts enhanced in the multi-scale strategies presented in section 3.3.

3.2 Methodology

We present each stage of the strategies separately. The clustering step is presented in section 3.2.1, The forecasting step in section 3.2.2 and the aggregation step in section 3.2.2.

3.2.1 Clustering

Recall that for an individual i we have access to its load $Y_{t,i}$ but also to non-temporal information $(I_i)_{1\leq i\leq n}$ e.g home heating or the type of insulation. We aim to group the individuals in clusters of similar properties. Clustering is an important field of machine learning and there are many ways to cluster individuals. We propose a hierarchical agglomerative clustering approach based on Gower's distance to measure the similarity as we have both quantitative and qualitative information.

Gower's distance

Let us denote $I_{i,k}$ (resp. $I_{l,k}$) the kth variable of the vector I_i (resp. I_l). The Gower's distance (Gower and Gower [1971]) between two individuals i and l is defined as follows:

$$d_{\text{gower}}(i,l) = \frac{\sum_{k=1}^{p} \delta(il,k) d(il,k)}{\sum_{k=1}^{p} \delta(il,k)}$$

where $d(il,k)$, the kth variable contribution to the total distance, is the distance between $I_{i,k}$ and $I_{l,k}$ given by

- Binary or nominal case :

$$d(il,k) = \begin{cases} 1 & \text{if the variables are equal} \\ 0 & \text{else} \end{cases}$$

- Continuous case:

$$d(il,k) = \frac{|I_{i,k} - I_{l,k}|}{\max(I_{\cdot,k}) - \min(I_{\cdot,k})}.$$

and where $\delta(ij,k)$ is equal to 0 if the variable $I[,k]$ is missing for the individual i or j (or both) or if the variable is equal to 0.

Hierarchical agglomerative clustering (HAC)

Many clustering algorithms exist and choosing the best one depends on the data and the objective. We consider for our purpose the hierarchical agglomerative clustering in order to obtain not only one partition but a sequence of partitions hierarchically organised.

Two notions are needed for hierarchical agglomerative clustering: a dissimilarity between sets of individuals and a linkage criterion. The dissimilarities are defined between each pair of individual and is a representation of how close the individuals are to each other. Once we begin to cluster the individuals, we need to define the dissimilarity between clusters which is done by the linkage criterion. We only consider the complete-linkage criterion which is defined between two sets A and B as $\max(D_{i,l}, i \in A, l \in B)$ where $D_{i,l}$ is the dissimilarity between the individuals i and l. The complete-linkage clustering is detailed in algorithm 6.

input: Dissimilarity matrix D
repeat
-Find the least dissimilar pair (r, v) of clusters
-Merge clusters r and v to form the next clustering.
-The new dissimilarity between the cluster (r, v) and a cluster g is defined as $D_{g,(r,v)} = \max\left(D_{g,r}, D_{g,v}\right)$.
until *All objects are in one cluster*

Algorithm 6: Hierarchical agglomerative clustering

We calculate a dissimilarity matrix using Gower's distance on the non-temporal information $(I_i)_{1\leq i\leq n}$ and we feed this dissimilarity to the HAC to get a hierarchical sequence of partitions.

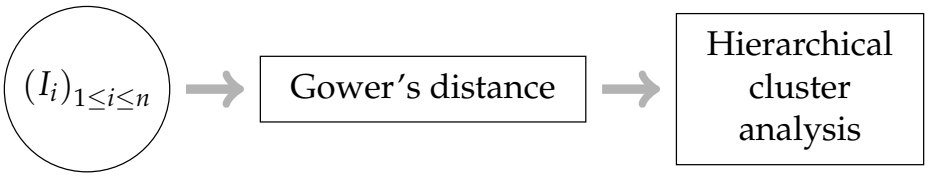

Figure 3.1: Skeleton of the clustering procedure.

Let us fix a strictly increasing sequence of number of clusters $(\kappa_j)_{1\leq j\leq K}$. We consider a sequence of partitions $(\mathcal{A}_j)_{1\leq j\leq K}$ where $\mathcal{A}_1 = \{1, \ldots, n\}$ and each partition $\mathcal{A}_j$ has κ_j clusters for $j \in \{1, \ldots, K\}$. We denote $\mathcal{A}_{j,q}$ the qth cluster of partition $\mathcal{A}_j$ such that $\mathcal{A}_j = \cup_{q=1}^{\kappa_j} \mathcal{A}_{j,q}$ and $\forall q \neq q', \mathcal{A}_{j,q} \cap \mathcal{A}_{j,q'} = \emptyset$. We denote by j the level associated to the partition $\mathcal{A}_j$.

We denote $Y_t^{\mathcal{A}_{j,q}}$ the load of the individuals belonging to the cluster $\mathcal{A}_{j,q}$:

$$Y_t^{\mathcal{A}_{j,q}} = \sum_{i\in\mathcal{A}_{j,q}} Y_{t,i} \tag{3.2.1}$$

and

$$X_t^{\mathcal{A}_{j,q}} = \sum_{i\in\mathcal{A}_{j,q}} w_i X_{t,i}$$

where w_i are weights depending on the nature of $X_{t,i}$. For example if $X_{t,i}$ is the temperature of a station close to the individual i at time t the weight could be $w_i = \frac{1}{Card(\mathcal{A}_{j,q})}$ and $X_t^{\mathcal{A}_{j,q}}$ corresponds to the mean temperature of the cluster $\mathcal{A}_{j,q}$ at the instant t.

We can represent the sequence of hierarchical partitions with a tree (Fig. 3.2) where the root is the single class: the n individuals together and then the individuals are clustered according to the dissimilarity matrix. Using eq. (3.2.1) we assign to each cluster its load (Fig. 3.3).

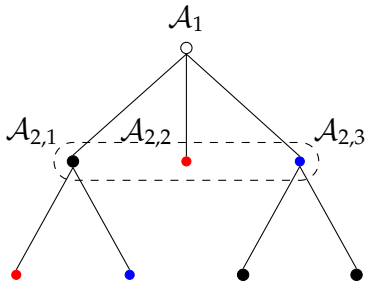

Figure 3.2: Hierarchical partitioning.

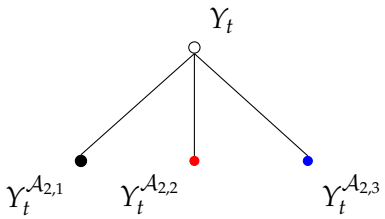

Figure 3.3: Load associated to the different clusters.

3.2.2 Forecasting methods

We present here the forecasting methods, first the random forests and then sequential expert aggregation.

Random forests

Let us suppose the following regression model

$$Y_t = f(U_t) + \epsilon_t$$

where U_t corresponds to the explanatory variables which are defined later in section 3.3, ϵ_t corresponds to the error at instant t and f the regression function. Random forests aim to estimate the regression function f from a training set $((U_1, Y_1), \ldots, (U_T, Y_T))$.

Random forests were introduced in 2001 by Breiman in Breiman [2001] and are since then one of the most popular algorithms in machine learning: they are known to perform well on many problems, are fast to compute and easy to tune. We compared, on the aggregated signal Y_t, random forests to others well-known methods and observed gains between 15% and 110% compared to lasso Tibshirani [1996], SVM Wang [2005], CART Breiman et al. [1984], bagging Breiman [1996], GAM Hastie and Tibshirani [1986]; Wood [2017] and loss of less than 3% compared to boosting

Chen and Guestrin [2016]; Friedman [2001]; Schapire and Freund [2012]. We choose random forests because of their good forecasting qualities and their robustness compared to boosting.

input: $((U_1, Y_1), \ldots, (U_T, Y_T))$
parameters: M, α_T, m_{try}
for *1 to* M **do**
  Draw randomly $\alpha_T \leq T$ points, with or without replacement
  Construct a tree:
  - At each node, select randomly m_{try} variables
  - Select the best split using the variance criterion among the previously chosen variables

end
output: mean of the M trees

Algorithm 7: Random forest.

A tree is a piecewise constant decomposition of the input space where each node of the tree corresponds to a binary test and each region is associated to a constant. A random forest is a collection of M random trees where each tree is grown independently according to the following procedure.

The first step is the bootstrap/subsampling: α_T points are selected with or without replacement among the T realisations. Then a tree is constructed on these selected points. At each node of the tree the best split (the variable and the location on this variable) is determined by minimising the variance. Instead of considering this criterion on all the variables the choice of inputs is restricted to a random subset of size m_{try} and then reiterate until stopping conditions are met.

Even if the theoretical settings of random forests was until recently restricted to the i.i.d case a theoretical study extending it to the time-dependent case is proposed in Goehry [2018]. In addition, applications on time series could be found in Fischer et al. [2017]; Kane et al. [2014]; Niu et al. [2018], in electricity load forecasting Dudek [2015]. In this paper we choose random forests to compute the forecasts because of their good flexibility and good accuracy at different sizes of aggregation. Another advantage is that random forests can easily incorporate with exogenous variables.

Sequential expert aggregation

In this section we provide a short description of sequential expert aggregation for forecasting. See Cesa-Bianchi and Lugosi [2006] for a complete presentation of these methods and Devaine et al. [2011] for a previous application on national load data.

We suppose to observe a bounded sequence of observations (here half-hourly total load of customers) $Y_1, \ldots, Y_T \in [0, B]$ that we want to forecast step by step at each time instant $t = 1, \ldots, T$. Each instance t, N experts provide forecasts $(\hat{Y}_t^1, \ldots, \hat{Y}_t^N) \in [0, B]^N$ of future observation Y_t. These experts could be the output of any forecasting model, the output of an algorithm (statistical model, physical model...) or a human based projection. After that, aggregation is conducted to build a mixture $\hat{Y}_t = \sum_{e=1}^N \hat{p}_{e,t} \hat{Y}_t^e$ where the weights $(\hat{p}_{e,t})_{1 \leq e \leq N}$, such that for all $e = 1, \ldots, N$, $\hat{p}_{e,t} \geq 0$ and $\sum_{e=1}^N \hat{p}_{e,t} = 1$, are to be suitably chosen. Then, Y_t is observed and instance $t + 1$ starts.

Performance of forecasts (experts and aggregation) are measured with a loss function. It could be any convex loss but for simplicity and in accordance to the RMSE criterion used in our case study we consider the square loss $\ell_t(x) = (Y_t - x)^2$. Thus, at time t expert e suffers loss $\ell_t(\hat{Y}_t^e) = (Y_t - \hat{Y}_t^e)^2$ and the aggregation $\ell_t(\hat{Y}_t) = (Y_t - \hat{Y}_t)^2$. We call Oracle an optimal forecast which is unknown in advance and usually hard to beat in terms of forecasting accuracy (see Cesa-Bianchi and Lugosi [2006]). We denote it by $\hat{Y}_t^*$. For example, it could be the best fixed convex aggregation or the best expert (best w.r.t the entire time interval performance, of course unknown a priori). The goal of aggregation algorithms is to minimise the total loss $\sum_{t=1}^{T}(Y_t - \hat{Y}_t)^2$ that can be expressed:

$$\frac{1}{T}\sum_{t=1}^{T}(Y_t - \hat{Y}_t)^2 \quad \triangleq \quad \frac{1}{T}\sum_{t=1}^{T}(Y_t - \hat{Y}_t^*)^2 + R_T\,,$$

where R_T is the so-called regret term, it is the error suffered by our algorithm relatively to the error of the oracle (see again Cesa-Bianchi and Lugosi [2006]). The aim is thus to propose algorithms that, regarding competitive oracles, achieve low regrets.

In our study we use the so-called ML-Poly of Gaillard et al. [2014], implemented in the R package Gaillard and Goude [2016] and successfully used for load and price forecasting in Gaillard and Goude [2015]; Gaillard et al. [2016]. Intuitively, the algorithm, described in algorithm 8, gives more weight to an expert who has a high regret which means that the expert suffers a lower loss than the aggregation. In addition, the time varying learning rate $\eta_{e,t}$ could be seen as a vector step size parameter of gradient descent varying with time. This makes this algorithm particularly interesting as no parameter tuning is needed.

initialisation: $\hat{\boldsymbol{p}}_1 = (1/N, \ldots, 1/N)$ and $\boldsymbol{R}_0 = (0, \ldots, 0)$
for t : *1 to* T **do**

- pick the learning rates: $\eta_{e,t-1} = 1/\left(1 + \sum_{s=1}^{t-1}\left(\ell_s(\hat{Y}_s) - \ell_s(Y_s^e)\right)^2\right)$
- calculate the weights $\hat{\boldsymbol{p}}_t$: $\hat{p}_{e,t} = \eta_{e,t-1}\,(R_{e,t-1})_+ \;/\; \boldsymbol{\eta}_{t-1} \cdot (\boldsymbol{R}_{t-1})_+$ where $\boldsymbol{Y}_+$ is the non-negative parts of $\boldsymbol{Y}$
- output prediction $\hat{Y}_t = \sum_{e=1}^{N} \hat{p}_{e,t}\hat{Y}_t^e$
- for each expert h update the regret: $R_{e,t} = R_{e,t-1} + \ell_t(\hat{Y}_t) - \ell_t(Y_t^e)$ $\boldsymbol{R}_t = (R_{1,t}, \ldots, R_{N,t})$

end

Algorithm 8: ML-Poly algorithm.

3.3 Bottom-up forecasting strategies

For a given cluster $\mathcal{A}_{j,q}$, we suppose the following model:

$$Y_t^{\mathcal{A}_{j,q}} = f\left(U_t^{\mathcal{A}_{j,q}}\right) + \epsilon_t \tag{3.3.1}$$

where $U_t^{\mathcal{A}_{j,q}} = \left(Y_{t-48}^{\mathcal{A}_{j,q}}, \ldots, Y_{t-48\cdot t_Y}^{\mathcal{A}_{j,q}}, X_t^{\mathcal{A}_{j,q}}, \ldots, X_{t-48\cdot t_X}^{\mathcal{A}_{j,q}}\right)$, t_Y denotes the load's maximum number of lags in days and t_X the maximum number of lags in days of the information X. We suppose the same generic model whether a cluster has 1 or 300 individuals and estimate f using a non-parametric method presented in section 3.2.2.

We detail in this section three new strategies to forecast the system load Y_t. The gist of these new strategies is the combination of diversity of the forecasts given by random forests and sequential aggregation. The first strategy is called the SSWA. We fix a partition $\mathcal{A}_j$, the SSWA is an aggregation of the forecasts given by the κ_j clusters which are computed on renormalised loads. The two other strategies are multi-scaled. Instead of only considering one partition, a sequence of partitions $(\mathcal{A}_j)_{1\leq j\leq L}$ is considered. The MSWA consists of one aggregation of all the forecasts given in the sequence of partition, namely one aggregation of the $\sum_{j=1}^{L}\kappa_j$ forecasts which are computed in the same way as in the SSWA strategy. The 2S-MSWA is a strategy in two steps. The first step is to compute the SSWA for each partition in the sequence $(\mathcal{A}_j)_{1\leq j\leq L}$. The second step is to aggregate the L SSWA predictions from the first step.

Strategy	Acronym
Single scale weighted aggregation	SSWA
Multi-scale weighted aggregation	MSWA
Two steps multi-scaled weighted aggregation	2S-MSWA

Single scale weighted aggregation (SSWA)

The goal is to mix different predictors obtained for each cluster on a given partition $\mathcal{A}_j$. We then first normalise $Y_t^{\mathcal{A}_{j,q}}\ \forall q \in [1, \ldots, \kappa_j]$ such that Y_t and $Y_t^{\mathcal{A}_{j,q}}$ have the same order of magnitude using a constant $c_{\mathcal{A}_{j,q}}$. We denote $Z_t^{\mathcal{A}_{j,q}}$ the load $Y_t^{\mathcal{A}_{j,q}}$ normalised:

$$Z_t^{\mathcal{A}_{j,q}} = c_{\mathcal{A}_{j,q}} Y_t^{\mathcal{A}_{j,q}}.$$

We normalise the loads since the goal is to forecast the system load and that in the last step we combine these forecasts. Since random forests are non-linear, we do the normalisation step before fitting the model.

We then estimate for each normalised load $Z_t^{\mathcal{A}_{j,q}}$ a predictor described in eq. (3.3.1) denoted $\hat{Z}_t^{\mathcal{A}_{j,q}}$.

For the jth partition we then estimate Y_t by

$$\hat{Y}_t^{\mathcal{A}_j} = \sum_{q=1}^{\kappa_j} \hat{p}_t^{\mathcal{A}_{j,q}} \hat{Z}_t^{\mathcal{A}_{j,q}} \tag{3.3.2}$$

where $\hat{p}_t^{\mathcal{A}_{j,q}}$ are positive weights such that $\forall t\ \sum_{q=1}^{\kappa_j} \hat{p}_t^{\mathcal{A}_{j,q}} = 1$. These weights could be either constant (for example uniform weights) or time dependent. The single

scaled weighted aggregation is summarised in algorithm 9.

input: partition $\mathcal{A}_j$, weights $\left(\hat{p}_t^{\mathcal{A}_{j,q}}\right)_{j,q,t}$
for $q = 1$ *to* κ_j **do**
  Normalise the sub-load for the ith cluster
  Train the forecasting model for the ith cluster
  Forecast the ith cluster on the test set
end
Estimate the system load

Algorithm 9: Single scale weighted aggregation.

A strategy which is commonly found in the bottom-up forecasting setting is: for a fixed partition $\mathcal{A}_j$, compute each forecast for each cluster without renormalising and then sum the κ_j forecasts. This can be seen as a sub-case of the SSWA taking $c_{\mathcal{A}_{j,q}} = 1$ and $\hat{p}_t^{\mathcal{A}_{j,q}} = 1$. We call this strategy Simple Aggregation Strategy (SAS).

Multi-scale weighted aggregation (MSWA)

The second strategy to forecast Y_t is to mix all the normalised estimators obtained for each cluster in each partition up to some level L:

$$\hat{Y}_{t,\,\text{MSWA}}^{\mathcal{A}_{1:L}} = \sum_{j=1}^{L} \sum_{q=1}^{\kappa_j} \hat{p}_t^{\mathcal{A}_{j,q}} \hat{Z}_t^{\mathcal{A}_{j,q}} \tag{3.3.3}$$

where $\hat{p}_t^{\mathcal{A}_{j,q}}$ are positive weights such that $\forall t \ \sum_{j=1}^{L} \sum_{q=1}^{\kappa_j} \hat{p}_t^{\mathcal{A}_{j,q}} = 1$. The multi-scale weighted aggregation is summarised in algorithm 10. The notation $\mathcal{A}_{1:L}$ means that we consider all the partitions from $\mathcal{A}_1$ to $\mathcal{A}_L$.

input: sequence of partitions $(\mathcal{A}_j)_{1\leq j\leq L}$,weights $\left(\hat{p}_t^{\mathcal{A}_{j,q}}\right)_{j,q,t}$
for $j = 1$ *to* L **do**
  for $q = 1$ *to* κ_j **do**
    Normalise the sub-load for the ith cluster
    Train the forecasting model for the ith cluster
    Forecast the ith cluster on the test set
  end
end
Estimate the system load

Algorithm 10: Multi-scale weighted aggregation.

Two steps multi-scaled weighted aggregation (2S-MSWA)

This last strategy consists of aggregating L single-scale weighted aggregation computed as in section 3.2.2 for each partition in $(\mathcal{A}_j)_{1\leq j\leq L}$:

$$\hat{Y}_{t,\,\text{2S-MSWA}}^{\mathcal{A}_{1:L}} = \sum_{j=1}^{L} \hat{p}_t^{\mathcal{A}_j} \hat{Y}_t^{\mathcal{A}_j} \tag{3.3.4}$$

where $\hat{p}_t^{\mathcal{A}_j}$ are positive weights such that $\forall t \ \sum_{j=1}^{L} \hat{p}_t^{\mathcal{A}_j} = 1$.

