Contribution à la modélisation et l'inférence de réseaux de régulation de gènes

Magali Champion

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *05/12/2014* **par :**

Magali CHAMPION

Contribution à la modélisation et l'inférence de réseau de régulation de gènes

---

JURY

Etienne BIRMELE	Professeur, Université Paris Descartes	Rapporteur
Christine CIERCO-AYROLLES	Chargée de recherche, INRA Toulouse	Co-Directrice
Sébastien GADAT	Professeur, Université Toulouse 1 Capitole	Directeur
Fabrice GAMBOA	Professeur, Université Paul Sabatier	Examinateur
Catherine MATIAS	Directrice de recherche CNRS	Rapporteur
Nicolas VAYATIS	Professeur, ENS Cachan	Président du Jury
Matthieu Vignes	Senior Lecturer, Massey University	Invité

École doctorale et spécialité :
MITT : Domaine Mathématiques : Mathématiques appliquées
Unité de Recherche :
Institut de Mathématiques de Toulouse
Directeur(s) de Thèse :
Sébastien GADAT, *Christine CIERCO-AYROLLES* et *Matthieu VIGNES*
Rapporteurs :
Catherine MATIAS et *Etienne BIRMELE*

Comme un vent irrésistible
la vie suit ses courbes invisibles et file
vers l'avant
Jilano

Remerciements

*H*ourra, j'ai réussi à arriver au terme de ces trois années de dur labeur ! Il ne me reste plus qu'à franchir l'ultime étape de ce parcours du combattant, moment que tout bon doctorant attend, les remerciements !

*A*lors pour commencer, je tiens à remercier les personnes sans qui cette thèse n'aurait certainement pas eu lieu, à savoir mes directeurs de thèse : Sébastien, parce qu'il a su me guider, non sans humour !, pendant ses (un peu plus que) trois années, le nouveau néo-zélandais Matthieu pour ses discussions passionées et ses corrections au feutre rose, version patte de mouche et Christine pour avoir su mettre un peu d'ordre dans ce monde masculin. Merci à tous les trois, travailler avec vous a été un vrai plaisir !

*R*apporter un manuscrit de thèse n'est pas une tâche facile. Je remercie donc également Catherine Matias et Etienne Birmelé d'avoir accepté de s'y atteler et d'avoir insisté sur certains points sensibles de ce manuscrit. Merci aussi à Nicolas Vayatis de présider ce jury et Fabrice Gamboa d'avoir accepté de représenter l'UPS.

*R*ares sont ceux qui ont la chance de pouvoir travailler dans deux laboratoires. Mon accueil au sein de l'unité MIA de l'INRA de Toulouse a été particulièrement agréable. Les responsables administratifs Fabienne, Nathalie et Alain font un travail formidable (quelle réactivité !). Un grand merci à Maman Hiep, qui m'a accueillie dès les premiers mois de ma thèse dans le bureau MIA20 : même si je n'aime pas forcément les fourmis et Céline Dion, tes micros (ou macros suivant les jours) siestes et tes éclats de voix "La Vie !" nous ont vraiment bien fait rire, et quels nems ! Merci aussi à Julia, pour qui la thèse n'est pas un long fleuve tranquille mais qui a su garder le sourire en toutes circonstances, et Charlotte, la bleue. Tu as débarqué comme un boulet de canon dans notre bureau tranquille et bouleversé nos vieilles habitudes de mamie mais je crois qu'on est toutes d'accord, on ne regrette rien ! Fais attention quand même à ne pas dégrader le matériel de l'INRA en évaluant correctement tes dimensions... J'en profite pour saluer les anciens doctorants Jimmy et Mathieu, qui se sont éclipsés alors que l'environnement doctoral se féminisait. Je remercie aussi les jeunes du labo, Anaïs et Damien, et les moins jeunes mais tout aussi drôles (sous réserve de comprendre leur humour) Sylvain, Damien et Régis. Merci aussi à Victor pour son aide bienvenue sur les derniers travaux de ma thèse.

Y-a-t'il meilleur concentré de doctorants ailleurs qu'à l'IMT ? Certainement pas. L'ex-bureau 201 en est la preuve vivante. Mes premiers remerciements sont donc destinés à mes anciennes cobureaux : Hélène, partie vivre de nouvelles aventures à Saclay (ton départ a fait du vide !), Anne-Claire, la cobureau du jeudi et vendredi, et ma "demi-sœur de papa n°1" Claire. Bien évidemment, tes petites discussions (et ronchonnades) intempestives me manquent là où je suis ! Je remercie aussi les anciens MIPiens Anne-Charline, Mathieu et Fabien qui ont contribué à rendre ces années (et leurs repas) très agréables. Merci aussi à Mélanie (notre super organisatrice), Malika (la récente joggeuse), Raphaël (le gaucher, je n'ai pas dit gauche !), Yurii (l'âme de Mickaël Jackson), Tatiana (pour ses aventures Savannahiennes), Stéphane (pour s'inquiéter encore et toujours "ça fait longtemps qu'on ne s'est pas vus"), Sébastien (parce que j'espère qu'il mettra

une bonne note à ma sœur), Claire D. (pour son coup de pouce pour les cours) et tous les autres que je ne cite pas mais qui ont su au cours des années renouveler la bonne humeur au labo... Je remercie également Gaëlle avec qui j'ai pris beaucoup de plaisir à travailler. Je suis très fière du travail qu'on a réussi à accomplir ! J'en profite pour saluer Loïc, qui a eu l'immense honneur de goûter mon dentifrice.

*P*our ceux qui m'ont fait découvrir les plaisirs de l'Université à mon arrivée en L3 et pour les (trop ?) nombreuses parties de cartes à la salle des photocopieuses, ou ailleurs, merci à JC, Matthieu, Marie-Anne, Sébastien, Victor et Aurélien, alias Monsieur Patate !

*O*ublier Anne et Auriane, avec qui j'ai souffert pendant deux longues années de prépa, serait vraiment impardonnable. Des remerciements semblent s'imposer aux traumatisées de Rosy, Lacroixes, machine et autres. Ces longues soirées du jeudi, les cours de français et de physique, n'auraient pas été les mêmes sans vous !

*T*ous les patineurs de Tarbes comme de Toulouse sont également remerciés pour m'avoir supportée ou pour supporter encore mon humour grinçant. Mes cibles favorites, Jennifer la vieille, Livie le schtroumpf et Alicia la blonde KK, mes rivaux Pierre et, tout récemment, Rosa, et mes amies les cervelles de piaf Delphine, Jessica et Gwen... Un grand merci particulier à Anne-Sophie et Kévin, qui sont devenus aujourd'hui beaucoup plus que des amis patineurs pour moi !

*T*out un paragraphe ne serait pas de trop pour remercier enfin Gracie, avec qui je partage la même vision du monde, Jack, le cousin biologique de Camille, et Olympe sa meilleure amie. Je remercie Jennifer pour ces parties endiablées de bataille navale et Aline, Pierre, Wiiill, Bonnie, Doriane pour leur soutien depuis l'âge de mes 12 ans. Je souhaite un très bon anniversaire à Max pour ses 45 ans et déjà 2 mois. Une grosse pensée pour Silvia et Justine, que je n'ai pas vue depuis (trop) longtemps, mais que je ne désespère pas de revoir très bientôt.

*E*t je conclus comme il se doit par la famille. Famille restreinte certes, mais soudée oui ! Tout d'abord, merci à mes parents, qui font que c'est encore un vrai plaisir de rentrer le week end, en dépit de leur grand âge, merci à tata Coco et ses rares séjours annuels pendant lesquels on dégoise bien, merci à l'autre Coco, qui est toujours ravi de nous voir, merci aux cousins (jeunes et moins jeunes) éparpillés dans tous les coins de la France et merci à ma grand-mère de Bayonne, qui pète encore la forme malgré ses 87 printemps. J'espère avoir autant d'énergie à ton âge ! Un grand merci particulier à mes deux sisters : Julie, la plus grande, que tous les anciens doctorants connaissent bien. Ces années à Toulouse n'ont pas été de tout repos (j'en ai ramassé des mouchoirs pliés !), mais on s'est bien marrées ! C'est plus maintenant que je vais faire des soirées roues, séries à gogo et vol d'Hector ! Et la plus petite, Camille -dit méga gouffa-, qui m'a récemment rejointe à Toulouse (certains croient même que j'héberge une ado !). Virage à 180° pour la cohabitation : très bien éduquée, elle fait la vaisselle, le ménage et travaille, je recommande vivement. Enfin une qui va redorer l'image des Champions à l'université, dont l'invasion Championesque ne fait que commencer ! !

*R*édiger ces dernières lignes n'est pas une chose facile pour moi. Mes derniers mots sont destinés à ma grand-mère puisque je n'aurais jamais imaginé qu'elle ne soit pas présente pour ma thèse. Il a fallu que tu partes pour que j'ai envie d'en savoir plus sur la vie que tu as mené, et quelle vie ! Berlinoise non francophone, fraîchement débarquée à 20 ans en France dans les années 45-50, pour suivre un garçon rencontré pendant la guerre, il fallait ta force de caractère (et ton grain de folie quand même) pour tenir ! Tu es partie trop vite, tes "binvouis" me manquent mamie !

Table des matières

Chapitre I

Introduction générale

Cette thèse aborde certains problèmes mathématiques posés par l'inférence de réseaux géniques. De tels réseaux sont des outils puissants de représentation et d'analyse de systèmes biologiques complexes à partir de données à haut débit. Ils permettent notamment de rechercher des liens fonctionnels entre entités biologiques (gènes, petites molécules...). Certaines études visent à identifier des liens causaux entre les sommets c'est-à-dire à identifier quel(s) gène(s) modifie(nt) (active(nt) ou inhibe(nt)) l'état d'un autre gène. En médecine, l'étude de tels réseaux de régulation a, par exemple, pour objectif de découvrir de nouveaux traitements susceptibles soit de bloquer une interaction (qui peut conduire une cellule à devenir cancéreuse) soit de favoriser une relation (pouvant aboutir à la destruction d'une cellule cancéreuse). En agronomie, l'objectif est d'extraire des informations sur l'organisation de certains réseaux géniques impliqués par exemple dans la réponse des plantes à différents stress et d'identifier quels gènes (nœuds) peuvent jouer un rôle de facteur clef dans la réponse coordonnée de la plante à son environnement.

Les recherches en génomique ont permis de réaliser des progrès considérables et ont conduit à l'acquisition de nouvelles connaissances qui sont à l'origine de l'ère post-génomique. Dans l'étude des données post-génomiques, une particularité, et par conséquent un des enjeux statistiques, réside dans la très grande dimension de l'espace des paramètres et dans le très petit nombre d'observations disponibles. Typiquement, les jeux de données sont caractérisés par un très grand nombre de gènes (plusieurs milliers), un bruit plutôt important et peu d'observations (quelques centaines dans les meilleurs cas). De plus, les réseaux étudiés dans ce manuscrit s'inscrivent dans le cadre de la génomique génétique, une branche de la biologie des systèmes qui combine des données discrètes et continues, correspondant aux niveaux d'expression des gènes, terme qui sera défini plus précisément par la suite, et à la présence de mutations sur la séquence d'ADN.

Une modélisation possible d'un tel réseau est donnée par l'équation :

$$E = E \cdot B + M \cdot A + \varepsilon,$$

où $E \in \mathcal{M}_{n,p}(\mathbb{R})$ est la matrice des p variables quantitatives observées sur n individus, $M \in \mathcal{M}_{n,m}(\mathbb{R})$ est la matrice des m variables discrètes observées et ε est un bruit gaussien. B à diagonale nulle et A sont des matrices inconnues dont les éléments non nuls décrivent la structure du réseau. En d'autres termes, pour un nœud donné (un gène), on cherche à savoir quels sont les autres gènes et les marqueurs (balises sur le génome) qui ont un effet sur ce gène. La Figure I.1 donne un aperçu du type de réseaux étudiés. Il s'agit donc d'un problème statistique de sélection de variables qui se résout classiquement en optimisant un critère pénalisé. L'objectif de cette thèse est d'estimer les matrices B et A dans le cadre de régression de grande dimension où le nombre d'individus n est petit devant le nombre de gènes p.

Ce chapitre a pour but d'introduire le contexte de cette thèse autour de trois principaux

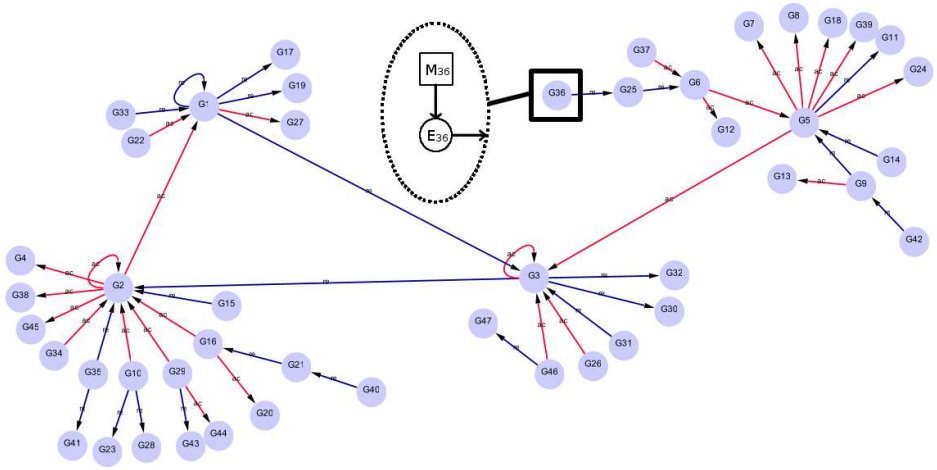


FIGURE I.1 – Une modélisation possible d'un réseau de régulation de gènes. Le marqueur 36 agit sur l'expression du gène 36, elle-même agissant sur l'expression du gène 25. La structure du réseau est inconnue. L'objectif est de la retrouver à partir de données d'observation sur les expressions des gènes et de données marqueurs sur un échantillon de taille n.

concepts, à savoir la sélection de variables pour l'apprentissage statistique supervisé, l'utilisation d'outils d'analyse convexe en statistiques et l'inférence de réseaux de régulation de gènes. Ainsi, la Section 1 propose une introduction générale aux problèmes statistiques d'apprentissage et plus particulièrement aux méthodes de sélection de variables. Dans la Section 2, nous présentons principalement des méthodes d'optimisation permettant de résoudre certains problèmes statistiques. La Section 3 est destinée aux applications de ces méthodes pour l'inférence de réseaux. Après avoir présenté quelques notions de génomique génétique nécessaires à la compréhension de ce manuscrit, une vue d'ensemble des méthodes existant dans la littérature est proposée. La Section 4 concerne enfin les contributions de cette thèse.

1 Méthodes de sélection de variables pour l'apprentissage statistique supervisé

Lorsqu'un phénomène physique, biologique ou autre, est trop complexe pour aboutir à une description analytique et une modélisation déterministe, il est courant d'avoir recours à un ensemble de techniques pour en décrire au mieux le comportement à partir d'une série d'observations. On parle alors de problème d'apprentissage.

On distingue usuellement deux types de problèmes d'apprentissage. Dans le cas d'apprentissage supervisé, les observations fournies se présentent sous la forme de couples entrée-sortie (X, Y), où la sortie Y, qui a été observée sur un même type d'échantillon que l'entrée X, est la variable à prédire. L'objectif est alors de trouver une fonction f susceptible de reproduire Y ayant observé X :

$$Y = f(X) + \varepsilon,$$

où ε symbolise l'erreur de mesure. S'il n'y aucune variable à expliquer, on parle alors de problèmes d'apprentissage non-supervisé, ou plus fréquemment de problèmes de classification non supervisée. Ceux-ci ont pour objectif de partitionner les entrées en plusieurs classes de façon à regrouper entre elles les observations de caractéristiques semblables.

Dans cette thèse, nous nous focaliserons sur des problèmes d'apprentissage statistique supervisé, pour lesquels l'ensemble d'apprentissage est constitué de n observations $Y_i \in \mathcal{Y}$, images de p variables explicatives $X_i^1, ..., X_i^p \in \mathcal{X}$ par une fonction f perturbée par un bruit :

$$\forall i \in [\![1, n]\!], \quad Y_i = f(X_i^1, ..., X_i^p) + \varepsilon_i, \tag{I.1}$$

où les $X_i := (X_i^1, ..., X_i^p)$ $(1 \leq i \leq n)$ sont des vecteurs indépendants et de même loi, et $(\varepsilon_i)_{1 \leq i \leq n}$ est une suite de variables aléatoires centrées, simulant la présence de bruit. En pratique, $\mathcal{X}$ et $\mathcal{Y}$ (domaines dans lesquels évoluent $(X_i)_{1 \leq i \leq n}$ et $(Y_i)_{1 \leq i \leq n}$) seront égaux à $\mathbb{R}$. Notons que les vecteurs $X^j := {}^t(X_1^j, ..., X_n^j)$ $(1 \leq j \leq p)$, où ${}^tU$ désigne la transposée du vecteur U, ne sont pas nécessairement indépendants. Pour un aperçu détaillé des problèmes d'apprentissage statistique supervisé, on pourra se référer à [Vap98], [Bis06] et [HTF09].

Les problèmes de sélection de variables consistent plus précisément à chercher les variables les plus pertinentes pour expliquer et prédire les valeurs prises par la variable à prédire. Nous renvoyons à titre d'information aux travaux de Guyon *et al.* [GE03] et Liu *et al.* [LM07].

La Section 1.1 vise à introduire d'une manière générale les outils et les enjeux de l'apprentissage statistique supervisé. Dans la Section 1.2, nous nous intéressons à des méthodes de sélection de variables dites pénalisées tandis que la Section 1.3 concerne enfin les méthodes basées sur de l'agrégation de modèles.

1.1 L'apprentissage statistique supervisé

L'objectif de cette section est de présenter une vue d'ensemble des techniques liées à l'apprentissage statistique supervisé. Elle permet notamment d'introduire des outils qui seront utiles dans la suite de ce manuscrit.

1.1.1 Risque et prévision

Les performances des méthodes d'apprentissage ou des modèles issus de la même méthode d'apprentissage, s'évaluent par leurs qualités de prévision. Pour prédire la réponse Y, on cherche une fonction $\hat{f}$ appartenant à l'ensemble $\mathcal{F} = \{f : \mathcal{X} \to \mathbb{R}, \quad f \text{ mesurable}\}$ telle que $\hat{Y} := \hat{f}(X)$ est proche de Y. $\hat{f}$ est alors appelé prédicteur, ou règle de prévision de f. La distance entre Y et $\hat{f}(X)$ est mesurée par une fonction dite de perte $l : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}^+$.

Les performances de la règle de prévision $\hat{f}$ sont alors mesurées en terme de risque réel, ou risque théorique, défini par :

$$R(\hat{f}) = \int_{\mathcal{X} \times \mathcal{Y}} l(y, \hat{f}(x)) dP(x, y),$$

où P désigne la loi jointe des observations (inconnue). Le meilleur prédicteur est obtenu en résolvant le problème suivant [Vap95] :

$$f^* = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \quad R(f), \tag{I.2}$$

et est plus connu sous le nom d'oracle.

Le risque empirique est défini comme la moyenne des pertes sur les points de l'échantillon d'apprentissage :

$$\hat{R}_n(\hat{f}) = \frac{1}{n} \sum_{i=1}^{n} l(Y_i, \hat{f}(X_i)).$$

D'après la loi des grands nombres, nous pouvons en déduire que, pour une règle de prévision $\hat{f}$ fixée, le risque empirique converge vers le risque réel lorsque la taille de l'échantillon d'apprentissage tend vers l'infini. On peut ainsi approcher l'oracle f^* par la mesure du minimum du risque empirique :

$$\hat{f} = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \ \hat{R}_n(f). \tag{I.3}$$

Remarquons que la minimisation du risque empirique sur l'ensemble des règles de prévision $\mathcal{F}$ possibles peut parfois s'avérer non judicieuse. Si le prédicteur $\hat{f}$ associé à (I.3) s'ajuste parfaitement aux données, il n'est pas forcément capable de s'appliquer à des données qui n'ont pas participé à son estimation. Ceci conduit au phénomène de sur-apprentissage. Si l'on souhaite faire de la prévision, il apparaît que le meilleur modèle n'est alors pas toujours celui qui ajuste le mieux le vrai modèle. Le choix du modèle est basé sur des critères de qualité de prévision pouvant par exemple privilégier des modèles de plus petite complexité, ou plus parcimonieux.

Considérons donc un sous-ensemble $\mathcal{H}$ de l'ensemble des règles de prévision possibles, permettant de restreindre l'espace de recherche pour $\hat{f}$. L'objectif peut alors consister à trouver $f^*_{\mathcal{H}}$ satisfaisant :

$$f^*_{\mathcal{H}} = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \ R(f),$$

approché par :

$$\hat{f} = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \ \hat{R}_n(f). \tag{I.4}$$

Supposons que l'espace d'hypothèses $\mathcal{H}$ soit fini, par exemple $\mathcal{H} = \{h_1, ..., h_M\}$. La règle de prévision $\hat{f}$ définie par l'Equation (I.4) imite alors l'oracle $f^*_{\mathcal{H}}$ associé à $\mathcal{H}$, puisqu'elle satisfait l'inégalité oracle suivante : pour tout $0 < \delta < 1$, avec probabilité au moins $1 - \delta$,

$$R(\hat{f}) \leq \min_{1 \leq j \leq M} R(h_j) + \sqrt{\frac{2}{n} \log\left(\frac{2M}{\delta}\right)}.$$

La théorie de Vapnik et Chervonenkis a permis d'étendre ces résultats à des espaces $\mathcal{H}$ plus généraux, notamment pour des problèmes de classification (pour plus de détails, se référer à [Vap95]). Elle fait principalement intervenir la notion de dimension de Vapnik-Chervonenkis (ou dimension VC) de $\mathcal{H}$, notée $\mathrm{VC}_{\mathcal{H}}$, qui mesure la richesse de l'espace $\mathcal{H}$. Si $\mathcal{H}$ est de VC-dimension finie, pour tout $0 < \delta < 1$, avec probabilité au moins $1 - \delta$, l'inégalité suivante est satisfaite :

$$R(\hat{f}) \leq \min_{f \in \mathcal{H}} R(f) + 4\sqrt{\frac{2(\mathrm{VC}_{\mathcal{H}} \log(n+1) + \log 2)}{n}} + \sqrt{\frac{2 \log(1/\delta)}{n}}.$$

La différence entre le risque réel de notre estimateur $\hat{f}$, donné par l'Equation (I.4), et celui de l'oracle peut se décomposer sous la forme :

$$R(\hat{f}) - R(f^*) = \underbrace{R(\hat{f}) - R(f^*_{\mathcal{H}})}_{\text{Erreur d'estimation}} + \underbrace{R(f^*_{\mathcal{H}}) - R(f^*)}_{\text{Erreur d'approximation}}.$$

L'erreur d'estimation est aléatoire, de par sa dépendance aux données. Elle permet de quantifier la difficulté d'estimer $f^*_{\mathcal{H}}$ et correspond à un terme de variance. L'erreur d'approximation (ou biais) mesure quant à elle à quel point l'espace d'hypothèse $\mathcal{H}$ est proche de la cible f^* et ne dépend pas des données.

Notons que plus l'espace d'hypothèse $\mathcal{H}$ est grand, plus l'erreur d'approximation peut être petite, mais plus l'erreur d'estimation peut être grande. En d'autres termes, plus un modèle est

complexe, plus il intègre de paramètres et plus il est capable de s'ajuster aux données (biais réduit) mais moins il est capable de s'appliquer à des données qui n'ont pas participé à son estimation. Un des enjeux en apprentissage statistique supervisé consiste alors à trouver un juste équilibre entre biais et variance. Une alternative possible à la minimisation du risque empirique consiste à ce titre à utiliser des algorithmes d'approximation, tels que l'algorithme $\mathbb{L}_2$-Boosting, largement étudié dans ce manuscrit. Comme nous le verrons par la suite, ils permettent de contrôler d'une part le biais d'estimation de f et, d'autre part, la variance d'approximation de f. De manière plus générale, cette problématique a principalement conduit à l'élaboration de méthodes de sélection de variables que nous présentons dans les Sections 1.2 et 1.3.

1.1.2 La grande dimension et le recouvrement du support

Un deuxième enjeu de la théorie de l'apprentissage statistique repose sur le principe de grande dimension, lorsque le nombre de variables p observées est très important devant la taille de l'échantillon n. Les travaux de Vapnik [Vap98] en théorie de l'apprentissage statistique ont conduit à s'intéresser à la présence de propriétés théoriques évaluant les performances des méthodes d'estimation lorsque l'on fait croître la taille de l'échantillon vers l'infini. Ce cadre d'étude s'est imposé en apprentissage ces dernières années avec l'émergence du *big data*, ou mégadonnées, qui se caractérise par le fait que les jeux de données sont gigantesques tant par leurs tailles (n grand) que par la dimension des données (p grand). Cependant, s'il est vrai que les résultats présentés concernent finalement le cas où $n, p \to +\infty$ à une vitesse maîtrisée, il faut souligner le fait que le problème central pour la reconstruction de réseaux biologiques se rapporte à des tailles d'échantillon n petites devant le nombre d'observations p.

Les principaux résultats que l'on peut espérer obtenir font alors intervenir des hypothèses concernant notamment un contrôle du nombre de variables $p := p_n$ en fonction de n. Parmi les propriétés théoriques basiques, on trouve celles qui concernent :

- la vitesse de convergence d'un estimateur qui nous donne une idée du comportement d'un estimateur, ou du risque de cet estimateur, lorsque n tend vers l'infini,
- la consistance de l'estimateur. La consistance est un outil plus fin que la convergence. Elle garantit la convergence en probabilité d'un estimateur vers la valeur théorique (inconnue mais supposée existante),
- les inégalités oracles. Elles permettent de comparer le risque de l'estimateur avec le risque de l'oracle, défini suivant l'Equation (I.2). Elles s'écrivent sous la forme :

$$R(\hat{f}) \leq (1+\eta)R(f^*) + r(n,p),$$

où $\eta \geq 0$ et $r(n,p)$ est un terme résiduel négligeable devant $R(f^*)$, pouvant dépendre de η. Dans le cas où $\eta = 0$, on parle d'inégalités oracles précises.

Les questions autour de la sélection de variables et de la grande dimension ont également conduit à des études concernant la parcimonie du signal reconstruit. En biologie des systèmes par exemple, les mécanismes sont beaucoup trop complexes pour qu'on en ait une appréhension fine. On cherche donc à identifier dans un premier temps les gènes clés, ceux qui sont essentiels au mécanisme étudié. L'hypothèse de parcimonie (ou de sparsité) signifie que la fonction f dépend d'un petit nombre de variables seulement. On appelle support de la fonction f, et on note $\mathcal{S}_f$ l'ensemble des variables dont dépend la fonction f. La sparsité, ou parcimonie, de f est alors définie par

$$S(f) = |\mathcal{S}_f|,$$

où $|.|$ est la notation pour le cardinal d'un ensemble.

Par analogie avec la consistance d'un estimateur, la consistance du support consiste à montrer la convergence en probabilité du support $\mathcal{S}_{\hat{f}}$ de l'estimateur $\hat{f}$ vers le vrai support de f et permet d'assurer un bon comportement de l'estimateur.

1.1.3 Vers une approche multi-tâches

On peut enfin parfois être amenés à considérer non plus une, mais plusieurs tâches de régression sur un même espace. Par analogie avec le modèle présenté au travers de l'Equation (I.1), la régression multi-tâches consiste à trouver une fonction f susceptible de reproduire m réponses $Y = (Y^1, ..., Y^m)$ ayant toujours observé X. Chacune des m coordonnées de Y est modélisée, suivant [HTF09], par une relation linéaire du type :

$$\forall i \in [\![1, m]\!], \quad Y^i = f^i(X) + \varepsilon^i, \tag{I.5}$$

où $(\varepsilon^i)_{1 \leq i \leq m}$ sont des termes de bruit supposés gaussiens, tels que ε^i est indépendant de ε^j, pour tout $i \neq j$.

Pour résoudre des problèmes d'apprentissage multi-tâches, une solution naturelle consiste à traiter ces différentes tâches indépendamment en résolvant m régressions. A tour de rôle, chaque coordonnée f^i de f est estimée à partir de la réponse Y^i correspondante et la matrice de design X. On peut cependant espérer gagner en efficacité en les considérant simultanément, du fait de leur possible similarité. Les enjeux statistiques pour ce type de modèle sont les mêmes que pour la régression uni-tâche : proposer un estimateur $\hat{f}$ de la fonction multivariée f et une méthode d'estimation numérique, garantissant de bonnes propriétés théoriques. Ce sujet sera l'objet du Chapitre II de ce mémoire.

1.2 Méthodes de sélection de variables pénalisées

L'apprentissage statistique supervisé a pour objectif de trouver une règle de prévision $\hat{f}$ suivant des critères de qualité présentés dans la Section 1.1. On pourrait penser que plus on augmente le nombre de variables décrivant chaque observation d'un échantillon, plus on dispose d'informations concernant ces observations et plus on en facilite et on améliore l'apprentissage du modèle. Cependant, la qualité de la prévision ne dépend pas du nombre d'informations à disposition mais essentiellement de la pertinence de ces informations. La sélection de variables est un processus très important en apprentissage statistique supervisé : à partir d'une série de variables candidates, le statisticien cherche les variables les plus pertinentes pour expliquer ou prédire les valeurs prises par la variable à prédire. Ceci conduit à rechercher des modèles parcimonieux, qui ont un nombre restreint de variables explicatives.

Il y a principalement deux méthodes de sélection de variables : la première consiste à simplifier le modèle appris en réduisant son nombre de variables (critères et algorithmes de sélection de variables, pénalisation en norme ℓ_1) tandis que la deuxième consiste à contraindre les paramètres du modèle en les réduisant (pénalisation en norme ℓ_2).

1.2.1 Critères de choix de modèles

Les premiers critères de choix de modèles apparaissant dans la littérature sont les critères de validation croisée PRESS [All71], le C_p de Mallows [Mal73], le critère d'information AIC [Aka74] ou le critère bayésien BIC [Sch78].

Le critère PRESS est l'ancêtre de la validation croisée dans sa version actuelle. Il s'appuie sur le principe qu'il ne faut pas utiliser le même échantillon à la fois pour construire et évaluer l'estimateur. Si l'on note $\hat{f}\left(X_{(i)}\right)$ la prévision de Y calculée sans tenir compte de la i-ème

observation $(Y_i, X_i^1, ..., X_i^p)$, la somme des erreurs quadratiques de prévision, ou critère PRESS, est définie par :

$$\text{PRESS} = \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \hat{f}\left(X_{(i)}\right) \right)^2.$$

La minimisation de ce critère permet de sélectionner des modèles ayant de bons pouvoirs prédictifs mais elle peut parfois être lourde à calculer pour des modèles complexes.

Une généralisation du critère PRESS consiste à couper aléatoirement l'échantillon d'origine en k groupes. L'échantillon test, permettant de calculer l'erreur faite sur chaque estimateur, est alors constitué à tour de rôle de l'un des k groupes. Les $k-1$ autres groupes, permettant d'estimer le paramètre, constituent l'ensemble d'apprentissage. Le modèle choisi est celui qui minimise l'erreur moyenne de prévision sur les échantillons tests. On parle alors de k-fold validation croisée. Le choix de k entre 5 et 15 est couramment $k = 10$ [MDA04]. Cette valeur est par exemple implémentée par défaut dans le logiciel R.

On peut parfois privilégier des critères dont le calcul est immédiat. C'est le cas par exemple du C_p de Mallows [Mal73], défini par :

$$C_p = \frac{\sum_{i=1}^{n} \left(Y_i - \hat{f}(X_i) \right)^2}{\hat{\sigma}^2} + 2p - n,$$

où $\hat{\sigma}^2$ est un estimateur de la variance de l'erreur de mesure ε. Dans le cas d'un modèle complet (non pénalisé), le C_p de Mallows vaut p. Il est alors d'usage de rechercher un modèle qui minimise le C_p de Mallows tout en fournissant une valeur proche de p.

Enfin, les derniers critères largement utilisés dans la littérature sont basés sur une forme pénalisée de la vraisemblance du modèle afin de favoriser des modèles parcimonieux :

$$\text{Crit}(\lambda) = -2 \log L + \text{pen}(\lambda),$$

où L est la vraisemblance du modèle considéré et $\text{pen}(\lambda)$ est une pénalité choisie au préalable. Le critère AIC, par exemple, fait appel à une pénalité correspondant au double du nombre de paramètres k du modèle $\text{pen}(\lambda) = 2k$. Il est très proche du C_p de Mallows, et en est même un équivalent dans le cas du modèle linéaire et si la variance des observations est connue. Une variante possible du critère AIC, donnée par le critère BIC, consiste à pénaliser les modèles plus complets en ajoutant à la vraisemblance une pénalité de l'ordre de $\log(n)k$.

Dans certains problèmes de sélection de variables, il peut parfois être souhaitable de laisser croître la taille du modèle, ou la complexité de l'espace sur lequel on minimise, avec le nombre d'observations. L'enjeu principal de ces méthodes pénalisées consiste alors à trouver une pénalité qui garantit une performance de sélection optimale. Les travaux de Birgé *et al.* [BM07], consacrés à l'étude des méthodes de calibration automatique de pénalités en sélection de modèles, sont basés sur une heuristique, appelée heuristique de pente, et ont permis de mettre en place un algorithme de calibration de pénalités optimales. Pour plus de détails, on pourra consulter [Mas07].

1.2.2 Algorithmes de sélection de variables

Les méthodes de sélection de variables consistent à rechercher le meilleur sous-ensemble de variables au sens d'un des critères précédents. La façon la plus simple de procéder est de tester tous les sous-ensembles de variables possibles, mais cela devient vite impossible lorsque p est grand.

Supposons que l'on dispose d'un système de fonctions $(g_j)_{j=1,\dots,p}$, encore appelé dictionnaire tel que la fonction f donnée par l'Equation (I.1), se décompose sous la forme :

$$f(X) = \sum_{j=1}^{p} \theta_j^0 g_j(X).$$

Notons alors $\theta^0 = (\theta_j^0)_{1\leq j\leq p}$ le vecteur dont les composantes sont les θ_j^0. Supposons de plus que nous désirions approximer le modèle d'intérêt par la représentation $\sum_{j=1}^{p} \theta_j g_j(X)$, où un petit nombre seulement des variables $(\theta_j)_{j=1,\dots,p}$ sont non nulles. Dans le cas où p est grand, les méthodes gloutonnes sont des méthodes d'estimation efficaces permettant de trouver une solution à ce problème d'apprentissage. Elles sont basées sur des choix de solutions locales optimales d'un problème dans le but d'obtenir une solution globale de ce problème. Ces algorithmes sont souvent utilisés en intelligence artificielle pour résoudre des problèmes d'optimisation combinatoire, de par leur implémentation intuitive et leur rapidité d'exécution.

Parmi les algorithmes gloutons les plus connus, on peut citer l'algorithme *Forward* qui consiste pas à pas à ajouter au modèle le prédicteur qui minimise le résidu, mesurant l'ajustement de la régression. La procédure s'arrête lorsque toutes les variables sont introduites ou lorsque la valeur des variables qu'il reste à ajouter au modèle ne dépasse pas un seuil donné. L'inconvénient de l'algorithme *Forward* est principalement que lors d'une itération donnée, il ne permet pas de corriger les erreurs de sélection faites durant les précédentes itérations, il est donc difficile de justifier l'optimalité globale de l'estimateur obtenu. Dans le but de corriger ces problèmes, l'algorithme *Backward* consiste cette fois à retirer pas à pas du modèle le prédicteur le moins informatif. Il démarre donc avec le modèle complet. Cette méthode est cependant plus coûteuse et très sensible au phénomène de surajustement de données.

Dans un but amélioratif, Zhang [Zha11] propose alors de mixer ces deux algorithmes *Forward* et *Backward* en un seul algorithme, l'*Adaptive Forward-Backward Greedy Algorithm* (FoBa). L'algorithme FoBa permet d'estimer le modèle de manière itérative en ajoutant à chaque étape le prédicteur le plus important au modèle (étape *Forward*) et en enlevant celui qui est jugé le moins informatif (étape *Backward*). Ces méthodes souffrent cependant d'instabilités numériques [Bre95], [Tib96].

1.2.3 Régularisation en norme ℓ_2 : la régression *ridge*

Afin d'introduire du biais dans l'estimateur pour en améliorer les propriétés théoriques, on peut procéder à une régularisation en norme ℓ_2. L'estimateur *ridge*, introduit par Hoerl *et al.* [HK70], est défini par un critère des moindres carrés auquel on ajoute une pénalité en norme ℓ_2 :

$$\hat{\theta}_{ridge} = \underset{\theta\in\mathbb{R}^p}{\operatorname{argmin}} \left\| Y - \sum_{j=1}^{p} \theta_j g_j \right\|_2^2 + \lambda \|\theta\|_2 ,$$

où $\lambda > 0$ est le paramètre de pénalisation. Par dérivation matricielle, on obtient une solution explicite à ce problème de minimisation sous la forme :

$$\hat{\theta}_{ridge} = \left({}^tXX + \lambda I_p\right)^{-1} {}^tXY,$$

où ${}^tX$ désigne la matrice transposée de X. Par opposition avec l'estimateur des moindres carrés, obtenu avec $\lambda = 0$, l'estimateur *ridge* n'est pas sensible au mauvais conditionnement de la matrice ${}^tXX$. Cependant, la régression *ridge* ne permet pas de faire de la sélection de variables, elle contraint seulement la norme du paramètre θ en limitant la taille de ses composantes, on parle alors d'effet de *shrinkage*.

1.2.4 Régularisation en norme ℓ_1 : la régression *Lasso*

Dans le but de rendre parcimonieux le signal reconstruit, c'est-à-dire d'en limiter le nombre de composantes non nulles, on ajoute au critère des moindres carrées une pénalité en norme ℓ_1. Il s'agit de l'estimateur *Lasso* [Tib96], largement étudié dans la communauté statistique :

$$\hat{\theta}_{Lasso} = \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \left\| Y - \sum_{j=1}^{p} \theta_j g_j \right\|_2^2 + \lambda \|\theta\|_1 .$$

Comme l'indique la Figure I.2 suivante, la norme ℓ_1 permet d'affiner la propriété de *shrinkage* de la régression *ridge*. Plus précisément, elle permet d'écraser les coefficients estimés vers 0 afin de produire des solutions parcimonieuses. Comme dans le cas de la régression *ridge*, si le paramètre de pénalisation λ vaut 0, on retrouve l'estimateur des moindres carrés. Si λ tend au contraire vers l'infini, on annule l'ensemble des composantes de $\hat{\theta}_{Lasso}$.

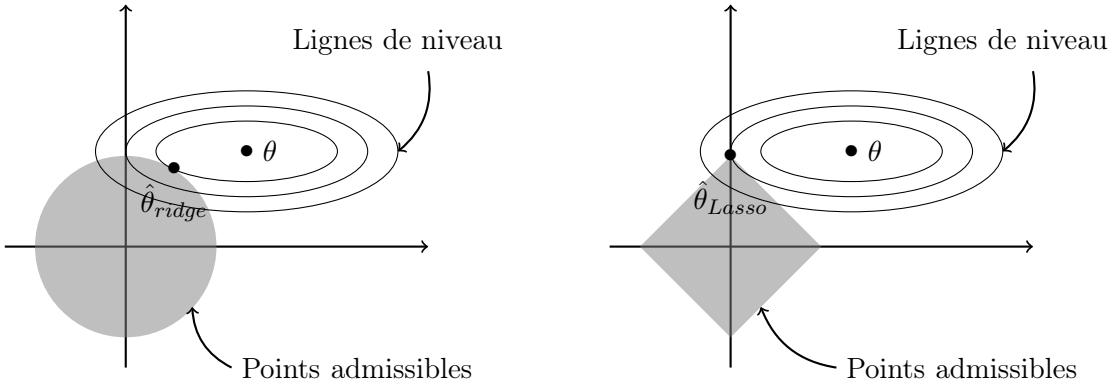


FIGURE I.2 – L'estimateur *Lasso* produit beaucoup de coefficients nuls. Les zones grises correspondent aux espaces de contrainte ($\|\theta\|_1 \leq \lambda$ pour l'estimateur *Lasso* et $\|\theta\|_2 \leq \lambda$ pour l'estimateur *ridge*), tandis que les ellipses représentent les lignes de niveau de la norme euclidienne. Les ellipses sont plus facilement en contact avec une face de la boule ℓ_1 de faible dimension, le point de contact correspondant à l'estimateur *Lasso* a donc plus de coefficients nuls.

De nombreux résultats théoriques ont été établis dans la littérature statistique mais ceux-ci ont été obtenus au prix d'hypothèses plus ou moins contraignantes concernant notamment la matrice de Gram $\Psi = {}^tXX$, en particulier, la condition de valeur propre restreinte [BRT09] suivante :

Hypothèse 1 (Condition Re(s,c_0))**.** *Une matrice $X \in \mathcal{M}_{n,p}(\mathbb{R})$ satisfait la condition de valeur propre restreinte Re(s,c_0) si :*

$$\kappa(s, c_0) = \min_{\substack{\mathcal{S} \subset \{1,\dots,p\} \\ |\mathcal{S}| \leq s}} \min_{\substack{\delta \neq 0 \\ \|\delta_{\mathcal{S}^C}\|_1 \leq c_0 \|\delta_{\mathcal{S}}\|_1}} \frac{\|X\delta\|}{\sqrt{n}\,\|\delta_{\mathcal{S}}\|} > 0,$$

où $\delta_{\mathcal{S}}$ désigne le vecteur δ restreint aux colonnes dont les éléments appartiennent à $\mathcal{S}$ et $\mathcal{S}^C$ est le complémentaire de $\mathcal{S}$.

Sous la condition que cette hypothèse soit satisfaite, Bickel *et al.* [BRT09] ont obtenu une inégalité oracle en prédiction pour l'estimateur *Lasso* :

Théorème 1.1 (Inégalité oracle en prédiction, [BRT09]). *On considère l'estimateur Lasso correspondant à la pénalité* $\lambda = A\sigma\sqrt{\frac{\log p}{n}}$, *avec* $A > 2\sqrt{2}$. *Supposons que la condition Re(s,*$3+4/\eta$*) est satisfaite pour la matrice* $D := (g_j(X_i))_{1\leq i\leq n, 1\leq j\leq p}$ *avec* $\eta > 0$. *Avec probabilité au moins* $1 - p^{1-A^2/8}$, *il existe alors* C_η *dépendant uniquement de* η *tel que :*

$$\frac{1}{n}\left\|X\hat{\theta}_{Lasso} - X\theta^0\right\|^2 \leq (1+\eta)\inf_{\substack{\theta\in\mathbb{R}^p \\ \|\theta\|_1\leq s}}\left\{\frac{1}{n}\left\|X\theta - X\theta^0\right\|^2 + \frac{C_\eta A^2\sigma^2 s}{\kappa^2(s, 3+4/\eta)}\frac{\log p}{n}\right\}.$$

L'algorithme le plus populaire pour résoudre le critère de minimisation correspondant à l'estimateur *Lasso* est le LARS (pour plus de détails sur cet algorithme, se référer à [EHJT04]).

L'estimateur *Lasso* a été largement étudié dans la littérature, ses limites théoriques et algorithmiques sont à présent bien déterminées. D'un point de vue théorique, les résultats relatifs à l'estimateur *Lasso* reposent ainsi sur une hypothèse implicite de faible corrélation des variables explicatives (hypothèse sur la matrice de Gram). Dans des problèmes d'estimation avec fortes corrélations entre les variables, l'algorithme LARS échoue à reconstituer le modèle. De nombreux auteurs se sont alors inspirés des travaux autour du *Lasso* pour lui apporter des améliorations.

1.2.5 Les dérivées du *Lasso*

Une première méthode dérivée du *Lasso* consiste à combiner la régression *ridge* et la régression *Lasso* en ajoutant au modèle une pénalité en norme ℓ_1 et ℓ_2. Il s'agit de l'*Elastic Net* [ZH05] :

$$\hat{\theta}_{EN} = \underset{\theta\in\mathbb{R}^p}{\operatorname{argmin}}\left\|Y - \sum_{j=1}^p \theta_j g_j\right\|^2 + \lambda\left(\alpha\|\theta\|_1 + (1-\alpha)\|\theta\|_2^2\right).$$

Par rapport à l'estimateur *Lasso*, le terme de pénalisation en norme ℓ_2 encourage les variables fortement corrélées à être moyennées, tandis que la pénalisation en norme ℓ_1 assure une solution de dimension limitée.

Il peut aussi parfois être intéressant de prendre en compte la structure de groupe des données : supposons que la matrice du dictionnaire $D := (g_1, ..., g_p)$ soit constituée de q blocs de tailles respectives $p_1, ..., p_q$ telles que $\sum_{i=1}^q p_i = p$. L'estimateur *Group Lasso* $\hat{\theta}_{GL}$ [YL06] est défini pour un certain $\lambda > 0$ comme solution du problème d'optimisation :

$$\hat{\theta}_{GL} = \underset{\theta=(\theta^1,...,\theta^q)}{\operatorname{argmin}}\frac{1}{2}\left\|Y - \sum_{j=1}^p \theta_j g_j\right\|_2^2 + \lambda\sum_{j=1}^q p_j^{1/2}\left\|\theta^j\right\|_2.$$

Dans le cas où D est constitué de blocs de taille unitaire, ce problème d'optimisation coïncide avec le problème d'optimisation lié à l'estimateur *Lasso*. Dans le cas contraire, le paramètre estimé a effectivement tendance à privilégier des structures de groupe.

Une application intéressante de l'estimateur *Group Lasso* concerne la régression multi-tâches donnée par l'Equation (I.5). Soit $\mathcal{D}$ le dictionnaire de fonctions $(g_j)_{j=1,...,p}$ permettant de décomposer chacune des fonctions (ou tâches) f_i $(1 \leq i \leq m)$. L'objectif à atteindre est l'approximation de chaque f_i sous la forme :

$$\forall i \in [\![1, m]\!], \quad \hat{f}_i = \sum_{j=1}^p \hat{\theta}_{i,j} g_j.$$

Pour chacune des tâches considérées, notons $\mathcal{S}_i$ le support de f_i :

$$\mathcal{S}_i = \{j \in [\![1, p]\!], \theta_{i,j} \neq 0\}.$$

Il est évidemment naturel de s'attendre à ce que les supports $\mathcal{S}_i$ $(1 \leq i \leq m)$ se chevauchent : $\cap_{1\leq i\leq m}\mathcal{S}_i \neq \emptyset$. Le problème d'estimation peut alors être traité en estimant, non pas chacun des supports $\mathcal{S}_i$ $(1 \leq i \leq m)$ indépendamment les uns des autres, mais en estimant un support global de $f := (f_1, ..., f_m)$, c'est-à-dire l'ensemble des variables intervenant dans au moins l'une des régressions, puis en estimant a posteriori les supports individuels (pour plus de détails, voir [OWJ11]). On se ramène ainsi à un problème plus simple puisque l'estimation de l'union des supports $\cup_{1\leq i\leq m}\mathcal{S}_i$ revient à estimer des groupes de variables.

Des résultats théoriques ont été obtenus pour l'application du *Group Lasso* à ce type de structures au prix d'hypothèses concernant la matrice de covariance de X. Obozinski *et al.* [OWJ11] ont ainsi mis en évidence l'existence d'un seuil, dépendant de n, p et la parcimonie totale s de f, $s = \sum_i |\mathcal{S}_i|$, en-dessous duquel le *Group Lasso* parvient à reconstruire exactement le support du signal avec grande probabilité.

1.3 Méthodes basées sur de l'agrégation de modèles

Face au très grand nombre de méthodes d'apprentissage statistique présentes dans la littérature, a emergé l'idée de les agréger pour tirer le meilleur parti de leurs avantages respectifs (voir par exemple [Vov90] et [LW94]). Parmi les principales procédures utilisées, on trouve celles qui reposent sur une construction aléatoire d'une famille de modèles, telles que le Bagging [Bre95] ou les forêts aléatoires [Bre01], et celles qui reposent sur une construction adaptative d'une famille de modèles, comme par exemple le Boosting [Fre90]. Ces procédures sont encore aujourd'hui largement plébiscitées pour leurs bonnes performances expérimentales (voir par exemple [Gha99], [RKA06]).

1.3.1 Le Bagging (ou Bootstrap aggregating)

Le principe du Bagging est le suivant : étant donné un échantillon $(X_i, Y_i)_{1\leq i\leq n}$ de taille n, on génère q échantillons en effectuant n tirages indépendants avec remise dans l'échantillon initial. Ce type d'échantillon est appelé échantillon Bootstrap. On construit alors un estimateur agrégé en moyennant les résultats obtenus pour les modèles associés à chacun de ces échantillons. Le Bagging a principalement pour effet de réduire la variance globale du modèle.

Le Bagging a donné naissance à toute une classe de familles de modèles. Ainsi, Bach [Bac08] a proposé une version Bootstrappée du *Lasso* dont l'idée est la suivante : pour une valeur donnée du paramètre de pénalisation λ du *Lasso*, on construit q estimateurs $(\hat{\mathcal{S}}_k)_{k=1...q}$ du support $\mathcal{S}$ de f à partir de q échantillons bootstraps. L'estimateur *Bolasso* est alors construit sur l'intersection de ces q ensembles :

$$\hat{\mathcal{S}}_{Bolasso} = \bigcap_{k\in\{1,...,q\}} \hat{\mathcal{S}}_k.$$

Les coefficients de régression sont enfin estimés par les moindres carrés. D'un point de vue théorique, Bach [Bac08] a montré la consistance du support avec grande probabilité, sous des hypothèses moins contraignantes que pour le *Lasso*.

1.3.2 Les forêts aléatoires

Dans le cas d'apprentissage par arbres binaires, les méthodes CART (Classification And Regression Trees) ont été introduites par Breiman *et al.* [BFOS84]. A chaque étape de cet algorithme, on partitionne une partie de l'espace en deux sous-parties. On associe à ce partitionnement un arbre binaire dont les nœuds sont associés aux éléments de cette partition, et une règle de découpe

d. La première étape de l'algorithme consiste alors à sélectionner la meilleure découpe d, c'est-à-dire celle qui minimise une fonction de coût donnée. Les arbres sont ainsi développés jusqu'à atteindre une règle d'arrêt. Une règle d'arrêt classique consiste par exemple à ne pas découper des nœuds qui contiennent moins qu'un certain nombre d'enfants. La deuxième étape de l'algorithme est l'élagage : parmi l'arbre maximal (entièrement développé), on cherche le meilleur sous-arbre, au sens de l'erreur de prédiction, permettant de rendre parcimonieux le modèle appris.

Pour améliorer les performances de ces méthodes d'apprentissage, Breiman [Bre01] propose d'ajouter une phase de randomisation : ce sont les forêts aléatoires. Le principe des forêts aléatoires est de faire de l'agrégation d'arbres binaires : en ajoutant du hasard dans le choix des variables qui interviennent dans les modèles, on espère ainsi améliorer les performances des méthodes d'estimation. On commence par générer des échantillons Bootstrap de l'échantillon de départ. Pour chacun de ces échantillons, une variante de CART est appliquée : pour découper un nœud, on cherche la meilleure coupure suivant un nombre aléatoire q de variables et l'arbre obtenu n'est pas élagué. La collection d'arbres est alors agrégée pour donner le prédicteur. L'entier q est fixé au début de la procédure et est le même pour tous les arbres construits mais les q variables servant à trouver la meilleure découpe sont choisies aléatoirement.

Les forêts aléatoires sont des méthodes d'estimation aux performances curieusement exceptionnelles, pour lesquelles on ne connaît à l'heure actuelle que très peu de résultats théoriques. Dans le cadre de la classification, des résultats de consistance ont notamment été obtenus par Biau *et al.* [BDL08]. Notons k le nombre de découpes effectuées et n le nombre d'observations de l'échantillon.

Théorème 1.2 ([BDL08]). *Si $k \to +\infty$ et $\frac{k}{n} \to 0$, alors le classifieur construit à partir des forêts aléatoires est consistant.*

L'idée de ce résultat est qu'il faut découper beaucoup d'arbres ($k \to +\infty$) pour réduire le biais, mais qu'il reste assez d'observations dans les feuilles des arbres ($\frac{k}{n} \to 0$) pour contrôler la variance.

1.3.3 Le Boosting

L'idée de base du Boosting est de combiner un ensemble de classifieurs en améliorant adaptivement les compétences des plus faibles d'entre eux. La méthode originale de Schapire [Sch90] a été améliorée par Schapire *et al.* [SF96] par le biais de l'algorithme Adaptive Boosting (*AdaBoost*) pour la prévision d'une variable binaire.

Le Boosting s'appuie sur le même principe que le Bagging : il construit un ensemble de classifieurs qui sont ensuite agrégés par une moyenne pondérée des résultats. Cependant, dans le cas du Boosting, cet ensemble de classifieurs est construit d'une façon récurrente et itérative. Plus précisement, chaque classifieur est une version adaptative du précédent en donnant plus de poids aux observations mal prédites. Pour le cas de la régression, Schapire *et al.* [SF96] a proposé l'algorithme AdaBoost.R.

Dans la littérature, on trouve plusieurs variantes des algorithmes de Boosting qui diffèrent par leurs façons de pondérer les observations mal prédites, leurs façons d'agréger les modèles ou leurs fonctions de perte. Ainsi, si l'AdaBoost est basé sur une fonction de perte exponentielle, le LogitBoost [FLNP00a] fait appel à une fonction de perte logistique, et le $\mathbb{L}_2$-Boosting [BY03], à une fonction de perte en norme ℓ_2. Notons que ce dernier est largement étudié dans ce manuscrit dans le cadre de la régression, pour laquelle nous ne faisons pas d'agrégation de modèles.

De la même manière que le Bagging, le Boosting permet de réduire la variance du modèle, mais également son biais, grâce à son étape d'agrégation. De même, les forêts aléatoires sont basées sur des modèles de faible biais (arbres complets) et permettent elles aussi de réduire

significativement la variance. Les performances numériques de ces deux méthodes sont donc sensiblement les mêmes.

2 Outils d'analyse convexe en statistiques

Rappelons la problématique qui nous intéresse. Nous avons à disposition n observations $(X_i, Y_i)_{1\leq i\leq n} \in \mathcal{X} \times \mathcal{Y}$ i.i.d., générées suivant le modèle :

$$\forall i \in [\![1, n]\!], \quad Y_i = f(X_i^1, ..., X_i^p) + \varepsilon_i,$$

où $(\varepsilon_i)_{1\leq i\leq n}$ est une suite de variables aléatoires indépendantes et centrées, indépendantes de tous les X_i et modélisant la présence de bruit sur la réponse Y.

L'objectif consiste à approximer la fonction f par une fonction linéaire d'éléments d'un dictionnaire $(g_j(X))_{j=1,...,p}$ dépendant des observations $X := (X^1, ..., X^p)$:

$$\hat{f}(X) = \sum_{j=1}^{p} \theta_j g_j(X). \tag{I.6}$$

Pour trouver une règle de prévision $\hat{f}$ susceptible d'avoir produit Y à partir des observations X, une première méthode consiste à minimiser le risque empirique $\hat{R}_n(f)$ défini par :

$$\hat{R}_n(f) = \frac{1}{n} \sum_{i=1}^{n} l(Y_i, \hat{f}(X_i)),$$

où l est une fonction de perte. Pour éviter les phénomènes de sur-apprentissage (voir Section 1), on s'intéressse plus particulièrement à la minimisation du risque empirique sous contraintes :

$$\hat{f} = \underset{f\in\mathcal{H}}{\operatorname{argmin}} \ \hat{R}_n(f), \tag{I.7}$$

où $\mathcal{H}$ désigne l'espace de recherche contraint. Nous avons présenté dans la section précédente des méthodes dites de sélection de variables nous permettant d'estimer f en ne conservant que les variables les plus pertinentes du modèle. Nous nous intéressons ici plus particulièrement aux techniques d'optimisation utilisées pour résoudre des problèmes d'optimisation plus généraux.

La Section 2.1 rappelle des notions d'optimisation utilisées dans mes travaux de thèse. Dans la Section 2.2, nous présentons les méthodes de descente, mises en œuvre pour résoudre des problèmes d'optimisation différentiables. La Section 2.3 concerne les notions de convergence et de complexité des algorithmes d'optimisation. Dans la Section 2.4, nous nous intéressons enfin à une adaptation des algorithmes gloutons à l'optimisation convexe.

2.1 Rappels d'optimisation

Dans cette section, nous présentons des notions d'optimisation nécessaires à la compréhension de la suite de cette introduction. Un problème d'optimisation peut être formulé comme suit :

$$\min_{x\in E\subset F} f(x), \tag{I.8}$$

où F est un espace de Banach, E est un sous-ensemble de F correspondant à l'ensemble des contraintes et f est une fonction de $E \subset F$ dans $\mathbb{R}$ supposée différentiable. f est appelée la fonction objectif ou fonction coût. Résoudre l'Equation (I.8) consiste à trouver une solution locale (faute de mieux) à ce problème.

2.1.1 Eléments d'analyse convexe

Pour une lecture plus aisée de ce chapitre, nous rappelons quelques notions d'analyse convexe. Pour plus de détails, on pourra se référer aux ouvrages [Bre83] et [HUL93]. Notons $\langle ., . \rangle$ le produit scalaire sur E et $\|.\|$ sa norme induite.

Définition 2.1. *Soit $f : E \to \mathbb{R}$ une fonction différentiable.*

— *f est convexe si :*

$$\forall \lambda \in [0,1], \forall x, y \in E, \quad f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y).$$

— *f est μ-fortement convexe si :*

$$\forall x, y \in E, \quad f(y) \geq f(x) + {}^t\nabla f(x)(y-x) + \frac{\mu}{2} \|y-x\|^2,$$

où ∇f désigne le gradient de la fonction f.

Les notions de fonctions convexes et fortement convexes sont particulièrement importantes en théorie de l'optimisation puisqu'elles permettent d'affiner les résultats de convergence des algorithmes d'optimisation (voir la Section 2.3). Il en va de même des fonctions Lipschitz différentiables, définies par :

Définition 2.2. *Soit $f : E \to \mathbb{R}$ une fonction différentiable. f est L-Lipschitz différentiable si :*

$$\forall x, y \in E, \quad |\nabla f(x) - \nabla f(y)| \leq L |x-y|,$$

où $L > 0$ est la constante de Lipschitz différentiabilité.

2.1.2 Optimisation sans contrainte

Les problèmes d'optimisation sans contrainte sont définis par un problème d'optimisation (I.8), pour lequel $E = F$. Le Théorème 2.1 suivant donne une condition suffisante pour qu'un point x soit une solution du problème de minimisation (I.8).

Théorème 2.1 (Condition d'optimalité sans contrainte)**.** *Soit $x \in E$ satisfaisant les deux conditions suivantes :*

(i) $\nabla f(x) = 0$,

(ii) la Hessienne $H(x)$ de f au point x est symétrique définie positive.

Alors x est un minimum local de f.

Remarquons que la condition (ii), ou condition du second ordre, revient à dire que f est localement convexe en x. En pratique, elle peut être difficile à vérifier. Dans le cas particulier où f est convexe, une condition suffisante d'optimalité pour le point x est $\nabla f(x) = 0$. En outre, si la fonction f est convexe, tout minimum (ou maximum) local de f est alors un minimum (ou maximum) global de f. Les points critiques (dont le gradient est nul) de la fonction f définissent donc ainsi des minima (ou maxima) globaux de f. Ce résultat rend particulièrement attrayante l'optimisation d'un critère convexe.

2.1.3 Optimisation sous contraintes

Les problèmes d'optimisation sous contraintes (I.8), pour lesquels l'ensemble admissible E est un sous-ensemble non vide de F, défini par des contraintes d'égalité ou d'inégalités de fonctions :

$$E = \{x \in F, h_i(x) = 0, i = 1, ..., p, g_j(x) \leq 0, j = 1, ..., q\},$$

où les fonctions $h : F \to \mathbb{R}^p$ et $g : F \to \mathbb{R}^q$ sont continues, constituent une classe de problèmes plus difficilement résolvables.

Les conditions d'optimalité dans le cadre de l'optimisation sous contraintes sont de la même forme que la condition (i) du premier ordre présentée dans le paragraphe précédent. Dans le cas où il y a plusieurs contraintes d'égalité ou d'inégalité, un minimum local de la fonction f est un point satisfaisant les conditions KKT (Karush, Kuhn, Tucker). Pour plus de détails sur ces conditions, on pourra se référer à [KT51].

2.2 Méthodes de descente pour la résolution de problèmes d'optimisation

Dans ce paragraphe, nous décrivons les méthodes d'optimisation dites de descente, utilisées pour résoudre des problèmes de minimisation avec ou sans contraintes. Supposons dans un premier temps qu'il n'y a pas de contraintes.

Le principe de base des méthodes de descente est le suivant : générer une suite de points $(x_k)_{k\geq 0}$ appartenant à E tels que :

$$f(x^{k+1}) \leq f(x^k), \quad \text{pour tout } k \geq 0,$$

où x^0 est un point choisi arbitrairement. Ce schéma de convergence permet notamment d'assurer la convergence de la suite $f(x^k)$ (si $f(x)$ est borné) et l'amélioration de la fonction objectif.

2.2.1 Méthodes de descente du gradient

Le plus connu des algorithmes associés à ces méthodes est l'algorithme de descente du gradient, obtenu en remplaçant f par son développement de Taylor du premier ordre au voisinage de x^k :

$$\begin{aligned} &x^0 \in E, \\ &x^{k+1} = x^k - \gamma \nabla f(x^k), \end{aligned} \tag{I.9}$$

où γ est le pas de descente.

L'algorithme de descente du gradient peut être légèrement modifié par un choix (local) optimal de pas de descente γ, à chacune des itérations de l'algorithme. L'algorithme de descente à pas optimal est ainsi défini de la manière suivante :

$$\begin{aligned} &x^0 \in E, \\ &x^{k+1} = x^k - \gamma_k \nabla f(x^k), \quad \text{où} \quad \gamma_k = \underset{\gamma>0}{\operatorname{argmin}} \left\{ f\left(x^k - \gamma \nabla f(x^k)\right) \right\}. \end{aligned} \tag{I.10}$$

Remarquons que les algorithmes de descente du gradient peuvent conduire à une approximation d'un minimum local de la fonction objectif mais peuvent requérir de nombreuses itérations pour trouver ce minimum à précision donnée. Un exemple d'applications des algorithmes de descente du gradient pour la minimisation de la fonction

$$f : (x, y) \in \mathbb{R}^2 \mapsto \frac{1}{2}x^2 + \frac{7}{2}y^2,$$

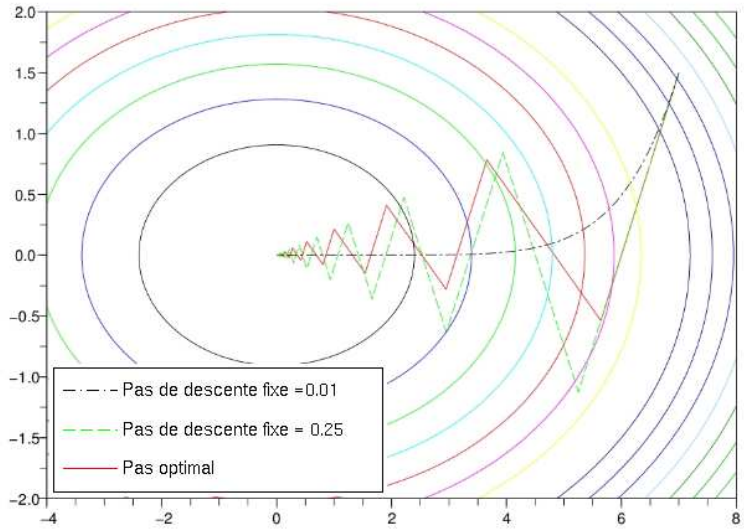


FIGURE I.3 – Itérations des algorithmes de descente de gradient à pas fixe $\gamma = 0.01$ et 0.25 et à pas optimal, pour la minimisation de la fonction $f(x, y) = 1/2x^2 + 7/2y^2$. Le point de départ est $(7.5, 1.5)$. Les algorithmes sont stoppés au bout de 1340 itérations, *resp.* 49 et 43 iterations, pour $\gamma = 0.01$, *resp.* $\gamma = 0.25$ et γ optimal.

est donnée par la Figure I.3.

Cette méthode a l'avantage d'être facile à implémenter et de posséder des garanties de convergence sous réserve notamment de conditions sur la structure de la fonction objectif f (voir la Section 2.3). Elle souffre en revanche d'un inconvénient de faible convergence dans le cas de problèmes mal conditionnés. Ces problèmes se manifestent par des surfaces d'erreur ressemblant à de longs ravins : la pente est forte dans certaines directions mais faible dans d'autres. Dans ce cas, le gradient ne pointe pas vers le minimum mais plutôt dans la direction du meilleur gain immédiat. Si le pas effectué dans cette direction est trop grand, l'optimum va osciller entre les deux côtés du ravin et réaliser peu de progrès dans les autres directions. Un moyen d'améliorer ces faiblesses des algorithmes de descente de gradient consiste à utiliser la méthode de Newton.

2.2.2 Méthode de Newton

Pour construire les méthodes de descente de gradient, nous avons remplacé f par son approximation linéaire au voisinage de l'itérée courante. Ces méthodes ne sont pas très performantes puisqu'elles ne tiennent pas compte de la courbure de la fonction qui est une information de second ordre. Afin d'améliorer les résultats des algorithmes de descente du gradient, la méthode de Newton consiste à remplacer la fonction f par son développement de Taylor de second ordre :

$$\begin{aligned} &x^0 \in E \\ &x^{k+1} = x^k - H(x^k)^{-1}\nabla f(x^k), \end{aligned} \tag{I.11}$$

où l'on suppose que la Hessienne $H(x^k)$ de f en x^k est définie positive. En pratique, cette méthode ne doit pas être appliquée en utilisant une inversion de la matrice Hessienne mais plutôt en utilisant :

$$x^{k+1} = x^k + d_k,$$

où d_k est l'unique solution du système linéaire :

$$H(x^k)d_k = -\nabla f(x^k).$$

d_k est alors appelé la direction de Newton.

Comme nous le verrons dans la Section 2.3, la méthode de Newton améliore considérablement la vitesse de convergence, notamment dans le cas où la Hessienne est définie positive. Cependant, le calcul de cette matrice Hessienne requiert un coût de calcul particulièrement élevé. De plus, dans le cas où celle-ci n'est pas définie positive, la méthode de Newton n'est pas régulière et peut faire des sauts incontrôlés.

2.2.3 Cas contraint

Afin de résoudre un problème d'optimisation sous contraintes, l'idée est de se ramener à la résolution d'un problème d'optimisation sans contraintes en introduisant le Lagrangien qui lui est associé, défini comme suit :

$$\begin{array}{rccll} L: & F \times \mathbb{R}^p \times \mathbb{R}^q & \longrightarrow & \mathbb{R} & \\ & (x, \lambda, \mu) & \longrightarrow & L(x, \lambda, \mu) & = f(x) + \displaystyle\sum_{i=1}^{p} \lambda_i h_i(x) + \sum_{j=1}^{q} \mu_j g_j(x) \\ & & & & = f(x) + {}^t\lambda h(x) + {}^t\mu g(x), \end{array}$$

où $\lambda = {}^t(\lambda_1, ..., \lambda_p)$ et $\mu = {}^t(\mu_1, ..., \mu_q)$.

Trouver une solution au problème primal (I.8), consiste alors à trouver un point optimal x^* et un multiplicateur de Lagrange (λ^*, μ^*) au problème dual, ou problème relaxé :

$$\min_{x,\lambda,\mu} L(x, \lambda, \mu).$$

Remarquons que trouver une solution au problème dual est équivalent à trouver les points-selles du Lagrangien (pour plus de détails, voir [Bre83]). Un exemple d'applications de relaxation de problèmes d'optimisation sous contraintes est l'estimateur *Lasso*. Celui-ci est défini par :

$$\begin{array}{rcl} \hat{\theta}_{Lasso} & = & \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \ \dfrac{1}{n} \left\| Y - X\theta \right\|_2^2 . \\ & & \text{tel que} \quad \left\| \theta \right\|_1 \leq t, \end{array} \tag{I.12}$$

mais plus généralement calculé sous la forme relaxée :

$$\hat{\theta}_{Lasso} = \underset{\theta \in \mathbb{R}^p}{\operatorname{argmin}} \ \left\{ \frac{1}{n} \left\| Y - X\theta \right\|_2 + \lambda \left\| \theta \right\|_1 \right\}. \tag{I.13}$$

Les problèmes (I.12) et (I.13) sont équivalents : pour tout $\lambda \geq 0$, il existe $t \geq 0$ tel que les solutions aux problèmes (I.12) et (I.13) coïncident et inversement. Pour plus de détails, on pourra consulter [OPT99].

Dans le cas où l'ensemble des contraintes E est convexe, les algorithmes de descente de gradient projeté permettent de trouver une solution au problème d'optimisation (I.8) en ajoutant à l'algorithme de descente de gradient une étape de projection (voir la Figure I.4 suivante). A chaque étape k de l'algorithme, pour s'assurer que le point courant x^{k+1}, satisfaisant l'Equation (I.10), appartient à E, on projette x^{k+1} sur E :

$$\begin{array}{l} x^0 \in E, \\ x^{k+1} = \operatorname{Proj}_E \left(x^k - \gamma_k \nabla f(x^k) \right), \quad \text{où} \quad \gamma_k = \underset{\gamma > 0}{\operatorname{argmin}} \left\{ f\left(x^k - \gamma \nabla f(x^k) \right) \right\}. \end{array} \tag{I.14}$$

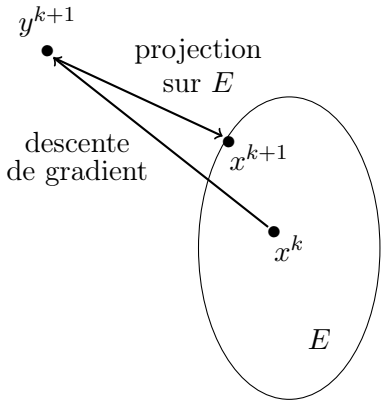


FIGURE I.4 – Illustration de l'algorithme de descente de gradient projeté.

2.3 Convergence, vitesse de convergence et complexité

Etudier la convergence d'un algorithme itératif, c'est étudier la convergence de la suite d'itérées générées par cet algorithme. Notons x^* un point limite, solution du problème de minimisation :

$$\min_{x \in E \subset F} f(x).$$

Supposons que l'on choisisse comme test d'arrêt de l'algorithme de descente le critère optimal $x^k = x^*$. Dans un monde idéal où tous les calculs sont exacts et la capacité de calcul illimitée, soit l'algorithme s'arrête après un nombre fini d'itérations, soit il construit une suite infinie de points x^1,...,x^k,... qui converge vers x^*. En pratique, un test d'arrêt devra être choisi pour que l'algorithme s'arrête toujours après un nombre fini d'itérations et que le dernier point soit suffisamment proche de x^*. Soit $\epsilon > 0$ la précision demandée, plusieurs critères d'arrêt existent dans la littérature, notamment la stagnation de la solution $\left\|x^{k+1} - x^*\right\| \leq \epsilon \left\|x^k\right\|$, mais aussi la stagnation de la valeur courante $\left|f(x^{k+1}) - f(x^k)\right| < \epsilon \left|f(x^k)\right|$, correspondant chacun à un type de convergence différent.

Il est bien entendu très important de garantir la convergence d'un algorithme sous certaines hypothèses mais la vitesse de convergence et la complexité sont également des facteurs à prendre en compte lors de l'utilisation d'un algorithme. En effet, ils garantissent un équilibre entre la précision, la stabilité et la vitesse de cet algorithme.

2.3.1 Vitesse de convergence

Il existe différents types de vitesse de convergence :

— linéaire :

$$\lim_{k \to +\infty} \frac{\left\|x^{k+1} - x^*\right\|}{\left\|x^k - x^*\right\|} = \tau,$$

où $\tau > 0$,

— superlinéaire :

$$\lim_{k \to +\infty} \frac{\left\|x^{k+1} - x^*\right\|}{\left\|x^k - x^*\right\|} = 0,$$

— d'ordre p (ou quadratique lorsque $p = 2$) :

$$\lim_{k \to +\infty} \frac{\left\|x^{k+1} - x^*\right\|}{\left\|x^k - x^*\right\|^p} = \tau.$$

Bien entendu, plus la vitesse de convergence de l'algorithme est grande, meilleures sont la convergence et la précision de l'algorithme.

Sans hypothèse sur la structure de f, il est difficile d'établir un résultat sur la convergence de l'algorithme de descente de gradient. Les premiers résultats de convergence ont été établis par Nesterov [Nes04] au prix des hypothèses suivantes :

(i) la dérivée de f est M-Lipschitz différentiable (condition du second ordre),

(ii) il existe un minimum local x^* de f tel que la Hessienne $H(x^*)$ de f en x^* est définie positive et satisfait :
$$lI_n \leq H(x^*) \leq LI_n,$$
où I_n est la matrice identité d'ordre n (dimension de l'espace E), ℓ, $L \in \mathbb{R}_+^*$,

(iii) le point de départ x^0 de l'algorithme est suffisamment proche de x^* :
$$\left\|x^0 - x^*\right\| \leq r_0 := \frac{2l}{M}.$$

Sous ces hypothèses, Nesterov [Nes04] obtient une vitesse de convergence linéaire pour l'algorithme de descente du gradient, donnée par le Théorème 2.2.

Théorème 2.2 ([Nes04])**.** *Sous les hypothèses (i), (ii) et (iii), la vitesse de convergence de l'algorithme de descente du gradient est linéaire :*
$$\forall k \in \mathbb{N}, \quad \left\|x^k - x^*\right\| \leq C\left(1 - \frac{l}{L+l}\right)^k,$$
où $C > 0$ est une constante dépendant de $\left\|x^0 - x^*\right\|$, l et M.

Remarquons que sous les mêmes hypothèses, un résultat similaire (voir Théorème 2.3 suivant) a été démontré par Nesterov [Nes04] pour la méthode de Newton.

Théorème 2.3 ([Nes04])**.** *Supposons que les hypothèses (i), (ii) et (iii) soient satisfaites avec $r_0 := \frac{2l}{3M}$. La vitesse de convergence de l'algorithme de Newton est alors quadratique :*
$$\forall k \in \mathbb{N}, \quad \left\|x^{k+1} - x^*\right\| \leq \frac{M\left\|x^k - x^*\right\|^2}{2\left(l - M\left\|x^k - x^*\right\|\right)}.$$

Notons que l'hypothèse (ii) sur la structure de la Hessienne n'est pas vérifiable en pratique et les hypothèses (i) et (iii) sont très restrictives. Cependant, on peut remplacer ces hypothèses en se restreignant à l'étude des fonctions objectifs convexes (voire fortement convexes), ce qui garantit notamment la convergence vers un minimum global (et non local !) de l'algorithme.

2.3.2 Complexité

La complexité d'un algorithme fait référence à la notion de temps de calcul nécessaire pour que l'algorithme fournisse une solution au problème d'optimisation, indépendamment de la machine utilisée ou du langage de programmation employé. Plus précisément, on définit la fonction de complexité $C(n)$ d'un algorithme comme le nombre maximal d'opérations élémentaires nécessaires au calcul pour une entrée de taille n. La complexité d'un algorithme est alors définie comme étant l'ordre de sa fonction de complexité. Elle permet de donner une idée du comportement asymptotique de cette dernière vers $+\infty$. L'exécution d'algorithmes d'ordre $\mathcal{O}(1)$ (ordre de grandeur constant) est ainsi par exemple indépendante du nombre de variables, tandis que

l'exécution d'un algorithme d'ordre $\mathcal{O}(n)$ (ordre de grandeur linéaire) dépend linéairement du nombre de variables.

L'étude de la complexité des algorithmes en optimisation, notamment par Papadimitriou [Pap94], a permis de répartir ces problèmes en différentes classes. Parmi les plus connus, on peut citer la classe des problèmes P, pour lesquels une solution peut être déterminée par un algorithme de complexité au plus polynomiale. Les problèmes de type NP correspondent à des problèmes pour lesquels si une solution possible est donnée, on peut vérifier cette solution en un temps polynomial. Ce sont des problèmes pour lesquels il existe un algorithme efficace. Les problèmes de type NP constituent une classe de problèmes difficilement résolubles.

Le choix d'un algorithme pour résoudre un problème d'optimisation nécessite un juste équilibre entre convergence de l'algorithme vers un point, si possible un minimum de la fonction objectif, et complexité de l'algorithme pour limiter le nombre d'opérations nécessaires au calcul de cette solution. Typiquement, la méthode de Newton assure une convergence plus rapide que l'algorithme de descente du gradient mais sa complexité est de l'ordre de $\mathcal{O}(n^3 + n^2 k)$, où n correspond à la dimension de l'espace de recherche E et k est le nombre d'itérations de l'algorithme, ce qui correspond à la formation et l'inversion de la matrice Hessienne. Lorsque la dimension de l'espace n devient trop importante, la complexité de l'algorithme explose et il devient inutilisable.

2.3.3 Notion d'oracle en optimisation

Pour obtenir l'itérée suivante, l'algorithme a besoin d'informations sur la fonction objectif f, notamment la valeur numérique de f en un point donné x et la valeur du gradient $\nabla f(x)$. Ces informations sont fournies par une boîte noire, *i.e.* par un sous-programme indépendant de l'algorithme d'optimisation choisi. Par analogie avec la complexité d'un algorithme, la complexité oracle correspond au nombre de fois où l'on fait appel à l'oracle pour obtenir une solution avec précision ϵ.

Remarquons que les résultats concernant la complexité des algorithmes dépendent essentiellement des hypothèses effectuées sur la structure de f (convexité, forte convexité, Lipschitz différentiabilité,...). Supposons que f est convexe et que l'ensemble E est inclus dans une boule euclidienne de rayon R. L'ensemble des résultats présentés dans la suite de ce paragraphe sont issus de [Nes04].

Dans un premier temps, on suppose que le gradient de f est uniformément borné par $L > 0$ sur E :

$$\forall x \in E, \quad |\nabla f(x)| \leq L.$$

Remarquons qu'une condition nécessaire pour cette hypothèse est que f soit L-Lipschitz sur E. L'algorithme de descente de gradient projeté satisfait alors le Théorème 2.4 suivant :

Théorème 2.4. *Supposons que f est L-Lipschitz. L'algorithme de descente de gradient projeté (I.14) admettant pour pas de descente $\gamma_k = \frac{R}{L\sqrt{k}}$. satisfait alors :*

$$\forall k \geq 0, \quad f\left(\frac{1}{k}\sum_{\ell=0}^{k-1} x^\ell\right) - f(x^*) \leq \frac{RL}{\sqrt{k}}.$$

Remarquons que le Théorème 2.4 concerne une moyenne des points construits par l'algorithme de descente de gradient projeté. Une conséquence du Théorème 2.4 est que la complexité de l'algorithme est indépendante de la dimension de l'espace, ce qui rend l'utilisation de cet algorithme particulièrement attrayante dans le cadre de l'optimisation en grande dimension. Cependant, afin d'atteindre une solution avec précision ϵ, l'algorithme requiert $\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$ appels à l'oracle, ce

qui n'est pas techniquement envisageable. Des hypothèses plus restrictives sur f peuvent fort heureusement améliorer la valeur de la complexité.

Dans un second temps, supposons donc que f est L-Lipschitz différentiable. Dans le cas non contraint ($E = F$), le Théorème 2.4 devient :

Théorème 2.5. *Supposons que f est L-Lipschitz différentiable. L'algorithme de descente de gradient (I.10) admettant pour pas de descente $\gamma_k = \frac{1}{L}$. satisfait :*

$$\forall k \geq 0, \quad f(x^k) - f(x^*) \leq \frac{2L \left\|x^0 - x^*\right\|^2}{k-1}.$$

Ce résultat provient de l'étude de la décroissance de la fonction objectif au cours des itérations de l'algorithme de descente de gradient, de l'ordre de $\frac{1}{2L} |\nabla f(x)|^2$. Dans le cas contraint, ce taux de décroissance est modifié par l'étape de projection et on montre alors que :

$$\forall k \geq 0, \quad f(x^k) - f(x^*) \leq \frac{3L \left\|x^0 - x^*\right\|^2 + f(x^0) - f(x^*)}{k}.$$

Pour obtenir une solution avec précision $\epsilon > 0$, l'algorithme de descente de gradient projeté ne requiert ainsi plus que $\mathcal{O}\left(\frac{LR^2}{\epsilon}\right)$ itérations.

Un cas particulier concerne les fonctions μ-fortement convexes et dont le gradient est uniformément borné par L, pour lesquelles Lacoste *et al.* [LJSB02] ont montré :

Théorème 2.6. *Supposons que f est L-Lipschitz et μ-fortement convexe. L'algorithme de descente de gradient projeté (I.14) admettant pour pas de descente $\gamma_k = \frac{R}{L\sqrt{k}}$. satisfait alors :*

$$\forall k \geq 0, \quad f\left(\sum_{\ell=0}^{k-1} \frac{2\ell}{k(k-1)} x^\ell\right) - f(x^*) \leq \frac{2L^2}{\mu k}.$$

Comme précédemment, le nombre d'itérations requis est de l'ordre de $\mathcal{O}\left(\frac{1}{\mu\epsilon}\right)$. On peut alors espérer améliorer grandement ces bornes de convergence en combinant les hypothèses f est μ-fortement convexe et f est L-Lipschitz différentiable. C'est l'objet du Théorème 2.7 suivant :

Théorème 2.7. *Supposons que f est μ-fortement convexe et L-Lipschitz différentiable. L'algorithme de descente de gradient projeté (I.14) admettant pour pas de descente $\gamma_k = \frac{2}{L+\mu}$. satisfait :*

$$\forall k \geq 0, \quad f(x^k) - f(x^*) \leq \frac{L}{2} \exp\left(-\frac{4(k-1)}{\frac{L}{\mu}+1}\right) \left\|x^0 - x^*\right\|^2.$$

Le Théorème 2.7 implique que la complexité oracle de l'algorithme de descente de gradient projeté est de l'ordre de $\mathcal{O}\left(\frac{L}{\mu} \log \frac{1}{\epsilon}\right)$. Une conséquence immédiate de ce résultat est que les fonctions μ-fortement convexes et L-Lipschitz différentiables peuvent être optimisées en très grande dimension avec une très bonne précision.

Le Tableau I.1 résume les différents taux de convergence obtenus sous les différentes hypothèses effectuées sur la structure de f.

Remarquons que les différents résultats obtenus donnent des bornes supérieures des taux de convergence de l'algorithme de descente du gradient projeté. Les travaux de Nemirovsky et Yudin [NY83] et Nesterov [Nes04] autour des procédures dites de boîtes noires ont permis d'obtenir des bornes inférieures pour la complexité oracle. Elles assurent notamment l'existence d'une fonction f pour laquelle les taux de convergence présentés dans le Tableau I.1 sont optimaux.

f	taux de convergence	nombre d'itérations
L-Lipschitz	$RL/\sqrt{t}$	R^2L^2/ϵ^2
L-Lipschitz différentiable	R^2L/t	R^2L/ϵ
μ-fortement convexe L-Lipschitz	$L^2/(\mu t)$	$L^2/(\mu\epsilon)$
μ-fortement convexe L-Lipschitz différentiable	$LR^2\exp(-t\mu/L)$	$L/\mu\log\left(LR^2/\epsilon\right)$

TABLE I.1 – Taux de convergence et nombre d'itérations nécessaires à l'obtention d'une solution avec précision ϵ pour l'algorithme de descente de gradient projeté.

2.4 Approximation gloutonne pour l'optimisation convexe

Pour résoudre un certain nombre de problèmes physiques ou mathématiques, on peut aussi être amené à trouver une solution à un problème d'optimisation sous la forme d'une combinaison linéaire parcimonieuse d'une famille d'éléments. Les méthodes d'approximation gloutonnes, étudiées notamment par Donoho *et al.* [DET07] et Temlyakov [Tem00], peuvent être adaptées au cadre de l'optimisation pour trouver une solution parcimonieuse à un problème d'optimisation donné [Temv1].

2.4.1 Approximation gloutonne

Soit F un espace de Banach, muni de la norme $\|.\|$ et du produit scalaire $\langle .,. \rangle$. Soit $\mathcal{D} := (g_1, ..., g_p)$ une famille d'éléments de F satisfaisant les conditions suivantes :

$$\forall j \in [\![1, p]\!], \quad \|g_j\| = 1 \quad \text{et} \quad \text{Span}(g_1, ..., g_p) = F,$$

où $\text{Span}(g_1, ..., g_p)$ désigne l'espace vectoriel engendré par la famille $(g_1, ..., g_p)$. Les problèmes d'approximation d'une fonction $f \in F$ par une combinaison linéaire des éléments de $\mathcal{D}$ ont été largement étudiés dans la littérature (pour plus de détails, se référer à [DeV98]). Les algorithmes gloutons sont des méthodes itératives visant à construire une approximation de f à l'aide d'une succession d'approximations locales. Le principe de base de ces méthodes consiste à chercher à chaque étape le meilleur élément à ajouter à l'approximation de f.

Notons $G_k(f)$ l'approximation de f à l'étape k et $R_k(f) := f - G_k(f)$ le résidu qui lui est associé. A l'étape k, l'algorithme Pure Greedy Algorithm (PGA) [FS81] et plus connu sous le nom de *Matching Pursuit*, ajoute au modèle l'élément $\varphi_k \in \mathcal{D}$, tel que :

$$\varphi_k = \underset{g \in \mathcal{D}}{\text{argmax}} \, |\langle g, R_{k-1}(f) \rangle| \,, \tag{I.15}$$

où $G_0(f) = 0$ et $R_0(f) = f$. L'approximation courante est alors définie par :

$$G_k(f) = G_{k-1}(f) + \langle R_{k-1}(f), \varphi_k \rangle \varphi_k.$$

Un autre exemple d'algorithme de type Greedy est l'Orthogonal Greedy Algorithm (OGA) [MZ93], ou *Orthogonal Matching Pursuit*. Etant donné l'élément $\varphi_k \in \mathcal{D}$ satisfaisant l'Equation (I.15), l'approximation courante est actualisée de la manière suivante :

$$G_k(f) = \mathrm{Proj}_k(f),$$

où $\mathrm{Proj}_k(f)$ est la projection orthogonale de f sur l'espace engendré par la famille d'éléments $(\varphi_1, ..., \varphi_k)$. De nombreux raffinements, améliorant la convergence globale de ces algorithmes, ont été apportées au cours des années. L'étude de ces algorithmes constitue la première partie de cette thèse, nous ne détaillerons donc pas davantage cette partie.

2.4.2 Algorithmes gloutons pour l'optimisation convexe

En théorie de l'optimisation, on cherche cette fois une solution parcimonieuse à un problème d'optimisation donné du type (I.8), où f est différentiable et E est un espace de Banach muni de la norme $\|.\|$ et du produit scalaire $\langle ., . \rangle$. Les algorithmes de type Greedy peuvent être relativement facilement aménagés pour résoudre ces problèmes d'optimisation.

Comme pour les problèmes d'approximation, la première difficulté consiste à trouver à chaque étape de l'algorithme le meilleur élément $\varphi_k \in \mathcal{D}$ à ajouter au modèle. Temlyakov [Temv1] propose pour cela deux types de stratégie :

— la stratégie "gradient-greedy" pour laquelle φ_k satisfait :

$$\langle -\nabla f(G_{k-1}), \varphi_k \rangle = \sup_{g \in \mathcal{D}} \langle -\nabla f(G_{k-1}), g \rangle,$$

où G_{k-1} est l'approximation courante de la solution au problème d'optimisation (I.8), initialisée à 0,

— la stratégie "f-greedy", pour laquelle φ_k satisfait :

$$\inf_{\gamma \in \mathbb{R}} f(G_{k-1} + \gamma \varphi_k) = \inf_{g \in \mathcal{D}, \gamma \in \mathbb{R}} f(G_{k-1} + \gamma g).$$

Dans un second temps, on définit un équivalent du pas de descente des algorithmes de descente dans la direction φ_k choisie. Un exemple de pas γ_k est :

$$f(G_{k-1} + \gamma_k \varphi_k) = \inf_{\gamma \in \mathbb{R}} f(G_{k-1} + \gamma \varphi_k).$$

La solution courante G_k est alors actualisée suivant l'équation :

$$G_k = G_{k-1} + \gamma_k \varphi_k.$$

Il existe de nombreuses variantes de pas permettant d'améliorer les performances globales de ces algorithmes (pour plus de détails, se référer à [Temv1]), mais cela ne constitue pas un réél intérêt pour cette thèse, nous ne poursuivrons donc pas plus cette description.

3 L'apprentissage de réseaux de régulation de gènes

En biologie, l'expression d'un gène donné est modulée par une cascade de régulations présentes à différentes étapes. Ces régulations peuvent avoir des effets positifs sur l'expression de ce gène, on parle alors d'activation, ou négatifs, on parle alors d'inhibition de l'expression de ce gène. Ces différentes régulations impliquent à la fois des protéines, des ARNs ou la séquence d'ADN elle-même. On représente généralement un réseau de régulation par un graphe où chaque nœud

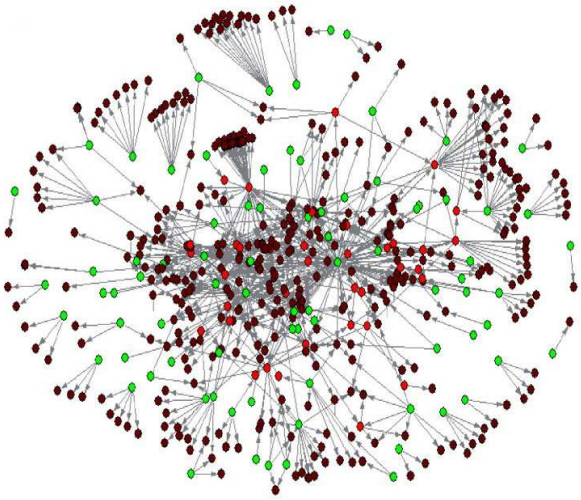


FIGURE I.5 – Un exemple de réseau de régulation de gènes chez la bactérie E. coli.

symbolise une entité biologique (gène, ARN, protéine) et les arcs entre les nœuds, les régulations. On parle alors de réseau de régulation biologique, dont un exemple est représenté Figure I.5.

La modélisation par réseau permet de mieux appréhender le fonctionnement d'un organisme. Dans cette thèse, nous nous sommes "limités" à des réseaux de *génomique génétique* pour lesquels les nœuds sont des marqueurs génétiques (données discrètes) et des données d'expression de gènes (données continues).

La Section 3.1 introduit les bases de biologie nécessaires à la compréhension du problème de reconstruction d'un réseau de régulation de gènes, au départ de l'ADN jusqu'au phénomène de régulation. La Section 3.2 concerne la modélisation mathématique des réseaux de régulation et les techniques d'inférence statistique permettant de rechercher la topologie d'un réseau. Dans la Section 3.3, nous présentons enfin les données utilisées ainsi que les différents critères d'évaluation permettant de comparer les techniques d'apprentissage.

3.1 Rappels de biologie

Le génome représente l'ensemble du matériel génétique nécessaire au développement de toute espèce vivante. L'ADN est le support de l'information génétique, transmis de génération en génération lors de la reproduction. C'est une molécule constituée de deux brins complémentaires en forme de double hélice composés d'une succession de nucléotides, comme représenté sur la Figure I.6. On trouve quatre nucléotides différents dans l'ADN, notés A, G, C et T, du nom des bases correspondantes adénine, guanine, cytosine et thymine, dont l'ordre d'enchaînement détermine l'information génétique. Ces nucléotides peuvent s'apparier : l'adénine est associée à la thymine tandis que la guanine est associée à la cytosine. L'ADN détermine la synthèse des protéines par l'intermédiaire de l'acide ribonucléique (ARN).

Par des techniques biochimiques, il est possible de déterminer la succession de ces bases sur le génome d'un organisme donné : c'est le séquençage du génome. En génétique, le séquençage permet de déterminer la séquence des gènes, voire des chromosomes, voire du génome complet. Les gènes correspondent à des portions d'ADN de taille variable et réparties le long des chromosomes.

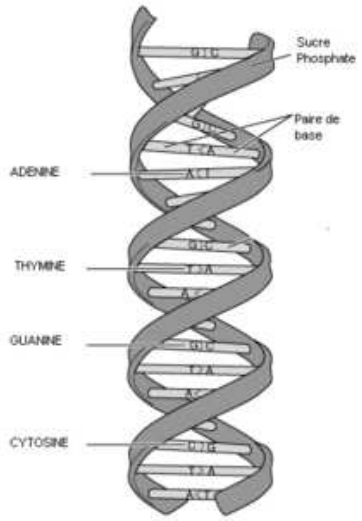


FIGURE I.6 – Structure de la molécule d'ADN.

3.1.1 Qu'est-ce qu'un marqueur ?

Dans deux génomes humains (ou autres), 99.9% de la séquence d'ADN est identique. Les 0.1% restants contiennent des variations de séquence, on parle alors de polymorphisme. Le type le plus commun de variations est le polymorphisme pour un nucléotide SNP (*Single Nucleotide Polymorphism*) pour lequel deux chromosomes diffèrent sur un segment donné par une seule paire de bases.

Un marqueur génétique est une portion d'ADN repérable spécifiquement. C'est un gène ou une portion d'ADN polymorphe qui a un emplacement connu sur le chromosome. De nos jours, les marqueurs les plus fréquemment utilisés dans les études sont les marqueurs SNP.

3.1.2 L'expression d'un gène

Le nombre de gènes varie suivant l'organisme indépendamment de la taille du génome allant de quelques centaines à plusieurs dizaines de milliers. Les gènes représentent les régions codantes de l'ADN car celles-ci codent pour la production de molécules spécifiques, les régions restantes, qui représentent une partie importante de l'ADN, sont dites non-codantes. On notera cependant qu'un gène n'est pas constitué d'une unique région codante, des régions non-codantes venant s'intercaler par endroits, un gène est donc une succession de régions codantes, les exons et de sections non-codantes, les introns. Le rôle des régions non-codantes, un temps ignoré, fait l'objet d'études de plus en plus nombreuses révélant des rôles multiples notamment dans le phénomène de régulation entre gènes.

Le rôle d'une partie des gènes est la production de protéines qui constituent les acteurs principaux de l'activité cellulaire. Si certaines protéines jouent un rôle d'enzyme au sein des cellules, ou encore le rôle de transmetteur d'information, d'autres protéines, appelées facteurs de transcription, permettent de réguler l'activité de certains gènes, comme nous le détaillerons par la suite.

La Figure I.7 représente les deux étapes principales nécessaires à la production d'une protéine à partir de l'ADN :

— l'étape de transcription de l'ADN consiste à synthétiser une molécule d'ARN, constituée

d'un seul brin qui est une copie complémentaire d'un des deux brins de la séquence d'ADN,
— elle est suivie de l'étape de traduction, au cours de laquelle l'ARN est transformé en protéine.

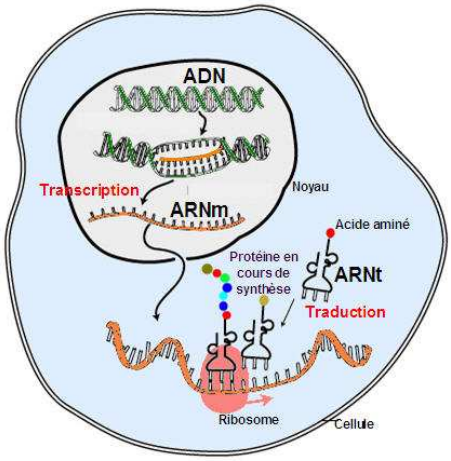


FIGURE I.7 – Les différentes étapes nécessaires à la production d'une protéine.

Il faut savoir que le niveau de protéines synthétisées à partir d'un gène n'est pas constant dans le temps et que des variations sont observées, dues notamment à des phénomènes de régulations. Avant de chercher à définir les régulations entre des gènes, il est nécessaire de préciser ce qui caractérise le niveau d'expression d'un gène. Celui-ci peut être défini de deux manières différentes : d'une part, par l'ensemble des protéines synthétisées, d'autre part, par l'ensemble des ARN transcrits à partir de ce gène, ces deux quantités n'étant pas équivalentes du fait des modifications pouvant intervenir au cours des étapes de traduction et de transcription de l'ADN.

Ces deux niveaux d'analyse n'apportent pas le même type d'informations et constituent même des approches complémentaires pour comprendre le fonctionnement global de la cellule. Cependant, afin de simplifier le problème, nous nous restreignons dans ce manuscrit à l'étude du niveau des transcrits. Dans toute la suite, l'expression d'un gène sera donc uniquement caractérisée par son niveau de transcrits.

De nos jours, les puces à ADN (voir Figure I.8) représentent la technique la plus utilisée afin de mesurer l'expression des gènes, dues principalement à leur faible coût et la possibilité de mesurer simultanément le niveau de transcrits de milliers de gènes. Concrètement, une puce est constituée d'une lame solide sur laquelle sont fixés des fragments d'ADN mono-brin correspondant aux gènes dont on souhaite mesurer l'expression. Ces fragments, les *sondes*, sont regroupés au sein de *spot*. Chaque *spot* contient plusieurs milliers de *sondes* correspondant au même fragment d'ADN et donc à un gène précis. Ces *spots* sont répartis sur le support de la puce sous la forme d'une grille. Le choix des fragments à inclure sur la puce est primordial du fait que seuls les ARNs correspondants aux *spots* présents seront mesurés.

3.2 Modélisation d'un réseau de régulation de gènes

Nous avons déjà mentionné à plusieurs reprises l'existence de régulations modulant l'expression des gènes dues à certaines protéines. Des régulations sont présentes à chaque étape nécessaires à la production d'une protéine (voir par exemple la Figure I.9). De plus, les mutations observées peuvent potentiellement induire une variabilité dans cette action. La prise en compte de

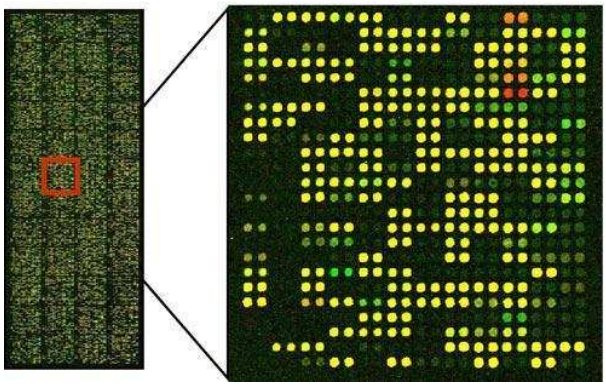


FIGURE I.8 – Exemple d'une puce à ADN, permettant de mesurer l'expression des gènes.

l'ensemble de ces régulations permet d'appréhender au mieux le fonctionnement réel d'un organisme, cependant, l'observation conjointe de l'ensemble de ses acteurs (gène, ARN et protéine) est complexe.

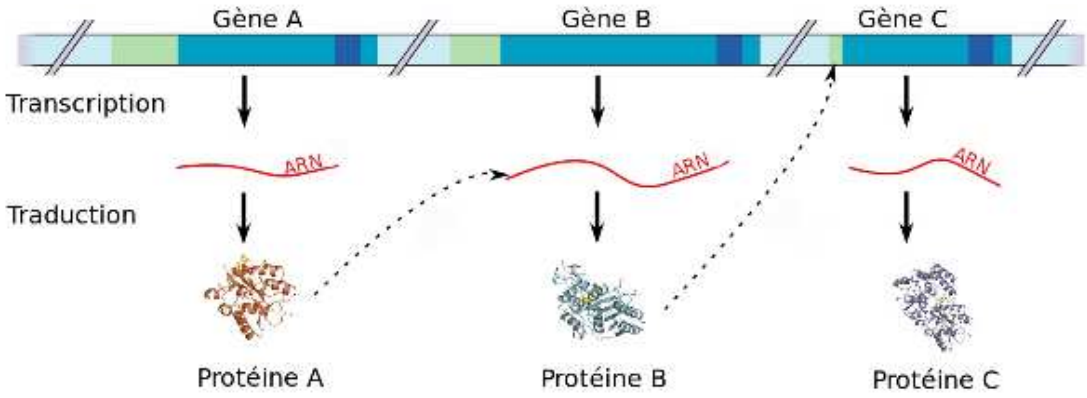


FIGURE I.9 – Exemples de régulations biologiques impliquant trois gènes ainsi que les ARN et protéines synthétisées. Deux types de régulations sont représentées : la protéine A régule la traduction du gène B, tandis que la protéine B régule la transcription du gène C.

Les régulations géniques peuvent être représentées sous forme de graphes, où chaque nœud représente une entité biologique (gène, ARN ou protéine) et les arcs entre ces nœuds symbolisent les régulations présentes. On parle alors de réseau de régulation biologique.

Le terme de réseau de régulation de gènes est un terme générique pour désigner un réseau pour lequel chaque nœud est assimilé à un gène et chaque arête représente une régulation d'un gène sur l'expression d'un autre gène. Un réseau de régulation de gènes peut donc à la fois désigner un réseau de protéines ou un réseau transcriptionnel. Dans cette thèse, nous considérons que l'expression d'un gène est caractérisée par son niveau de transcrits, et, par conséquent, les réseaux étudiés sont des réseaux transcriptionnels.

L'apprentissage de réseaux de régulation de gènes est une problématique de recherche très active ces dernières années, dont les applications biologiques possibles sont nombreuses.

L'objectif de cette section est de proposer une modélisation d'un réseau de régulation de

gènes. Une large variété de formalismes mathématiques ont été proposés dans ce but. Dans cette section, nous considérons différents modèles statistiques qui ont été choisis pour leur capacité à inférer des régulations géniques à partir de données d'expression. On peut distinguer deux types de réseau selon que l'on modélise ou pas l'évolution temporelle de la quantité de transcrits : les réseaux dynamiques et les réseaux statiques. La première partie de cette section vise à décrire les approches dynamiques des réseaux biologiques. Nous nous intéresserons par la suite un peu plus longuement à différentes méthodes statiques pour l'apprentissage de réseaux.

3.2.1 Approches dynamiques

Les réseaux dynamiques visent à caractériser le comportement dynamique des réseaux biologiques pour pouvoir prédire les effets des perturbations d'un gène sur les autres gènes. Plusieurs modèles dynamiques ont été considérés dans le cadre d'inférence de réseaux, en particulier les réseaux booléens [Tho73] et les réseaux dynamiques bayésiens ([RJFD10] et [LBD$^+$10]). Pour les réseaux booléens, par exemple, la méthode consiste à discrétiser les données d'expression temporelles : à un instant donné, un gène est soit actif ($= 1$), soit inactif ($= 0$). L'objectif est alors de trouver des règles logiques permettant de trouver l'état d'un gène à l'instant $t + 1$ à partir de l'état de ce gène et des autres gènes à l'instant t.

L'inconvénient majeur des méthodes dynamiques pour la reconstruction de réseaux de régulation de gènes est certainement qu'elles nécessitent des données cinétiques, souvent très coûteuses, pour pouvoir être mises en oeuvre.

Au contraire des approches dynamiques, les méthodes statiques ignorent les aspects temporels. Plusieurs d'entre elles ont donné des résultats prometteurs dans la reconstruction de réseaux, nous en citons certaines, parmi les plus connues.

3.2.2 Réseaux de co-expression

Plusieurs travaux dans la littérature ([dlFBHM04], [DLS00]) construisent des réseaux d'interactions par comparaison de paires de gènes. Ces méthodes diffèrent essentiellement par la mesure utilisée pour relier deux gènes.

Une première méthode d'apprentissage, développée par exemple par Carter *et al.* [CBGB04], consiste à utiliser la corrélation de Pearson (ou coefficient de corrélation linéaire) pour mesurer la relation entre chaque paire de gènes (X^i, X^j) en éliminant les interactions ayant une faible significativité. Ce coefficient de corrélation linéaire est défini par :

$$r(X^i, X^j) = \frac{\mathrm{Cov}(X^i, X^j)}{\sqrt{\mathrm{Var}(X^i)\mathrm{Var}(X^j)}},$$

où $\mathrm{Cov}(.,.)$, *resp.* $\mathrm{Var}(.)$, désigne la covariance entre deux variables, *resp.* la variance d'une variable. Il appartient à $[-1, 1]$ et permet de distinguer les régulations activatrices des régulations inhibitrices.

Pour reconstruire un réseau de régulation, on mesure alors le degré d'indépendance entre chacun des nœuds du réseau, puis on définit un seuil au-delà duquel la dépendance est jugée suffisante pour supposer l'existence d'une régulation, et donc d'une arête au sein du graphe. La Figure I.10 suivante donne un aperçu de cette méthode.

L'estimation du seuil est une étape déterminante pour la reconstruction du réseau. Le principal inconvénient de cette méthode réside dans le fait que seules les interactions entre paire de gènes sont considérées, alors qu'un gène est souvent régulé par plusieurs régulateurs. Plus précisément, elle ne permet pas de distinguer les régulations directes du type $X^j \to X^i$ des relations

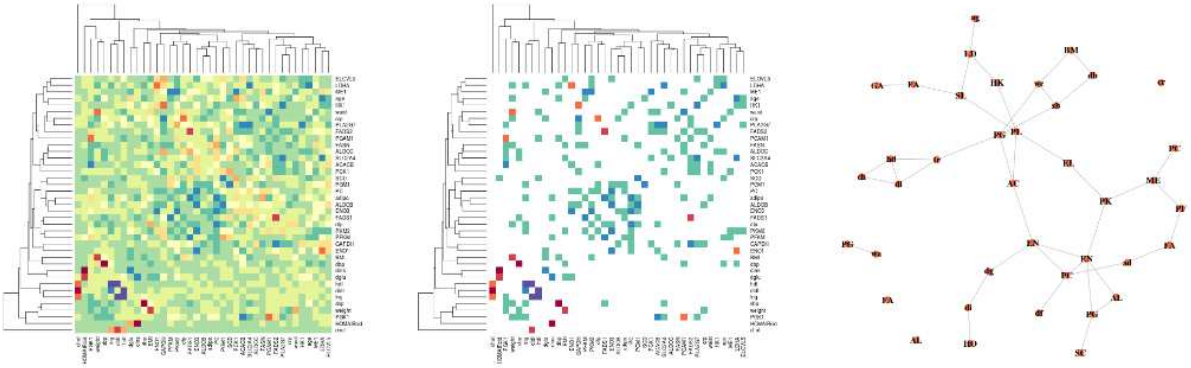


FIGURE I.10 – Première approche statique pour reconstruire un réseau : on commence par calculer les corrélations entre chacun des gènes (à gauche), on seuille (au centre) puis on reconstruit le réseau (à droite) en ne considérant que les corrélations qui sont supérieures au seuil, défini au préalable.

indirectes du type $X^j \to X^k \to X^i$, où la dépendance entre X^i et X^j est due à une troisième variable X^k.

Afin de supprimer les relations indirectes dans le réseau reconstruit, l'une des possibilités consiste à mesurer des corrélations partielles entre les variables. On définit le coefficient de corrélation partielle de Pearson d'ordre un entre deux variables X^i et X^j sachant une troisième variable X^k, aussi appelée variable de conditionnement, par :

$$r(X^i, X^j | X^k) = \frac{r(X^i, X^j) - r(X^i, X^k) r(X^j, X^k)}{\sqrt{1 - r(X^i, X^k)^2}\sqrt{1 - r(X^j, X^k)^2}}.$$

Les gènes X^i et X^j sont alors considérés liés si :

$$\forall k \neq i, j, \quad r\left(X^i, X^j | X^k\right) \neq 0,$$

ou si les corrélations partielles sont significativement importantes devant un seuil à définir. Les mesures de dépendances partielles d'ordre un permettent de détecter les situations où deux gènes ont une frontière d'indépendance d'au plus un gène dans le vrai réseau.

Afin de couvrir l'ensemble des relations indirectes possibles, il est donc nécessaire d'étendre les notions de corrélation partielles pour des ensembles de conditionnement d'ordre k supérieur à un. Leur nombre croissant de façon exponentielle avec le nombre de gènes du réseau, cela devient très vite impossible. Une alternative possible consiste à considérer directement l'ensemble de conditionnement maximal composé de tous les autres gènes du réseau : les gènes X^i et X^j sont liés si et seulement si

$$r\left(X^i, X^j | (X^k)_{k \neq i,j}\right) \neq 0. \tag{I.16}$$

Cependant, pour pouvoir estimer correctement ces coefficients de corrélation, le nombre d'observations nécessite d'être assez grand, ce qui est souvent loin d'être le cas.

Notons le cas particulier des modèles graphiques linéaires gaussiens, pour lesquels les données d'expression des gènes du réseau suivent une loi gaussienne centrée de matrice de covariance Σ. Le coefficient de corrélation partielle donné par l'Equation (I.16) s'écrit alors de manière analytique

sous la forme :

$$r\left(X^i, X^j | (X^k)_{k\neq i,j}\right) = -\frac{S_i^j}{\sqrt{S_i^i S_j^j}},$$

où $S := \Sigma^{-1}$ est l'inverse de la matrice de covariance, aussi appelée matrice de précision. Pour reconstruire le réseau, il faut donc estimer cette matrice de précision (dans le cas où la matrice de covariance du modèle est inconnue) afin d'en mesurer ces coefficients significativement non nuls.

3.2.3 Modélisation par réseau bayésien

Les réseaux bayésiens constituent la seconde grande classe de réseaux largement étudiée dans la littérature. Dans la suite de ce paragraphe, nous adopterons la formalisation des réseaux bayésiens de [Pea78]. Ce sont des modèles permettant de décrire les relations de probabilités conditionnelles entre des faits. Cette représentation repose sur un graphe orienté sans cycle, ou *Directed Acyclic Graph* en anglais (DAG), dans lequel chaque nœud possède une table de probabilités conditionnelles et où chaque arête représente une dépendance directe entre les variables reliées.

Plus formellement, un réseau bayésien $B := (\mathcal{G}, P_\mathcal{G})$ est défini par la donnée d'un DAG $\mathcal{G} = (V, A)$, où $V := \{X^1, ..., X^p\}$ est un ensemble de variables aléatoires assimilées aux nœuds du graphe, reliées par un ensemble d'arêtes orientées A, et un ensemble de probabilités conditionnelles $P_\mathcal{G} := \{P_1, ..., P_p\}$ telles que :

$$\forall i \in [\![1, p]\!], \quad P_i = \mathbb{P}\left(X^i | \operatorname{Pa}(X^i)\right),$$

où $\operatorname{Pa}(X^i)$ désigne l'ensemble des parents du nœud X^i au sein du graphe.

En s'appuyant sur les relations d'indépendances conditionnelles existant entre les variables, ces réseaux représentent alors une distribution de probabilités jointes sur V, définie par :

$$\mathbb{P}(V) = \prod_{i=1}^{p} \mathbb{P}\left(X^i | \operatorname{Pa}(X^i)\right).$$

Une première approche utilisant ce type de modèle pour l'apprentissage de réseaux de régulation à partir de données d'expression, a été réalisée par Friedman *et al.* [FLNP00b]. Dans ces travaux, chaque gène est représenté par une variable discrète et le réseau de régulation est modélisé par un graphe orienté acyclique qui suggère l'influence causale entre les gènes.

L'apprentissage de réseaux bayésiens est encore actuellement un champ de recherche très actif et consiste à trouver un réseau bayésien modélisant les données disponibles. Il existe deux grandes familles d'approche pour apprendre la structure d'un réseau bayésien : la première approche consiste à tester les indépendances conditionnelles, et à chercher une structure de réseau cohérente avec les dépendances et indépendances observées. Cette approche traite un nombre très limité de variables et est moins utilisée pour l'inférence de réseaux de régulation. La deuxième approche utilise une fonction de score, qui mesure l'adéquation des (in)dépendances codées dans le réseau avec les données. On cherche alors un réseau maximisant ce score (pour un exemple de score, on peut se référer à [HC95]).

Avec suffisamment de données, on peut montrer que l'apprentissage d'un réseau par ces méthodes converge. Cependant, la recherche d'une structure optimale est un problème NP-difficile ([Pea78] et [CHM04]). La recherche exhaustive du meilleur réseau, guidé par une fonction de score, est mathématiquement triviale : il suffit de calculer le score de tous les graphes et sélectionner celui qui a le meilleur score. Ce qui rend irréalisable une recherche exhaustive est sans aucun

doute le nombre super-exponentiel de graphes candidats. Une solution consiste donc à restreindre l'espace d'hypothèses, en limitant par exemple le nombre de parents possibles pour chaque nœud, et/ou à utiliser des techniques d'exploration heuristiques ([Fri04], [Chi02] et [VMV+12]).

3.2.4 Modélisation par arbres de décision

De nombreuses méthodes pour l'inférence de réseaux sont ainsi basées sur la création d'un score, permettant de classer les régulations au sein du réseau. Mathématiquement, cette stratégie est très intimement liée au problème de sélection de variables en statistique, présenté en détails dans la Section 1.2. Plus précisement, pour tous les gènes j du réseau, on considère le problème de régression suivant :

$$E^j = f_j(E^{-j}, M) + \varepsilon,$$

où E^j désigne le niveau d'expression du gène j, $E^{-j} = \{E^k, k \neq j\}$ est l'ensemble des niveaux d'expression des gènes régulateurs du gène j, M constitue l'ensemble des données marqueurs, et ε est un bruit blanc. Ces modèles sont plus connus sous le nom de *Structural Equation Models* (SEM).

Pour apprendre les fonctions $(f_j)_{j=1,\dots,p}$ à partir des données dont on dispose, des méthodes statistiques linéaires ou non-linéaires peuvent être utilisées. Ainsi, dans un cadre non-linéaire, les méthodes basées sur des arbres de régression permettent de reconstruire ces réseaux de régulation de gènes (pour plus de détails sur les arbres de régressions et les forêts aléatoires, se réferer à la Section 1.3). L'utilisation des arbres de régression pour la reconstruction de réseaux de régulation de gènes pour des données d'expression a été initialement proposée par Huynh [HTIWG10].

3.2.5 Modélisation par régressions linéaires pénalisées

Une manière naturelle de reconstruire un réseau de régulation de gènes consiste à considérer indépendamment les uns des autres chacun des gènes j du réseau, et à supposer que son niveau d'expression E^j s'écrit comme une fonction linéaire des niveaux d'expression de chacun des autres gènes du réseau, et de tous les marqueurs M :

$$\forall j \in [\![1, p]\!], \quad E^j = \sum_{i=1}^{p} \alpha_i M^i + \sum_{\substack{i=1 \\ i \neq j}}^{p} \beta_i E^i + \varepsilon, \tag{I.17}$$

où $\alpha := (\alpha_i)_{1 \leq i \leq p}$ est le vecteur de taille p, mesurant les effets linéaires des polymorphismes sur le niveau d'expression E^j du gène j, $\beta := (\beta_i)_{1 \leq i \leq p}$ est le vecteur de taille p correspondant aux effets des niveaux d'expression de tous les autres gènes du réseau (on supposera $\beta_j = 0$ pour éviter la régression triviale de E^j sur E^j) et ε est un terme d'erreur, supposé gaussien.

Remarquons que l'idée d'utiliser des régressions linéaires pour modéliser des réseaux de régulation de gènes provient de Meinshausen *et al.* [MB06], qui ont fait le lien entre les coefficients non nuls de la matrice de précision S et les coefficients issus de la régression d'un gène sur les autres gènes du réseau. La problématique principale qui se pose dans ce cadre d'étude concerne la grande dimension des données : le nombre de gènes p du réseau étudié est très grand devant la taille n de l'échantillon observé. L'estimateur empirique $\hat{\Sigma}_n^{-1}$ de la matrice de précision S n'est alors, en général, pas inversible, ce qui empêche l'estimation du réseau. Au contraire, la structure des données ($p >> n$) et le fait que l'on se restreint à un réseau parcimonieux nous incitent à utiliser des méthodes de sélection de variables, dont une liste non exhaustive est donnée dans la Section 1.2 de ce chapitre.

Parmi les différentes techniques permettant de résoudre ce type de modèle, Meinshausen *et al.* [MB06] utilisent la régression *Lasso*. On peut réécrire le modèle donné par l'Equation (I.17) sous la forme :

$$Y = X\theta^0 + \varepsilon,$$

où Y constitue à tour de rôle la donnée d'expression de chacun des gènes du réseau, X constitue l'ensemble des observations (données d'expression et polymorphismes des gènes) et $\theta^0 = (\alpha, \beta)$ est le paramètre d'intérêt à estimer. L'estimation du paramètre θ^0 s'effectue alors en minimisant la somme des erreurs quadratiques sous une contrainte en norme ℓ_1 :

$$\hat{\theta}_{Lasso} = \underset{\theta \in \mathbb{R}^{2p}}{\operatorname{argmin}} \|Y - X\theta\|_2^2 + \lambda \|\theta\|_1 .$$

Pour plus de détails sur l'estimateur *Lasso*, on pourra se référer à la Section 1.2.

Une deuxième méthode d'estimation est basée sur le *Dantzig Selector* de [CT07]. Celle-ci offre une alternative intéressante à l'estimation par le *Lasso*. Elle utilise une pénalité en norme ℓ_1 sur les paramètres, sous contrainte d'une borne supérieure sur la corrélation entre les variables explicatives et les résidus :

$$\hat{\theta}_{DS} = \underset{\theta \in \mathbb{R}^{2p}}{\operatorname{argmin}} \quad \|\theta\|_1 , \quad \text{sous contraintes } \left\| {}^t X(Y - X\theta) \right\|_\infty \leq \delta,$$

où $\|.\|_\infty$ est la norme infinie, définie comme la composante maximale du vecteur, et δ est un seuil à fixer. On cherche donc à sélectionner le modèle possédant le moins de variables explicatives, tout en assurant que la part des résidus restant à estimer ne dépasse par le seuil δ. Pour $\delta = 0$, la contrainte implique une corrélation nulle entre le résidu et les variables du modèle, ce qui correspond à la présence de toutes les variables dans le modèle. Au contraire, lorsque $\delta \to +\infty$, la contrainte sur les résidus se relâche totalement et les coefficients du paramètre θ sont alors mis à zéro.

Notons qu'une méta-analyse, consistant en un consensus des méthodes réseaux bayésiens, *Lasso* et *Dantzig Selector*, a été proposée par Vignes *et al.* [VVA$^+$11] pour l'apprentissage de réseaux de régulation de gènes. Les résultats obtenus pour cette méthode d'estimation ont été particulièrement bons, les classant premiers du challenge DREAM5, dont nous parlerons dans la section suivante.

3.3 Comparaison des approches

Après avoir présenté les différentes approches utilisées afin de reconstruire des réseaux de régulation de gènes, nous nous intéressons plus particulièrement à l'utilisation des méthodes basées sur de la régression linéaire pénalisée (Equation (I.17)). L'objectif de cette section est de présenter de manière un peu plus générale la forme et la manière dont sont générées nos données. Nous évoquerons alors la compétition internationale DREAM5, dont l'objectif est d'apprendre un réseau de régulation de gènes à partir de données de génomique génétique. Nous présentons ensuite la modélisation graphique que nous choisissons pour représenter de tels réseaux dans le cas des données de génomique génétique. Nous nous intéressons enfin aux différents critères de comparaison des approches existantes.

3.3.1 Description des données simulées

Etant donnée la difficulté d'obtenir et de travailler sur des données réelles, les différentes méthodes étudiées sont testées sur des jeux de données simulés. Les données de génomique génétique utilisées ont été fournies par Alberto de la Fuente et ses collègues du CRS4 Bioinformatica (Pula,

Italia) et simulées via le logiciel SysGenSIM [PSHdlF11]. Pour chaque réseau, modélisé par un graphe orienté, le logiciel simule le comportement du réseau à l'aide d'une équation différentielle ordinaire dont nous n'expliciterons pas l'expression (pour plus de détails, voir [PSHdlF11]).

Un jeu de données est défini par un échantillon composé de n individus, issus d'un croisement de type *Recombinant Inbred Lines*, (dont les individus sont génétiquement identiques), dont on mesure les niveaux d'expression de p gènes et dont on relève p marqueurs. Un jeu de données est ainsi composé de :

— une matrice E de taille $n \times p$ où le coefficient E_i^j correspond à la donnée d'expression du gène j pour l'individu i,

— une matrice M de taille $n \times p$ où le coefficient M_i^j exprime le polymorphisme du marqueur j, associé au gène j, de l'individu i. Il s'agit d'une variable binaire.

Précisons qu'il n'y a pas équivalence gènes-marqueurs, ce qui sous-entend qu'on peut très bien associer plusieurs marqueurs à un même gène. Afin de simplifier notre étude, nous nous sommes restreints au cas où on associe un unique marqueur à un gène dans cette thèse. Nous proposons ainsi une modélisation fusionnée des réseaux de régulation de gènes (voir paragraphe suivant).

Il faut noter que depuis l'année 2006, la compétition internationale DREAM [SMC07] propose chaque année différents challenges visant à une meilleure compréhension des réseaux biologiques. Ces challenges permettent en particulier de comparer les méthodes existantes autour de l'apprentissage de réseaux. En 2010, l'édition DREAM5 proposait un challenge basé sur les données fournies par le logiciel SysGenSIM.

3.3.2 Modélisation graphique d'un réseau

Nous avons vu précedemment que les niveaux d'expression des gènes d'un réseau sont à la fois régulés par les niveaux d'expression de tous les autres gènes du réseau ainsi que par des ploymorphismes correspondant à des mutations observées à l'aide de marqueurs. Les régulations apprises sont donc toujours orientées d'un marqueur M^i ou de l'expression d'un gène E^i vers l'expression d'un autre gène E^j.

Deux modélisations graphiques sont possibles pour représenter un tel réseau. La première d'entre elles distingue clairement les variables représentant l'expression des gènes de celles associées aux marqueurs. On parle alors de modèle non-fusionné, dont un exemple est donné par la Figure I.11.

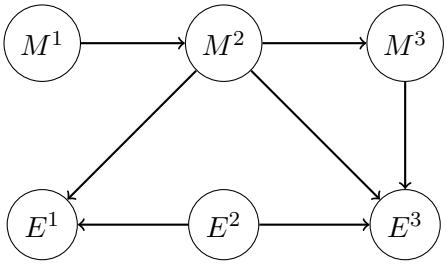


FIGURE I.11 – Exemple de réseau de régulation non-fusionné faisant intervenir trois gènes. Le niveau d'expression E^3 du gène 3 régule l'activité des gènes 1 et 2 tandis que les marqueurs M^1, M^2 et M^3, associés aux gènes 1, 2 et 3 sont liés entre eux suivant la relation $M^i \rightarrow M^{i+1}$ et viennent réguler les niveaux d'expression des gènes 1 et 3.

Le principal inconvénient de cette modélisation réside dans le nombre doublé de variables

par rapport au nombre de gènes observés. De plus, la compréhension d'un réseau de régulation non-fusionné nécessite des connaissances biologiques supplémentaires qui dépassent notre champ de compétences. D'un point de vue pratique, nous nous intéressons donc à une modélisation alternative qui rassemble au sein d'une même variable l'expression d'un gène et le marqueur qui lui est associé. De cette fusion résulte une nouvelle variable G^i dont le domaine sera le produit cartésien de celui de M^i et E^i, comme décrit par la Figure I.12.

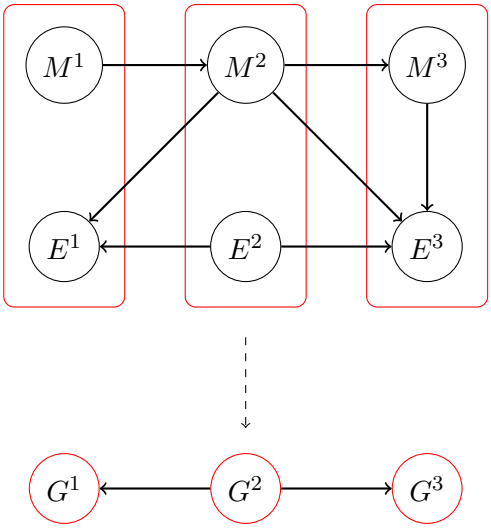


FIGURE I.12 – Exemple de fusion de réseau de régulation comportant trois gènes et trois marqueurs en un modèle ne comportant que trois variables.

3.3.3 Critères d'évaluation

Afin de comparer les performances des méthodes pour l'apprentissage de réseaux de régulation de gènes, nous utilisons des critères basés sur la structure du réseau reconstruit par rapport à la structure du vrai réseau (connu dans le cas de données simulées). Pour cela, on classe l'ensemble des arêtes possibles en quatre catégories suivant le tableau I.2 : les vrais positifs TP (*True positive*), les faux positifs FP (*False positive*), les vrais négatifs TN (*True negative*) et les faux négatifs FN (*False negative*).

		Prédiction		
		arête	pas d'arête	
Réalité	arête	TP	FN	S
	pas d'arête	FP	TN	$p - S$
		$\hat{S}$	$p - \hat{S}$	p

TABLE I.2 – Matrice de confusion permettant de trier les arêtes possibles au sein d'un graphe en quatre catégories. S désigne le nombre total d'arêtes dans le vrai graphe, tandis que $\hat{S}$ est le nombre d'arêtes prédites par l'une des méthodes d'apprentissage de réseau.

Ces quatre mesures nous permettent alors de définir les quantités suivantes :

— la précision

$$\text{Précision} = \frac{TP}{\hat{S}},$$

qui indique le nombre d'arêtes correctement prédites par rapport au nombre total d'arêtes apprises,

— la sensibilité, ou *recall* en anglais,

$$\text{Sensibilité} = \frac{TP}{S},$$

qui indique le nombre d'arêtes correctement prédites par rapport au nombre total d'arêtes à apprendre.

Remarquons que les méthodes de sélection de variables, présentées dans la Section 1.2, construisent différents modèles, du plus parcimonieux (aucune arête apprise dans le réseau) au plus complet (tous les nœuds du réseau sont reliés entre eux). La précision et la sensibilité sont alors calculées pour l'ensemble de ces modèles, conduisant au tracé de courbes précision-recall, dont on peut voir un exemple ci-dessous (voir Figure I.13). La mesure de l'aire sous la courbe AUC (*Area under curve*) peut aussi permettre de quantifier la performance d'une méthode. Pour plus de détails, on pourra se référer à [Faw06].

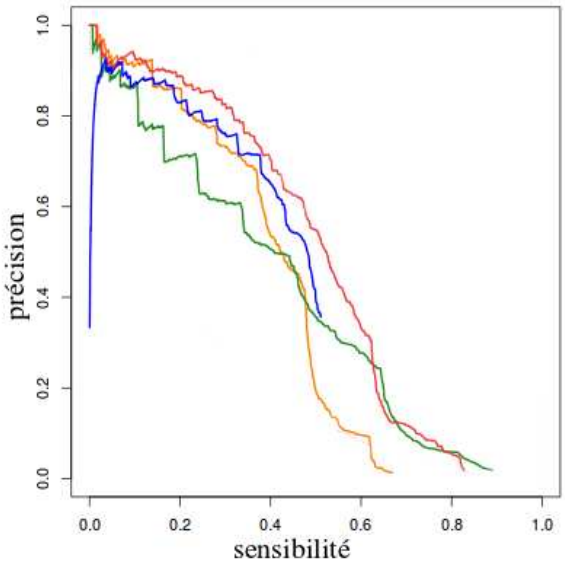


FIGURE I.13 – Courbes précision-recall pour un réseau du challenge DREAM5 pour quatre types de méthodes d'apprentissage : les réseaux bayésiens (en bleu), le *Dantzig selector* (en orange), le *Lasso* (en vert) et une méta-analyse de ces trois méthodes (en rouge) proposée par [VVA+11].

4 Contributions de cette thèse

Les travaux présentés dans cette thèse sont regroupés en trois chapitres. Le Chapitre II concerne l'étude des algorithmes de $\mathbb{L}_2$-Boosting pour la régression parcimonieuse dans le contexte de la grande dimension. Le Chapitre III propose une application de ces algorithmes à l'analyse de sensibilité. Enfin, dans le Chapitre IV, nous nous intéressons au développement de méthodes d'optimisation pour retrouver la structure de réseaux de régulation de gènes.

4.1 Les algorithmes de $\mathbb{L}_2$-Boosting

On se place dans le cadre d'un problème d'apprentissage supervisé : on observe un échantillon $(Y_i, X_i)_{1\leq i\leq n}$ de taille n, où chacune des réponses Y_i est l'image de p variables $X_i^1, ..., X_i^p$ par une fonction f perturbée par un bruit :

$$\forall i \in [\![1, p]\!], \quad Y_i = f(X_i^1, ..., X_i^p) + \varepsilon_i,$$

où la fonction f s'écrit comme une combinaison linéaire d'éléments d'un dictionnaire $\mathcal{D} = (g_1, ..., g_p)$.

Dans le Chapitre II de ce manuscrit, nous nous intéressons à l'estimation de la fonction f par l'algorithme d'approximation $\mathbb{L}_2$-Boosting. Les algorithmes de Boosting font partie de la famille des algorithmes gloutons, ils ont pour but de construire une approximation globale de f à l'aide d'une succession d'optimisations locales. En particulier, l'algorithme $\mathbb{L}_2$-Boosting [BY03] fait intervenir une fonction de perte ℓ_2 et consiste à ajouter progressivement au modèle l'élément du dictionnaire le plus corrélé avec le résidu (ce qui reste à estimer).

Les travaux de Bühlmann [Büh06] autour de ces algorithmes ont permis de montrer que cette méthode d'estimation était consistante en grande dimension. Elle se démarque ainsi de l'estimateur *Lasso* car les hypothèses effectuées ne font pas intervenir la forme du design X. En ce qui concerne le support de la fonction f, nous montrons qu'avec grande probabilité, la consistance du support est satisfaite. Ce résultat n'est possible qu'au prix d'hypothèses supplémentaires sur la structure du dictionnaire. Notons $\mathcal{S}$ le support de f, D la matrice contenant les éléments du dictionnaire et $D_{\mathcal{S}}$ la matrice restreinte aux éléments de D qui sont dans $\mathcal{S}$. Les hypothèses que nous faisons sont les suivantes :

Hypothèse 2. *La matrice $D_{\mathcal{S}}$ satisfait :*

$$\max_{j\notin\mathcal{S}} \left\| D_{\mathcal{S}}^{+} g_j \right\|_1 < 1,$$

où D^+ est la notation pour le pseudo-inverse de D lorsque D n'est pas inversible.

Hypothèse 3. *Il existe $\lambda_{min} > 0$, indépendant de n tel que :*

$$\inf_{\beta, \mathrm{Supp}(\beta)} \frac{\|D\beta\|^2}{\|\beta\|^2} \geq \lambda_{min}.$$

Hypothèse 4. *La fonction f se décompose sous la forme $f = \sum_{j\in\mathcal{S}} a_j g_j$ où :*

$$\exists \kappa \in (0,1), \forall j \in \mathcal{S}, \quad |a_j| \geq h(n)^{-\kappa},$$

où $h(n)$ est une fonction de n.

L'Hypothèse 2 est la condition d'*exact recovery* [Tro04]. Elle permet d'assurer que seuls les bons coefficients de f sont ajoutés au modèle. Les Hypothèses 3 et 4 sont nécessaires pour montrer que tous les éléments du support sont retrouvés. L'Hypothèse 3 est la condition d'isométrie restreinte [CT05]. Notons que la forme de la fonction h de l'Hypothèse 4 dépend du nombre de variables p autorisées dans le modèle. Plus précisément, dans le cadre de la grande dimension, p peut croître comme une fonction de n : plus l'hypothèse sur la croissance de p devant n est restrictive, moins la valeur des coefficients non nuls de f doit être grande, et donc moins l'Hypothèse 4 est restrictive.

La deuxième partie de ce travail a consisté à étendre l'algorithme $\mathbb{L}_2$-Boosting au cas multi-tâches, pour lequel la fonction $f = (f^1, ..., f^m)$ à prédire est susceptible d'avoir produit m réponses $Y^1, ..., Y^m$:

$$\forall i \in [\![1, m]\!], Y^i = f^i(X) + \varepsilon^i.$$

On peut tout d'abord généraliser l'algorithme précédent en choisissant d'ajouter au modèle, à chaque étape k de l'algorithme, le meilleur élément du dictionnaire associé à la meilleure réponse :

$$(i_k, \varphi_k) = \underset{i \in [\![1,m]\!], g \in \mathcal{D}}{\operatorname{argmax}} \langle g, Y - G_{k-1}(f^i) \rangle,$$

où $G_{k-1}(f^i)$ est l'approximation de f^i à l'étape $k-1$. Lutz *et al.* [LB06] montrent que cette méthode est consistante en étendant les résultats obtenus par [Büh06] dans le cas univarié. Cependant, nous pensons que ce type de choix n'est pas optimal pour certains types de structure de données, notamment lorsque f est constitué de composantes déséquilibrées. Nous proposons donc deux nouvelles méthodes d'estimation, basées sur des choix de couples (réponse, élément du dictionnaire) différents, pour lesquelles nous obtenons des résultats numériques compétitifs avec l'état de l'art. Comme dans le cas univarié, nous montrons que ces deux algorithmes sont consistants, et de plus, qu'avec grande probabilité, nous recouvrons le support de f.

Ces travaux ont fait l'objet d'une publication, à paraître dans *Journal of Statistical Planning and Inference* [CCAGV14].

4.2 Application à l'analyse de sensibilité dans le cas de variables dépendantes

L'analyse de sensibilité permet d'analyser un modèle mathématique en étudiant l'impact de la variabilité des facteurs d'entrée du modèle sur la variable de sortie. Pour cela, une méthode classiquement utilisée est basée sur l'analyse de la variance fonctionnelle.

On considère le modèle :

$$Y = f(X) + \varepsilon, \tag{I.18}$$

où f est une fonction mesurable du vecteur aléatoire $X = (X^1, ..., X^p)$ et ε est une variable aléatoire indépendante de X qui modélise la variabilité de la réponse Y par rapport à f. On suppose dans un premier temps que les variables d'entrée sont **indépendantes**.

Pour quantifier l'importance de la variable d'entrée X^i sur la sortie Y, nous étudions les fluctuations de la variance de Y si on fixe la variable X^i à une valeur arbitraire x, $\mathbb{E}\left(\mathrm{Var}(Y|X^i = x)\right)$. Plus cette quantité est petite, plus le fait de fixer X^i réduit la variance et plus X^i est influente dans le modèle. Cette quantité est en revanche difficile à calculer en pratique. Remarquons que la variance se décompose sous la forme :

$$\mathrm{Var}(Y) = \mathrm{Var}\left(\mathbb{E}(Y|X^i = x)\right) + \mathbb{E}\left(\mathrm{Var}(Y|X^i = x)\right).$$

On en déduit, de façon équivalente, qu'on peut étudier la quantité $\mathrm{Var}\left(\mathbb{E}(Y|X^i = x)\right)$, qui est cette fois d'autant plus grande que la variable X^i est importante vis-à-vis de Y.

On définit alors l'indice de sensibilité (ou indice de Sobol) du premier ordre de Y à X^i [Sob01] mesurant l'effet d'une variable donnée sur la sortie par :

$$S_i = \frac{\mathrm{Var}\left(\mathbb{E}(Y|X^i = x)\right)}{\mathrm{Var}(Y)}. \tag{I.19}$$

Pour définir les indices de sensibilité d'ordre supérieur à un, mesurant les effets d'interaction entre les variables, Sobol s'appuie sur la décomposition de la fonction f sous forme de fonctionnelles d'ANOVA [Sob67]. La décomposition fonctionnelle d'ANOVA consiste à exprimer f

comme une somme de fonctions de dimension croissante :

$$\begin{aligned} f(X) &= f_\emptyset + \sum_{i=1}^{p} f_i(X^i) + \sum_{1\leq i<j\leq p} f_{ij}(X^i, X^j) + \cdots + f_{1,\cdots,p}(X) \\ &= \sum_{u\in S} f_u(X^u), \end{aligned} \tag{I.20}$$

où S est l'ensemble des sous-ensembles de $[\![1,p]\!]$, et où, pour tout $u \in S$, on note X^u le sous-vecteur de X défini par $X^u := (X^i)_{i\in u}$.

La fonction f se décompose ainsi en une somme de 2^p fonctions, où $f_\emptyset$ est une constante, f_i, $i \in [\![1,p]\!]$ sont es effets principaux, les fonctions $f_{i,j}$, $i < j \in [\![1,p]\!]$ représentent les effets d'interaction d'ordre 2...