The strategy of Wang et al. [2018b] is a sub-case of this one in which $\hat{Y}_t^{\mathcal{A}_j}$ is the sum of the forecast at the level j and the weights are computed based on an optimisation problem based on the minimisation of the mean absolute percentage error.

3.4 Case study

3.4.1 Data

The following data set was published by the Commission for energy regulation Commission for Energy Regulation [2011]. It is composed of 4225 individual loads, each having 48 half hours reports per day over the year 2010. This data set was already studied in many papers among them Quilumba et al. [2015b] and Wang et al. [2018b] for forecasting in the same context smart meters data. We have access to the energy consumption but also other information as the consumer's pricing and the mean temperature in Ireland throughout the year. The location of each client is unknown in order to prevent a link between load and localised weather information to preserve privacy.

The data set is composed of three main categories: residential, small and medium-sized companies and the others. As in Devijver et al. [to appear], to exclude outliers we consider a random subset of 500 individuals from the residential population among the 90% whose loads are the closest to the mean of loads.

More precisely, we consider 487 individuals for which external information (as the pricing) is available. We have 16799 observations of the load for each consumer as well as the global temperature. An example over one day of the 487 loads and the system load we want to predict can be found in Figs. 3.4 and 3.5 (left side). We also present the load over one week individually (respectively for the system load) in Fig. 3.4 (right side) (resp. Fig. 3.5 (right side)).

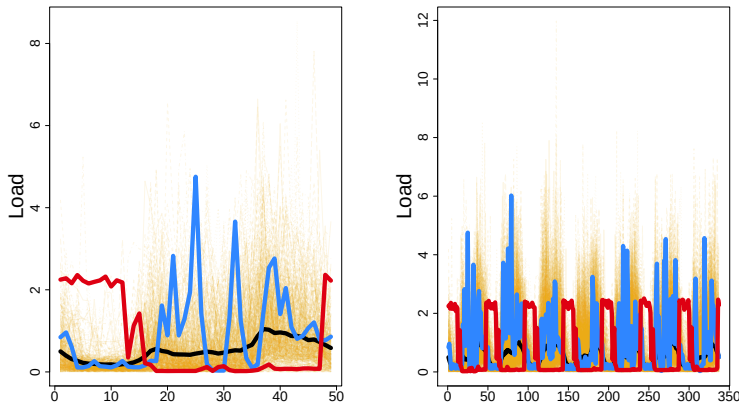

Figure 3.4: Instantaneous load of all the individuals over one day (left) and one week (right). Black line represents the mean load, blue and red line represent two individual loads among the 487.

Besides the load, we have for each consumer i the following vectorial information I_i: Tariffication, Social class, Heat house, Heat water, Double-glassed windows,

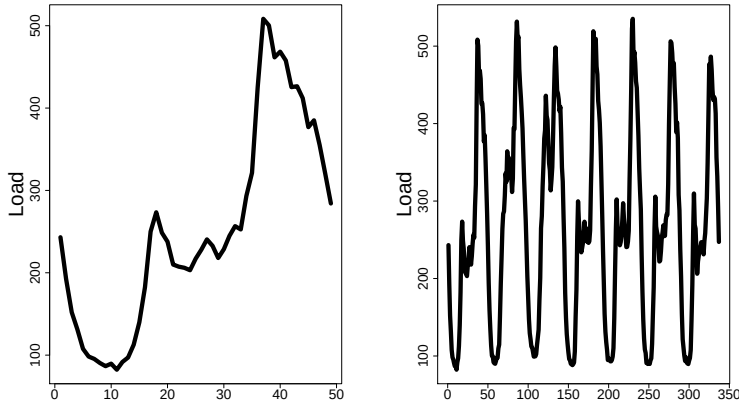

Figure 3.5: Instantaneous system load over one day (left) and one week (right).

Household appliance, Year's building. The choice of these variables is based on domain experience. A further study could be interesting but is not the subject of this work.

Note that in this context, the information $(I_i)_{1\leq i\leq n}$ is not directly related to the load $(Y_{t,i})_{\substack{1\leq i\leq n\\1\leq t\leq T}}$ but could also be done. Regarding this data set we tried to add to $(I_i)_{1\leq i\leq n}$ load features as the load factor, night/lunch impact based on the work in Figueiredo et al. [2005] or the M-shape model from Al-Otaibi et al. [2016] but these failed to do better than the first mentioned vector. Moreover, in the context of energy forecasting it could be more interesting to cluster the individuals independently of their loads, e.g if we get a new consumer, we could directly predict its load based on other information (on a preliminary survey for example).

3.4.2 Experiments

In this experiment X corresponds to the temperature and take $t_Y = t_X = 7$ days (corresponding to 7×48 half-hours) in the model eq. (3.3.1). We also add the instant of the day (which is the number of the half-hour of the day), the day of the week and a binary variable indicating whether the day is a weekend or not. The two last variables seem redundant but some profiles may need finer information regarding the day than others. The robustness of random forests allows us to use both variable and let the method choose the optimal one for forecasting.

Note that in our experiment we consider temperature observations as an entry. We consciously do this as we focus on comparing different forecasting strategies without incorporating meteorological forecast error. In practice, meteorological forecasts can be easily plugged in our models without loss of generality.

We decomposed the dataset in two parts, a training set and a test set. The first 80% (until October) of the data is used to adjust the normalisation's constants and to train the random forest models and the test set is composed of the last 20%. On the test set, we update the aggregation weights each day in a rolling simulation fashion. In order to analyse the procedures we use the root mean square error RMSE defined

as follows:

$$\text{RMSE}\left(Y, \hat{Y}\right) = \sqrt{\frac{1}{|\text{Test set}|} \sum_{t \in \text{Test set}} \left(Y_t - \hat{Y}_t\right)^2}.$$

We also looked at the MAE and MAPE but the results are fundamentally the same as for the RMSE. For consistency purposes, we only give the analysis for the RMSE. We provide at the end of this section the analysis of the small differences for the MAE and MAPE along with a summary of the best performance for each strategy under the different error measures in table 3.1.

The random forests are computed using the package *ranger* Wright and Ziegler [2017]. We set the parameter m_{try} to the number of explanatory variables equal to 17 and set to default the other parameters.

We now discuss the different settings for the strategies. We first address the renormalisation part. For a given partition $\mathcal{A}_j$, we compute a constant $c_{\mathcal{A}_{j,q}}$ for each cluster $\mathcal{A}_{j,q}$ for $q \in [1, \dots, \kappa_j]$ such that the following equality is verified:

$$\sum_{t \in \text{Train set}} Y_t = c_{\mathcal{A}_{j,q}} \sum_{t \in \text{Train set}} Y_t^{\mathcal{A}_{j,q}}. \tag{3.4.1}$$

We consider three methods to compute the $\hat{p}_t^{\mathcal{A}_{j,q}}$ for the strategies. Since the load of the clusters are of the same order as the system load, we can simply use the uniform weights:

$$\hat{p}_t^{\mathcal{A}_{j,q}} = \hat{p}^{\mathcal{A}_{j,q}} = \frac{1}{\kappa_j} \quad \forall q \in [1, \dots, \kappa_j].$$

We also consider weights proportional to the number of individuals per cluster:

$$\hat{p}^{\mathcal{A}_{j,q}} = \frac{N(\mathcal{A}_{j,q})}{\sum_{q=1}^{\kappa_j} N(\mathcal{A}_{j,q})}$$

where $N(\mathcal{A}_{j,q})$ denotes the number of individuals belonging to the group $\mathcal{A}_{j,q}$. This choice comes from the intuition that if a cluster has a large number of individuals relatively to the total number of individuals its load have a greater impact than the other ones. Note that the estimator with proportional weights is really close to the first strategy, the information of the system load being somehow included in the model by the normalisation.

The last choice is to compute the weights with the sequential aggregation method recalled in section 3.2.2.

Weights computation

The first question we want to address is relative to the weights computing methods. We compare the SSWA using the set of weights expressed before to a baseline and to the simple aggregation strategy. The baseline is the model in which we forecast the system load Y_t by a random forest following the model eq. (3.3.1) as if there were only one class, i.e using directly the system load for the load lags.

We quickly notice that for the strategy SSWA the only interesting weights to consider are the ones optimised by the algorithm *ML-Poly* (Fig. 3.6). We can also note

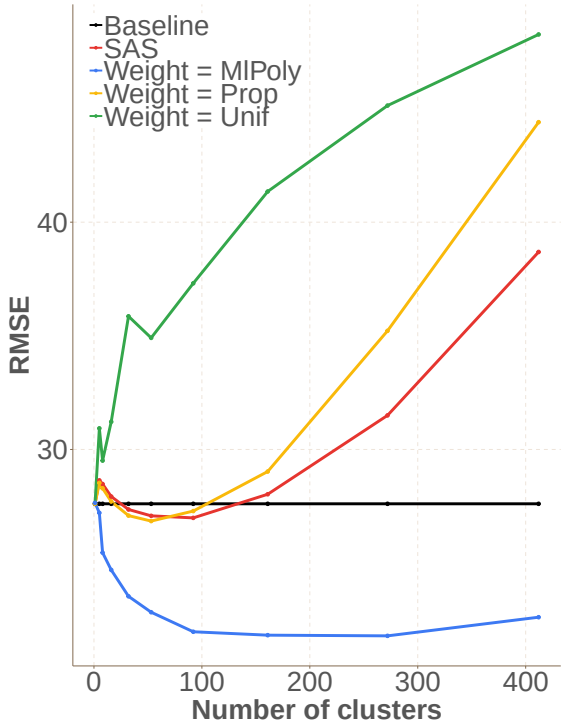

Figure 3.6: RMSE comparison of the strategies according to the number of clusters, baseline *vs* SSWA.

that the first strategy and the SSWA with sequential weights reach their minimum for the same number of clusters but the latter having a lower RMSE. The other observation we make is that except the sequential weights the intuitive strategy and the other weight choices get quickly worse than the baseline.

However, when the number of experts increases the interpretability of the sequential weights is harder whereas the static weights are easily understood.

Contribution of the clustering

An arising question is about the usefulness of the clustering; is the HAC depending on $(I_i)_{1\leq i\leq n}$ of significant impact on the forecasting performance or could a simple disaggregation be enough? We compare the HAC to a random disaggregation. More precisely, for a given number of clusters κ_j, we construct $\mathcal{A}_{j,q}$ for $q \in \{1,\dots,\kappa_j\}$ such that each individual $i \in \{1,\dots,n\}$ is attributed uniformly among the κ_j clusters (without replacement) i.e

$$\mathbb{P}\left(\text{Individual } i \in \mathcal{A}_{j,q}\right) = \frac{1}{\kappa_j}.$$

Note that we lose the hierarchical characteristic with this disaggregation as well as the number of individuals in each cluster (relatively to HAC clustering). However varying κ_j we have a sequence of partitions and we can apply the bottom-up forecasting strategies. We choose to disaggregate this way to compare the HAC to a form of disaggregation in which there is not a single information about whatsoever other than the number of clusters. It should be noted that, for a given level of partition, the following results are based on one run over the data.

We compare the three strategies: SSWA, MSWA and 2S-MSWA with the weights sequentially optimised using *ML-Poly* according to the results in section 3.4.2 using the HAC and the random disaggregation (Fig. 3.7). We note that all three strategies under the HAC give smooth decreasing errors when the number of clusters increases and go slightly back up when the number of clusters is close to the number of individuals for the strategies SSWA and MSWA (Fig. 3.7 blue lines). Regarding the random disaggregation, the SSWA's error (Fig. 3.7 red continuous line) follows with a few fluctuations the SSWA error's curve under the HAC. The other two strategies are again smoother and do even better than under the HAC. In both cases we have an optimal point which is located around 100 clusters for the SSWA strategy and 400 clusters for the 2S-MSWA strategy. Besides the 2S-MSWA strategy, the errors for both clustering procedures get slightly worse when the number of clusters gets closer to the number of individuals but note that this rising is almost negligible considering the number of individuals we try to forecast in each cluster. Indeed, at this point we use the load of one individual to model the system load and yet combining the forecasts in a intelligible way leads to a good performance. On Fig. 3.7, we see that up to 50 clusters all the methods benefit equally from disaggregation. For finer partitions, we observe a difference between single scale and multi-scale strategies. Indeed, for single scale strategies, increasing the number of clusters leads to increase the forecasting error for both random and hierarchical partitions. On the contrary, multi-scale strategies benefit from finer disaggregation, especially for the 2S-MSWA strategies.

We get a decrease in the error of the forecast of the system load in both ways of clustering and a slightly better performance under the random disaggregation. We conclude that the disaggregation is definitely useful but not as we thought, a random partition can be as good or even better than an HAC.

Which forecasting strategy is the best

Recall that the goal is to get the best forecast of the system load. According to the previous answers, we can let down the interpretability since we can take a random partition and still do better than an HAC and in both cases the best results are for a high number of clusters.

The idea of the second strategy MSWA is to combine up to some level all the forecasts associated to the clusters. It is obvious that doing so we improve significantly the forecast of the system load (continuous *vs* dotted lines Fig. 3.7). The two-step strategy 2S-MSWA, when looking at the RMSE, improves the forecast even more being less penalised by the number of experts but comparing different error measures (table 3.1) both multi-scale strategies are fairly equivalent in terms of performance. The conclusion of this point is that it is always useful to add coarser partitions to the aggregation and get extreme gains compared to a single scale strategy.

Summing up the weights for each level obtained for the MSWA strategy (Fig. 3.10) we note that the finer partition weights is always higher than the coarser partitions weights. When looking at the mean over time of the weights per level for the 2S-MSWA (Fig. 3.8), we observe that the more we add finer partitions the less the previous important level are and it seems that the weights converge to uniform weights over the partitions when κ_j converges to the number of individuals: there is no useless partition. Also note that there is a persistent V-shape in the weights and this

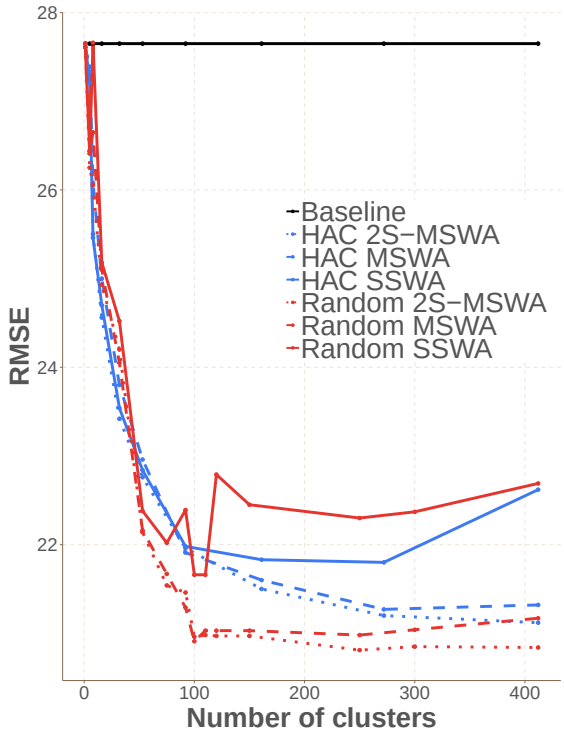

Figure 3.7: RMSE comparison of the strategies according to the number of clusters, HAC *vs* random clustering.
For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.

is probably due to the hierarchical structure of the expert clusters: the forecasts obtained at one resolution are close to the sum of the forecasts of the scale below. This phenomenon disappears when we use the random disaggregation. The partitions having around 100 individuals contribute the most and the coarser partitions get almost no weights.

We conclude that both strategies MSWA and 2S-MSWA give good forecasts close to the real load even though there is a high variance among the single forecasts of the clusters (transparent curves in Fig. 3.11). A summary of the best performance for each strategy is given in table 3.1. When looking at different error measures as the MAPE or MAE, the only noted difference is that the best performance for the SAS, SSWA HAC and 2S-MSWA Random are obtained for a smaller number of clusters but the previous results are still verified. Due to space limitations, we omit the graphics but give the best performance for the two other measures in table 3.1.

3.4.3 Discussion and future work

We showed that a simple random disaggregation may be as good as an HAC. We may question the choice of I_i in the experiment. However we tried the case where I_i incorporate information about the load (based on the feature of Figueiredo et al. [2005] or the M-shape model from Al-Otaibi et al. [2016]) but no gain was noticeable and confirms the previous point. We think it is due to the nature of the individuals which are in this case quite homogeneous. We think that in the case where individuals are more different (e.g dataset with loads of companies and residentials)

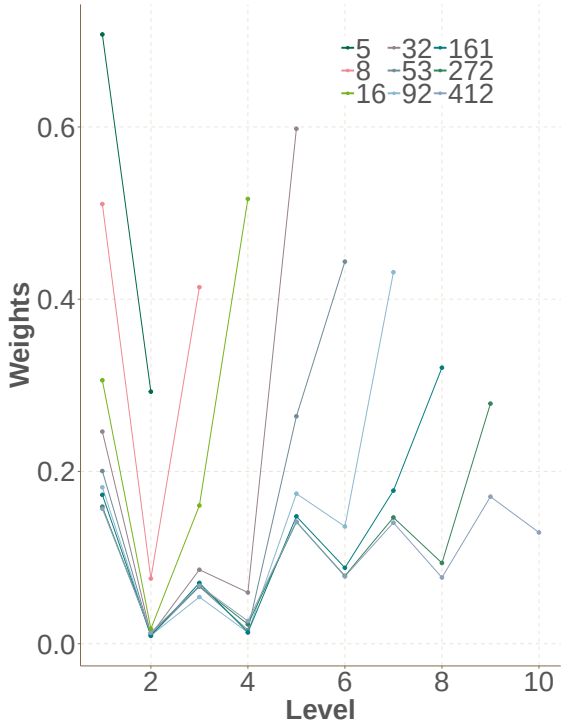

Figure 3.8: Mean over time of the weights according to the levels for the 2S-MSWA HAC.

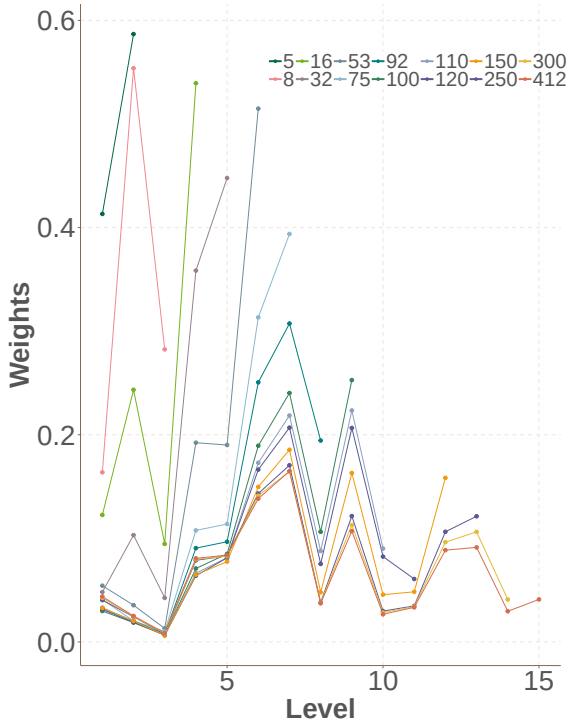

Figure 3.9: Mean over time of the weights according to the levels for the 2S-MSWA Random.

an HAC may be more contributional. Nevertheless, even though individuals are homogeneous, it is always useful to disaggregate.