Cette décomposition n'est naturellement pas unique car il peut exister une infinité de choix de ses composantes. Pour assurer l'unicité d'une telle décomposition, dans le cas où les entrées sont indépendantes, il est nécessaire d'imposer des contraintes d'orthogonalité sur les composantes :

$$\forall i \in u, \forall u \in S, \quad \int f_u(X^u) dX^i = 0. \tag{I.21}$$

Une conséquence de la condition (I.21) est que les termes de la décomposition sont deux à deux orthogonaux, c'est-à-dire :

$$\forall u \neq v \in S, \quad \int f_u(X^u) f_v(X^v) dX = 0.$$

La décomposition (I.20) sous les contraintes (I.21) assure la décomposition de la variance globale du modèle en une somme de variances partielles :

$$\begin{aligned} \mathrm{Var}(Y) &= \sum_{i=1}^{p} \mathrm{Var}(f_i(X_i)) + \sum_{1\leq i<j\leq p} \mathrm{Var}(f_{ij}(X_i, X_j)) + \cdots + \mathrm{Var}(f_{1,\cdots,p}(X)) \\ &= \sum_{u\in S} \mathrm{Var}(f_u(X^u)). \end{aligned}$$

On généralise alors la notion d'indices de Sobol d'ordre $|u|$ dans le cas de variables indépendantes :

Définition 4.1 (Indices de sensibilité)**.** *L'indice de sensibilité d'ordre $|u|$ mesurant la contribution de X^u sur Y est donné par :*

$$S_u = \frac{\mathrm{Var}(f_u(X^u))}{\mathrm{Var}(Y)} = \frac{\mathrm{Var}\left(\mathbb{E}(Y|X^u)\right) + \sum_{v\subset u}(-1)^{|u|-|v|}\mathrm{Var}\left(\mathbb{E}(Y|X^v)\right)}{\mathrm{Var}(Y)}.$$

Cette définition témoigne de l'attrait de la décomposition fonctionnelle d'ANOVA pour obtenir une définition claire des indices de sensibilité. En pratique, il est rare de pouvoir calculer analytiquement les indices de Sobol puisqu'il faut pour cela connaître les lois des entrées et la forme de la réponse. On utilise alors des méthodes d'estimation traditionnelles pour décomposer f suivant l'Equation (I.20) et estimer les indices de Sobol.

Pour effectuer une analyse de sensibilité, on suppose généralement que les variables d'entrées sont indépendantes. Mais cela reste évidemment une hypothèse non réaliste et il faut donc envisager le cas où la sortie du modèle Y dépend de plusieurs paramètres d'entrées non indépendants. Les propriétés d'orthogonalité de la décomposition ne sont alors plus vérifiées et la définition des

indices de Sobol, donnée par le biais de la Définition 4.1 peut mener à de fausses interprétations des effets d'interaction. De plus, les méthodes d'estimation classiques ne peuvent pas être appliquées directement au cas non-indépendant et doivent être généralisées.

Dans le cas de variables non indépendantes, Chastaing *et al.* [CGP12] proposent une généralisation de la décomposition de Hoeffding sous des hypothèses concernant la loi jointe des données. Cette décomposition consiste à exprimer f comme une somme de fonctions de dimension croissante, non pas mutuellement orthogonales, mais hiérarchiquement orthogonales. Elle permet aussi de définir proprement les indices de Sobol.

Plaçons nous dans le cadre du modèle (I.18), où les variables $X^1, ..., X^p$ sont non indépendantes. Soit S une famille de sous-ensembles de $[\![1, p]\!]$. Pour tout $u \in S$, on note X^u le sous-vecteur de X défini par $X^u := (X^i)_{i \in u}$ et X^{-u} le vecteur complémentaire à X^u. La Définition 4.2 a pour but d'introduire la notion d'orthogonalité hiérarchique.

Définition 4.2 (Orthogonalité hiérarchique)**.** *On dit que la suite de fonctions $(f_u(X^u))_{u \in S}$ satisfait la propriété d'orthogonalité hiérarchique, si $\forall u \in S^*$, $f_u(X^u) \in H_u$, où*

$$H_u := \{h_u(X^u), \langle h_u, h_v \rangle = 0, \forall\ v \subset u, \forall\ h_v \in H_v\}.$$

Suivant les travaux de [CGP12], la fonction f se décompose alors de manière unique sous la forme d'une somme de fonctions $f_\emptyset, f_1, ..., f_{\{1,...,p\}} \in H_\emptyset \times H_1 \times ... \times H_{\{1,...,p\}}$ hiérarchiquement orthogonales :

$$\begin{aligned} f(X) &= f_\emptyset + \sum_{i=1}^{p} f_i(X^i) + \sum_{1 \le i < j \le p} f_{ij}(X^i, X^j) + \cdots + f_{1,\cdots,p}(X) \\ &= \sum_{u \in S} f_u(X^u), \end{aligned} \tag{I.22}$$

où les fonctions $f_\emptyset, f_1, ..., f_{\{1,...,p\}}$ vérifient les propriétés d'orthogonalité hiérarchique.

En pratique, il n'est pas évident d'obtenir la décomposition de Hoeffding (I.22). Une méthode numérique, basée sur une procédure du type orthogonalisation de Gram-Schmidt, a été proposée par [CGP72] et permet de construire une base orthogonale $(\phi_i)_i$ pour les ensembles $(H_u)_{u \in S}$, ainsi qu'une approximation de cette base grâce à un premier échantillon d'observations $(Y_k, X_k)_{1 \le k \le n}$. Notre apport consiste notamment à montrer que celle-ci est proche de la base théorique. Pour estimer de manière parcimonieuse la décomposition (I.22) dans la base approximée, nous proposons ensuite d'utiliser un algorithme de $\mathbb{L}_2$-Boosting sur un deuxième échantillon d'observations. Nous montrons enfin la consistance de cette procédure.

Ce travail a été effectué en collaboration avec Gaëlle Chastaing, Sébastien Gadat et Clémentine Prieur et a fait l'objet d'une publication, à paraître dans *Statistica Sinica* [CCGP14].

4.3 Estimation de graphes acycliques dirigés

Enfin, le Chapitre IV de cette thèse est consacré au développement de méthodes d'optimisation pour l'apprentissage de réseaux de régulation de gènes, modélisés sous la forme de graphes acycliques dirigés (DAG). Rappelons que les travaux de [MB06] ont conduit à s'intéresser à des modèles de régressions linéaires du type :

$$\forall i \in [\![1, p]\!], \quad X^i = \sum_{j=1}^{p} G^i_j X^j + \varepsilon^i,$$

où $X := (X^1, ..., X^p)$ constitue le vecteur des données d'observation des p gènes du réseau, ε^i représente la présence de bruit pour l'observation du gène X^i et G^i_j symbolise l'effet du gène X^j sur le gène X^i.

Ce travail fait suite à une remarque de [Büh13], qui décompose la matrice $G = (G^j_i)_{1 \leq i,j \leq p}$ d'un DAG comme une fonction de deux variables (P, T) où P est une matrice de permutation et T est une matrice triangulaire inférieure stricte. De manière plus précise, ce couple de matrices permet de dissocier deux informations essentielles sur la structure du graphe : l'ordre des variables (quel est le gène le plus important ?) - information contenue dans P - et la parcimonie du graphe - information contenue dans T.

Une des difficultés majeures liée à l'inférence de réseaux est la non-identifiabilité du modèle : les données d'observation sont en général insuffisantes pour retrouver la structure exacte du réseau. Cependant, sous des hypothèses de *fidélité* du graphe notamment, on peut espérer retrouver une classe d'équivalence du réseau d'origine [Pea00], dont les éléments sont des graphes dont le squelette est identique mais dont l'orientation des arêtes peut être partiellement différentes.

Dans le cas où les variances sur le bruit des variables sont égales, ce problème a été résolu par Peters *et al.* [PB14] : le modèle est alors identifiable. Dans ce cadre particulier, nous nous intéressons à la minimisation de la log-vraisemblance que nous pénalisons à l'aide de la norme ℓ_1 afin de rendre parcimonieux le modèle reconstruit tout en facilitant la résolution du problème (critère convexe) :

$$\min_{P,T} \frac{1}{n} \left\| X - XPT^tP \right\|_F^2 + \lambda \left\| T \right\|_1, \tag{I.23}$$

où $\lambda > 0$ est le paramètre de pénalisation.

D'un point de vue théorique, les travaux de [vdGB13] autour de la pénalité en norme ℓ_0 nous ont permis d'obtenir des inégalités en prédiction et en estimation pour l'estimateur considéré. Ces résultats proviennent notamment d'une étude rigoureuse de l'estimation de l'ordre des variables au sein du graphe, et nécessitent malheureusement des hypothèses restrictives sur la dimension du modèle, n'incluant pas le cadre de la grande dimension :

Hypothèse 5. *Le nombre de variables p du réseau satisfait :*

$$p^3 \log p = \mathcal{O}(n),$$

où n représente la taille de l'échantillon.

D'un point de vue algorithmique cette fois, en collaboration avec Victor Picheny, nous avons développé un algorithme génétique, spécifiquement dédié à la résolution de ce problème. Cet algorithme permet d'explorer intelligemment l'ensemble des matrices de permutation afin de trouver une solution au problème d'optimisation (I.23).

Chapter II

Sparse regression and support recovery with $\mathbb{L}_2$-Boosting algorithms

This chapter has been accepted in a sligthly different form as [CCAGV14], as joint work with Christine Cierco-Ayrolles, Sébastien Gadat and Matthieu Vignes.

Abstract

This chapter focuses on the analysis of $\mathbb{L}_2$-Boosting algorithms for linear regressions. Consistency results were obtained for high-dimensional models when the number of predictors grows exponentially with the sample size n. We propose a new result for Weak Greedy Algorithms that deals with the support recovery, provided that reasonable assumptions on the regression parameter are fulfilled. For the sake of clarity, we also present some results in the deterministic case. Finally, we propose two multi-task versions of $\mathbb{L}_2$-Boosting for which we can extend these stability results, provided that assumptions on the restricted isometry of the representation and on the sparsity of the model are fulfilled. The interest of these two algorithms is demonstrated on various datasets.

1 Introduction

Context of our work This chapter presents a study of *Weak Greedy Algorithms* (WGA) and statistical $\mathbb{L}_2$-Boosting procedures derived from these WGA. These methods are dedicated to the approximation or estimation of several parameters that encode the relationships between input variables X and any response Y through a noisy linear representation $Y = f(X) + \varepsilon$, where ε models the amount of noise in the data. We assume that f may be linearly spanned on a predefined dictionary of functions $(g_j)_{j=1\dots p}$:

$$f(x) = \sum_{j=1}^{p} a_j g_j(x). \tag{II.1}$$

We aim at recovering unknown coefficients $(a_j)_{j=1\dots p}$ when one n-sample $(X_k, Y_k)_{k=1\dots n}$ is observed in the high-dimensional paradigm. Moreover, we are also interested in extending the Boosting methods to the multi-task situation described in [HTF09]: Y is described by m coordinates $Y = (Y^1, \dots, Y^m)$, and each one is modelled by a linear relationship $Y^i = f^i(X) + \varepsilon^i$. These relationships are now parametrised through the family of unknown coefficients $(a_{i,j})_{1\leq i\leq m, 1\leq j\leq p}$.

In both univariate or multivariate situations, we are primarily interested in the recovery of the structure (*i.e.* non-zero elements) of the matrix $A = (a_j)_{1 \leq j \leq p}$, when a limited amount of observations n is available compared to the large dimension p of the feature space. In brief, the goal is to identify significant relationships between variables X and Y. We formulate this paradigm as a feature selection problem: we seek relevant elements of the dictionary $(g_j(X))_{j=1...p}$ that explain (in)dependencies in a measured dataset.

Feature selection algorithms can be split into three large families: exhaustive subset exploration, subspace methods, and forward algorithms with shrinkage. The exhaustive search suffers from an obvious hard combinatorial problem (see [Hoc83]) and subspace methods such as [Gad08] are generally time consuming. In contrast, forward algorithms are fast, and shrinkage of greedy algorithms aims to reduce overfitting in stepwise subset selection (see [HTF09]). However, as pointed out by [ST07], *collinearities* may confuse greedy stepwise algorithms and subsequent estimates, which is not the case for the two other families of methods. Another main difficulty in our setting is that we often cope with *high-dimensional* situations where thousands of variables can be measured and where, at most, only a few hundred measures are manageable. For example, this is the case when dealing with biological network reconstruction, a problem that can be cast in a multivariate variable selection framework to decipher which regulatory relationships between entities actually dictate the evolution of the system [VVA$^+$11] [OM12]. Several strategies were proposed to circumvent these hindrances in a statistical framework. Among them, in addition to a control on the isometry effect of the matrix X, the leading assumption of the *sparsity* of the solution A leads to satisfactory estimations. All the more, it is a quite reasonable hypothesis in terms of the nature of some practical problems. We clarify this notion of sparsity and give bounds for the applicability of our results. Note that [Wai09] and [Ver12b] established the limit of the statistical estimation feasibility of latent structures in random matrices with Gaussian noise and Gaussian Graphical Model frameworks, respectively.

Related works Among the large number of recent advances on linear regression within a sparse univariate setting, we focused our point of view and investigate the use of Weak Greedy Algorithms for estimating regression parameters of Equation (II.1). Since the pioneering works of Schapire [Sch90] and Schapire and Freund [SF96], there has been an abundant literature on *Boosting algorithms* (as an example, see [BY10] for a review). Friedman [FLNP00a] gave a statistical view of Boosting and related it to the maximisation of the likelihood in a logistic regression scenario (see [Rid99]). Subsequent papers also proposed algorithmic extensions (*e.g.*, a functional gradient descent algorithm with $\mathbb{L}_2$ loss function, [BY03]). For prediction or classification purposes, Boosting techniques were shown to be particularly suited to large dataset problems. Indeed, just like the Lasso [Tib96] and the Dantzig Selector [CT07], which are two classical methods devoted to solving regression problems, Boosting uses variable selection, local optimisation and shrinkage. Even though Lasso, Dantzig and Elastic net ([ZH05]) estimators are inspired by penalised M-estimator methods and appear to be different from the greedy approach, like boosting methods, it is worthful to observe that, from an algorithmic point of view, these methods are very similar in terms of their practical implementation. Their behaviour is stepwise and based on correlation computed on the predicted residuals. We refer to [MRY07] for an extended comparison of such algorithms.

In a multivariate setting, some authors such as [LPvdGT11] or [OWJ11] use the geometric structure of an $\mathbb{L}_1$ ball derived from the Lasso approach . Others adopt a model selection strategy (see [SAB12]). Some authors also propose to use greedy algorithms such as Orthogonal Matching Pursuit developed in [ER10] or Basis Pursuit [GN08]. More recently, due to their attractive computational properties and to their ability to deal with high-dimensional predictors, Boosting

algorithms have been adapted and applied to bioinformatics, for microarray data analysis as well as for gene network inference ([Büh06] and [ADH09]).

Organisation of the paper The works of [Tem00] and [TZ11] provide estimates of the rate of the approximation of a function by means of greedy algorithms, which inspired our present work. Section 2 is dedicated to Weak Greedy Algorithms. We first recall some key results needed for our purpose. Section 2.1 may be omitted by readers familiar with such algorithms. In Section 2.2, we then provide a description of the behaviour of the $\mathbb{L}_2$-Boosting algorithm in reasonable noisy situations and in Section 2.3, we obtain a new result on support recovery. In Section 3, we describe two new extensions of this algorithm, referred to as Boost-Boost algorithms, dedicated to the multi-task regression problem. We also establish consistency results under some mild sparsity and isometry conditions. Section 4 is dedicated to a comparison of the performances of the Boosting methodology we propose with several approaches (Bootstrap Lasso [Bac08], Random Forests [Bre01] and remMap [PZB$^+$10]) on several simulated datasets. The features of these datasets allow us to conclude that the two new Boosting algorithms are competitive with other state-of-the art methods, even when the theoretical assumptions of our results are challenged.

2 Greedy algorithms

In this section, we mainly describe some essential and useful results on greedy algorithms that build approximations of any functional data f by stepwise iterations. In the deterministic case (*i.e.*, noiseless setting), we will refer to 'approximations' of f . In the noisy case, these approximations of f will be designated as 'sequential estimators'. Results on Weak Greedy Algorithms are derived from those of Temlyakov [Tem00] and adapted to our particular setting. We slightly enrich the presentation by adding some supplementary shrinkage parameters, which offers additional flexibility in the noisy setting. It will in fact be necessary to understand the behaviour of the WGA with shrinkage to show statistical consistency of the Boosting method.

2.1 A review of the Weak Greedy Algorithm (WGA)

Let H be a Hilbert space and $\|.\|$ denote its associated norm, which is derived from the inner product $\langle , \rangle$ on H. We define a *dictionary* as a (finite) subset $\mathcal{D} = (g_1, \ldots, g_p)$ of H, which satisfies:

$$\forall g_i \in \mathcal{D}, \quad \|g_i\| = 1 \quad \text{and} \quad \overline{\text{Span}\,\mathcal{D}} = H.$$

Greedy algorithms generate iterative approximations of any $f \in H$, using linear combination of elements of $\mathcal{D}$. Consistent with the notations of [Tem00], let $G_k(f)$ (as opposed to $R_k(f)$) denote the approximation of f (as opposed to the residual) at step k of the algorithm. These quantities are linked through the following equation:

$$R_k(f) = f - G_k(f).$$

At step k, we select an element $\varphi_k \in \mathcal{D}$, which provides a sufficient amount of information on residual $R_{k-1}(f)$. The first shrinkage parameter ν stands for a tolerance towards the optimal correlation between the current residual and any dictionary element. It offers some flexibility in the choice of the new element plugged into the model. Though the elements φ_k selected by (II.2) along the algorithm may not be uniquely defined, the convergence of the algorithm is still guaranteed by our next results. The second shrinkage parameter γ is the standard step-length

Algorithm 1: Weak Greedy Algorithm (WGA)

Input: Function f, $(\nu, \gamma) \in (0,1]^2$ (shrinkage parameters), k_{up} (number of iterations).
Initialisation: $G_0(f) = 0$ and $R_0(f) = f$.
for $k = 1$ *to* k_{up} **do**

Step 1 Select φ_k in $\mathcal{D}$ such that:

$$|\langle \varphi_k, R_{k-1}(f)\rangle| \geq \nu \max_{g\in\mathcal{D}} |\langle g, R_{k-1}(f)\rangle|, \tag{II.2}$$

Step 2 Compute the current approximation and residual:

$$\begin{aligned} G_k(f) &= G_{k-1}(f) + \gamma\langle R_{k-1}(f), \varphi_k\rangle\varphi_k \\ R_k(f) &= R_{k-1}(f) - \gamma\langle R_{k-1}(f), \varphi_k\rangle\varphi_k. \end{aligned} \tag{II.3}$$

end

parameter of the Boosting algorithm. It avoids a binary add-on, and actually smoothly inserts the new predictor into the approximation of f.

When the two shrinkage parameters equal 1, this algorithm is also known as Pure Greedy Algorithm (PGA). Refinements of PGA, or WGA, including a barycentre average between $G_{k-1}(f)$ and $\langle R_{k-1}(f), \varphi_k\rangle\varphi_k$, may improve the algorithm convergence rate (see Section 2.1.2 below). However, we decide to only consider the simplest version of WGA, because these improvements generally disappear in the noisy framework from a theoretical point of view (see [Büh06]).

2.1.1 Convergence of the WGA

Following the arguments developed in [Tem00], we can extend their results and obtain a polynomial approximation rate:

Theorem 2.1 (Temlyakov, 2000)**.** *Let $B > 0$ and assume that $f \in \mathcal{A}(\mathcal{D}, B)$, where*

$$\mathcal{A}(\mathcal{D}, B) = \left\{ f = \sum_{j=1}^{p} a_j g_j, \quad \textit{with} \quad \sum_{j=1}^{p} |a_j| \leq B \right\},$$

then, for a suitable constant C_B that only depends on B, at each step k of the algorithm:

$$\|R_k(f)\| \leq C_B(1 + \nu^2\gamma(2-\gamma)k)^{-\frac{\nu(2-\gamma)}{2(2+\nu(2-\gamma))}}.$$

Proof. The proof of Theorem 2.1 is already given in [Tem00] for a shrinkage parameter γ set to 1 and is generalized in [Büh06] for $\gamma \neq 1$. We write it for any choice of γ and ν. Let $\nu, \gamma \in [0,1]$. We define the sequences $(v_k)_{k\geq 0}$ and $(w_k)_{k\geq 1}$ as:

$$\begin{aligned} &\forall k \geq 1, \quad w_k = w_{k-1} + \gamma\,|\langle R_{k-1}(f), \varphi_k\rangle| \quad \text{where } \varphi_k \text{ is given by Equation (II.2)}, \\ &\forall k \geq 0, \quad v_k = \|R_k(f)\|^2. \end{aligned}$$

Remark that $R_k(f) \in \mathcal{A}(\mathcal{D}, w_k)$ with the initialization $w_0 = B$. Then, the following inequality is satisfied for v_k:

$$\begin{aligned} v_{k-1} = \|R_{k-1}(f)\|^2 &= |\langle R_{k-1}(f), R_{k-1}(f)\rangle| \\ &\leq w_{k-1} \sup_{g_j\in\mathcal{D}} |\langle R_{k-1}(f), g_j\rangle| \\ &\leq w_{k-1}\nu^{-1}\,|\langle R_{k-1}(f), \varphi_k\rangle|. \end{aligned} \tag{II.4}$$

By definition, v_k satisfies the recursive relation $v_k = v_{k-1} - \gamma(2-\gamma)\langle R_{k-1}(f), \varphi_k\rangle^2$, and we deduce from Equation (II.4) that:

$$v_k \leq v_{k-1} - \nu^2\gamma(2-\gamma)\frac{v_{k-1}^2}{w_{k-1}^2} = v_{k-1}\left(1 - \nu^2\gamma(2-\gamma)\frac{v_{k-1}}{w_{k-1}^2}\right).$$

Since $(w_k)_{k\geq 1}$ is an increasing sequel, we then have:

$$v_k w_k^{-2} \leq v_{k-1}\left(1 - \nu^2\gamma(2-\gamma)\frac{v_{k-1}}{w_{k-1}^2}\right) w_{k-1}^{-2} = v_{k-1}w_{k-1}^{-2}\left(1 - \nu^2\gamma(2-\gamma)v_{k-1}w_{k-1}^{-2}\right).$$

Lemma 2.1 is a technical lemma useful to obtain an upper bound for $(v_k w_k^{-2})_{k\geq 1}$.

Lemma 2.1. *Let $\alpha > 0$ and $(u_k)_{k\geq 0}$ be a sequence of real numbers such that $u_0 \leq 1$ and*

$$\forall k \geq 1, \qquad u_k \leq u_{k-1}(1 - \alpha u_{k-1}).$$

Then $(u_k)_{k\geq 0}$ converges to 0 and $u_k \leq (1+\alpha k)^{-1}$.

For all $k \geq 0$, denote $u_k := v_k w_k^{-2}$. Then, we can apply Lemma 2.1 to the sequence $(u_k)_{k\geq 0}$ with $\alpha = \nu^2\gamma(2-\gamma)$:

$$v_k w_k^{-2} \leq (1 + \nu^2\gamma(2-\gamma)k)^{-1}. \tag{II.5}$$

Using Inequality (II.4) in the recursive relation $v_k = v_{k-1} - \gamma(2-\gamma)\langle R_{k-1}(f), \varphi_k\rangle^2$, we can show that:

$$v_k \leq v_{k-1}\left(1 - \gamma(2-\gamma)\nu\frac{|\langle R_{k-1}(f), \varphi_k\rangle|}{w_{k-1}}\right), \tag{II.6}$$

and by definition of w_k,

$$w_k = w_{k-1}\left(1 + \gamma\frac{|\langle R_{k-1}(f), \varphi_k\rangle|}{w_{k-1}}\right). \tag{II.7}$$

We deduce from Equations (II.6) and (II.7) and inequality $(1+x)^\alpha \leq 1 + \alpha x$, for some $0 \leq \alpha \leq 1$ and $x \geq 0$ (derived by a convexity argument), that:

$$\begin{aligned} v_k w_k^{\nu(2-\gamma)} &\leq v_{k-1}w_{k-1}^{\nu(2-\gamma)}\left(1 - \gamma^2(2-\gamma)^2\nu^2\frac{|\langle R_{k-1}, \varphi_k\rangle|^2}{w_{k-1}^2}\right) \\ &\leq v_{k-1}w_{k-1}^{\nu(2-\gamma)} \\ &\leq \|f\|^2 B^{\nu(2-\gamma)}. \end{aligned} \tag{II.8}$$

We conclude the proof of Theorem 2.1 with Equations (II.5) and (II.8) by observing that:

$$\begin{aligned} \left(\|R_k(f)\|^2\right)^{2+\nu(2-\gamma)} = v_k^{2+\nu(2-\gamma)} &= (v_k w_k^{-2})^{\nu(2-\gamma)}(v_k w_k^{\nu(2-\gamma)})^2 \\ &\leq C_B^2(1 + \nu^2\gamma(2-\gamma)k)^{-\nu(2-\gamma)}. \end{aligned}$$

□

2.1.2 Refinements of the WGA

If the two shrinkage parameters ν and γ of the Boosting algorithm are set to 1, Theorem 2.1 provides the following estimate:

$$\|R_k(f)\| \leq C_B k^{-1/6}.$$

The WGA is then also known as the Pure Greedy Algorithm (PGA). Some modifications of the PGA can improve this approximation property, considering relaxed variants of the algorithm. As an example, at each step k of the algorithm, we replace Step 2 of Algorithm 1 by:

$$\begin{aligned} G_k(f) &= \left(1 - \frac{1}{k}\right) G_{k-1}(f) + \frac{1}{k}\langle R_{k-1}(f), \varphi_k\rangle \varphi_k, \\ R_k(f) &= f - G_k(f), \end{aligned}$$

where φ_k satisfies:

$$|\langle \varphi_k, R_{k-1}(f)\rangle| = \max_{g\in\mathcal{D}} |\langle g, R_{k-1}(f)\rangle| .$$

For any function $f \in \mathcal{A}(\mathcal{D}, B)$, the relaxed PGA provides the approximation order:

$$\forall k, \quad \|R_k(f)\| \leq C_B k^{-1/2}.$$

A weak version of this algorithm can also be defined, including a shrinkage parameter, for which we obtain too a polynomial approximation rate (for more details, see [DT96]).

2.2 The Boosting algorithm in the noisy regression framework

This section aims at extending the previous results to several noisy situations. We present a noisy version of WGA, and we clarify the consistency result of [Büh06] by careful considerations on the empirical residuals instead of the theoretical ones (which are in fact unavailable, see Remark 1).

2.2.1 The noisy Boosting algorithm

In this paragraph, we consider an unknown $f \in H$, and we observe some i.i.d. real random variables $(X_i, Y_i)_{1\leq i\leq n}$, with arbitrary distributions. We cast the following regression model on the dictionary $\mathcal{D}$:

$$\forall i = 1 \dots n, \qquad Y_i = f(X_i) + \varepsilon_i, \qquad \text{where} \qquad f = \sum_{j=1}^{p_n} a_j g_j(X_i). \tag{II.9}$$

In the noisy framework, $\mathcal{D}$ is composed of p n-dimensional vectors, and we denote D the matrix which coefficients are $D := g_j(X_i)_{1\leq i\leq n, 1\leq j\leq p}$. The Hilbert space $\mathbb{L}_2(P) := \{f, \|f\|^2 = \int f^2(x)dP(x) < \infty\}$, is endowed with the inner product $\langle f, g\rangle = \int {}^t f(x)g(x)dP_X(x)$, where P_X is the unknown law of the random variables X. Let us define the empirical inner product $\langle , \rangle_{\text{n}}$ as:

$$\forall (h_1, h_2) \in H, \qquad \langle h_1, h_2\rangle_{\text{n}} := \frac{1}{n}\sum_{i=1}^{n} h_1(X_i)h_2(X_i) \quad \text{and} \quad \|h_1\|_{\text{n}}^2 := \frac{1}{n}\sum_{i=1}^{n} h_1(X_i)^2.$$

The empirical WGA is analogous to the coupled Equations (II.2) and (II.3), replacing $\langle , \rangle$ by the empirical inner product $\langle , \rangle_{\text{n}}$.

Algorithm 2: Noisy Weak Greedy Algorithm

Input: Observations $(X_i, Y_i)_{i=\{1\dots n\}}$, $\gamma \in (0,1]$ (shrinkage parameter), k_{up} (number of iterations).

Initialisation: $\hat{G}_0(f) = 0$.

for $k = 1$ *to* k_{up} **do**

Step 1: Select $\varphi_k \in \mathcal{D}$ such that:

$$\left|\langle Y - \hat{G}_{k-1}(f), \varphi_k\rangle_{\mathrm{n}}\right| = \max_{1\le j\le p_n} \left|\langle Y - \hat{G}_{k-1}(f), g_j\rangle_{\mathrm{n}}\right|. \tag{II.10}$$

Step 2: Compute the current approximation and residual:

$$\hat{G}_k(f) = \hat{G}_{k-1}(f) + \gamma\langle Y - \hat{G}_{k-1}(f), \varphi_k\rangle_{\mathrm{n}}\varphi_k. \tag{II.11}$$

end

Remark 1. *The theoretical residual $\hat{R}_k(f) = \boldsymbol{f} - \hat{G}_k(f)$ cannot be used for the WGA (see Equations (II.10) and (II.11)) even with the empirical inner product, since f is not observed. Hence, only the observed residuals at step k, $Y - \hat{G}_k$, can be used in the algorithm. This point is not so clear in the initial work of [Büh06], since notations used in its proofs are read as if $\hat{R}_k(f) = f - \hat{G}_k(f)$ was available. We write more explicit proofs in Section 2.4.2.*

2.2.2 Stability of the Boosting algorithm

We will use the following two notations below: for any sequences $(a_n)_{n\ge0}$ and $(b_n)_{n\ge0}$ and a random sequence $(X_n)_{n\ge0}$, $a_n = \underset{n\to+\infty}{\mathcal{O}}(b_n)$ means that a_n/b_n is a bounded sequence, and $X_n = \underset{n\to+\infty}{o_P}(1)$ means that $\forall\varepsilon > 0$, $\lim_{n\to+\infty} \mathbb{P}(|X_n| \ge \varepsilon) = 0$. We recall here the standard assumptions on high-dimensional models.

Hypotheses $\mathbf{H_{dim}}$

$\mathbf{H_{dim-1}}$ For any $g_j \in \mathcal{D}$: $\mathbb{E}\left(g_j(X)^2\right) = 1$ and $\sup_{1\le j\le p_n, n\in\mathbb{N}} \|g_j(X)\|_\infty < \infty$.

$\mathbf{H_{dim-2}}$ The number of predictors p_n satisfies

$$p_n = \underset{n\to+\infty}{\mathcal{O}}\left(\exp(Cn^{1-\xi})\right), \text{ where } 0 < \xi \le 1 \text{ and } C > 0.$$

$\mathbf{H_{dim-3}}$ $(\varepsilon_i)_{i=1\dots n}$ are i.i.d centred variables in $\mathbb{R}$, independent from $(X_i)_{i=1\dots n}$, satisfying

$$\mathbb{E}|\varepsilon|^t < \infty, \quad \text{for some } t > \frac{4}{\xi}, \text{ where } \xi \text{ is given in } \mathbf{H_{dim-2}}.$$

$\mathbf{H_{dim-4}}$ The sequence $(a_j)_{1\le j\le p_n}$ satisfies:

$$\sup_{n\in\mathbb{N}} \sum_{j=1}^{p_n} |a_j| < \infty.$$

Assumption $\mathbf{H_{dim-1}}$ is clearly satisfied for compactly supported real polynomials or Fourier expansions with trigonometric polynomials. Assumption $\mathbf{H_{dim-2}}$ bounds the high dimensional

setting and states that $\log(p_n)$ should be, at the most, on the same order as n. Assumption $\mathbf{H_{dim-3}}$ specifies the noise and especially the size of its tail distribution. It must be centred with at least a bounded second moment. This hypothesis is required to apply the uniform law of large numbers and is satisfied by a great number of distributions, such as Gaussian or Laplace ones. The last assumption $\mathbf{H_{dim-4}}$ is a sparsity hypothesis on the unknown signal. It is trivially satisfied when the decomposition $(a_j)_{j=1...p_n}$ of f is bounded and has a fixed sparsity index: Card $\{i|a_i \neq 0\} \leq S$. Remark that it could be generalised to $\sum_{j=1}^{p_n} |a_j| \underset{n\to+\infty}{\longrightarrow} +\infty$ at the expense of additional restrictions on ξ and p_n.

We then formulate the first important result of the Boosting algorithm, obtained by [Büh06], which represents a *stability result*.

Theorem 2.2 (Consistency of WGA)**.** *Consider Algorithm 2 presented above and assume that Hypotheses* $\mathbf{H_{dim}}$ *are fulfilled. A sequence* $k_n := C\log(n)$ *then exists, with* $C < \xi/4\log(3)$*, so that:*

$$\mathbb{E}\left\|f - \hat{G}_{k_n}(f)\right\|_n^2 = \underset{n\to+\infty}{o_P}(1).$$

We only give the outline of the proof here. Details can be found in Section 2.4. A straightforward calculation shows that the theoretical residuals are updated as:

$$\hat{R}_k(f) = \hat{R}_{k-1}(f) - \gamma\langle\hat{R}_{k-1}(f), \varphi_k\rangle_n\varphi_k - \gamma\langle\varepsilon, \varphi_k\rangle_n\varphi_k. \tag{II.12}$$

The proof then results from the study of a *phantom* algorithm, which reproduces the behaviour of the deterministic WGA. In this algorithm, the inner product $\langle,\rangle$ replaces its empirical counterpart, and the (random) sample-driven choice of dictionary element $(\varphi_k)_{k\geq 0}$ is governed by Equation (II.10) of Algorithm 2. The phantom residuals are initialised by $\tilde{R}_0(f) = \hat{R}_0(f) = f$ and satisfy at step k:

$$\tilde{R}_k(f) = \tilde{R}_{k-1}(f) - \gamma\langle\tilde{R}_{k-1}(f), \varphi_k\rangle\varphi_k, \tag{II.13}$$

where φ_k is chosen using Equation (II.10). On the one hand, we establish an analogue of Equation (II.2) for φ_k which can allow us to apply Theorem 2.1 to the phantom residual $\tilde{R}_k(f)$. On the other hand, we provide an upper bound for the difference between $\hat{R}_k(f)$ and $\tilde{R}_k(f)$. The proof then results from a careful balance between these two steps.

2.3 Stability of support recovery

2.3.1 Ultra-high dimensional case

This paragraph presents our main results in the univariate case for the ultra-high dimensional case. We prove the stability of the support recovery with the noisy WGA. Provided that assumptions on the amplitude of the active coefficients of f and the structure of the dictionary are fulfilled, the WGA exactly recovers the support of f with high probability. This result is related to the previous work of [Tro04] and [Zha09] for recovering sparse signals using Orthogonal Matching Pursuit.

To state the theorem, we denote D as the $n \times p$ matrix whose columns are the p elements $(g_1, ..., g_p)$ of the dictionary $\mathcal{D}$. In the following text, $D_\mathcal{S}$ will be the matrix D restricted to the elements of $\mathcal{D}$ that are in $\mathcal{S} \subset [\![1, p]\!]$. Since $D_\mathcal{S}$ is not squared and therefore not invertible, D^+ is written as its pseudo-inverse. If we denote $\mathcal{S}$ as the support of f and S as its cardinality, we can then make the following assumptions.

Hypotheses $\mathbf{H_S}$: The matrix $D_{\mathcal{S}}$ satisfies:

$$\max_{j \notin \mathcal{S}} \left\| D_{\mathcal{S}}^{+} g_j \right\|_1 < 1.$$

This assumption is also known as the exact recovery condition (see [Tro04]). It will ensure that only active coefficients of f can be selected along the iterations of Algorithm 2 (noisy Boosting algorithm).

Hypotheses $\mathbf{H_{RE^-}}$: A $\lambda_{min} > 0$ independent of n exists so that:

$$\inf_{\beta, \mathrm{Supp}(\beta) \subset \mathcal{S}} \|D\beta\|^2 / \|\beta\|^2 \geq \lambda_{min}.$$

λ_{min} of Assumption $\mathbf{H_{RE^-}}$ is the smallest eigenvalue of the restricted matrix ${}^t D_{\mathcal{S}} D_{\mathcal{S}}$. Assumption $\mathbf{H_{RE^-}}$ stands for the restricted isometry condition [CT05] or the sparse eigenvalue condition (e.g., [Zha09] and [ZY06]). Remark that our assumption is different from that of [Zha09] since we assume that $\forall j, \|g_j\| = 1$. For more details about this assumption, see Section 3.1.2.

Hypotheses $\mathbf{H_{SNR}}$: Elements $(a_j)_{1 \leq j \leq p_n}$ satisfy:

$$\exists \kappa \in (0,1), \forall j \in \mathcal{S}, \qquad |a_j| \geq \log(n)^{-\kappa}.$$

Remark that the greater the number of variables is allowed to grow with n, the larger the value of active coefficients of f are and the more restrictive Assumption $\mathbf{H_{SNR}}$ is (see Section 2.3.2 below).

Theorem 2.3 (Support recovery)**.** *(i) Assume that Hypotheses $\mathbf{H_{dim}}$ and $\mathbf{H_S}$ hold. Then, with high probability, only active coefficients are selected by Equation (II.10) along iterations of Algorithm 2.*

(ii) Moreover, if Hypotheses $\mathbf{H_{RE^-}}$ and $\mathbf{H_{SNR}}$ hold with a sufficiently small $\kappa < \kappa^*$ (κ^* only depending on γ), then Algorithm 2 fully recovers the support of f with high probability.

Similar results are already known for other algorithms devoted to sparse problems (see [GN08] for Basis Pursuit algorithms, and [Tro04], [CJ11] or [Zha09] for Orthogonal Matching Pursuit (OMP)). It is also known for other signal reconstruction algorithms [OWJ11], [CW11], [Zha09], which also rely on a sparsity assumption. Our assumption is stronger than the condition obtained by [Zha09] since active coefficients should be bounded from below by a power of $\log(n)^{-1}$ instead of $\log(p)^{1/2} n^{-1/2}$ in Theorem 4 of [Zha09]. However, obtaining optimal conditions on active coefficients is not straightforward and beyond the scope of this paper. The *weak* aspect of WGA seems harder to handle compared to the treatment of OMP (for example) because the amplitude of the remaining coefficients on active variables has to be recursively bound from one iteration to the next, according to the size of shrinkage parameters.

Let $\rho := \max_{1 \leq i \neq j \leq n} |\langle g_i, g_j \rangle|$ be the coherence of the dictionary $\mathcal{D}$. For non-orthogonal dictionaries, which are common settings of real high-dimensional datasets, the coherence is non-null. A sufficient condition to obtain the support recovery result would then be $\rho(2S-1) < 1$, where $S := |\mathcal{S}|$ is the number of non-null coordinates of f, combined with $\mathbf{H_{SNR}}$. However, it should be observed that this assumption is clearly more restrictive than $\mathbf{H_{RE^-}}$ when the number of predictors p_n becomes large.

In summary, a trade-off between signal sparsity, dimensionality, signal-to-noise ratio and sample size has to be reached. We provide explicit constant bounds for results on similar problems.

Very interesting discussions can be found in [Wai09] (see their Theorems 1 and 2 for sufficient and necessary conditions for an *exhaustive search decoder* to succeed with high probability in recovering a function support) and in the section on *Sparsity and ultra-high dimensionality* of [Ver12b].

2.3.2 High dimensional case

In this paragraph, we restrict our study to high-dimensional models, where the number of predictors should be, at the most, on the same order of n: $p_n = \underset{n\to+\infty}{\mathcal{O}}(n^a)$ with $a > 0$ (reinforcement of $\mathbf{H_{dim-2}}$). Then, provided that Assumption $\mathbf{H^+_{SNR}}$ below is fulfilled, Theorem 2.3 still holds.

Hypotheses $\mathbf{H^+_{SNR}}$: Elements $(a_j)_{1\leq j\leq p_n}$ satisfy:

$$\exists \kappa \in (0,1), \forall j \in \mathcal{S}, \qquad |a_j| \geq n^{-\kappa}.$$

The proof of this result is given in Section 2.4.3. As a consequence, in the high-dimensional case, Assumption $\mathbf{H^+_{SNR}}$ is less restrictive than Assumption $\mathbf{H_{SNR}}$. Moreover, to ensure the consistency of Algorithm 2, the number of iterations is then allowed to grow with n, and the algorithm converges faster and can easily recover even small active coefficients of the true function f.

2.4 Proof of stability results for Boosting algorithms

2.4.1 Concentration inequalities

We begin by recalling some technical results. The first one is a concentration inequality that aims at comparing the sum of any independent random variables with its expected value. In the litterature, there exists many versions of the Bernstein's inequality. We propose here one, mainly used by [Büh06]:

Theorem 2.4 (Bernstein's inequality)**.** *Let Z_1,...,Z_n be random variables i.i.d such that*

$$\mathbb{E}(Z_i^2) \leq \sigma,$$
$$\|Z_i\|_\infty \leq c.$$

Then, the following concentration inequality is satisfied:

$$\forall t > 0, \quad \mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n Z_i - \mathbb{E}(Z_i)\right| > t\right) \leq 2\exp\left(-\frac{nt^2}{2(\sigma^2 + ct)}\right).$$

Lemma 2.2, given in [Büh06], provides a uniform law of large numbers, in order to compare inner products $\langle , \rangle_n$ and $\langle , \rangle$. It is useful to prove the theorems of Section 2.2.2 and 2.3, and does not rely on Boosting arguments.

Lemma 2.2. *Assume that Hypotheses $\mathbf{H_{dim}}$ are fulfilled on dictionary $\mathcal{D}$, f and ε, with $0 < \xi < 1$ as given in $\mathbf{H_{dim-2}}$, then:*

(i) $\sup_{1\leq i,j\leq p_n} |\langle g_i, g_j\rangle_n - \langle g_i, g_j\rangle| = \zeta_{n,1} = \mathcal{O}_P(n^{-\xi/2})$,

(ii) $\sup_{1\leq i\leq p_n} |\langle g_i, \varepsilon\rangle_n| = \zeta_{n,2} = \mathcal{O}_P(n^{-\xi/2})$,

(iii) $\sup_{1\leq i\leq p_n} |\langle f, g_i\rangle_n - \langle f, g_i\rangle| = \zeta_{n,3} = \mathcal{O}_P(n^{-\xi/2})$.

Proof. Denote $M := \sup_{1\leq j\leq p_n, n\in\mathbb{N}} \|g_j(X)\|_\infty$ in $\mathbf{H_{dim-1}}$. Then, Theorem 2.4 applied to random variables $(g_i(X_k))_{1\leq k\leq n}$ implies that, for all $t>0$:

$$\mathbb{P}\left(n^{\xi/2}\sup_{1\leq i,j\leq p_n}|\langle g_i,g_j\rangle_{\mathrm{n}}-\langle g_i,g_j\rangle|>t\right)\leq 2p_n^2\exp\left(-\frac{t^2n^{1-\xi}}{2(M^8+M^2tn^{-\xi/2})}\right),$$

and (i) follows using Assumption $\mathbf{H_{dim-2}}$.

Consider now (ii). Let $(K_n)_{n\in\mathbb{N}}$ be a sequel of real numbers. We define the truncated variables $(\varepsilon_k^t)_{k\in\mathbb{N}}$ as:

$$\varepsilon_k^t=\begin{cases}\varepsilon_k & \text{if } |\varepsilon_k|\leq K_n\\ sg(\varepsilon_k)K_n & \text{otherwise,}\end{cases}$$

where $sg(\varepsilon)$ denotes the sign of ε. We then have, for any $t>0$,

$$\begin{aligned}\mathbb{P}\left(n^{\xi/2}\sup_{1\leq i\leq p_n}|\langle g_i,\varepsilon\rangle_{\mathrm{n}}|>t\right) &\leq \mathbb{P}\left(n^{\xi/2}\sup_{1\leq i\leq p_n}\left|\langle g_i,\varepsilon-\varepsilon^t\rangle_{\mathrm{n}}\right|>t/3\right)\\ &\quad+\mathbb{P}\left(n^{\xi/2}\sup_{1\leq i\leq p_n}\left|\langle g_i,\varepsilon^t\rangle_{\mathrm{n}}-\langle g_i,\varepsilon^t\rangle\right|>t/3\right)\\ &\quad+\mathbb{P}\left(n^{\xi/2}\sup_{1\leq i\leq p_n}|\langle g_i,\varepsilon_t\rangle|>t/3\right)\\ &= I+II+III\end{aligned}$$

The first term of the previous inequality can be bounded as follows:

$$\begin{aligned}I &\leq \mathbb{P}(\exists 1\leq i\leq n, |\varepsilon_i|>K_n)\\ &\leq n\mathbb{P}(|\varepsilon|>K_n)\\ &\leq nK_n^{-s}\mathbb{E}(|\varepsilon|^s),\end{aligned}$$

using Bienaymé-Tchebychev inequality. If we set $K_n:=n^{\xi/4}$, we then deduce that I can become as small as possible with $s>4/\xi$ and Assumption $\mathbf{H_{dim-3}}$.

For II, we apply Theorem 2.4 to random variables $(g_i(X_k)\varepsilon_k^t)_{1\leq k\leq n}$:

$$II\leq 2p_n\exp\left(-\frac{(t^2/9)n^{1-\xi}}{2(M^4K_n^4+MK_nt/3n^{-\xi/2})}\right).$$

With $K_n=n^{\xi/4}$, we can then show with Assumption $\mathbf{H_{dim-2}}$ that $II\underset{n\to+\infty}{\longrightarrow}0$.

To finish, consider III. On the one hand, since $\mathbb{E}(g_i\varepsilon)=0$,

$$III=\mathbb{P}\left(n^{\xi/2}\sup_{1\leq i\leq p_n}\left|\langle g_i,\varepsilon^t-\varepsilon\rangle\right|>t/3\right).$$

On another hand,

$$\left|\langle g_i,\varepsilon^t-\varepsilon\rangle\right|\leq M\left|\mathbb{E}(\varepsilon-\varepsilon^t)\right|,$$

where

$$\begin{aligned}\left|\mathbb{E}(\varepsilon-\varepsilon^t)\right| &= \int_{|x|>K_n}(x-sg(x)K_n)dP_\varepsilon(x)\\ &\leq \int 1_{|x|>K_n}(|x|+K_n)dP_\varepsilon(x)\\ &\leq \left(\mathbb{E}\left|\varepsilon\right|^2\right)^{1/2}\left(\mathbb{P}\left(|\varepsilon|>K_n\right)\right)^{1/2}+K_n\mathbb{P}\left(|\varepsilon|>K_n\right).\end{aligned}$$

Using again Bienaymé-Tchebychev inequality, the following inequality is satisfied:

$$\left|\mathbb{E}(\varepsilon - \varepsilon^t)\right| \leq K_n^{-s/2} \left(\mathbb{E}\left|\varepsilon\right|^s\right)^{1/2} \left(\mathbb{E}\left|\varepsilon\right|^2\right)^{1/2} + K_n^{1-s}\mathbb{E}\left|\varepsilon\right|^s.$$

Hence, setting $K_n = n^{\xi/4}$, we can show that $III = 0$ for n large enough from Assumption $\mathbf{H_{dim-3}}$ and (ii) follows.

To prove the last point, we use the following bound:

$$\sup_{1\leq i\leq p_n} |\langle f, g_i\rangle_{\mathrm{n}} - \langle f, g_i\rangle| \leq \sum_{j=1}^{p_n} |a_j| \sup_{1\leq i,j\leq p_n} |\langle g_j, g_i\rangle_{\mathrm{n}} - \langle g_j, g_i\rangle|.$$

We conclude the proof of Lemma 2.2 with (i) and Assumption $\mathbf{H_{dim-4}}$. □

Denote $\zeta_n = \max\{\zeta_{n,1}, \zeta_{n,2}, \zeta_{n,3}, \zeta_{n,4}\} = \mathcal{O}_P(n^{-\xi/2})$. The following lemma (lemma 2 from [Büh06]) also holds.

Lemma 2.3. *Under Hypotheses* $\mathbf{H_{dim}}$*, a constant* $0 < C < +\infty$ *exists, independent of* n *and* k*, so that on set* $\Omega_n = \{\omega, |\zeta_n(\omega)| < 1/2\}$*:*

$$\sup_{1\leq j\leq p_n} \left|\langle \hat{R}_k(f), g_j\rangle_n - \langle \tilde{R}_k(f), g_j\rangle\right| \leq C\left(\frac{5}{2}\right)^k \zeta_n.$$

Proof. This lemma is given in [Büh06], but their notations are confusing since residuals $\hat{R}_k$ are used to compute φ_k instead of $Y - \hat{G}_k$ (see Remark 1 of Section 2.2). It is nevertheless possible to generalise its application field using Lemma 2.2. First, assume that $k = 0$. The desired inequality follows directly from point (iii) of Lemma 2.2. We now extend the proof by an inductive argument.

Denote $A_n(k,j) = \langle \hat{R}_k(f), g_j\rangle_{\mathrm{n}} - \langle \tilde{R}_k(f), g_j\rangle$. Then, from the recursive relations of Equations (II.12) and (II.13), we obtain:

$$\begin{aligned} A_n(k,j) &= \langle \hat{R}_{k-1}(f) - \gamma\langle \hat{R}_{k-1}(f), \varphi_k\rangle_{\mathrm{n}}\varphi_k - \gamma\langle \varepsilon, \varphi_k\rangle_{\mathrm{n}}\varphi_k, g_j\rangle_{\mathrm{n}} \\ &\quad -\langle \tilde{R}_{k-1}(f) - \gamma\langle \tilde{R}_{k-1}(f), \varphi_k\rangle\varphi_k, g_j\rangle \\ &= A_n(k-1,j) - \gamma \underbrace{\langle \tilde{R}_{k-1}(f), \varphi_k\rangle \left(\langle \varphi_k, g_j\rangle_{\mathrm{n}} - \langle \varphi_k, g_j\rangle\right)}_{=I} \\ &\quad -\gamma \underbrace{\langle \varphi_k, g_j\rangle_{\mathrm{n}} \left(\langle \hat{R}_{k-1}(f), \varphi_k\rangle_{\mathrm{n}} - \langle \tilde{R}_{k-1}(f), \varphi_k\rangle\right)}_{=II} - \gamma \underbrace{\langle \varepsilon, \varphi_k\rangle_{\mathrm{n}}\langle \varphi_k, g_j\rangle_{\mathrm{n}}}_{=III}. \end{aligned}$$

Expanding Equation (II.13) yields $\left\|\tilde{R}_k(f)\right\|^2 = \left\|\tilde{R}_{k-1}(f)\right\|^2 - \gamma(2-\gamma)\langle \tilde{R}_{k-1}(f), \varphi_k\rangle^2$. From the last equality, we deduce $\left\|\tilde{R}_k(f)\right\|^2 \leq \left\|\tilde{R}_{k-1}(f)\right\|^2 \leq \ldots \leq \|f\|^2$ and Lemma 2.2 (i) shows that

$$\sup_{1\leq j\leq p_n} |I| \leq \left\|\tilde{R}_{k-1}(f)\right\| \|\varphi_k\| \zeta_n \leq \|f\|\zeta_n.$$

Moreover,

$$\begin{aligned} \sup_{1\leq j\leq p_n} |II| &\leq \sup_{1\leq j\leq p_n} |\langle \varphi_k, g_j\rangle_{\mathrm{n}}| \sup_{1\leq j\leq p_n} |A_n(k-1,j)| \\ &\leq \left(\sup_{1\leq j\leq p_n} |\langle \varphi_k, g_j\rangle| + \zeta_n\right) \sup_{1\leq j\leq p_n} |A_n(k-1,j)| \quad \text{using } (i) \text{ of Lemma 2.2} \\ &\leq (1+\zeta_n) \sup_{1\leq j\leq p_n} |A_n(k-1,j)|. \end{aligned}$$

Finally, using (i) and (ii) from Lemma 2.2:

$$\sup_{1\leq j\leq p_n}|III| \leq \sup_{1\leq j\leq p_n}|\langle\varphi_k, g_j\rangle_{\mathrm{n}}| \sup_{1\leq j\leq p_n}|\langle\varepsilon, g_j\rangle_{\mathrm{n}}| \leq (1+\zeta_n)\zeta_n.$$

Using our bounds on I, II and III, and $\gamma < 1$, we obtain on Ω_n

$$\begin{aligned}\sup_{1\leq j\leq p_n}|A_n(k,j)| &\leq \sup_{1\leq j\leq p_n}|A_n(k-1,j)| + \zeta_n\|f\| + (1+\zeta_n)\sup_{1\leq j\leq p_n}|A_n(k-1,j)| + (1+\zeta_n)\zeta_n \\ &\leq \frac{5}{2}\sup_{1\leq j\leq p_n}|A_n(k-1,j)| + \zeta_n\left(\|f\| + \frac{3}{2}\right).\end{aligned}$$

A simple induction yields:

$$\begin{aligned}\sup_{1\leq j\leq p_n}|A_n(k,j)| &\leq \left(\frac{5}{2}\right)^k \underbrace{\sup_{1\leq j\leq p_n}|A_n(0,j)|}_{\leq\zeta_n} + \zeta_n\left(\|f\| + \frac{3}{2}\right)\sum_{\ell=0}^{k-1}\left(\frac{5}{2}\right)^\ell \\ &\leq \left(\frac{5}{2}\right)^k \zeta_n\left(1 + \left(\sup_{n\in\mathbb{N}}\sum_{j=1}^{p_n}|a_j| + \frac{3}{2}\right)\sum_{\ell=1}^{\infty}\left(\frac{5}{2}\right)^{-\ell}\right),\end{aligned}$$

which ends the proof of (i) by setting $C = 1 + \left(\sup_{n\in\mathbb{N}}\sum_{j=1}^{p_n}|a_j| + \frac{3}{2}\right)\sum_{\ell=1}^{\infty}\left(\frac{5}{2}\right)^{-\ell}$. □

2.4.2 Proof of consistency result

We aim then to apply Theorem 2.1 to the semi-population $\tilde{R}_k(f)$ version of $\hat{R}_k(f)$. This will be possible with high probability when $n \to +\infty$. We first observe that Lemma 2.3 holds when replacing the theoretical residual $\hat{R}_k(f)$ with the observed residual $Y - \hat{G}_k(f)$, thanks to Lemma 2.2 (ii). Hence, on the set Ω_n, by definition of φ_k:

$$\begin{aligned}\left|\langle Y - \hat{G}_{k-1}(f), \varphi_k\rangle_{\mathrm{n}}\right| &= \sup_{1\leq j\leq p_n}\left|\langle Y - \hat{G}_{k-1}(f), g_j\rangle_{\mathrm{n}}\right| \\ &= \sup_{1\leq j\leq p_n}\left\{\left|\langle\tilde{R}_{k-1}(f), g_j\rangle\right| - C\left(\frac{5}{2}\right)^{k-1}\zeta_n\right\}. \qquad \text{(II.14)}\end{aligned}$$

Applying Lemma 2.3 again on the set Ω_n, we have:

$$\begin{aligned}\left|\langle\tilde{R}_{k-1}(f), \varphi_k\rangle\right| &\geq \left|\langle Y - \hat{G}_{k-1}(f), \varphi_k\rangle_{\mathrm{n}}\right| - C\left(\frac{5}{2}\right)^{k-1}\zeta_n \\ &\geq \sup_{1\leq j\leq p_n}\left|\langle\tilde{R}_{k-1}(f), g_j\rangle\right| - 2C\left(\frac{5}{2}\right)^{k-1}\zeta_n. \qquad \text{(II.15)}\end{aligned}$$

Let $\tilde{\Omega}_n = \left\{\omega,\ \ \forall k \leq k_n,\ \ \sup_{1\leq j\leq p_n}\left|\langle\tilde{R}_{k-1}(f), g_j\rangle\right| > 4C\left(\frac{5}{2}\right)^{k-1}\zeta_n\right\}$. We deduce the following inequality from Equation (II.15):

$$\left|\langle\tilde{R}_{k-1}(f), \varphi_k\rangle\right| \geq \frac{1}{2}\sup_{1\leq j\leq p_n}\left|\langle\tilde{R}_{k-1}(f), g_j\rangle\right|. \qquad \text{(II.16)}$$

Consequently, on the set $\Omega_n \cap \tilde{\Omega}_n$, we can apply Theorem 2.1 to the family $(\tilde{R}_k(f^i))_k$, since it satisfies a WGA with constants $\tilde{\nu} = 1/2$.

$$\left\|\tilde{R}_k(f)\right\| \leq C_B \left(1 + \frac{1}{4}\gamma(2-\gamma)k\right)^{-\frac{2-\gamma}{2(6-\gamma)}}. \tag{II.17}$$

Now consider the set $\tilde{\Omega}_n^C = \left\{\omega,\ \ \exists\, k \leq k_n \ \sup_{1\leq j\leq p_n} |\langle \tilde{R}_{k-1}(f), g_j\rangle| \leq 4C\left(\frac{5}{2}\right)^{k-1}\zeta_n\right\}$. Remark that:

$$\begin{aligned}\left\|\tilde{R}_k(f)\right\|^2 &= \langle \tilde{R}_k(f), f - \gamma\sum_{j=0}^{k-1}\langle \tilde{R}_j(f), \varphi_j\rangle\varphi_j\rangle \\ &\leq \left(\sum_{j=1}^{p}|a_j| + \gamma\sum_{j=0}^{k-1}\left|\langle \tilde{R}_j(f), \varphi_j\rangle\right|\right)\sup_{1\leq j\leq p}\left|\langle \tilde{R}_k(f), g_j\rangle\right|.\end{aligned}$$

Then, since $\left\|\tilde{R}_k(f)\right\|$ is non-increasing and by definition of $\tilde{\Omega}_n^C$, we deduce that on $\tilde{\Omega}_n^C$,

$$\left\|\tilde{R}_k(f)\right\|^2 \leq 4C\left(\frac{5}{2}\right)^k \zeta_n \left(\sum_{j=1}^{p}|a_j| + \gamma k\,\|f\|\right), \tag{II.18}$$

Hence, on $(\Omega_n \cap \tilde{\Omega}_n) \cup \tilde{\Omega}_n^C$, using Equations (II.17) and (II.18),

$$\left\|\tilde{R}_k(f)\right\|^2 \leq C_B^2\left(1 + \frac{1}{4}\gamma(2-\gamma)k\right)^{-\frac{2-\gamma}{6-\gamma}} + 4C\left(\frac{5}{2}\right)^k \zeta_n\left(\sum_{j=1}^{p}|a| + \gamma k\,\|f\|\right). \tag{II.19}$$

To conclude, remark that $\mathbb{P}\left((\Omega_n \cap \tilde{\Omega}_n) \cup \tilde{\Omega}_n^C\right) \geq \mathbb{P}(\Omega_n) \underset{n\to+\infty}{\longrightarrow} 1$. Inequality (II.19) holds almost surely for all ω and for a sequence $k_n < (\xi/4\log(3))\log(n)$, which grows sufficiently slowly:

$$\left\|\tilde{R}_{k_n}(f)\right\| = \underset{n\to+\infty}{o_P}(1). \tag{II.20}$$

To end the proof, let $k \geq 1$ and consider $A_k = \left\|\hat{R}_k(f) - \tilde{R}_k(f)\right\|$. By definition:

$$\begin{aligned}A_k &= \left\|\hat{R}_{k-1}(f) - \gamma\langle \hat{R}_{k-1}(f), \varphi_k\rangle_{\mathrm{n}}\varphi_k - \gamma\langle \varepsilon, \varphi_k\rangle_{\mathrm{n}}\varphi_k - \left(\tilde{R}_{k-1}(f) - \gamma\langle \tilde{R}_{k-1}(f), \varphi_k\rangle\varphi_k\right)\right\| \\ &\leq A_{k-1} + \gamma\left|\langle Y - \hat{G}_{k-1}(f), \varphi_k\rangle_{\mathrm{n}} - \langle \tilde{R}_{k-1}(f), \varphi_k\rangle\right|. \end{aligned}\tag{II.21}$$

Under Hypotheses $\mathbf{H_{dim}}$, we deduce the following inequality on Ω_n from Equation (II.21):

$$A_k \leq A_{k-1} + \gamma\left(C\left(\frac{5}{2}\right)^{k-1} + 1\right)\zeta_n. \tag{II.22}$$

Using $A_0 = 0$, we deduce recursively from Equation (II.22) that, on Ω_n, since $k := k_n$ grows sufficiently slowly:

$$A_{k_n} \xrightarrow[n\to+\infty]{\mathbb{P}} 0. \tag{II.23}$$

Finally observe that $\left\|\hat{R}_{k_n}(f)\right\| \leq \left\|\tilde{R}_{k_n}(f)\right\| + A_{k_n}$. The conclusion holds using Equation (II.20) and (II.23).

2.4.3 Proof of support recovery

Ultra-high dimensional case We now detail the proof of Theorem 2.3 which represents the exact recovery of the support with high probability. It should be recalled that we denote as S (respectively $\mathcal{S}$) the sparsity (respectively the support) of f. We suppose that the current residuals could be decomposed on $\mathcal{D}$ as $\hat{R}_k(f) = \sum_{j=1}^{p_n} \theta_j^k g_j$, where $(\theta_j^k)_j$ is S_k-sparse, with support $\mathcal{S}_k$.

Proof of (i). The aim of the first part of the proof is to show that along the iterations of Boosting, we only select elements of the support of f using Equation (II.10). Since $\mathcal{S}_0 = \mathcal{S}$, we only have to show that $(\mathcal{S}_k)_{k\geq 0}$ is non-increasing, which implies that successive residual supports satisfy $\mathcal{S}_k \subset \mathcal{S}_{k-1}$. At the initial step $k = 0$, $S_0 = S$ and $\mathcal{S}_0 = \mathcal{S}$. The proof works now by induction, and we assume that $\mathcal{S}_{k-1} \subset \mathcal{S}$. Using the same outline of proof of Lemma 2.3, we have:

$$\forall g_j \in \mathcal{D}, \qquad \left|\langle Y - \hat{G}_{k-1}(f), g_j\rangle_{\mathrm{n}} - \langle \hat{R}_{k-1}(f), g_j\rangle\right| \leq C\zeta_n \left(\frac{5}{2}\right)^{k-1}. \tag{II.24}$$

On the one hand, we deduce from Equation (II.24) below that:

$$\forall j \in \mathcal{S}_{k-1}, \quad \left|\langle Y - \hat{G}_{k-1}(f), g_j\rangle_{\mathrm{n}}\right| \geq \left|\langle \hat{R}_{k-1}(f), g_j\rangle\right| - C\zeta_n \left(\frac{5}{2}\right)^{k-1}. \tag{II.25}$$

On the other hand, for $j \notin \mathcal{S}_{k-1}$, we also have:

$$\left|\langle Y - \hat{G}_{k-1}(f), g_j\rangle_{\mathrm{n}}\right| \leq \left|\langle \hat{R}_{k-1}(f), g_j\rangle\right| + C\zeta_n \left(\frac{5}{2}\right)^{k-1}. \tag{II.26}$$

Now denote $M_k := \max_{j\in\mathcal{S}_{k-1}} \left|\langle \hat{R}_{k-1}(f), g_j\rangle\right|$ and $M_k^C := \max_{j\notin\mathcal{S}_{k-1}} \left|\langle \hat{R}_{k-1}(f), g_j\rangle\right|$. We recall that element j is selected at step k following Equation (II.10). Hence, we deduce from Equations (II.25) and (II.26) that $j \in \mathcal{S}_k$ is in $\mathcal{S}_{k-1}$ if the following inequality is satisfied:

$$M_k > M_k^C + 2C\zeta_n \left(\frac{5}{2}\right)^{k-1}. \tag{II.27}$$

The next step of the proof consists in comparing the two quantities M_k and M_k^C. Note that M and M^C can be rewritten as $\left\|{}^t D_{\mathcal{S}_{k-1}} \hat{R}_{k-1}(f)\right\|_\infty$ and $\left\|{}^t D_{\mathcal{S}_{k-1}^C} \hat{R}_{k-1}(f)\right\|_\infty$. Following the arguments of [Tro04], we have:

$$\begin{aligned}\frac{M_k^C}{M_k} &= \frac{\left\|{}^t D_{\mathcal{S}_{k-1}^C}\, {}^t D^+_{\mathcal{S}_{k-1}}\, {}^t D_{\mathcal{S}_{k-1}} \hat{R}_{k-1}(f)\right\|_\infty}{\left\|{}^t D_{\mathcal{S}_{k-1}} \hat{R}_{k-1}(f)\right\|_\infty} \\ &\leq \left\|{}^t D_{\mathcal{S}_{k-1}^C}\, {}^t D^+_{\mathcal{S}_{k-1}}\right\|_{\infty,\infty},\end{aligned}$$

where $\|.\|_{q,q}$ is the subordonate norm of the space $(\mathbb{R}^q, \|.\|_q)$. In particular, the norm $\|.\|_{\infty,\infty}$ equals the maximum absolute row of its arguments and we also have:

$$\frac{M_k^C}{M_k} \leq \left\|D^+_{\mathcal{S}_{k-1}} D_{\mathcal{S}_{k-1}^C}\right\|_{1,1} = \max_{j\notin\mathcal{S}_{k-1}} \left\|D^+_{\mathcal{S}_{k-1}} g_j\right\|_1.$$

Using Assumption $\mathbf{H_S}$ and the recursive assumption $\mathcal{S}_{k-1} \subset \mathcal{S}$, we obtain that $M_k > M_k^C$.

The end of the proof of (i) follows with Equation (II.27) for $k := k_n$ given by Theorem 2.2 which implies that $\zeta_n(5/2)^k \to 0$. □

Proof of (ii). The second part of the proof consists in checking that, along the iterations of the Boosting algorithm, every correct element of the dictionary is chosen at least once.

Assume that one element j_0 of $\mathcal{S}$ is never selected. Then, if we denote as $\theta^k = (\theta_j^k)_{1\leq j\leq p_n}$ the decomposition of $\hat{R}_{k-1}(f)$ on $\mathcal{D}$, we obtain:

$$\left\|\theta^k\right\|^2 = \sum_j (\theta_j^k)^2 \geq (\theta_{j_0}^k)^2 = a_{j_0}^2, \tag{II.28}$$

where a_{j_0} is the true coefficient of f associated to the element g_{j_0}.

Moreover, note that

$$\left\|\hat{R}_{k-1}(f)\right\|^2 = \left\|D\theta^k\right\|^2 \geq \lambda_{min}\left\|\theta^k\right\|^2, \tag{II.29}$$

with $\lambda_{min} := \inf_{\beta,\mathrm{Supp}(\beta)\subset\mathcal{S}} \|D\beta\|^2 / \|\beta\|^2 > 0$ by Assumption $\mathbf{H_{RE}}^-$.

Equation (II.29) deserves special attention since $\left(\left\|\hat{R}_{k-1}(f)\right\|\right)_k$ decreases with k. More precisely, Equations (II.19) and (II.22) of Section 2.4.2 provide the following bound for $\left\|\hat{R}_{k-1}(f)\right\|$:

$$\left\|\hat{R}_{k-1}(f)\right\|^2 \leq (C\log(n))^{-\alpha},$$

where $\alpha := \frac{2-\gamma}{6-\gamma}$.

The sought contradiction is obtained using Assumption $\mathbf{H_{SNR}}$ in Equation (II.28) as soon as

$$\lambda_{min}\log(n)^{-2\kappa} \geq (C\log(n))^{-\alpha},$$

i.e., when $\kappa < \kappa^* := (2-\gamma)/2(6-\gamma)$. This ends the proof of the support consistency. □

High-dimensional case We explain here the proof of Theorem 2.3 in the high-dimensional case, when the number of predictors satisfies $p_n = \underset{n\to+\infty}{\mathcal{O}}(n^a)$, with $a > 0$. Following carefully the proof of Theorems 2.2 and 2.3 in the ultra-high dimensional case, we can show that ζ_n in the uniform law of large numbers 2.2 is in the order of $\mathcal{O}_P\left(\exp\left(-n^{1-\xi}\right)\right)$.

The number of iterations of Algorithm 2 is then allowed to grow with n since $k_n := Cn^\beta$, with $\beta < 1-\xi$, which ensures that $\left(\frac{5}{2}\right)^{k_n}\zeta_n$ is small enough. The decrease of the theoretical residuals $\left(\left\|\hat{R}_k\right\|^2\right)_k$ is finally on the order of $Cn^{-\beta\alpha}$, where C depends on the shrinkage parameters γ and ξ, although α depends on the rate of approximation of the boosting ($\alpha = (2-\gamma)/(2(6-\gamma))$). Now Theorem 2.3 follows with $\kappa < \kappa^* := \beta\alpha/2$.

3 A new $\mathbb{L}_2$-Boosting algorithm for multi-task situations

In this section, our purpose is to extend the above algorithm and results to the multi-task situation. The main focus of this work lies in the choice of the optimal task to be boosted. We therefore propose a new algorithm that follows the initial spirit of iterative Boosting (see [Sch99] for further details) and the multi-task structure of f. We first establish an approximation result in the deterministic setting and we then extend the stability results of Theorems 2.2 and 2.3 to the so called Boost-Boost algorithm for noisy multi-task regression.

3.1 Multi-task Boost-Boost algorithms

3.1.1 Description of the algorithm

Let $H_m := H^{\otimes m}$ denote the Hilbert space obtained by m-tensorisation with the inner product:

$$\forall (f, \tilde{f}) \in H_m^2, \qquad \langle f, \tilde{f} \rangle_{H_m} = \sum_{i=1}^{m} \langle f^i, \tilde{f}^i \rangle_H.$$

Given any dictionary $\mathcal{D}$ on H, each element $f \in H_m$ will be described by its m coordinates $f = (f^1, \dots f^m)$, where each f^i is spanned on $\mathcal{D}$, with unknown coefficients:

$$\forall i \in [\![1, m_n]\!], \qquad f^i = \sum_{j=1}^{p_n} a_{i,j} g_j. \tag{II.30}$$

A canonical extension of WGA to the multi-task problem can be computed as follows (Algorithm 3).

Algorithm 3: Boost-Boost algorithm

Input: $f = (f^1, ..., f^m)$, $(\gamma, \mu, \nu) \in (0, 1]^3$ (shrinkage parameters), k_{up} (number of iterations).

Initialisation: $G_0(f) = 0_{H_m}$ and $R_0(f) = f$.

for $k = 1$ *to* k_{up} **do**

Step 1: Select f^{i_k} according to:

$$\left\| R_{k-1}(f^{i_k}) \right\|^2 \geq \mu \max_{1 \leq i \leq m} \left\| R_{k-1}(f^i) \right\|^2, \qquad \text{[Residual } L^2 \text{ norm]} \tag{II.31}$$

or to

$$\sum_{j=1}^{p} \langle R_{k-1}(f^{i_k}), g_j \rangle^2 \geq \mu \max_{1 \leq i \leq m} \sum_{j=1}^{p} \langle R_{k-1}(f^i), g_j \rangle^2, \qquad [\mathcal{D}\text{-Correlation sum}] \tag{II.32}$$

Step 2: Select $\varphi_k \in \mathcal{D}$ such that:

$$\left| \langle R_{k-1}(f^{i_k}), \varphi_k \rangle \right| \geq \nu \max_{1 \leq j \leq p} \left| \langle R_{k-1}(f^{i_k}), g_j \rangle \right|. \tag{II.33}$$

Step 3: Compute the current approximation:

$$\begin{aligned} &G_k(f^i) = G_{k-1}(f^i), \quad \forall i \neq i_k, \\ &G_k(f^{i_k}) = G_{k-1}(f^{i_k}) + \gamma \langle R_{k-1}(f^{i_k}), \varphi_k \rangle \varphi_k. \end{aligned} \tag{II.34}$$

Step 4: Compute the current residual: $R_k(f) = f - G_k(f)$.

end

In the multi-task framework at step k, it is crucial to choose the coordinate from among the residuals that is meaningful and thus *most* needs improvement, as well as the best regressor $\varphi_k \in \mathcal{D}$. The main idea is to focus on the coordinates that are still poorly approximated. We introduce a new shrinkage parameter $\mu \in (0, 1]$. It allows a tolerance towards the optimal choice

of the coordinate to be boosted, relying on either the Residual L^2 norm -Equation (II.31)- or on the $\mathcal{D}$-Correlation sum -Equation (II.32).

Note that this latter choice is rather different from the choice proposed in [GN08], which uses the multichannel energy and sums the correlations of each coordinate of the residuals to any element of the dictionary. Comments on pros and cons of minimising the Residual L^2 norm or the $\mathcal{D}$-Correlation sum viewed as the correlated residual can be found in [CT07] (page 2316). Although [CT07] tends toward a final advantage for the $\mathcal{D}$-Correlation sum alternative, we also consider the Residual L^2 norm that seems more natural. In fact, it relies on the norm of the residuals themselves instead of the sum of information gathered by individual regressors on each residual. Moreover, conclusions of [CT07] are more particularly focused on an orthogonal design matrix.

We can discuss the added value brought by the Residual L^2 norm Boost-Boost algorithm. Compared to running m times standard WGA on each coordinates of the residuals, the proposed algorithm is efficient when the coordinates of the residuals are unbalanced, *i.e.* when few columns possess most of the information to be predicted. In contrast, when WGA is applied to well balanced tasks, there is no clear advantage to using the Residual L^2 norm Boost-Boost algorithm.

We use coupled criteria of Equations (II.31) and (II.33) in the Residual L^2 norm Boost-Boost algorithm, whereas we use criteria of Equations (II.32) and (II.33) in its $\mathcal{D}$-Correlation sum counterpart.

3.1.2 Approximation results in the deterministic setting

We consider the sequence of functions $(R_k(f))_k$ recursively built according to our Boost-Boost Algorithm 3 with either choice (II.31) or (II.32). Since $\overline{\operatorname{Span}\mathcal{D}} = H$, for any $f \in H_m$, each f^i can be decomposed in H, and we denote S^i as the minimal amount of sparsity for such a representation. Denote D as the $n \times p$ matrix whose columns are the p elements $(g_1, ..., g_p)$ of the dictionary $\mathcal{D}$. As previously, $D_{\mathcal{S}}$ will be the matrix D restricted to the elements of $\mathcal{D}$ that are in $\mathcal{S} \subset [\![1, ,]\!]p$. We then prove a first approximation result provided that the following assumption is true.

Hypotheses $\mathbf{H_{RE^+}}$ A $\lambda_{max} < \infty$ independent of n exists so that

$$\sup_{\beta, \operatorname{Supp}(\beta) \subset \mathcal{S}} \|D\beta\|^2 / \|\beta\|^2 \leq \lambda_{max}.$$

λ_{max} of Assumption $\mathbf{H_{RE^+}}$ is the largest eigenvalue of the restricted matrix ${}^tD_{\mathcal{S}}D_{\mathcal{S}}$. Remark that

$$\begin{aligned} \forall u \in \mathbb{R}^S, \quad {}^tu{}^tD_{\mathcal{S}}D_{\mathcal{S}}u = \|D_{\mathcal{S}}u\|^2 &\leq \|u\|^2 \sum_{j \in \mathcal{S}} \|g_j\|^2 \\ &\leq S \|u\|^2 . \end{aligned} \tag{II.35}$$

Then, denote v as the eigenvector associated with the largest eigenvalue λ_{max} of ${}^tD_{\mathcal{S}}D_{\mathcal{S}}$. Equation (II.35) then makes it possible to write:

$${}^tv\lambda_{max}v \leq S \|v\|^2 ,$$

which directly implies the following bound for λ_{max}: $\lambda_{max} \leq S$. Then, if S is kept fixed independent from n, Assumption $\mathbf{H_{RE^+}}$ trivially holds.

On the other hand, if S is allowed to grow with n as $S/n \to_{n\to+\infty} l$, [BCT11] proves that the expected value of λ_{max} is also bounded for the special Wishart matrices:

$$\mathbb{E}(\lambda_{max}) \underset{n\to+\infty}{\longrightarrow} (1+\sqrt{l})^2.$$

Moreover, they show that fluctuations of λ_{max} around $\mathbb{E}\lambda_{max}$ are exponentially small with n, that is:

$$\mathbb{P}\left(\lambda_{max} > \mathbb{E}\lambda_{max} + \varepsilon\right) \underset{n\to+\infty}{\longrightarrow} 0, \quad \text{exponentially fast with } n.$$

In the case of matrices with subgaussian entries, with probability $1 - c\exp(-S)$, [Ver12a] also provides the following bound for λ_{min} and λ_{max}:

$$\sqrt{S/n} - c \leq \lambda_{min} \leq \lambda_{max} \leq \sqrt{S/n} + c.$$

The following theorem presents a convergence rate for the two Boost-Boost algorithms.

Theorem 3.1 (Convergence of the Boost-Boost Algorithm)**.** *Let $f = (f^1, \dots f^m) \in H_m$ so that, for any coordinate i, $f^i \in \mathcal{A}(\mathcal{D}, B)$.*

(i) A suitable constant C_B exists that only depends on B so that the approximations provided by the Residual L^2 norm Boost-Boost algorithm satisfy, for all $k \geq m$

$$\sup_{1\leq i\leq m} \left\|R_k(f^i)\right\| \leq C_B \mu^{-\frac{1}{2}} \nu^{-\frac{\nu(2-\gamma)}{2+\nu(2-\gamma)}} \left(\gamma(2-\gamma)\right)^{-\frac{\nu(2-\gamma)}{2(2+\nu(2-\gamma))}} \left(\frac{k}{m}\right)^{-\frac{\nu(2-\gamma)}{2(2+\nu(2-\gamma))}}.$$

(ii) Assume that Hypotheses $\mathbf{H_{RE^-}}$ and $\mathbf{H_{RE^+}}$ hold. A suitable constant $C_{\lambda_{min},B}$ then exists so that the approximations provided by the $\mathcal{D}$-Correlation sum Boost-Boost algorithm satisfy, for all $k \geq m$

$$\sup_{1\leq i\leq m} \left\|R_k(f^i)\right\| \leq C_{\lambda_{min},B} \mu^{-\frac{1}{2}} \nu^{-\frac{\nu(2-\gamma)}{2+\nu(2-\gamma)}} \left(\gamma(2-\gamma)\right)^{-\frac{\nu(2-\gamma)}{2(2+\nu(2-\gamma))}} \left(\frac{k}{m}\right)^{-\frac{\nu(2-\gamma)}{2(2+\nu(2-\gamma))}}.$$

Remark 2. *Note first that Theorem 3.1 is a uniform result over the m_n coordinates. Then, note that Assumptions $\mathbf{H_{RE^-}}$ and $\mathbf{H_{RE^+}}$ are needed to obtain the second part of the theorem since we have to compare each coordinates of the residual with the coordinate chosen at step k. For the Residual L^2 norm Boost-Boost algorithm, this comparison trivially holds.*

3.1.3 Proof of Theorem 3.1

We breakdown the proof of Theorem 3.1 into several steps here. It should be recalled that $\mathcal{D} = \{(g_j), 1 \leq j \leq p\}$ is a dictionary that spans H. We set any $f = (f^1, \dots, f^m) \in H_m$ such that $f^i \in \mathcal{A}(\mathcal{D}, B)$.

The first key remark is that, if we denote $s_i(k)$ as the number of steps in which i is invoked until step k, for all $i \in [\![1, m]\!]$, we deduce from Theorem 2.1 that:

$$\forall k \geq 1, \quad \left\|R_{k-1}(f^i)\right\| \leq C_B(1 + \nu^2\gamma(2-\gamma)s_i(k-1))^{-\frac{\nu(2-\gamma)}{2(2+\nu(2-\gamma))}}. \tag{II.36}$$

The second key point of the proof consists in comparing $R_k(f^i)$ and $R_k(f^{i_k})$, where i_k is chosen using Equation (II.31) or (II.32). For the Boost-Boost Residual L^2 norm algorithm, this step is not pivotal since, using Equation (II.31),

$$\sup_{1\leq i\leq m} \left\|R_k(f^i)\right\| \leq \mu^{-1/2} \left\|R_k(f^{i_k})\right\|. \tag{II.37}$$

However, for the Boost-Boost $\mathcal{D}$-Correlation sum algorithm, we can prove the following lemma:

Lemma 3.1. *Suppose that Assumptions* $\mathbf{H_{RE^-}}$ *and* $\mathbf{H_{RE^+}}$ *hold. Then, for any* k*:*

$$\sup_{1\leq i\leq m} \left\|R_{k-1}(f^i)\right\|^2 \leq \mu^{-1}\left\|R_{k-1}(f^{i_k})\right\|^2 \left(\frac{\lambda_{max}}{\lambda_{min}}\right)^3,$$

where λ_{min} *and* λ_{max} *(given by Assumptions* $\mathbf{H_{RE^-}}$ *and* $\mathbf{H_{RE^+}}$*) are the smallest and the largest eigenvalues of* ${}^tD_{\mathcal{S}}D_{\mathcal{S}}$*.*

Proof of Lemma 3.1. Assume that each residual $R_k(f^i)$ is expanded on $\mathcal{D}$ at step k as: $R_k(f^i) = \sum_{j=1}^p \theta_{i,j}^k g_j$, where $(\theta_{i,j}^k)_{1\leq j\leq p}$ is S_k^i-sparse, with support $\mathcal{S}_k^i$. Note that, along the iterations of the Boost-Boost algorithm, an incorrect element of the dictionary cannot be selected using Equation (II.33) (see Theorem 3.4 for some supplementary details). We observe then that Assumptions $\mathbf{H_{RE^-}}$ and $\mathbf{H_{RE^+}}$ imply that at each step, each approximation is at most S-sparse. We present an elementary lemma that will be very useful until the end of the proof.

Lemma 3.2. *Let* $\mathcal{D} = (g_1, ..., g_p)$ *a dictionary on* H*. Denote* D *as the matrix whose columns are the elements of* $\mathcal{D}$ *and for any* $\mathcal{S} \subset [\![1, p]\!]$*,* $D_{\mathcal{S}}$ *the matrix restricted to the elements of* $\mathcal{D}$ *that are in* $\mathcal{S}$*. Then, if we denote* λ_{min} *and* λ_{max} *as the smallest and the largest eigenvalues of the restricted matrix* ${}^tD_{\mathcal{S}}D_{\mathcal{S}}$*, the two propositions hold.*

(i) For any $\mathcal{S}$*-sparse family* $(a_j)_{1\leq j\leq p}$*, we have:*

$$\lambda_{min}\left(\sum_{j=1}^p |a_j|^2\right) \leq \left\|\sum_{j=1}^p a_j g_j\right\|^2 \leq \lambda_{max}\left(\sum_{j=1}^p |a_j|^2\right).$$

(ii) For any function f *spanned on* $\mathcal{D}$ *as* $f = \sum_{j=1}^p a_j g_j$*, where* $(a_j)_j$ *is* S *-sparse, we have*

$$\lambda_{min}^2 \sum_{j=1}^p |a_j|^2 \leq \sum_{j=1}^p |\langle f, g_j\rangle|^2 \leq \lambda_{max}^2 \sum_{j=1}^p |a_j|^2.$$

Proof of Lemma 3.2. Since λ_{min} and λ_{max} are the largest and the smallest eigenvalues of ${}^tD_{\mathcal{S}}D_{\mathcal{S}}$, the following inequality holds:

$$\forall \beta | \operatorname{Supp}(\beta) \subset \mathcal{S}, \quad \lambda_{min} \leq \frac{\|D\beta\|^2}{\|\beta\|^2} \leq \lambda_{max}.$$

The end of the proof of (i) follows for any $\mathcal{S}$-sparse family $(a_j)_j$.

To prove (ii), remark that:

$$\forall j \in [\![1, p]\!], \quad \langle f, g_j\rangle = \sum_i a_i \left({}^tDD\right)_i^j.$$

Since ${}^tD_{\mathcal{S}}D_{\mathcal{S}}$ is symmetric, it smallest and largest eigenvalues equal λ_{min}^2 and λ_{max}^2 and, using tha fact that $(a_j)_j$ is a $\mathcal{S}$-sparse family, we conclude that:

$$\lambda_{min}^2 \sum_{j=1}^p |a_j|^2 \leq \sum_{j=1}^p |\langle f, g_j\rangle|^2 \leq \lambda_{max}^2 \sum_{j=1}^p |a_j|^2.$$

□

Now, let $i \neq i_k$. By Lemma 3.2 (right hand side -r.h.s.- of (ii) and left hand side -l.h.s.- of (i)) combined with Assumption $\mathbf{H_{RE^-}}$, we have

$$\sum_{j=1}^{p} \left|\langle R_{k-1}(f^{i_k}), g_j\rangle\right|^2 \leq \left\|R_{k-1}(f^{i_k})\right\|^2 \frac{\lambda_{max}^2}{\lambda_{min}}. \tag{II.38}$$

Moreover Lemma 3.2 again (l.h.s. of (ii) and r.h.s. of (i)) and Assumption $\mathbf{H_{RE^+}}$ show that

$$\forall 1 \leq i \leq m, \quad \sum_{j=1}^{p} \left|\langle R_{k-1}(f^{i}), g_j\rangle\right|^2 \geq \left\|R_{k-1}(f^{i})\right\|^2 \frac{\lambda_{min}^2}{\lambda_{max}}. \tag{II.39}$$

By definition of i_k (see Equation (II.32) in the Boost-Boost algorithm), we deduce that:

$$\begin{aligned} \forall i \in [\![1, m]\!], \qquad \sum_{j=1}^{p} \left|\langle R_{k-1}(f^{i_k}), g_j\rangle\right|^2 &\geq \mu \sum_{j=1}^{p} \left|\langle R_{k-1}(f^{i}), g_j\rangle\right|^2 \\ &\geq \mu \left\|R_{k-1}(f^{i})\right\|^2 \frac{\lambda_{min}^2}{\lambda_{max}}. \end{aligned} \tag{II.40}$$

The conclusion follows by using Equations (II.38) and (II.40). □

To conclude, we consider the Euclidean division of k by m: $k = mK + d$, where the remainder d is not greater than the divisor m. A coordinate $i^* \in \{1 \dots m\}$, that is selected at least K times by Equation (II.31) or (II.32) exists, hence $s_{i^*}(k) \geq K$. We also denote k^* as the last step which selects i^* before step k. Since $\left(\left\|R_k(f^i)\right\|\right)_k$ is a non-increasing sequence along the iterations of the algorithm, by Equation (II.36), we have that:

$$\left\|R_{k-1}(f^{i^*})\right\| \leq \left\|R_{k^*-1}(f^{i^*})\right\| \leq C_B(1 + \nu^2\gamma(2-\gamma)(K-1))^{-\frac{\nu(2-\gamma)}{2(2+\nu(2-\gamma))}}. \tag{II.41}$$

The conclusion holds noting that $\frac{k}{m} - 1 \leq K \leq \frac{k}{m}$ and $\nu < 1$, and using our bounds (II.37) for the Boost-Boost Residual L^2 norm algorithm, or Lemma 3.1 for the Boost-Boost $\mathcal{D}$-Correlation sum algorithm.

3.2 Stability of the Boost-Boost algorithms for noisy multi-task regression

3.2.1 The noisy Boost-Boost algorithm

The noisy WGA for the multi-task problem is described by Algorithm 4 where we replace the inner product $\langle , \rangle$ by the empirical inner product $\langle , \rangle_n$.

3.2.2 Stability of the noisy Boost-Boost algorithms

We establish a theoretical convergence result for these two versions of the multi-task WGA. We first state several assumptions adapted to the multi-task setting.

Hypotheses $\mathbf{H^{Mult}_{dim}}$

$\mathbf{H^{Mult}_{dim-1}}$ For any $g_j \in \mathcal{D}$: $\mathbb{E}[g_j(X)^2] = 1$ and $\sup_{1 \leq j \leq p_n, n \in \mathbb{N}} \|g_j(X)\|_\infty < \infty$.

$\mathbf{H^{Mult}_{dim-2}}$ $\xi \in (0,1), C > 0$ exist so that the number of predictors and tasks (p_n, m_n) satisfies

$$p_n \vee m_n = \underset{n \to +\infty}{\mathcal{O}} \left(\exp\left(Cn^{1-\xi}\right)\right).$$

Algorithm 4: Noisy Boost-Boost algorithm

Input: Observations $(X_i, Y_i)_{1\leq i\leq n}$, $\gamma \in (0,1]$ (shrinkage parameter), k_{up} (number of iterations).

Initialisation: $\hat{G}_0(f) = 0_{H_m}$.

for $k = 1$ *to* k_{up} **do**

Step 1: Select i_k according to:

$$\left\|Y^{i_k} - \hat{G}_{k-1}(f^{i_k})\right\|_{\mathrm{n}}^2 = \max_{1\leq i\leq m}\left\|Y^{i} - \hat{G}_{k-1}(f^{i})\right\|_{\mathrm{n}}^2, \qquad \text{[Residual } L^2 \text{ norm]}$$

or to

$$\sum_{j=1}^{p}\langle Y^{i_k} - \hat{G}_{k-1}(f^{i_k}), g_j\rangle_{\mathrm{n}}^2 = \max_{1\leq i\leq m}\sum_{j=1}^{p}\langle Y^{i} - \hat{G}_{k-1}(f^{i}), g_j\rangle_{\mathrm{n}}^2. \quad [\mathcal{D}\text{-Correlation sum}]$$

Step 2: Select $\varphi_k \in \mathcal{D}$ such that:

$$\left|\langle Y^{i_k} - \hat{G}_{k-1}(f^{i_k}), \varphi_k\rangle_{\mathrm{n}}\right| = \max_{1\leq j\leq p}\left|\langle Y^{i} - \hat{G}_{k-1}(f^{i}), g_j\rangle_{\mathrm{n}}\right|.$$

Step 3: Compute the current approximation:

$$\hat{G}_k(f^i) = \hat{G}_{k-1}(f^i), \quad \forall i \neq i_k,$$
$$\hat{G}_k(f^{i_k}) = \hat{G}_{k-1}(f^{i_k}) + \gamma\langle Y^{i_k} - \hat{G}_{k-1}(f^{i_k}), \varphi_k\rangle_{\mathrm{n}}\varphi_k.$$

end

$\mathbf{H}^{\mathbf{Mult}}_{\mathbf{dim-3}}$ $(\varepsilon_i)_{i=1\ldots n}$ are i.i.d centered in $\mathbb{R}^{m_n}$, independent from $(X_i)_{i=1\ldots n}$ so that for some $t > \frac{4}{\xi}$, where ξ is defined in $\mathbf{H}^{\mathbf{Mult}}_{\mathbf{dim-2}}$,

$$\sup_{1\leq j\leq m_n, n\in\mathbb{N}} \mathbb{E}\left|\varepsilon^j\right|^t < \infty.$$

Moreover, the variance of ε^j does not depend on j: $\forall (j,\tilde{j}) \in [\![1, m_n]\!]^2$, $\quad \mathbb{E}\left|\varepsilon^j\right|^2 = \mathbb{E}\left|\varepsilon^{\tilde{j}}\right|^2$.

$\mathbf{H}^{\mathbf{Mult}}_{\mathbf{dim-4}}$ The sequence $(a_{i,j})_{1\leq j\leq p_n, 1\leq i\leq m_n}$ satisfies:

$$\sup_{n\in\mathbb{N}, 1\leq i\leq m_n}\sum_{j=1}^{p_n}|a_{i,j}| < \infty.$$

It should be noted that a critical change appears in Hypothesis $\mathbf{H}^{\mathbf{Mult}}_{\mathbf{dim-3}}$. Indeed, all tasks should be of equal variances. We thus need to normalise the data before applying the Boost-Boost algorithms.

We can therefore derive a result on the consistency of the Residual L^2 norm Boost-Boost algorithm. This extends the result of Theorem 2.2 for univariate WGA.

Theorem 3.2 (Consistency of the Boost-Boost Residual L^2 norm algorithm)**.** *Assume that Hypotheses* $\mathbf{H}^{\mathbf{Mult}}_{\mathbf{dim}}$, $\mathbf{H}_{\mathbf{RE}^-}$ *and* $\mathbf{H}_{\mathbf{RE}^+}$ *are fulfilled. A sequence* $k_n := C\log(n)$ *then exists, with* $C < \xi/4\log(3)$*, so that:*

$$\sup_{1\leq i\leq m_n}\left\{\mathbb{E}\|f^i - \hat{G}_{k_n}(f^i)\|_{(n)}^2\right\} = \underset{n\to+\infty}{o_P}(1).$$

As regards the Boost-Boost algorithm defined with the sum of correlations, if the number of predictors p_n satisfies a more restrictive assumption than $\mathbf{H}^{\mathbf{Mult}}_{\mathbf{dim-2}}$, we prove a similar result.

Theorem 3.3 (Consistency of the Boost-Boost $\mathcal{D}$-Correlation sum algorithm)**.** *Assume that Hypotheses* $\mathbf{H}_{\mathbf{RE}^-}$ *and* $\mathbf{H}_{\mathbf{RE}^+}$ *are fulfilled and* $\mathbf{H}^{\mathbf{Mult}}_{\mathbf{dim}}$ *holds with* $p_n = \underset{n\to+\infty}{\mathcal{O}}\left(n^{\xi/4}\right)$*. A sequence* $k_n := C\log(n)$ *then exists with* $C < \xi/8\log(3)$ *so that:*

$$\sup_{1\le i\le m_n} \left\{\mathbb{E}\left\|f^i - \hat{G}_{k_n}(f^i)\right\|_n^2\right\} = \underset{n\to+\infty}{o_P}(1).$$

We concede that Assumption $\mathbf{H}^{\mathbf{Mult}}_{\mathbf{dim-2}}$ includes the very high dimensional case. Theorem 3.3 has a slightly more restrictive assumption, and encompasses the high dimensional perspective from a theoretical point of view. A proof of these theorems is given in Section 3.3.

3.2.3 Stability of support recovery

We can also obtain a consistency result for the support of the Boost-Boost algorithms.

Theorem 3.4 (Support recovery)**.** *Assume Hypotheses* $\mathbf{H}^{\mathbf{Mult}}_{\mathbf{dim}}$, $\mathbf{H}_{\mathbf{S}}$, $\mathbf{H}_{\mathbf{RE}^-}$ *and* $\mathbf{H}_{\mathbf{RE}^+}$ *are fulfilled, then the two propositions hold.*

(*i*) *With high probability, only active coefficients are selected along iterations of Algorithm 4.*

(*ii*) *Moreover, if Assumption* $\mathbf{H}_{\mathbf{SNR}}$ *holds with a sufficiently small* $\kappa < \kappa^*$ *(with* κ^* *depending on* γ*), then both Boost-Boost procedures fully recover the support of* f *with high probability.*

A proof of this result is given in Section 3.3.

3.3 Proof of stability results for multi-task $\mathbb{L}_2$-Boosting algorithms

3.3.1 Proof of Theorem 3.4

We begin this section by clarifying the proof of Theorem 3.4 since this result is needed to prove all others multi-task results. The proof proceeds in the same way as in Section 2.4.3. Our focus is on the choice of the regressor to add in the model, regardless the column chosen to be regressed in the previous step. Therefore, in order to simplify notations, index i may be omitted and we can do exactly the same computations.

3.3.2 Proof of Theorems 3.2 and 3.3

The proof of consistency results in the multi-task case is the same as in Section 2.4.2. Hence, we consider a semi-population version of the two Boost-Boost algorithms: let $(\tilde{R}_k(f))_k$ be the phantom residuals, that are now living in H_m, initialised by $\tilde{R}_0(f) = f$, and satisfy at step k:

$$\begin{aligned}
&\tilde{R}_k(f^i) = \tilde{R}_{k-1}(f^i) \ \text{ if } \ i \neq i_k,\\
&\tilde{R}_k(f^{i_k}) = \tilde{R}_{k-1}(f^{i_k}) - \gamma\langle \tilde{R}_{k-1}(f^{i_k}), \varphi_k\rangle\varphi_k,
\end{aligned} \tag{II.42}$$

where (i_k, φ_k) is chosen according to Algorithm 4.

As previously explained, we aim at applying Theorem 3.1 to the phantom residuals. This will be possible if we can show an analogue of Equations (II.31) (for the Residual L^2 norm) or (II.32) (for the $\mathcal{D}$-Correlation sum) and (II.33). Remark that, from Theorem 3.4, sparsity of both residuals $\tilde{R}_k(f)$ and $\hat{R}_k(f)$ does not exceed S with high probability if we choose γ small enough in Equation (II.34).

We begin the proof by recalling Lemma 2.2. In the multi-task case, this lemma can be easily extended as follows:

Lemma 3.3. *Assume that Hypotheses* $\mathbf{H}^{\mathbf{Mult}}_{\mathbf{dim}}$ *are fulfilled on* $\mathcal{D}$, f *and* ε, *with* $0 < \xi < 1$ *as given in* $\mathbf{H}^{\mathbf{Mult}}_{\mathbf{dim-2}}$, *then:*

(i) $\sup_{1\leq i,j\leq p_n} |\langle g_i, g_j\rangle_n - \langle g_i, g_j\rangle| = \zeta_{n,1} = \mathcal{O}_P(n^{-\xi/2})$,

(ii) $\sup_{1\leq i\leq p_n, 1\leq j\leq m_n} \left|\langle g_i, \varepsilon^j\rangle_n\right| = \zeta_{n,2} = \mathcal{O}_P(n^{-\xi/2})$,

(iii) $\sup_{1\leq i\leq m_n, 1\leq j\leq p_n} \left|\langle f^i, g_j\rangle_n - \langle f^i, g_j\rangle\right| = \zeta_{n,3} = \mathcal{O}_P(n^{-\xi/2})$.

(iv) $\sup_{1\leq i\leq m_n} \left|\left\|\varepsilon^i\right\|_n^2 - \mathbb{E}(|\varepsilon^i|^2)\right| = \zeta_{n,4} = \mathcal{O}_P(n^{-\xi/2})$.

Proof. The first three points of Lemma 3.3 are the same as (i), (ii) and (iii) of Lemma 2.2. The fourth point is something new, but the schema of its proof is close to the proof of (ii) of Lemma 2.2. Let $(K_n)_{n\in\mathbb{N}}$. We begin by considering the truncated variables:

$$\forall i \in [\![1, m_n]\!], \quad \varepsilon_k^{t_i} = \begin{cases} \varepsilon_k^i & \text{if } |\varepsilon_k| \leq K_n \\ sg(\varepsilon_k^i) K_n & \text{otherwise.} \end{cases}$$

For all $t > 0$, the following inequality holds:

$$\begin{aligned} \mathbb{P}\left(n^{\xi/2} \sup_{1\leq i\leq m_n} \left|\left\|\varepsilon^i\right\|_{\mathrm{n}}^2 - \mathbb{E}\left(\left|\varepsilon^i\right|^2\right)\right| > t\right) \quad \leq \quad & \mathbb{P}\left(n^{\xi/2} \sup_{1\leq i\leq m_n} \left|\left\|\varepsilon^{t_i}\right\|_{\mathrm{n}}^2 - \mathbb{E}\left(\left|\varepsilon^{t_i}\right|^2\right)\right| > t/3\right) \\ & + \mathbb{P}\left(n^{\xi/2} \sup_{1\leq i\leq m_n} \left|\left\|\varepsilon^i\right\|_{\mathrm{n}}^2 - \left\|\varepsilon^{t_i}\right\|_{\mathrm{n}}^2\right| > t/3\right) \\ & + \mathbb{P}\left(n^{\xi/2} \sup_{1\leq i\leq m_n} \left|\mathbb{E}\left(\left|\varepsilon^{t_i}\right|^2\right) - \mathbb{E}\left(\left|\varepsilon^i\right|^2\right)\right| > t/3\right) \\ & = I + II + III. \end{aligned}$$

For I, we apply Theorem 2.4 to random variables $\left(\left(\varepsilon_k^{t_i}\right)^2\right)_{1\leq k\leq n}$ which satisfy $\left\|\left(\varepsilon_k^{t_i}\right)^2\right\|_\infty \leq K_n^2$:

$$I \leq 2m_n \exp\left(-\frac{(t^2/9)n^{1-\xi}}{2\left(K_n^8 + K_n^2 t/3n^{-\xi/2}\right)}\right).$$

If we set $K_n = n^{\xi/4}$, we can then show that I becomes arbitrarily small when $n \to +\infty$ with Assumption $\mathbf{H}^{\mathbf{Mult}}_{\mathbf{dim-2}}$.

The second term can also be bounded as follows:

$$\begin{aligned} II \quad &\leq \quad \mathbb{P}\left(\exists 1 \leq k \leq n, \left|\varepsilon_k^i\right| > K_n\right) \\ &\leq \quad n\mathbb{P}\left(\left|\varepsilon_k^i\right| > K_n\right) \\ &\leq \quad nK_n^{-s}\mathbb{E}\left(\left|\varepsilon^i\right|^s\right), \end{aligned}$$

and we conclude that $II = \underset{n\to+\infty}{o}(1)$ with $s > 4/\xi$ (Assumption $\mathbf{H}^{\mathbf{Mult}}_{\mathbf{dim-3}}$).

For III, observe that:

$$\begin{aligned} \left|\mathbb{E}\left(\left(\varepsilon^i\right)^2 - \left(\varepsilon^{t_i}\right)^2\right)\right| \quad &\leq \quad \mathbb{E}\left(\left|\varepsilon^i\right|^4\right)^{1/2} \mathbb{P}\left(\left|\varepsilon^i\right| > K_n\right)^{1/2} + K_n^2 \mathbb{P}\left(\left|\varepsilon^i\right| > K_n\right) \\ &\leq \quad K_n^{-s/2}\mathbb{E}\left(\left|\varepsilon^i\right|^4\right)^{1/2} \mathbb{E}\left(\left|\varepsilon^i\right|^s\right)^{1/2} + K_n^{2-s}\mathbb{E}\left(\left|\varepsilon^i\right|^s\right). \end{aligned}$$

To conclude, remark that $s > 4$, which implies:

$$\left|\mathbb{E}\left(\left(\varepsilon^i\right)^2 - \left(\varepsilon^{t_i}\right)^2\right)\right| = o\left(K_n^{-2}\right) = o\left(n^{-\xi/2}\right),$$

and $III = 0$ for n large enough. □

Denoting $\zeta_n = \max\{\zeta_{n,1}, \zeta_{n,2}, \zeta_{n,3}, \zeta_{n,4}\} = \mathcal{O}_P(n^{-\xi/2})$, we can show that Lemma 2.3 is still true for the i_k-th coordinate of f. Moreover, let $i \neq i_k$. Since $\hat{R}_k(f^i) = \hat{R}_{k'}(f^i)$ for all $k' \leq k$ such that i_k is not selected between step k' and k (see Equation (II.34)), we can easily extend Lemma 2.3 to each coordinate of f:

$$\sup_{1\leq i\leq m_n} \sup_{1\leq j\leq p_n} \left|\langle \hat{R}_k(f^i), g_j\rangle_{\mathrm{n}} - \langle \tilde{R}_k(f^i), g_j\rangle\right| \leq C\left(\frac{5}{2}\right)^k \zeta_n. \tag{II.43}$$

Using this extension of Lemma 2.2, the same calculations detailed in Section 2.4.2 can be done. Hence, considering the i_k-th coordinate of f chosen by Equations (II.31) or (II.32), on the set Ω_n, inequality (II.16) also holds:

$$\left|\langle \tilde{R}(f^{i_k}), \varphi_k\rangle\right| \geq \frac{1}{2} \sup_{1\leq j\leq p_n} \left|\langle \tilde{R}_{k-1}(f^{i_k}), g_j\rangle\right|.$$

Consider now the Boost-Boost Residual L^2 norm algorithm. To obtain an analogue of (II.31), we need the following lemma, which compares the norms of both residuals:

Lemma 3.4. *Under Hypotheses* $\mathbf{H}^{\mathbf{Mult}}_{\mathbf{dim}}$*, a constant* $0 < C < +\infty$ *exists, independent of* n *and* k*, so that on the set* $\Omega_n = \{\omega, |\zeta_n(\omega)| < 1/2\}$*:*

$$\sup_{1\leq i\leq m_n} \left|\left\|\hat{R}_{k-1}(f^i)\right\|_n^2 - \left\|\tilde{R}_{k-1}(f^i)\right\|_n^2\right| \leq C\left(2\left(\frac{5}{2}\right)^{k-1} + S\right) S\zeta_n.$$

Proof. Consider the two residual sequences $(\hat{R}_k(f))_{k\geq 0}$ and $(\tilde{R}_k(f))_{k\geq 0}$, expanded on $\mathcal{D}$ as: $\hat{R}_{k-1}(f^i) = \sum_j \theta^k_{i,j} g_j$, and $\tilde{R}_{k-1}(f^i) = \sum_j \tilde{\theta}^k_{i,j} g_j$. Hence,

$$\left|\left\|\hat{R}_{k-1}(f^i)\right\|_{\mathrm{n}}^2 - \left\|\tilde{R}_{k-1}(f^i)\right\|_{\mathrm{n}}^2\right| \leq \underbrace{\left|\sum_{j=1}^{p_n} \theta^k_{i,j}\left(\langle \hat{R}_{k-1}(f^i), g_j\rangle_{\mathrm{n}} - \langle \tilde{R}_{k-1}(f^i), g_j\rangle\right)\right|}_{I}$$

$$+\underbrace{\left|\sum_{j=1}^{p_n} \tilde{\theta}^k_{i,j}\left(\langle \hat{R}_{k-1}(f^i), g_j\rangle_{\mathrm{n}} - \langle \tilde{R}_{k-1}(f^i), g_j\rangle\right)\right|}_{II} + \underbrace{\left|\sum_{j=1}^{p_n} \theta^k_{i,j}\langle \tilde{R}_{k-1}(f^i), g_j\rangle - \sum_{j=1}^{S} \tilde{\theta}^k_{i,j}\langle \hat{R}_{k-1}(f^i), g_j\rangle_{\mathrm{n}}\right|}_{III}.$$

By Equation (II.43), we can provide two upper bounds for I and II:

$$I \leq C\left(\frac{5}{2}\right)^{k-1} \sum_{j=1}^{p_n} \left|\theta^k_{i,j}\right| \zeta_n \quad \text{and} \quad II \leq C\left(\frac{5}{2}\right)^{k-1} \sum_{j=1}^{p_n} \left|\tilde{\theta}^k_{i,j}\right| \zeta_n.$$

Denoting $M := \max_{1\leq j\leq S}\left\{\left|\theta^k_{i,j}\right|, \left|\tilde{\theta}^k_{i,j}\right|\right\}$, the following inequality holds for I and II:

$$I \vee II \leq CMS\left(\frac{5}{2}\right)^{k-1} \zeta_n.$$

To conclude, using Lemma (3.3), we have:

$$III \leq \sum_{j=1}^{p_n} \left|\tilde{a}^k_{i,j}\right| \sum_{j'=1}^{p_n} \left|a^k_{i,j}\right| \left|\langle g_j, g_{j'}\rangle - \langle g_j, g_{j'}\rangle_{\mathrm{n}}\right| \leq S^2 M^2 \zeta_n.$$

and the conclusion follows using our last bounds. □

Since Lemma 3.4 is not directly applicable to the observed residual $Y - \hat{G}_k(f)$, the same calculation cannot be performed to obtain an analogue of Equation (II.31). However, we can compare the norm of the theoretical and observed residuals:

$$\begin{aligned}\sup_{1\le i\le m_n} \left\|Y^i - \hat{G}_{k-1}(f^i)\right\|_{\mathrm{n}}^2 &= \left\|\hat{R}_{k-1}(f^i) + \varepsilon^i\right\|_{\mathrm{n}}^2 \\ &= \left\|\hat{R}_{k-1}(f^i)\right\|_{\mathrm{n}}^2 + \left\|\varepsilon^i\right\|_{\mathrm{n}}^2 + 2\langle \hat{R}_{k-1}(f^i), \varepsilon^i\rangle_{\mathrm{n}}.\end{aligned}$$

Note that, using Lemma 3.3, we obtain: $\left|\langle \hat{R}_k(f^i), \varepsilon^i\rangle_{\mathrm{n}}\right| \le MS\zeta_n$, where M is defined in the proof of Lemma 3.4. Hence, we have for all i:

$$\left\|\hat{R}_{k-1}(f^i)\right\|_{\mathrm{n}}^2 + \left\|\varepsilon^i\right\|_{\mathrm{n}}^2 - 2MS\zeta_n \le \left\|Y^i - \hat{G}_{k-1}(f^i)\right\|_{\mathrm{n}}^2 \le \left\|\hat{R}_{k-1}(f^i)\right\|_{\mathrm{n}}^2 + \left\|\varepsilon^i\right\|_{\mathrm{n}}^2 + 2MS\zeta_n. \quad \text{(II.44)}$$

It should be recalled that $\mathbb{E}(|\varepsilon^i|^2)$ does not depend on i from Assumption $\mathbf{H}^{\mathbf{Mult}}_{\mathbf{dim-3}}$, and is denoted by σ^2. Then, an application of Lemma 3.3 (iv) to Equation (II.44) yields

$$\left\|\hat{R}_{k-1}(f^i)\right\|_{\mathrm{n}}^2 + \sigma^2 - (1+2MS)\zeta_n \le \left\|Y^i - \hat{G}_{k-1}(f^i)\right\|_{\mathrm{n}}^2 \le \left\|\hat{R}_{k-1}(f^i)\right\|_{\mathrm{n}}^2 + \sigma^2 + (1+2MS)\zeta_n. \quad \text{(II.45)}$$

Hence, on Ω_n, by definition of i_k, Equation (II.45) and Lemma 3.4, we can write:

$$\begin{aligned}\left\|Y^{i_k} - \hat{G}_{k-1}(f^{i_k})\right\|_{\mathrm{n}}^2 &\ge \sup_{1\le i\le m_n} \left\|Y^i - \hat{G}_{k-1}(f^i)\right\|_{\mathrm{n}}^2 \\ &\ge \sup_{1\le i\le m_n} \left\{\left\|\hat{R}_{k-1}(f^i)\right\|_{\mathrm{n}}^2 + \sigma^2\right\} - (1+2MS)\zeta_n \\ &\ge \sup_{1\le i\le m_n} \left\{\left\|\tilde{R}_{k-1}(f^i)\right\|^2 + \sigma^2\right\} - C\left(2\left(\frac{5}{2}\right)^{k-1} + S\right) S\zeta_n \\ &\quad -(1+2MS)\zeta_n. \qquad \text{(II.46)}\end{aligned}$$

Using the same calculus on the set Ω_n once again:

$$\begin{aligned}\left\|\tilde{R}_{k-1}(f^{i_k})\right\|^2 &\ge \left\|\hat{R}_{k-1}(f^{i_k})\right\|_{\mathrm{n}}^2 - C\left(2\left(\frac{5}{2}\right)^{k-1} + S\right) S\zeta_n \\ &\ge \left\|Y^{i_k} - \hat{G}_{k-1}(f^{i_k})\right\|_{\mathrm{n}}^2 - \sigma^2 - (1+2MS)\zeta_n - C\left(2\left(\frac{5}{2}\right)^{k-1} + S\right) S\zeta_n \\ &\ge \sup_{1\le i\le m_n} \left\{\left\|\tilde{R}_{k-1}(f^i)\right\|^2 + \sigma^2\right\} - \sigma^2 - 2(1+2MS)\zeta_n \\ &\quad -2C\left(2\left(\frac{5}{2}\right)^{k-1} + S\right) S\zeta_n, \quad \text{by Equation (II.46).} \qquad \text{(II.47)}\end{aligned}$$

We then obtain from Equation (II.47) that:

$$\left\|\tilde{R}_{k-1}(f^{i_k})\right\|^2 \ge \sup_{1\le i\le m_n} \left\|\tilde{R}_{k-1}(f^i)\right\|^2 - 2(1+2MS)\zeta_n - 2C\left(2\left(\frac{5}{2}\right)^{k-1} + S\right) S\zeta_n. \quad \text{(II.48)}$$

Let $\check{\Omega}_n^1 = \left\{\omega, \quad \forall k \leq k_n \quad \sup_{1\leq i\leq m_n} \left\|\tilde{R}_{k-1}(f^i)\right\|^2 > 4\left(1+2MS+C\left(2\left(\frac{5}{2}\right)^{k-1}+S\right)S\right)\zeta_n\right\}$. We deduce from Equation (II.48) the following inequality on set $\Omega_n \cap \check{\Omega}_n^1$:

$$\left\|\tilde{R}_{k-1}(f^{i_k})\right\|^2 \geq \frac{1}{2}\sup_{1\leq i\leq m_n}\left\|\tilde{R}_{k-1}(f^i)\right\|^2.$$

Finally, consider the Boost-Boost $\mathcal{D}$-Correlation sum algorithm. To obtain an analogue of Equation (II.32), the following lemma is needed:

Lemma 3.5. *Under Hypotheses* $\mathbf{H}_{\mathbf{dim}}^{\mathbf{Mult}}$*, a constant* $0 < C < +\infty$ *exists, independent of* n *and* k *so that, on the set* $\Omega_n = \{\omega, |\zeta_n(\omega)| < 1/2\}$*:*

$$\sup_{1\leq i\leq m}\sup_{1\leq j\leq p_n}\left|\langle\hat{R}_k(f^i), g_j\rangle_n^2 - \langle\tilde{R}_k(f^i), g_j\rangle^2\right| \leq C\left(\frac{5}{2}\right)^{2k}\zeta_n.$$

Proof. Let $k \geq 1$, $i \in [\![1, m_n]\!]$. We have the following equality:

$$\begin{aligned}&\left|\langle\hat{R}_k(f^i), g_j\rangle_{\mathrm{n}}^2 - \langle\tilde{R}_k(f^i), g_j\rangle^2\right| \\ &\qquad = \left|\langle\hat{R}_k(f^i), g_j\rangle_{\mathrm{n}} - \langle\tilde{R}_k(f^i), g_j\rangle\right|\left|\langle\hat{R}_k(f^i), g_j\rangle_{\mathrm{n}} + \langle\tilde{R}_k(f^i), g_j\rangle\right|, \quad \text{(II.49)}\end{aligned}$$

where $\left|\langle\hat{R}_k(f^i), g_j\rangle_{\mathrm{n}} - \langle\tilde{R}_k(f^i), g_j\rangle\right| \leq C\left(\frac{5}{2}\right)^k\zeta_n$ by Equation (II.43).

Moreover, using the recursive equation for $(\hat{R}_k(f^{i_k}))_k$, we can obtain the following bounds:

$$\begin{aligned}\left|\langle\hat{R}_k(f^{i_k}), g_j\rangle_{\mathrm{n}}\right| &\leq \left|\langle\hat{R}_{k-1}(f^{i_k}), g_j\rangle_{\mathrm{n}}\right| + \gamma\left|\langle\hat{R}_{k-1}(f^{i_k}), \varphi_k\rangle_{\mathrm{n}}\langle\varphi_k, g_j\rangle_{\mathrm{n}}\right| \\ &\quad +\gamma\left|\langle\varepsilon^{i_k}, \varphi_k\rangle_{\mathrm{n}}\langle g_j, \varphi_k\rangle_{\mathrm{n}}\right| \\ &\leq \sup_{1\leq j\leq p_n}\left|\langle\hat{R}_{k-1}(f^{i_k}), g_j\rangle_{\mathrm{n}}\right|(1+\gamma\left|\langle\varphi_k, g_j\rangle_{\mathrm{n}}\right|) + \gamma\zeta_n(1+\zeta_n) \\ &\leq M_{k-1}^{i_k}(1+\gamma(1+\zeta_n)) + \gamma\zeta_n(1+\zeta_n),\end{aligned}$$

where $M_k^i := \sup_{1\leq j\leq p_n}\left|\langle\hat{R}_k(f^i), g_j\rangle_{\mathrm{n}}\right|$. Note that for $i \neq i_k$, $M_k^i = M_{k-1}^i$.

On Ω_n, we hence have for a suitable constant $C > 0$:

$$M_k^i \leq M_{k-1}^i\left(1+\frac{3}{2}\gamma\right) + C\ldots \leq \left(1+\frac{3}{2}\gamma\right)^k\left(\sup_{n\in\mathbb{N}}\sum_{j=1}^{p_n}|a_{i,j}| + \frac{3}{2}\right) + C. \quad \text{(II.50)}$$

By Equation (II.13), $\left\|\tilde{R}_k(f^i)\right\|$ is non-increasing. Hence $\left\|\tilde{R}_k(f^i)\right\| \leq \left\|f^i\right\|$. The Cauchy-Schwarz inequality allows us to write that:

$$\left|\langle\tilde{R}_k(f^i), g_j\rangle\right| \leq \left\|\tilde{R}_k(f^i)\right\| \leq \left\|f^i\right\|. \quad \text{(II.51)}$$

The conclusion therefore holds using Equations (II.50) and (II.51) in Equation (II.49) for a large enough constant C. □

Observe that Lemma 3.5 remains true if we change the observed residual by the theoretical residual. Hence, on the set Ω_n,

$$\begin{aligned}\sum_{j=1}^{p_n}\left|\langle Y^{i_k}-\hat{G}_{k-1}(f^{i_k}),g_j\rangle_{\mathrm{n}}\right|^2 &\geq \sup_{1\leq i\leq m_n}\sum_{j=1}^{p_n}\left|\langle Y^{i}-\hat{G}_{k-1}(f^{i}),g_j\rangle_{\mathrm{n}}\right|^2\\ &\geq \sup_{1\leq i\leq m_n}\sum_{j=1}^{p_n}\left(\left|\langle \tilde{R}_{k-1}(f^{i}),g_j\rangle_{\mathrm{n}}\right|^2-C\left(\frac{5}{2}\right)^{2(k-1)}\zeta_n\right)\\ &\geq \sup_{1\leq i\leq m_n}\sum_{j=1}^{p_n}\left|\langle \tilde{R}_{k-1}(f^{i}),g_j\rangle_{\mathrm{n}}\right|^2-Cp_n\left(\frac{5}{2}\right)^{2(k-1)}\zeta_n. \quad \text{(II.52)}\end{aligned}$$

Therefore, using Lemma 3.5 again, on Ω_n:

$$\begin{aligned}\sum_{j=1}^{p_n}\left|\langle \tilde{R}_{k-1}(f^{i_k}),g_j\rangle\right|^2 &\geq \sum_{j=1}^{p_n}\left|\langle Y^{i_k}-\hat{G}_{k-1}(f^{i_k}),g_j\rangle_{\mathrm{n}}\right|^2-Cp_n\left(\frac{5}{2}\right)^{2(k-1)}\zeta_n\\ &\geq \sup_{1\leq i\leq m_n}\sum_{j=1}^{p_n}\left|\langle \tilde{R}_{k-1}(f^{i}),g_j\rangle\right|^2-2Cp_n\left(\frac{5}{2}\right)^{2(k-1)}\zeta_n, \quad \text{(II.53)}\end{aligned}$$

by Equation (II.52).

Let $\check{\Omega}_n^2=\left\{\omega,\ \ \forall k\leq k_n \quad \sup_{1\leq i\leq m_n}\sum_{j=1}^{p_n}\left|\langle \tilde{R}_{k-1}(f^{i}),g_j\rangle\right|^2>4Cp_n\left(\frac{5}{2}\right)^{2(k-1)}\zeta_n\right\}$. We deduce from Equation (II.53) the following inequality on $\Omega_n\cap\check{\Omega}_n^2$:

$$\sum_{j=1}^{p_n}\left|\langle \tilde{R}_{k-1}(f^{i_k}),g_j\rangle\right|^2\geq\frac{1}{2}\sup_{1\leq i\leq m_n}\sum_{j=1}^{p_n}\left|\langle \tilde{R}_{k-1}(f^{i}),g_j\rangle\right|^2.$$

Consequently, on $\Omega_n\cap\tilde{\Omega}_n\cap\check{\Omega}_n^1$ and $\Omega_n\cap\tilde{\Omega}_n\cap\check{\Omega}_n^2$, we can apply Theorem 3.1 to family $(\tilde{R}_k(f^i))_k$, since it satisfies a deterministic Boost-Boost algorithm with constants $\tilde{\mu}=1/2$, $\tilde{\nu}=1/2$, and has a bounded sparsity S.

Let us now consider the set $\left(\check{\Omega}_n^2\right)^C$. Using Equation (II.39), we obtain

$$\left\|\tilde{R}_k(f^i)\right\|^2\leq\frac{\lambda_{max}}{\lambda_{min}^2}\sum_{j=1}^{p_n}\left|\langle \tilde{R}_k(f^{i}),g_j\rangle\right|^2\leq 4\frac{\lambda_{max}}{\lambda_{min}^2}Cp_n\left(\frac{5}{2}\right)^{2k}\zeta_n.$$

On the set $\left(\check{\Omega}_n^1\right)^C$, we also have:

$$\left\|\tilde{R}_k(f^i)\right\|^2\leq 4\left(1+2MS+C\left(2\left(\frac{5}{2}\right)^k+S\right)S\right)\zeta_n.$$

The end of the proof follows as in Section 2.4.2 by noting that:

$$\mathbb{P}\left((\Omega_n\cap\tilde{\Omega}_n)\cup\tilde{\Omega}_n^C\cup(\check{\Omega}_n^{1,2})^C\right)\geq\mathbb{P}(\Omega_n)\underset{n\to+\infty}{\longrightarrow}1.$$

Remark that the conclusion holds for a sequence k_n that grows sufficiently slowly: for the Boost-Boost Residual L^2 norm algorithm, k_n is allowed to grow as $(\xi/4\log(3))\log(n)$, whereas k_n can only grow as $(\xi/8\log(3))\log(n)$ for the Boost-Boost $\mathcal{D}$-Correlation sum algorithm.

4 Numerical applications

This section is dedicated to simulation studies to assess the practical performances of our method. We compare it to existing methods, namely the Bootstrap Lasso [Bac08], Random Forests [Bre01] and the recently proposed remMap [PZB$^+$10]. The aim of these applications is twofold. Firstly, we assess the performance of our algorithms in light of expected theoretical results and as compared to other state-of-the-art methods. Secondly, we demonstrate the ability of our algorithm to analyse datasets, that have features encountered in real situations. Three types of data sets are used. The two first types are challenging multivariate, noisy, linear datasets with different characteristics, either uni-dimensional or multi-dimensional. The third type consists in a simulated dataset that mimics the behaviour of a complex biological system through observed measurements.

First, we introduce the numerically-driven stopping criterion we used for our algorithms. Then, we briefly present the competing methods and the criteria we used to assess the merits of the different methods. Finally, we discuss the obtained results for each datasets we used. For the sake of convenience, we will shortcut the notation p_n to p as well as m_n to m in the sequel.

4.1 Stopping criterion

An important issue when implementing a Boosting method, or any other model estimation procedure from a dataset, is linked to the definition of a stopping rule. It ideally guarantees that the algorithm ran long enough to provide informative conclusions without over-fitting the data. Cross-Validation (CV) or informative criteria, such as AIC or BIC address this issue. In the univariate case, implementing an AIC criterion seems to be very computationaly attractive. In the multivariate setting, the AIC criterion is defined as follows for a selected sub-model M_d, with $d \leq p$ covariates:

$$AIC(M_d) = \log\left(\hat{\sigma}^2(M_d)\right) + \frac{2md}{n},$$

where $\hat{\sigma}^2(M_d)$ is the maximum likelihood estimator of the error covariance matrix.

To apply AIC for the Boost-Boost algorithm, we have to determine the number of parameters, or effective degrees of freedom, of the current approximation as a function of the number of iterations. One solution consists in defining d as the number of non-zeros estimated coefficients of $(a_{i,j})_{1\leq i\leq m, 1\leq j\leq p}$. However, as pointed out by [HTF09], many regression methods are based on regularized least squares and so degrees of freedom defined in terms of dimensionality are not useful. In the general case, the number of degrees of freedom is equal to the trace of a hat-matrix $\hat{H}$ (see [HTF09] or [BY03]). $\hat{H}$ is defined as the operator that enables the estimation $\hat{G}(f)$ from the true parameter f only. Consider the linear model $Y = X\beta + \varepsilon$, then, the hat-matrix is defined as the matrix that maps the vector of observed values Y to the vector of fitted values, denoted $X\hat{\beta}$:

$$\hat{H}Y = X\hat{\beta}.$$

For the least-square estimation, $\hat{\beta} = ({}^tXX)^{-1\,t}XY$ and the hat-matrix $\hat{H}$ satisfies:

$$\hat{H} = X({}^tXX)^{-1\,t}X.$$

Denote $\hat{H}_k$ the number of degrees of freedom for the Boost-Boost algorithms at step k of the algorithm. To obtain the degrees of freedom per response variable, we divide the total number of degrees of freedom by m. The AIC criterion then becomes:

$$AIC(k) = \log\left(\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{G}_k(f_i)\right)^t\left(Y_i - \hat{G}_k(f_i)\right)\right) + \frac{2\,\mathrm{trace}(\hat{H}_k)}{n}.$$

However, as pointed by [LB06], the computation of the hat matrix at step k has a complexity of $\mathcal{O}(n^2p + n^3m^2k)$, and thus becomes not feasible if n, p or m are too large. For example, the computation of the hat matrix at the initialization of the algorithm (iteration $k = 1$) with $n = 100$ and $p = m = 250$ requires $6 \cdot 10^8$ operations, which takes around 7 seconds on an actual standard computer. Consequently, a typical run of the algorithm requires hundreds of iterations, which would last almost 10 hours, this does not look reasonnable for practical purpose.

We hence choose to use 5-fold cross-validation to assess the optimal number of iterations. Finally, it should be noted that cross-validation should be carefully performed, as pointed out by the erratum of [GWBV02]. It is imperative not to use the same dataset to both optimise the feature space and compute the prediction error. We refer the interested reader to the former erratum of Guyon and several comments detailed in [AM02].

Remark that, in our simulation study, the cross-validation error E_{CV} decreases along the step of the Boosting algorithm while new variables are added in the model (see for instance Figure II.1). The selected model was the one estimated after the first iteration that made the ratio of the total variation in cross-validation error:

$$\left| \frac{E_{CV} - E_{min}}{E_{max} - E_{min}} \right|,$$

where E_{max} and E_{min} are the maximal and the minimal values of the cross-validation error, below a 5% threshold.

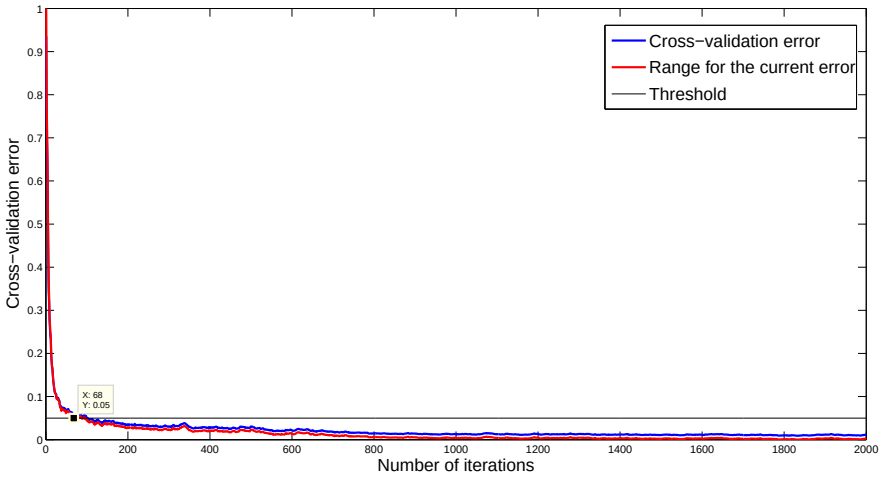


Figure II.1: The cross-validation error (in blue) and the range $|(E_{CV} - E_{min})/(E_{max} - E_{min})|$ (in red) along the iterations of the Boosting algorithm. The threshold, used to define our stopping criterion, is represented in black.

4.2 Calibration of parameters

Two parameters are required for the Boosting algorithm. The calibration of the maximal number of iterations seems very significant. As pointed in Figure II.2, if k_{up} is too small, we will never choose the "best" estimator. However, choosing a large k_{up} makes the algorithm slower, without improving the numerical results: the estimated number of iterations is quite the same.

The maximal number of iterations is then chosen such that the number of iterations estimated by cross validation no longer changes.

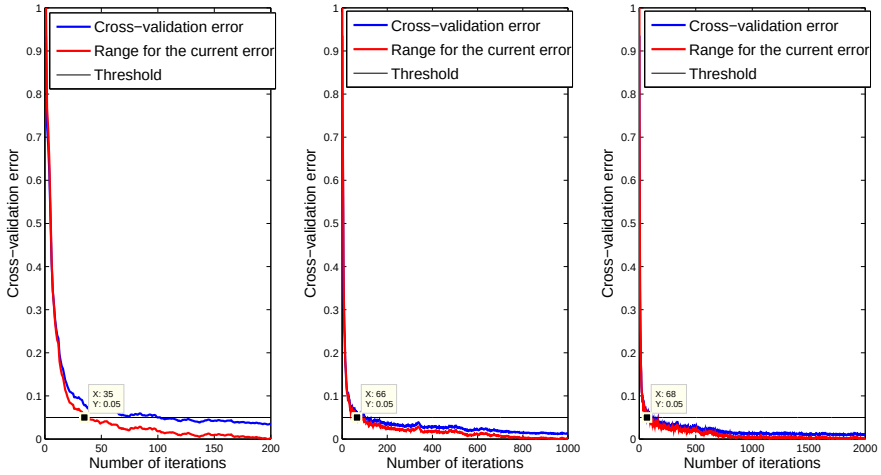


Figure II.2: The cross-validation error (in blue) and the range $|(E_{CV} - E_{min})/(E_{max} - E_{min})|$ (in red) along the iterations of the Boosting algorithm for three maximal numbers of iterations $k_{up} = 200$, 1000 and 2000. The threshold, used to define our stopping criterion, is represented in black. The maximal number of iterations is selected such that the estimated number of iterations no longer changes.

The second parameter is the shrinkage parameter γ. Remark that small step size also makes the Boosting algorithm slower and requires a larger maximal number of number of iterations. However, the computational cost seems to be advantageous compared to the performances. We hence set γ to 0.2.

4.3 Algorithms and methods

We used our two proposed Boost-Boost algorithms (denoted "$\mathcal{D}$-Corr" for the Boost-Boost $\mathcal{D}$-Correlation sum algorithm and "L^2 norm" for the Boost-Boost Residual L^2 norm algorithm) with a shrinkage factor $\gamma = 0.2$. When the number of responses m is set to 1, these two algorithms are similar to Algorithm 2 and will both be referred to as "WGA".

We compared them to a bootstrapped version of the Lasso, denoted "BootLasso" thereafter. The idea of this algorithm is essentially that of the algorithm proposed by Bach [Bac08]: it uses bootstrapped estimates of the active regression set based on a Lasso penalty. In [Bac08], only variables that are selected in every bootstrap are kept in the model, and actual coefficient values are estimated from a straightforward least square procedure. Due to high-dimensional settings and empirical observations, we slightly relaxed the condition for a variable to be selected: at a given penalty level, the procedure keeps a variable if more than 80% of bootstrapped samples lead to select it in the model. We computed a 5-fold cross-validation unknown parameter estimates. The R package `glmnet` v1.9 − 5 was used for the BootLasso simulations.

The second approach we used is a random forest algorithm [Bre01] in regression, known to be suited to reveal interactions in a dataset, denoted as "RForests". It consists in a set (the forest) of regression trees. The randomisation is combined into 'bagging' of samples and random selection of feature sets at each split in every tree. For each regression, predictors are ranked according

to their importance, that computes the squared error loss when using a shuffled version of the variable instead of the original one. We filtered for variables that have a negative importance. Such variables are highly non-informative since shuffling their sample values leads to an increased prediction accuracy; this can happen for small sample sizes or if terminal leaves are not pruned at the end of the tree-building process. No stopping criterion is implemented since it would require storing all partial depth trees of the forest and would be very memory-consuming. However, in each forest, we artificially introduced a random variable made up of a random blend of values observed on any variable in the data for each sample. The rationale is that any variable that achieves a lower importance than this random variable is not informative and should be discarded from the model. For each forest, we repeated this random variable inclusion a hundred times. We selected a variable if its importance was at least 85 times out of 100 higher than that of the artificially introduced random variable, their importance could serve to rank them. We also computed a final prediction $\mathbb{L}_2$-error for the whole forest and model selection metrics associated with correctly predicted relationships. The R package `randomForest` v$4.6-7$ was used for the RForests simulations. Notice that the total running time for RForests is linear in the size of the output variables. Hence, when $m = 250$ (correlated covariates or correlated noise), the total running time is nearly four days. We hence present partial results in these two cases on a very limited number of networks (5).

Finally, we compared our method to "remMap" (REgularized Multivariate regression for identifying MAster Predictors) that essentially produces sparse models with several very important regulatory variables in a high-dimensional setting. We refer to it as REM later in the paper. More specifically, REM uses an $\mathbb{L}_1$-norm penalty to control the overall sparsity of the coefficient matrix of the multivariate linear regression model. In addition, REM imposes a "group sparse" penalty, which is pasted from the group lasso penalty [YL06]. This penalty puts a constraint on the $\mathbb{L}_2$ norm of regression coefficients for each predictor, which controls the total number of predictors entering the model and consequently facilitates the detection of so-called master predictors. We used the R package `remMap` v$0.1-0$ in our simulations. Parameter tuning was performed using the built-in cross-validation function. We varied parameters for DS1 and DS3 from 10^{-5} to 10^5 with a 10-fold multiplicative increment; for DS2, DS4 and DREAM datasets, the package could only run with parameters varying from 10^{-2} to 10^2. Lastly, in the very high-dimensional settings of our scenario ($p = m = 250$), the built-in cross-validation function of the `remMap` package wouldn't allow us to visit parameters outside the range 10^{-1} to 10^1, with over 24 hours of computation per network.

4.4 Numerical results

Performance measurments The performances are first measured through the normalised prediction error, also known as the mean square error:

$$MSE = \left\| Y - \hat{G}_{\hat{k}}(f) \right\|_{\mathrm{n}}^2,$$

where $\hat{G}_{\hat{k}}(f)$ denotes the approximation of f at the end of the Boosting algorithm.

To measure the support recovery of the estimator, we also report the rate of coefficients inferred by mistake, the false positives, denoted FP thereafter, and not detected, the false negatives, denoted FN thereafter.

First dataset We use two toy examples in both univariate ($m = 1$) and multi-task ($m = 5$) situations, with noisy linear datasets with different characteristics. They are simulated according

to a linear modelling:

$$Y = XA + \varepsilon = f(X) + \varepsilon,$$

where Y is a $n \times m$ response matrix, X is a $n \times p$ observation matrix, ε is an additional Gaussian noise and A is the $p \times m$ S-sparse parameter matrix that encodes relationships to be inferred. Covariates are generated according to a multi-variate Gaussian distribution $\forall i, X_i \sim \mathcal{N}(0, I_p)$. Errors are generated according to a multi-variate normal distribution with an identity covariance matrix. Non-zero A-coefficients are set equal to 10 when $(p, m, S) = (250, 1, 5)$ and 1 for all other datasets.

In all our simulations, we always generate $n = 100$ observations; this situation corresponds to either moderate or very high-dimensional settings depending on the number of explanatory variables (p) or on the number of response variables (m). Unless otherwise stated, all experiments are replicated 100 times and results are averaged over these replicates.

Prediction performances of tested methods are detailed in Table II.1. In the first three simulation settings, when $m = 1$, the prediction performances of the Boosting algorithms are quite similar to those of the BootLasso and RF ones (see Table II.1), but when the number of predictors is set to $1,000$, BootLasso results are poorer. REM seems to achieve a better prediction than other approaches, especially in the very high-dimensional setting ($p = 1,000$ while $m = 1$). This is still the case when $p = 250$ and $m = 5$ or 250.

(p,m,S)	(250,1,5)	(250,1,10)	(1000,1,20)	(250,5,50)	(250,250,1250)
WGA	0.21	0.23	0.42	$\emptyset$	$\emptyset$
$\mathcal{D}$-Corr	$\emptyset$	$\emptyset$	$\emptyset$	0.39	0.36
L^2 norm	$\emptyset$	$\emptyset$	$\emptyset$	0.40	0.38
BootLasso	0.30	0.28	0.78	0.31	0.40
RForests	0.18	0.25	0.49	0.41	0.20*
REM	0.33	0.18	0.08	0.21	0.19

Table II.1: First dataset: MSE for the Boosting algorithms, with a shrinkage factor $\gamma = 0.2$, compared to the BootLasso, RForests and REM; the sample size n is set to 100. (*: for five simulated replicate datasets only as the running time for RForest was four days per network).

(p,m,S)	(250,1,5)		(250,1,10)		(1000,1,20)		(250,5,50)		(250,250,1250)	
	FP	FN	FP	FN	FP	FN	FP	FN	FP	FN
WGA	0.00	0.00	0.43	0.10	0.62	41.5	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\mathcal{D}$-Corr	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	0.84	3.42	0.10	0.65
L^2 norm	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	0.85	4.68	0.09	0.73
BootLasso	0.00	19.00	0.03	30.70	0.00	89.25	0.10	31.80	0.00	32.03
RForests	2.10	0.20	3.67	23.10	1.01	60.25	3.29	32.02	2.47*	2.76*
REM	0.58	0.00	1.49	0.00	5.53	6.65	2.66	0.00	2.35	0.00

Table II.2: First dataset: Percentage of false positive FP coefficients and false negative FN coefficients for the Boosting algorithms, with a shrinkage factor $\gamma = 0.2$, compared to the BootLasso, RForests and REM; the sample size n is set to 100. (*: for five simulated replicate datasets only as the running time for RForest was four days per network)

Looking at the accuracy results of Table II.2 at the same time is instructive: neither BootLasso nor RF succeed at recovering the structure of f, with the FN rate much higher than that of

the $\mathbb{L}_2$-Boosting and REM approaches. In the moderately high-dimensional univariate setting $(p, m) = (250, 1)$, WGA and REM almost always recover the full model with few FP, while BootLasso and RF miss one third and one fourth of the correct edges, respectively. Figures in the high-dimensional univariate case $(p, m) = (1, 000, 1)$ confirm this trend with a better precision for WGA, whereas REM achieves a better recall. This probably explains the much lower MSE for REM: the model selected in the REM framework is much richer and contains the vast majority of relevant relationships at the price of a low precision (just below 30%). In contrast, the model built by WGA is sparser with fewer FP, but misses some correct relationships. We therefore empirically observe here that MSE is not too informative for feature selection, as reported by [HTF09], for example. The conclusion we can draw follow the same tendency in the high-dimensional multivariate settings $(p, m) = (250, 5)$ and $(p, m) = (250, 250)$. Again, REM is more comprehensive in retrieving actual edges, but it produces much more FP relationships than the multivariate boosting algorithms we presented.