We also considered the strategy from Wang et al. [2018b]. We decomposed the

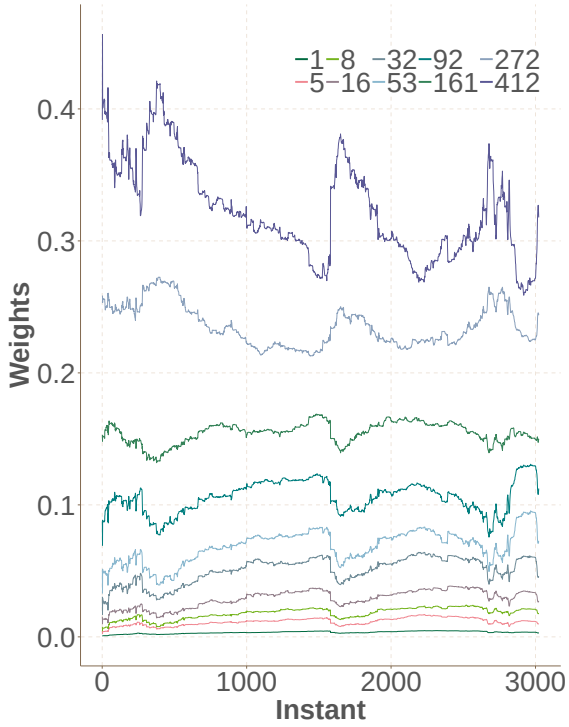

Figure 3.10: Weights according to the levels when $k_K = 412$.

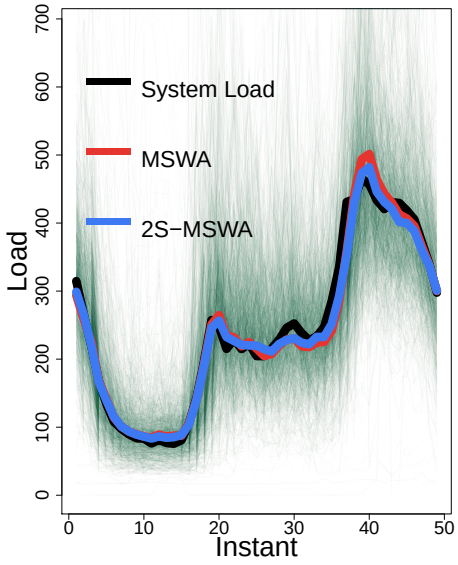

Figure 3.11: Measured one day load and prediction of the values using the multi-scale strategies for the random disaggregation.

dataset in 60% for the training of the forecasts (using random forests instead of ANN), 20% for the optimisation of the weights and 20% for the test, the rest remaining the same (including the clustering phase). Their optimisation problem is based on the minimisation of the MAPE. It does better than the baseline when looking at the MAPE (see table 3.1) but fail to exploit information coming from finer partitions and leads to worse performance than our three strategies. Many reasons

Table 3.1: Recap of the best RMSE and MAE for each strategy

	Base-line	SAS	Strategy Wang et al. [2018b]	SSWA	MSWA HAC	MSWA Ran-dom	2S-MSWA HAC	2S-MSWA Random
Best RMSE	27.61	26.99 (92)	28.89	21.97 (272)	21.27 (272)	20.91 (100)	21.12 (412)	20.81 (250)
Best MAE	19.28	19.17 (32)	19.82	15.28 (92)	14.78 (272)	14.18 (100)	14.64 (412)	14.29 (120)
Best MAPE in %	7.43	7.04 (32)	7.25	6.02 (92)	5.72 (272)	5.5 (100)	5.66 (412)	5.51 (100)

may lead to such a distinct performance. One being that the weights in their strategy are constant through time whereas in ours they are adapted each day according to the past.

In this paper we also studied the weights to understand which partitions contribute in the forecasts. We observe a persistent V-shape in the weights for the HAC partitioning highlighting the complementary between coarser and finer partitions not noticeable with the random partitions. An interesting future work would be to analyse the variable importance throughout the clusters which is easily computable in the case of random forests and see if there are variables which are more dominant and see if some clusters are more thermosensitive (or others external factors dependent) than the others.

Acknowledgment

The authors acknowledge CER Smart Metering Project for Electricity Customer Behaviour Trial, 2009-2010 data accessed via the Irish Social Science Data Archive - www.ucd.ie/issda.

3.5 Bibliography

R. Al-Otaibi, N. Jin, T. Wilcox, and P. Flach. Feature construction and calibration for clustering daily load curves from smart-meter data. *IEEE Transactions on Industrial Informatics*, 12:645–654, 2016. 74, 78

L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996. 68

L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001. 65, 68

L. Breiman, J. Friedman, C. Stone, and R. Olshen. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984. 68

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006. 65, 69, 70

T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016. 69

Commission for Energy Regulation. Electricity smart metering customer behaviour trials (cbt) findings report. http://www.cer.ie/docs/000340/cer11080(a)(i).pdf, 2011. 65, 73

M. Devaine, Y. Goude, and G. Stoltz. Forecasting the electricity consumption by aggregating specialized experts; a review of the sequential aggregation of specialized experts, with application to Slovakian and French country-wide one-day-ahead (half-)hourly predictions. preprint: hal-00484940, 2011. 69

E. Devijver, Y. Goude, and J.-M. Poggi. Clustering electricity consumers using high dimensional regression mixture models. *Applied Stochastic Models in Business and Industry*, to appear. 73

G. Dudek. Short-term load forecasting using random forests. In *Intelligent Systems'2014*, volume 323, pages 821–828, Cham, 2015. Springer International Publishing. 66, 69

V. Figueiredo, F. Rodrigues, Z. Vale, and J. B. Gouveia. An electric energy consumer characterization framework based on data mining techniques. *IEEE Transactions on Power Systems*, 20:596–602, 2005. 74, 78

A. Fischer, L. Montuelle, M. Mougeot, and D. Picard. Statistical learning for wind power: A modeling and stability study towards forecasting. *Wind Energy*, 20: 2037–2047, 2017. 69

D. Fischer, J. Scherer, A. Flunk, N. Kreifels, K. Byskov-Lindberg, and B. Wille-Haussmann. Impact of hp, chp, pv and evs on households' electric load profiles. In *2015 IEEE Eindhoven PowerTech*, pages 1–6, 2015. 64

J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 29:1189–1232, 2001. 69

P. Gaillard and Y. Goude. Forecasting electricity consumption by aggregating experts; how to design a good set of experts. In *Modeling and stochastic learning for forecasting in high dimensions*, volume 217, pages 95–115. Springer, 2015. 70

P. Gaillard and Y. Goude. Opera: Online prediction by expert aggregation. r package, 2016. 70

P. Gaillard, G. Stoltz, and T. van Erven. A second-order bound with excess losses. In *Proceedings of The 27th Conference on Learning Theory*, volume 35 of *Proceedings of Machine Learning Research*, pages 176–196. PMLR, 2014. 70

P. Gaillard, Y. Goude, and R. Nedellec. Additive models and robust aggregation for gefcom2014 probabilistic electric load and electricity price forecasting. *International Journal of Forecasting*, 32:1038 – 1050, 2016. 70

B. Goehry. Random forests for time-dependent processes. preprint: hal-01955331, 2018. 69

J. C. Gower and J. C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, 1971. 66

T. Hastie and R. Tibshirani. Generalized additive models. *Statist. Sci.*, 1:297–310, 1986. 68

M. J. Kane, N. Price, M. Scotch, and P. Rabinowitz. Comparison of arima and random forest time series models for prediction of avian influenza h5n1 outbreaks. *BMC Bioinformatics*, 15:276, 2014. 69

H. Kikusato, K. Mori, S. Yoshizawa, Y. Fujimoto, H. Asano, Y. Hayashi, A. Kawashima, S. Inagaki, and T. Suzuki. Electric vehicle charge–discharge management for utilization of photovoltaic by coordination between home and grid energy management systems. *IEEE Transactions on Smart Grid*, 10:3186–3197, 2019. 64

A. Lahouar and J. Ben Hadj Slama. Random forests model for one day ahead load forecasting. In *IREC2015 The Sixth International Renewable Energy Congress*, pages 1–6, 2015. 66

F. Lezama, J. Soares, P. Hernandez-Leal, M. Kaisers, T. Pinto, and Z. M. A. do Vale. Local energy markets: Paving the path towards fully transactive energy systems. *IEEE Transactions on Power Systems*, pages 1–1, 2018. 64

J. Moon, Y. Kim, M. Son, and E. Hwang. Hybrid short-term load forecasting scheme using random forest and multilayer perceptron. *Energies*, 11:3283, 2018. 66

D. Niu, D. Pu, S. Dai, et al. Ultra-short-term wind-power forecasting based on the weighted random forest optimized by the niche immune lion algorithm. *Energies*, 11:1–21, 2018. 69

F. L. Quilumba, W. Lee, H. Huang, D. Y. Wang, and R. L. Szabados. Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities. *IEEE Transactions on Smart Grid*, 6:911–918, 2015a. 65

F. L. Quilumba, W.-J. Lee, H. Huang, D. Y. Wang, and R. L. Szabados. Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities. *IEEE Trans. Smart Grid*, 6:911–918, 2015b. 73

R. Schapire and Y. Freund. *Boosting: Foundations and Algorithms*. Adaptive computation and machine learning. MIT Press, 2012. 69

R. Sevlian and R. Rajagopal. A scaling law for short term load forecasting on varying levels of aggregation. *International Journal of Electrical Power & Energy Systems*, 98: 350 – 361, 2018. 65

H. Shareef, M. S. Ahmed, A. Mohamed, and E. A. Hassan. Review on home energy management system considering demand responses, smart technologies, and intelligent controllers. *IEEE Access*, 6:24498–24509, 2018. 65

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58:267–288, 1996. 68

L. Wang. *Support Vector Machines: Theory and Applications*. Studies in Fuzziness and Soft Computing. Springer Berlin Heidelberg, 2005. 68

Y. Wang, Q. Chen, C. Kang, Q. Xia, and M. Luo. Sparse and redundant representation-based smart meter data compression and pattern extraction. *IEEE Transactions on Power Systems*, 32:2142–2151, 2017. 65

Y. Wang, Q. Chen, T. Hong, and C. Kang. Review of smart meter data analytics: Applications, methodologies, and challenges. *IEEE Transactions on Smart Grid*, 10: 1–1, 2018a. 65

Y. Wang, Q. Chen, M. Sun, C. Kang, and Q. Xia. An ensemble forecasting method for the aggregated load with subprofiles. *IEEE Transactions on Smart Grid*, 9:3906–3908, 2018b. 65, 73, 79, 81

S. N. Wood. *Generalized additive models: an introduction with R*. Chapman and Hall/CRC, 2017. 68

M. Wright and A. Ziegler. ranger: A fast implementation of random forests for high dimensional data in c++ and r. *Journal of Statistical Software, Articles*, 77:1–17, 2017. 75

Chapter 4

A variant of random forests for time series

In collaboration with Hui YAN, *Yannig* GOUDE, *Pascal* MASSART *and Jean-Michel* POGGI

Abstract

Random forests were introduced in 2001 by Breiman and have since become a popular learning algorithm, for both regression and classification. However, when dealing with time series, random forests do not integrate the time-dependent structure, implicitly supposing that the observations are independent. We propose a variant of the random forests designed for time series. The idea is to replace the standard bootstrap with dependent bootstrap to subsample time series during the tree construction phase to take time dependence into account.

Sommaire

4.1 Introduction

Random forests were introduced in 2001 by Breiman in Breiman [2001] and are since then one of the most popular algorithms in machine learning Fernández-Delgado et al. [2014]. The popularity comes from the wide range of applications in which they are known to perform well on even high dimensional, are fast to compute and easy to tune. Successful applications can be cited: chemo-informatics Svetnik et al. [2003], ecology Cutler et al. [2007]; Prasad et al. [2006], 3D object recognition Shotton et al. [2013] and time series prediction Dudek [2015]; Fischer et al. [2017]; Kane et al. [2014]; Lahouar and Ben Hadj Slama [2015]; Moon et al. [2018].

Suppose that we have a random sequence $(X_t, Y_t)_{t\in\mathbb{Z}} \in \mathcal{X} \times \mathcal{Y}$ such that

$$Y_t = f(X_t) + \epsilon_t \tag{4.1.1}$$

and the error ϵ_t is such that $\mathbb{E}[\epsilon_t|U_t] = 0$. The purpose of random forests is to estimate, by only observing a training sample $\mathcal{D}_n = ((X_1, Y_1), \ldots, (X_n, Y_n))$, the regression function

$$\forall x \in \mathcal{X}, f(x) = \mathbb{E}[Y_t|X_t = x].$$

Random forests can be related to two main sources, regression trees Breiman et al. [1984] and bagging Breiman [1996]. Regression trees are constructed by a recursive partitioning of the input space based on some criterion to estimate the regression function f. At each step of the tree construction, a split is selected (a variable and a location on the variable) based on the evaluation of the criterion among all the admissible splits based on all the variables. The cell is cut in two on the selected split and the previous step is reiterated on the new cells. A tree is then a piecewise constant decomposition of the input space. We can associated to the input space partitioning a binary tree where each node corresponds to a test matching how the input space was cut. An illustration is given in Fig. 4.1 of a partitioning in the two-dimensional space and its associated binary tree. The principle of bagging (short form of bootstrap aggregating) is to create M randomly generated training sets by randomly sampling a_n observations with or without replacement from the set $\mathcal{D}_n$ and to construct on each set a predictor. Once the predictors are constructed, the bagging prediction for a new observation x is an aggregation, generally the empirical mean, of the predictions given by the M predictors for the point x. This procedure aims to improve stability and accuracy of the base predictor. In the context of random forests the predictors are regression trees. In order to explain the random forest procedure we then have to explicit the construction of one tree.

The first step is the bootstrap/subsampling: a_n points are selected with or without replacement among the n realisations. Then a tree is constructed based on these a_n selected points. At each node of the tree the best split (the variable and the location on this variable) is determined by minimising the intra-node variance. This is commonly called the CART criterion introduced in Breiman et al. [1984]. Instead of minimising this criterion among all the admissible splits based on all the variables the choice of inputs is restricted to a random subset of fixed size m_{try}. This procedure is then iterated on each node produced after binary splitting until stopping conditions are met. The first stopping rule is when the variance in a node is equal to zero. Since this is rarely the case a second condition is that the number of observations in a node must be greater than a given threshold.

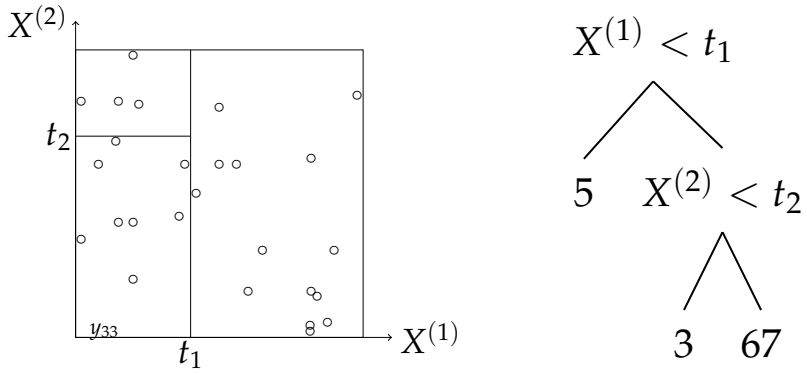

Figure 4.1: A partitioning of $[0,1]^2$ and the associated binary tree.

Even if the theoretical settings of random forests was until recently restricted to the i.i.d case, a theoretical study extending it to the time-dependent case is proposed in Goehry [2018]. In addition, applications on time series could be found, as previously cited, in Fischer et al. [2017]; Kane et al. [2014], in electricity load forecasting Dudek [2015], Lahouar and Ben Hadj Slama [2015], Moon et al. [2018].

The bootstrap step determines which observations are chosen to construct a tree. The original bootstrap which we call standard (or i.i.d) bootstrap from Efron [1979] consists of randomly drawing α_n observations among the n with or without replacement. Note that we use here an abuse of language, the bootstrap is standardly defined as drawing n observations among the n observations with replacement. The goal of this bootstrap is to replicate the distribution of $\mathcal{D}_n$. However, this is adapted to the case of independent and identically distributed observations. When the data has an underlying dependence structure as for time series the i.i.d hypothesis is not verified anymore and using the standard bootstrap destroys the dependent structure. We illustrate this phenomenon for a dataset from Miller and Meggers [2017] which is described in section 4.3.1. We observe in Fig. 4.2 the original load over the month of January. Using the standard bootstrap we obtain the series in Fig. 4.3 and immediately note that the structure we had in the original series is all gone. By contrast, using a moving block bootstrap, described in section 4.2, using a block length of 24 hours we recover similar patterns as in the original series of Fig. 4.4.

We list here a few papers using blocks bootstrap in the forecasting literature. The first one is Cordeiro and Neves [2009] in which they use a sieve bootstrap to perform bagging with exponential smoothing models. They use exponential smoothing to decompose the data, then fit an autoregressive model to the residuals, and generate new residuals from this AR process. Finally, they fit the exponential smoothing model that was used for decomposition to all bootstrapped series. Another work is from Bergmeir et al. [2016] who propose a method of bagging which is as follows. After applying a Box-Cox transformation to the data, the series is decomposed into trend, seasonal and remainder components. The remainder component is then bootstrapped using the moving block bootstrap, defined in section 4.2, the trend and seasonal components are added back, and the Box-Cox transformation is inverted. For each one of these bootstrapped time series, a model among several exponential smoothing models is chosen, using the bias-corrected AIC. Then, point forecasts are calculated using all the different models and the resulting forecasts are combined using the median. We refer to Cavaliere et al. [2015] for more details about the re-

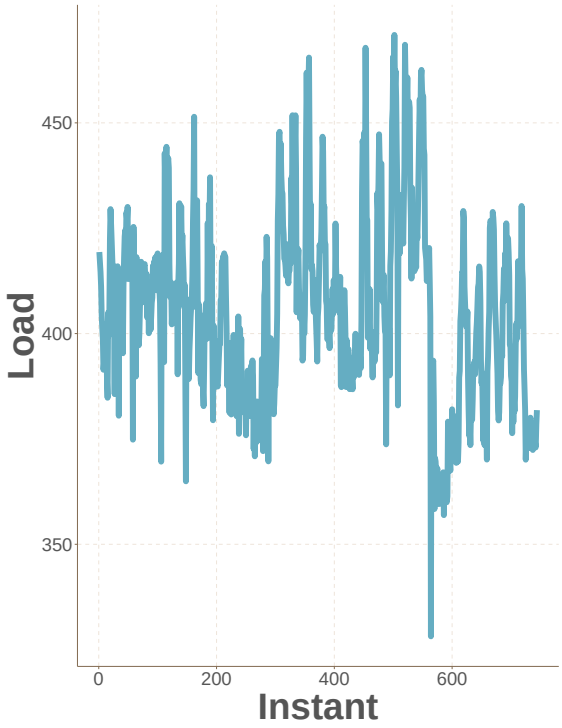

Figure 4.2: Original load.

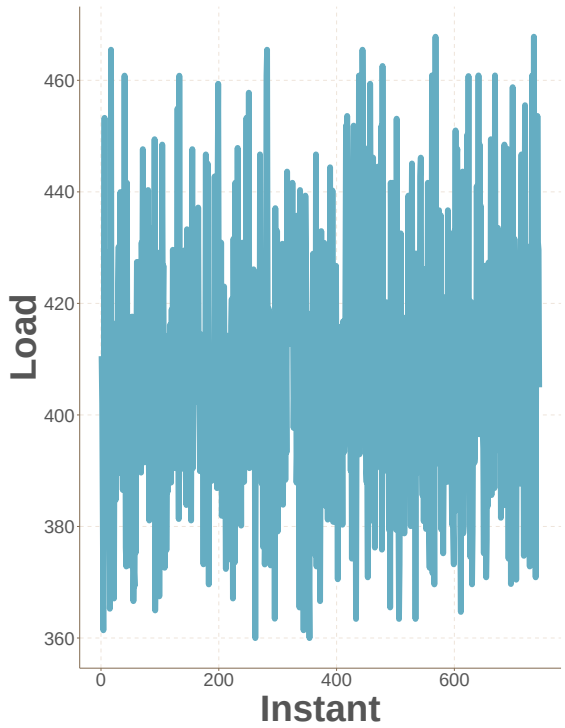

Figure 4.3: Bootstraped load.