In addition to the performance value, Figure II.3 represents the norm of each coordinate of the residual along the iterations of the Boost-Boost $\mathcal{D}$-Correlation sum algorithm when the number of predictors p is equal to 250 and the number of responses m is equal to 5 (A then includes $S = 50$ non-zero coefficients). Figure II.3 shows that no residual coordinate is preferred along the iterations of the Boost-Boost $\mathcal{D}$-Correlation sum algorithm.

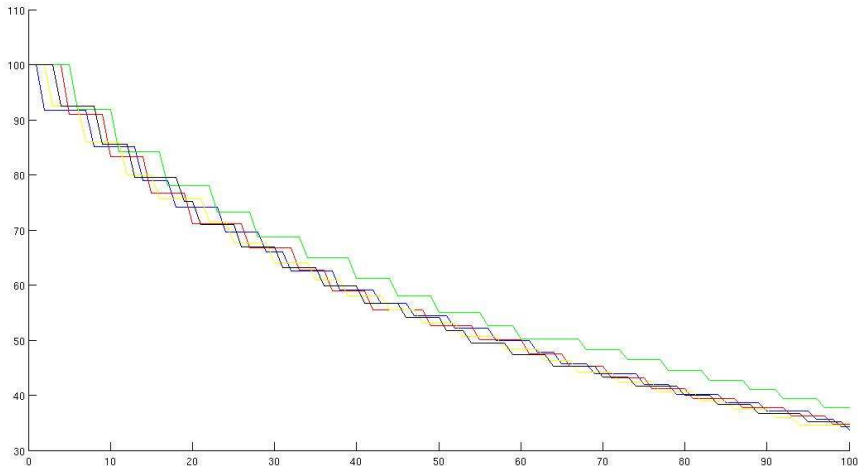


Figure II.3: First dataset $(p, m, S) = (250, 5, 50)$: Norm of each coordinate of residuals along the first 100 iterations of the Boost-Boost $\mathcal{D}$-Correlation sum algorithm; the sample size n is set to 100.

Second dataset The following dataset stands for a more extreme situation. It is specifically designed to illustrate the theoretical results we presented on permissive sparsity and the lower bound of regression parameters. The idea is to consider a column-wise unbalanced design with highly correlated predictors or highly correlated noise coordinates (correlations can be as strong as ± 0.9). More precisely, we generate the second dataset with $p = 250$ and $m = 250$ as follows. For the first task (first column of X), we fix 10 non-zero coefficients and set their value to 500. For each task from 2 to 241, we choose 10 coefficients and set their value to 1. The last 9 columns have respectively 9, 8,... 1 non-zero coefficients, which are also set to 1. At last, we first generate

in the first case some high correlations among covariates according to a multivariate Gaussian distribution with covariance matrix V so that $V_i^j = 0.9(-1)^{|i-j|}$. Then, we also generate some high correlations among the error terms according to the same multivariate Gaussian distribution with covariance V . Table II.3 shows performances of the proposed algorithms on this dataset.

Assumption $\mathbf{H_{SNR}}$ may not be fulfilled here, but we are interested in the robustness of the studied Boost-Boost algorithms in such a scenario. Results indicate that the Boost-Boost $\mathcal{D}$-Correlation sum algorithm and REM perform better overall. Their overall recall is quite poor (about $71.26 - 75.60\%$ of FN elements for REM and $74.20 - 83.36\%$ for the Boosting algorithm). REM includes more irrelevant regressors in the model (with a rate of $4.38 - 7.07\%$ of FP elements for Boosting algorithms and $5.34 - 14.33\%$ for REM), probably because of the very high correlation levels between predictors or because of the intricate correlated noise we artifically added to the data. The latter seems indeed to be an even more challenging obstacle here. We recall here that in these 2 scenarii, a 1% in FP rate implies a difference of just over 600 falsely predicted edges. The algorithms we proposed were designed to deal partly with the correlation between responses when it's not too high and when the noise is not too high neither. It seems here that the correlated noise is a more difficult situation to tackle, perhaps only because of the choice we made to simulate it. The overall low recalls (or high FN rates) can be explained by the highly unbalanced design between columns as well. Moreover, Boosting algorithms and REM identify much richer models than BootLasso and RF do, quite beyond the $\frac{10}{2{,}455} \approx 0.41\%$ of TP in the first column whose coefficients dominate, even if their precision is not as good. On the opposite, RForest and BootLasso do tend to produce reliable coefficients (at least in identifying non-zero values) but at the price of a very poor coverage.

MSE are also quite high in this scenarii, mainly because the coefficient matrix includes many coefficients with values set to 500. Hence, the effect of imprecisely estimated coefficients can have quite a large impact on MSE values, even it is actually a true coefficient. $\mathcal{D}$- Correlation sum, L^2-norm and REM again achieve the best MSE among tested approaches, with REM taking the advantage again because of richer, less precise models.

	Correlated covariates			Correlated noises		
	FP (60,045)	FN (2,455)	MSE	FP (60,045)	FN (2,455)	MSE
$\mathcal{D}$-Corr	4.39	74.20	0.63	7.07	83.14	0.60
L^2 norm	4.38	74.50	0.63	6.94	83.38	0.61
BootLasso	0.81	77.21	0.82	0.76	87.64	1.21
RForests*	2.27	78.63	0.84	0.79	97.15	0.93
REM	5.35	71.26	0.62	14.33	75.60	0.47

Table II.3: Second dataset: Percentage of false positive FP parameters (number of coefficients not to be predicted between brackets) and false negative FN parameters (number of coefficients to be predicted between brackets) and MSE for the Boosting algorithms, with a shrinkage factor $\gamma = 0.2$, compared with the BootLasso, RForests and REM; the sample size n is set to 100. We also indicate the number of edges to retrieve: $2,455$ and the number of potential FP: $250 * 250 - 2455 = 60,045$. (*: for five simulated replicate datasets only as the running time for RForest was four days per network).

Third dataset The last dataset mimics activation and inhibition relationships that exist between genes in the gene regulatory network of a living organism and is very close to a real data situation. This dataset, for which $p = 100$, is exactly the one that was provided by the DREAM

Project [DRE] in their Challenge 2 on the "In Silico Network Challenge" (more precisely the *InSilico_Size100_Multifactorial*). First, a directed network structure is chosen. Its features can be regarded as features of a biological network, *e.g.* in terms of degree distribution. Coupled ordinary differential equations (ODEs) then drive the quantitative impact of gene expression on each other, the expression of a gene roughly representing its activity in the system. For example, if gene 1 is linked to gene 2 with a positive effect going from 1 to 2, then increasing the expression of gene 1 (as operator *do*, see [Pea00]) will increase the expression of gene 2. However, increasing the expression of gene 2 does not have a direct effect on gene 1. Lastly, the system of ODEs is solved numerically to obtain steady states of the expression of the genes after technical and biological noises are created. We denote as A the $n \times p$ expression matrix of p genes for n individuals. This simulation process is highly non-linear compared to the first two scenarios described above.

The goal was to automatically retrieve network structure encoded in matrix A from data only. Samples were obtained by multifactorial perturbations of the network using GeneNetWeaver [SMF11] for simulations. A multifactorial perturbation is the simultaneous effect of numerous minor random perturbations in the network. It therefore measures a deviation from the equilibrium of the system. This could be seen as changes in the network due to very small environmental changes or genetic diversity in the population. Additional details and a discussion on the biological plausability (network structure, the use of chemical Langevin differential equations, system and experimental noise) of such datasets can be found in [MSMF09].

	FP (9,695)	FN (205)	MSE
$\mathcal{D}$-Corr	21.37	47.75	0.45
L^2 norm	18.98	50.20	0.50
BootLasso	1.40	77.93	0.32
RForests	7.68	68.98	0.20
REM	7.05	78.53	0.01

Table II.4: Third dataset: Percentage of false positive FP parameters (number of coefficents not to be predicted between brackets) and false negative FN parameters (number of coefficients to be predicted between brackets) and MSE for the Boosting algorithms, with a shrinkage factor $\gamma = 0.2$, compared to the BootLasso, RForests and REM.

The results of tested methods on this last dataset are presented in Table II.4. In this scenario, our two multivariate (we recall that $m = p = 100$) $\mathbb{L}_2$-Boosting algorithms both suffer from higher MSE. It also exhibits higher FP rates than other competing methods: $\approx 20\%$ vs 1.4, 7.7 and 7.1% for BootLasso, RF and REM, respectively. Many FP coefficients may imply an increase in MSE whereas the three other tested methods focus on fewer correct edges.

What can first be considered as a pitfall can be turned into a strength: recall can be close to (for L^2 norm) or even higher than ($\mathcal{D}$-Correlation sum) 50%, whereas other approaches reach 31% at best (RF). In other words, the $\mathcal{D}$-Correlation sum can retrieve more than half of the 205 edges to be predicted, at the price of producing more FP predictions, considered as noise from a model prediction perspective in the delivered list. RF is on average only able to grab 84 out of the 205 correct edges, but the prediction list is cleaner in a sense. A specifically designed variant of the RF approach that we tested was deemed the best performer for this challenge by the DREAM4 organisers [HTIWG10]. Our algorithm would have been ranked 2$^{\text{nd}}$.

For the sake of completeness, we computed the smallest and the largest eigenvalues of the restricted matrix ${}^tD_{\mathcal{S}}D_{\mathcal{S}}$, that are involved in the key Assumptions $\mathbf{H_{RE^-}}$ and $\mathbf{H_{RE^+}}$. We also provided the measured value of $\rho := \max_{j\notin\mathcal{S}} \|D_{\mathcal{S}}^{+} g_j\|_1$ of Assumption $\mathbf{H_S}$ in Table II.5 for the three datasets, which quantifies the coherence of the dictionary: favorable situations correspond to small values of ρ, ideally lower than 1.

	λ_{min}	λ_{max}	$\lambda_{max}/\lambda_{min}$	ρ
First dataset $(p, m, S) = (250, 1, 5)$	75.31	130.43	1.73	0.82
First dataset $(p, m, S) = (250, 1, 10)$	59.03	143.78	2.43	1.52
First dataset $(p, m, S) = (1000, 1, 20)$	37.66	190.71	5.06	2.80
First dataset $(p, m, S) = (250, 5, 50)$	49.14	157.20	3.20	1.71
First dataset $(p, m, S) = (250, 250, 50)$	52.78	151.76	2.88	1.10
Second dataset				
Correlated covariates	3.95	921.39	233.44	5.47
Correlated noises	41.12	181.49	4.41	1.88
Third dataset	19.29	233.83	12.12	1.57

Table II.5: Smallest and largest eigenvalue of the restricted matrix, ratio of these eigenvalues and computation of $\rho := \max_{j\notin\mathcal{S}} \|D_{\mathcal{S}}^{+} g_j\|_1$ for the three datasets.

Regarding the first dataset, we obtain a larger value than 0 for λ_{min} and a moderate value of λ_{max}. This implies a reasonable value of $\lambda_{max}/\lambda_{min}$. This situation is thus acceptable according to the bound given by Equations (33) and (35) (see Lemma 3.2). Concerning Assumption $\mathbf{H_S}$, for each range of parameters on the first dataset, ρ is not very far from 1, which explains the good numerical results. We have to particularly emphasize the first simulation study where $(p, m, S) = (250, 1, 5)$. With a coherence value ρ lower than 1, the WGA reaches to recover the true support of A. $\lambda_{max}/\lambda_{min}$ and ρ values for the second dataset support our numerical analysis (see Table II.3) that shows that this is a very difficult dataset. This situation is clearly less favourable for the sparse estimation provided by our Boosting procedures than for the first dataset. This is perhaps less visible for the second simulated setting, where additional noise was is correlated. Clearly in this latter case, hypothesis $\mathbf{H^{Mult}_{dim-3}}$ is violated because the noise coordinates are not i.i.d. anymore. We however have no numerical indicator to quantify this.

For the last dataset, we can observe that $\mathbf{H_S}$ yields a moderate value of ρ but that the ratio of the restricted eigenvalues is quite large (compared to thoses obtained in the first dataset) and it is difficult to recover the support of the true network.

Taken together, this numerically shows that both $\mathbf{H_S}$ and $\mathbf{H_{RE^-}}$, $\mathbf{H_{RE^+}}$ are important to obtain good reconstruction properties. These assumptions then seem complementary and not redundant. However, the practical use of the proposed algorithms advocates a certain tolerance of the method towards divergence from the hypotheses that condition our theoretical results.

Concluding remarks

In this chapter, we studied WGA and established a support recovery result for solving linear regression in high-dimensional settings. We then proposed two statistically funded $\mathbb{L}_2$-Boosting algorithms derived thereupon in a multivariate framework. The algorithms were developped to sequentially estimate unknown parameters in a high-dimensional regression framework: significant possibly correlated regressor functions from a dictionnary need be identified, relative coefficients need be estimated and noise can disturb the observations. Consistency of two variants of the algorithms was proved in Theorem 3.2 for the L^2 norm variant and in Theorem 3.3 for the $\mathcal{D}$-Correlation sum variant. An important Support Recovery result (Theorem 3.4) under mild assumption on sparsity of the regression function and on the restricted isometry of the X matrix then generalises the univariate result to the multi-task framework. Using the MSE of the model, we derived a simple yet effective stopping criterion for our algorithms.

We then illustrated the proposed algorithms in a variety of simulated datasets in order to determine the ability of the proposed method to compete with state-of-the-art methods when the data is high-dimensional, noisy and the active elements can be unbalanced. Even if the algorithms we propose are not superior in all settings, we observed, for example, that they are very competitive in situations such as those of the DREAM4 In Silico Multifactorial Network Challenge. Without fine parameter tuning and with a very small computing time, our generic method would have ranked 2^{nd} in this challenge. Moreover, it has the ability to quickly produce a rich prediction list of edges at an acceptable quality level, which might reveal novel regulatory mechanisms on real biological datasets.

Chapter III

$\mathbb{L}_2$-Boosting on a generalized Hoeffding decomposition for dependent variables

This chapter has been accepted under a slightly form as [CCGP14], as joint work with Gaëlle Chastaing, Sébastien Gadat and Clémentine Prieur.

Abstract

This chapter is dedicated to the study of an estimator of the generalized Hoeffding decomposition. We build such an estimator using an empirical Gram-Schmidt approach and derive a consistency rate in a large dimensional setting. We then apply a greedy algorithm with these previous estimators to a sensitivity analysis. We also establish the consistency of this $\mathbb{L}_2$-Boosting under sparsity assumptions of the signal to be analyzed. The chapter concludes with numerical experiments, that demonstrate the low computational cost of our method, as well as its efficiency on the standard benchmark of sensitivity analysis.

1 Introduction

In many scientific fields, it is desirable to extend a multivariate regression model as a specific sum of increasing dimension functions. Functional ANOVA decompositions and High Dimensional Representation Models (HDMR) ([Hoo07] and [LRY$^+$10]) are well known expansions that make it possible to understand model behavior and to detect how inputs interact with each other.

When input variables are independent, Hoeffding establishes the uniqueness of the decomposition provided that the summands are mutually orthogonal [Hoe48]. However, in practice, this assumption is sometimes difficult to justify, or may even be wrong (see [LR10] for an application to correlated ionosonde data, or [JLD06], who studied an adjusted neutron spectrum inferred from a correlated dependent nuclear dataset).

When inputs are correlated, the orthogonality properties of the classical Sobol decomposition [Sob93] are no longer satisfied. As pointed out by several authors ([Hoo07] and [DVWG09]), a global sensitivity analysis based on this decomposition may lead to erroneous conclusions. Following the work of [Sto94], later applied in Machine Learning by [Hoo07], and to sensitivity analysis by [CGP12], we consider a hierarchically orthogonal decomposition in this work, whose uniqueness has been proved under mild conditions on the dependence structure of the inputs [CGP12]. In other words, any model function can be uniquely decomposed as a sum of *hierarchically orthogonal* component functions. Two summands are considered to be *hierarchically*

orthogonal whenever all of the variables included in one of them are also involved in the other. For a better understanding, this generalized ANOVA expansion will be referred to as a Hierarchically Orthogonal Functional Decomposition (HOFD).

It is of great importance to develop estimation procedures since the analytical formulation for HOFD is rarely available. In this chapter, we focus on an alternative method proposed in [CGP72] to estimate the HOFD components. It consists in constructing a hierarchically orthogonal basis from a suitable Hilbert orthonormal basis. Inspired by the usual Gram-Schmidt algorithm, the procedure recursively builds a multidimensional basis for each component that satisfies the identifiability constraints imposed on this summand. Each component is then well approximated on a truncated basis, where the unknown coefficients are deduced by solving an ordinary least-squares regression. Nevertheless, in a high-dimensional paradigm, this procedure suffers from a curse of dimensionality. Moreover, it is numerically observed that only a few of coefficients are not close to zero, meaning that only a small number of predictors restore the major part of the information contained in the components. Thus, it is important to be able to select the most relevant representative functions, and to then identify the HOFD with a limited computational budget.

With this in mind, we suggest in this article to transform the ordinary least-squares regression into a penalized regression, as has been proposed in [CGP72]. We focus here on the $\mathbb{L}_2$-Boosting to deal with the ℓ_0 penalization, developed by [Fri01]. The $\mathbb{L}_2$-Boosting is a greedy strategy that performs variable selection and shrinkage. The choice of such an algorithm is motivated by the fact that the $\mathbb{L}_2$-Boosting is very intuitive and easy to implement. It is also closely related (from the practical point of view) to the LARS algorithm proposed by [EHJT04], which solves the Lasso regression (see *e.g.* [Büh13] and [Tib96]). The $\mathbb{L}_2$-Boosting and the LARS both select predictors using the maximal correlation with the current residuals.

The question that naturally arises now is the following: provided that the theoretical procedure of component reconstruction is well tailored, do the estimators obtained by the $\mathbb{L}_2$-Boosting converge to the theoretical true sparse parameters when the number of observations tends to infinity? The goal of this chapter is to provide an overall consistent estimation of a signal spanned into a large dimensional dictionary derived from a HOFD. Hence, our work significantly improves the results of [CGP72]: we first address the convergence rate of the empirical HOFD and then use this result to obtain a sparse estimator of the unknown signal. We will need to manage sufficient theoretical conditions to ensure the consistency of our estimator. In addition, we discuss these conditions and provide some numerical examples in which such conditions are fulfilled.

The chapter is organized as follows. The notations used in this work are presented in Section 2.1. Section 2.2 provides the HOFD representation of the model function. In Section 2.3, we review the procedure detailed in [CGP72] that consists in constructing well- tailored hierarchically orthogonal basis to represent the components of the HOFD. Finally, we highlight the curse of dimensionality that we are exposed to, and present the $\mathbb{L}_2$-Boosting. Section 3 describes our main theoretical results on the proposed algorithms and the proofs of the two main theorems are given in Section 4. One interesting application of the general theory is the global sensitivity analysis (SA). In Section 5, we apply the $\mathbb{L}_2$-Boosting to estimate the generalized sensitivity indices defined in [CGP12]. After recalling the form of these indices, we numerically compare the $\mathbb{L}_2$-Boosting performance with a Lasso strategy and the Forward-Backward algorithm, proposed by [Zha11].

2 Estimation of the generalized Hoeffding decomposition components

2.1 Model and notations

We consider a measurable function f of a random real vector $X = (X^1, \cdots, X^p)$ of $\mathbb{R}^p$, $p \geq 1$. The response variable Y is a real-valued random variable defined as

$$Y = f(X) + \varepsilon, \tag{III.1}$$

where ε stands for a centered random variable independent of X and models the variability of the response around its theoretical unknown value f. We denote the distribution law of X by P_X, which is unknown in our setting, and we assume that P_X admits a density function p_X with respect to the Lebesgue measure on $\mathbb{R}^p$. Note that P_X is not necessarily a tensor product of univariate distributions since the components of X may be correlated.

Furthermore, we suppose that $f \in L^2_{\mathbb{R}}(\mathbb{R}^p, \mathcal{B}(\mathbb{R}^p), P_X)$, where $\mathcal{B}(\mathbb{R}^p)$ denotes the Borel set of $\mathbb{R}^p$. The Hilbert space $L^2_{\mathbb{R}}(\mathbb{R}^p, \mathcal{B}(\mathbb{R}^p), P_X)$ is denoted by $L^2_{\mathbb{R}}$, for which we use the inner product $\langle \cdot, \cdot \rangle$, and the norm $\|\cdot\|$ as follows:

$$\forall h, g \in L^2_{\mathbb{R}}, \qquad \langle h, g \rangle = \int h(x) g(x) p_X \, dx = \mathbb{E}(h(X)g(X)),$$
$$\|h\|^2 = \langle h, h \rangle = \mathbb{E}\left(h(X)^2\right),$$

where $\mathbb{E}(\cdot)$ stands for the expected value. Further, $\mathrm{Var}(\cdot) = \mathbb{E}[(\cdot - \mathbb{E}(\cdot))^2]$ denotes the variance, and $\mathrm{Cov}(\cdot, *) = \mathbb{E}[(\cdot - \mathbb{E}(\cdot))(* - \mathbb{E}(*))]$ the covariance.

For any $1 \leq i \leq p$, we denote the marginal distribution of X_i by P_{X^i} and naturally extend the former notation to $L^2_{\mathbb{R}}(\mathbb{R}, \mathcal{B}(\mathbb{R}), P_{X^i}) := L^2_{\mathbb{R},i}$.

2.2 The generalized Hoeffding decomposition

Let S be the collection of all subsets of $[\![1, p]\!]$. We also define $S^* := S \backslash \{\emptyset\}$. For $u \in S$, the subvector X^u of X is defined as $X^u := (X^i)_{i \in u}$. Conventionally, for $u = \emptyset$, $X^u = 1$. The marginal distribution (*resp.* density) of X^u is denoted by P_{X^u} (*resp.* p_{X^u}).

A functional ANOVA decomposition consists in expanding f as a sum of increasing dimension functions:

$$\begin{aligned} f(X) &= f_\emptyset + \sum_{i=1}^{p} f_i(X^i) + \sum_{1 \leq i < j \leq p} f_{ij}(X^i, X^j) + \cdots + f_{1,\cdots,p}(X) \\ &= \sum_{u \in S} f_u(X^u), \end{aligned} \tag{III.2}$$

where $f_\emptyset$ is a constant term, f_i, $i \in [\![1, p]\!]$ are the main effects, $f_{ij}, f_{ijk}, \cdots$, $i, j, k \in [\![1, p]\!]$ are the interaction effects, and the last component $f_{1,\cdots,p}$ is the residual.

Decomposition (III.2) is generally not unique. However, under mild assumptions on the joint density p_X (see Assumptions (C.1) and (C.2) in [CGP12]), the decomposition is unique under some additional orthogonality assumptions.

More precisely, let us introduce $H_\emptyset = H^0_\emptyset$ the set of constant functions, and for all $u \in S^*$, $H_u := L^2_{\mathbb{R}}(\mathbb{R}^{|u|}, \mathcal{B}(\mathbb{R}^{|u|}), P_{X^u})$. We then define H^0_u, $u \in S \backslash \emptyset$ as follows:

$$H^0_u = \left\{ h_u \in H_u, \langle h_u, h_v \rangle = 0, \forall v \subset u, \forall h_v \in H^0_v \right\},$$

where $\subset$ denotes the strict inclusion.

Definition 2.1 (Hierarchical Orthogonal Functional Decomposition - HOFD)**.** *Under Assumptions (C.1) and (C.2) in [CGP12], the decomposition (III.2) is unique as soon as we assume* $f_u \in H_u^0$ *for all* $u \in S$.

Remark 3. *The components of the HOFD* (III.2) *are assumed to be hierarchically orthogonal, that is,* $\langle f_u, f_v \rangle = 0 \ \forall v \subset u$.

To obtain more information about the HOFD, the reader is referred to [Hoo07] and [CGP12]. In this chapter, we are interested in estimating the summands in (III.2). As underlined in [Hua98], estimating all components of (III.2) suffers from a curse of dimensionality, leading to an intractable problem in practice. To bypass this issue, we will assume (without loss of generality) that f is centered, so that $f_\emptyset = 0$. Furthermore, most of the models are only governed by low-order interaction effects, as pointed out in [CLMM09], [Bla09] and [LRY$^+$10]. We thus suppose that f is well approximated by:

$$f(X) \simeq \tilde{f}(X) = \sum_{\substack{u \in S^* \\ |u| \leq d}} f_u(X^u), \quad d \ll p, \tag{III.3}$$

so that interactions of order $\geq d+1$ can be neglected. The choice of d, which is directly related to the notion of effective dimension in the superposition sense (see Definition 1 in [WF03]) is addressed in [MZKS13], but is not of great interest in the present work, so that it is assumed to be fixed by the user. Even by choosing a small d, the number of components in (III.3) can become prohibitive if the number of inputs p is high. We therefore are interested in estimation procedures under sparse assumptions when the number of variables p is large.

In the next section, we recall the procedure devoted to identify components of (III.3). As a result of this strategy, we highlight the curse of dimensionality when p becomes large, and we propose to use a greedy $\mathbb{L}_2$-Boosting to tackle this issue.

2.3 Practical determination of the sparse HOFD

We propose a two-step estimation procedure in this section to identify the components in (III.3). The first one is a simplified version of the Hierarchical Orthogonal Gram-Schmidt (HOGS) procedure developed in [CGP12] and the second consists of a sparse estimation in the dictionary learnt by the empirical HOGS.

To carry out this two-step procedure, we assume that we observe two independent and identically distributed samples $(Y_r, X_r)_{r=1,\cdots,n_1}$ and $(Y_s, X_s)_{s=1,\cdots,n_2}$ from the distribution of (Y, X) (the initial sample can be splitted in such two samples). We define the empirical inner product $\langle \cdot, \cdot \rangle_{\mathrm{n}}$ and the empirical norm $\|\cdot\|_{\mathrm{n}}$ associated to a n-sample as

$$\langle h, g \rangle_{\mathrm{n}} = \frac{1}{n} \sum_{k=1}^{n} h(X_k) g(X_k), \quad \|h\|_{\mathrm{n}} = \langle h, h \rangle_{\mathrm{n}}.$$

Also, for $u = (u_1, \cdots, u_t) \in S$, we define the multi-index $\boldsymbol{l_u} = (l_{u_1}, \cdots, l_{u_t}) \in \mathbb{N}^t$. We use the notation Span $\{B\}$ to define the set of all finite linear combinations of elements of B, also referred to as the linear span of B.

Step 1 and Step 2 of our sparse HOFD procedure will be described in detail below.

Remark 4. *The procedure could be extented to any higher order approximation, but we think that the description of the methodology for* $d = 2$ *provides a better understanding. We have thus chosen to only describe this situation for the sake of clarity.*

2.3.1 Step 1: Hierarchically Orthogonal Gram-Schmidt Procedure

For each $i \in [\![1,p]\!]$, let $\{1\,,\ \psi^i_{\boldsymbol{l_i}}\,,\ \boldsymbol{l_i} \in \mathbb{N}^*\}$ denote an orthonormal basis of $H_i := L^2(\mathbb{R}, \mathcal{B}(\mathbb{R}), P_{X^i})$. For $L \in \mathbb{N}^*$, for $i \neq j \in [\![1,p]\!]$, we set

$$H^L_\emptyset = \text{Span}\,\{1\} \quad \text{and} \quad H^L_i = \text{Span}\left\{1, \psi^i_1, \cdots, \psi^i_L\right\},$$

as well as:

$$H^L_{ij} = \text{Span}\left\{1, \psi^i_1, \cdots, \psi^i_L, \psi^j_1, \cdots, \psi^j_L, \psi^i_1 \otimes \psi^j_1, \cdots, \psi^i_L \otimes \psi^j_L\right\},$$

where $\otimes$ denotes the tensor product between two elements of the basis. We define $H^{L,0}_u$, the approximation of H^0_u, as:

$$H^{L,0}_u = \left\{h_u \in H^L_u,\ \langle h_u, h_v\rangle = 0, \forall\ v \subset u, \forall\ h_v \in H^{L,0}_v\right\},$$

The recursive procedure below aims at constructing a basis for $H^{L,0}_i$ and a basis for $H^{L,0}_{ij}$ for any $i \neq j \in [\![1,p]\!]$.

Initialization For any $1 \leq i \leq p$, define $\phi^i_{\boldsymbol{l_i}} := \psi^i_{\boldsymbol{l_i}}, \quad \forall \boldsymbol{l_i} \in [\![1,L]\!]$. Then, as a result of the orthogonality of $\{\psi^i_{\boldsymbol{l_i}}\,,\ l_i \in \mathbb{N}\}$, we obtain:

$$H^{L,0}_i := \text{Span}\left\{\phi^i_1, \cdots \phi^i_L\right\}.$$

For this step, we just need the orthogonality of the constant function equal to 1 with each of the $\psi^i_{\boldsymbol{l_i}}\,,\ \boldsymbol{l_i} \in \mathbb{N}^*$. However, orthogonality is needed for the proof of the consistency of the $\mathbb{L}_2$-Boosting procedure (see Section 3).

Second order interactions Let $u = \{i,j\}$, with $i \neq j \in [\![1,p]\!]$ and consider the spaces $H^{L,0}_i$ and $H^{L,0}_j$ constructed according to the step of initialization. Since the dimension of H^L_{ij} is equal to L^2+2L+1, and the approximation space $H^{L,0}_{ij}$ is subject to $2L+1$ constraints, its dimension is then equal to L^2. We want to construct a basis for $H^{L,0}_{ij}$, which satisfies the hierarchical orthogonal constraints. To build such a basis, we proceed as follows:

1. Set

$$\phi^{ij}_{\boldsymbol{l_{ij}}}(X^i, X^j) = \phi^i_{\boldsymbol{l_i}}(X^i) \times \phi^j_{\boldsymbol{l_j}}(X^j) + \sum_{k=1}^{L} \lambda^i_{k,\boldsymbol{l_{ij}}} \phi^i_k(X^i) + \sum_{k=1}^{L} \lambda^j_{k,\boldsymbol{l_{ij}}} \phi^j_k(X^j) + C_{\boldsymbol{l_{ij}}}, \quad \text{(III.4)}$$

   with $\boldsymbol{l_{ij}} = (\boldsymbol{l_i}, \boldsymbol{l_j}) \in [\![1,L]\!]^2$.

2. Compute the constants $\left(C_{\boldsymbol{l_{ij}}}, \left(\lambda^i_{k,\boldsymbol{l_{ij}}}\right)_{k=1}^L, \left(\lambda^j_{k,\boldsymbol{l_{ij}}}\right)_{k=1}^L\right)$ by resolving the following constraints:

$$\begin{aligned} &\langle \phi^{ij}_{\boldsymbol{l_{ij}}}, \phi^i_k\rangle = 0, \quad \forall\ k \in [\![1,L]\!], \\ &\langle \phi^{ij}_{\boldsymbol{l_{ij}}}, \phi^j_k\rangle = 0, \quad \forall\ k \in [\![1,L]\!], \\ &\langle \phi^{ij}_{\boldsymbol{l_{ij}}}, 1\rangle = 0. \end{aligned} \quad \text{(III.5)}$$

   Removing the constant term $C_{\boldsymbol{l_{ij}}}$, the linear system (III.4) with the constraints (III.5) leads to the linear system:

$$A^{ij}\boldsymbol{\lambda}^{\boldsymbol{l_{ij}}} = D^{\boldsymbol{l_{ij}}}, \quad \text{(III.6)}$$

where

$$A^{ij} = \begin{pmatrix} B^{ii} & B^{ij} \\ {}^tB^{ij} & B^{jj} \end{pmatrix}, \quad \text{with } B^{ij} = \begin{pmatrix} \langle \phi_1^i, \phi_1^j \rangle & \cdots & \langle \phi_1^i, \phi_L^j \rangle \\ \vdots & & \vdots \\ \langle \phi_L^i, \phi_1^j \rangle & \cdots & \langle \phi_L^i, \phi_L^j \rangle \end{pmatrix}, \tag{III.7}$$

and

$$\boldsymbol{\lambda}^{\boldsymbol{l_{ij}}} = \begin{pmatrix} \lambda^i_{1,\boldsymbol{l_{ij}}} \\ \vdots \\ \lambda^i_{L,\boldsymbol{l_{ij}}} \\ \lambda^j_{1,\boldsymbol{l_{ij}}} \\ \vdots \\ \lambda^j_{L,\boldsymbol{l_{ij}}} \end{pmatrix}, \quad D^{\boldsymbol{l_{ij}}} = - \begin{pmatrix} \langle \phi^i_{\boldsymbol{l_i}} \times \phi^j_{\boldsymbol{l_j}}, \phi^i_1 \rangle \\ \vdots \\ \langle \phi^i_{\boldsymbol{l_i}} \times \phi^j_{\boldsymbol{l_j}}, \phi^i_L \rangle \\ \langle \phi^i_{\boldsymbol{l_i}} \times \phi^j_{\boldsymbol{l_j}}, \phi^j_1 \rangle \\ \vdots \\ \langle \phi^i_{\boldsymbol{l_i}} \times \phi^j_{\boldsymbol{l_j}}, \phi^j_L \rangle \end{pmatrix}. \tag{III.8}$$

As shown in [CGP72], $A^{\boldsymbol{l_{ij}}}$ is a definite positive Gramian matrix and (III.6) provides a unique solution in $\boldsymbol{\lambda}^{\boldsymbol{l_{ij}}}$. Then, $C_{\boldsymbol{l_{ij}}}$ is deduced with

$$C_{\boldsymbol{l_{ij}}} = -\mathbb{E}\Big[\phi^i_{\boldsymbol{l_i}} \otimes \phi^j_{\boldsymbol{l_j}}(X^i, X^j) + \sum_{k=1}^{L} \lambda^i_{k,\boldsymbol{l_{ij}}} \phi^i_k(X^i) + \sum_{k=1}^{L} \lambda^j_{k,\boldsymbol{l_{ij}}} \phi^j_k(X^j)\Big]. \tag{III.9}$$

3. At the end, set

$$H_{ij}^{L,0} = \text{Span}\left\{\phi^{ij}_{\boldsymbol{l_{ij}}}, \boldsymbol{l_{ij}} = (l_i, l_j) \in [\![1, L]\!]^2\right\}.$$

Higher interactions This construction can be extended to any $|u| \geq 3$ (for more details, see [CGP72]). Just note that the dimension of the approximation space $H_u^{L,0}$ is given by $L_u = L^{|u|}$, where $|u|$ denotes the cardinality of u.

Empirical procedure Algorithm 5 below proposes an empirical version of the HOGS procedure. It consists in substituting the inner product $\langle \cdot, \cdot \rangle$ by its empirical version $\langle \cdot, \cdot \rangle_{n_1}$ obtained with the first data set $(Y_r, X_r)_{r=1,\cdots,n_1}$.

Algorithm 5: Empirical HOFD (EHOFD)

Input: Orthonormal system $(\psi^i_{\boldsymbol{l_i}})_{\boldsymbol{l_i}=0}^{L}$ of H_i, $i \in [\![1, p]\!]$, i.i.d. observations $\mathcal{O}_1 := (Y_r, X_r)_{r=1,\cdots,n_1}$ of (III.1), threshold $|u_{max}|$

Initialization: for any $i \in [\![1, p]\!]$ and $l_i \in [\![1, L]\!]$, define first $\hat{\phi}^i_{\boldsymbol{l_i}, n_1} = \psi^i_{\boldsymbol{l_i}}$.

Step 1 For any u such that $2 \leq |u| \leq |u_{max}|$, write the matrix $\left(\hat{A}^u_{n_1}\right)$ as well as $\left(\hat{D}^{\boldsymbol{l_u}}_{n_1}\right)$ obtained using the former expressions with $\langle \cdot, \cdot \rangle_{n_1}$.

Step 2 Solve (III.6) with the empirical inner product $\langle \cdot, \cdot \rangle_{n_1}$. Then, compute $\left(\hat{\boldsymbol{\lambda}}^{\boldsymbol{l_{ij}}}_{n_1}\right)$ and $\left(\hat{C}^{n_1}_{\boldsymbol{l_{ij}}}\right)$ with Equation (III.9).

Step 3 The empirical version of the basis given by (III.4) is then:

$$\forall u \in [\![2, |u_{max}|]\!], \quad \hat{H}_u^{L,0,n_1} = \text{Span}\left\{\hat{\phi}^u_{1,n_1}, \cdots, \hat{\phi}^u_{L^{|u|},n_1}\right\}.$$

2.3.2 Step 2: Greedy selection of Sparse HOFD

Each component f_u of the HOFD defined in Definition 2.1 is a projection onto H_u^0. Since, for $u \in S_n^*$, the space $\hat{H}_u^{L,0,n_1}$ well approximates H_u^0, it is then natural to approximate f by:

$$f(X) \simeq \bar{f}(X) = \sum_{\substack{u \in S^* \\ |u| \leq d}} \bar{f}_u(X^u), \text{ with } \bar{f}_u(X^u) = \sum_{\boldsymbol{l_u}} \beta^u_{\boldsymbol{l_u}} \hat{\phi}^u_{\boldsymbol{l_u},n_1}(X^u),$$

where $\boldsymbol{l_u}$ is the multi-index $\boldsymbol{l_u} = (l_i)_{i \in u} \in [\![1, L]\!]^{|u|}$. For the sake of clarity (since there is no ambiguity), we will omit the summation support of $\boldsymbol{l_u}$ in the sequel.

We consider now the second sample $(Y_s, X_s)_{s=1,\cdots,n_2}$ and we attempt to recover the unknown coefficients $\left(\beta^u_{\boldsymbol{l_u}}\right)_{\boldsymbol{l_u}, |u| \leq d}$ on the regression problem,

$$Y_s = \bar{f}(X_s) + \varepsilon^s, \quad s = 1, \cdots, n_2.$$

However, the number of coefficients is equal to $\sum_{k=1}^{d} \binom{p}{k} L^k$. When p becomes large, the usual least-squares estimator is not adapted to estimate the coefficients $(\beta^u_{\boldsymbol{l_u}})_{\boldsymbol{l_u},u}$. We then use the penalized regression,

$$(\hat{\beta}^u_{\boldsymbol{l_u}}) \in \underset{\beta^u_{\boldsymbol{l_u}} \in \mathbb{R}}{\operatorname{argmin}} \frac{1}{n_2} \sum_{s=1}^{n_2} \left[Y_s - \sum_{\substack{u \in S^* \\ |u| \leq d}} \sum_{\boldsymbol{l_u}} \beta^u_{\boldsymbol{l_u}} \hat{\phi}^u_{\boldsymbol{l_u},n_1}(X^u_s) \right]^2 + \lambda J(\beta^1_1, \cdots, \beta^u_{\boldsymbol{l_u}}, \cdots), \tag{III.10}$$

where $J(\cdot)$ is the ℓ_0-penalty, i.e.

$$J(\beta^1_1, \cdots, \beta^u_{\boldsymbol{l_u}}, \cdots) = \sum_{\substack{u \in S^* \\ |u| \leq d}} \sum_{\boldsymbol{l_u}} \mathbb{1}(\beta^u_{\boldsymbol{l_u}} \neq 0).$$

Of course, such an optimisation procedure is not tractable. In the following, we have chosen to use the so-called $\mathbb{L}_2$-Boosting procedure instead of the widespread Lasso estimator. Indeed, this choice is motivated by two reasons.

— First, from a technical point of view, the empirical HOGS will produce a noisy estimation of the theoretical dictionary, in which the true signal f is expanded. Hence, the arguments produced for the Lasso estimation would have to be completely adjusted to this situation with errors in the variables. Moreover, as an M-estimator, such a modification is far from being trivial (see [CH05] for an example of oracle inequalities derived from M estimators with noise in the variables). In contrast, the approximation obtained in the empirical HOGS can be easily handled with the Boosting algorithm since we just have to quantify how the empirical inner products built with noisy variables are close to theoretical ones. Our proofs rely precisely on this strategy: we obtain a uniform bound on our statistical estimation of the HOGS dictionary, and then take advantage of the sequential description of the Boosting with empirical inner products.

— Second, in order to obtain consistent estimation with the Boosting procedure, we do not need to make any coherence assumption on the dictionary (such as the RIP assumption of [CT07] or the weakest RE(s, c_0) assumption of [BRT09]). Such assumptions are generally necessary to assert some consistency results for the Dantzig and Lasso procedures, such as Sparse Oracle Inequalities (SOI), for example. Nevertheless, it would be only reasonable here to impose these latter assumptions on the *theoretical* version of the HOGS although it seems difficult to deduce coherence results on the *empirical* HOGS from coherence results

on the *theoretical* version of the HOGS. Our Theorem 3.2 below will not produce a SOI in expectation and our results will instead be expressed in probability. We will discuss the asymptotics involved in Theorem 3.2 after its statement, and underline the differences with the state of the art results on the Lasso estimator.

Mimicking the notation of [Tem00] and [CCAGV14], we define the dictionary $\mathcal{D}$ of functions as

$$\mathcal{D} = \{\hat{\phi}^1_{1,n_1}, \cdots \hat{\phi}^1_{L,n_1}, \cdots, \hat{\phi}^u_{1,n_1}, \cdots, \hat{\phi}^u_{L_u,n_1}, \cdots\}.$$

The quantity $G_k(\bar{f})$ denotes the approximation of $\bar{f}$ at step k, as a linear combination of elements of $\mathcal{D}$. At the end of the algorithm, the estimation of $\bar{f}$ is denoted $\hat{f}$. The $\mathbb{L}_2$-Boosting is described in Algorithm 6.

Algorithm 6: The $\mathbb{L}_2$-Boosting

Input: Observations $\mathcal{O}_2 := (Y_s, X_s)_{s=1,\cdots,n_2}$, shrinkage parameter $\gamma \in]0,1]$ and number of iterations $k_{up} \in \mathbb{N}^*$.
Initialization: $G_0(\bar{f}) = 0$.
for $k = 1$ *to* k_{up} **do**

Step 1: Select $\hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1} \in \mathcal{D}$ such that

$$\left|\langle Y - G_{k-1}(\bar{f}), \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}\rangle_{n_2}\right| = \max_{\hat{\phi}^u_{\boldsymbol{l_u},n_1} \in \mathcal{D}} \left|\langle Y - G_{k-1}(\bar{f}), \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle_{n_2}\right|. \tag{III.11}$$

Step 2: Compute the new approximation of $\bar{f}$ as

$$G_k(\bar{f}) = G_{k-1}(\bar{f}) + \gamma\langle Y - G_{k-1}(\bar{f}), \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}\rangle_{n_2} \cdot \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}. \tag{III.12}$$

end
Output: $\hat{f} = G_{k_{up}}(\bar{f})$.

For any step k, Algorithm 6 selects a function from $\mathcal{D}$ that provides sufficient information about the residual $Y - G_{k-1}(\bar{f})$. The shrinkage parameter γ is the standard step-length parameter of the Boosting algorithm. It actually smoothly inserts the next predictor in the model, making possible a refinement of the greedy algorithm, and may statistically guarantees its convergence rate. In a deterministic setting, the shrinkage parameter is not really useful and may be set to 1 (see [Tem00] for further details). It is paticularly useful from a practical point of view to smooth the Boosting iterations.

2.3.3 An algorithm for our new sparse HOFD procedure

Algorithm 7 below now provides a simplified description of our sparse HOFD procedure, whose steps have been described above.

We now obtain a strategy to estimate the components of the decomposition (III.3) in a high-dimensional paradigm. We aim to show that the obtained estimators are consistent, and that the two-step procedure (summarized in Algorithm 7) is numerically convincing. The next section is devoted to the asymptotic properties of the estimators.

Algorithm 7: Greedy Hierarchically Orthogonal Functional Decomposition

Input: Orthonormal system $(\psi^i_{\boldsymbol{l_i}})_{l_i=0}^{L}$ of $L^2(\mathbb{R}, \mathcal{B}(\mathbb{R}), P_{X^i})$, $i \in [\![1, p]\!]$, i.i.d. observations $\mathcal{O} := (Y_j, X_j)_{j=1\dots n}$ of (III.1)

Initialization: Split $\mathcal{O}$ in a partition $\mathcal{O}_1 \cup \mathcal{O}_2$ of size (n_1, n_2).

Step 1: For any $u \in S$, use Algorithm 5 with observations $\mathcal{O}_1$ to construct the set

$$\hat{H}_u^{L,0,n_1} := \text{Span}\left\{\hat{\phi}^u_{1,n_1}, \cdots, \hat{\phi}^u_{L,n_1}\right\},$$

approximation of $H_u^{L,0}$.

Step 2: Use an $\mathbb{L}_2$-Boosting algorithm on $\mathcal{O}_2$ with the random dictionary $\mathcal{D} = \{\hat{\phi}^1_{1,n_1}, \cdots \hat{\phi}^1_{L,n_1}, \cdots, \hat{\phi}^u_{1,n_1}, \cdots, \hat{\phi}^u_{L,n_1}, \cdots\}$ to obtain the Sparse Hierarchically Orthogonal Decomposition (see Algorithm 6).

3 Consistency of the estimator

In this section, we study the asymptotic properties of the estimator $\hat{f}$ obtained from Algorithm 7 described in Section 2.3.3. To do this, we restrict our study to the case of $d = 2$ and assume that f is well approximated by first and second order interaction components (see Remark 5 below). Hence, the observed signal Y may be represented as

$$Y = \tilde{f}(X) + \varepsilon, \quad \tilde{f}(X) = \sum_{\substack{u \in S_n^* \\ |u| \leq 2}} \sum_{\boldsymbol{l_u}} \beta^{u,0}_{\boldsymbol{l_u}} \phi^u_{\boldsymbol{l_u}}(X^u) \in H_u^L,$$

where $\boldsymbol{\beta}^0 = (\beta^{u,0}_{\boldsymbol{l_u}})_{\boldsymbol{l_u},u}$ is the true parameter that expands $\tilde{f}$, and the functions $\left(\phi^u_{\boldsymbol{l_u}}\right)_{\boldsymbol{l_u}}$, $|u| \leq 2$ are constructed according to the HOFD described in Section 2.2.

We assume that we have in hand an n-sample of observations, divided into two samples $\mathcal{O}_1$ and $\mathcal{O}_2$. Samples in $\mathcal{O}_1$ (*resp.* in $\mathcal{O}_2$) of size $n_1 = n/2$ (*resp.* of size $n_2 = n/2$) are used for the construction of $\left(\hat{\phi}^u_{\boldsymbol{l_u},n_1}\right)_{\boldsymbol{l_u},u}$ described in Algorithm 5 (*resp.* for the $\mathbb{L}_2$-Boosting Algorithm 6 to estimate $\left(\beta^u_{\boldsymbol{l_u}}\right)_{\boldsymbol{l_u},u}$).

The goal of this section is to study the consistency of $\hat{f} = G_{k_n}(\bar{f})$ when the sample size n tends to infinity. Its objective is also to determine an optimal number of steps k_n to get a consistent estimator from Algorithm 6.

Remark 5. *We choose the truncature order $d = 2$ in order to simplify the presentation, but it may be extended to arbitrary larger thresholds independent of the sample size n. This choice is legitimate as soon as the function f is well approximated by low interaction components and this assumption is well suited for many practical situations ([RASS99] and [Sob01]). Indeed, a data-dependent choice of d_n (with $d_n \longrightarrow +\infty$ as $n \to +\infty$) would rely on a smoothness assumption on the signal f with respect to the order of the considered interactions by exploiting the size of the bias term induced by the truncature given in Theorem 5 of [Sob01]. However, this challenging task is far beyond the scope of this work and we have chosen to leave this problem open.*

3.1 Assumptions

We first briefly recall some notation: for all sequences $(a_n)_{n\geq 0}$, $(b_n)_{n\geq 0}$, we write $a_n = \underset{n\to+\infty}{\mathcal{O}}(b_n)$ when a_n/b_n is a bounded sequence for n large enough. Now, for any random sequence

$(X_n)_{n\geq 0}$, $X_n = \mathcal{O}_P(a_n)$ means that $|X_n/a_n|$ is bounded in probability.

We have chosen to present our assumptions in three parts to deal with the dimension, the noise and the sparseness of the entries.

Bounded Assumptions ($\mathbf{H_b}$) The first set of hypotheses matches the *bounded case* and is adapted to the special situation of bounded support for the random variable X, for example, when each X_j follows a uniform law on a compact set $\mathcal{K}_j \subset K$ where K is a compact set of $\mathbb{R}$ independent of $j \in [\![1, p]\!]$. It is referred to as ($\mathbf{H_b}$) in the sequel and corresponds to the following three conditions:

($\mathbf{H_b^1}$) $M := \sup_{\substack{i\in[\![1,p]\!] \\ l_i\in[\![1,L]\!]}} \left\|\phi^i_{\boldsymbol{l_i}}(X_i)\right\|_\infty < +\infty$.

($\mathbf{H_b^2}$) The number of variables p_n satisfies

$$p_n = \underset{n\to+\infty}{\mathcal{O}}(\exp(Cn^{1-\xi})), \text{ where } 0 < \xi \leq 1 \text{ and } C > 0.$$

($\mathbf{H_b^{3,\vartheta}}$) The Gram matrices A^{ij} introduced in (III.6) satisfies:

$$\exists C > 0 \quad \forall (i,j) \in [\![1, p_n]\!]^2, \qquad \det(A^{ij}) \geq Cn^{-\vartheta},$$

where det denotes the determinant of a matrix.

Roughly speaking, this will be the favorable situation from a technical point of view since it will be possible to apply a matricial Hoeffding type inequality. It may be possible to slightly relax such a hypothesis using a sub-exponential tail argument. For the sake of simplicity, we have chosen to only restrict our work to the settings of ($\mathbf{H_b}$).

Regardless of the joint law of the random variables $(X^1, \ldots, X^p)$, it is always possible to build an orthonormal basis $(\phi^i_{\boldsymbol{l_i}})_{1\leq l_i\leq L}$ from a bounded (frequency truncated) Fourier basis and, therefore, ($\mathbf{H_b^1}$) is not as restrictive in practice.

Assumption ($\mathbf{H_b^2}$) deals with the high dimensional situation. We are in fact interested in practical situations where the number of variables can be much larger than the number of observations n. Hence, in our mathematical study, the number of variables p_n can grow exponentially fast with the number of observations n. This obviously implies that the collection of subsets u also depends on n and will now be denoted S_n^*. As a consequence, S_n^* also increases rapidly and is much larger than n.

Note that Hypothesis ($\mathbf{H_b^{3,\vartheta}}$) stands for a lower bound of the determinant of the Gram matrices involved in the HOFD. It is shown in [CGP72] that each of these Gram matrices is invertible and, as a result, each $\det(A^{ij})$ is positive. Nevertheless, if $\vartheta = 0$, this hypothesis assumes that such an invertibility is *uniform* over all choices of tensor (i, j). This hypothesis may be too strong for a large number of variables $p_n \to +\infty$ when $\vartheta = 0$. However, when $\vartheta > 0$, Hypothesis ($\mathbf{H_b^{3,\vartheta}}$) drastically relaxes the case $\vartheta = 0$ and becomes very weak. The verification of ($\mathbf{H_b^{3,\vartheta}}$) requires the computation of an order of p_n^2 determinants of size $L^2 \times L^2$. We have checked this assumption in our experiments. However, for very large values of n, this may become impossible from a numerical point of view.

In the following, the parameters ϑ and ξ will be related each other and we will obtain a consistency result of the sparse HOFD up to the condition $\vartheta < \xi/2$. This constraint implicitely limits the size of p_n since $\log p_n = \underset{n\to+\infty}{\mathcal{O}}(n^{1-\xi})$.

Noise Assumption ($\mathbf{H}_{\varepsilon,\mathbf{q}}$) We will assume the noise measurement ε to obtain some bounded moments of sufficiently high order, which is true for Gaussian or bounded noise. This assumption is given by:

($\mathbf{H}_{\varepsilon,\mathbf{q}}$) $\mathbb{E}(|\varepsilon|^q) < \infty$, for one $q \in \mathbb{R}_+$.

Sparsity Assumption ($\mathbf{H}_{\mathbf{s},\alpha}$) The last assumption concerns the sparse representation of the unknown signal described by Y in the basis $(\phi^u_{\boldsymbol{l_u}}(X^u))_u$. Such a hypothesis will be useful to assess the statistical performance of the $\mathbb{L}_2$-Boosting and will be refered to as ($\mathbf{H}_{\mathbf{s},\alpha}$) below. It is legitimate due to our high dimension setting and our motivation to identify the main interactions X^u.

($\mathbf{H}_{\mathbf{s},\alpha}$) There exists $\alpha > 0$ such that the parameter $\boldsymbol{\beta}^0$ satisfies :

$$\|\beta^0\|_1 := \sum_{\substack{u \in S^* \\ |u| \leq d}} \left| \sum_{l_u=1}^{L} \beta^{u,0}_{l_u} \right| = \underset{n \to +\infty}{\mathcal{O}}(n^\alpha).$$

3.2 Main results

We recall below that $\|.\|$ is the $\mathbb{L}_2$ norm on functions decomposed in the orthonormal basis $(\phi^u_{\boldsymbol{l_u}})_u$. We first provide our main result on the efficiency of the EHOFD (Algorithm 5).

Theorem 3.1. *Assume that* ($\mathbf{H_b}$) *holds with* ξ *(resp.* ϑ*) given by* ($\mathbf{H^2_b}$) *(resp.* ($\mathbf{H^{3,\vartheta}_b}$)*) and that there exists a constant* Λ *such that* $\left\|\boldsymbol{\lambda}^{\boldsymbol{l_{ij}}}\right\|_2 \leq \Lambda$ *for any couple* (i,j)*. Then, if* $\vartheta < \xi/2$*, the sequence of estimators* $\left(\hat{\phi}^u_{\boldsymbol{l_u},n_1}\right)_u$ *satisfies:*

$$\sup_{\substack{u \in S^*_n, |u| \leq d \\ \boldsymbol{l_u}}} \left\| \hat{\phi}^u_{\boldsymbol{l_u},n_1} - \phi^u_{\boldsymbol{l_u}} \right\| = \zeta_{n,0} = \mathcal{O}_P(n^{\vartheta - \xi/2}).$$

The proof of Theorem 3.1 is given in Section 4.3. Let us mention the contribution of Theorem 3.1 compared to the results obtained in [CGP72]. Proposition 5.1 of [CGP72] leads to an almost sure convergence of their estimator without any quantitative rate when the number of functions in the HOFD is kept fixed and does not grow with the number of observations n. In contrast, in our high dimensional paradigm, we allow S^*_n to grow with n and also obtain an almost sure result associated with a convergence rate. This will be essential for the derivation of our next result.

Our second main result concerns the $\mathbb{L}_2$-Boosting that recovers the unknown $\tilde{f}$ up to a preprocessing estimation of $(\hat{\phi}^u_{\boldsymbol{l_u},n_1})_{\boldsymbol{l_u},u}$ on a first sample $\mathcal{O}_1$. Such a result is satisfied provided the sparsity Assumptions ($\mathbf{H}_{\mathbf{s},\alpha}$) holds. To the best of our knowledge, such a high dimensional inference with noise in the variables appears to be novel. As already pointed out above, the greedy Boosting seems to be a well-tailored approach to handle noisy dictionaries in comparison to a penalized regression strategy, which relies on a somewhat unverifiable "RIP-type" hypothesis on the learned dictionary.

Theorem 3.2 (Consistency of the $\mathbb{L}_2$-Boosting)**.** *Consider an estimation* $\hat{f}$ *of* $\tilde{f}$ *from an i.i.d. n-sample broken up into* $\mathcal{O}_1 \cup \mathcal{O}_2$*. Assume that functions* $\left(\hat{\phi}^u_{\boldsymbol{l_u},n_1}\right)_{\boldsymbol{l_u},u}$ *are estimated from the first sample* $\mathcal{O}_1$ *under* ($\mathbf{H_b}$) *with* $\vartheta < \xi/2$*, and that there exists a constant* Λ *such that* $\left\|\boldsymbol{\lambda}^{\boldsymbol{l_{ij}}}\right\|_2 \leq \Lambda$ *for any couple* (i,j)*.*

Then, $\hat{f}$ is defined by (III.12) of Algorithm 6 on $\mathcal{O}_2$ as:

$$\hat{f}(X) = G_{k_n}(\bar{f}), \quad \textit{with} \quad \bar{f} = \sum_{\substack{u \in S_n^* \\ |u| \leq d}} \sum_{\boldsymbol{l_u}} \beta^{u,0}_{\boldsymbol{l_u}} \hat{\phi}^u_{\boldsymbol{l_u},n_1}(X^u).$$

If we assume that $(\mathbf{H}_{\mathbf{s},\alpha})$ and $(\mathbf{H}_{\varepsilon,\mathbf{q}})$ are satisfied with $q > 4/\xi$ and $\alpha < \xi/4 - \vartheta/2$, then, a sequence $k_n := C \log n$ exists, with $C < \frac{\xi/2-\vartheta-2\alpha}{2\cdot\log 3}$, such that

$$\left\| \hat{f} - \tilde{f} \right\| \xrightarrow{\mathbb{P}} 0, \textit{when} \quad n \to +\infty.$$

In particular, for Gaussian noises that possess arbitrary large moments, the constraint on q disappears and Theorem 3.2 can be applied as soon as $\xi < 1$.

Let us discuss the asymptotic setting involved in our Theorem. First, our result is a result in probability, rather than in expectation. It is a frequently encountered fact that SOI in expectation are derived with additional assumptions on the coherence of the dictionary; some detailed discussions can be found in [BRT09] and [RT11]. With some coherence and boundedness assumptions, [BRT09] deduced convergence rates of the Lasso estimator in expectation as soon as:

$$\|\beta^0\|_0 \frac{\log(p)}{n} \longrightarrow 0. \tag{III.13}$$

Later, [RT11] extended the study of the Lasso behavior with a result on the Lasso estimator on bounded variables without any coherence assumption and showed a consistency result in probability when:

$$\left\|\beta^0\right\|_1 \sqrt{\frac{\log(p)}{n}} \longrightarrow 0. \tag{III.14}$$

Hence, the rate is damaged by the appearance of the $\sqrt{.}$ in (III.14) in comparison with (III.13). Concerning the Boosting algorithm, [CCAGV14] also obtained consistency results in probability under the asymptotic setting given in (III.14) without a coherence assumption. It should be observed that our results with a noisy dictionary requires that

$$\left(\inf_{i,j} \det(A^{ij})\right)^{-1} \left\|\beta^0\right\|_1^2 \sqrt{\frac{\log p}{n}} \underset{n\to+\infty}{\longrightarrow} 0 \ \text{ a.s.}, \tag{III.15}$$

which is a stronger assumption in comparison with (III.14). From a technical point of view, the asymptotic setting is due to Inequality (III.37) where $\left\|\beta^0\right\|_1^2 \zeta_n$ appears instead of $\left\|\beta^0\right\|_1 \zeta_n$ for Boosting algorithms without noise on the variables (see the proof of Theorem 3.2 in Section 4.4).

In favorable cases where all linear systems defined through the Gram matrices A^{ij} are well conditioned, $\vartheta = 0$ and the condition becomes $\left\|\beta^0\right\|_1^2 \sqrt{\frac{\log p}{n}} \longrightarrow 0$, and there is still a price to pay for the preliminary estimation of the elements of the HOGS. Theorem 3.2 can be applied only for sequences of coefficients such that $\left\|\beta^{u,0}_{\boldsymbol{l_u}}\right\|_1 \lesssim n^{1/4}$. Note also that the degeneracy of the Gram determinants must be strictly larger than $n^{-1/2}$. For example, when $\vartheta = 1/4$, the norm $\left\|\beta^{u,0}_{\boldsymbol{l_u}}\right\|_1$ cannot be larger than $n^{1/8}$.

We briefly describe the proof below and provide the technical details in Section 4.4.

Sketch of Proof of Theorem 3.2. Mimicking the scheme of [Büh06] and [CCAGV14], the proof first consists in defining the theoretical residual of Algorithm 6 at step k as

$$\begin{aligned} R_k(\bar{f}) &= \bar{f} - G_k(\bar{f}) \\ &= \bar{f} - G_{k-1}(\bar{f}) - \gamma \langle Y - G_{k-1}(\bar{f}), \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1} \rangle_{n_2} \cdot \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}. \end{aligned} \tag{III.16}$$

Furthermore, following the work of [CCAGV14], we introduce a *phantom* residual in order to reproduce the behaviour of a deterministic Boosting, studied in [Tem00]. This *phantom* algorithm is the theoretical $\mathbb{L}_2$-Boosting, performed using the randomly chosen elements of the dictionary by Equations (III.11) and (III.12), but updated using the deterministic inner product. The *phantom* residuals $\tilde{R}_k(\bar{f})$, $k \geq 0$, are defined as follows,

$$\begin{cases} \tilde{R}_0(\bar{f}) = \bar{f}, \\ \tilde{R}_k(\bar{f}) = \tilde{R}_{k-1}(\bar{f}) - \gamma \langle \tilde{R}_{k-1}(\bar{f}), \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1} \rangle \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}, \end{cases} \tag{III.17}$$

where $\hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}$ has been selected with Equation (III.11) of Algorithm 6. The aim is to decompose the quantity $\left\| \hat{f} - \tilde{f} \right\|$ to introduce the theoretical residuals and the *phantom* ones,

$$\left\| \hat{f} - \tilde{f} \right\| = \left\| G_{k_n}(\bar{f}) - \tilde{f} \right\| \leq \left\| \bar{f} - \tilde{f} \right\| + \left\| R_{k_n}(\bar{f}) - \tilde{R}_{k_n}(\bar{f}) \right\| + \left\| \tilde{R}_{k_n}(\bar{f}) \right\|. \tag{III.18}$$

We then have to show that each term of the right-hand side of (III.18) converges towards zero in probability. □

4 Proofs of Theorem 3.1 and 3.2

We present here the proofs of Theorems 3.1 and 3.2 of Section 3. Section 4.1 sets the notation that will be used all along this paragraph. Section 4.2 quotes a concentration inequality on random matrices that will be exploited in the rest of the work. We develop the proofs in Section 4.3 and 4.4.

4.1 Notations

Let us first recall some standard notation on matricial norms. For any square matrix M, its spectral radius $\rho(M)$ will refer to as the largest absolute value of the elements of its spectrum, denoted $Sp(M)$:

$$\rho(M) := \max_{\alpha \in Sp(M)} |\alpha|.$$

Moreover, $\|\!|M|\!\|_2$ is the euclidean endomorphism norm and is given by

$$\|\!|M|\!\|_2 := \sqrt{\rho(M^t M)},$$

where ${}^tM$ is the transpose of M. Note that for self-adjoint matrices, $\|\!|M|\!\|_2 = \rho(M)$. At last, the Frobenius norm of M is given by

$$\|M\|_F := \sqrt{\text{Trace}\left({}^tMM\right)}.$$

We finally denote $\preceq$ the semi-definite order on self-adjoint matrices, which is defined for all self-adjoint matrices M_1 and M_2 of size q as:

$$M_1 \preceq M_2 \;\text{ iff } \forall u \in \mathbb{R}^q, \quad {}^tuM_1u \leq {}^tuM_2u.$$

4.2 Hoeffding's type Inequality for random bounded matrices

For the sake of completeness, we quote here Theorem 1.3 of [Tro12].

Theorem 4.1 (Matrix Hoeffding: bounded case)**.** *Consider a finite sequence $(X_k)_{1\leq k\leq n}$ of independent random self-adjoint matrices with dimension d, and let $(A_k)_{1\leq k\leq n}$ a deterministic sequence of self-adjoint matrices. Assume that*

$$\forall 1 \leq k \leq n \qquad \mathbb{E}X_k = 0 \qquad \textit{and} \qquad X_k^2 \preceq A_k^2 \quad a.s.$$

Then, for all $t \geq 0$

$$\mathbb{P}\left(\rho\left(\sum_{k=1}^{n} X_k\right) \geq t\right) \leq d e^{-t^2/8\sigma^2}, \qquad \textit{where} \qquad \sigma^2 = \rho\left(\sum_{k=1}^{n} A_k^2\right).$$

In our work, a more precise concentration inequality such as the Bernstein one (see Theorem 6.1 of [Tro12] is useless since we do not consider any asymptotic on L (the number of basis functions for each variables X^j). Such asymptotic setting is far beyond the scope of this work and we let this problem open for a future work.

4.3 Proof of Theorem 3.1

Consider any subset $u = (u_1, ..., u_t) \in S_n^*$ with $t \geq 1$. If $t = 1$, and $L \geq 1$, set $u = \{i\}$. Then, the **Initialization** step of Algorithm 5 implies that

$$\hat{\phi}^i_{\boldsymbol{l_i},n_1} = \phi^i_{\boldsymbol{l_i}}, \quad \forall\, l_i \in [\![1, L]\!].$$

Therefore, we obviously have that $\sup_{\substack{i\in[\![1,p]\!]\\ l_i\in[\![1,L]\!]}} \left\|\hat{\phi}^i_{\boldsymbol{l_i},n_1} - \phi^i_{\boldsymbol{l_i}}\right\| = 0$.

Now, for $t = 2$, let $u = \{i, j\}$, with $i \neq j \in [\![1, p]\!]$, and $\boldsymbol{l_{ij}} = (\boldsymbol{l_i}, \boldsymbol{l_j}) \in [\![1, L]\!]^2$, and remind that $\phi^{ij}_{\boldsymbol{l_{ij}}}$ is defined as:

$$\phi^{ij}_{\boldsymbol{l_{ij}}}(X^i, X^j) = \phi^i_{\boldsymbol{l_i}}(X^i) \times \phi^j_{\boldsymbol{l_j}}(X^j) + \sum_{k=1}^{L} \lambda^i_{k,\boldsymbol{l_{ij}}} \phi^i_k(X^i) + \sum_{k=1}^{L} \lambda^j_{k,\boldsymbol{l_{ij}}} \phi^j_k(X^j) + C_{\boldsymbol{l_{ij}}},$$

where $\left(C_{\boldsymbol{l_{ij}}}, \left(\lambda^i_{k,\boldsymbol{l_{ij}}}\right)_k, \left(\lambda^j_{k,\boldsymbol{l_{ij}}}\right)_k\right)$ are given as the solutions of (III.5).

When removing $C_{\boldsymbol{l_{ij}}}$, the resolution of (III.5) leads to the resolution of a linear system of the type:

$$A^{ij}\boldsymbol{\lambda}^{\boldsymbol{l_{ij}}} = D^{\boldsymbol{l_{ij}}}, \tag{III.19}$$

where $\boldsymbol{\lambda}^{\boldsymbol{l_{ij}}}$, A^{ij} and $D^{\boldsymbol{l_{ij}}}$ are defined by Equations (III.7) and (III.8).

Consider now $\hat{\phi}^{ij}_{\boldsymbol{l_{ij}},n_1}$ that is decomposed on the dictionary as follows:

$$\hat{\phi}^{ij}_{\boldsymbol{l_{ij}},n_1}(X^i, X^j) = \phi^i_{\boldsymbol{l_i}}(X^i) \times \phi^j_{\boldsymbol{l_j}}(X^j) + \sum_{k=1}^{L} \hat{\lambda}^i_{k,\boldsymbol{l_{ij}},n_1} \phi^i_k(X^i) + \sum_{k=1}^{L} \hat{\lambda}^j_{k,\boldsymbol{l_{ij}},n_1} \phi^j_k(X^j) + \hat{C}^{n_1}_{\boldsymbol{l_{ij}}},$$

where $\left(\hat{C}^{n_1}_{\boldsymbol{l_{ij}}}, \left(\hat{\lambda}^i_{k,\boldsymbol{l_{ij}},n_1}\right)_k, \left(\hat{\lambda}^j_{k,\boldsymbol{l_{ij}},n_1}\right)_k\right)$ are given as solutions of the following *random* equalities:

$$\begin{aligned}
&\langle \hat{\phi}^{ij}_{\boldsymbol{l_{ij}},n_1}, \phi^i_k \rangle_{n_1} = 0, \quad \forall\, k \in [\![1, L]\!],\\
&\langle \hat{\phi}^{ij}_{\boldsymbol{l_{ij}},n_1}, \phi^j_k \rangle_{n_1} = 0, \quad \forall\, k \in [\![1, L]\!],\\
&\langle \hat{\phi}^{ij}_{\boldsymbol{l_{ij}},n_1}, 1 \rangle_{n_1} = 0.
\end{aligned} \tag{III.20}$$

When removing $\hat{C}^{n_1}_{\boldsymbol{l_{ij}}}$, the resolution of (III.20) can also lead to the resolution of a linear system of the type:

$$\hat{A}^{ij}_{n_1}\hat{\boldsymbol{\lambda}}^{\boldsymbol{l_{ij}}}_{n_1} = \hat{D}^{\boldsymbol{l_{ij}}}_{n_1}, \tag{III.21}$$

where

$$\hat{\boldsymbol{\lambda}}^{\boldsymbol{l_{ij}}}_{n_1} = \begin{pmatrix} \hat{\lambda}^i_{1,\boldsymbol{l_{ij}},n_1} \\ \vdots \\ \hat{\lambda}^i_{L,\boldsymbol{l_{ij}},n_1} \\ \hat{\lambda}^j_{1,\boldsymbol{l_{ij}},n_1} \\ \vdots \\ \hat{\lambda}^j_{L,\boldsymbol{l_{ij}},n_1} \end{pmatrix},$$

and $\hat{A}^{ij}_{n_1}$ (*resp.* $\hat{D}^{\boldsymbol{l_{ij}}}_{n_1}$) are obtained from A^{ij} (*resp.* $D^{\boldsymbol{l_{ij}}}$) by changing the theoretical inner product by its empirical version.

Remark 6. *Remark that A^{ij} depends on (i,j) as well as $\boldsymbol{\lambda}^{\boldsymbol{l_{ij}}}$ and $D^{\boldsymbol{l_{ij}}}$ depend on (i,j) and $\boldsymbol{l_{ij}}$, but we will deliberately omit these indexes in the sequel for the sake of convenience (when no confusion is possible). For instance, when a couple (i,j) is handled, we will frequently use the notation $A, \boldsymbol{\lambda}, D, C, \lambda^i_k, \lambda^j_k$ instead of $A^{ij}, \boldsymbol{\lambda}^{\boldsymbol{l_{ij}}}, D^{\boldsymbol{l_{ij}}}, C_{\boldsymbol{l_{ij}}}, \lambda^i_{k,\boldsymbol{l_{ij}}}$ and $\lambda^j_{k,\boldsymbol{l_{ij}}}$. This will be also the case for the estimators $\hat{A}_{n_1}, \hat{\boldsymbol{\lambda}}_{n_1}, \hat{D}_{n_1}, \hat{C}^{n_1}, \hat{\lambda}^i_{k,n_1}$ and $\hat{\lambda}^j_{k,n_1}$.*

Then, the following useful lemma compares the two matrices $\hat{A}_{n_1}$ and A.

Lemma 4.1. *Under Assumption $(\mathbf{H_b})$, and for any ξ given by $(\mathbf{H^2_b})$, one has*

$$\sup_{1\le i,j\le p_n} \left|\!\left|\!\left| \hat{A}_{n_1} - A \right|\!\right|\!\right|_2 = \mathcal{O}_P(n^{-\xi/2}).$$

Proof of Lemma 4.1. First consider one couple (i,j) and note that $\left|\!\left|\!\left| \hat{A}_{n_1} - A \right|\!\right|\!\right|_2 = \rho(\hat{A}_{n_1} - A)$, since $\hat{A}_{n_1} - A$ is self-adjoint. To obtain a concentration inequality on the matricial norm $\left|\!\left|\!\left| \hat{A}_{n_1} - A \right|\!\right|\!\right|_2$, we use the result of [Tro12](see Theorem 4.1), which give concentration inequalities for the largest eigenvalue of self-adjoint matrices.

Remark that $\hat{A}_{n_1} - A$ can be written as follows:

$$\hat{A}_{n_1} - A = \frac{1}{n_1}\sum_{r=1}^{n_1} \Theta_{r,ij}, \quad \Theta_{r,ij} = \begin{pmatrix} \Theta^{ii}_r & \Theta^{ij}_r \\ {}^t\Theta^{ij}_r & \Theta^{jj}_r \end{pmatrix}, \quad \forall\, r \in [\![1, n_1]\!],$$

where, for all $k, m \in [\![1, L]\!]$, $(\Theta^{i_1 i_2}_r)_{k,m} = \phi^{i_1}_k(X^{i_1}_r)\phi^{i_2}_m(X^{i_2}_r) - \mathbb{E}[\phi^{i_1}_k(X^{i_1})\phi^{i_2}_m(X^{i_2})]$ with $i_1, i_2 \in \{i,j\}$. Since the observations $(X_r)_{r=1,\cdots,n_1}$ are independent, $\Theta_{1,ij}, \cdots, \Theta_{n_1,ij}$ is a sequence of independent, random, centered, and self-adjoint matrices. Moreover, for all $u \in \mathbb{R}^{2L}$, and all $r \in [\![1, n_1]\!]$,

$${}^t u\Theta^2_{r,ij}u = \|\Theta_{r,ij}u\|_2^2 \le \|u\|_2^2 \|\Theta_{r,ij}\|_F^2,$$

where

$$\begin{aligned} \|\Theta_{r,ij}\|_F^2 &\le (2L)^2 \left(\max_{k,m\in[\![1,L]\!]} |(\Theta_{r,ij})_{k,m}| \right)^2 \\ &\le (2L)^2 \left(\max_{\substack{k,m\in[\![1,L]\!] \\ i_1,i_2\in\{i,j\}}} \left| \phi^{i_1}_k(x^r_{i_1})\phi^{i_2}_m(x^r_{i_2}) - \mathbb{E}[\phi^{i_1}_k(X_{i_1})\phi^{i_2}_m(X_{i_2})] \right| \right)^2 \\ &\le 16L^2M^4 \quad \text{by } (\mathbf{H^1_b}). \end{aligned}$$

We then deduce that each element of the sum satisfies $X_{l,ij}^2 \preceq 16L^2M^4\mathrm{I}_{L^2}$, where I_{L^2} denotes the identity matrix of size L^2.

Applying now the Hoeffding's type Inequality stated as Theorem 4.1 of [Tro12] to our sequence $\Theta_{1,ij}, \cdots, \Theta_{n_1,ij}$, with $\sigma^2 = 16n_1L^2M^4$, we then obtain that

$$\forall t \geq 0 \qquad \mathbb{P}\left(\rho\left(\frac{1}{n_1}\sum_{r=1}^{n_1}\Theta_{r,ij}\right) \geq t\right) \leq 2Le^{-\frac{(n_1t)^2}{8\sigma^2}},$$

Considering now the whole set of estimators $\hat{A}_{n_1}$, we obtain

$$\forall t \geq 0 \qquad \mathbb{P}\left(\sup_{1\leq i,j\leq p_n}\rho\left(\frac{1}{n_1}\sum_{r=1}^{n_1}\Theta_{r,ij}\right) \geq t\right) \leq 2Lp_n^2e^{-\frac{(n_1t)^2}{8\sigma^2}},$$

Now, we take $t = \gamma n_1^{-\xi/2}$, where $\gamma > 0$, and $0 < \xi \leq 1$ given in $(\mathbf{H_b^2})$. Then, the following inequality holds:

$$\mathbb{P}\left(\sup_{1\leq i,j\leq p_n}\rho\left(\hat{A}_{n_1} - A\right) \geq \gamma n^{-\xi/2}\right) \leq 2Lp_n^2e^{-\frac{{n_1}^{1-\xi}\gamma^2}{128L^2M^4}}. \tag{III.22}$$

Since $p_n = \underset{n\to+\infty}{\mathcal{O}}(\exp(Cn^{1-\xi}))$ by Assumption $(\mathbf{H_b^2})$, the right-hand side of the previous inequality becomes arbitrarily small for n sufficiently large and $\gamma > 0$ large enough. The end of the proof follows using Inequality (III.22). □

Similarly, we can show that the estimated quantity $\hat{D}_{n_1}$ is not far from the theoretical D with high probability.

Lemma 4.2. *Under Assumptions* $(\mathbf{H_b})$*, and for any* ξ *given by* $(\mathbf{H_b^2})$*, one has*

$$\sup_{i,j,\boldsymbol{l_{ij}}}\left\|\hat{D}_{n_1} - D\right\|_2 = \mathcal{O}_P(n^{-\xi/2}).$$

Proof of Lemma 4.2. First consider one couple (i,j). We aim to apply another concentration inequality on $\left\|\hat{D}_{n_1} - D\right\|_2$. Remark that $\left\|\hat{D}_{n_1} - D\right\|_2$ can be written as:

$$\begin{aligned}
\left\|\hat{D}_{n_1} - D\right\|_2 &= \left(\sum_{k=1}^{L}\left(\langle\phi_{\boldsymbol{l_i}}^i \times \phi_{\boldsymbol{l_j}}^j, \phi_k^i\rangle_{n_1} - \langle\phi_{\boldsymbol{l_i}}^i \times \phi_{\boldsymbol{l_j}}^j, \phi_k^i\rangle\right)^2 + \right.\\
&\qquad \left.\sum_{k=1}^{L}\left(\langle\phi_{\boldsymbol{l_i}}^i \times \phi_{\boldsymbol{l_j}}^j, \phi_k^j\rangle_{n_1} - \langle\phi_{\boldsymbol{l_i}}^i \times \phi_{\boldsymbol{l_j}}^j, \phi_k^j\rangle\right)^2\right)^{1/2}\\
&\leq \sum_{k=1}^{L}\left|\frac{1}{n_1}\sum_{r=1}^{n_1}\phi_{\boldsymbol{l_i}}^i(X_r^i)\phi_{\boldsymbol{l_j}}^j(X_r^j)\phi_k^i(X_r^i) - \langle\phi_{\boldsymbol{l_i}}^i \times \phi_{\boldsymbol{l_j}}^j, \phi_k^i\rangle\right| +\\
&\qquad \sum_{k=1}^{L}\left|\frac{1}{n_1}\sum_{r=1}^{n_1}\phi_{\boldsymbol{l_i}}^i(X_r^i)\phi_{\boldsymbol{l_j}}^j(X_r^j)\phi_k^j(X_r^j) - \langle\phi_{\boldsymbol{l_i}}^i \times \phi_{\boldsymbol{l_j}}^j, \phi_k^j\rangle\right|.
\end{aligned}$$

Now, Bernstein's Inequality (see Chapter II for instance) implies that, for all $\gamma > 0$,

$$\begin{aligned}
\mathbb{P}\left(n_1^{\xi/2}\left\|\hat{D}_{n_1} - D\right\|_2 \geq \gamma\right) &\leq \mathbb{P}\left(n_1^{\xi/2}\sum_{k=1}^{L}\left|\frac{1}{n_1}\sum_{r=1}^{n_1}\phi^i_{\boldsymbol{l_i}}(X^i_r)\phi^j_{\boldsymbol{l_j}}(X^j_r)\phi^i_k(X^i_r) - \langle\phi^i_{\boldsymbol{l_i}} \times \phi^j_{\boldsymbol{l_j}}, \phi^i_k\rangle\right| > \gamma/2\right) \\
&+ \mathbb{P}\left(n_1^{\xi/2}\sum_{k=1}^{L}\left|\frac{1}{n_1}\sum_{r=1}^{n_1}\phi^i_{\boldsymbol{l_i}}(X^i_r)\phi^j_{\boldsymbol{l_j}}(X^j_r)\phi^i_k(X^i_r) - \langle\phi^i_{\boldsymbol{l_i}} \times \phi^j_{\boldsymbol{l_j}}, \phi^i_k\rangle\right| > \gamma/2\right) \\
&\leq 4L\exp\left(-\frac{1}{8}\frac{\gamma^2 n_1^{1-\xi}}{M^6 + M^3\gamma/6n_1^{-\xi/2}}\right),
\end{aligned}$$

which gives:

$$\mathbb{P}\left(\sup_{i,j,\boldsymbol{l_{ij}}}\left\|\hat{D}_{n_1} - D\right\|_2 \geq \gamma n_1^{-\xi/2}\right) \leq 4L \times L^2 p_n^2 \exp\left(-\frac{1}{8}\frac{\gamma^2 {n_1}^{1-\xi}}{M^6 + M^3\gamma/6{n_1}^{-\xi/2}}\right). \tag{III.23}$$

Now, since $n_1 = n/2$, Assumption $(\mathbf{H_b^2})$ implies that the right-hand side of Inequality (III.23) can also become arbitrarily small for n sufficiently large, which concludes the proof. □

The next lemma then compares the estimated $\hat{\boldsymbol{\lambda}}_{n_1}$ with $\boldsymbol{\lambda}$.

Lemma 4.3. *Under Assumptions* $(\mathbf{H_b})$*, with* $\vartheta < \xi/2$*, we have:*

$$\sup_{i,j,\boldsymbol{l_{ij}}}\left\|\hat{\boldsymbol{\lambda}}_{n_1} - \boldsymbol{\lambda}\right\|_2 = \mathcal{O}_P(n^{\vartheta - \xi/2}).$$

Proof of Lemma 4.3. Fix any couple (i, j), $\boldsymbol{\lambda}$ and $\hat{\boldsymbol{\lambda}}_{n_1}$ satisfy Equations (III.19) and (III.21). Hence,

$$\begin{aligned}
A(\hat{\boldsymbol{\lambda}}_{n_1} - \boldsymbol{\lambda}) - A\hat{\boldsymbol{\lambda}}_{n_1} &= -D = \hat{D}_{n_1} - D - \hat{D}_{n_1} \\
&= (\hat{D}_{n_1} - D) - \hat{A}_{n_1}\hat{\boldsymbol{\lambda}}_{n_1},
\end{aligned}$$

which can be equivalently rewritten as:

$$A(\hat{\boldsymbol{\lambda}}_{n_1} - \boldsymbol{\lambda}) = (\hat{D}_{n_1} - D) + (A - \hat{A}_{n_1})\hat{\boldsymbol{\lambda}}_{n_1}.$$

Since the matrix A is positive definite, it follows that

$$\hat{\boldsymbol{\lambda}}_{n_1} - \boldsymbol{\lambda} = A^{-1}[(A - \hat{A}_{n_1})\hat{\boldsymbol{\lambda}}_{n_1}] + A^{-1}(\hat{D}_{n_1} - D),$$

or

$$\begin{aligned}
&\hat{\boldsymbol{\lambda}}_{n_1} - \boldsymbol{\lambda} = A^{-1}(A - \hat{A}_{n_1})(\hat{\boldsymbol{\lambda}}_{n_1} - \boldsymbol{\lambda}) + A^{-1}(A - \hat{A}_{n_1})\boldsymbol{\lambda} + A^{-1}(\hat{D}_{n_1} - D) \\
\Leftrightarrow\quad &\left(\mathrm{I} - A^{-1}(A - \hat{A}_{n_1})\right)(\hat{\boldsymbol{\lambda}}_{n_1} - \boldsymbol{\lambda}) = A^{-1}(A - \hat{A}_{n_1})\boldsymbol{\lambda} + A^{-1}(\hat{D}_{n_1} - D).
\end{aligned} \tag{III.24}$$

Remark that $\left\|\hat{A}_{n_1} - A\right\|_2 = \mathcal{O}_P(n^{-\xi/2})$ by Lemma 4.1. Hence, with high probability and for n large enough $\mathrm{I} - A^{-1}(A - \hat{A}_{n_1})$ is invertible, and Inequality (III.24) can be rewritten as:

$$\hat{\boldsymbol{\lambda}}_{n_1} - \boldsymbol{\lambda} = \left(\mathrm{I} - A^{-1}(A - \hat{A}_{n_1})\right)^{-1}\left(A^{-1}(A - \hat{A}_{n_1})\boldsymbol{\lambda} + A^{-1}(\hat{D}_{n_1} - D)\right).$$

We then deduce that,

$$\begin{aligned}\left\|\hat{\boldsymbol{\lambda}}_{n_1}-\boldsymbol{\lambda}\right\|_2 &\leq \left|\!\left|\!\left|\left(\mathrm{I}-A^{-1}(A-\hat{A}_{n_1})\right)^{-1}\right|\!\right|\!\right|_2 \times \left(\left|\!\left|\!\left|A^{-1}[A-\hat{A}_{n_1}]\right|\!\right|\!\right|_2 \|\boldsymbol{\lambda}\|_2 + \left\|A^{-1}(\hat{D}_{n_1}-D)\right\|_2\right)\\ &\leq \left|\!\left|\!\left|\left(\mathrm{I}-A^{-1}(A-\hat{A}_{n_1})\right)^{-1}\right|\!\right|\!\right|_2 \\ &\quad \times \left(\left|\!\left|\!\left|A^{-1}\right|\!\right|\!\right|_2 \left|\!\left|\!\left|A-\hat{A}_{n_1}\right|\!\right|\!\right|_2 \|\boldsymbol{\lambda}\|_2 + \left|\!\left|\!\left|A^{-1}\right|\!\right|\!\right|_2 \left\|\hat{D}_{n_1}-D\right\|_2\right). \end{aligned} \tag{III.25}$$

A uniform bound for $\left|\!\left|\!\left|A^{-1}\right|\!\right|\!\right|_2$ (over all couples (i,j)) can be easily obtain since A (and obviously A^{-1}) is Hermitian.

$$\left|\!\left|\!\left|A^{-1}\right|\!\right|\!\right|_2 \leq \max_{(i',j')\in[\![1,p_n]\!]^2} \rho\left(\left(A^{i'j'}\right)^{-1}\right).$$

Simple algebra then yields

$$\rho\left(\left(A^{i'j'}\right)^{-1}\right) \leq \mathrm{Trace}\left(\left(A^{i'j'}\right)^{-1}\right) = \frac{\mathrm{Trace}\left({}^t\mathrm{Com}(A^{i'j'})\right)}{\det(A^{i'j'})} = \frac{1}{\det(A^{i'j'})}\sum_{k=1}^{2L}\mathrm{Com}(A^{i'j'})_k^k,$$

where $\mathrm{Com}(A^{ij})$ is the cofactor matrix associated to A^{ij}. Now, recall the classical inequality (that can be found in [Bul98]): for any symetric definite positive matrix squared M of size $q\times q$

$$\det(M) \leq \prod_{k=1}^{q}\left|M_k^k\right|.$$

This last inequality applied to the determinant involved in $\mathrm{Com}(A^{i'j'})_k^k$ associated with $(\mathbf{H_b^1})$ implies

$$\forall k\in[\![1,2L]\!], \qquad \left|\mathrm{Com}(A^{i'j'})_k^k\right| \leq \left(M^2\right)^{2L-1}.$$

We then deduce from $(\mathbf{H_b^{3,\vartheta}})$ that there exists a constant $C>0$ such that:

$$\begin{aligned}\left|\!\left|\!\left|A^{-1}\right|\!\right|\!\right|_2 &\leq \max_{(i,j)\in[\![1,p_n]\!]^2}\frac{2LM^{4L-2}}{\det(A^{i'j'})}\\ &\leq 2C^{-1}LM^{4L-2}n^{\vartheta}. \end{aligned} \tag{III.26}$$

Similarly, if we denote $\Delta_{n_1}=A-\hat{A}_{n_1}$, we have

$$\begin{aligned}\left|\!\left|\!\left|\left(\mathrm{I}-A^{-1}(A-\hat{A}_{n_1})\right)^{-1}\right|\!\right|\!\right|_2 &= \rho\left(\left(I-A^{-1}\Delta_{n_1}\right)^{-1}\right)\\ &= \max_{\alpha\in Sp(A^{-1}\Delta_{n_1})}\frac{1}{|1-\alpha|},\end{aligned}$$

using the fact that $A-\hat{A}_{n_1}$ is self-adjoint. We have seen that $\rho(A^{-1})\leq 2C^{-1}LM^{4L-2}n^{\vartheta}$ and Lemma 4.1 yields $\rho\left(\Delta_{n_1}\right)=\mathcal{O}_P(n^{-\xi/2})$. As a consequence, we have

$$\max_{\alpha\in Sp(A^{-1}\Delta_{n_1})}|\alpha| \leq \rho(A^{-1})\rho\left(\Delta_{n_1}\right) = \mathcal{O}_P(n^{\vartheta-\xi/2}).$$

At last, remark that

$$\max_{\alpha\in Sp(A^{-1}\Delta_{n_1})}\frac{1}{|1-\alpha|}-1 = \max_{\alpha\in Sp(A^{-1}\Delta_{n_1})}\frac{1-|1-\alpha|}{|1-\alpha|}.$$

We know that for n large enough, each absolute value of $\alpha \in Sp(A^{-1}\Delta_{n_1})$ becomes smaller than $1/2$ with a probability tending to one. Hence, we have with probability tending to one

$$\max_{\alpha \in Sp(A^{-1}\Delta_{n_1})} \left| \frac{1-|1-\alpha|}{|1-\alpha|} \right| \leq \max_{\alpha \in Sp(A^{-1}\Delta_{n_1})} \frac{|\alpha|}{1-\alpha} \leq 2\rho(A^{-1}\Delta_{n_1}).$$

Since $\rho(A^{-1}\Delta_{n_1}) = \mathcal{O}_P(n^{\vartheta-\xi/2})$, we deduce

$$\sup_{i,j,\boldsymbol{l_{ij}}} \left\| \left(\mathrm{I} - A^{-1}(A - \hat{A}_{n_1}) \right)^{-1} \right\|_2 \leq 1 + 2LM^{4L-2}C^{-1}\mathcal{O}_P(n^{\vartheta-\xi/2}). \tag{III.27}$$

To conclude the proof, we can now apply the same argument as the one used in Lemmas 4.1 and 4.2 with Bernstein's Inequality, using Equations (III.26) and (III.27), and the assumption on the uniform bound $\|\boldsymbol{\lambda}\|_2 \leq \Lambda$ over all the couples (i,j) for the norm $\left\|\boldsymbol{\lambda}^{\boldsymbol{l_{ij}}}\right\|_2$. □

The last lemma finally compares the constant $\hat{C}^{n_1}$ with C.

Lemma 4.4. *Under Assumptions* $(\mathbf{H_b})$*, we have:*

$$\sup_{i,j,\boldsymbol{l_{ij}}} \left| \hat{C}^{n_1} - C \right| = \mathcal{O}_P(n^{-\xi/2}).$$

Proof of Lemma 4.4. For any couple (i,j), remark that constants $\hat{C}^{n_1}$ and C satisfy:

$$C = -\langle \phi^i_{\boldsymbol{l_i}} \times \phi^j_{\boldsymbol{l_j}}, 1 \rangle \quad \text{and} \quad \hat{C}^{n_1} = -\langle \phi^i_{\boldsymbol{l_i}} \times \phi^j_{\boldsymbol{l_j}}, 1 \rangle_{n_1}.$$

If we denote

$$\Delta_{i,j,\boldsymbol{l_{ij}}} := \frac{1}{n_1} \sum_{r=1}^{n_1} \phi^i_{\boldsymbol{l_i}}(X^i_r)\phi^j_{\boldsymbol{l_j}}(X^j_r) - \mathbb{E}(\phi^i_{\boldsymbol{l_i}}(X^i)\phi^j_{\boldsymbol{l_j}}(X^j)),$$

we can apply again Bernstein's Inequality on $(\phi^i_{\boldsymbol{l_i}}(X^i_r)\phi^j_{\boldsymbol{l_j}}(X^j_r))_{r=1,\cdots,n_1}$. From $(\mathbf{H^1_b})$, these independent random variables are bounded by M^2 and

$$\begin{aligned}
\mathbb{P}\left(\sup_{i,j,\boldsymbol{l_{ij}}} \left|\Delta_{i,j,\boldsymbol{l_{ij}}}\right| \geq \gamma n_1^{-\xi/2} \right) &\leq \sum_{i,j,\boldsymbol{l_{ij}}} \mathbb{P}\left(\left|\Delta_{i,j,\boldsymbol{l_{ij}}}\right| \geq \gamma n_1^{-\xi/2} \right) \\
&\leq \sum_{i,j,\boldsymbol{l_{ij}}} 2\exp\left(-\frac{1}{2} \frac{\gamma^2 n_1^{1-\xi}}{M^4 + M^2\gamma/3 n_1^{-\xi/2}} \right) \\
&\leq 2L^2 p_n^2 \exp\left(-\frac{1}{2} \frac{\gamma^2 n_1^{1-\xi}}{M^4 + M^2\gamma/3 n_1^{-\xi/2}} \right).
\end{aligned}$$

Under Assumption $(\mathbf{H^2_b})$, the right-hand side of this inequality can be arbitrarly small for n large enough, which ends the proof. □

To finish the proof of Theorem 3.1, remark that:

$$\begin{aligned}
\left\| \hat{\phi}^{ij}_{\boldsymbol{l_{ij}},n_1} - \phi^{ij}_{\boldsymbol{l_{ij}}} \right\| &= \left\| \sum_{k=1}^{L} \left(\hat{\lambda}^i_{k,n_1} - \lambda^i_k \right) \phi^i_k + \sum_{k=1}^{L} \left(\hat{\lambda}^j_{k,n_1} - \lambda^j_k \right) \phi^j_k + (\hat{C}^{n_1} - C) \right\| \\
&\leq \underbrace{\left\| \sum_{k=1}^{L} \left(\hat{\lambda}^i_{k,n_1} - \lambda^i_k \right) \phi^i_k + \sum_{k=1}^{L} \left(\hat{\lambda}^j_{k,n_1} - \lambda^j_k \right) \phi^j_k \right\|}_{=I} + \left| \hat{C}^{n_1} - C \right|.
\end{aligned}$$

Moreover,

$$\begin{aligned} I^2 &= \int \left(\sum_{k=1}^{L} \left(\hat{\lambda}^i_{k,n_1} - \lambda^i_k \right) \phi^i_k + \sum_{k=1}^{L} \left(\hat{\lambda}^j_{k,n_1} - \lambda^j_k \right) \phi^j_k \right)^2 p_{X^i,X^j} dx_i dx_j \\ &= \underbrace{\int \left(\sum_{k=1}^{L} \left(\hat{\lambda}^i_{k,n_1} - \lambda^i_k \right) \phi^i_k \right)^2 p_{X^i} dx_i}_{=I_1} + \underbrace{\int \left(\sum_{k=1}^{L} \left(\hat{\lambda}^j_{k,n_1} - \lambda^j_k \right) \phi^j_k \right)^2 p_{X^j} dx_j}_{=I_2} \\ &\quad \underbrace{+ 2 \int \left(\sum_{k=1}^{L} \left(\hat{\lambda}^i_{k,n_1} - \lambda^i_k \right) \phi^i_k \right) \left(\sum_{k=1}^{L} \left(\hat{\lambda}^i_{k,n_1} - \lambda^i_k \right) \phi^i_k \right) p_{X^i,X^j} dx_i dx_j}_{=I_3} . \end{aligned}$$

Using the inequality $2ab \leq a^2 + b^2$, we thus deduce that $I_3 \leq I_1 + I_2$, and

$$\begin{aligned} I_1 &= \int \sum_{k=1}^{L} \sum_{m=1}^{L} \left(\hat{\lambda}^i_{k,n_1} - \lambda^i_k \right) \left(\hat{\lambda}^i_{m,n_1} - \lambda^i_m \right) \phi^i_k(x_i) \phi^i_m(x_i) p_{X^i} dx_i \\ &= \sum_{k=1}^{L} \left(\hat{\lambda}^i_{k,n_1} - \lambda^i_k \right)^2 \quad \text{by orthonormality.} \end{aligned}$$

And the same equality is satisfied for I_2: $I_2 = \sum_{k=1}^{L} \left(\hat{\lambda}^j_{k,n_1} - \lambda^j_k \right)^2$.

Consequently, we obtain

$$\begin{aligned} \left\| \hat{\phi}^{ij}_{\boldsymbol{l_{ij}},n_1} - \phi^{ij}_{\boldsymbol{l_{ij}}} \right\| &\leq \sqrt{2 \left[\sum_{k=1}^{L} \left(\hat{\lambda}^i_{k,n_1} - \lambda^i_k \right)^2 + \sum_{k=1}^{L} \left(\hat{\lambda}^j_{k,n_1} - \lambda^j_k \right)^2 \right]} + \left| \hat{C}^{n_1} - C \right| \\ &= \sqrt{2} \left\| \hat{\boldsymbol{\lambda}}_{n_1} - \boldsymbol{\lambda} \right\|_2 + \left| \hat{C}^{n_1} - C \right| . \end{aligned}$$

The end of the proof follows with Lemmas 4.3 and 4.4.

4.4 Proof of Theorem 3.2

We recall first that $\langle , \rangle$ denotes the theoretical inner product based on the law P_X (and $\|\|$ is the derived Hilbertian norm). A careful inspection of the Gram-Schmidt procedure used to build the HOFD shows that

$$M^* := \sup_{u, \boldsymbol{l_u}} \left\| \phi^u_{\boldsymbol{l_u}}(X^u) \right\|_\infty < \infty,$$

provided that ($\mathbf{H^1_b}$) holds.

Now, remark that the EHOFD is obtained through the first sample $\mathcal{O}_1$ which determines the first empirical inner product $\langle , \rangle_{n_1}$ although the $\mathbb{L}^2$-Boosting depends on the second sample $\mathcal{O}_2$. Indeed, $\mathcal{O}_2$ determines the second empirical inner product $\langle , \rangle_{n_2}$. Hence, $\langle , \rangle_{n_2}$ uses observations which are *independent* to the ones used to build the HOFD.

We begin this section with a lemma which establishes that the estimated functions $\hat{\phi}^u_{\boldsymbol{l_u},n_1}$ (which result in the EHOFD) are bounded.

Lemma 4.5. *Under Assumption* $(\mathbf{H_b})$*, define*

$$N_{n_1} := \sup_{u,\boldsymbol{l_u}} \left\| \hat{\phi}^u_{\boldsymbol{l_u},n_1}(X^u) \right\|_\infty .$$

Then, we have:

$$N_{n_1} - M^* = \mathcal{O}_P(n^{\vartheta-\xi/2}).$$

Proof of Lemma 4.5. Using the decomposition of $\hat{\phi}^u_{\boldsymbol{l_u},n_1}$ on the dictionary, Assumption $(\mathbf{H^2_b})$ and Cauchy-Schwarz Inequality, there exists a fixed constant $C > 0$ such that for all $u \in S$, $\boldsymbol{l_u}$:

$$\forall x \in \mathbb{R}^p \qquad \left|\hat{\phi}^u_{\boldsymbol{l_u},n_1}(x) - \phi^u_{\boldsymbol{l_u}}(x)\right| \leq CM\sqrt{L}\sqrt{\left\|\hat{\boldsymbol{\lambda}}_{n_1} - \boldsymbol{\lambda}\right\|_2} + \left\|\hat{C}^{n_1}_{\boldsymbol{l_u}} - C_{\boldsymbol{l_u}}\right\|.$$

The conclusion then follows using Lemmas 4.3 and 4.4. □

We now present a key lemma which compares the elements $(\phi^u_{\boldsymbol{l_u}})_{\boldsymbol{l_u},u}$ with their estimated version $(\hat{\phi}^u_{\boldsymbol{l_u},n_1})_{\boldsymbol{l_u},u}$.

Lemma 4.6. *Assume that* $(\mathbf{H_b})$ *holds with* $\xi \in (0,1)$*, that the noise* ε *satisfies* $(\mathbf{H_{\varepsilon,q}})$ *with* $q > 4/\xi$ *and that* $(\mathbf{H_{s,\alpha}})$ *is fullfilled. Then, the following inequalities hold,*

(*i*)

$$\sup_{u,v,\boldsymbol{l_u},\boldsymbol{l_v}} \left|\langle\hat{\phi}^u_{\boldsymbol{l_u},n_1}, \hat{\phi}^v_{\boldsymbol{l_v},n_1}\rangle - \langle\phi^u_{\boldsymbol{l_u}}, \phi^v_{\boldsymbol{l_v}}\rangle\right| = \zeta_{n,1} = \mathcal{O}_P(n^{\vartheta-\xi/2}),$$

(*ii*)

$$\sup_{u,v,\boldsymbol{l_u},\boldsymbol{l_v}} \left|\langle\hat{\phi}^u_{\boldsymbol{l_u},n_1}, \hat{\phi}^v_{\boldsymbol{l_v},n_1}\rangle_{n_2} - \langle\phi^u_{\boldsymbol{l_u}}, \phi^v_{\boldsymbol{l_v}}\rangle\right| = \zeta_{n,2} = \mathcal{O}_P(n^{\vartheta-\xi/2}),$$

(*iii*)

$$\sup_{u,v,\boldsymbol{l_u},\boldsymbol{l_v}} \left|\langle\varepsilon, \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle_{n_2}\right| = \zeta_{n,3} = \mathcal{O}_P(n^{-\xi/2}),$$

(*iv*)

$$\sup_{u,\boldsymbol{l_u}} \left|\langle\tilde{f}, \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle_{n_2} - \langle\tilde{f}, \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle\right| = \|\beta^0\|_1 \mathcal{O}_P(n^{-\xi/2}).$$

In the sequel, we will denote $\zeta_n := \max_{i\in[\![1,3]\!]}\{\zeta_{n,i}\}$.

Proof of Lemma 4.6. **Assertion** (*i*) Let $u, v \in S$, $\boldsymbol{l_u} \in [\![1,L]\!]^{|u|}$ and $\boldsymbol{l_v} \in [\![1,L]\!]^{|v|}$. Then, we have

$$\begin{aligned}
\left|\langle\hat{\phi}^u_{\boldsymbol{l_u},n_1}, \hat{\phi}^v_{\boldsymbol{l_v},n_1}\rangle - \langle\phi^u_{\boldsymbol{l_u}}, \phi^v_{\boldsymbol{l_v}}\rangle\right| &\leq \left|\langle\hat{\phi}^u_{\boldsymbol{l_u},n_1} - \phi^u_{\boldsymbol{l_u}}, \hat{\phi}^v_{\boldsymbol{l_v},n_1}\rangle - \langle\phi^u_{\boldsymbol{l_u}}, \phi^v_{\boldsymbol{l_v}} - \hat{\phi}^v_{\boldsymbol{l_v},n_1}\rangle\right| \\
&\leq \left\|\hat{\phi}^u_{\boldsymbol{l_u},n_1} - \phi^u_{\boldsymbol{l_u}}\right\| \left\|\hat{\phi}^v_{\boldsymbol{l_v},n_1}\right\| + \left\|\phi^u_{\boldsymbol{l_u}}\right\| \left\|\hat{\phi}^v_{\boldsymbol{l_v},n_1} - \phi^v_{\boldsymbol{l_v}}\right\| \\
&\leq \left\|\hat{\phi}^u_{\boldsymbol{l_u},n_1} - \phi^u_{\boldsymbol{l_u}}\right\| \left(\left\|\hat{\phi}^v_{\boldsymbol{l_v},n_1} - \phi^v_{\boldsymbol{l_v}}\right\| + 1\right) + \left\|\hat{\phi}^v_{\boldsymbol{l_v},n_1} - \phi^v_{\boldsymbol{l_v}}\right\|,
\end{aligned}$$

and the conclusion holds applying Theorem 3.1.

Assertion (*ii*) We breakdown the term in two parts:

$$\left|\langle\hat{\phi}^u_{\boldsymbol{l_u},n_1}, \hat{\phi}^v_{\boldsymbol{l_v},n_1}\rangle_{n_2} - \langle\phi^u_{\boldsymbol{l_u}}, \phi^v_{\boldsymbol{l_v}}\rangle\right| \leq \underbrace{\left|\langle\hat{\phi}^u_{\boldsymbol{l_u},n_1}, \hat{\phi}^v_{\boldsymbol{l_v},n_1}\rangle_{n_2} - \langle\hat{\phi}^u_{\boldsymbol{l_u},n_1}, \hat{\phi}^v_{\boldsymbol{l_v},n_1}\rangle\right|}_{=I} + \underbrace{\left|\langle\hat{\phi}^u_{\boldsymbol{l_u},n_1}, \hat{\phi}^v_{\boldsymbol{l_v},n_1}\rangle - \langle\phi^u_{\boldsymbol{l_u}}, \phi^v_{\boldsymbol{l_v}}\rangle\right|}_{=II}.$$

Assertion (i) implies that,

$$\sup_{u,v,\boldsymbol{l_u},\boldsymbol{l_v}} |II| = \mathcal{O}_P(n^{\vartheta-\xi/2}).$$

To control $\sup_{u,v,\boldsymbol{l_u},\boldsymbol{l_v}} |I|$, we use Bernstein's inequality to the family of independent random variables $\left(\hat{\phi}^u_{\boldsymbol{l_u},n_1}(X^u_s)\hat{\phi}^v_{\boldsymbol{l_v},n_1}(X^v_s)\right)_{s=1\ldots n_2}$ and we denote

$$\Delta_{u,v,\boldsymbol{l_u},\boldsymbol{l_v}} = \left|\frac{1}{n_2}\sum_{s=1}^{n_2}\hat{\phi}^u_{\boldsymbol{l_u},n_1}(X^u_s)\hat{\phi}^v_{\boldsymbol{l_v},n_1}(X^v_s) - \mathbb{E}\left(\hat{\phi}^u_{\boldsymbol{l_u},n_1}(X^u)\hat{\phi}^v_{\boldsymbol{l_v},n_1}(X^v)\right)\right|.$$

Then, Bernstein's inequality implies that

$$\begin{aligned}
\mathbb{P}\left(\sup_{u,v,\boldsymbol{l_u},\boldsymbol{l_v}}\Delta_{u,v,\boldsymbol{l_u},\boldsymbol{l_v}} \geq \gamma n_2^{-\xi/2}\right) &\leq \mathbb{P}\left(\sup_{u,v,\boldsymbol{l_u},\boldsymbol{l_v}}\Delta_{u,v,\boldsymbol{l_u},\boldsymbol{l_v}} \geq \gamma n_2^{-\xi/2}\&N_{n_1} < M^*+1\right)\\
&\quad+\mathbb{P}\left(\sup_{u,v,\boldsymbol{l_u},\boldsymbol{l_v}}\Delta_{u,v,\boldsymbol{l_u},\boldsymbol{l_v}} \geq \gamma n_2^{-\xi/2}\&N_{n_1} > M^*+1\right)\\
&\leq 64L^4p_n^4\exp\left(-\frac{1}{2}\frac{\gamma^2{n_2}^{1-\xi}}{(M^*+1)^4+(M^*+1)^2\gamma/3{n_2}^{-\xi/2)}}\right)\\
&\quad+\mathbb{P}\left(N_{n_1} > M^*+1\right).
\end{aligned}$$

Lemma 4.5 and Assumption $(\mathbf{H^2_b})$ yields (ii).

Assertion (iii) The proof follows the roadmap of (ii) of Lemma 2.2 of Chapter II. We thus define the truncated variable $(\varepsilon^t_s)_{s\in[\![1,n_2]\!]}$,

$$\varepsilon^t_s = \begin{cases}\varepsilon_s & \text{if } |\varepsilon_s| \leq K_n\\ sg(\varepsilon_s)K_n & \text{if } |\varepsilon_s| > K_n,\end{cases}$$

where $sg(\varepsilon)$ is the sign of ε. Then, for $\gamma > 0$, we have:

$$\begin{aligned}
\mathbb{P}\left(n_2^{\xi/2}\sup_{u,\boldsymbol{l_u}}\left|\langle\hat{\phi}^u_{\boldsymbol{l_u},n_1},\varepsilon\rangle_{n_2}\right| > \gamma\right) &\leq \mathbb{P}\left(n_2^{\xi/2}\sup_{u,\boldsymbol{l_u}}\left|\langle\hat{\phi}^u_{\boldsymbol{l_u},n_1},\varepsilon-\varepsilon^t\rangle_{n_2}\right| > \gamma/3\right)\\
&\quad+\mathbb{P}\left(n_2^{\xi/2}\sup_{u,\boldsymbol{l_u}}\left|\langle\hat{\phi}^u_{\boldsymbol{l_u},n_1},\varepsilon^t\rangle_{n_2} - \langle\hat{\phi}^u_{\boldsymbol{l_u},n_1},\varepsilon^t\rangle\right| > \gamma/3\right)\\
&\quad+\mathbb{P}\left(n_2^{\xi/2}\sup_{u,\boldsymbol{l_u}}\left|\langle\hat{\phi}^u_{\boldsymbol{l_u},n_1},\varepsilon^t\rangle\right| > \gamma/3\right)\\
&= I + II + III.
\end{aligned}$$

Term I: We can bound I using the following simple inclusion:

$$\begin{aligned}
\left\{n_2^{\xi/2}\sup_{u,\boldsymbol{l_u}}\left|\langle\hat{\phi}^u_{\boldsymbol{l_u},n_1},\varepsilon^t\rangle_{n_2} - \langle\hat{\phi}^u_{\boldsymbol{l_u},n_1},\varepsilon^t\rangle\right| > \gamma/3\right\} &\subset \left\{\text{there exists s such that } \varepsilon_s - \varepsilon^t_s \neq 0\right\}\\
&= \left\{\text{there exists s such that } |\varepsilon_s| > K_n\right\}.
\end{aligned}$$

Hence,

$$\begin{aligned}
I &\leq \mathbb{P}(\text{some } |\varepsilon_s| > K_n)\\
&\leq n_2\mathbb{P}(|\varepsilon| > K_n) \leq n_2K_n^{-q}\mathbb{E}(|\varepsilon|^q) = \underset{n\to+\infty}{\mathcal{O}}(n^{1-q\xi/4}),
\end{aligned}$$

where $n_2 = n/2$ and we have chosen $K_n := n^{\xi/4}$ since $q > 4/\xi$ by Assumption of the Lemma. Hence, I can become arbitrarily small.

Term II: Using again Bernstein's Inequality to the family of independent random variables $(\hat{\phi}^u_{\boldsymbol{l_u},n_1}(X^u_s)\varepsilon^t_s)_{s=1,\cdots,n_2}$ and considering the two events $\{N_{n_1} > M^* + 1\}$ and $\{N_{n_1} < M^* + 1\}$, we have that:

$$II \leq 2Lp_n \exp\left(-\frac{1}{2}\frac{(\gamma^2/9){n_2}^{1-\xi}}{(M^*+1)^4\sigma^2 + (M^*+1)K_n\gamma/9{n_2}^{-\xi/2}}\right) + \mathbb{P}(N_{n_1} > M^* + 1),$$

where $\sigma^2 := \mathbb{E}(|\varepsilon|^2)$. We can then make the right-hand side of the previous inequality arbitrarily small owing to $(\mathbf{H^2_b})$ with $K_n = n^{\xi/2}$.

Term III: By assumption, $\mathbb{E}\left(\phi^u_{\boldsymbol{l_u}}(X^u)\varepsilon\right) = 0$. We then have:

$$\begin{aligned} III &\leq \mathbb{P}\left(n_2^{\xi/2}\sup_{u,\boldsymbol{l_u}}\left|\mathbb{E}[(\hat{\phi}^u_{\boldsymbol{l_u},n_1} - \phi^u_{\boldsymbol{l_u}})(X^u)\varepsilon^t]\right| > \gamma/6\right) + \mathbb{P}\left(n_2^{\xi/2}\sup_{u,\boldsymbol{l_u}}\left|\mathbb{E}[\phi^u_{\boldsymbol{l_u}}(X^u)(\varepsilon - \varepsilon^t)]\right| > \gamma/6\right) \\ &= III_1 + III_2, \end{aligned}$$

with,

$$\begin{aligned} III_1 &= \mathbb{P}\left(n_2^{\xi/2}\sup_{u,\boldsymbol{l_u}}\left|\mathbb{E}[(\hat{\phi}^u_{\boldsymbol{l_u},n_1} - \phi^u_{\boldsymbol{l_u}})(X^u)]\right|\left|\mathbb{E}(\varepsilon^t)\right| > \gamma/6\right) \\ &\leq \mathbb{P}\left(n_2^{\xi/2}\sup_{u,\boldsymbol{l_u}}\left|\mathbb{E}[(\hat{\phi}^u_{\boldsymbol{l_u},n_1} - \phi^u_{\boldsymbol{l_u}})(X^u)]\right|\left|\mathbb{E}(\varepsilon^t)\right| > \gamma/6\right) \\ &\leq \mathbb{1}_{\{n_2^{\xi/2}\sup_{u,\boldsymbol{l_u}}\left|\mathbb{E}[(\hat{\phi}^u_{\boldsymbol{l_u},n_1} - \phi^u_{\boldsymbol{l_u}})(X^u)]\right|\left|\mathbb{E}(\varepsilon^t)\right| > \gamma/6\}}. \end{aligned}$$

Moreover, one has

$$\begin{aligned} \left|\mathbb{E}(\varepsilon^t)\right| &= \left|\int_{|x|\leq K_n} x dP_\varepsilon(x) + \int_{|x|>K_n} sg(x)K_n dP_\varepsilon(x)\right| = \left|\int_{|x|>K_n}(sg(x)K_n - x)dP_\varepsilon(x)\right| \\ &\leq \int \mathbb{1}_{|x|>K_n}(K_n + |x|)dP_\varepsilon(x) \\ &\leq K_n\mathbb{P}(|\varepsilon| > K_n) + \int |x|\,\mathbb{1}_{|x|>K_n}dP_\varepsilon(x) \\ &\leq K_n^{1-q}\mathbb{E}(|\varepsilon|^q) + \mathbb{E}(\varepsilon^2)^{1/2}K_n^{-q/2}\mathbb{E}(|\varepsilon|^q)^{1/2} \quad \text{by the Tchebychev Inequality , (III.28)} \end{aligned}$$

since $0 < \xi < 1$ and $q > 4/\xi > 4$. Then, set $K_n = n^{\xi/4}$, we obtain:

$$n_2^{\xi/2}\left\|\hat{\phi}^u_{\boldsymbol{l_u},n_1} - \phi^u_{\boldsymbol{l_u}}\right\|\left|\mathbb{E}(\varepsilon^t)\right| \leq n_2^{\xi/2}o(1)o(n^{-\xi/2}) = o(1),$$

when o is the usual Landau notation of relative insignificance.

Hence, $III_1 = 0$ for n large enough. For III_2, one has

$$III_2 \leq \mathbb{1}_{\{n_2^{\xi/2}\sup_{u,\boldsymbol{l_u}}\left|\mathbb{E}[\phi^u_{\boldsymbol{l_u}}(X^u)(\varepsilon - \varepsilon^t)]\right| > \gamma/6\}},$$

and, by independance,

$$\left|\mathbb{E}[\phi^u_{\boldsymbol{l_u}}(X^u)(\varepsilon - \varepsilon^t)]\right| = \left|\mathbb{E}[\phi^u_{\boldsymbol{l_u}}(X^u)]\right|\left|\mathbb{E}(\varepsilon - \varepsilon^t)\right| \leq M^*\left|\mathbb{E}(\varepsilon - \varepsilon^t)\right|.$$

Equation (III.28) then implies,

$$\left|\mathbb{E}(\varepsilon - \varepsilon^t)\right| = \left|\int_{|x|>K_n} (sg(x)K_n - x)dP_\varepsilon(x)\right| \leq o(n^{-\xi/2}).$$

Thus, III is arbitrarily small for n and γ large enough and (iii) holds.

Assertion (iv) Remark that,

$$\sup_{u,\boldsymbol{l_u}} \left|\langle \tilde{f}, \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle_{n_2} - \langle \tilde{f}, \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle\right| \leq \|\beta^0\|_1 \sup_{u,\boldsymbol{l_u}} \left|\langle \phi^v_{\boldsymbol{l_v}}, \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle_{n_2} - \langle \phi^v_{\boldsymbol{l_v}}, \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle\right|.$$

Now, $(\mathbf{H_{s,\alpha}})$ and Bernstein's Inequality implies

$$\mathbb{P}\left(\sup_{u,\boldsymbol{l_u}} \left|\langle \phi^v_{\boldsymbol{l_v}}, \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle_{n_2} - \langle \phi^v_{\boldsymbol{l_v}}, \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle\right| \geq \gamma {n_2}^{-\xi/2}\right) \leq \mathbb{P}(N_{n_1} > M^* + 1)$$
$$+ 2Lp_n \exp\left(-\frac{1}{2}\frac{\gamma^2 {n_2}^{1-\xi}}{(M^*+1)^4 + (M^*+1)^2\gamma/3{n_2}^{-\xi/2}}\right),$$

which implies with Assumption $(\mathbf{H^2_b})$ that:

$$\sup_{u,\boldsymbol{l_u}} \left|\langle \phi^v_{\boldsymbol{l_v}}, \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle_{n_2} - \langle \phi^v_{\boldsymbol{l_v}}, \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle\right| = \mathcal{O}_P(n^{-\xi/2}).$$

□

The following lemma, similar to Lemma 2.3 from Chapter II, then holds:

Lemma 4.7. *Under Assumptions* $(\mathbf{H_b})$, $(\mathbf{H_{\varepsilon,q}})$ *with* $q > 4/\xi$, *there exists a constant* $C > 0$ *such that, on the set* $\Omega_n = \{\omega, |\zeta_n(\omega)| < 1/2\}$*:*

$$\sup_{u,\boldsymbol{l_u}} \left|\langle Y - G_k(\bar{f}), \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle_{n_2} - \langle \tilde{R}_k(\bar{f}), \phi^u_{\boldsymbol{l_u}}\rangle\right| \leq \left(\frac{5}{2}\right)^k (1 + C\|\beta^0\|_1)\zeta_n.$$

Proof of Lemma 4.7. Denote $A_n(k,u) = \langle Y - G_k(\bar{f}), \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle_{n_2} - \langle \tilde{R}_k(\bar{f}), \phi^u_{\boldsymbol{l_u}}\rangle$. Assume first that $k = 0$,

$$\begin{aligned}
\sup_{u,\boldsymbol{l_u}}|A_n(0,u)| &= \sup_u |\langle Y, \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle_{n_2} - \langle \bar{f}, \phi^u_{\boldsymbol{l_u}}\rangle| \\
&\leq \sup_{u,\boldsymbol{l_u}} \left\{\left|\langle \tilde{f}, \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle_{n_2} - \langle \tilde{f}, \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle\right| + \left|\langle \tilde{f} - \bar{f}, \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle\right| + \left|\langle \bar{f}, \hat{\phi}^u_{\boldsymbol{l_u},n_1} - \phi^u_{\boldsymbol{l_u}}\rangle\right|\right\} \\
&\quad + \sup_{u,\boldsymbol{l_u}} \left|\langle \varepsilon, \hat{\phi}^u_{\boldsymbol{l_u},n_1}\rangle_{n_2}\right| \\
&\leq (1 + 4\|\beta^0\|_1)\zeta_n \quad \text{by } (iii) - (iv) \text{ of Lemma 4.6 and Theorem 3.1.}
\end{aligned}$$

From the main document, we remind that

$$G_k(\bar{f}) = G_{k-1}(\bar{f}) + \gamma\langle Y - G_{k-1}(\bar{f}), \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}\rangle_{n_2} \cdot \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}, \tag{III.29}$$

$$\begin{aligned}
R_k(\bar{f}) &= \bar{f} - G_k(\bar{f}) \\
&= \bar{f} - G_{k-1}(\bar{f}) - \gamma\langle Y - G_{k-1}(\bar{f}), \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}\rangle_{n_2} \cdot \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1},
\end{aligned} \tag{III.30}$$

and

$$\left\{\begin{array}{l} \tilde{R}_0(\bar{f}) = \bar{f} \\ \tilde{R}_k(\bar{f}) = \tilde{R}_{k-1}(\bar{f}) - \gamma\langle \tilde{R}_{k-1}(\bar{f}), \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}\rangle \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}. \end{array}\right. \tag{III.31}$$

The recursive relations (III.29) and (III.31), leads to, for any $k \geq 0$:

$$\begin{aligned} A_n(k,u) &= \langle Y - G_{k-1}(\bar{f}) - \gamma\langle Y - G_{k-1}(\bar{f}), \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}\rangle_{n_2} \cdot \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}, \hat{\phi}^{u}_{\boldsymbol{l_u},n_1}\rangle_{\mathrm{n}} \\ &\quad -\langle \tilde{R}_{k-1}(\bar{f}) - \gamma\langle \tilde{R}_{k-1}(\bar{f}), \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}\rangle \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}, \phi^{u}_{\boldsymbol{l_u}}\rangle \\ &\leq A_n(k-1,u) \\ &\quad -\gamma \underbrace{\left(\langle Y - G_{k-1}(\bar{f}), \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}\rangle_{n_2} - \langle \tilde{R}_{k-1}(\bar{f}), \phi^{u_k}_{\boldsymbol{l_{u_k}}}\rangle\right)\langle \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}, \hat{\phi}^{u}_{\boldsymbol{l_u},n_1}\rangle_{n_2}}_{=I} \\ &\quad +\gamma \underbrace{\langle \tilde{R}_{k-1}(\bar{f}), \phi^{u_k}_{\boldsymbol{l_{u_k}}}\rangle \left(\langle \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}, \phi^{u}_{\boldsymbol{l_u}}\rangle - \langle \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}, \hat{\phi}^{u}_{\boldsymbol{l_u},n_1}\rangle_{n_2}\right)}_{=II} \\ &\quad +\gamma \underbrace{\langle \tilde{R}_{k-1}(\bar{f}), \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1} - \phi^{u_k}_{\boldsymbol{l_{u_k}}}\rangle \langle \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}, \phi^{u}_{\boldsymbol{l_u}}\rangle}_{=III}. \end{aligned}$$

On the one hand, using assertion (ii) of Lemma 4.6, and the Cauchy-Schwarz inequality (with $\left\|\phi^{u}_{\boldsymbol{l_u}}\right\| = 1$), it comes

$$\begin{aligned} \sup_{u,\boldsymbol{l_u}}|I| &\leq \sup_{u,\boldsymbol{l_u}}|\langle \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}, \hat{\phi}^{u}_{\boldsymbol{l_u},n_1}\rangle_{n_2}| \sup_{u,\boldsymbol{l_u}}|A_n(k-1,u)| \\ &\leq (\sup_{u,\boldsymbol{l_u}}|\langle \phi^{u_k}_{\boldsymbol{l_{u_k}}}, \phi^{u}_{\boldsymbol{l_u}}\rangle| + \zeta_n) \sup_{u,\boldsymbol{l_u}}|A_n(k-1,u)| \\ &\leq (1+\zeta_n) \sup_{u,\boldsymbol{l_u}}|A_n(k-1,u)|. \end{aligned}$$

Consider now the phantom residual, from its recursive relation, we can show that $\left\|\tilde{R}_k(\bar{f})\right\|^2 = \left\|\tilde{R}_{k-1}(\bar{f})\right\|^2 - \gamma(2-\gamma)\langle \tilde{R}_{k-1}(\bar{f}), \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}\rangle^2 \leq \left\|\tilde{R}_{k-1}(\bar{f})\right\|^2$ and we deduce

$$\left\|\tilde{R}_k(\bar{f})\right\|^2 \leq \left\|\bar{f}\right\|^2. \tag{III.32}$$

Then,

$$\begin{aligned} \sup_{u,\boldsymbol{l_u}}|II| &\leq \left\|\tilde{R}_{k-1}(\bar{f})\right\| \left\|\phi^{u_k}_{\boldsymbol{l_{u_k}}}\right\| \sup_{u,\boldsymbol{l_u}}|\langle \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}, \phi^{u}_{\boldsymbol{l_u}}\rangle - \langle \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}, \hat{\phi}^{u}_{\boldsymbol{l_u},n_1}\rangle_{n_2}| \\ &\leq \left\|\bar{f}\right\| \sup_{u,\boldsymbol{l_u}}|\langle \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}, \phi^{u}_{\boldsymbol{l_u}}\rangle - \langle \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}, \hat{\phi}^{u}_{\boldsymbol{l_u},n_1}\rangle_{n_2}|, \end{aligned}$$

with

$$\begin{aligned} |\langle \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}, \phi^{u}_{\boldsymbol{l_u}}\rangle - \langle \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}, \hat{\phi}^{u}_{\boldsymbol{l_u},n_1}\rangle_{n_2}| &\leq |\langle \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}, \hat{\phi}^{u}_{\boldsymbol{l_u},n_1}\rangle_{n_2} - \langle \phi^{u_k}_{\boldsymbol{l_{u_k}}}, \phi^{u}_{\boldsymbol{l_u}}\rangle| \\ &\quad +|\langle \phi^{u_k}_{\boldsymbol{l_{u_k}}} - \hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}, \phi^{u}_{\boldsymbol{l_u}}\rangle|. \end{aligned}$$

Using again assertion (ii) from Lemma 4.6 and Theorem 3.1, we obtain the following bound for II,

$$\begin{aligned} \sup_{u,\boldsymbol{l_u}}|II| &\leq \left\|\bar{f}\right\| \left(\zeta_n + \sup_{u,\boldsymbol{l_u}} \left\|\phi^{u}_{\boldsymbol{l_u}} - \hat{\phi}^{u}_{\boldsymbol{l_u},n_1}\right\|\right) \\ &\leq 2\zeta_n \left\|\bar{f}\right\|. \end{aligned}$$

Finally, Theorem 3.1 gives

$$\begin{aligned}\sup_{u,\boldsymbol{l_u}}|III| &\leq \sup_{u,\boldsymbol{l_u}}\left\|\tilde{R}_{k-1}(\bar{f})\right\|\left\|\hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}-\phi^{u_k}_{\boldsymbol{l_{u_k}}}\right\|\left\|\hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}\right\|\left\|\phi^{u}_{\boldsymbol{l_u}}\right\| \\ &\leq \left\|\bar{f}\right\|\zeta_n.\end{aligned}$$

Our bounds on I, II and III, and $\gamma<1$ yields on $\Omega_n=\{\zeta_n<1/2\}$ that

$$\begin{aligned}\sup_{u,\boldsymbol{l_u}}|A_n(k,u)| &\leq \sup_{u,\boldsymbol{l_u}}|A_n(k-1,u)|+(1+\zeta_n)\sup_{u,\boldsymbol{l_u}}|A_n(k-1,u)|+3\zeta_n\left\|\bar{f}\right\| \\ &\leq \frac{5}{2}\sup_{u,\boldsymbol{l_u}}|A_n(k-1,u)|+3\zeta_n\left\|\bar{f}\right\|.\end{aligned}$$

A simple induction yields:

$$\begin{aligned}\sup_{u,\boldsymbol{l_u}}|A_n(k,u)| &\leq \left(\frac{5}{2}\right)^k\underbrace{\sup_{u,\boldsymbol{l_u}}|A_n(0,u)|}_{\leq(1+4\|\beta^0\|_1)\zeta_n}+3\zeta_n\left\|\bar{f}\right\|\sum_{\ell=0}^{k-1}\left(\frac{5}{2}\right)^\ell \\ &\leq \left(\frac{5}{2}\right)^k\zeta_n\left(1+\|\beta^0\|_1\left(4+6\sum_{\ell=1}^{\infty}\left(\frac{5}{2}\right)^{-\ell}\right)\right),\end{aligned}$$

which ends the proof with $C=14$. □

We then aim at applying Theorem 2.1 from Chapter II to the phantom residuals $(\tilde{R}_k(\bar{f}))_k$. Using the notation of Chapter II, this will be possible if we can show that the phantom residuals follows a theoretical Boosting with a shrinkage parameter $\nu\in[0,1]$. Thanks to Lemma 4.7 and by definiton of $\hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}$, one has

$$\begin{aligned}\left|\langle Y-G_{k-1}(\bar{f}),\hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}\rangle_{n_2}\right| &= \sup_{u,\boldsymbol{l_u}}\left|\langle Y-G_{k-1}(\bar{f}),\hat{\phi}^{u}_{\boldsymbol{l_u},n_1}\rangle_{n_2}\right| \\ &\geq \sup_{u,\boldsymbol{l_u}}\left\{\left|\langle\tilde{R}_{k-1}(\bar{f}),\phi^u_{\boldsymbol{l_u}}\rangle\right|-C\left(\frac{5}{2}\right)^{k-1}\zeta_n\|\beta^0\|_1\right\}. \quad \text{(III.33)}\end{aligned}$$

Applying again Lemma 4.7 on the set Ω_n, we obtain:

$$\begin{aligned}\left|\langle\tilde{R}_{k-1}(\bar{f}),\phi^{u_k}_{\boldsymbol{l_{u_k}}}\rangle\right| &\geq \left|\langle Y-G_{k-1}(\bar{f}),\hat{\phi}^{u_k}_{\boldsymbol{l_{u,k}},n_1}\rangle_{n_2}\right|-C\left(\tfrac{5}{2}\right)^{k-1}\zeta_n\|\beta^0\|_1 \\ &\geq \sup_{u,\boldsymbol{l_u}}\left|\langle\tilde{R}_{k-1}(\bar{f}),\phi^u_{\boldsymbol{l_u}}\rangle\right|-2C\left(\tfrac{5}{2}\right)^{k-1}\zeta_n\|\beta^0\|_1. \quad \text{(III.34)}\end{aligned}$$

Consider now the set

$$\tilde{\Omega}_n=\left\{\omega,\quad\forall k\leq k_n,\quad\sup_{u,\boldsymbol{l_u}}\left|\langle\tilde{R}_{k-1}(\bar{f}),\phi^u_{\boldsymbol{l_u}}\rangle\right|>4C\left(\frac{5}{2}\right)^{k-1}\zeta_n\|\beta^0\|_1\right\}.$$

We deduce from Equation (III.34) the following inequality on $\Omega_n\cap\tilde{\Omega}_n$:

$$\left|\langle\tilde{R}_{k-1}(\bar{f}),\phi^{u_k}_{\boldsymbol{l_{u_k}}}\rangle\right|\geq\frac{1}{2}\sup_{u,\boldsymbol{l_u}}\left|\langle\tilde{R}_{k-1}(\bar{f}),\phi^u_{\boldsymbol{l_u}}\rangle\right|. \quad \text{(III.35)}$$

Consequently, on $\Omega_n \cap \tilde{\Omega}_n$, the family $(\tilde{R}_k(\bar{f}))_k$ satisfies a theoretical Boosting, given by Algorithm 1 of Chapter II, with constant $\nu = 1/2$ and we have:

$$\left\|\tilde{R}_k(\bar{f})\right\| \leq C' \left(1 + \frac{1}{4}\gamma(2-\gamma)k\right)^{-\frac{2-\gamma}{2(6-\gamma)}}. \tag{III.36}$$

Consider now the complementary set

$$\tilde{\Omega}_n^C = \left\{\omega, \quad \exists\, k \leq k_n \quad \sup_{u,\boldsymbol{l_u}} \left|\langle \tilde{R}_{k-1}(\bar{f}), \phi^u_{\boldsymbol{l_u}}\rangle\right| \leq 4C \left(\frac{5}{2}\right)^{k-1} \zeta_n \|\beta^0\|_1 \right\}.$$

Remark that

$$\begin{aligned}
\left\|\tilde{R}_k(\bar{f})\right\|^2 &= \langle \tilde{R}_k(\bar{f}), \bar{f} - \gamma \sum_{j=0}^{k-1} \langle \tilde{R}_j(\bar{f}), \hat{\phi}^{u_j}_{\boldsymbol{l_{u_j}}, n_1}\rangle \hat{\phi}^{u_j}_{\boldsymbol{l_{u_j}}, n_1}\rangle \\
&\leq \|\beta^0\|_1 \sup_{u,\boldsymbol{l_u}} \left|\langle \tilde{R}_k(\bar{f}), \hat{\phi}^u_{\boldsymbol{l_u}, n_1}\rangle\right| + \gamma \sum_{j=0}^{k-1} \left|\langle \tilde{R}_j(\bar{f}), \hat{\phi}^{u_j}_{\boldsymbol{l_{u_j}}, n_1}\rangle\right| \sup_{u,\boldsymbol{l_u}} \left|\langle \tilde{R}_k(\bar{f}), \hat{\phi}^u_{\boldsymbol{l_u}, n_1}\rangle\right|.
\end{aligned}$$

Moreover,

$$\begin{aligned}
\sup_{u,\boldsymbol{l_u}} \left|\langle \tilde{R}_k(\bar{f}), \hat{\phi}^u_{\boldsymbol{l_u}, n_1}\rangle\right| &\leq \sup_{u,\boldsymbol{l_u}} \left|\langle \tilde{R}_k(\bar{f}), \phi^u_{\boldsymbol{l_u}}\rangle\right| + \sup_{u,\boldsymbol{l_u}} \left|\langle \tilde{R}_k(\bar{f}), \hat{\phi}^u_{\boldsymbol{l_u}, n_1} - \phi^u_{\boldsymbol{l_u}}\rangle\right| \\
&\leq \sup_{u,\boldsymbol{l_u}} \left|\langle \tilde{R}_k(\bar{f}), \phi^u_{\boldsymbol{l_u}}\rangle\right| + 2\|\beta^0\|_1 \zeta_n \quad \text{by Theorem 1 and (III.32).}
\end{aligned}$$

We hence have

$$\begin{aligned}
\left\|\tilde{R}_k(\bar{f})\right\|^2 &\leq \left(\|\beta^0\|_1 + \gamma \sum_{j=0}^{k-1} \left|\langle \tilde{R}_j(\bar{f}), \hat{\phi}^{u_j}_{\boldsymbol{l_{u_j}}, n_1}\rangle\right|\right) \left(\sup_{u,\boldsymbol{l_u}} \left|\langle \tilde{R}_k(\bar{f}), \phi^u_{\boldsymbol{l_u}}\rangle\right| + 2\|\beta^0\|_1 \zeta_n\right) \\
&\leq \|\beta^0\|_1 (1 + 2\gamma k) \left(\sup_{u,\boldsymbol{l_u}} \left|\langle \tilde{R}_k(\bar{f}), \phi^u_{\boldsymbol{l_u}}\rangle\right| + 2\|\beta^0\|_1 \zeta_n\right) \\
&\leq 4C\|\beta^0\|_1^2 \zeta_n (1 + 2\gamma k) \left(\frac{5}{2}\right)^k \quad \text{on} \quad \tilde{\Omega}_n^C.
\end{aligned} \tag{III.37}$$

Finally, on the set $(\Omega_n \cap \tilde{\Omega}_n) \cup \tilde{\Omega}_n^C$, by Equations (III.36) and (III.37),

$$\left\|\tilde{R}_k(\bar{f})\right\|^2 \leq C'^2 \left(1 + \frac{1}{4}\gamma(2-\gamma)k\right)^{-\frac{2-\gamma}{6-\gamma}} + 4C\|\beta^0\|_1^2 \zeta_n (1 + 2\gamma k) \left(\frac{5}{2}\right)^k. \tag{III.38}$$

To conclude the first part of the proof, remark that

$$\mathbb{P}\left((\Omega_n \cap \tilde{\Omega}_n) \cup \tilde{\Omega}_n^C\right) \geq \mathbb{P}(\Omega_n) \underset{n\to+\infty}{\longrightarrow} 1.$$

On this set, Inequality (III.38) holds almost surely, and for $k_n < \frac{\xi/2 - \vartheta - 2\alpha}{2\log(3)} \log(n)$, which grows sufficiently slowly, we get

$$\left\|\tilde{R}_{k_n}(\bar{f})\right\| \xrightarrow[n\to+\infty]{\mathbb{P}} 0. \tag{III.39}$$

Consider now $A_k := \left\| R_k(\bar{f}) - \tilde{R}_k(\bar{f}) \right\|$ for $k \geq 1$. By definitions reminded in (III.30)-(III.31), we have:

$$\begin{aligned} A_k &\leq A_{k-1} + \gamma \left| \langle Y - G_{k-1}(\bar{f}), \hat{\phi}^{u_k}_{\boldsymbol{l}_{\boldsymbol{u},\boldsymbol{k}},n_1} \rangle_{n_2} - \langle \tilde{R}_{k-1}(\bar{f}), \hat{\phi}^{u_k}_{\boldsymbol{l}_{\boldsymbol{u},\boldsymbol{k}},n_1} \rangle \right| \\ &\leq A_{k-1} + \gamma \left| \langle Y - G_{k-1}(\bar{f}), \hat{\phi}^{u_k}_{\boldsymbol{l}_{\boldsymbol{u},\boldsymbol{k}},n_1} \rangle_{n_2} - \langle \tilde{R}_{k-1}(\bar{f}), \phi^{u_k}_{\boldsymbol{l}_{\boldsymbol{u}_\boldsymbol{k}}} \rangle \right| \\ &\quad + \gamma \left| \langle \tilde{R}_{k-1}(\bar{f}), \hat{\phi}^{u_k}_{\boldsymbol{l}_{\boldsymbol{u},\boldsymbol{k}},n_1} - \phi^{u_k}_{\boldsymbol{l}_{\boldsymbol{u}_\boldsymbol{k}}} \rangle \right|. \end{aligned} \tag{III.40}$$

By Lemma 4.7, we then deduce the following inequality on Ω_n:

$$A_k \leq A_{k-1} + \gamma \left(\frac{5}{2}\right)^{k-1} \left(1 + C\|\beta^0\|_1\right) \zeta_n + 2\gamma \|\beta^0\|_1 \zeta_n. \tag{III.41}$$

Since $A_0 = 0$, we deduce recursively from Equation (III.41) that, on Ω_n,

$$A_{k_n} \xrightarrow[n \to +\infty]{\mathbb{P}} 0.$$

Finally, as

$$\left\| \hat{f} - \tilde{f} \right\| = \left\| G_{k_n}(\bar{f}) - \tilde{f} \right\| \leq \left\| \bar{f} - \tilde{f} \right\| + \left\| R_{k_n}(\bar{f}) - \tilde{R}_{k_n}(\bar{f}) \right\| + \left\| \tilde{R}_{k_n}(\bar{f}) \right\|, \tag{III.42}$$

it remains to deal with the term $\left\| \bar{f} - \tilde{f} \right\|$. As,

$$\left\| \bar{f} - \tilde{f} \right\| \leq \|\beta^0\|_1 \left\| \phi^u_{\boldsymbol{l}_\boldsymbol{u}} - \hat{\phi}^u_{\boldsymbol{l}_\boldsymbol{u},n_1} \right\|,$$

and the proof follows using $(\mathbf{H}_{\mathbf{s},\alpha})$ with $\alpha < \xi/4 - \vartheta/2$ and Theorem 3.1.