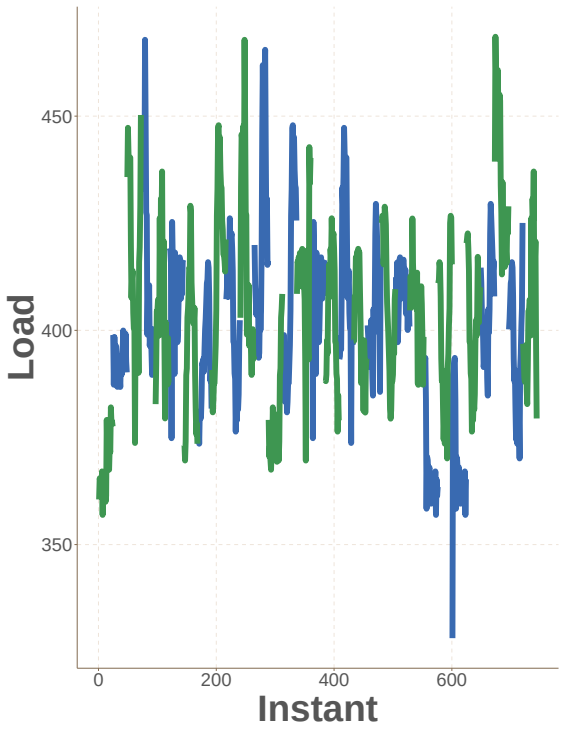

Figure 4.4: Block bootstrapped load with block size of 24h.

cent developments in bootstraps methods for dependent data.

The aim of this work is to show that the forecasting performance could be improved by replacing the bootstrap step by what we call block bootstrap variants, to subsample time series during the tree construction phase and thereby keep the dependent structure. Since random forests were already introduced in this introduction. The next section presents the different block bootstrap variants, the new algorithm and a new way to compute the variable importance. We then present two numerical experiments, the first one is based on a simulated time series and the second is an application to load forecasting on a real world dataset.

4.2 Random forests for time series

4.2.1 Block bootstrap variants

Non-overlapping block bootstrap A first variant is found in Carlstein [1986]: the *non-overlapping block bootstrap*. The idea is to construct a number of non-overlapping blocks and then to draw uniformly, with replacement, among the constructed blocks. More precisely, let l_n be the size of a block and $B \geq 1$ the greatest integer such that $l_n B \leq n$. The blocks are then constructed the following way

$$B_b = \left(X_{(b-1)l_n+1}, \ldots, X_{bl_n}\right) \quad b = 1, \ldots, B.$$

The bootstrap set $\mathcal{D}_n^\star$ is then obtained by drawing K blocks, $(B_1^\star, \ldots, B_K^\star)$, uniformly with replacement in the collection of non-overlapping blocks$(B_b)_{1\leq b\leq B}$ for a suitably chosen K.

Moving block bootstrap Kunsch [1989] and Liu and Singh [1992] introduced the so-called *moving block bootstrap*. The idea is, instead of picking randomly one observation among the n observations as for the standard bootstrap, the moving block bootstrap pick randomly a block of l_n consecutive observations. Repeating this step and concatenating all the selected blocks, we get a new time series with a preserved structure at least in each block. More precisely, let us denote by $B_{i,l_n} = ((X_i, Y_i), \ldots, (X_{i+l_n-1}, Y_{i+l_n-1})$ the block of size l_n beginning with the observation (X_i, Y_i) for $i \in \{1, \ldots, n-l+1\}$. The procedure then consists to draw randomly K indices $(I_j)_{1\leq k\leq K}$ uniformly on the set $\{1, \ldots, n-l_n+1\}$ and associate one block to each index, $(B_{I_k})_{1\leq k\leq K}$. The bootstrap set is then defined as $\mathcal{D}_n^\star = (B_{I_1}, \ldots, B_{I_K})$.

Circular block bootstrap When studying the moving block bootstrap we can note that less weight is given to certain parts of the time series which also leads in theory to non negligible bias when computing the mean. A way to correct this issue is given in Politis and Romano [1992] introducing the so-called *circular block bootstrap*. The idea is to wrap the time series writing $X_i := X_{i_n}$ where $i_n = i \mod n, X_0 := X_n$ and then use the same procedure as in the moving block bootstrap where the index I is drawn uniformly on the set $\{1, \ldots, n\}$ instead.

Note that in each above variant, taking $l_n = 1$ we recover the standard bootstrap of Efron [1979]. For a given number of selected observations in each tree a_n the number of blocks K is such that $K = \frac{a_n}{l_n}$.

4.2.2 Proposed random forest for time series

Our proposition in order to incorporate the dependence structure is by replacing the first step for the construction of a random tree in the random forest building procedure, namely replacing the standard bootstrap step with one of the block bootstrap variant recalled in section 4.2.1. The adapted algorithm is found in algorithm 11 highlighting in blue the novelty.

input: $((X_1, Y_1), \ldots, (X_n, Y_n))$
parameters: $M, \alpha_n, m_{try}, \tau_n, l_n$
stopping criteria: the variance in the node is zero or the number of observations in a node is below the threshold τ_n
for $j \leftarrow 1$ *to* M **do**
Construct the jth tree:

- Draw $\alpha_n \leq n$ observations using a block bootstrap variant with parameter l_n.
- Repeat recursively on each resulting node the following steps until a stopping criterion is met:
  - At each node, select randomly m_{try} variables
  - Select the best split using the variance criterion among the previously chosen variables.
  - Cut on the chosen split.

end
output for a new observation x: mean of the M predictions given by the trees for x.

Algorithm 11: Random forest for time series.

4.2.3 Bloc permutation importance

Random forests can be used to rank with respect to a decreasing order of importance the variables. One way to measure the significance of a variable is the *Mean Decrease Accuracy* introduced in Breiman [2001] which stems from the idea that if a variable is not important, then permuting its value should not change the prediction accuracy.

For each tree, we have access to the so-called *out-of-bag* observations denoted by OOB_m, composed of the observations not included in the bootstrap sample $\mathcal{D}_n^m$ used to construct the mth tree. The OOB_m sample can then be used to estimate the out-of-bag error denoted by $errOOB_m$. In order to compute the importance of the variable $X^{(j)}$, the values of the jth variable are randomly permuted in the OOB sample and compute for each tree an out-of-bag error estimation for the permuted observations. The importance of the variable $X^{(j)}$ is then obtained by averaging the difference between the out-of-bag error before and after permutation. More formally, if, for the mth tree, we denote by $\widetilde{errOOB}_m^j$ the OOB_m sample's error when the jth variable is permuted, then the importance of the variable $X^{(j)}$ is defined by

$$VI\left(X^{(j)}\right) = \frac{1}{M}\sum_{m=1}^{M}\left(\widetilde{errOOB}_m^j - errOOB_m\right).$$

The higher the increase in the prediction error after the permutation of the jth variable in the out-of-bag observations, the more important the variable is. However, if the permutation of $X^{(j)}$ doesn't change much the error prediction then the importance of the considered variable is small.

In the case of dependent observations we are faced with the same issue as in the construction of the random forests, namely the permutation of variable in the out-of-bag observations does not preserve the dependent structure. In the case where block instead of standard bootstrap is used in the random forest we introduce a new variable importance computation: the *block (permutation) variable importance*. However, using a block bootstrap variant doesn't necessarily lead to a out-of-bag observations with constant number of consecutive observations but we solve this issue in the following. Let us first suppose that the out-of-bag observations can be separated in blocks of size of the block size parameter in the forest l_n and denote by B_m^* the blocks in the out-of-observations for the mth tree. In order to compute the importance of the jth variable, the permutation of the considered variable is done by only permuting the blocks in B_m^* and preserving the structure in each block. We can then compute a block permuted out-of-bag error estimation for the jth variable denoted by $\overline{errOOB_m^j}$. The block variable importance for the jth variable is then defined by

$$VI\left(X^{(j)}\right) = \frac{1}{M}\sum_{m=1}^{M}\left(\overline{errOOB_m^j} - errOOB_m\right).$$

The out-of-bag observations stemming from the block bootstrap with parameter l_n is not necessarily composed of blocks of the size l_n but the non-overlapping block bootstrap. In order to obtain an OOB sample which has the same block size as in the construction of the random forest we adapt the obtained out-of-bag observations to get a new set of blocks of out-of-bag observations. The construction of the latter is as it follows. If a block of consecutive observations in the out-of-bag observations is of the right length l_n we add it to the block out-of-bag observations and if the length is larger than l_n and less than $2l_n$ we draw a random subset of consecutive observations of length l_n. If a block of consecutive observations in the out-of-observations has a length less than l_n then the block is not kept. Then the block out-of-bag observations is composed of the kept block observations of length l_n and satisfies the conditions to compute the block permutation variable importance as previously defined.

4.3 Numerical experiments

We consider two experiments in this work, one is a simulation study, the other is a real world application of load forecasting on one of the building dataset from Miller and Meggers [2017] which is composed of different building loads with hourly observations.

We run the experiments by implementing the extra features we propose in this paper as an extension of the R package *ranger* Wright and Ziegler [2017] (see appendix C), and thus inherit the availability in both C++ and R. In the following experiments, the results are obtained over 50 runs. The parameters of the random forest are set to default excepted the parameter m_{try} which is optimised on a validation set and the block size parameter for which we carry out an in-depth analysis.

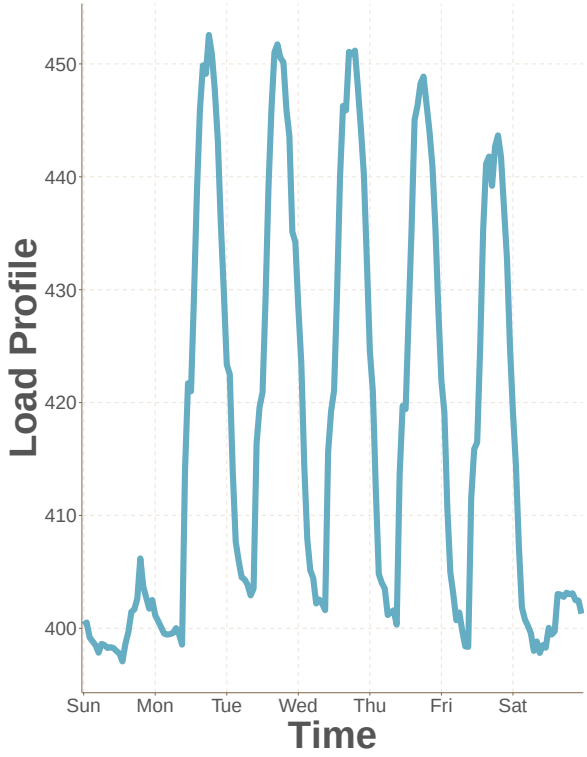

Figure 4.5: Weekly profile hourly sampled of the UnivLab Patrick dataset.

4.3.1 Load forecasting application

This experiment is based on the so-called building loads, a collection of 507 whole buildings electrical meters made publicly available. We refer to the paper Miller and Meggers [2017] for a complete description of the collection. We consider one specific building in the building data genome project called *UnivLab Patrick.* This building belongs to the college laboratory category located in the New York time zone and has an area of around 7054 square meters. We have access to its electricity load from the 1st January 2015 to the 31th December 2015 with a sampling rate of one observation per hour. The weekly profile is found in Fig. 4.5. We also have access to exogenous variables: the temperature as well as to the schedule of the building, indicating if a day is ordinary, a break or a holiday.

Let us denote by Y_t the system load of the building at hour t. In this experiment, we aim to forecast at a horizon of 24 hours using the model described in eq. (4.1.1) where X_t is of the form

$$X_t = (Y_{t-24}, Y_{t-168}, \text{Temp}_t, \text{Schedule}_t, \text{Hour}_t, \text{InstantWeek}_t, \text{DayType}_t, \text{Toy}_t)$$

with

- Temp_t corresponds to the temperature at instant t;
- Schedule_t take three values: Regular, Break, Holiday;
- Hour_t corresponds to the hour of the day at instant t;
- InstantWeek_t corresponds to the hour in the month;

- DayType_t corresponds to the day of the week;
- Toy_t corresponds to the day of the year divided by 366.

4.3.2 Results

The selected value for m_{try} according to the performance on the validation set is $m_{try} = 2$. For this parameter we computed the different variants varying the block size parameters multiple of 6 hours up to 90 hours. We first optimise the performances on the validation set, looking for the best block size value minimising the RMSE and then plug it in for the test set. The performance are resumed in Fig. 4.6. We observe an improvement for the three variants with an improvement up to 11% for the mean RMSE compared to the standard random forest. We also show the evolution of the performance according to the block size parameter in Fig. 4.7. We observe for the three variants a similar pattern in the evolution of the performance, namely a decrease for which the three variants performs better than the standard random forest and then an increase. We note that, even if the performance get worse when the block size is large, we also have a large window for which the performance is far better for these three variants with an optimal block size parameter of around 24 hours also corresponding to the forecasting horizon.

Computing the variable importance for blocks of size 24 hours we obtain Figs. 4.8a to 4.9b. We observe that the difference between the standard variable importance and the block variable importance is essentially noticeable for the non-overlapping block bootstrap variant. The most evident difference is for the variable *Hour* for which the importance is set to zero using the block variable importance. Since the blocks are of length 24 hours and always beginning at the same time, permuting the blocks won't change the out-of-bag error since each permutation is replaced by an identical copy and thus the output from this procedure for the variable *Hour.*

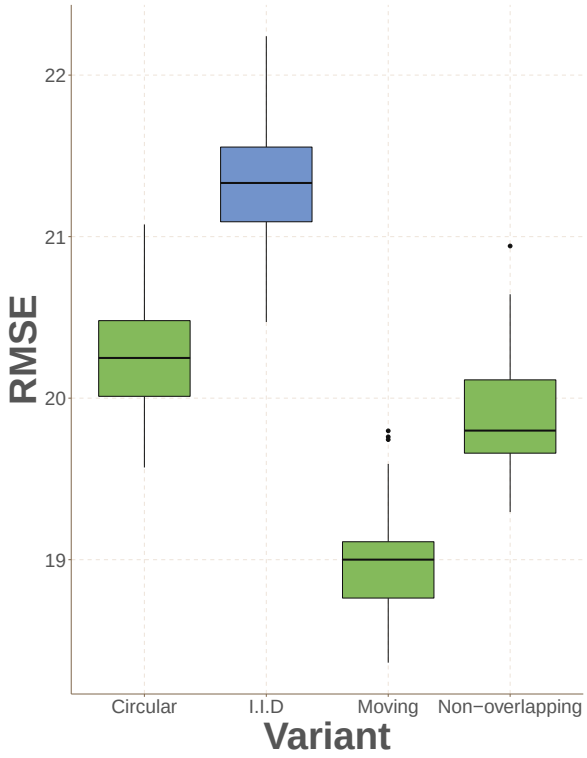

Figure 4.6: Performances of the different variants for $m_{try} = 2$.

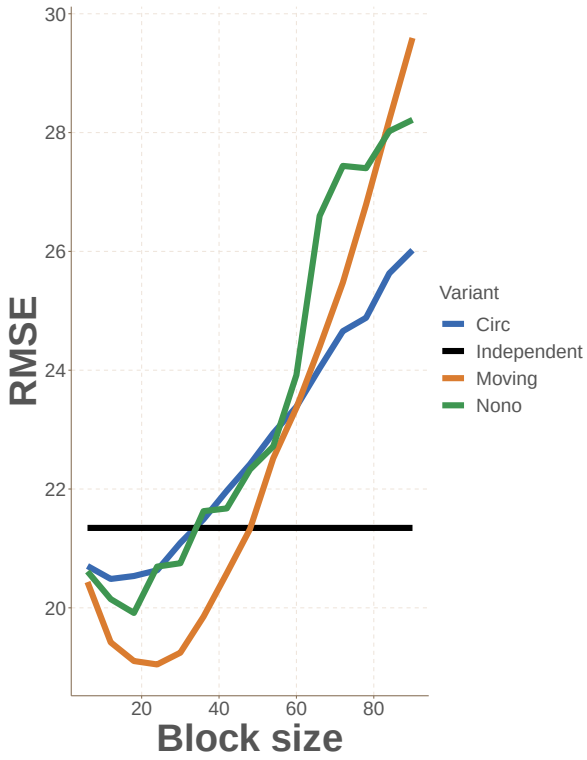

Figure 4.7: Performances of the variants for $m_{try} = 2$ when the block size changes.

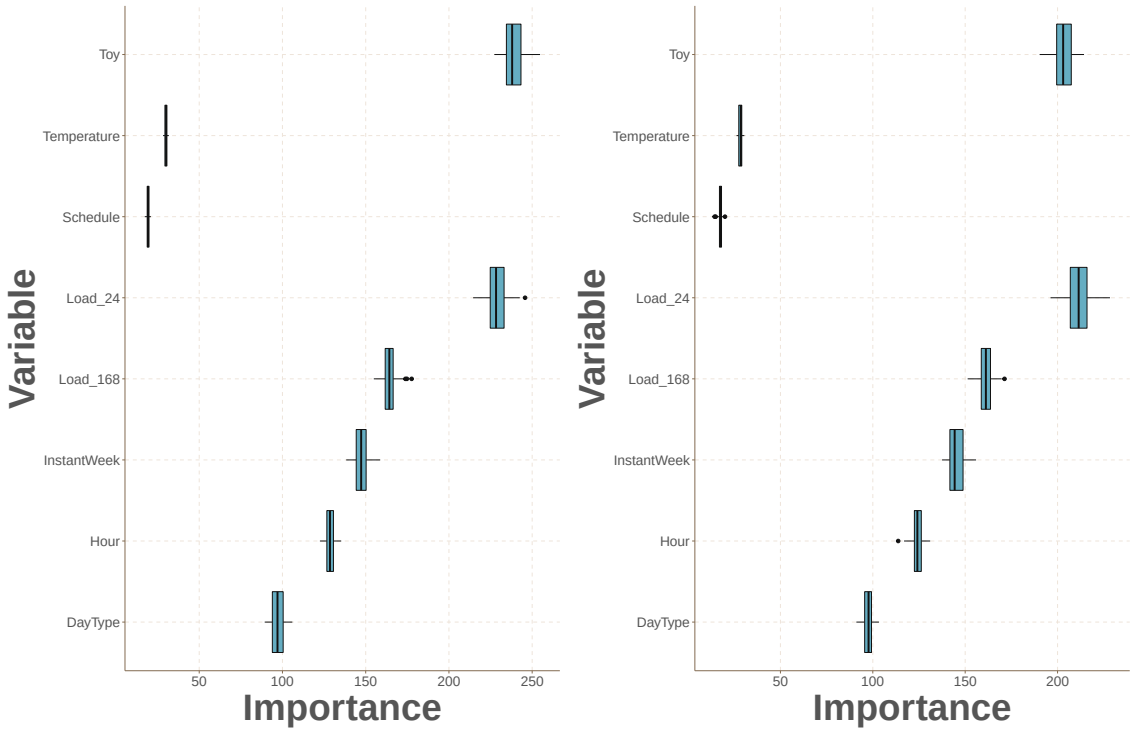

Figure 4.8: Variable importance based on the moving variant: a standard permutation, and b 24h blocks permutation.