5 Numerical Applications

This section is devoted to the numerical efficiency of the two-step procedure given by Algorithm 7, and primarily focuses on the practical use of the HOFD through sensitivity analysis (SA). SA aims to identify the most contributive variables to the variability of a regression model ([SCS00] and [CIBN05]). The most common quantification is a variance-based index, known as the Sobol index ([Sob93]). This measure relies on the Hoeffding decomposition that provides an elegant and meaningful theoretical framework when inputs are known to be independent. However, as mentioned in the introduction, the interpretation of such indices may be irrelevant when strong dependencies arise. The HOFD presented in Section 2.2 is of great interest in this situation because it provides a general and rigorous multivariate regression extension that can be used to define sensitivity indices well-tailored to dependent inputs. As detailed in [CGP12], the model variance can be expanded as follows:

$$\mathrm{Var}(Y) = \sum_{u \in S^*_n} \left(\mathrm{Var}(f_u(X^u)) + \sum_{u \cap v \neq u,v} \mathrm{Cov}\left(f_u(X^u), f_v(X^v)\right) \right).$$

Therefore, to measure the contribution of X^u, for $|u| \geq 1$, in terms of variability in the model, it is then quite natural to define a sensitivity index S_u as follows:

$$S_u = \frac{\mathrm{Var}(f_u(X^u)) + \sum_{u \cap v \neq u,v} \mathrm{Cov}(f_u(X^u), f_v(X^v))}{\mathrm{Var}(Y)}. \tag{III.43}$$

Furthermore, we deduce the empirical estimation of (III.43) once we have applied the procedure described in Algorithm 7 to obtain $(\hat{f}_u, \hat{f}_v, u \cap v \neq u, v)$.

5.1 Description

We end this work with a short simulation study, focused primarily on the performance of the greedy selection algorithm for the prediction of generalized sensitivity indices. Since the estimation of these indices consists in estimating the summands of the generalized functional ANOVA decomposition (referred to as HOFD), we begin by constructing a hierarchically orthogonal system of functions to approximate the components. As pointed out above (see Assumption $(\mathbf{H}_{\mathbf{b}}^{\mathbf{3},\vartheta})$ in Theorem 3.1 and 3.2), the invertibility of each linear system plays an important role in our theoretical study. For each situation, we therefore measured the degeneracy of the matrices involved, given by:

$$d(A) = \inf_{i,j\in[\![1,p]\!]} \det(A^{ij}).$$

We then use a variable selection method to select a sparse number of predictors. The goal is to numerically compare three variable selection methods: the $\mathbb{L}_2$-Boosting, the Forward-Backward greedy algorithm (referred to as FoBa below), and the Lasso estimator. As pointed out above, we have an n-sample of i.i.d. observations $(Y_s, X_s)_{s=1,\cdots,n}$ broken down into two samples of size $n_1 = n_2 = n/2$. The first sample is used to construct the system of functions according to Algorithm 5. The second sample is used to solve the penalized regression problem given by (III.10) and illustrated here:

$$(\hat{\beta}^u_{\boldsymbol{l_u}})_{\boldsymbol{l_u},u} \in \underset{\beta^u_{\boldsymbol{l_u}}\in\mathbb{R}}{\operatorname{argmin}} \frac{1}{n_2}\sum_{s=1}^{n_2}\left[y^s - \sum_{\substack{u\in S\\|u|\le d}}\sum_{\boldsymbol{l_u}} \beta^u_{\boldsymbol{l_u}}\hat{\phi}^u_{\boldsymbol{l_u},n_1}(X^u_s)\right]^2 + \lambda J(\beta^1_1,\cdots,\beta^u_{\boldsymbol{l_u}},\cdots).$$

We will now briefly describe how we use the Lasso, the FoBa and the Boosting.

5.2 Feature selection Algorithms

FoBa procedure The FoBa algorithm, as well as the $\mathbb{L}_2$-Boosting, use a greedy exploration to minimize the previous criterion when $J(\cdot)$ is a ℓ_0 penalty, *i.e.*,

$$J(\beta^1_1,\cdots,\beta^u_{\boldsymbol{l_u}},\cdots) = \sum_{\substack{u\in S^*_n\\|u|\le d}}\sum_{\boldsymbol{l_u}} \mathbb{1}(\beta^u_{\boldsymbol{l_u}}\neq 0).$$

This algorithm is an iterative scheme that sequentially deletes or selects an element of $\mathcal{D}$ that has the least impact on the fit, respectively, that significantly reduces the model residual. This algorithm is described in [Zha11] and used for HOFD in [CGP72]. We refer to these references for a more in-depth description of this algorithm. This procedure depends on two shrinkage parameters, ϵ and δ. The parameter ϵ is the stopping criterion that predetermines if a large number of predictors is going to be introduced into the model. The second parameter, $\delta \in]0,1]$, offers a flexibility in the *backward* step since it allows the algorithm to smoothly eliminate a predictor at each step.

In our numerical experiments, we have found a well suited behaviour of the FoBa procedure with $\epsilon = 10^{-2}$ and $\delta = 1/2$.

Calibration of the Boosting We have fixed the shrinkage parameter to $\gamma = 0.7$ as it yields a suitable value for high dimensional regression, even though we do not have found some extreme differences when γ varies in $[0.5; 1[$. Since the optimal value for k_{up} is unknown in practice, we use a C_p-Mallows type criterion to fix the optimal number of iterations. This stopping criterion

is much more important than the choice of the shrinkage parameter. It is of course induced by γ since it depends on the sequence of the Boosting iterations.

Like in the LARS algorithm, we follow the recommendations of [EHJT04] to select the best solution. First, we define a large number of iterations, say K. For each step $k \in \{1, \cdots, K\}$, the Boosting algorithm computes an estimation of the solution $\hat{\boldsymbol{\beta}}(k)$. On the basis of this, we compute the following quantity,

$$E_k^{\text{Boost}} = \frac{1}{n_2} \sum_{s=1}^{n_2} \left(Y_s - \sum_{\hat{\phi}^u_{\boldsymbol{l_u},n_1} \in \mathcal{D}} \hat{\beta}^u_{\boldsymbol{l_u}}(k) \hat{\phi}^u_{\boldsymbol{l_u},n_1}(X^u_s) \right)^2 - n_2 + 2k,$$

where the implied set of functions $\hat{\phi}^u_{\boldsymbol{l_u},n_1}$ has been selected through the first k steps of the algorithm. Finally, we choose the optimal number of selected functions $\hat{k}_{up}$ such that:

$$\hat{k}_{\text{up}} = \underset{k=1,\cdots,K}{\operatorname{argmin}} E_k^{\text{Boost}}.$$

Lasso algorithm Since the ℓ_0 strategy is very difficult to handle and may suffer from a lack of robustness, the ℓ_0 penalty is often replaced by the $\lambda \times \ell_1$ strategy that yields the Lasso estimator for a given penalization parameter $\lambda > 0$, *i.e.*,

$$J(\beta_1^1, \cdots, \beta^u_{\boldsymbol{l_u}}, \cdots) = \sum_{\substack{u \in S_n^* \\ |u| \le d}} \sum_{\boldsymbol{l_u}} \left| \beta^u_{\boldsymbol{l_u}} \right|.$$

Several algorithms have been proposed in the literature to solve the Lasso regression. One of the most popular is the LARS method, described in [EHJT04], because it performs a solution that coincides with the theoretical regularization path $\{\hat{\boldsymbol{\beta}}(\lambda),\ \lambda \in \mathbb{R}^+\}$. However, the LARS strategy is very expensive in large Lasso problems. To make a good numerical comparison with the greedy algorithms, we choose to perform a coordinate descent algorithm proposed by [Fu98], and [FHHT07] because of its low computational cost compared to the LARS implementation. The tuning parameter λ is first selected by generalized cross-validation, and the Lasso Coordinate Descent (LCD) algorithm is performed with the R `lassoshooting` package.

5.3 Datasets

Each experiment on each dataset was randomly reproduced 50 times to compute the Monte-Carlo errors. Since each dataset has very few instances, the size L of the initial orthonormal systems has to be small. Here, we arbitrarily choose $5 \le L \le 8$ and the approximation performance do not suffer from the sensitivity of L in these models.

First dataset: the Ishigami function Well known in sensitivity analysis, the analytical form of the Ishigami model is given by,

$$Y = \sin(X_1) + a \sin^2(X_2) + b X_3^4 \sin(X_1),$$

where we set $a = 7$ and $b = 0.1$, and where it is assumed that the inputs are independent. In the numerical experience, we consider the following cases.

1. For all $i = 1, 2, 3$, the inputs are uniformly distributed on $[-\pi, \pi]$. We choose $n = 300$ observations, with the first eight Legendre basis functions ($L = 8$).

2. For all $i = 1, 2, 3$, the inputs are uniformly distributed on $[-\pi, \pi]$. We choose $n = 300$ observations, with the first eight Fourier basis functions.

Each time, the number of predictors is $m_n = pL + \binom{p}{2}L^2 = 408 \geq n$.

Second dataset: the g-Sobol function This function is referred in [SCS00], and is given by

$$Y = \prod_{i=1}^{p} \frac{|4X_i - 2| + a_i}{1 + a_i}, \quad a_i \geq 0,$$

where the inputs X_i are independent and uniformly distributed over $[0, 1]$. The analytical Sobol indices are given by

$$S_u = \frac{1}{D} \prod_{i \in u} D_i, \quad D_i = \frac{1}{3(1 + a_i)^2}, \quad D = \prod_{i=1}^{p} (D_i + 1) - 1, \ \forall\ u \subseteq [\![1, p]\!].$$

Here, we take $p = 25$ and $a = (0, 0, 0, 1, 1, 2, 3, 4.5, 4.5, 4.5, 9, 9, 9, 9, 9, 99, \cdots, 99)$. For the construction of the hierarchical basis functions, we choose the first five Legendre polynomials ($L = 5$). We use $n = 2000$ evaluations of the model and the number of predictors $m_n = pL + \binom{p}{2}L^2 = 7625$, which clearly exceeds the sample size n.

Third dataset: dependent inputs The third data set stands for a rarely investigated situation, where the inputs are correlated. As proposed by [MT12], we generate a sample set according to the following distribution: X^1 and X^2 are uniformly sampled in the set $\mathcal{S}$:

$$\mathcal{S} := \left\{(x_1, x_2) \in [-1, 1]^2 \,|\, 2x_1^2 - 1 \leq x_2 \leq 2x_1^2\right\}.$$

Furthermore, X^3 is also sampled uniformly in $[-1; 1]$. Then, Y is built following

$$Y = X^1 + X^2 + X^3.$$

The inputs X^1 and X^2 are clearly not independent and we do not exactly know the analytical Sobol indices. We choose $n = 100$ observations, with the first six Legendre basis functions ($L = 6$).

5.4 The tank pressure model

This real case study concerns a shell closed by a cap and subject to an internal pressure. Figure III.1 illustrates a simulation of tank distortion. We are interested in the von Mises stress, detailed in [vM13] on the point y indicated in Figure III.1. The von Mises stress makes it possible to predict material yielding that occurs when the material yield strength is reached. The selected point y corresponds to the point for which the von Mises stress is maximal in the tank. Therefore, we want to prevent the tank from material damage induced by plastic deformations. In order to provide a large panel of tanks able to resist the internal pressure, a manufacturer wants to know the parameters that contribute the most to the von Mises criterion variability. In the model that we propose, the von Mises criterion depends on three geometrical parameters: the shell internal radius (R_{int}), the shell thickness (T_{shell}), and the cap thickness (T_{cap}). It also depends on five physical parameters concerning Young's modulus (E_{shell} and E_{cap}) and the yield strength ($\sigma_{y,shell}$ and $\sigma_{y,cap}$) of the shell and the cap. The last parameter is the internal pressure (P_{int}) applied to the shell. There exists some strong correlations between some of the inputs of the system

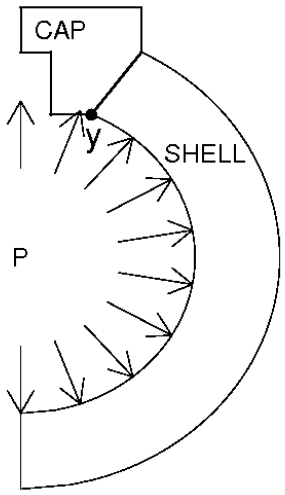


Figure III.1: Tank distortion at point y

Inputs	Distribution
R_{int}	$\mathcal{U}([1800; 2200])$, $\gamma(R_{int}, T_{shell}) = 0.85$
T_{shell}	$\mathcal{U}([360; 440])$, $\gamma(T_{shell}, T_{cap}) = 0.3$
T_{cap}	$\mathcal{U}([180; 220])$, $\gamma(T_{cap}, R_{int}) = 0.3$
E_{cap}	$\alpha N(\mu, \Sigma) + (1-\alpha)N(\mu, \Omega)$
$\sigma_{y,cap}$	$\alpha = 0.02$, $\mu = \begin{pmatrix} 210 \\ 500 \end{pmatrix}$, $\Sigma = \begin{pmatrix} 350 & 0 \\ 0 & 29 \end{pmatrix}$, $\Omega = \begin{pmatrix} 175 & 81 \\ 81 & 417 \end{pmatrix}$
E_{shell}	$\alpha N(\mu, \Sigma) + (1-\alpha)N(\mu, \Omega)$
$\sigma_{y,shell}$	$\alpha = 0.02$, $\mu = \begin{pmatrix} 70 \\ 300 \end{pmatrix}$, $\Sigma = \begin{pmatrix} 117 & 0 \\ 0 & 500 \end{pmatrix}$, $\Omega = \begin{pmatrix} 58 & 37 \\ 37 & 250 \end{pmatrix}$
P_{int}	$N(80, 10)$

Table III.1: Description of inputs of the shell model

owing to the constraints of manufacturing processes, for instance between the shell radius and its thickness. The system is modeled by a 2D finite element ASTER code. Input distributions are provided in Table III.1.

The geometrical parameters are uniformly distributed because of the large choice left for tank construction. The correlation γ between the geometrical parameters is induced by the constraints linked to manufacturing processes. The physical inputs are normally distributed and their uncertainty is due to the manufacturing process and the properties of the elementary constituent variabilities. The large variability of P_{int} in the model corresponds to the different internal pressure values that could be applied to the shell by the user.

To measure the contribution of the correlated inputs to the output variability, we estimate the generalized sensitivity indices. We do $n = 1000$ simulations. We use the first Hermite basis functions, whose maximum degree is 5 for every parameter.

5.5 Results

We consider both the estimation of the sensitivity indices, the ability to select the good representation of the different signals, and the computation time needed to obtain the sparse representation. "Greedy" refers to the Foba procedure as well as "LCD" refers to the Lasso coordinate descent method. Our method is, of course, referred to as "Boosting".

Sensitivity estimation Figures III.2 and III.3 provide the dispersion of the sensitivity indices estimated by our three methods on the Ishigami function. We can see that the three methods behave well with the two basis functions. Note that handling the Fourier basis is, as expected, more suitable for the Ishigami function than the Legendre basis (see the sensitivity index S_3 in Figures III.2 and III.3). For the sake of clarity, Figure III.4 only represents the first ten sensitivity indices. We can also draw similar conclusions with Figure III.4, where the three methods lead to the same conclusion. It should also be noted that the standard deviations of each method seem to be relatively equivalent. Figure III.5 represents the estimated sensitivity indices when the inputs are correlated. The analytical results are obviously unknown, but we obtain similar results for the three methods.

Finally, as illustrated in Figure III.6, the most contributive parameter to the von Mises criterion variability is the internal pressure P_{int}, which is not surprising. Concerning the geometric characteristics, the main parameters of the three methods are cap thickness, T_{cap}, and shell thickness, T_{shell}, using their expensive code, although the shell internal radius does not seem to be that important.

Computation time and accuracy The performances of the three methods are illustrated in Table III.2, on the basis of their computational cost and the accuracy of the feature selection.

Regarding the statistical accuracy, it should be noted that each estimator of high dimensional regression possesses a comparable dispersion on all the datasets and performs quite similarly on the first dataset. The Lasso estimator seems a little bit unprecise in the third data-set in comparison with the FoBa and Boosting methods. At last, the LCD method is also outperformed on the third data-set (with dependent inputs): it selects a significantly larger number of sensitivity indices in comparison with Boosting and FoBa methods (for instance, the indices S_{13} and S_{23} are certainly equals to 0 owing to the definition of Y). This may be due to the influence of the dependency among the inputs X^1 and X^2 in this data-set on the Lasso estimator.

Furthermore, it clearly appears in Table III.2 that our proposed $\mathbb{L}_2$-Boosting is the fastest method. This is particularly true on the 25-dimension g-Sobol function where the fraction of additional time required by the LCD algorithm in comparison to the $\mathbb{L}_2$-Boosting is about 100. Although we do not have access to the theoretical support recovery $\|\boldsymbol{\beta}\|_0$, we can observe that the results of the $\mathbb{L}_2$-Boosting are equivalent to those of other algorithms in terms of its feature selection ability. Hence, for the same degree of accuracy, our method seems to be much faster.

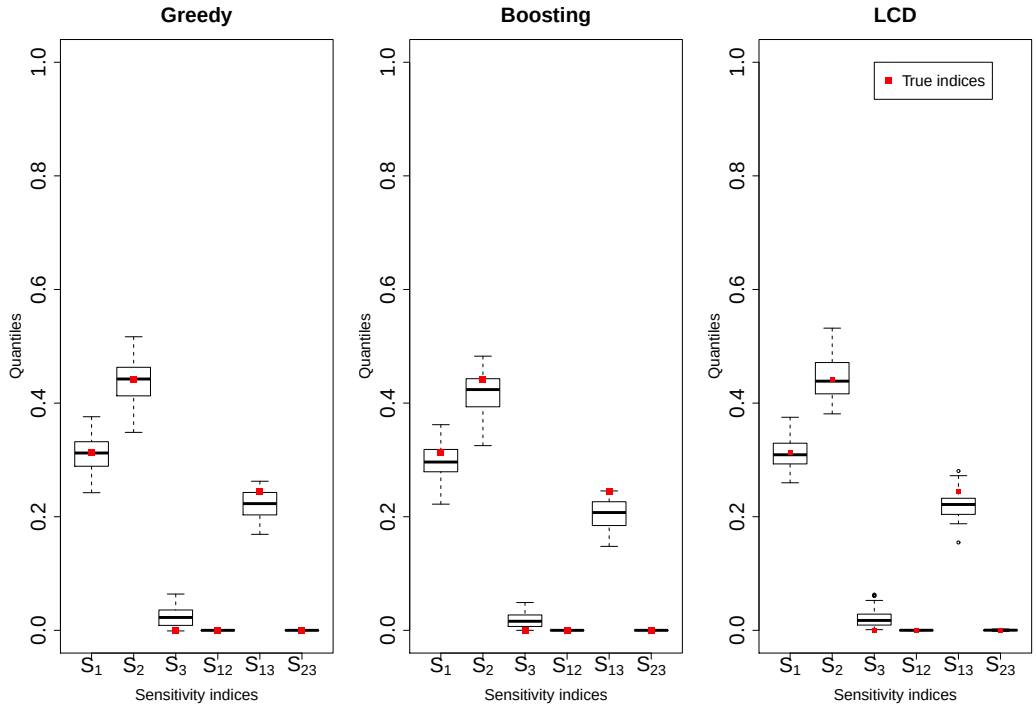


Figure III.2: Representation of the first-order components on the First dataset (Ishigami function) described through the Legendre basis.

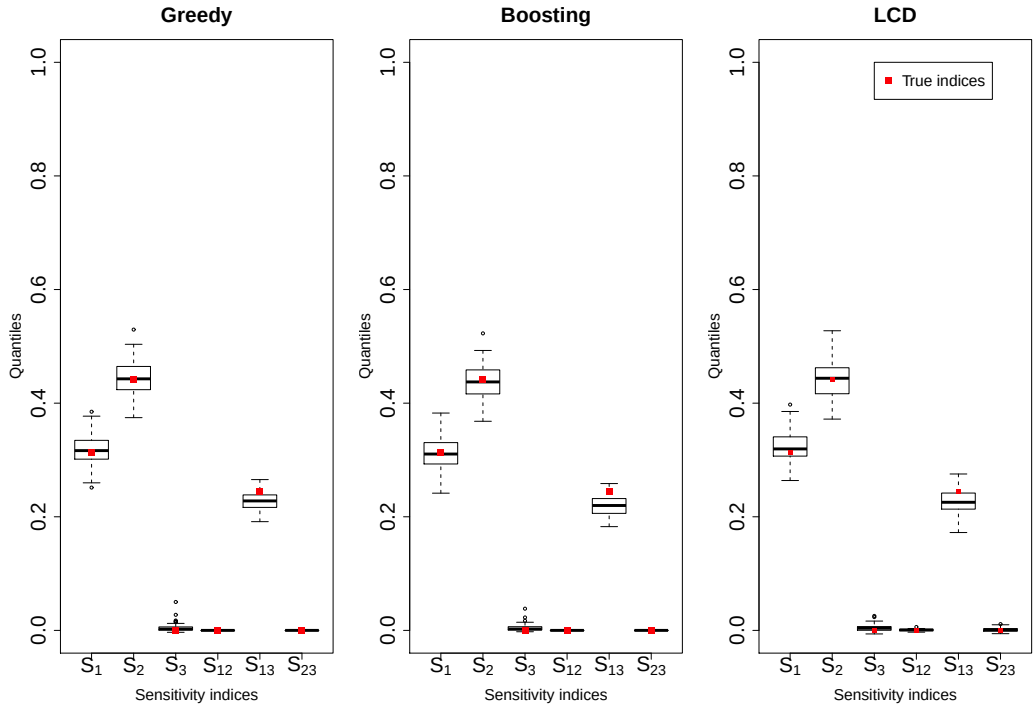


Figure III.3: Representation of the first-order components on the First dataset (Ishigami function) described through the Fourier basis.

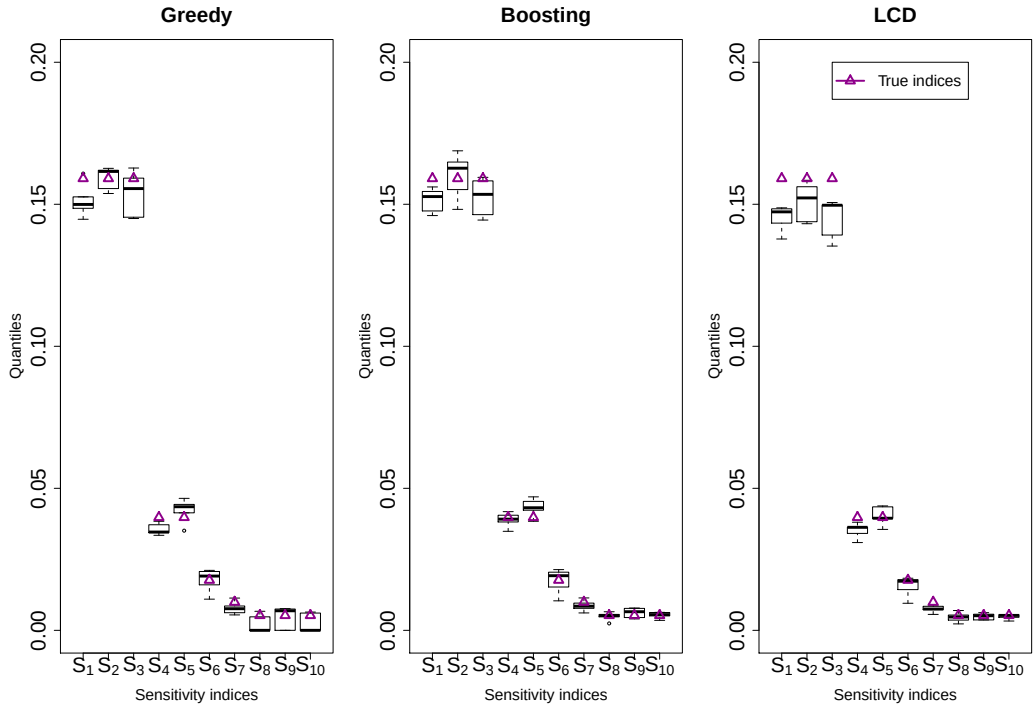


Figure III.4: Representation of the first-order components on the Second dataset (g-Sobol function).

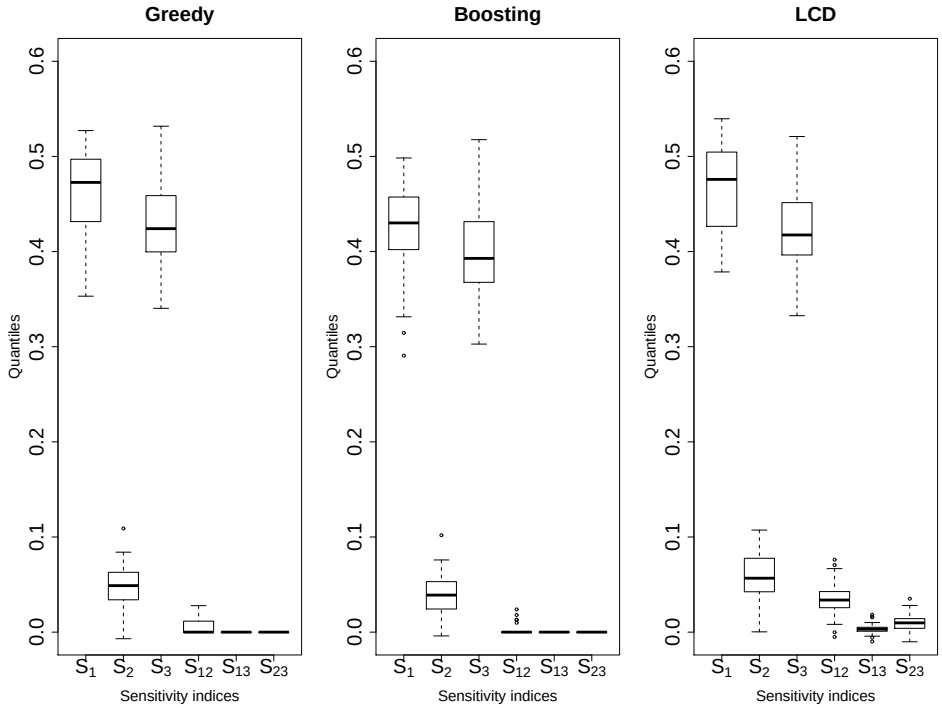
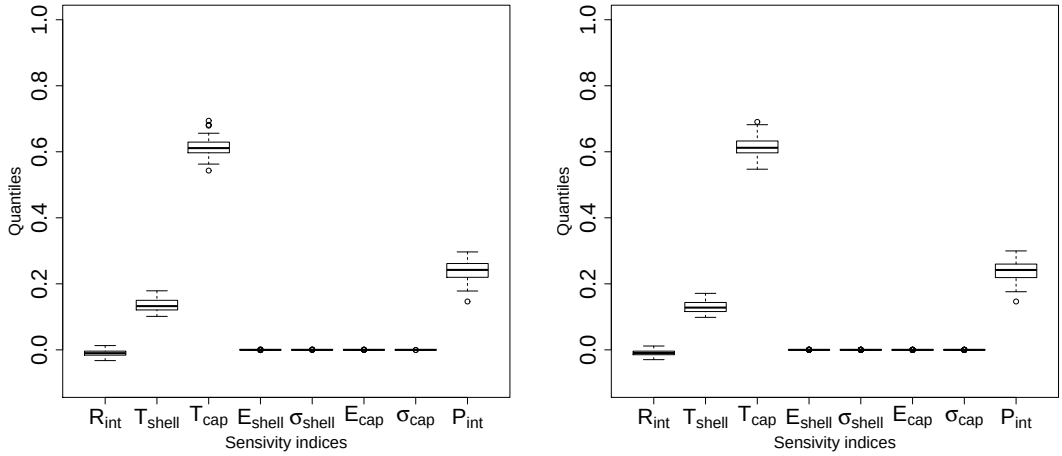
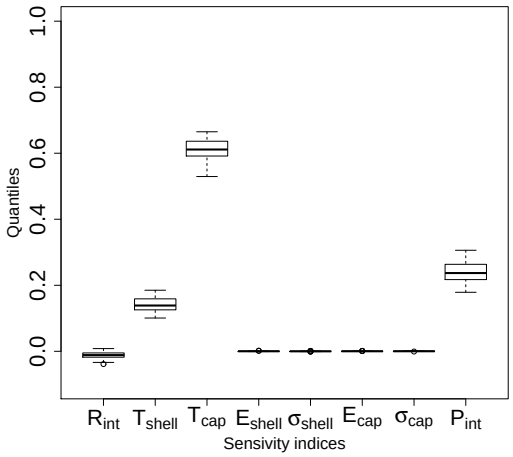


Figure III.5: Representation of the first-order components on the third dataset (dependent inputs).



(a) Greedy (b) Boosting



(c) LCD

Figure III.6: Dispersion of the first order sensitivity indices of the tank model parameters for the 3 methods.

Note that we have computed the maximal "degeneracy" that is involved in the resolution of the linear systems and quantified by Assumption $(\mathbf{H_b^{3,\vartheta}})$ in column 2 of Table III.3. In many cases, we obtain a significantly larger value than 0. The third column of Table III.3 shows the admissible size of the parameter ϑ, and we can check that the number of variables p_n allowed by $(\mathbf{H_b^2})$ and the balance between ξ and ϑ (ξ should be greater than 2ϑ in our theoretical results) is not restrictive since $n^{1-2\vartheta}$ is always significantly greater than $log(m_n)$ in Table III.3.

Data set	Procedure	$\left\|\hat{\boldsymbol{\beta}}\right\|_0$	Elapsed Time (in sec.)
Ishigami function Case 1	$\mathbb{L}_2$-Boosting	19	0.0941
	FoBa	21	2.2917
	LCD	20	2.25
Ishigami function Case 2	$\mathbb{L}_2$-Boosting	15	0.0884
	FoBa	12	1.0752
	LCD	13.9	0.41
g-Sobol function	$\mathbb{L}_2$-Boosting	99	49.8
	FoBa	22.4	827.9
	LCD	91.8	5047.4
Dependent inputs	$\mathbb{L}_2$-Boosting	4.14	0.028
	FoBa	4.76	0.1056
	LCD	24.1	0.061
Tank pressure model	$\mathbb{L}_2$-Boosting	10	0.0266
	FoBa	22	0.3741
	LCD	23	0.15

Table III.2: Features of the three algorithms

Dataset	Degeneracy $d(A)$	$\vartheta \geq \frac{\log(1/d(A))}{log(n)}$	$n^{1-2\vartheta}$	$\log(m_n)$
Ishigami function Case 1	0.6388	$[0.0786, +\infty[$	122.3821	6.0113
Ishigami function Case 2	0.76	$[0.0481, +\infty[$	173.3094	6.0113
g-Sobol function	0.9745	$[0.0034, +\infty[$	1899	8.9392
Dependent inputs	0.628	$[0.101, +\infty[$	39.4457	4.8363

Table III.3: Degeneracy of the linear systems and admissible size of m_n ($n^{1-2\vartheta}$ should be greater than $\log(m_n)$).

Conclusion and perspectives

This work provides a rigorous framework for the hierarchically orthogonal Gram-Schmidt procedure in a high-dimensional paradigm, with the use of the greedy $\mathbb{L}_2$-Boosting. Overall, the procedure falls into the category of sparse estimation with a noisy dictionary, and we demonstrate its consistency up to some mild assumptions on the structure of the real underlying basis. From a mathematical point of view, assumption ($\mathbf{H_b^1}$) presents a restrictive condition, and to relax it would open a wider class of basis functions for applications. We leave this development open for a future study, which could be based either on the development of a concentration inequality for unbounded random matrices or on a truncating argument. It also appears that our algorithm produces very satisfactory numerical results through our three datasets as a result of its very low computational cost. It can also be extended with some further numerical work to a larger truncation order of $d \geq 3$. Such an improvement may also be of interest from a theoretical point of view when dealing with a function that smoothly depends on the interaction order. In particular, a data-driven adaptive choice of d may be of practical interest in the future.

Chapter IV

Estimation of sparse directed acyclic graphs: theoretical framework and Genetic Algorithms

This chapter presents a joint work with Victor Picheny on the development of optimization methods to infer gene regulatory networks. It will be subject of publication next year.

1 Introduction

In the present work, we are interested in the inference of gene regulatory networks (GRN), which model activation and inhibition relationships that exist between genes. These relationships represent the quantitative impact of gene expression on each other, where the expression of a gene roughly represents its activity in the system. As explained by [Pea00], in order to obtain causal statements from observational data only (the expression data) without external interventions like gene knockouts, a standard hypothesis consists in assuming that the data are generated by a Directed Acyclic Graph (DAG). DAGs and corresponding directed graphical models are key concepts for causal inference (see for example [SGS00]).

A variety of mathematical formalisms have been proposed to represent the complex behavior of known gene regulation networks: continuous or discrete, *e.g.* with Boolean networks [Tho73], defined over time, using ordinary differential equations [BdB07] or dynamic Bayesian networks [RJFD10], [LBD$^+$10], or in stationary states [FNP99]. In this chapter, we focus on statistical models particularly adapted to infer gene regulations from expression data. A natural approach, developped by [MB06] to solve the network inference problem is to consider that each gene X^i $(1 \leq i \leq p)$ can be represented as a linear function of all other genes X^j $(j \neq i)$ through the Gaussian Structural Equation Model (SEM):

$$\forall i \in [\![1, p]\!], \quad X^i = \sum_{j=1}^{p} G_j^i X^j + \varepsilon,$$

where ε is a Gaussian residual error term, and G_j^i encodes the relationships from gene X^i to gene X^j.

One of the most challenging problems consists in inferring causality in the graph: the available observational data are in general not sufficient to infer the true DAG that generates the data. More precisely, the observational data provide set of conditional dependencies that only determine an equivalence class of DAGs. An equivalence class of DAGs corresponds to the same probability

distribution [KF09]. So proceeding to inference on dependent or independent relationships from observational data only cannot lead to unambiguous causal structure in general. This approach relies on the assumption that the joint distribution is Markov and faithful with respect to the true graph, *i.e.* the conditional independencies given by the data are only induced by the true DAG [SGS00]. Such an assumption is a major principle of the causal inference. There is a large existing litterature on estimating the equivalence class of DAGs (see for instance [Chi02] and [SDLC93]). One of the most famous methods consists in using the PC-algorithm of [SGS00]. [KB07] provides consistency results for the PC-algorithm in the high-dimensional sparse setting.

However, if we consider particular SEMs, one can show that the graphical structure is identifiable from the observational data only. As an example, the recent works of [PMJS11] and [PB14] on networks with Gaussian data with *equal noise variances* have shown that the true DAG can be identified from the distribution.

In the setting of observational Gaussian data with equal noise variances, we focus on the problem of estimating DAGs. From a computational statistics point of view, a challenging problem deals with the high dimension: the number of genes in a typical genome is much larger than the number of samples. Common methods used for GRN inference in a high-dimensional paradigm are based on penalized linear regressions [MB06] or penalized Bayesian Networks [FLNP00a]. Our framework is to rely on the statistically popular negative log-likelihood. To ensure that the estimated graph is sparse enough, mimicking the works of [Tib96], we consider here a ℓ_1-penalized maximum log-likelihood estimator. The theoretical high-dimensional analysis of the estimator we propose is not trivial due to the unknown order among the variables. For a known order, Shojaie *et al.* present some results for the estimation of high-dimensional DAGs [SM10]. The case with unknown order has already been studied by [vdGB13] for the ℓ_0-penalized maximum log-likelihood. We then relate and compare our results to those obtained by [vdGB13].

From a computational point of view, the ℓ_1-norm makes the criterion to maximize convex, with respect to the graph structure, and finding a solution of this estimation problem is then reduced to a discrete optimization over the non-convex set of DAGs. Following remarks from [Büh13], a new parametrization of the set of DAGs breaks down this problem in two parts: the estimation of the variables order and the estimation of the DAG. For a known order, the latter can be solved using a popular convex optimization algorithm. However, the estimation of the variables order leads to a discrete (and non-convex) optimization problem.

This work is organized as follows: Section 2 is dedicated to a review of networks modelling, we thus present reminders on graph theory and establish a relation between graphs and the corresponding joint distribution. In Section 3, we mainly focus on the estimation of DAGs when the noise variances of each variable are the same. In Section 4, we are interested in a theoretical analysis of the ℓ_1-penalized maximum likelihood estimator. Following the works of [BRT09] and [vdGB13], we prove convergence rate both in prediction and estimation. In Section 5, we propose two algorithms devoted to find a solution of the maximization of the ℓ_1-penalized log-likelihood estimator. The first one, presented in Section 5.2, is an alternating procedure of optimization, consisting in freezing alternately each variable. The second one, presented in Section 5.3, is a hybrid convex Genetic Algorithm suited to the structure of the problem at hand. In Section 6, we apply our method to toy datasets to assess its performance in terms of ability to recover the structure of the initial graph that generates the data and show that it compares favorably to the state-of-the-art.

2 A review of networks modelling

2.1 Reminders on graph theory

We start this section with needed definitions for graphs. A graph $\mathcal{G} = (V, E)$ consists of a finite set of elements V (*e.g.* $\{1, ..., p\}$), called nodes, and a set of pairs of elements of V, $E \subset V^2$, called edges. In our work, the nodes of a graph represent some finite family of random variables $X := (X^1, ..., X^p)$ and the edges between the nodes of the graph, some relationships between these variables. In a slight abuse of notation, we sometimes identify the nodes $j \in V$ with the variables X^j. Denote $\mathcal{L}(X)$ the joint distribution of the observations $X^1, ..., X^p$. Most of the definitions we present here can be found in [SGS00] or [Lau96].

Definition 2.1 (Graph terminology)**.** *Let $\mathcal{G} = (V, E)$ a graph, with $V = \{1, ..., p\}$ and corresponding random variables $X^1, ..., X^p$.*

- *An edge between two nodes is called directed if this edge has an arrowhead, i.e. $X^i \leftarrow X^j$ or $X^i \rightarrow X^j$. If not, this edge is said not directed,*
- *X^i is a parent of X^j if $(i, j) \in E$, i.e. if $X^i \rightarrow X^j$. If $X^i \leftarrow X^j$, X^i is called a child of X^j. Denote $\mathrm{Pa}_{\mathcal{G}}(j)$, respectively $\mathrm{Ch}_{\mathcal{G}}(j)$ the set of parents, respectively children, of X^j. X^i and X^j are said to be adjacent if either X^i is a parent of X^j, or X^j is a parent of X^i.*
- *A v-structure is a triplet of nodes (i, j, k) such that one of the nodes is a child of the two others, which are not adjacent: $X^i \rightarrow X^j \leftarrow X^k$,*
- *A path $(X^{i_1}, ..., X^{i_n})$ is a sequence of distinct nodes of the graph, such that X^{i_k} and $X^{i_{k+1}}$ are adjacent, for all $k = 1, ..., n-1$. This path is directed if, for all $k = 1, ..., n-1$, X^{i_k} is parent of $X^{i_{k+1}}$. Then, for all $k \geq 2$, X^{i_k} is a descendant of X^{i_1}. Denote $\mathrm{Des}_{\mathcal{G}}(i)$, resp. $\mathrm{ND}_{\mathcal{G}}(i)$, the set of descendants of i in $\mathcal{G}$, resp. non-descendants of i.*
- *A (directed) cycle is a (directed) path, which begins and ends with the same node,*
- *If there is no cycle in the graph and all its edges are directed, $\mathcal{G}$ is said to be a Directed Acyclic Graph (DAG).*
- *The skeleton of $\mathcal{G}$ is obtained by removing the orientation from the edges.*

Then, we define the notion of d-separability for subsets of vertices as follows:

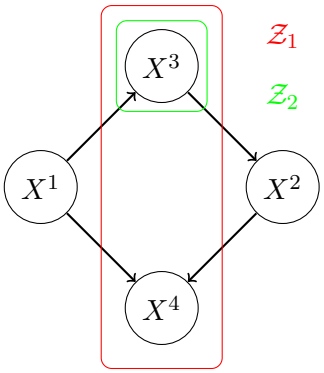
Definition 2.2 (d-separability)**.** *Variables X^i and X^j are blocked by a subset of nodes $\mathcal{Z}$ (with neither X^i nor X^j in this set) whenever there exists a node X^k such that one of the two conditions hold:*

1. *$X^k \in \mathcal{Z}$ and $X^i \rightarrow X^k \rightarrow X^j$, or $X^i \leftarrow X^k \rightarrow X^j$, or $X^i \leftarrow X^k \leftarrow X^j$,*
2. *$X^i \rightarrow X^k \leftarrow X^j$ and neither X^k nor any of its descendants is in $\mathcal{Z}$.*

Two disjoint subsets of nodes $\mathcal{X}$ and $\mathcal{Y}$ are said to be d-separated by a third (also disjoint) subset $\mathcal{Z}$ if every path between nodes in $\mathcal{X}$ and $\mathcal{Y}$ is blocked by $\mathcal{Z}$.

Example 1 enlights the definition of d-separability in a graph.

Example 1. *Consider the following graph $\mathcal{G}$:*



Nodes X^1 and X^2 are d-separated by $\mathcal{Z}_2$ but not d-separated by $\mathcal{Z}_1$ since the path $X^1 \to X^4 \leftarrow X^2$ isn't blocked by X^4.

A standard assumption for the joint distribution $\mathcal{L}(X)$ that generates a DAG $\mathcal{G}$ is the Markov condition. Throughout this work, $\perp$ denotes the conditional independence between two variables or groups of variables under the joint distribution $\mathcal{L}(X)$.

Definition 2.3 (Markovian distribution)**.** *The joint distribution $\mathcal{L}(X)$ is said to be Markov with respect to the DAG $\mathcal{G}$ if for all disjoint sets $\mathcal{X}$ and $\mathcal{Y}$ d-separeted by $\mathcal{Z}$, $\mathcal{X}$ and $\mathcal{Y}$ are conditionally independent given $\mathcal{Z}$:*

$$\mathcal{X}, \mathcal{Y} \quad d-sep. \ by \quad \mathcal{Z} \Rightarrow \mathcal{X} \perp \mathcal{Y} | \mathcal{Z}.$$

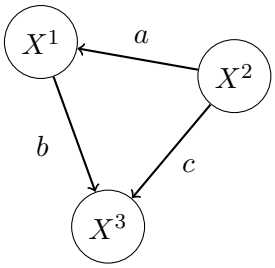
Given a graph, the Markov condition defines a set of conditionally independent variables that corresponds to the d-separated variables. However, other independent variables may appear in the graph: if the conditional independences induced by the distribution $\mathcal{L}(X)$ are encoded by the Markov condition, $\mathcal{L}(X)$ is said to be faithful with respect to $\mathcal{G}$.

Definition 2.4 (Faithfulness)**.** *The distribution $\mathcal{L}(X)$ is called faithful with respect to $\mathcal{G}$ if the conditional independences of $\mathcal{L}$ are the same as those encoded by $\mathcal{G}$ via d-separability:*

$$\mathcal{X}, \mathcal{Y} \quad d-sep. \ by \quad \mathcal{Z} \Leftrightarrow \mathcal{X} \perp \mathcal{Y} | \mathcal{Z}.$$

In graph theory, the faithfulness assumption is particularly needed to establish the connection between graph and distribution. We give below an example of distribution that is not faithful with respect to some DAG $\mathcal{G}$. This is achieved by making variables be independent, whereas they should not be according to the graph structure.

Example 2. *Consider the following graph $\mathcal{G}$:*



Corresponding to this graph, we generate a joint distribution by the following equations:

$$\begin{cases} X^1 &= aX^2 + \varepsilon^1, \\ X^2 &= \varepsilon^2, \\ X^3 &= cX^1 + bX^2 + \varepsilon^3, \end{cases}$$

where $\varepsilon^1, \varepsilon^2$ and ε^3 are some Gaussian noises, and we assume $ac+b=0$. Remark that this is an example of a linear Gaussian Structural Equation Model that we formally define below in Section 2.2. Then, we have:

$$\begin{aligned} X^3 &= c(aX^2+\varepsilon^1)+bX^2+\varepsilon^3 \\ &= \varepsilon^1+\varepsilon^3. \end{aligned}$$

As a consequence, X^3 and X^2 are conditionally independent given X^1. However, since X^2 and X^3 are not d-separated by X^1 in the graph $\mathcal{G}$, the joint distribution is not faithful with respect to $\mathcal{G}$.

Given a DAG $\mathcal{G}$, we denote by $\mathcal{M}(\mathcal{G})$ the set of distributions that are Markov with respect to $\mathcal{G}$:

$$\mathcal{M}(\mathcal{G}) = \{\mathcal{L}(X), \mathcal{L}(X) \text{ is Markov with respect to } \mathcal{G}\}.$$

Two DAGs $\mathcal{G}_1$ and $\mathcal{G}_2$ are said to be Markov equivalent if:

$$\mathcal{M}(\mathcal{G}_1) = \mathcal{M}(\mathcal{G}_2).$$

This is the case if and only if $\mathcal{G}_1$ and $\mathcal{G}_2$ share the same set of d-separations, which means that the Markov condition entails the same set of conditional independencies. The set of DAGs that are Markov equivalent to some DAG $\mathcal{G}$ forms the Markov equivalence class of $\mathcal{G}$. Under assumption of faithfulness, this equivalence class can be described by a Completed Partially Directed Acyclic Graph (CPDAG) using Proposition 2.1 from [VP91].

Proposition 2.1. *Two graphs are Markov-equivalent if and only if they have the same skeleton ant the same v-stuctures.*

Graphically, given a DAG $\mathcal{G}$, its Markov equivalence class is obtained by removing the orientation of edges not include in a v-structure. Figure IV.1 gives an example of two graphs Markov equivalent.

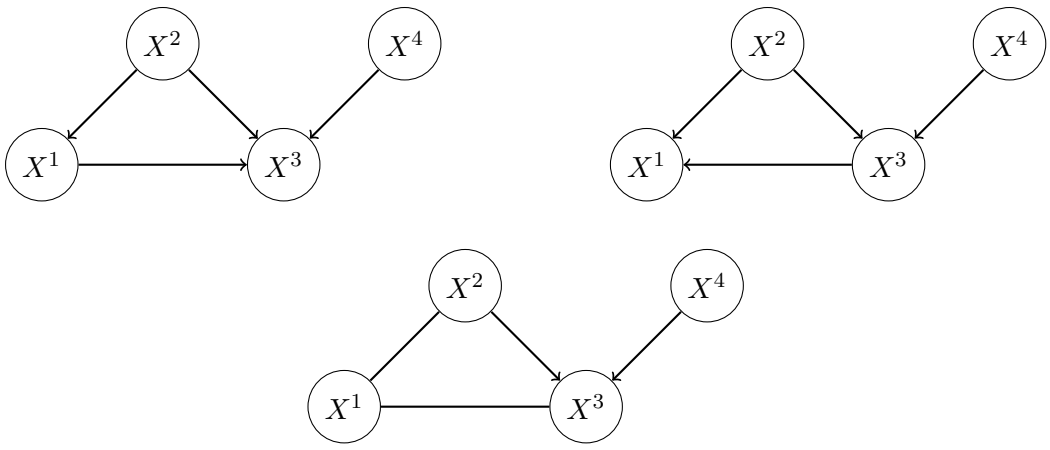


Figure IV.1: At the top, two graphs Markov equivalent (they have the same skeleton and the same only v-structure $X^2 \to X^3 \leftarrow X^4$). At the bottom, their Markov equivalence class is a CPDAG.

2.2 Structural Equation Models (SEM) for inferring DAGs

In this section, we present the model we are working on and its graphical representation. Consider the following model: let $X = (X^1, ..., X^p)$ variables satisfying the following equation:

$$\forall i \in [\![1, p]\!], \quad X^i = f_i\left(X^{\mathcal{S}^i}, \varepsilon^i\right), \tag{IV.1}$$

where $\varepsilon := (\varepsilon^1, ..., \varepsilon^p)$ are noise parameters, supposed to be independent and identically distributed according to the law $\mathcal{L}(\varepsilon)$, and $(f_i)_{1\leq i\leq p}$ are unknown functions to be estimated. $(\mathcal{S}^i)_{1\leq i\leq p}$ correspond to subsets of $[\![1, p]\!]$, and for $i \in [\![1, p]\!]$, $X^{\mathcal{S}^i}$ denotes the set of variables with corresponding indices in $\mathcal{S}^i$.

This model is known as a Structural Equation Model (SEM). The graph associated to a SEM (IV.1) is obtained by drawing directed edges from each variable X^j with $j \in \mathcal{S}^i$ to its direct consequence X^i. According to the previous notations, for $i \in [\![1, p]\!]$, the set $\mathcal{S}^i$ corresponds to the parents of node X^i. In [Pea00], Pearl shows that the distribution $\mathcal{L}(X)$ generated by a SEM is Markov with respect to the graph. We also assume that $\mathcal{L}(X)$ is faithful with respect to $\mathcal{G}$.

As shown by [MB06], when inferring DAGs, we often consider linear functions f_i to encode relationships between gene expression levels:

$$\forall i \in [\![1, p]\!], \quad X^i = \sum_{j\in\mathcal{S}^i} G^i_j X^j + \varepsilon^i, \tag{IV.2}$$

where $G^i := {}^t(G^i_1, ..., G^i_p)$ is the p-vector of linear effects of expression levels of all genes on gene X^i.

2.3 Identifiability

The question of identifiability of the model is a central concern for statisticians. Given a joint distribution $\mathcal{L}(X) = \mathcal{L}(X^1, ..., X^p)$ of Model (IV.1), is it possible to recover the true (unknown) graph $\mathcal{G}_0$ associated to the model? Obviously not always: the joint distribution of the observations is certainly Markov with respect to a large number of different DAGs that are different from the true DAG. Some supplementary assumptions have to be supposed to obtain identifiability of the model.

Provided that $\mathcal{L}(X)$ is faithful with respect to the true DAG $\mathcal{G}_0$, the equivalence class of Markov of $\mathcal{G}_0$ can be obtained (see Proposition 2.2 below). Indeed, the conditional independencies induced by $\mathcal{L}(X)$ are the same as those encoded by any graph $\mathcal{G}$ Markov equivalent to $\mathcal{G}_0$. If $\mathcal{G}$ is not Markov equivalent to $\mathcal{G}_0$, then, there is at least one conditional independence in $\mathcal{G}$ that is not in $\mathcal{G}_0$, or vice versa. But then, $\mathcal{L}(X)$ cannot be faithful with respect to $\mathcal{G}$.

Proposition 2.2 (Identifiability of the Markov equivalence class [Pea00])**.** *If $\mathcal{L}(X)$ is Markov and faithful with respect to the graph $\mathcal{G}_0$, then, the Markov equivalence class of $\mathcal{G}_0$ is identifiable from $\mathcal{L}(X)$.*

The estimation of the Markov equivalence class of $\mathcal{G}_0$, which may still be large, is one of the most challenging problem when inferring a DAG. The methods developed to this end try to avoid checking all possible conditional independencies in $\mathcal{L}(X)$. A first approach is provided by [SGS00]: the PC-algorithm is based on a clever hierarchical scheme for multiple testing conditional independencies among the variables of the graph. The procedure, developped in [SG91], starts by forming the complete undirected graph, then reduces this graph by removing edges using zero order conditional independence relations, reduces it again with first order conditional

independence relations, and so on. Due to the faithfulness assumption and assuming sparsity of the DAG, the algorithm is computationally feasible for settings where the number of nodes p is in the thousands.

Following the works of [AMP97], it is possible to infer a subset of the Markov equivalence class by careful considerations on the edges to be oriented. The so-called essential graph, is defined as the class of graphs that have:

— the same skeleton and the same v-structures,

— the same directed edges.

We do not detail more this section, but the interested reader can refer to [AMP97].

3 Estimation of DAGs

As presented in Section 2, observational data alone are not sufficient in general to orient the edges in a DAG. In Gaussian SEMs with linear functions, the GRN can be identified from the joint distribution only up to a Markov equivalence class (assuming faithfulness). In this work, we restrict our study to a particular case of Model (IV.2) for which the DAG becomes identifiable. Section 3.1 is thus dedicated to the presentation of the model. In Section 3.2, we recall the proof of identifiability from [PB14]. In Section 3.3, we finally propose a method of estimation of the true DAG based on the maximization of the the log-likelihood.

3.1 The settings: restriction to Gaussian SEMs with same error variances

The model is the following: all the functions f_i of Equation (IV.1) are linear and the noise parameters are i.i.d, distributed according to a Gaussian law $\mathcal{N}(0, \sigma^{\mathbf{2}})$:

$$\forall i \in [\![1,p]\!], \qquad X^i = \sum_{\substack{1\leq j\leq p \\ j\neq i}} (G_0)^i_j X^j + \varepsilon^i, \tag{IV.3}$$
$$\varepsilon^i \sim \mathcal{N}(0, \sigma^{\mathbf{2}}).$$

Denote $\mathcal{G}_0$ the graph induced by Equation (IV.3), for which the directed edges of $\mathcal{G}_0$ correspond to the non-zero coefficients of the matrix $G_0 = ((G_0)^i_j)_{1\leq i,j\leq p}$. As an example, if $(G_0)^i_j$ equals 0, then, X^i cannot be a parent of X^j. Following the previous notations, denote $\mathcal{S}^i$ the set of parents of X^i, corresponding to the non-zero coefficients of the i-th column of G_0:

$$\forall i \in [\![1,p]\!], \quad \mathcal{S}^i = \left\{ j \in [\![1,p]\!], (G_0)^i_j \neq 0 \right\}.$$

Assume that we observe n i.i.d observations from the model given by Equation (IV.3). We denote by X the $n \times p$ observation matrix, composed of n i.i.d rows, distributed according to a $\mathcal{N}(0, \Sigma_0)$. The relations between the variables in a row, given by Equation (IV.3), can be represented as:

$$X = XG_0 + \varepsilon.$$

Since the noise variances are the same for all variables, the covariance matrix of ε equals the identity matrix up to a multiplicative scalar σ^2. Remark that Lemma 3.1 below (see Section 3.2) implies that ε^j is independent of X^i as soon as $(G_0)^i_j = 0$. We aim at estimating the non-zero coefficients of matrix G_0 to recover the true DAG that generates the data.

3.2 Identifiability of the model

Under assumption of faithfulness, Theorem 3.1, introduced by [PB14], shows the identifiability of Model (IV.3) in the case of equal noise variances. It ensures that we may recover the true graph given the observations.

Theorem 3.1 (Identifiability [PB14]). *Let $\mathcal{L}(X)$ generated according to Equation (IV.3) and assume that it is Markov and faithful with respect to the graph $\mathcal{G}_0$. Then, the graph $\mathcal{G}_0$ is identifiable given $\mathcal{L}(X)$.*

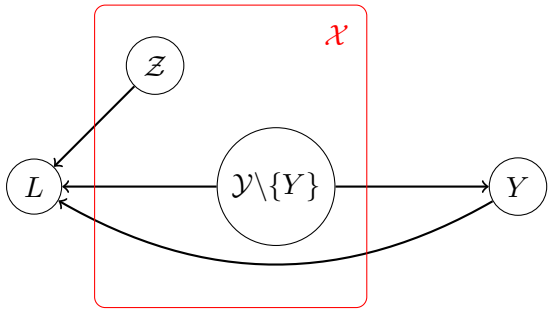
The proof of this result consists in proving that two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ that have the same joint law are identical. Assuming faithfulness, these two graphs are necessarily equivalent: they have the same skeleton and only few edges are differently oriented. Then, beginning with the sink node L of graph $\mathcal{G}_1$ (the node without outcoming edges), we recursively show that all the adjacent nodes of L in $\mathcal{G}_2$ are also parents of L. We conclude the proof by considering all the generations in the graphs.

Proof. Let $\mathcal{G}_1$ and $\mathcal{G}_2$ two graphs, generated according to the same joint law $\mathcal{L}(X)$. Remark that there exists a node L that has no descendant in the DAG $\mathcal{G}_1$. Assuming faithfulness, Proposition 2.2 implies that the graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are Markov equivalent. As a consequence, $\mathcal{G}_1$ and $\mathcal{G}_2$ can be decomposed as follows:

Graph $\mathcal{G}_1$ Graph $\mathcal{G}_2$

We partition the parents of L in $\mathcal{G}_1$ into $\mathcal{Z}$ and $\mathcal{Y}$. $\mathcal{Z}$ are also parents of L in $\mathcal{G}_2$ whereas $\mathcal{Y}$ are children of L in $\mathcal{G}_2$. We then aim at proving that $\mathcal{Y} = \emptyset$.

Consider the subgraph $\mathcal{G}_{1|\mathcal{Y}}$ of $\mathcal{G}_1$ restricted to the elements of $\mathcal{Y}$. This subgraph is still a DAG and there exists at least a node $Y \in \mathcal{Y}$ that has no children in $\mathcal{G}_{1|\mathcal{Y}}$:



Graph $\mathcal{G}_1$

Then, denote $\mathcal{X} = \mathcal{Z} \cup \mathcal{Y}\backslash\{Y\}$. Let $x = (z, y) \in \mathcal{X}$. The three following lemmas, given by [PB14], are purely technical. The first one deals with the structure of the considered graphs. The second and the third one mainly use arguments of independence, Gaussian vectors and conditional laws. We do not prove them here.

Lemma 3.1. *Let $\mathcal{G}$ be a graph, generated according to a law $\mathcal{L}(X)$ that follows a SEM as in (IV.1). Let X^i be a random variable. Then, the following result holds:*

$$\forall \mathcal{Y} \subset \mathrm{ND}_{\mathcal{G}}(X^i), \quad \varepsilon^i \perp \mathcal{Y}.$$

Proof of Lemma 3.1. Let $\mathcal{Y} \subset \mathrm{ND}_{\mathcal{G}}(X^i)$. Then, each element Y of $\mathcal{Y}$ can be written as a function of $\mathrm{Pa}(Y)$:

$$\forall Y \in \mathcal{Y}, \quad Y = f_Y\left(\mathrm{Pa}(Y), \varepsilon^{\mathrm{Pa}(Y)}\right).$$

Moreover, one can substitute recursively the parents of Y by the corresponding functional equations. After a finite number of iterations, we obtain $Y = f_{Y^1,...,Y^t}(\varepsilon^{Y^1}, ..., \varepsilon^{Y^t})$, where $\{Y^1, ..., Y^t\}$ is the set of all ancestors of Y, which does not contain X^i by assumption. Since all noise variables are jointly independent, we conclude the proof for all $Y \in \mathcal{Y}$. □

Lemma 3.2. *Let A, B, C and D random variables taking values in $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and $\mathcal{D}$. Let $f : \mathcal{A}, \mathcal{B}, \mathcal{C} \to \mathbb{R}$ be a measurable function. If $C \perp (A, B, D)$, then, for all $b \in \mathcal{B}, d \in \mathcal{D}$ such that the joint density $p_{B,D}(b, d) > 0$, we have:*

$$f(A, B, C)_{|B=b,D=d} = f\left(A_{|B=b,D=d}, b, C\right).$$

Remark that, for clarity purpose, the equalities above are equalities in distribution: the notation $f(A, B, C)_{|B=b,D=d}$ means the law of $f(A, B, C)$ given $B = b$ and $D = d$.

Lemma 3.3. *Let $(A_1, ..., A_m) \sim \mathcal{N}\left((\mu_1, ..., \mu_m)^T, \Sigma\right)$ where Σ is strictly positive definite. Let $A_1^* = A_{1|(A_2,...,A_m)=(a_2,...,a_m)}$, with $(a_2, ..., a_m) \in \mathbb{R}^{m-1}$. Then,*

$$var(A_1^*) \leq var(A_1).$$

On the one hand, in the graph $\mathcal{G}_1$, L can be written as a function of its set of parents $\mathcal{X}$ and Y:

$$L = f_L\left(\mathcal{X}, Y, \varepsilon^L\right).$$

Remark that node L has no descendants in $\mathcal{G}_1$. Using Lemma 3.1, we thus deduce that $\varepsilon^L \perp (\mathcal{X}, Y)$. The following equality holds using Lemma 3.2 :

$$\begin{aligned} L_{|\mathcal{X}=x} &= f_L\left(x, Y_{|\mathcal{X}=x}, \varepsilon^L\right) \\ &= f_L(x) + \beta Y_{|\mathcal{X}=x} + \varepsilon^L, \end{aligned}$$

where $\beta > 0$ is the coefficient that encodes the relationships between L and Y. We then have:

$$\mathrm{Var}\left(L_{|\mathcal{X}=x}\right) = \beta^2 \mathrm{Var}\left(Y_{|\mathcal{X}=x}\right) + \sigma^2 > \sigma^2. \tag{IV.4}$$

On the other hand, on the graph $\mathcal{G}_2$, we have:

$$L = f_L\left(\mathcal{Z}, \varepsilon^L\right).$$

Now, some elements of $\mathcal{X}$ are descendants of L in $\mathcal{G}_2$. We thus deduce that $\varepsilon^L \not\perp \mathcal{X}$, and we have:

$$\begin{aligned} L_{|\mathcal{X}=x} &= f_L\left(z, \varepsilon^L_{|\mathcal{X}=x}\right) \\ &= f_L(z) + \varepsilon^L_{|\mathcal{X}=x}. \end{aligned}$$

Then, using Lemma 3.3 the following equality holds:

$$\mathrm{Var}\left(L_{|\mathcal{X}=x}\right) = \mathrm{Var}\left(\varepsilon^L_{|\mathcal{X}=x}\right) \leq \mathrm{Var}\left(\varepsilon^L\right) = \sigma^2. \tag{IV.5}$$

Equation (IV.4) contradicts Equation (IV.5): all edges adjacent to L are directed toward L. To end the proof of Theorem 3.1, we process recursively, crossing the graph $\mathcal{G}_1$ along its edges. □

Provided the identifiability of the model, we aim at finding an estimator of the true graph, which satisfies good theoretical properties. In the next section, we consider the log-likelihood estimator.

3.3 The ℓ_1-penalized maximum likelihood estimator

To estimate the non-zero coefficients of the unknown matrix G_0, a statistically popular score function is the negative log-likelihood score, penalized with the dimension of the model:

$$\hat{G} = \underset{G \in \mathcal{G}_{DAG}}{\operatorname{argmin}} \left\{ l(G) + \lambda \operatorname{pen}(G) \right\}, \tag{IV.6}$$

where $l(.)$ is the log-likelihood, pen(.) a penalization function, λ a parameter which controls the amount of penalization and $\mathcal{G}_{DAG}$ the set of matrices compatible with a DAG. To make the estimated graph sparse enough, one solution consists first in considering a ℓ_0-penalty. From a computational point of view, the main difficulty to optimize (IV.6) over the space of DAGs is to explore this set. Some algorithms, such as the dynamic programming method [SM06], propose to optimize the ℓ_0-penalized log-likelihood using a particular decomposition of the objective function. The greedy equivalent search algorithms [HB12] restrict the search space (set of DAGs) to the smaller space of equivalence classes. They provide an efficient algorithm without enumerating all the equivalent DAGs.

The main advantage of the ℓ_0-penalty is that the objective function to optimize is constant over the Markov equivalence class, which is identifiable. However, according to Proposition 2.2, in the particular case we focus on, the model is identifiable. We thus propose to make the criterion convex in Equation (IV.6) by considering the ℓ_1-norm penalty. The price to pay for this relaxation is a bias, which should be controlled by thresholding the estimator [vdGBZ11].

$$\hat{G} = \underset{G \in \mathcal{G}_{DAG}}{\operatorname{argmin}} \left\{ l(G, \sigma) + \lambda \|G\|_1 \right\}.$$

In this setting, the bigger the λ, the sparser the estimated graph.

The log-likelihood of Model (IV.3) is given by Proposition 3.1 below.

Proposition 3.1. *Assume that we observe an n i.i.d. sample of Equation (IV.3). Then, the log-likelihood of the model is:*

$$l(G) = \frac{1}{n} \sum_{k=1}^{n} \sum_{j=1}^{p} \left((X - XG)_k^j \right)^2 = \frac{1}{n} \|X - XG\|_F^2 .$$

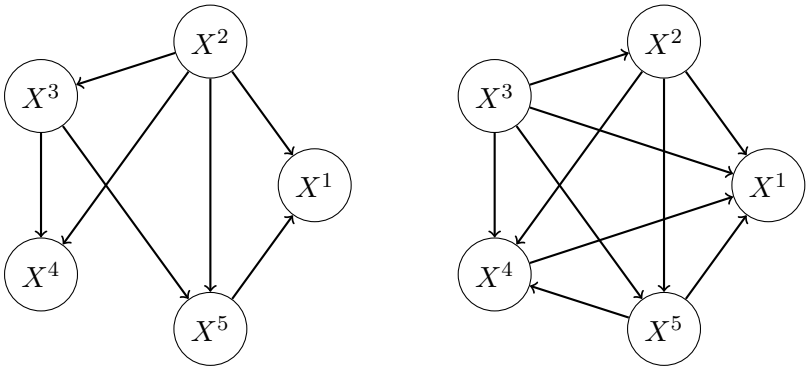
Given Proposition 3.1, the problem of minimization of the penalized log-likelihood can be written as:

$$\hat{G} = \underset{G \in \mathcal{G}_{DAG}}{\operatorname{argmin}} \left\{ \frac{1}{n} \|X - XG\|_F^2 + \lambda \|G\|_1 \right\}. \tag{IV.7}$$

3.4 A new formulation for the estimator

The optimization problem in (IV.7) is compuationally extremely challenging because of the large dimension of $\mathcal{G}_{DAG}$ and its non-convexity. A key property is that any DAG leads to a partial order, denoted $\leq$, on its vertice, where $X^i \leq X^j$ is equivalent to "node X^i has a larger number of parents than node X^j". This ordering is not unique in general, except when each node of the graph has a different number of parents [CLRS09] (see Example 3 below for more explanations). This is the case when the graph is complete, *i.e.* when all nodes are connected to each other. On the contrary, the sparser the graph, the more orderings of the variables exist.

Example 3. *Consider the two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ given by:*



Graph $\mathcal{G}_1$ *Graph $\mathcal{G}_2$*

For graph $\mathcal{G}_1$ (on the left), several ordering of the nodes are possible such that $X^5 \leq X^4 \leq X^1 \leq X^3 \leq X^2$ or $X^4 \leq X^1 \leq X^5 \leq X^3 \leq X^2$. Since nodes $\{X^5, X^4, X^1\}$ have the same number of parents, six orders are possible for $\mathcal{G}_1$.

Now, looking at the graph $\mathcal{G}_2$ (on the right), a unique hierarchical order exists between the nodes of the graph, given by $X^1 \leq X^4 \leq X^5 \leq X^2 \leq X^3$.

Proposition 3.2 below then gives an equivalent condition for a matrix to be compatible with a DAG.

Proposition 3.2. *A matrix G is compatible with a DAG $\mathcal{G}$ if and only if there exists a permutation matrix P and a stricly lower triangular matrix T such that:*

$$G = PT^tP.$$

Proof. Proposition 3.2 is pointed by [Büh13] and we propose here an original proof of this result.

Let G a matrix defined as $G = PT^tP$, where P is a permutation matrix and T is a stricly lower triangular matrix. We aim at showing that G is compatible with a DAG. One has to remark that the matrix T is compatible with a DAG. As a consequence, ${}^tPGP$, obtained from G permuting the nodes of the graph $\mathcal{G}$ associated to G, is also compatible with a DAG. This ends the first part of the proof.

For the second part of the proof, we assume that G is compatible with a DAG. We provide here an algorithm devoted to write G as a combination of a permutation matrix and a stricly lower triangular matrix. The result is proved by induction on the dimension of the matrix G.