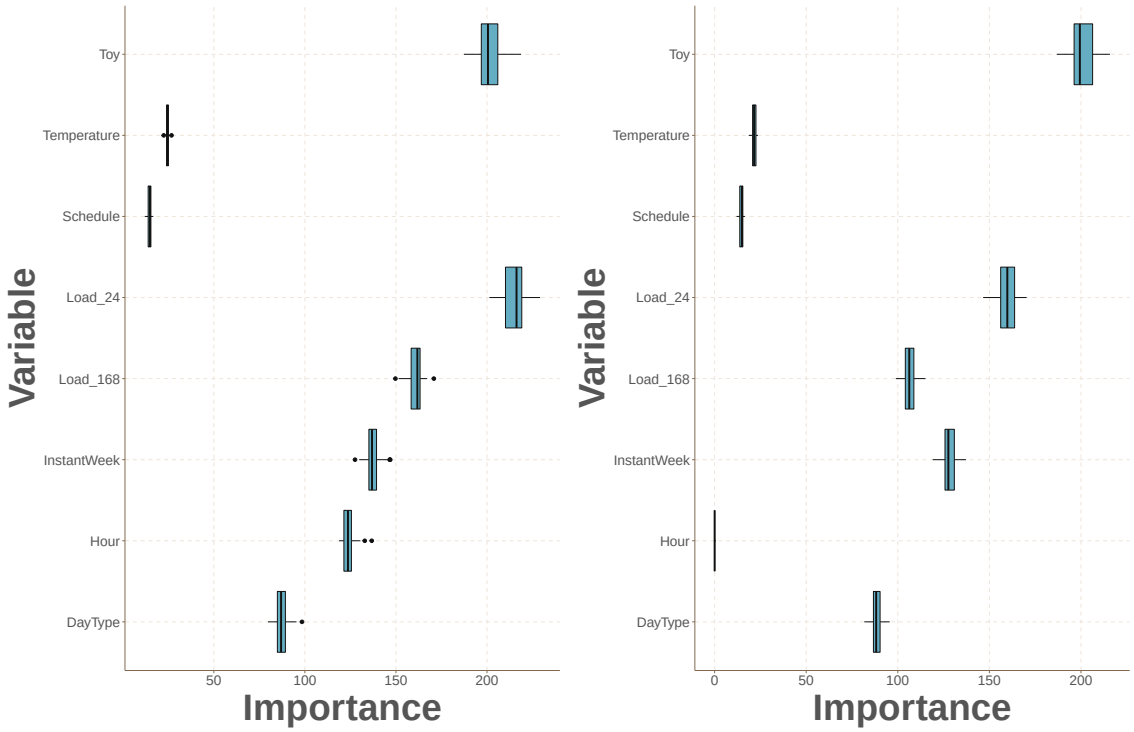

Figure 4.9: Variable importance based on the non-overlapping variant: a standard permutation, and b 24h blocks permutation.

4.4 Bibliography

C. Bergmeir, R. J. Hyndman, and J. M. Benítez. Bagging exponential smoothing methods using stl decomposition and box–cox transformation. *International journal of forecasting*, 32:303–312, 2016. 87

L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, Aug 1996. 86

L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001. 86, 90

L. Breiman, J. Friedman, C. Stone, and R. Olshen. *Classification and Regression Trees*. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984. 86

E. Carlstein. The use of subseries values for estimating the variance of a general statistic from a stationary sequence. *Ann. Statist.*, 14:1171–1179, 1986. 89

G. Cavaliere, D. N. Politis, A. Rahbek, P. Bertail, S. Clémençon, J. Tressou, et al. Recent developments in bootstrap methods for dependent data. *Journal of Time Series Analysis*, 36:462–480, 2015. 87

C. Cordeiro and M. Neves. Forecasting time series with boot.expos procedure. *REVSTAT-Statistical Journal*, 7:135–149, 2009. 87

D. R. Cutler, T. C. Edwards, K. H. Beard, A. Cutler, K. T. Hess, J. Gibson, and J. J. Lawler. Random forests for classification in ecology. *Ecology*, 88:2783–2792, 2007. 86

G. Dudek. Short-term load forecasting using random forests. In *Intelligent Systems'2014*, volume 323, pages 821–828, Cham, 2015. Springer International Publishing. 86, 87

B. Efron. Bootstrap methods: Another look at the jackknife. *Ann. Statist.*, 7:1–26, 1979. 87, 89

M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, 15:3133–3181, 2014. 86

A. Fischer, L. Montuelle, M. Mougeot, and D. Picard. Statistical learning for wind power: a modeling and stability study towards forecasting. *Wind Energy*, 20:2037–2047, 2017. 86, 87

B. Goehry. Random forests for time-dependent processes. preprint: hal-01955331, 2018. 87

M. J. Kane, N. Price, M. Scotch, and P. Rabinowitz. Comparison of arima and random forest time series models for prediction of avian influenza h5n1 outbreaks. *BMC bioinformatics*, 15:276, 2014. 86, 87

H. R. Kunsch. The jackknife and the bootstrap for general stationary observations. *Ann. Statist.*, 17:1217–1241, 1989. 89

A. Lahouar and J. Ben Hadj Slama. Random forests model for one day ahead load forecasting. In *IREC2015 The Sixth International Renewable Energy Congress*, pages 1–6, 2015. 86, 87

R. Y. Liu and K. Singh. Moving blocks jackknife and bootstrap capture weak dependence. In *Exploring the limits of bootstrap (East Lansing, MI, 1990)*, Wiley Ser. Probab. Math. Statist. Probab. Math. Statist., pages 225–248. Wiley, New York, 1992. 89

C. Miller and F. Meggers. The building data genome project: An open, public data set from non-residential building electrical meters. *Energy Procedia*, 122:439 – 444, 2017. 87, 91, 92

J. Moon, Y. Kim, M. Son, and E. Hwang. Hybrid short-term load forecasting scheme using random forest and multilayer perceptron. *Energies*, 11:3283, 2018. 86, 87

D. N. Politis and J. P. Romano. A circular block-resampling procedure for stationary data. In *Exploring the limits of bootstrap (East Lansing, MI, 1990)*, Wiley Ser. Probab. Math. Statist. Probab. Math. Statist., pages 263–270. Wiley, New York, 1992. 89

A. M. Prasad, L. R. Iverson, and A. Liaw. Newer classification and regression tree techniques: bagging and random forests for ecological prediction. *Ecosystems*, 9: 181–199, 2006. 86

J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56:116–124, 2013. 86

V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston. Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43:1947–1958, 2003. 86

M. Wright and A. Ziegler. ranger: A fast implementation of random forests for high dimensional data in c++ and r. *Journal of Statistical Software, Articles*, 77:1–17, 2017. 91

Appendix A

Complément au chapitre 3 : Analyse de l'importance des variables pour la stratégie SSWA

Dans ce travail, nous étudions l'importance des variables suite au travail *Aggregation of Multi-scale Experts for Bottom-up Load Forecasting* (cf. Chapitre 3). Rappelons que l'objectif ici est de prévoir la consommation totale notée Y_t, composée des consommations de n individus $(Y_{t,i})_{\substack{t\in\mathbb{Z}\\1\leq i\leq n}}$. Nous avons également comme informations supplémentaires des facteurs externes $(X_t)_{t\in\mathbb{Z}}$ (ici la température du pays) et un vecteur non-temporelle I_i pour chaque individu (par exemple les informations propres à l'individu comme la tarification ou son statut social). Nous supposons également le modèle suivant, pour un groupe $\mathcal{A}_{j,q}$ donné,

$$Y_t^{\mathcal{A}_{j,q}} = f\left(Y_{t-48}^{\mathcal{A}_{j,q}}, \ldots, Y_{t-7\times 48}^{\mathcal{A}_{j,q}}, X_t, \ldots, X_{t-7\times 48}\right) + \epsilon_t$$

où ϵ_t correspond à l'erreur à l'instant t.

Il existe divers façons de calculer l'importance des variables. Nous utilisons la méthode *Mean Decrease Accuracy* de Breiman [2001] qui part du principe que si une variable n'est pas importante alors permuter ces valeurs ne change pas la précision de la prédiction. Plus précisément, l'importance des variables, pour une forêt aléatoire, est obtenue de la manière suivante : pour chaque arbre de la forêt, nous avons accès aux observations dites *out-of-bag* notées OOB_m, composées des observations non incluses dans l'échantillon bootstrap $\mathcal{D}_n^m$ utilisé pour construire le $m$ème arbre. L'échantillon OOB_m peut alors être utilisé pour estimer l'erreur out-of-bag notée $errOOOB_m$. Pour calculer l'importance de la variable $X^{(\tilde{i})}$, les valeurs de la $\tilde{i}$ème variable sont permutées aléatoirement dans l'échantillon OOB et calculent pour chaque arbre une estimation des erreurs out-of-bag pour les observations permutées. L'importance de la variable $X^{(\tilde{i})}$ est alors obtenue en faisant la moyenne de la différence entre l'erreur out-of-bag avant et après permutation. Plus formellement, si, pour le $m$ème arbre, on note par $\widetilde{errOOOB}_m^{\tilde{i}}$ l'erreur de l'échantillon OOB_m lorsque la $\tilde{i}$ème variable est permutée, alors l'importance de la variable $X^{(\tilde{i})}$ est

définie par

$$Imp\left(X^{(\tilde{i})}\right) = \frac{1}{M}\sum_{m=1}^{M}\left(\widetilde{errOOOB}_m^{\tilde{i}} - errOOOB_m\right).$$

Plus l'augmentation de l'erreur de prédiction après la permutation de la variable $\tilde{i}$ème dans les observations out-of-bag est élevée, plus la variable est importante. Cependant, si la permutation de $X^{(\tilde{i})}$ ne change pas beaucoup la prédiction d'erreur alors l'importance de la variable considérée est faible.

A.1 Importance de référence

Nous prenons pour référence l'importance des variables obtenue en appliquant une forêt aléatoire directement pour la prévision de la consommation totale Y_t, sans étape de clustering. Nous retrouvons en Fig. A.1 l'importance de la forêt précédente. Nous détaillons les importances pour la consommation en Fig. A.2, pour la température en Fig. A.3 et les autres variables en Fig. A.4. Nous remarquons que la variable la plus importante pour la stratégie consistant à prévoir la consommation directement à partir d'une seule forêt aléatoire est la variable correspondante à la consommation totale d'y à 7 jours $Y_{t-7\times48}$, suivi de la consommation du jour précédent Y_{t-24} et l'instant de la journée, représentant la demi-heure qui est à prévoir. Concernant la température, la variable la plus importante est la température du jour d'avant, même si la température à l'instant prévu est disponible. La forêt reconstruit en quelque sorte une température lissée, moyenne de la température brute et de la température de la veille. Cela représente l'inertie des bâtiments à une variation de température. Cependant, sur ce jeu de données la température semble très peu utile. Cela peut se justifier par le fait qu'il s'agit ici d'une température globale, en Irlande où la température est peu variable et du fait que le chauffage électrique est bien moins développé qu'en France par exemple. Enfin, confirmant l'intuition, la variable weekend qui correspond à une variable binaire selon si le moment à prévoir est un jour du weekend ou un jour de la semaine a plus d'importance pour la prévision de la consommation totale que la variable correspondant au jour.

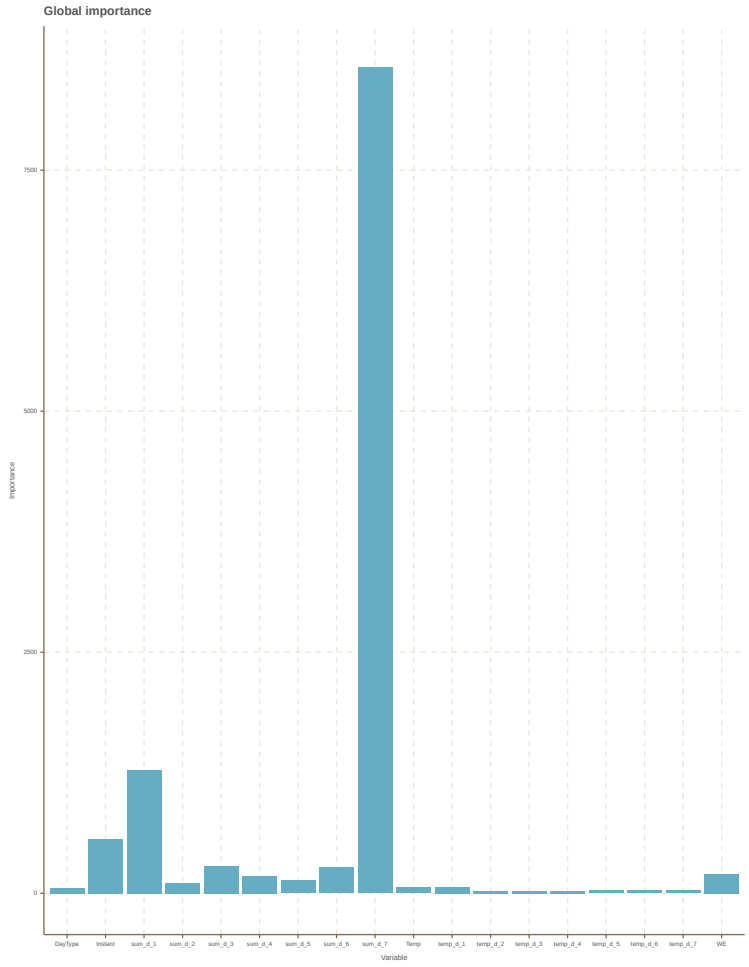

Figure A.1: Importance des variables pour le modèle global.

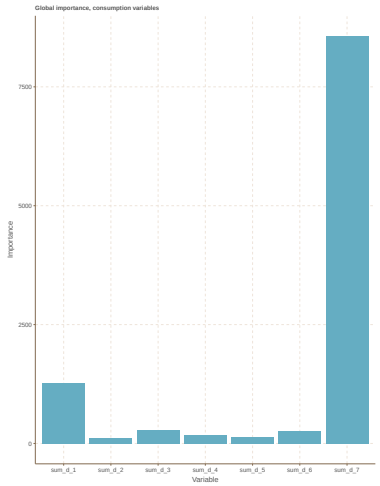

Figure A.2: Importance des lags de consommation pour le modèle global.

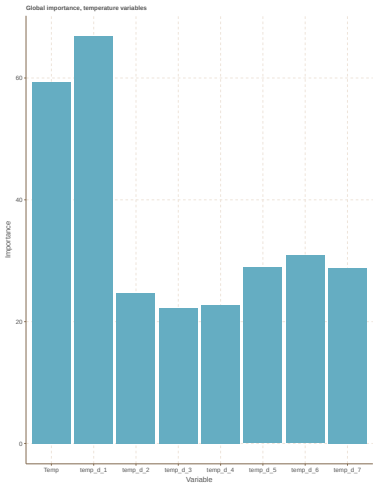

Figure A.3: Importance des lags de température pour le modèle global.

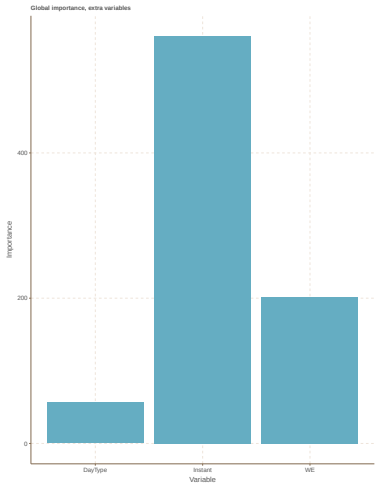

Figure A.4: Importance des autres variables pour le modèle global.

A.2 Importance des variables groupe par groupe

Nous regardons maintenant l'importance de chacun des groupes découlant des trois premières étapes de la stratégie SSWA HAC, c'est-à-dire suite à un clustering hiérarchique sur les individus en utilisant l'information $(I_i)_{1\leq i\leq n}$, la renormalisation des consommations de chaque groupe et enfin la prévision de chacun des groupes par une forêt aléatoire. Pour une partition $\mathcal{A}_j$ fixée, nous observons alors pour chaque groupe $\mathcal{A}_{j,q}$ l'importance des variables obtenue par la forêt aléatoire associée au groupe. Afin de pouvoir comparer au mieux les différentes importances de variables obtenues, nous les normalisons par le maximum de l'importance dans chaque groupe. L'importance normalisée de la $\tilde{i}$ème variable pour le groupe $\mathcal{A}_{j,q}$ s'écrit alors

$$\widetilde{Imp}\left(X^{(\tilde{i}),\mathcal{A}_{j,q}}\right) = \frac{Imp\left(X^{(\tilde{i}),\mathcal{A}_{j,q}}\right)}{\max_{\tilde{l}=1,\dots,p}\left(Imp\left(X^{(\tilde{l}),\mathcal{A}_{j,q}}\right)\right)}$$

où p représente le nombre total de variables que nous avons pour la prévision de la consommation. Nous donnons d'abord en Figs. A.5 to A.7 le nombre d'individus par groupe pour chaque partition. Nous avons ensuite en Figs. A.8 to A.16 les importances des variables pour 5 groupes, 8 groupes et 16 groupes (aller au delà devient trop difficile pour l'analyse). Plus précisément, nous observons en ordonnée l'importance de la variable considérée et en abscisse le numéro du groupe associé à la partition. Nous observons que plus nous augmentons le nombre de groupes, plus les groupes sont sensibles à d'autres variables, en particulier les groupes ayant peu d'individus. Nous observons que pour les petits groupes, la variable Instant est celle qui se place toujours en première place loin devant les autres, contrairement au modèle direct sans clustering. Que cette variable Instant soit importante n'est cependant pas très surprenant étant donné que la consommation dépend sans aucun doute du moment de la journée: un individu ne consommera pas la même quantité d'électricité entre minuit et 7h du matin qu'à 19h lors du repas. Cependant, il est surprenant que les plus les groupes sont petits, plus cette variable est considérée cruciale pour la prévision. Un autre constat concerne la consommation passée d'une semaine $Y_{t-7\times 48}$: plus le nombre d'individus dans les groupes diminuent moins cette variable est importante, pendant que la consommation du jour passé Y_{t-48} semble aussi bien voire meilleure pour la prévision que la précédente. Ceci peut s'expliquer par le fait que les petits groupes possèdent une forte volatilité et une saisonnalité moins forte que les grands groupes qui impliquent que le jour d'avant donne plus d'information que la semaine d'avant. Concernant la température, il semble que les importances pour les différents lags de température sont de plus en plus important en allant vers des partitions plus fines et semble autant important que la consommation pour certains groupes, voire même plus comme pour le groupe 14 dans la partition à 16 groupes. Concernant les variables DayType et Weekend, nous avons vu que pour prévoir la consommation totale à partir du modèle direct la variable Weekend est suffisante pour la prévision mais nous observons qu'à partir de 16 groupes il semble utile de rajouter une information supplémentaire concernant le jour pour certains groupes mais ces deux variables sont clairement très peu importantes dans ce contexte comparées à l'autre variable calendaire qui est l'instant.

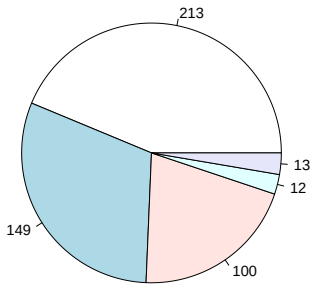

Figure A.5: Nombre d'individus par groupe pour la partition de 5 groupes obtenu par HAC.

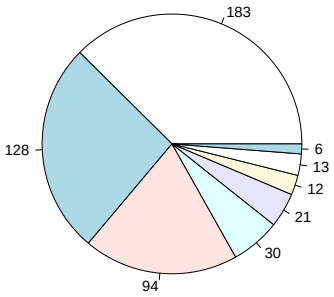

Figure A.6: Nombre d'individus par groupe pour la partition de 8 groupes obtenu par HAC.

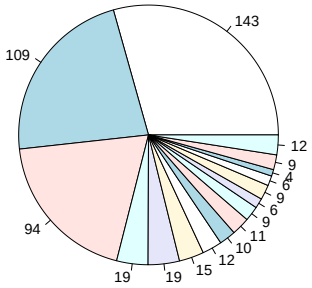

Figure A.7: Nombre d'individus par groupe pour la partition de 16 groupes obtenu par HAC.

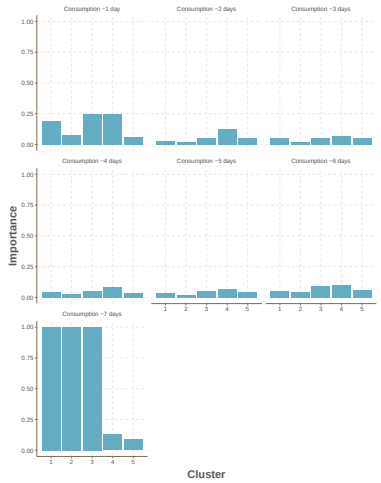

Figure A.8: Importance des lags de consommation pour une partition à 5 groupes.

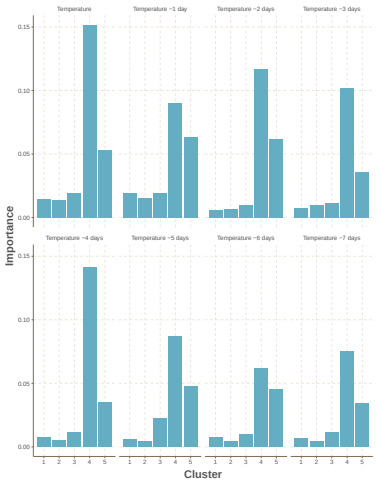

Figure A.9: Importance des lags de température pour une partition à 5 groupes.

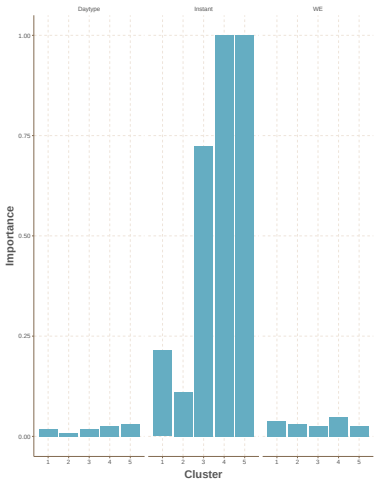

Figure A.10: Importance des autres variables pour une partition à 5 groupes.

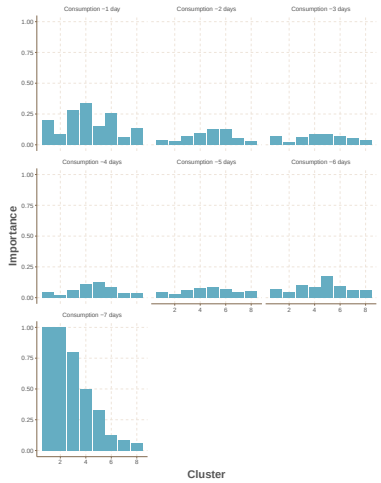

Figure A.11: Importance des lags de consommation pour une partition à 8 groupes.

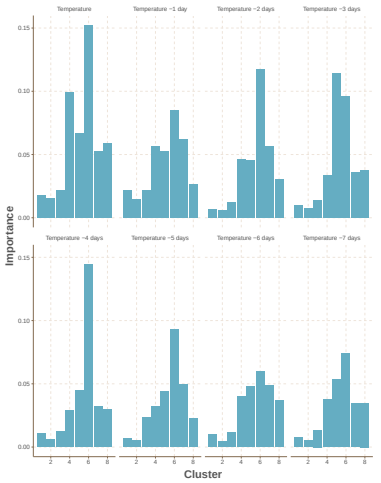

Figure A.12: Importance des lags de température pour une partition à 8 groupes.

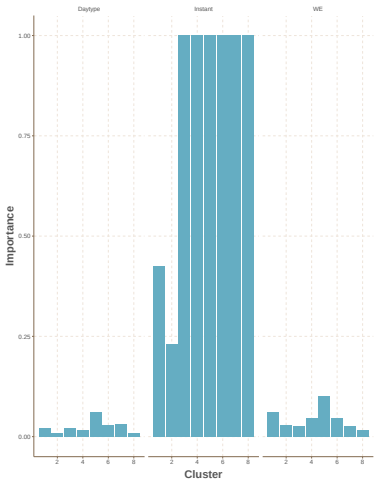

Figure A.13: Importance des autres variables pour une partition à 8 groupes.

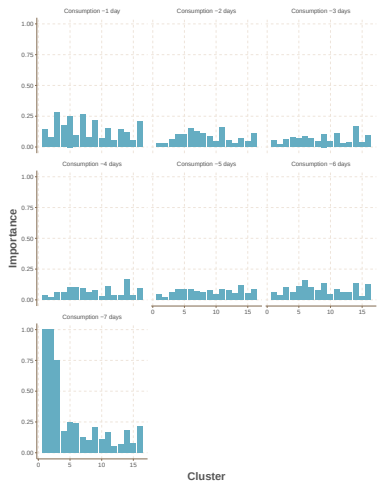

Figure A.14: Importance des lags de consommation pour une partition à 16 groupes.

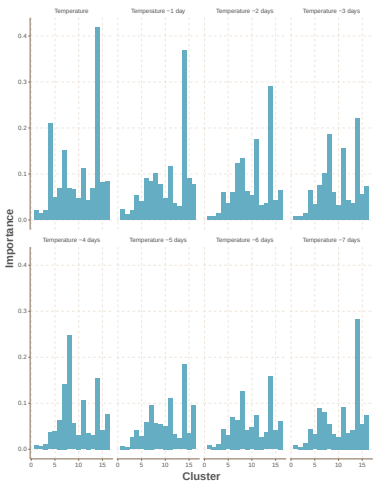

Figure A.15: Importance des lags de température pour une partition à 16 groupes.

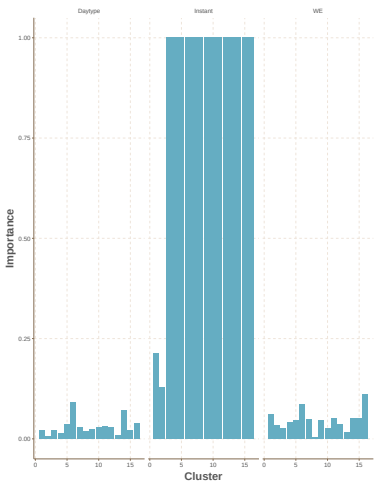

Figure A.16: Importance des autres variables pour une partition à 16 groupes.

A.3 Importance des variables agrégées

Revenons maintenant à l'objectif de la prévision de la consommation totale. Selon la stratégie SSWA, chaque groupe ne contribue pas de la même manière mais uniquement selon un certain poids pouvant dépendre du temps. Pour rappel, la prévision SSWA à un instant t à partir d'une partition $\mathcal{A}_j$ fixée est donnée par

$$\hat{Y}_t^{\mathcal{A}_j} = \sum_{q=1}^{k_j} p_t^{\mathcal{A}_{j,q}} \hat{Z}_t^{\mathcal{A}_{j,q}}.$$

Nous comparons l'importance du modèle directe à l'importance des modèles pour chaque partition en agrégeant les importances de chaque groupe. Les importances n'étant pas du même ordre de grandeur pour chaque groupe, dues à de plus grandes erreurs commises sur certains groupes ne prévoyant qu'un faible nombre de personnes, nous normalisons, comme précédemment, par l'importance maximale dans chaque groupe afin d'éviter d'éventuels phénomènes d'écrasement dus à l'échelle. Nous utilisons deux façons d'agréger l'importance. Une façon est par une agrégation utilisant les poids évoluant dans le temps $p_t^{\mathcal{A}_{j,q}}$ et avons donc une importance des variables évoluant également selon le temps. La deuxième façon d'agréger est un résumé de l'importance en prenant une moyenne sur le temps. Plus précisément pour l'importance temporelle, nous définissons l'importance de la variable $X^{(\tilde{i})}$ par

$$Imp\left(X^{(\tilde{i}),\mathcal{A}_j}\right)_t = \sum_{q=1}^{k_j} p_t^{\mathcal{A}_{j,q}} \widetilde{Imp}\left(X^{(\tilde{i}),\mathcal{A}_{j,q}}\right).$$

La deuxième notion d'importance est défini pour la variable $X^{\tilde{i}}$ de la façon suivante :

$$Imp\left(X^{(\tilde{i}),\mathcal{A}_j}\right) = \sum_{q=1}^{k_j} p^{\mathcal{A}_{j,q}} \widetilde{Imp}\left(X^{(\tilde{i}),\mathcal{A}_{j,q}}\right)$$

où

$$p^{\mathcal{A}_{j,q}} = \frac{1}{T}\sum_{t=1}^{T} p_t^{\mathcal{A}_{j,q}}.$$

Gardons cependant à l'esprit que les importances de variables obtenues pour les groupes d'une partition émanent de modèles construits pour prévoir la consommation de chacun des groupes et non pas la consommation totale. Nous les comparons cependant en utilisant les poids optimisés estimés pour la prévision de la consommation totale.

Nous observons en Figs. A.17 to A.19 l'évolution temporelle de l'importance de variable pour une partition donnée. La première chose à remarquer est que l'importance varie énormément au cours du temps. Pour la stratégie avec 5 groupes, la variable de consommation $Y_{t-7\times48}$ domine la majeure partie du temps suivit de très près par la variable *Instant*. Cependant, pour les stratégies avec 8 et 16 groupes, les variables *Instant* et la consommation $Y_{t-7\times48}$ dominent à tour de rôle l'importance mais l'importance de l'instant augmente, tout comme dans l'analyse

groupe par groupe, en considérant des partitions plus fines. Un autre point intéressant émanent de cette forme d'importance est, bien que l'importance de l'instant a augmenté entre la partition à 5 groupes et la partition à 16 groupes, la forme des courbes des importances est très proche en particulier pour les partitions de 8 et 16 groupes.

L'évolution en moyenne de l'importance des variables lorsque des partitions de plus en plus fines sont considérées se trouve en Figs. A.20 to A.23 avec en ordonnée l'importance des variables et en abscisse le niveau de partition considéré (ici nous allons jusqu'à la partition composée de 92 groupes). Les observations faites pour chaque groupe individuellement avant l'agrégation restent vérifiées après l'agrégation des groupes pour la prévision de la consommation totale. Plus les partitions sont fines, plus les variables comme l'instant ou les températures deviennent importantes contrairement à l'importance de la consommation passé d'il y a 7 jours qui décroît. Cependant l'importance de la consommation du jour d'avant semble rester constante quelque soit la partition considérée. Pour les partitions plus fines, il semble que les autres lags de consommation et températures semblent être plus proche en terme d'importance contrairement au modèle direct où la température ne jouait aucun rôle en comparaison. Concernant les variables DayType et Weekend, nous avons observé qu'en observant les groupes individuellement, certains groupes nécessitaient l'information supplémentaire du jour pour la prévision. Cependant, après agrégation, la variable DayType semble moins intéressante face à la variable Weekend et donc les groupes qui avaient besoin de l'information du jour sont moins considérés dans le mélange SSWA.

En dépit de cette remarque, nous observons finalement en pratique quelque chose de très similaire à ce qu'on obtenait en regardant les importances groupe par groupe. Précisément, plus la partition est fine, moins la variable qui était la plus importante au départ, la consommation passé d'une semaine $Y_{t-7\times48}$, est significative. Cependant, bien que l'importance des lags de températures croit en allant vers les partitions plus fines, nous retrouvons les variables les plus importantes du modèle de base sans clustering excepté la variable *Instant* qui prend la première place.

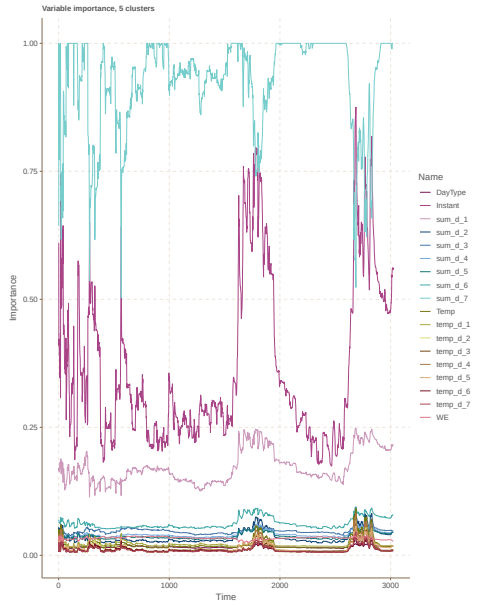

Figure A.17: Importance temporelle des variables pour une partition à 5 groupes.

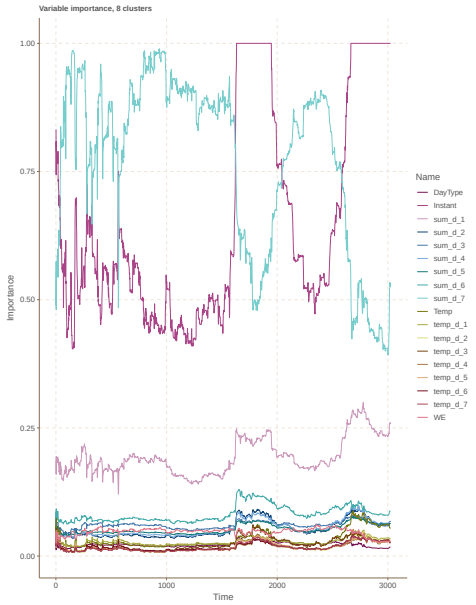

Figure A.18: Importance temporelle des variables pour une partition à 8 groupes.

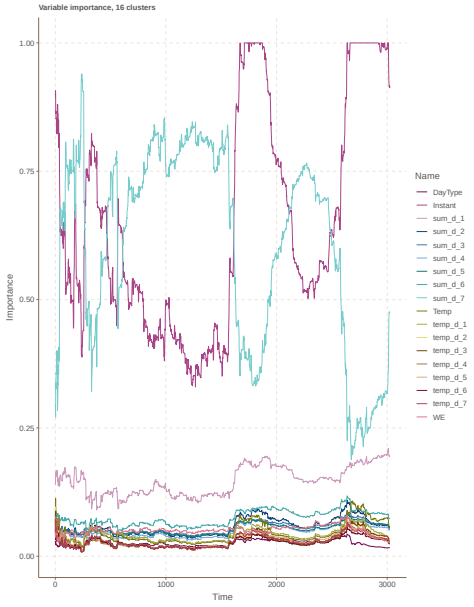

Figure A.19: Importance temporelle des variables pour une partition à 16 groupes.

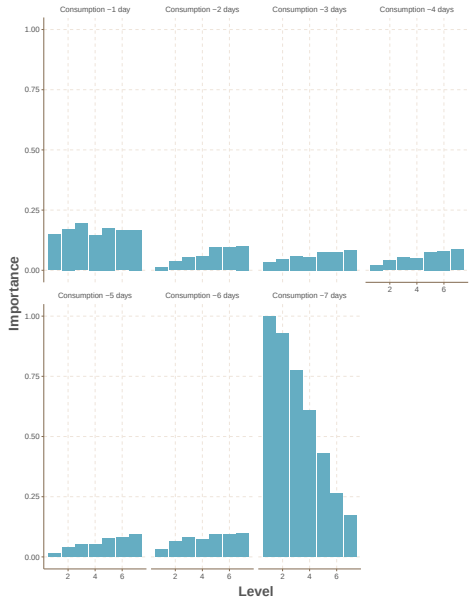

Figure A.20: Importance des lags de consommation selon le niveau de partitionnement.

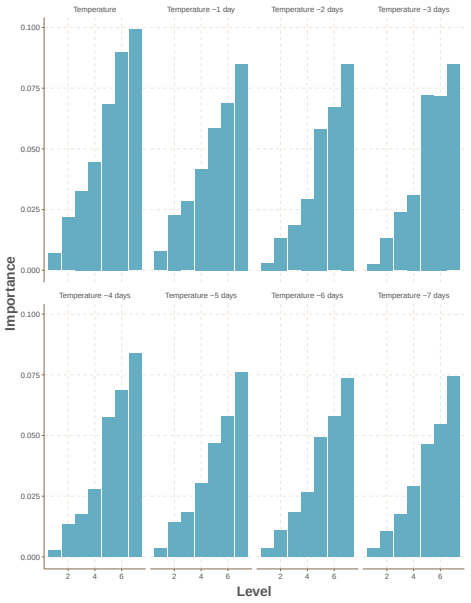

Figure A.21: Importance des lags de température selon le niveau de partitionnement.

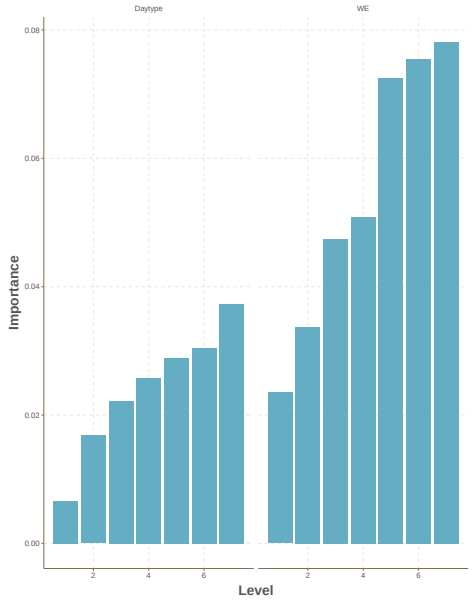

Figure A.22: Importance des variables *Jour* et *Week* selon le niveau de partitionnement.

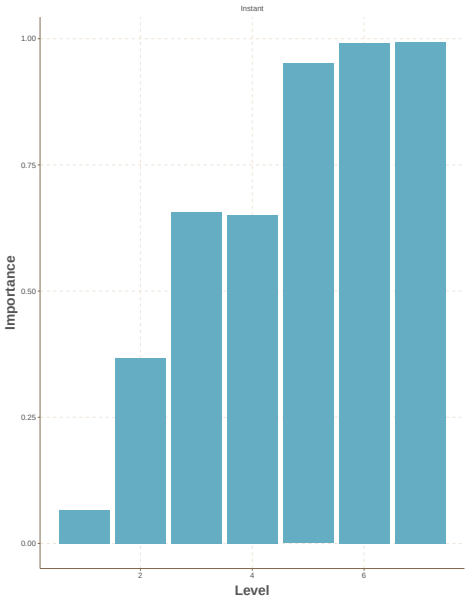

Figure A.23: Importance de la variable *Instant* selon le niveau de partitionnement.

A.4 Bibliographie

L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001. I

Appendix B

Complément au chapitre 3 : Forêts aléatoires pour la génération d'experts en vue de la prévision désagrégée

En collaboration avec Yannig GOUDE, *Pascal* MASSART *et Jean-Michel* POGGI

Sommaire

B.1 Principe

La stratégie de prévision combine plusieurs ingrédients : classification, prévision et mélange. Dans la première partie du contrat, la classification était en quelque sorte séparée des deux autres étapes. Dans cette deuxième partie du contrat nous passons d'un problème en deux étapes (classification des individus puis prévision) à un problème en une seule étape. L'approche exposée ici est un peu différente de celle projetée initialement à cause du fait que la classification sur la base de variables exogènes non temporelles s'est avérée peu compétitive par rapport à un partitionnement au hasard des individus. Autrement dit, dans le schéma deux éléments comptent : le fait de désagréger d'une part et le niveau de désagrégation (combien de groupes et combien de personnes par groupe). C'est sur cette base que les variantes ci-dessous s'appuient.

Nous disposons de n séries temporelles $(Y_{t,i})_{\substack{1\leq t\leq T\\1\leq i\leq n}}$ correspondant aux consommations électriques individuelles de n individus et nous voulons prévoir la consommation totale Y_t à une horizon h :

$$Y_{t+h} = \sum_{i=1}^{n} Y_{t+h,i}.$$

Nous avons également des variables exogènes dépendantes du temps $(X_{t,i})_{\substack{1\leq t\leq T\\1\leq i\leq n}}$ ainsi que des informations relatives aux individus $(I_i)_{1\leq i\leq n}$ qui sont non temporelles.

Contrairement au travail précédent, nous souhaitons classifier et construire des prévisions simultanément pour ensuite mélanger les prévisions pour prévoir la consommation totale. Plus précisément, nous construisons différents modèles de prévision pour différentes échelles et les combinons ensuite avec une stratégie de mélange. Nous avons considéré pour cela deux stratégies que nous allons détailler dans la section suivante.

B.2 Deux stratégies de prévision

Nous supposons pour un groupe d'individus de taille G donné le modèle suivant:

$$Y_t^G = f\left(U_t^G\right) + \epsilon_t \tag{B.2.1}$$

où $U_t^G = \left(Y_{t-1}^G, \ldots, Y_{t-t_Y}^G, X_t^G, \ldots, X_{t-t_X}^G, \widetilde{I}^G\right)$, t_Y le décalage maximum pour la consommation et t_X le décalage maximum pour l'information X. Nous supposons la même forme de modèle que le groupe ait 1 ou 300 individus et estimerons f par une stratégie non-paramétrique.

Nous présentons maintenant deux stratégies avec trois variantes pour chacune.

B.2.1 Première stratégie

Prévisions à échelle fixe

Commençons par fixer G un nombre d'individus par groupe et B un nombre de groupes. Nous construisons B groupes en tirant B fois G individus avec remise parmi les n individus. La consommation pour un groupe b de G individus s'écrit:

$$Y_t^{G,b} = \sum_{i\in b, |b|=G} Y_{t,i}.$$

Étant donné que nous voulons prévoir la consommation totale et que nous voulons que chacune des prévisions locales puisse être considérée comme une prévision du total, nous renormalisons les $Y_t^{G,b}$ de telle sorte que la consommation totale Y_t et la consommation d'un groupe b de G individus $Y_t^{G,b}$ soient du même ordre de grandeur. Pour faire cela, nous cherchons une constante c_b^G telle que

$$\sum_{t\in\text{Train set}} Y_t = c_b^G \sum_{t\in\text{Train set}} Y_t^{G,b}$$

et notons

$$Z_t^{G,b} = c_b^G Y_t^{G,b}$$

la variable de consommation renormalisée.

Nous avons alors sous forme matricielle, pour un nombre d'individus par groupe G et un nombre de groupes B fixés :

$$\begin{pmatrix} Z_1^{G,1} & Z_0^{G,1} & \dots & Z_{1-t_Y}^{G,1} & X_1^{G,1} & \dots & X_{1-t_X}^{G,1} & \widetilde{I}^{G,1} \\ \dots & & & & & & & \\ Z_T^{G,1} & Z_{T-1}^{G,1} & \dots & Z_{T-t_Y}^{G,1} & X_T^{G,1} & \dots & X_{T-t_X}^{G,1} & \widetilde{I}^{G,1} \\ \dots & & & & & & & \\ Z_1^{G,B} & Z_0^{G,B} & \dots & Z_{1-t_Y}^{G,B} & X_1^{G,B} & \dots & X_{1-t_X}^{G,B} & \widetilde{I}^{G,B} \\ \dots & & & & & & & \\ Z_T^{G,B} & Z_{T-1}^{G,B} & \dots & Z_{T-t_Y}^{G,B} & X_T^{G,B} & \dots & X_{T-t_X}^{G,B} & \widetilde{I}^{G,B} \end{pmatrix} \tag{B.2.2}$$

où $\widetilde{I}^{G,b}$ représente une synthèse de l'information du groupe b obtenue par exemple par vote majoritaire des modalités (cas qualitatif) ou la moyenne des variables (cas quantitatif).

Pour un couple (G, B) fixé nous calculons un estimateur pour un sous-ensemble de la population de taille G à partir de la matrice B.2.2 que l'on notera $\hat{Z}_t^{G^B}$. Par exemple en prenant $G = 1$ nous cherchons à prévoir un individu type de la population. Pour $G = n$ nous essayons de prévoir directement la consommation totale. Pour construire ces estimateurs nous utilisons comme dans la deuxième partie du contrat les forêts aléatoires Breiman [2001].

Supposons par exemple que $T = 10000$ et $B = 10$, notre matrice précédente aurait alors 100000 lignes et le calcul de notre estimateur pourrait devenir très long. Nous introduisons un autre paramètre s qui correspond au pourcentage temporel de consommation pris dans le calcul. Plus précisément, au lieu de prendre les T observations pour chaque bloc nous en prenons un ensemble aléatoire $S = s \times T << T$ construit en prenant S points avec remise parmi les T et répétons la procédure pour chaque bloc b. Nous notons $\left(t_i^b\right)_{\substack{1 \le i \le S \\ 1 \le b \le B}}$ les points retenus dans cette procédure et nous obtenons alors la forme suivante pour la matrice de calcul de l'estimateur:

$$\begin{pmatrix} Z_{t_1^1}^{G,1} & Z_{t_1^1-1}^{G,1} & \dots & Z_{t_1^1-t_Y}^{G,1} & X_{t_1^1}^{G,1} & \dots & X_{t_1^1-t_X}^{G,1} & \widetilde{I}^{G,1} \\ \dots & & & & & & & \\ Z_{t_S^1}^{G,1} & Z_{t_S^1-1}^{G,1} & \dots & Z_{t_S^1-t_Y}^{G,1} & X_{t_S^1}^{G,1} & \dots & X_{t_S^1-t_X}^{G,1} & \widetilde{I}^{G,1} \\ \dots & & & & & & & \\ Z_{t_1^B}^{G,B} & Z_{t_1^B-1}^{G,B} & \dots & Z_{t_1^B-t_Y}^{G,B} & X_{t_1^B}^{G,B} & \dots & X_{t_1^B-t_X}^{G,B} & \widetilde{I}^{G,B} \\ \dots & & & & & & & \\ Z_{t_S^B}^{G,B} & Z_{t_S^B-1}^{G,B} & \dots & Z_{t_S^B-t_Y}^{G,B} & X_{t_S^B}^{G,B} & \dots & X_{t_S^B-t_X}^{G,B} & \widetilde{I}^{G,B} \end{pmatrix}. \tag{B.2.3}$$

Mélange de prévisions à différentes échelles

Nous faisons ensuite varier le paramètre d'échelle (le nombre d'individus par bloc) $G \in \mathcal{G}$ et avons un ensemble de prévisions pour différents sous-ensembles de la

population. Fixons (s, B), nous calculons un estimateur de la consommation totale Y_t par le mélange suivant :

$$\hat{Y}_t^B = \sum_{G \in \mathcal{G}} p_t^{G^B} \hat{Z}_t^{G^B} \tag{B.2.4}$$

où $p_t^{G^B}$ est le poids associé à la prévision du sous-ensemble de notre population de taille G à l'instant t pour un nombre de blocs B fixé et tel que pour tout $t, \sum_{G \in \mathcal{G}} p_t^{G^B} = 1$. Nous optimisons ces poids par l'algorithme *Ml-Poly* (Gaillard and Goude [2015, 2016]; Gaillard et al. [2014, 2016]).

Nous allons considérer deux autres façons de mélanger. Fixons s. La deuxième variante sera un mélange en deux étapes. Nous faisons varier le paramètre B dans un certain ensemble $\mathcal{B}$, calculer l'estimateur $\hat{Y}_t^B$ à partir de la première variante (mélange B.2.4) pour chacun de ces $B \in \mathcal{B}$ et mélanger les $|\mathcal{B}|$ estimateurs sortants. Cette variante s'écrit de la façon suivante :

$$\hat{Y}_t = \sum_{B \in \mathcal{B}} p_t^B \hat{Y}_t^B$$

où pour tout $t, \sum_{B \in \mathcal{B}} p_t^B = 1$.

Enfin, la dernière variante, au lieu de calculer l'estimateur de la consommation totale en deux étapes, le mélange est cette fois effectué directement et qu'une seule fois sur tous les estimateurs:

$$\hat{Y}_t = \sum_{B \in \mathcal{B}} \sum_{G \in \mathcal{G}} p_t^{G,B} \hat{Z}_t^{G^B}$$

avec pour tout $t, \sum_{B \in \mathcal{B}} \sum_{G \in \mathcal{G}} p_t^{G,B} = 1$.

Les poids de ces deux dernières variantes sont également optimisés avec l'algorithme *ML-Poly*.

B.2.2 Deuxième stratégie

Prévisions à échelle fixe

Dans cette deuxième stratégie nous reprenons la matrice précédente cependant, cette fois au lieu de créer un seul estimateur pour toute la matrice, nous calculons un estimateur pour chaque bloc b pour (s, B, G) donné. Remarquons que comme l'estimateur est associé à un seul bloc l'estimateur ne dépend plus de $\widetilde{I}^{G,b}$ et calculons maintenant un estimateur pour un sous-ensemble de la population de taille G sur une matrice bien plus petite (qui a seulement S lignes contrairement à la première stratégie où il y en a $S \times B$).

$$\left.\begin{pmatrix} Z_{t_1^b}^{G,b} & Z_{t_1^b-1}^{G,b} & \dots & Z_{t_1^b-t_Y}^{G,b} & X_{t_1^b}^{G,b} & \dots & X_{t_1^b-t_X}^{G,b} \\ \dots & & & & & & \\ Z_{t_S^b}^{G,b} & Z_{t_S^b-1}^{G,b} & \dots & Z_{t_S^b-t_Y}^{G,b} & X_{t_S^b}^{G,b} & \dots & X_{t_S^b-t_X}^{G,b} \end{pmatrix}\right\} \to \hat{f} \tag{B.2.5}$$

Nous notons $\hat{Z}_t^{G,B,b}$ l'estimateur pour cette sous-matrice.

Mélange de prévisions à différentes échelles

Pour (s, B) fixé. La première variante consister à calculer une prévision de la consommation totale Y_t par le mélange suivant :

$$\hat{Y}_t^B = \sum_{G \in \mathcal{G}} \sum_{b \in \{1,\dots,B\}} p_t^{G^B,b} \hat{Z}_t^{G,B,b} \tag{B.2.6}$$

où pour tout t, $\sum_{G\in\mathcal{G}} \sum_{b\in\{1,\dots,B\}} p_t^{G^B,b} = 1$.

Nous considérons également les deux variantes de mélange précédentes. La deuxième variante s'écrit exactement de la même façon soit:

$$\hat{Y}_t = \sum_{B \in \mathcal{B}} p_t^B \hat{Y}_t^B$$

où pour tout t, $\sum_{B\in\mathcal{B}} p_t^B = 1$ et en prenant pour $\hat{Y}_t^B$ l'estimateur précédent (mélange B.2.6).

La dernière variante s'écrit légèrement différent dans cette deuxième stratégie :

$$\hat{Y}_t = \sum_{B \in \mathcal{B}} \sum_{G \in \mathcal{G}} \sum_{b \in \{1,\dots,B\}} p_t^{G,B,b} \hat{Z}_t^{G,B,b}$$

avec pour tout t, $\sum_{B\in\mathcal{B}} \sum_{G\in\mathcal{G}} \sum_{b\in\{1,\dots,B\}} p_t^{G,B,b} = 1$.

Les poids de ces trois variantes sont également optimisés comme dans la première stratégie avec l'algorithme *ML-Poly*.

Nous ferons référence à la $i$ème stratégie et $j$ème variante associée par $S_i V_j$.

B.3 Expériences numériques

Dans notre cas d'étude sur des données individuelles de consommation irlandaises (Commission for Energy Regulation [2011]) nous souhaitons prévoir la consommation totale Y_t à une horizon de $h = 48$ demi-heures. Dans ce jeu de données nous avons pour les données exogènes $X_{t,i}$: la température, l'instant de la journée, le jour de la semaine et une variable week-end ou non. Ces données sont dans notre cas indépendantes de l'individu et donc $X_t^{G,b} = X_t$ quel que soit G et b. Pour les informations individuelles non temporelles $(I_i)_{1\leq i\leq n}$ nous considérons les variables d'un questionnaire donné aux individus : Tarification, CSP, Propriété, Chauffage maison, Chauffage eau, Double vitrage fenêtre, Appareils électroménagers, Année de construction ainsi que des indicateurs sur les courbes de consommation.

Nous découpons le jeu de données en un jeu d'entraînement correspondant aux premiers 50% des données et un jeu de test avec le reste des données pour estimer les erreurs des stratégies en utilisant le RMSE définit par :

$$\text{RMSE}\left(Y, \hat{Y}\right) = \sqrt{\frac{1}{|\text{Test set}|} \sum_{t \in \text{Test set}} \left(Y_t - \hat{Y}_t\right)^2}.$$

Nous considérons les valeurs des paramètres suivantes pour les deux stratégies :

$$\begin{aligned} \mathcal{G} &= \{5, 10, 25, 40, 50, 75, 100, 150, 200, 250, 300, 350, 400, n\}, \\ B &= \{1, \ldots, 10\}, \\ s &= \{0.1, \ldots, 0.9\}, \\ t_Y = t_X &= 7 \times 48 \text{ demi-heures.} \end{aligned}$$

Nous distinguons à présent la construction des estimateurs pour les deux stratégies.

Forêts aléatoires pour la première stratégie.

Nous construisons 4 types de forêts aléatoires. Pour (s, B, G) fixé nous avons :

- une première forêt aléatoire qui consistera à prendre G individus (sans faire B répétitions) et entraîner sur le jeu d'entraînement total et sans les informations non-temporelles (questionnaire ou indicateurs de courbes) des individus i.e sans I.
- La deuxième forêt sera construite à partir de la matrice B.2.3 définie lors de l'explication de la première stratégie sans informations relatives aux individus.
- La troisième forêt aléatoire sera construite également à partir de la matrice B.2.3 en prenant pour $\widetilde{I}$ les informations du questionnaire des individus.
- De la même manière pour le dernier type de forêts aléatoires en prenant pour $\widetilde{I}$ les informations sur la consommation des individus.

Nous espérons à partir des deux derniers type de forêts aléatoires d'avoir une sorte de classification directement dans la régression en utilisant les informations sur les consommateurs.

Forêts aléatoires pour la deuxième stratégie

Nous avons pour cette stratégie seulement deux types de forêts aléatoires. La première forêt est construite de la même manière que la première forêt de la stratégie précédente. La deuxième type de forêt est construite comme indiqué en section B.2.2 à partir d'une matrice de la forme donnée en B.2.5.

Contrairement à la stratégie précédente où il n'y a que 4 forêts aléatoires pour un triplet (s, B, G), ici nous en avons $1 + B$.

Nous utilisons pour référence une forêt aléatoire construit directement sur la consommation globale selon le modèle B.2.1 qui donne sur ce jeu de données un RMSE de 27.6. Toutes les forêts aléatoires sont construites à l'aide du package *ranger* (Wright and Ziegler [2017]) avec les paramètres par défaut.

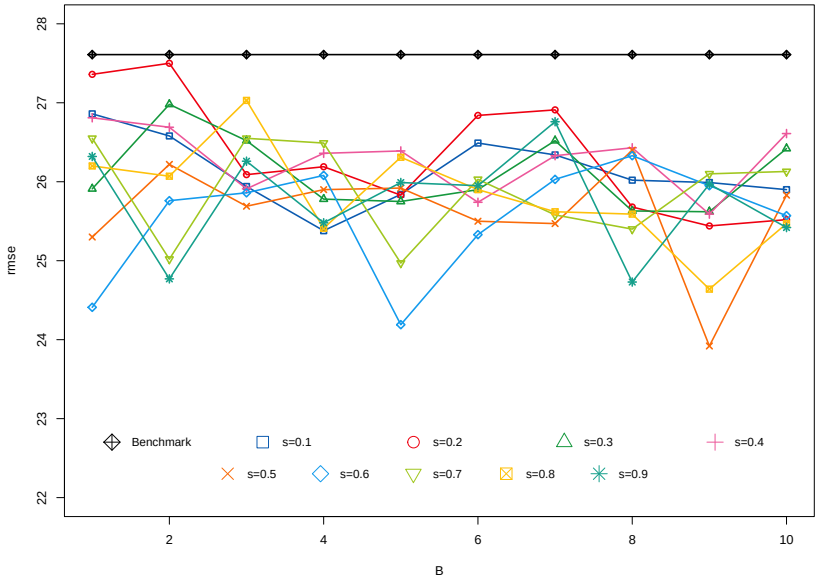

Figure B.1: Comparaison des RMSE pour la première stratégie pour différent choix de s en fonction de B.

B.3.1 Résultats pour la première stratégie

Nous trouvons les résultants pour la variante S_1V_1 en Fig. B.1. Nous remarquons directement que cette variante fait mieux que la référence quel que soit le choix de s et de B. Cependant il est difficile de dire quel choix est optimal. En effet pour un s fixé les courbes sont en dent de scie et il semble ne pas avoir de lien entre s et B pour le meilleur résultat possible (voir Table B.1 pour les choix optimaux de B pour un s donné).

Table B.1: Meilleur choix de B pour un s donné (Première stratégie).

s	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
B	4	9	9	9	9	5	5	9	8
RMSE	25.38	25.44	25.62	25.59	23.92	24.19	24.97	24.64	24.73

Nous prenons maintenant les meilleurs couples (S, B) pour la première variante. Les trois variantes sont meilleures que la référence (Fig. B.2) mais il est flagrant que la première variante est la moins bonne des trois variantes (excepté les points $s = \{0.5, 0.9\}$ où elle fait tout aussi bien que la deuxième variante) de plus le paramètre B est éliminé d'une certaine manière par le mélange pour les variantes S_1V_2 et S_1V_3.

Pour la deuxième variante S_2V_2 qui consiste à mélanger les estimateurs obtenus précédemment sur tout $\mathcal{B}$ (mélange en deux étapes) a de meilleures performances que les deux autres variantes pour $s = \{0.2, \ldots, 0.7\}$ et est optimal pour $s = 0.5$ qui sera également la meilleure performance que l'on peut obtenir pour cette stratégie.

Enfin pour la dernière variante S_2V_3 qui consiste à mélanger tous les estimateurs en une seule étape, a de meilleures performances que les trois autres pour $s = \{0.1, 0.8, 0.9\}$ et est également optimal pour $s = 0.5$ mais reste plus difficile à analyser par la nature du mélange.

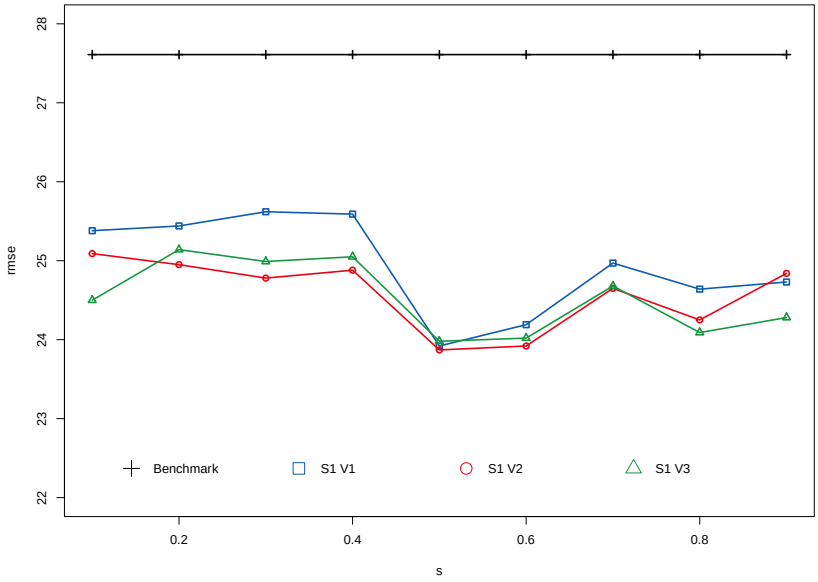

Figure B.2: Comparaison des RMSE entre S_1V_1, S_1V_2 et S_1V_3 pour différent choix de s.

B.3.2 Deuxième stratégie

Nous observons d'abord, comme précédemment, que pour la première variante S_2V_1 (Fig. B.3) fait mieux que la référence quel que soit (s, B). Cependant nous retrouvons aussi comme dans la première stratégie des courbes en dent de scie pour s fixé et il ne semble à nouveau ne pas avoir de lien entre s et B. Nous donnons en Table B.2 le choix optimal de B pour un s donné.

Table B.2: Meilleur choix de B pour un s donné (Deuxième stratégie).

s	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
B	8	10	4	5	8	5	4	6	10
RMSE	23.51	24.54	23.82	24.05	23.96	24.28	23.70	23.93	24.44

Nous prenons pour la première variante le B optimal pour chaque s (Fig. B.2) et obtenons une performance meilleure que la deuxième S_2V_2 et troisième variantes S_3V_3 uniquement pour $s = 0.8$.

La deuxième variante S_2V_2 semble mieux prévoir que les deux autres variantes pour $s = \{0.5, 0.6, 0.7\}$ et a l'avantage de ne pas avoir à faire de choix au préalable sur le paramètre B contrairement à la première variante. La dernière variante S_2V_3 est celle qui prévoit le mieux pour $s = \{0.1, 0.2, 0.3, 0.4, 0.9\}$. Cette dernière a également l'avantage, comme la variante précédente S_2V_2, d'être très efficace quel que soit le paramètre de la stratégie mais reste difficilement analysable comme tous les estimateurs à un jeu de paramètres $(s, \mathcal{B}, \mathcal{G})$ fixe sont agrégés.

Le point final est que ces trois stratégies semblent mieux fonctionner quand s est petit (excepté le point $s = 0.2$), en particulier optimal pour $s = 0.1$ et avec la meilleure RMSE pour la variante S_2V_3, c'est à dire qu'entraîner les estimateurs sur tout l'échantillon ne semble pas utile ce qui réduit également considérablement le temps de calcul.

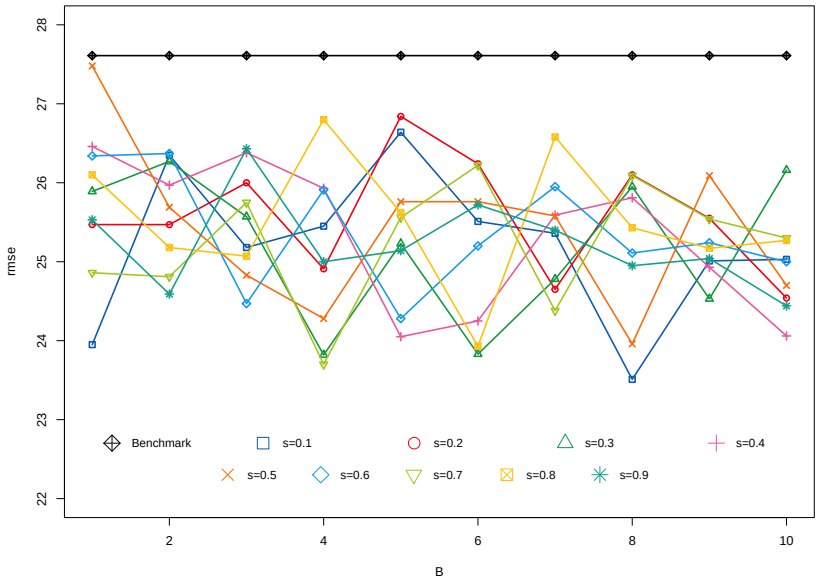

Figure B.3: Comparaison des RMSE pour la deuxième stratégie S_2V_1 pour différent choix de s en fonction de B.

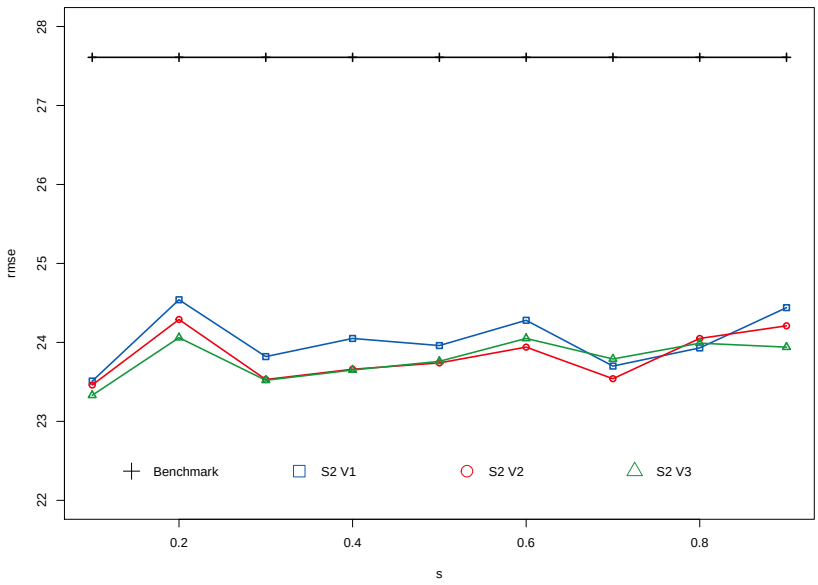

Figure B.4: Comparaison du RMSE des variantes S_2V_1, S_2V_2 et S_2V_3 en fonction de s.

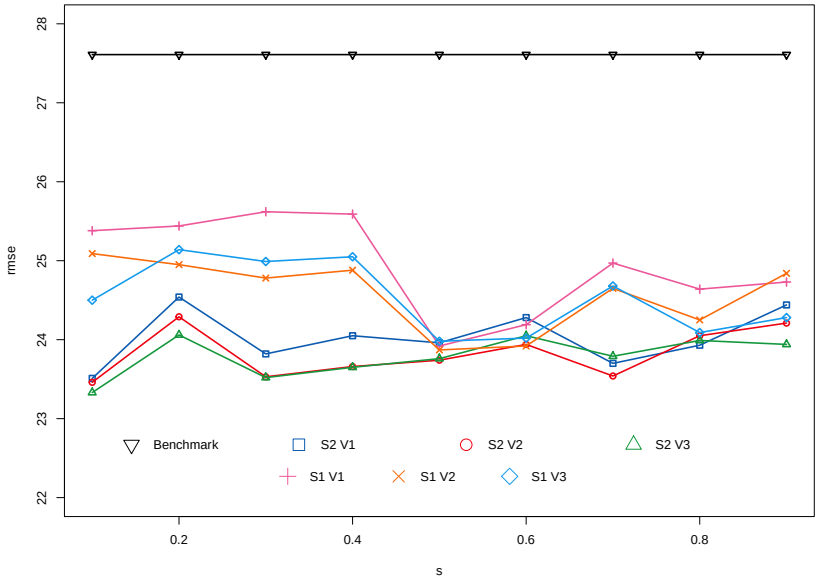

Figure B.5: Comparaison entre les deux stratégies.

B.3.3 Comparaison

Nous remarquons une majeure domination en terme de performance de prévision de la deuxième stratégie excepté aux points $s = \{0.5, 0.6\}$ où les deux variantes S_1V_2 et S_1V_3 font aussi bien que les variantes S_2V_2 et S_2V_3 (Fig. B.5).

Nous souhaitons également comparer ces deux stratégies aux travaux précédents et voir si l'on obtient de meilleures performances. Nous remettons le graphique des résultats des travaux précédent en Fig. B.6.

La première stratégie atteint au mieux un RMSE de 23.87 avec la variante S_1V_2 ce qui correspond à l'erreur qu'on obtient autour de 40 clusters dans les travaux précédent. Pour la deuxième stratégie nous obtenons au mieux un RMSE de 23.33. Dans les deux cas nous n'arrivons pas à retrouver les performances précédentes.

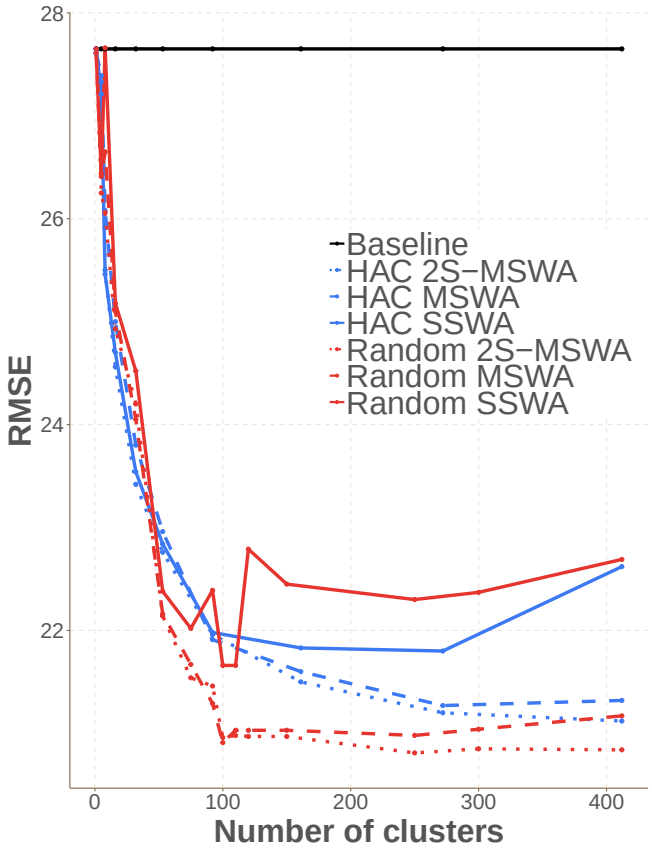

Figure B.6: Comparaison des erreurs entre les stratégies des travaux précédents en fonction du nombre de clusters, HAC *vs* désagrégation aléatoire.

B.4 Conclusion

Nous n'arrivons pas à retrouver les performances des travaux précédentes et notons qu'il est toujours difficile d'intégrer la nouvelle information liée à la consommation ou les habitudes des individus et ce quel que soit le nombre d'individus dans les agrégats. Cependant cette expérience confirme que la désagrégation couplée avec une aggrégation "experte" sont fondamentales dans l'amélioration de la prévision.

La deuxième stratégie montre cependant qu'en rajoutant de la diversité dans les estimateurs en les calculant uniquement sur des sous-matrices de petite taille (rappelons que les meilleures performances sont obtenues pour $s = 0.1$) il est possible d'améliorer nettement la prévision de la consommation totale en créant des agrégats d'individus de différentes tailles et sans avoir d'informations non-temporelle spécifiques aux individus.

B.5 Bibliographie

L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001. XV

Commission for Energy Regulation. Electricity smart metering customer behaviour trials (cbt) findings report. http://www.cer.ie/docs/000340/cer11080(a)(i).pdf, 2011. XVII

P. Gaillard and Y. Goude. Forecasting electricity consumption by aggregating experts; how to design a good set of experts. In *Modeling and stochastic learning for forecasting in high dimensions*, volume 217, pages 95–115. Springer, 2015. XVI

P. Gaillard and Y. Goude. Opera: Online prediction by expert aggregation. r package, 2016. XVI

P. Gaillard, G. Stoltz, and T. van Erven. A second-order bound with excess losses. In *Proceedings of The 27th Conference on Learning Theory*, volume 35 of *Proceedings of Machine Learning Research*, pages 176–196. PMLR, 2014. XVI

P. Gaillard, Y. Goude, and R. Nedellec. Additive models and robust aggregation for gefcom2014 probabilistic electric load and electricity price forecasting. *International Journal of Forecasting*, 32:1038 – 1050, 2016. XVI

M. Wright and A. Ziegler. ranger: A fast implementation of random forests for high dimensional data in c++ and r. *Journal of Statistical Software, Articles*, 77:1–17, 2017. XVIII

Appendix C

Complement to chapter 4: rangerts

C.1 The original package

ranger (https://github.com/imbs-hl/ranger) is an open source R package on github, created and maintained by Marvin N. Wright, with a clear explanation in the article Wright and Ziegler [2017].

C.2 What does rangerts do

The main idea of this modified version is to test the random forest algorithm using the block bootstrapping, in order to take time dependence structure of the data into account, during the tree construction phase, instead of the standard bootstrap (Efron [1979]).

In order to benefit from the efficient implementation of the ranger package, we based our implemetation on their C++ codes and we added five different kinds of block bootstrapping: circular blocks (Politis and Romano [1992]), moving blocks (Kunsch [1989]; Liu and Singh [1992]), non-overlapping blocks (Carlstein [1986]), stationary blocks (Politis and Romano [1994]) and seasonal blocks (Dudek et al. [2014]).

C.3 New parameters

All functions are the same as in the ranger package as well as the main function name. We added four parameters in the ranger function: *bootstrap.ts*, *block.size*, *period* and *by.end*. All these parameters are to be used with caution together with the parameters already included in the ranger original function.

- *bootstrap.ts*: string parameter, empty (= NULL) for standard bootstrap, or takes its value in nonoverlapping, moving, stationary, circular, and seasonal. By default = NULL. Some research works have demonstrated that moving might be more beneficial.

- *block.size*: the number of observations per block, by default = 10. In the stationary block bootstrapping mode, based on the geometric law with parameter $p = \frac{1}{block.size}$.

- *period*: the number of observations per period, if seasonal bootstrap is selected.
- *by.end*: boolean, by default = TRUE, build block from the end to the start of time series.

C.4 Installation

To install the development version from GitHub using devtools, run

```
# quiet = TRUE to mask c++ compilation messages, optional
devtools::install_github("BenjaminGoehry/BlocRF/rangerts",
                         quiet = T)
# to get the a default guide for the rangerts package, use
browseVignettes("rangerts")
```

C.5 Example

This example is based on the load forecasting application done in chapter 4 based on the dataset from Miller and Meggers [2017]. The goal is to forecast the load at a horizon of 24 hours of a building denoted by Load_t based on the following features:

- Load_{24} (resp. Load_{168}) corresponds to the 24 hour lagged load (resp. 168 hour lagged load);
- Temp_t corresponds to the temperature at instant t;
- Schedule_t take three values: Regular, Break, Holiday;
- Hour_t corresponds to the hour of the day at instant t;
- InstantWeek_t corresponds to the hour in the month;
- DayType_t corresponds to the day of the week;
- Toy_t corresponds to the day of the year divided by 366.

```
library(rangerts)
# to check the function ranger function helper
?rangerts::ranger

# load consumption dataset
dataset_elec <- readRDS("dataset_elec.RDS")

# split train and test
d_train <- dataset_elec %>%
  dplyr::filter(Test == 0) %>%
  dplyr::select(- Test)

d_test <- dataset_elec %>%
  dplyr::filter(Test == 1) %>%
  dplyr::select(- Test)

# set formula
```

```
eq_Load ← as.formula("Load_s ~ Load_24 + Load_168 + Temp_s +
                        Schedule_s + Hour_s + InstantWeek_s +
                        DayType_s + Toy_s"
                        )

# set general parameters
nb_trees ← 1000
Mtry ← 3 #floor(sqrt(ncol(df_train)))
block_size ← 24

# Use case 1
# the default ranger with the standard bootstrap
forest_iid ← rangerts::ranger(eq_Load,
                              data = d_train,
                              num.trees = nb_trees,
                              mtry = Mtry,
                              seed = 1, # for reproductibility
                              )

# Use case 2
# the nonoverlapping block bootstrap variant
forest_noov ← rangerts::ranger(eq_Load,
                              data = d_train,
                              num.trees = nb_trees,
                              mtry = Mtry,
                              seed = 1,
                              bootstrap.ts = "nonoverlapping",
                              block.size = block_size
                              )

# Use case 3
# the moving block bootstrap mode
forest_mv ← rangerts::ranger(eq_Load,
                              data = d_train,
                              num.trees = nb_trees,
                              mtry = Mtry,
                              seed = 1,
                              bootstrap.ts = "moving",
                              block.size = block_size
                              )

# Use case 4
# the circular block bootstrap mode
forest_cr ← rangerts::ranger(eq_Load,
                              data = d_train,
                              num.trees = nb_trees,
                              mtry = Mtry,
                              seed = 1,
                              bootstrap.ts = "circular",
                              block.size = block_size
                              )

# final model list
forests_List ← list(forest_iid,
                    forest_no,
                    forest_mv,
```

```
                    forest_cr)

# compare rmse and mape
algo_spec <- c("standard",
               "nonoverlapping",
               "moving",
               "circular")

rmse <- purrr::map_dbl(forests_List,
               ~ yardstick::rmse_vec(d_test$Load,
                                     predict(.x, df_test)$predictions))
cbind(algo_spec, round(rmse, 2))

mape <- purrr::map_dbl(forests_List,
               ~ yardstick::mape_vec(d_test$Load,
                                     predict(.x, df_test)$predictions))
cbind(algo_spec, round(mape, 2))
```

C.6 Bibliography

E. Carlstein. The use of subseries values for estimating the variance of a general statistic from a stationary sequence. *Ann. Statist.*, 14:1171–1179, 1986. XXV

A. E. Dudek, J. Leśkow, E. Paparoditis, and D. N. Politis. A generalized block bootstrap for seasonal time series. *Journal of Time Series Analysis*, 35:89–114, 2014. XXV

B. Efron. Bootstrap methods: Another look at the jackknife. *Ann. Statist.*, 7:1–26, 1979. XXV

H. R. Kunsch. The jackknife and the bootstrap for general stationary observations. *Ann. Statist.*, 17:1217–1241, 1989. XXV

R. Y. Liu and K. Singh. Moving blocks jackknife and bootstrap capture weak dependence. In *Exploring the limits of bootstrap (East Lansing, MI, 1990)*, Wiley Ser. Probab. Math. Statist. Probab. Math. Statist., pages 225–248. Wiley, New York, 1992. XXV

C. Miller and F. Meggers. The building data genome project: An open, public data set from non-residential building electrical meters. *Energy Procedia*, 122:439 – 444, 2017. XXVI

D. N. Politis and J. P. Romano. A circular block-resampling procedure for stationary data. In *Exploring the limits of bootstrap (East Lansing, MI, 1990)*, Wiley Ser. Probab. Math. Statist. Probab. Math. Statist., pages 263–270. Wiley, New York, 1992. XXV

D. N. Politis and J. P. Romano. The stationary bootstrap. *Journal of the American Statistical Association*, 89:1303–1313, 1994. XXV

M. Wright and A. Ziegler. ranger: A fast implementation of random forests for high dimensional data in c++ and r. *Journal of Statistical Software, Articles*, 77:1–17, 2017. XXV

université PARIS-SACLAY | École doctorale de mathématiques Hadamard (EDMH)

Titre : Prévision multi-échelle par agrégation de forêts aléatoires. Application à la consommation électrique.

Mots Clefs : statistiques, prévision, forêts aléatoires

Résumé : Cette thèse comporte deux objectifs. Un premier objectif concerne la prévision d'une charge totale dans le contexte des Smart Grids par des approches qui reposent sur la méthode de prévision ascendante. Le deuxième objectif repose quant à lui sur l'étude des forêts aléatoires dans le cadre d'observations dépendantes, plus précisément des séries temporelles. Nous étendons dans ce cadre les résultats de consistance des forêts aléatoires originelles de Breiman ainsi que des vitesses de convergence pour une forêt aléatoire simplifiée qui ont été tout deux jusqu'ici uniquement établis pour des observations indépendantes et identiquement distribuées. La dernière contribution sur les forêts aléatoires décrit une nouvelle méthodologie qui permet d'incorporer la structure dépendante des données dans la construction des forêts et permettre ainsi un gain en performance dans le cas des séries temporelles, avec une application à la prévision de la consommation d'un bâtiment.

Title : Multi-scale forecasting by aggregation of random forests. Application to load forecasting.

Keys words : statistics, forecasting, random forests

Abstract : This thesis has two objectives. A first objective concerns the forecast of a total load in the context of Smart Grids using approaches that are based on the bottom-up forecasting method. The second objective is based on the study of random forests when observations are dependent, more precisely on time series. In this context, we are extending the consistency results of Breiman's random forests as well as the convergence rates for a simplified random forest that have both been hitherto only established for independent and identically distributed observations. The last contribution on random forests describes a new methodology that incorporates the time-dependent structure in the construction of forests and thus have a gain in performance in the case of time series, illustrated with an application of load forecasting of a building.