Let G a 2×2 matrix compatible with a DAG, composed of two nodes. G can be necessarily written as:

$$\begin{pmatrix} 0 & 0 \\ a & 0 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} 0 & a \\ 0 & 0 \end{pmatrix}.$$

The conclusion holds with $P = I_2$ and $T = G$ for the first case. For the second case, set $P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and $T = \begin{pmatrix} 0 & 0 \\ a & 0 \end{pmatrix}$. Assume then that the result is true for matrices compatible with DAGs of size stricly smaller than p. Let G a $p \times p$ matrix compatible with a DAG $\mathcal{G}$. Remark that there exists a node c such that $\mathrm{Ch}_{\mathcal{G}}(c) = \emptyset$ and a node p such that $\mathrm{Pa}_{\mathcal{G}}(p) = \emptyset$. Then, the c-th row of G and the p-th column of G equal zero. Consider now the permutation P_1, that consists in switching column p of G with its last column, and row e of G with its first line. We

then have:

$$ {}^tP_1GP_1 = \begin{pmatrix} 0 & \cdots & 0 \\ \times & \tilde{G} & \vdots \\ \times & \cdots & 0 \end{pmatrix}. $$

$\tilde{G}$ is a squared matrix of size $p-2$, and corresponds to a subgraph of $\mathcal{G}$. $\tilde{G}$ is still a DAG. By induction, there exists a permutation matrix P_2 of size $p-2$ and a stricly lower triangular matrix $\tilde{T}$ such that ${}^tP_2\tilde{G}P_2 = \tilde{T}$. Then, consider the permutation matrix P defined as:

$$ P = P_1 \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \vdots \\ \vdots & & P_2 & \\ 0 & \cdots & 0 & 1 \end{pmatrix}. $$

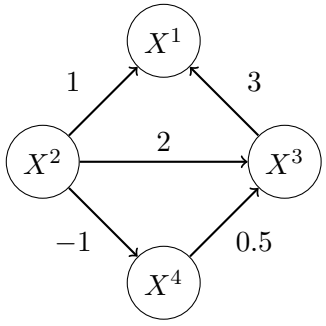
We thus have,

$$ \begin{aligned} {}^tPGP &= \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \vdots \\ \vdots & & {}^tP_2 & \\ 0 & \cdots & 0 & 1 \end{pmatrix} {}^tP_1GP_1 \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \vdots \\ \vdots & & P_2 & \\ 0 & \cdots & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \vdots \\ \vdots & & {}^tP_2 & \\ 0 & \cdots & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & \cdots & 0 \\ \times & \tilde{G} & \vdots \\ \times & \cdots & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \vdots \\ \vdots & & P_2 & \\ 0 & \cdots & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & \cdots & 0 \\ \times & \tilde{T} & \vdots \\ \times & \cdots & 0 \end{pmatrix}, \end{aligned} $$

which ends the proof. □

Example 4 below gives an example of decomposition of the matrix G associated to a DAG $\mathcal{G}$ as $G = PT^tP$.

Example 4. *Consider the graph $\mathcal{G}$ given by:*



The corresponding matrix G can be decomposed as follows:

$$G = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & -1 \\ 3 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{P} \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 1 & 2 & -1 & 0 \end{pmatrix}}_{T} {}^{t} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Graphically, the permutation matrix fixes an ordering of the nodes of the graph: the first column of P indicates that the node with the largest number of parents is X^1, whereas the node with the smallest number of parents is X^2 (see the last column of P). The permutation matrix is thus associated to a complete graph. The stricly lower triangular matrix T fixes the graph structure (non-zero entries of G_0). For more details, see Figure IV.2.

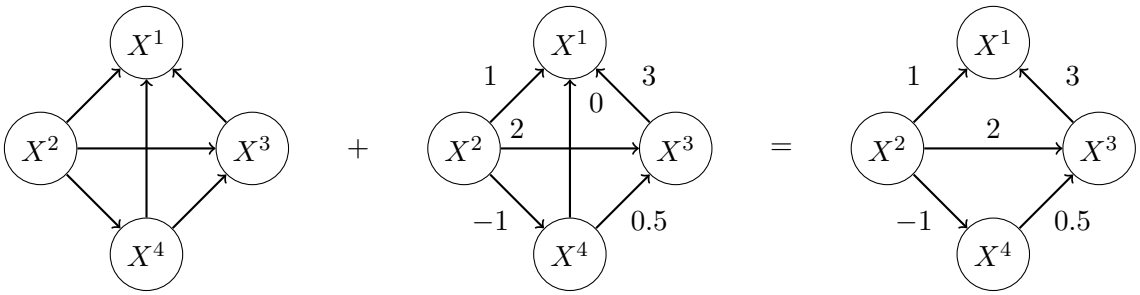


Figure IV.2: Effects of matrices P and T on the graph $\mathcal{G}$: a combination between ordering of the variables and sparsity of the graph.

Using Proposition 3.2, the estimator given by Equation (IV.7) leads to the following optimization problem:

$$(\hat{P}, \hat{T}) = \underset{P \in \mathbb{P}_p(\mathbb{R}), T \in \mathbb{T}_p(\mathbb{R})}{\operatorname{argmin}} \left\{ \frac{1}{n} \left\| X - XPT^tP \right\|_F^2 + \lambda \|T\|_1 \right\}, \tag{IV.8}$$

where $\mathbb{P}_p(\mathbb{R})$ is the set of permutation matrices and $\mathbb{T}_p(\mathbb{R})$, the set of stricly lower triangular matrices. This new parametrization is particularly useful to separate the DAG structure search in two tasks: the ordering estimation and the graph structure learning. It allows us to obtain theoretical bounds both in prediction and estimation (see Section 4).

From a computational point of view, Equation (IV.8) seems to be more tractable too. First, since we add edges to estimate from the initial sparse graph. Then, since the problem of optimizing the log-likelihood over the space of DAGs is reduced to a discrete exploration of the set of permutation matrices. In the litterature, there exists a large number of algorithms devoted to solve discrete optimization problems such that simulated annealing [Kir84], Genetic Algorithms [Mic94],... The optimization procedure we used is presented in the dedicated Section 5.

4 Main theoretical results

The aim of this section is to provide a convergence rate of our estimation, both in prediction and estimation for the ℓ_1-penalized maximum likelihood estimator considered in Equation (IV.7).

Following the work of [vdGB13] on the ℓ_0-penalized maximum likelihood estimator and the work of [BRT09] on the Lasso and the Dantzig Selector, we obtain two convergence results under some mild sparsity assumptions, when the number of variables is large, but upper bounded by a function $\varphi(n)$ of the sample size n.

4.1 The order of the variables

As presented in Section 3.4, the problem of maximization of the log-likelihood (IV.7) on the set of matrices G compatible with a DAG can be written as the problem of optimization (IV.8), introducing two variables $(P, T) \in \mathbb{P}_p(\mathbb{R}) \times \mathbb{T}_p(\mathbb{R})$ given by Proposition 3.2.

Denote Π_0 the set of permutation matrices defined as follows:

$$\Pi_0 = \left\{ P \in \mathbb{P}_p(\mathbb{R}), {}^tPG_0P \text{ is strictly lower triangular} \right\}.$$

An interesting question is: "does the estimated order of variables $\hat{P}$ given by Equation (IV.8) is in Π_0"? This problem is the key point of this section. Let $G = PT{}^tP$ a matrix of a DAG $\mathcal{G}$. In a slight abuse of notation, we identify P with the permutation that hierarchically orders the variables of the DAG $\mathcal{G}$ (see Example 1).

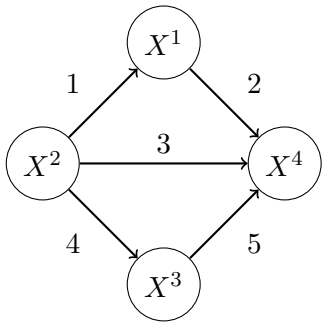
Remark that given Model (IV.3), the vector $XG_0^j = \sum_{i=1}^p (G_0)_i^j X^i$ is the projection of X^j on the linear space spanned by $(X^i)_{i \in \mathrm{Pa}(X^j)}$. Moreover, ε^j corresponds to the anti-projection, *i.e.* what is left after projecting, namely $\varepsilon^j = X^j - XG_0^j$ and satisfies Lemma 3.1: as soon as $(G_0)_j^i = 0$, ε^j is independent of X^i.

For a permutation $P \in \mathbb{P}_p(\mathbb{R})$, we denote by $X(P)$ the matrix obtained from X after permutation of its columns: $X(P) := (X^{P(p)}, ..., X^{P(1)})$. Then, we define the matrix $G_0(P)$ as the matrix G_0 given in this new basis and $\varepsilon(P)$ as the residual term (or anti-projection):

$$\varepsilon^j(P) = X^j - XG_0^j(P).$$

$G_0(P)$ is also stricly lower triangular, after permutations of its rows and columns. An example of such a basis change is provided in Example 5 below. For more details on this procedure, one can also refer to [vdGB13].

Example 5. *Consider the graph $\mathcal{G}_0$ given by:*



The corresponding matrix G_0 can be written as $G_0 = P_0T_0{}^tP_0$, where:

$$P_0 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \quad \textit{and} \quad T_0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 \\ 3 & 1 & 4 & 0 \end{pmatrix}.$$

Let $P \in \mathbb{P}_p(\mathbb{R})$ any permutation matrix, e.g:

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Then, $X(P)$ is defined by switching the columns of X: $X(P) := (X^4, X^2X^3, X^1)$ and $G_0(P)$ becomes:

$$G_0(P) = PT_0{}^tP = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 3 & 4 & 1 & 0 \end{pmatrix}.$$

Lemma 4.1 below provides information about the structure of $\varepsilon(P)$.

Lemma 4.1. *The variables $(\varepsilon^j(P))_{1\leq j\leq p}$ are independent and the covariance matrix $\Omega_0(P)$ of $\varepsilon(P)$ is diagonal.*

Proof. Denote by $\mathcal{G}_0(P)$ the graph associated to $G_0(P)$. A consequence for Lemma 3.1 is that, for all $j \in [\![1,p]\!], \varepsilon^j(P) \perp \left(X^k\right)_{k\in \mathrm{Pa}(j)}$. Moreover, for any $k \in \mathrm{Pa}(j)$, X^k can be written as a linear combination of $\left(X^{k'}\right)_{k'\in \mathrm{Pa}(k)}$, as ε^k. We thus deduce that ε^j is independent to everything used before. This implies that all error terms are independent.

As a consequence, the covariance matrix $\Omega_0(P)$ of $\varepsilon(P)$ is diagonal. □

To simplify the theoretical results and proofs, until the end of this work, we assume that the noise variances σ^2 in (IV.3) are equal to 1. Since the covariance matrix $\Omega_0(P)$ of $\varepsilon(P)$ is diagonal, we denote $(\omega_j^2(P))_{1\leq j\leq p}$ their coefficients. When $P := \hat{P}$, we denote $\hat{G}_0 := G_0(\hat{P})$, $\hat{\varepsilon} := \varepsilon(\hat{P})$, $\hat{\Omega}_0 := \Omega_0(\hat{P})$ and $\hat{\omega}_j := \omega_j(\hat{P})$.

An interesting point is certainly the link between the error ε^j associated to variable X^j and the error $\varepsilon^j(P)$. Since the covariance matrix Σ_0 of X satisfies:

$$\Sigma_0 = {}^t(I-G_0)^{-1}(I-G_0)^{-1} = {}^t(I-G_0(P))^{-1}\Omega_0(P)(I-G_0(P))^{-1}, \tag{IV.9}$$

we thus have:

$$\begin{aligned} \|\varepsilon\|_F^2 = \|X(I-G_0)\|_F^2 &= \mathrm{trace}\left(X(I-G_0)^t(X(I-G_0))\right) \\ &= \mathrm{trace}\left(X(I-G_0(P))\Omega_0(P)^{-1\,t}(I-G_0(P))^tX\right). \end{aligned}$$

Then, with $\Omega_0(P) = \mathrm{diag}(\omega_1^2(P), ..., \omega_p^2(P))$, we deduce:

$$\begin{aligned} \|\varepsilon\|_F^2 &= \sum_{i,j}\left((X(I-G_0(P)))_i^j\right)^2\frac{1}{\omega_j^2(P)} \\ &= \sum_{j=1}^p \frac{\sum_{k=1}^n\left(\varepsilon_k^j(P)\right)^2}{\omega_j^2(P)}. \end{aligned} \tag{IV.10}$$

Another important remark is given by Lemma 4.2 below.

Lemma 4.2. *Let $P \in \mathbb{P}_p(\mathbb{R})$ a permutation matrix. Then,*

$$\det(\Sigma_0) = \det(\Omega_0(P)) = 1.$$

Proof. From Equation (IV.9), we deduce that:

$$\begin{aligned}\det(\Sigma_0) &= \det\left({}^t(I-G_0)^{-1}(I-G_0)^{-1}\right)\\ &= \det\left({}^t(I-G_0(P))^{-1}\Omega_0(P)(I-G_0(P))^{-1}\right). \qquad \text{(IV.11)}\end{aligned}$$

Since $G_0 = P_0 T_0 {}^t P_0$ with $P_0 \in \mathbb{P}_p(\mathbb{R})$ and $T_0 \in \mathbb{T}_p(\mathbb{R})$ using Proposition 3.2, we have:

$$(I-G_0)^{-1} = \left(I - P_0 T_0 {}^t P_0\right)^{-1} = P_0 (I-T_0)^{-1} {}^t P_0,$$

and

$$\begin{aligned}\det\left((I-G_0)^{-1}\right) &= \det\left((I-T_0)^{-1}\right)\\ &= \left(\det(I-T_0)\right)^{-1} = 1,\end{aligned}$$

using $T_0 \in \mathbb{T}_p(\mathbb{R})$. The same conclusion holds for $\det\left((I-G_0(P))^{-1}\right)$. This ends the proof with Equation (IV.11). □

4.2 Assumptions on the model

We now introduce the assumptions we used to obtain statistical properties of the estimator we consider. The first assumption deals with the covariance matrix Σ_0 of the design matrix X.

Hypothesis $\mathbf{H_{cov}}$ There exists σ_0^2, independent of p and n, such that:

$$\max_{1\le j\le p} (\Sigma_0)_j^j \le \sigma_0^2.$$

Assumption $\mathbf{H_{cov}}$ is needed to obtain bounds for the variances of $\varepsilon(P)$, uniformly over the set of permutations:

$$\forall j \in [\![1,p]\!], \quad |\omega_j(P)|^2 \le \sigma_0^2.$$

Indeed, for node j, the model gives:

$$X^j = \sum_{k\in \mathrm{Pa}(j)} (G_0(P))_k^j X^k + \varepsilon^j(P),$$

where $\varepsilon^j(P) \perp X^{\mathrm{Pa}(j)}$ with Lemma 3.1 since $\mathrm{Pa}(j) \subset \mathrm{ND}(j)$. Then, by independence, we deduce that:

$$\mathrm{Var}(\varepsilon^j(P)) = |\omega_j(P)|^2 \le \mathrm{Var}(X^j) \le \sigma_0^2.$$

From a numerical point of view, this assumption is clearly non-restrictive since we standardize the data. Assumption $\mathbf{H_{cov}}$ is trivially satisfied with $\sigma_0^2 = 1$.

For a matrix $M \in \mathcal{M}_{p,p}(\mathbb{R})$ and a subset $\mathcal{S}$ of $[\![1,p]\!]^2$, we denote by $M_{\mathcal{S}}$ the matrix in $\mathcal{M}_{p,p}(\mathbb{R})$ that has the same coordinates as M on $\mathcal{S}$ and zero coordinates on the complementary set $\mathcal{S}^C$ of $\mathcal{S}$.

Hypothesis $\mathbf{H_{RE}(s)}$ For some integer s such that $1 \leq s \leq p^2$, the following condition holds:

$$\kappa(s) := \min_{\substack{\mathcal{S} \subset [\![1,p]\!]^2 \\ |\mathcal{S}| \leq s}} \min_{\substack{M \neq 0 \\ \|M_{\mathcal{S}^C}\|_1 \leq 3\|M_{\mathcal{S}}\|_1}} \frac{\|XM\|_F}{\sqrt{n}\,\|M_{\mathcal{S}}\|_F} > 0.$$

Assumption $\mathbf{H_{RE}(s)}$ is very similar to the restricted eigenvalue condition of [BRT09]. Under the sparsity scenario, when the number of observations p is large, the Gram matrix $\frac{X^t X}{n}$ is degenerate:

$$\min_{\delta \in \mathbb{R}^p, \delta \neq 0} \frac{\|X\delta\|_2}{\sqrt{n}\,\|\delta\|_2} = 0. \tag{IV.12}$$

Note that the least squared method of estimation then does not work in this case, since it requires positive definiteness of the Gram matrix. To derive some statistical properties of the Lasso and the Dantzig selector, [BRT09] introduces the set of vectors:

$$\{\delta \in \mathbb{R}^p, \|\delta_{\mathcal{S}^C}\|_1 \leq c_0 \|\delta_{\mathcal{S}}\|_1\}, \tag{IV.13}$$

where $\mathcal{S} \subset [\![1,p]\!]$. With probability close to 1, the residual vector of the Lasso satisfies the constraint (IV.13) and we thus require a "restricted" positive definiteness of the Gram matrix: we assume that Equation (IV.12) is valid for vectors satisfying Equation (IV.13).

In our matricial context, one can show that the residual matrix $M := \hat{G} - G_0$ satisfies a similar inequality with large probability:

$$\|M_{\mathcal{S}^C}\|_1 \leq 3\|M_{\mathcal{S}}\|_1,$$

and we thus define a restricted positive definite assumption.

Assumption $\mathbf{H_s}$ below deals with the structure of the DAG $\mathcal{G}_0$ that generates the data and is composed of two parts. The first one ensures that the maximal weight of the edges is bounded. The second one is certainly the most restrictive assumption but plays an important role in the proofs of the theoretical results. It consists in assuming that the graph is sparse enough, with regards to the dimension p and the sample size n.

Hypotheses $\mathbf{H_s}$

$\mathbf{H_{s-1}}$ The maximal value of the adjacency matrix G_0 of the graph $\mathcal{G}_0$ is bounded:

$$\|G_0\|_\infty := \max_{1 \leq i,j \leq p} \left|(G_0)_i^j\right| < +\infty.$$

$\mathbf{H_{s-2}}$ Denote $s_{0,j}$ the number of parents of node j in the graph $\mathcal{G}_0$. Then, the maximal number of parents in the graph, denoted s_{max}, satisfies:

$$s_{max} := \max_{1 \leq j \leq p} s_{0,j} \leq C\sqrt{\frac{n}{\log p}}p^{-3/2},$$

where C is a constant depending on $\|G_0\|_\infty$ and σ_0.

Assumption $\mathbf{H_s}$ is in particular needed to show that the minimal eigenvalue of Σ_0, denoted λ_{min} thereafter, is not too small (see for instance the proof of Lemma 4.5). Let $\chi_{\Sigma_0}(\lambda)$ be the characteristic polynomial of Σ_0 and denote by $(\lambda_1, ..., \lambda_p)$ its p eigenvalues. Since Σ_0 is symmetric definite positive, for all $i \in [\![1,p]\!]$, λ_i is non-negative and Lemma 4.2 implies:

$$\det(\Sigma_0) = \prod_{i=1}^{p} \lambda_i = 1. \tag{IV.14}$$

On the one hand, $\chi_{\Sigma_0}(\lambda) = \prod_{i=1}^{p}(\lambda - \lambda_i)$, and the derivative $\chi'_{\Sigma_0}(\lambda)$ taken for $\lambda = 0$ gives:

$$\chi'_{\Sigma_0}(\lambda) = (-1)^{p-1} \sum_{i=1}^{p} \prod_{j \neq i} \lambda_i.$$

A minor bound for $\left|\chi'_{\Sigma_0}(\lambda)\right|$ is then $\left|\prod_{j\neq i} \lambda_i\right|$ for a given $i \in [\![1, p]\!]$. In particular, considering the index i that corresponds to the smallest eigenvalue and using (IV.14) with $\lambda_i \geq 0$, for all i, we obtain:

$$\left|\chi'_{\Sigma_0}(\lambda)\right| \geq \left| \prod_{\lambda_i \neq \lambda_{min}} \lambda_i \right| = \frac{1}{\lambda_{min}}. \tag{IV.15}$$

On the other hand, for a given matrix $M \in \mathcal{M}_{p,p}(\mathbb{R})$, the derivative of the characteristic polynomial χ_M of M [PP12] is given by:

$$\chi'_M(\lambda) = -\det(M - \lambda I_p) \operatorname{trace}\left((M - \lambda I_p)^{-1}\right).$$

From Lemma 4.2, we thus have:

$$\left|\chi'_{\Sigma_0}(0)\right| = \operatorname{trace}(\Sigma_0^{-1}) = \|I_p - G_0\|_F^2 .$$

Since G_0 is compatible with a DAG, the diagonal of G_0 is null, and we have:

$$\begin{aligned} \chi_{\Sigma_0}(0)' = \|I_p - G_0\|_F^2 &= \sum_{i=1}^{p} 1 + \sum_{i=1}^{p} \sum_{\substack{j=1 \\ j<i}}^{p} \left((G_0)_i^j\right)^2 \\ &\leq p\left(1 + s_{max} \|G_0\|_\infty^2\right) \\ &\leq p \max\left(1, \|G_0\|_\infty^2\right)(1 + s_{max}), \end{aligned}$$

which directly implies from Equation (IV.15) the following bound for λ_{min}:

$$\lambda_{min} \geq \frac{1}{p \max\left(1, \|G_0\|_\infty^2\right)(1 + s_{max})}.$$

This point is not detailed in the works of [vdGB13] even if such an assumption is clearly needed to obtain Theorem 7.3 of [vdGB13].

Remark that Assumption $\mathbf{H_s}$ requires a careful balance between the number of variables p and the sample size n. If n is too small with regards to p, a large p implies quite a restrictive $\mathbf{H_s}$. Following the works of [vdGB13], a necessary condition to obtain bounds in prediction for the ℓ_0-penalized likelihood estimator is to ensure that $p \log p = \mathcal{O}(n)$. However, assuming $n = p \log p$ implies that:

$$s_{max} \leq \frac{C}{p},$$

in $\mathbf{H_{s-2}}$, which is clearly too restrictive when $p \to +\infty$. To overcome this difficulty, a solution could be to reinforce the dimension assumption to obtain a relaxed condition on s_{max}.

Hypothesis $\mathbf{H_{dim}}$ The number of predictors p satisfies:

$$p^3 \log p = \mathcal{O}(n).$$

Assumption $\mathbf{H_{dim}}$ strongly bounds the high-dimensional setting and states that $p^3 \log p$ should be, at the most, on the same order as n. The considered problem is obviously non-trivial and requires a sufficient amount of information.

As a consequence of $\mathbf{H_{dim}}$, an equivalent condition for $\mathbf{H_{s-2}}$ is:

$$s_{max} \leq C.$$

The last assumption is an identifiability condition needed to ensure that the estimated permutation $\hat{P}$ is in Π_0. Given two probability distributions P and Q, denote $\mathrm{D}_{KL}(P||Q)$ the Kullback-Leibler divergence of Q from P, defined as:

$$\mathrm{D}_{KL}(P||Q) = \int \log\left(\frac{dP}{dQ}\right) dP.$$

Hypothesis $\mathbf{H_{id}}$ There exists a constant $\eta > 0$, such that

$$\eta \leq C\frac{n}{p \log p},$$

where $C > 0$, such that, for all permutations $P \notin \Pi_0$,

$$\frac{\mathrm{D}_{KL}\left(\mathcal{N}(0, \Omega_0(P))||\mathcal{N}(0, I_p)\right)}{p} \geq \frac{1}{2\sqrt{\eta}}.$$

In probability theory and information theory, the Kullback-Leibler divergence (also called information divergence) is a non-symmetric measure of the disparity between two probability distributions. Assumption $\mathbf{H_{id}}$ means that the information lost when $\varepsilon(P)$, with $P \notin \Pi_0$, is used to approximate ε is large enough. Remark that the Kullback-Leibler divergence for two multivariate normal distributions $\mathcal{N}_p(\mu_1, \Sigma_1)$ and $\mathcal{N}_p(\mu_2, \Sigma_2)$ is given by:

$$\begin{aligned}&\mathrm{D}_{KL}\left(\mathcal{N}(\mu_1, \Sigma_1)||\mathcal{N}(\mu_2, \Sigma_2)\right)\\ &\qquad = \frac{1}{2}\left(\mathrm{trace}\left(\Sigma_2^{-1}\Sigma_1\right) + {}^t(\mu_2 - \mu_1)\Sigma_2^{-1}(\mu_2 - \mu_1) - p - \log\left(\frac{\det\Sigma_1}{\det\Sigma_2}\right)\right). \quad \text{(IV.16)}\end{aligned}$$

We thus deduce that:

$$\mathrm{D}_{KL}\left(\mathcal{N}(0, \Omega_0(P))||\mathcal{N}(0, I_p)\right) = \frac{1}{2}\left(\mathrm{trace}\left(\Omega_0(P)\right) - p - \log\left(\frac{1}{\det\Omega_0(P)}\right)\right).$$

Remind that $\Omega_0(P)$ is diagonal, and its diagonal elements are $\omega_j^2(P)$. Moreover, from Lemma (4.2), we have $\det\Omega_0(P) = 1$. Assumption $\mathbf{H_{id}}$ then implies:

$$\frac{1}{2p}\sum_{j=1}^{p}\left(\omega_j(P)^2 - 1\right) \geq \frac{1}{2\sqrt{\eta}}.$$

Cauchy-Schwarz inequality then yields the following inequality:

$$\sum_{j=1}^{p}\left(\omega_j(P)^2 - 1\right)^2 \geq \frac{1}{\eta}. \quad \text{(IV.17)}$$

The "omega-min" condition, introduced by [vdGB13], is traduced here more naturally using the KL divergence and is a separation hypothesis. From Assumption $\mathbf{H_{dim}}$, a sufficient condition for $\mathbf{H_{id}}$ is to assume that:

$$\frac{\mathrm{D}_{KL}\left(\mathcal{N}(0, \Omega_0(P))||\mathcal{N}(0, I_p)\right)}{p} \geq \frac{1}{2p}.$$

4.3 Inequality in prediction and estimation

4.3.1 The main result

Theorem 4.1. *Assume that Assumptions* $\mathbf{H_{cov}}$, $\mathbf{H_{RE}}(s)$, $\mathbf{H_s}$, $\mathbf{H_{dim}}$ *and* $\mathbf{H_{id}}$ *are satisfied. Consider the estimator defined by Equation (IV.7), with*

$$\lambda = 2C\sqrt{\frac{\log p}{n} s_{max}}.$$

Then, with probability at least $1-\frac{5}{p}$, *we have* $\hat{P} \in \Pi_0$. *Moreover, with at least the same probability, the following inequalities hold:*

$$\frac{1}{n}\left\|X\hat{G} - XG_0\right\|_F^2 \leq \frac{16C^2 s_{max}^2 \log p}{n\kappa^2(s,3)}. \tag{IV.18}$$

$$\left\|\hat{G} - G_0\right\|_1 \leq \frac{16C}{\kappa^2(s,3)}\sqrt{\frac{\log p}{n}} s_{max}^{3/2}. \tag{IV.19}$$

Remark that if we don't assume some hypotheses on s_{max}, s_{max} can grow with p at the order of p. Then, the penalty $\lambda = 2C\sqrt{\frac{\log p}{n} s_{max}}$ is allowed not to be small and the two inequalities (IV.18) and (IV.19) don't bring rate for convergence for prediction and estimation. Taking s_{max} as in Assumption $\mathbf{H_s}$ let us ensure that λs_{max} is not too large (see the proof below for more explanations). As a consequence, the worst prediction and estimation rate of convergence we obtain from Equation (IV.18) corresponds to the case when $s_{max} = \sqrt{\frac{n}{\log p}} p^{-3/2}$:

$$\begin{cases} \frac{1}{n}\left\|X\hat{G} - XG_0\right\|_F^2 \leq \frac{16C^2}{\kappa^2(s,3)} p^{-3}, \\ \left\|\hat{G} - G_0\right\|_1 \leq \frac{16C}{\kappa^2(s,3)} \left(\frac{n}{p^9 \log p}\right)^{1/4}. \end{cases}$$

4.3.2 Proof of Theorem 4.1

To prove the first point of Theorem 4.1, we assume that the permutation we estimate is a wrong permutation: $\hat{P} \notin \Pi_0$. We aim at obtaining a contradiction of Assumption $\mathbf{H_{id}}$.

Using the definition of $\hat{G}$, given in Equation (IV.7), one has:

$$\begin{aligned} \frac{1}{n}\left\|X - X\hat{G}\right\|_F^2 + \lambda\left\|\hat{G}\right\|_1 &\leq \frac{1}{n}\left\|X - XG_0\right\|_F^2 + \lambda\left\|G_0\right\|_1 \\ &\leq \frac{1}{n}\left\|\varepsilon\right\|_F^2 + \lambda\left\|G_0\right\|_1. \end{aligned} \tag{IV.20}$$

From Equations (IV.20) and (IV.10), we then obtain:

$$\frac{1}{n}\left\|X - X\hat{G}\right\|_F^2 + \lambda\left\|\hat{G}\right\|_1 \leq \frac{1}{n}\sum_{j=1}^{p} \frac{\sum_{k=1}^{n}\left(\hat{\varepsilon}_k^j\right)^2}{\left|\hat{\omega}_j\right|^2} + \lambda\left\|G_0\right\|_1. \tag{IV.21}$$

On another hand, we can bound the difference between $X\hat{G}$ and $X\hat{G}_0$ using some triangular inequalities:

$$\begin{aligned} \frac{1}{n}\left\|X\hat{G} - X\hat{G}_0\right\|_F^2 &= \frac{1}{n}\left(\left\|X\hat{G}\right\|_F^2 + \left\|X\hat{G}_0\right\|_F^2 - 2\langle X\hat{G}, X\hat{G}_0\rangle_F\right) \\ &\leq \frac{1}{n}\left(\left\|X - X\hat{G}\right\|_F^2 - \left\|X\right\|_F^2 + 2\langle X, X\hat{G}\rangle_F\right) + \frac{1}{n}\left\|X\hat{G}_0\right\|_F^2 \\ &\quad -\frac{2}{n}\langle X\hat{G}, X\hat{G}_0\rangle_F. \end{aligned}$$

From Equation (IV.21), we then have:

$$\begin{aligned}
\frac{1}{n}\left\|X\hat{G}-X\hat{G}_0\right\|_F^2+\lambda\left\|\hat{G}\right\|_1 &\leq \frac{1}{n}\sum_{j=1}^{p}\frac{\sum_{k=1}^{n}\left(\hat{\varepsilon}_k^j\right)^2}{\left|\hat{\omega}_j\right|^2}+\lambda\left\|G_0\right\|_1-\frac{1}{n}\left\|X\right\|_F^2\\
&\quad+\frac{2}{n}\langle X-X\hat{G}_0,X\hat{G}\rangle_F+\frac{1}{n}\left\|X\hat{G}_0\right\|_F^2\\
&\leq \frac{1}{n}\sum_{j=1}^{p}\frac{\sum_{k=1}^{n}\left(\hat{\varepsilon}_k^j\right)^2}{\left|\hat{\omega}_j\right|^2}+\lambda\left\|G_0\right\|_1+\frac{2}{n}\langle X-X\hat{G}_0,X\hat{G}\rangle_F\\
&\quad-\frac{1}{n}\left(\left\|X-X\hat{G}_0\right\|_F^2+\left\|X\hat{G}_0\right\|_F^2+2\langle X-X\hat{G}_0,X\hat{G}_0\rangle_F\right)\\
&\quad+\frac{1}{n}\left\|X\hat{G}_0\right\|_F^2\\
&\leq \frac{1}{n}\sum_{j=1}^{p}\frac{\sum_{k=1}^{n}\left(\hat{\varepsilon}_k^j\right)^2}{\left|\hat{\omega}_j\right|^2}+\frac{2}{n}\langle\hat{\varepsilon},X\hat{G}-X\hat{G}_0\rangle_F-\frac{1}{n}\left\|\hat{\varepsilon}\right\|_F^2+\lambda\left\|G_0\right\|_1\\
&\leq \frac{1}{n}\sum_{j=1}^{p}\left(\frac{1}{\left|\hat{\omega}_j\right|^2}-1\right)\sum_{k=1}^{n}\left(\hat{\varepsilon}_k^j\right)^2+\frac{2}{n}\langle\hat{\varepsilon},X\hat{G}-X\hat{G}_0\rangle_F+\lambda\left\|G_0\right\|_1,
\end{aligned}$$

and we finally obtain:

$$\begin{aligned}
\frac{1}{n}\left\|X\hat{G}-X\hat{G}_0\right\|_F^2+\lambda\left\|\hat{G}\right\|_1 &\leq \sum_{j=1}^{p}\frac{\left|\hat{\omega}_j\right|^2-\frac{1}{n}\sum_{k=1}^{n}\left(\hat{\varepsilon}_k^j\right)^2}{\left|\hat{\omega}_j\right|^2}\left(\left|\hat{\omega}_j\right|^2-1\right)+\sum_{j=1}^{p}\left(1-\left|\hat{\omega}_j\right|^2\right)\\
&\quad+\frac{2}{n}\langle\hat{\varepsilon},X\hat{G}-X\hat{G}_0\rangle_F+\lambda\left\|G_0\right\|_1\\
&\leq \underbrace{\sqrt{\sum_{j=1}^{p}\left(\frac{\left|\hat{\omega}_j\right|^2-\frac{1}{n}\sum_{k=1}^{n}\left(\hat{\varepsilon}_k^j\right)^2}{\left|\hat{\omega}_j\right|^2}\right)^2}}_{=I}\sqrt{\sum_{j=1}^{p}\left(\left|\hat{\omega}_j\right|^2-1\right)^2}\\
&\quad+\underbrace{\sum_{j=1}^{p}\left(1-\left|\hat{\omega}_j\right|^2\right)}_{=II}\\
&\quad+\frac{2}{n}\underbrace{\langle\hat{\varepsilon},X\hat{G}-X\hat{G}_0\rangle_F}_{=III}+\lambda\left\|G_0\right\|_1. \qquad \text{(IV.22)}
\end{aligned}$$

Then, Lemmas 4.3 and 4.4 below aim at bounding the terms I and III, with large probability.

Lemma 4.3. *Assume that Assumption* $\mathbf{H_{dim}}$ *is satisfied. Then, with probability at least* $1-\frac{2}{p}$, *there exists* $C>0$ *such that we have:*

$$\sum_{j=1}^{p}\left(\frac{\left|\hat{\omega}_j\right|^2-\frac{1}{n}\sum_{k=1}^{n}\left(\hat{\varepsilon}_k^j\right)^2}{\left|\hat{\omega}_j\right|^2}\right)^2\leq C\frac{\log p}{n}\left(p+\hat{s}_0\right),$$

where $\hat{s}_0:=\left\|\hat{G}_0\right\|_0$ *is the number of non-zero coefficients of* $\hat{G}_0$.

Proof of Lemma 4.3. The proof of this result is given in [vdGB13]. For a better understanding, we recall key elements of the proof here. Denote by:

$$Z_j(P) := \frac{\frac{1}{n}\sum_{k=1}^n \left(\varepsilon_k^j(P)\right)^2 - |\omega_j(P)|^2}{|\omega_j(P)|^2},$$

and assume that $G_0(P)$, connected with $\varepsilon(P)$ by $\varepsilon(P) = X - XG_0(P)$ is $s_0(P)$-sparse. Using Bernstein-like concentration inequalities, we can show that:

$$\mathbb{P}\left(\exists P, \sum_{j=1}^p Z_j(P)^2 \geq 8\left(\frac{pt + (1+8\alpha)s_0(P)\log(p) + 2p\log p}{n}\right) + 8\left(\frac{4p(t^2+\log^2 p)}{n^2}\right)\right) \leq 2e^{-t},$$

for all $t \geq 0$, where α is some constant such that $p^4 \leq \alpha n$. Then, the conclusion holds, with $P = \hat{P}$ and $t = \log p$. □

Lemma 4.4. *Assume that Assumptions* $\mathbf{H_{cov}}$ *and* $\mathbf{H_{dim}}$ *are satisfied. Then, with probability at least* $1 - \frac{1}{p}$*, there exists* $C > 0$ *such that:*

$$\frac{2}{n}\langle \hat{\varepsilon}, X\hat{G} - X\hat{G}_0\rangle_F \leq C\sqrt{\frac{\log p}{n}} \max_j \sqrt{\hat{s}_{0,j}} \left\|\hat{G} - \hat{G}_0\right\|_1,$$

where $\hat{s}_{0,j}$ *is the notation for the sparsity of the vector* $\hat{G}_0^j$*.*

Proof of Lemma 4.4. Remark that:

$$\langle \hat{\varepsilon}, X\hat{G} - X\hat{G}_0\rangle_F = \sum_{i,j}\left(\hat{G} - \hat{G}_0\right)_i^j \sum_k X_k^i \hat{\varepsilon}_k^j.$$

To obtain this result, we aim at showing that, uniformly over the set of permutation matrices P and uniformly on $1 \leq i \leq p$, $\sum_k X_k^i \varepsilon_k^j(P)$ is bounded.

Let $(V_k)_{k=1,..,n}$ *i.i.d* random variables generated according to a $\mathcal{N}(0,1)$ distribution. A standard concentration inequality gives then:

$$\mathbb{P}\left(\frac{1}{n}\sum_{k=1}^n V_k \geq t\right) \leq \exp(-nt^2).$$

Let $P \in \mathbb{P}_p(\mathbb{R})$. Denote by $\tilde{\mathcal{G}}_i(P) = \{\beta \in \mathbb{R}^p, \forall \beta_j = 0, X^i \perp \varepsilon^j(P),$ with $\varepsilon^j(P) = X^j - \sum_k X^k\beta_k\}$, in such a way that $G_0^j(P) \in \tilde{G}_i(P)$. Then, let $A^i(P)$ the set defined as:

$$A^i(P) = \left\{\exists \beta \in \tilde{\mathcal{G}}_i(P), \ \frac{2}{n}\sum_k X_k^i \varepsilon_k^j(P) \geq 2\sigma_0 \frac{\sqrt{t + s_i(P)\log p + 2\log p}}{n}, \ \text{with } \varepsilon^j(P) = X^j - \textstyle\sum_k X^k\beta_k\right\},$$

where $s_i(P)$ is the sparsity of $\beta \in \tilde{\mathcal{G}}_i(P)$.

Under Assumption $\mathbf{H_{cov}}$, the random variables $\varepsilon^j(P)$ follow a $\mathcal{N}(0, \omega_j^2(P))$, where $\left|\omega_j^2(P)\right| \leq \sigma_0^2$. We thus deduce that:

$$\mathbb{P}(A^i(P)) \leq \exp\left(-\left(t + s_i(P)\log(p) + 2\log p\right)\right).$$

Let $m \in [\![1,p]\!]$. We now let P vary over all permutations such that $s_i(P) = m$, and we denote Π_m this set. On Π_m, node j has exactly m parents, and there exists at most $\binom{p}{m}$ possibilities for P. We then have:

$$\begin{aligned}\mathbb{P}\left(\bigcup_{P\in\Pi_m} A^i(P)\right) &\leq \binom{p}{m}\exp\left(-\left(t+s_i(P)\log(p)+2\log p\right)\right)\\ &\leq \exp\left(-\left(t+2\log p\right)\right).\end{aligned}$$

For m and i varying as possible, we conclude that:

$$\mathbb{P}\left(\bigcup_{i\in[\![1,p]\!]}\bigcup_{P\in\Pi} A^i(P)\right) \leq \exp\left(-t\right).$$

Then, with probability at least $1-e^{-t}$,

$$\begin{aligned}\langle\hat{\varepsilon}, X\hat{G}-X\hat{G}_0\rangle_F &\leq \sum_{i,j} 2\sigma_0\sqrt{\frac{t+\hat{s}_{0,j}(P)\log p+2\log p}{n}}\left(\hat{G}-\hat{G}_0\right)_i^j\\ &\leq 2\sigma_0\max_j\sqrt{\frac{t+\hat{s}_{0,j}\log p+2\log p}{n}}\left\|\hat{G}-\hat{G}_0\right\|_1,\end{aligned}$$

which ends the proof with $t=\log p$. □

The last term II of Equation (IV.22) is bounded using inequality $\ln(1+x)\leq x-\frac{1}{2(1+c_0^2)}x^2$, satisfied for $-1\leq x\leq c_0$, to $x=|\hat{\omega}_j|^2-1$, which satisfies $-1\leq x\leq \sigma_0^2-1$ (Hypotheses $\mathbf{H_{cov}}$):

$$\sum_{j=1}^p \ln\left(|\hat{\omega}_j|^2\right) \leq \sum_{j=1}^p\left(|\hat{\omega}_j|^2-1\right)-\frac{1}{2\sigma_0^4}\sum_{j=1}^p\left(|\hat{\omega}_j|^2-1\right)^2.$$

Moreover, using lemma (4.2) yields $\det(\Sigma_0)=\prod\hat{\omega}_j^2=1$. We thus deduce that $\sum_j\ln\left(|\hat{\omega}_j|^2\right)=0$ and we finally obtain:

$$II \leq -\frac{1}{2\sigma_0^4}\sum_{j=1}^p\left(|\hat{\omega}_j|^2-1\right)^2. \tag{IV.23}$$

From Lemmas 4.3 and 4.4 and Equation (IV.23), the following inequality is deduced from Equation (IV.22), with probability at least $1-\frac{3}{p}$:

$$\begin{aligned}\frac{1}{n}\left\|X\hat{G}-X\hat{G}_0\right\|_F^2+\lambda\left\|\hat{G}\right\|_1 \leq C\sqrt{\frac{\log p}{n}}\sqrt{p+\hat{s}_0}\sqrt{\sum_{j=1}^p\left(|\hat{\omega}_j|^2-1\right)^2-\frac{1}{2\sigma_0^4}\sum_{j=1}^p\left(|\hat{\omega}_j|^2-1\right)^2}\\ +C\sqrt{\frac{\log p}{n}}\max_j\sqrt{\hat{s}_{0,j}}\left\|\hat{G}-\hat{G}_0\right\|_1+\lambda\left\|G_0\right\|_1.\end{aligned}$$

Let $\delta>0$ such that $\delta\leq\frac{1}{2\sigma_0^4}$, using $2xy\leq\frac{x^2}{a}+ay^2$ with $a=2\delta$, we can show with probability at least $1-\frac{3}{p}$ that:

$$\begin{aligned}\frac{1}{n}\left\|X\hat{G}-X\hat{G}_0\right\|_F^2+\lambda\left\|\hat{G}\right\|_1 \leq \frac{C\log p}{4\delta n}(p+\hat{s}_0)+\left(\delta-\frac{1}{2\sigma_0^4}\right)\sum_{j=1}^p\left(|\hat{\omega}_j|^2-1\right)^2\\ +C\sqrt{\frac{\log p}{n}}\max_j\sqrt{\hat{s}_{0,j}}\left\|\hat{G}-\hat{G}_0\right\|_1+\lambda\left\|G_0\right\|_1. \quad \text{(IV.24)}\end{aligned}$$

Lemma 4.5. *Assume that Assumptions* $\mathbf{H_{cov}}$, $\mathbf{H_{dim}}$ *and* $\mathbf{H_s}$ *hold. Then, with probability at least* $1-\frac{2}{p}$*:*

$$\frac{1}{n}\left\|X\left(\hat{G}-\hat{G}_0\right)\right\|_F^2 \geq \left(\frac{3\lambda_{min}}{4}-2\sqrt{\frac{\log p}{n}}-3\sigma_0\sqrt{\frac{2p\log p}{n}}\right)^2\left\|\hat{G}-\hat{G}_0\right\|_F^2.$$

Proof of Lemma 4.5. This result is a consequence of Theorem 7.3 of [vdGB13]: for all $t>0$, with probability at least $1-2e^{-t}$, we have:

$$\frac{1}{n}\|X\beta\|_2 \geq \left(\frac{3\lambda_{min}}{4}-\sqrt{\frac{2(t+\log p)}{n}}-3\sigma_0\sqrt{\frac{s_\beta\log p}{n}}\right)\|\beta\|_2,$$

uniformly on $\beta\in\mathbb{R}^p$, where s_β is the number of non-zero coefficients of β.

Moreover,

$$\frac{1}{n}\left\|X\left(\hat{G}-\hat{G}_0\right)\right\|_F^2=\frac{1}{n}\sum_j\left\|X\left(\hat{G}-\hat{G}_0\right)^j\right\|_2^2.$$

Let $\beta=\left(\hat{G}-\hat{G}_0\right)^j$ ($j\in[\![1,p]\!]$), we then deduce, with probability at least $1-2e^{-t}$, that:

$$\frac{1}{n}\left\|X\left(\hat{G}-\hat{G}_0\right)^j\right\|_2 \geq \left(\frac{3\lambda_{min}}{4}-\sqrt{\frac{2(t+\log p)}{n}}-3\sigma_0\sqrt{\frac{s_{\left(\hat{G}-\hat{G}_0\right)^j}\log p}{n}}\right)\left\|\left(\hat{G}-\hat{G}_0\right)^j\right\|_2,$$

where the quantity between brackets is non-negative by Assumption $\mathbf{H_s}$. The conclusion holds using $s_{\left(\hat{G}-\hat{G}_0\right)^j}\leq 2p$ and setting $t=\log p$. □

From lemma 4.5 and Equation (IV.24), we deduce that:

$$\left(\frac{3\lambda_{min}}{4}-2\sqrt{\frac{\log p}{n}}-3\sigma_0\sqrt{\frac{2p\log p}{n}}\right)^2\left\|\hat{G}-\hat{G}_0\right\|_F^2-C\sqrt{\frac{\log p}{n}}\max_j\sqrt{\hat{s}_{0,j}}\left\|\hat{G}-\hat{G}_0\right\|_1$$
$$+\left(\frac{1}{2\sigma_0^4}-\delta\right)\sum_{j=1}^p\left(|\hat{\omega}_j|^2-1\right)^2+\lambda\left\|\hat{G}\right\|_1\leq\frac{C\log p}{4\delta n}(p+\hat{s}_0)+\lambda\|G_0\|_1.$$

For all $j\in[\![1,p]\!]$, we finally use the Cauchy-Schwarz inequality:

$$\left\|\left(\hat{G}-\hat{G}_0\right)^j\right\|_1\leq\sqrt{s_{\left(\hat{G}-\hat{G}_0\right)^j}}\left\|\left(\hat{G}-\hat{G}_0\right)^j\right\|_2\leq\sqrt{2p}\left\|\left(\hat{G}-\hat{G}_0\right)^j\right\|_2,$$

which gives:

$$\max_j\sqrt{\hat{s}_{0,j}}\left\|\hat{G}-\hat{G}_0\right\|_1\leq\sqrt{2}p\left\|\hat{G}-\hat{G}_0\right\|_F.$$

Therefore,

$$\underbrace{\left(\left(\frac{3\lambda_{min}}{4}-2\sqrt{\frac{\log p}{n}}-3\sigma_0\sqrt{\frac{2p\log p}{n}}\right)^2\left\|\hat{G}-\hat{G}_0\right\|_F-Cp\sqrt{\frac{2\log p}{n}}\right)\left\|\hat{G}-\hat{G}_0\right\|_F}_{=A}$$
$$+\left(\frac{1}{2\sigma_0^4}-\delta\right)\sum_{j=1}^p\left(|\hat{\omega}_j|^2-1\right)^2+\lambda\left\|\hat{G}\right\|_1$$
$$\leq\frac{C\log p}{4\delta n}(p+\hat{s}_0)+\lambda\|G_0\|_1\leq\frac{Cp^2\log p}{4\delta n}+\lambda s_{max}p\|G_0\|_\infty, \quad \text{(IV.25)}$$

where we have used $\|G_0\|_1 = \sum_{i,j} |G_0|_i^j \leq \sum_{i=1}^p s_{max} \max_j |G_0|_i^j$.

Lemma 4.6 below gives us a bound for A.

Lemma 4.6.

$$\left(\left(\frac{3\lambda_{min}}{4} - 2\sqrt{\frac{\log p}{n}} - 3\sigma_0\sqrt{\frac{2p\log p}{n}}\right)^2 \left\|\hat{G} - \hat{G}_0\right\|_F - Cp\sqrt{\frac{2\log p}{n}}\right) \left\|\hat{G} - \hat{G}_0\right\|_F \geq -\frac{Cp^2\frac{\log p}{n}}{2\left(\frac{3\lambda_{min}}{4} - 2\sqrt{\frac{\log p}{n}} - 3\sigma_0\sqrt{\frac{2p\log p}{n}}\right)^2}.$$

Proof of Lemma 4.6. A is minimal as soon as $\left\|\hat{G} - \hat{G}_0\right\|_F = \frac{Cp\sqrt{\frac{2\log p}{n}}}{2\left(\frac{3\lambda_{min}}{4} - 2\sqrt{\frac{\log p}{n}} - 3\sigma_0\sqrt{\frac{2p\log p}{n}}\right)^2}$, and equals:

$$\begin{aligned} A &= \left(\frac{Cp\log p}{\sqrt{2}n} - C\sqrt{2}\frac{p\log p}{n}\right) \frac{\frac{Cp\log p}{n}}{\sqrt{2}\left(\frac{3\lambda_{min}}{4} - 2\sqrt{\frac{\log p}{n}} - 3\sigma_0\sqrt{\frac{2p\log p}{n}}\right)^2} \\ &= -\frac{Cp^2\frac{\log p}{n}}{2\left(\frac{3\lambda_{min}}{4} - 2\sqrt{\frac{\log p}{n}} - 3\sigma_0\sqrt{\frac{2p\log p}{n}}\right)^2}. \end{aligned}$$

□

If η given in Assumption $\mathbf{H_{id}}$ satisfies

$$\eta \leq \frac{\frac{1}{2\sigma_0^4} - \delta}{\frac{C\log p}{4\delta n}p^2 + \lambda p s_{max}\|G_0\|_\infty + \frac{Cp^2\log p}{2n\left(\frac{3\lambda_{min}}{4} - 2\sqrt{\frac{\log p}{n}} - 3\sigma_0\sqrt{\frac{2p\log p}{n}}\right)^2}} \cdot p, \tag{IV.26}$$

Equation (IV.25) then contradicts Assumption $\mathbf{H_{id}}$. Setting $\lambda = C\sqrt{\frac{\log p}{n}s_{max}}$ with s_{max} chosen as in Assumption $\mathbf{H_{s-2}}$, η satisfies Equation (IV.26) with probability at least $1 - \frac{5}{p}$. We then deduce that the estimated permutation $\hat{P}$ is a good permutation, *i.e.* $P \in \Pi_0$.

For the second part of the proof, we repeat the same process, with $\hat{P} \in \Pi_0$. As a consequence, $\hat{\omega}_j = 1$, for all j. Equation (IV.24) gives:

$$\frac{1}{n}\left\|X\hat{G} - XG_0\right\|_F^2 + \lambda\left\|\hat{G}\right\|_1 \leq \frac{\lambda}{2}\left\|\hat{G} - G_0\right\|_1 + \lambda\|G_0\|_1. \tag{IV.27}$$

We then have:

$$\frac{\lambda}{2}\left\|\hat{G} - G_0\right\|_1 \leq \lambda\left(\|G_0\|_1 - \left\|\hat{G}\right\|_1 + \left\|\hat{G} - G_0\right\|_1\right),$$

where $\|G_0\|_1 - \left\|\hat{G}\right\|_1 + \left\|\hat{G} - G_0\right\|_1 = \sum_{i,j}\left(\left|(G_0)_i^j\right| - \left|\hat{G}_i^j\right| + \left|\left(\hat{G} - G_0\right)_i^j\right|\right) := \sum_{i,j} M_i^j$, with

$$M_i^j \begin{cases} = 0 & \text{if } (i,j) \notin \mathcal{S}_0 \\ \leq 2\left|\left(\hat{G} - G_0\right)_i^j\right| & \text{otherwise.} \end{cases}$$

We finally obtain:

$$\frac{\lambda}{2}\left\|\hat{G}-G_0\right\|_1 \leq 2\lambda\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_1.$$

Using $\left\|\hat{G}-G_0\right\|_1 = \left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0^C}\right\|_1 + \left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_1$, where $\mathcal{S}_0^C$ is the notation for the complementary set of $\mathcal{S}_0$, the following inequality holds:

$$\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0^C}\right\|_1 \leq 3\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_1. \tag{IV.28}$$

We now apply Assumption $\mathbf{H_{RE}(s)}$ of [BRT09], to the matrix $\hat{G}-G_0$ which satisfies Equation (IV.28):

$$\kappa(s,3)\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_F \leq \frac{1}{\sqrt{n}}\left\|X\hat{G}-XG\right\|_F. \tag{IV.29}$$

From Equation (IV.27) and using the same calculus as previously, we can show that:

$$\frac{1}{n}\left\|X\hat{G}-XG_0\right\|_F^2 + \frac{\lambda}{2}\left\|\hat{G}-G_0\right\|_1 \leq 2\lambda\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_1. \tag{IV.30}$$

The Cauchy-Schwarz inequality now gives:

$$\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_1 \leq \sqrt{s_{max}}\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_F. \tag{IV.31}$$

From Equations (IV.29), (IV.30) and (IV.31), we finally obtain:

$$\frac{1}{n}\left\|X\hat{G}-XG_0\right\|_F^2 + \frac{\lambda}{2}\left\|\hat{G}-G_0\right\|_1 \leq \frac{2\lambda\sqrt{s_{max}}}{\kappa(s,3)\sqrt{n}}\left\|X\hat{G}-XG_0\right\|_F.$$

As a conclusion:

$$\frac{1}{n}\left\|X\hat{G}-XG_0\right\|_F^2 \leq \frac{4\lambda^2 s_{max}}{\kappa^2(s,3)}.$$

The proof of the inequality in prediction follows with the definition of λ.

To obtain an inequality on estimation, remind that:

$$\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0^C}\right\|_1 \leq 3\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_1.$$

We thus have

$$\begin{aligned}\left\|\hat{G}-G_0\right\|_1 &= \left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_1 + \left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0^C}\right\|_1 \\ &\leq 4\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_1 \\ &\leq 4\sqrt{s_{max}}\left\|\left(\hat{G}-G_0\right)_{\mathcal{S}_0}\right\|_F,\end{aligned}$$

with Cauchy-Schwarz inequality. Then, using again $\mathbf{H_{RE}(s)}$ to $\hat{G}-\hat{G}_0$ which satisfies Equation (IV.27):

$$\begin{aligned}\left\|\hat{G}-G_0\right\|_1 &\leq \frac{4\sqrt{s_{max}}}{\kappa(s,3)}\frac{\left\|X\left(\hat{G}-G_0\right)\right\|_F}{\sqrt{n}} \\ &\leq \frac{16C}{\kappa^2(s,3)}\sqrt{\frac{\log p}{n}}s_{max}^{3/2},\end{aligned}$$

where we have used the inequality of prediction (IV.18). This ends the proof.

5 Two optimization computational procedures

This section is dedicated to the computation of the estimator presented in Equation (IV.7). As explained in Section 4, the main difficulties occur when investigating the set of matrices compatible with a DAG, which is hard to parametrize. Using Proposition 3.2, the task of solving (IV.7) is now reduced to finding $(P, T) \in \mathbb{P}_p(\mathbb{R}) \times \mathbb{T}_p(\mathbb{R})$ that minimizes Equation (IV.8). Given a permutation matrix P, the criterion (IV.8) is convex in T and we can easily calculate the corresponding best parameter T by an explicit algorithm. The problem is then reduced to a discrete optimization over the set of permutations, which is non-convex and can still be large.

Section 5.1 is dedicated to reminders on optimization. In Section 5.2, we present a first naive procedure of optimization based on an alternating minimization, with relaxation of the set of permutation matrices. A second procedure, derived from a Genetic Algorithm, is proposed in Section 5.3.

5.1 Reminders on optimization

We begin this section with reminders on optimization, which will be very useful in the sequel. A problem of optimization can be formulated as follows:

$$\min_{x \in E \subset F} f(x), \tag{IV.32}$$

where F is a Banach space, the set of constraints E is a subset of F and the objective function f to minimize is any function from $E \subset F$ to $\mathbb{R}$.

5.1.1 Elements of convex analysis

In this paragraph, we recall some notions derived from convex analysis. For more details, see also [Bre83] and [HUL93]. Consider a Banach space F, with the inner product $\langle ., . \rangle$ and the induced norm $\|.\|$. Let E be a subset of F.

Definition 5.1. *E is said to be convex if:*

$$\forall x, y \in E, \forall \lambda \in [0,1], \quad \lambda x + (1-\lambda) y \in E.$$

E is a convex cone if:

$$\forall x, y \in E, \forall \lambda_1, \lambda_2 \in \mathbb{R}^+, \quad \lambda_1 x + \lambda_2 y \in E.$$

Definition 5.2. *Let $f : E \to \mathbb{R}$ be a function. f is convex if:*

$$\forall \lambda \in [0,1], \forall x, y \in E, \quad f(\lambda x + (1-\lambda) y) \leq \lambda f(x) + (1-\lambda) f(y).$$

We are also particularly interested in the differentiable Lipschitz class of functions, defined as follows:

Definition 5.3. *Let $f : E \to \mathbb{R}$ a differentiable function. f is said to be Lipschitz differentiable if:*

$$\forall x, y \in E, \quad |\nabla f(x) - \nabla f(y)| \leq L \, |x - y| \, ,$$

where $L > 0$ is called the constant of Lipschitz differentiability and $\nabla .$ is the notation for the gradient.

Proposition 5.1 below, introduced by [Pol87], enables to show the convergence of minimization processes for Lipschitz differentiable functions (for more details see Section 5.1.3).

Proposition 5.1. *A function f L-Lipschtiz differentiable satisfies:*

$$\forall x, y \in E, \quad |f(y) - f(x) - \langle \nabla f(x), y - x \rangle| \leq \frac{L}{2} |y - x|^2 .$$

Moreover, if f is convex, the following inequality holds:

$$\forall x, y \in E, \quad f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} |y - x|^2. \tag{IV.33}$$

Proof. Let $f : E \to \mathbb{R}$ a convex and Lipschtiz differentiable function. Then, the Taylor theorem implies

$$\begin{aligned} f(x + d) &= f(x) + \int_0^1 \langle \nabla f(x + td), d \rangle dt \\ &= f(x) + \int_0^1 \langle \nabla f(x + td) - \nabla f(x), d \rangle dt + \langle \nabla f(x), d \rangle. \end{aligned}$$

Moreover,

$$\begin{aligned} |\langle \nabla f(x + td) - \nabla f(x), d \rangle| &\leq \|\nabla f(x + td) - \nabla f(x)\| \, \|d\| \\ &\leq L \|x + td - x\| \, \|d\| , \quad \text{since } f \text{ is Lipschitz differentiable} \\ &\leq Lt \|d\|^2 . \end{aligned}$$

We thus deduce:

$$|f(x + d) - f(x) - \langle \nabla f(x), d \rangle| \leq \int_0^1 Lt \|d\|^2 = \frac{L}{2} \|d\|^2 . \tag{IV.34}$$

Using Equation (IV.34) with $d = y - x$, we finally obtain the first part of Proposition 5.1. Since f is convex, $0 \leq f(y) - f(x) - \langle \nabla f(x), y - x \rangle$, which ends the proof. □

For the non-differentiable functions, we extend the definition of differential through the notion of subdifferential:

Definition 5.4. *Let $f : E \to \mathbb{R}$ a convex function. The subdifferential of f at point $x \in E$ is defined as:*

$$\partial f(x) = \{\eta \in E, f(x) + \langle \eta, y - x \rangle \leq f(y), \forall y \in E\} .$$

$\eta \in \partial f(x)$ is called the subgradient of f.

From a geometrical point of view, the subdifferential of f at point x can be viewed as the set of hyperplans that go through the point $(x, f(x))$ and which graph is under the graph of f. If f is differentiable at x, the subdifferential $\partial f(x)$ is reduced to a unique point.

Example 6. *Let $f(x) = |x|$. Its subdifferential is defined as:*

$$\begin{cases} [-1, 1] & \text{if } x{=}0 \\ \text{sign}(x) & \text{otherwise,} \end{cases}$$

where sign(x) *is the notation for the sign of x.*

Indeed, if $x \neq 0$, f is differentiable at x and its derivative equals ± 1 following the sign of x. Then, assume that $x = 0$, and let $\eta \in \mathbb{R}$ such that, for all $y \in \mathbb{R}$, $f(0) + \langle \eta, y \rangle \leq f(y)$. We then have

— for $y \geq 0$, $\eta y \leq y$ i.e $\eta \leq 1$.

— for $y \leq 0$, $\eta y \leq -y$ i.e $\eta \geq -1$.

We deduce that $\eta \in [-1, 1]$. Reciprocally, let $\eta \in [-1, 1]$ and $y \in \mathbb{R}$:

— if $y \geq 0$, $-y \leq \eta y \leq y$ i.e $\eta y \leq |y|$.

— if $y \leq 0$, $y \leq \eta y \leq -y$ i.e $\eta y \leq |y|$.

As a conclusion, $\forall y \in \mathbb{R}, f(0) + \langle \eta, y \rangle \leq f(y)$.

5.1.2 Projection theorems

In this paragraph, we recall some elements that deal with the projection on a convex space. For more details, see [HUL93].

Theorem 5.1 (Projection on a closed convex space)**.** *Let E a non-empty closed convex subspace of a Banach space F. Then, for all $x \in F$, there exists a unique $\bar{x} \in E$ such that:*

$$\|x - \bar{x}\| = \inf_{y \in E} \|x - y\| .$$

Moreover, $\bar{x}$ satisfies the following property:

$$\begin{cases} \bar{x} \in E, \\ \forall y \in E, \quad \langle x - \bar{x}, y - \bar{x} \rangle \leq 0. \end{cases}$$

$\bar{x}$ is called the projection of x on E, and denoted $\bar{x} = \mathrm{Proj}_E(x)$.

From Theorem 5.1, if we assume that E is a closed convex subspace of F, we then deduce the following characterization of the projection:

Corollary 5.1 (Characterization of the projection on a closed convex subspace)**.** *Let E a non-empty closed convex subspace of F and $x \in F$. Then, the projection point $\bar{x}$ of x on E satisfies:*

$$\begin{cases} \bar{x} \in E, \\ x - \bar{x} \in E^{\perp}, \end{cases}$$

where $E^{\perp}$ is the orthogonal subspace of E.

5.1.3 Convex optimization without constraints

In this paragraph, we restrict our study to convex optimization problem without constraints, where the objective function f to minimize is convex and the set of constraints $E = F$. Optimization problems with convex data form an important class of optimization problems since any local minimum must be a global minimum. The convexity of f then makes its optimization easier. Theorem 5.2 below gives necessary conditions for a point x to be a solution of (IV.32).

Theorem 5.2 (Optimality condition for convex fonctions)**.** *Let f be the objective function in (IV.32) and assume that f is convex.*

1. *If f is differentiable, the two following items are equivalent:*
   — *x belongs to the set of solutions of (IV.32),*
   — $\nabla f(x) = 0$.
2. *If not, the two following items are equivalent:*
   — *x belongs to the set of solutions of (IV.32),*
   — $0 \in \partial f(x)$.

When f is differentiable, a first method of optimization devoted to solve (IV.32) consists in applying descent algorithms. Descent algorithms work as follows: given an arbitrary point x_0, we generate a sequence of point $(x_k)_{k \in \mathbb{N}}$ such that:

$$\forall k \in \mathbb{N}, f(x_{k+1}) \leq f(x_k).$$

For example, the gradient descent algorithm consists in replacing f by its Taylor series, at the neighborhood of x_k:

$$\begin{cases} x^0 \in E, \\ x^{k+1} = x^k - \gamma \nabla f(x^k), \end{cases}$$

where x^0 is any initial point and γ is the fixed descent step. The gradient descent algorithm is one of the easiest and the most popular method to minimize a differentiable function.

Under the assumption that f is L-Lipschitz differentiable, Polyak shows the convergence of this algorithm [Pol87]. However, following the works of Nesterov [Nes04], the rate of convergence is particularly sensitive to the choice of the Lipschitz constant L (linear dependence) and the distance between the initial point x^0 and the minimum (quadratic dependence).

The gradient descent algorithm can be slightly modified, with a locally optimal choice of step size γ on every iteration, to improve the convergence. The gradient descent algorithm with optimal step is defined as follows:

$$\begin{cases} x^0 \in E, \\ x^{k+1} = x^k - \gamma_k \nabla f(x^k), \quad \text{where} \quad \gamma_k = \underset{\gamma>0}{\operatorname{argmin}} \left\{ f\left(x^k - \gamma \nabla f(x^k)\right) \right\}. \end{cases} \tag{IV.35}$$

5.1.4 Optimization under linear constraints

In this paragraph, we consider now optimization problems given by Equation (IV.32), where the set of constraints E is defined by some equalities:

$$E = \{x \in F, h_i(x) = 0, i = 1, ..., p\},$$

where the function $h : F \to \mathbb{R}^p$ is continuous.

When the set of constraints E is convex, the projection of any point $x \in F$ on E is well defined, and the projected descent gradient algorithm, introduced by [Pol87] and defined as:

$$\begin{cases} x^0 \in F, \\ x^{k+1} = \operatorname{Proj}_E\left(x^k - \gamma_k \nabla f(x^k)\right), \quad \text{where} \quad \gamma_k = \underset{\gamma>0}{\operatorname{argmin}} \left\{ f\left(x^k - \gamma \nabla f(x^k)\right) \right\}, \end{cases}$$

generalizes the gradient descent algorithm.

Assume that the set of constraints E is a subset of $\mathbb{R}^n$, defined by *linear* constraints of equality, *i.e.* $(h_i)_{1 \le i \le p}$ are linear. Then, E can be written as a convex polyedron:

$$E = \{x \in \mathbb{R}^n, Ax = b, x \ge 0\},$$

where $A \in \mathcal{M}_{p,n}(\mathbb{R})$ is a $p \times n$ matrix and $b \in \mathbb{R}^p$ is a p vector. If the function f to minimize is also linear, (IV.32) can be solved using the simplex algorithm. The simplex algorithm is a popular algorithm for linear programming. Following the works of [PS82], if (IV.32) has a solution, one of the extreme points of the convex polyedron E, which can be viewed as vertexes of E, is a solution of (IV.32). The simplex algorithm then consists in finding a solution of (IV.32) within the vertices of E, without investigating the whole set of vertices. At each step of the algorithm, we switch from one vertex to another that has a lower score, following an edge (see Figure IV.3).

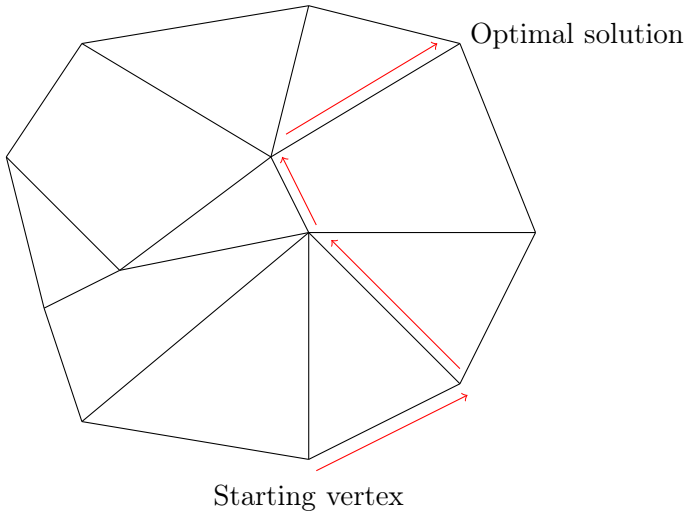


Figure IV.3: Illustration of the simplex algorithm: the algorithm begins at a starting vertex and moves along the edges of the polytope, until it reaches the vertex of the optimal solution.

5.2 A first method of optimization based on alternate minimization

To solve (IV.8), a popular approach consists in using an alternating minimization: iteratively, keep one of the variables (P, T) fixed and optimize over the other, then switch and repeat (for instance, see [CT84]). More precisely, denote $f(P,T) = \frac{1}{n}\left\|X - XPT^tP\right\|_F^2 + \lambda \left\|T\right\|_1$ the objective function to minimize, and consider the two induced optimization problems:

$$\min_{T \in \mathbb{T}_p(\mathbb{R})} f(P,T), \quad \text{for a given } P \in \mathbb{P}_p(\mathbb{R}), \tag{IV.36}$$

$$\min_{P \in \mathbb{P}_p(\mathbb{R})} f(P,T), \quad \text{for a given } T \in \mathbb{T}_p(\mathbb{R}). \tag{IV.37}$$

Remark that Problem (IV.36) is a classical optimization problem under constraints on the ℓ_1-norm of the matrix T. Since the objective function is convex in T, there exists a unique solution of (IV.36). As regards (IV.37), the objective function is differentiable. Since the space of constraints $\mathbb{P}_p(\mathbb{R})$ is non-convex, we propose to relax the constraints, considering the set of extreme points of the permutation matrices: the set of bistochastic matrices [HJ85], denoted $\mathbb{B}_p(\mathbb{R})$ thereafter. A solution is then given using projected gradient descent algorithm. While the overall problem is difficult to solve, each sub-problem can be solved efficiently.

Let $P_0 \in \mathbb{P}_p(\mathbb{R})$ a permutation matrix. Figure IV.4 represents the structure of the alternating optimization algorithm.

In Section 5.2.1, we carrefully study the set of stricly lower triangular matrices and we provide an explicit formulae for the unique $T \in \mathbb{T}_p(\mathbb{R})$ solution of (IV.36) given $P \in \mathbb{P}_p(\mathbb{R})$. In Section 5.2.2, we focus on the set of bistochastic matrices. We propose an algorithm devoted to find a solution of (IV.37), with relaxed constraints. In Section 5.2.3, we are finally interested in the approximation of any bistochastic matrix by a permutation matrix.

5.2.1 Minimization over $T \in \mathbb{T}_p(\mathbb{R})$

The aim of this section is to solve (IV.36) obtained from (IV.8) freezing the variable P.

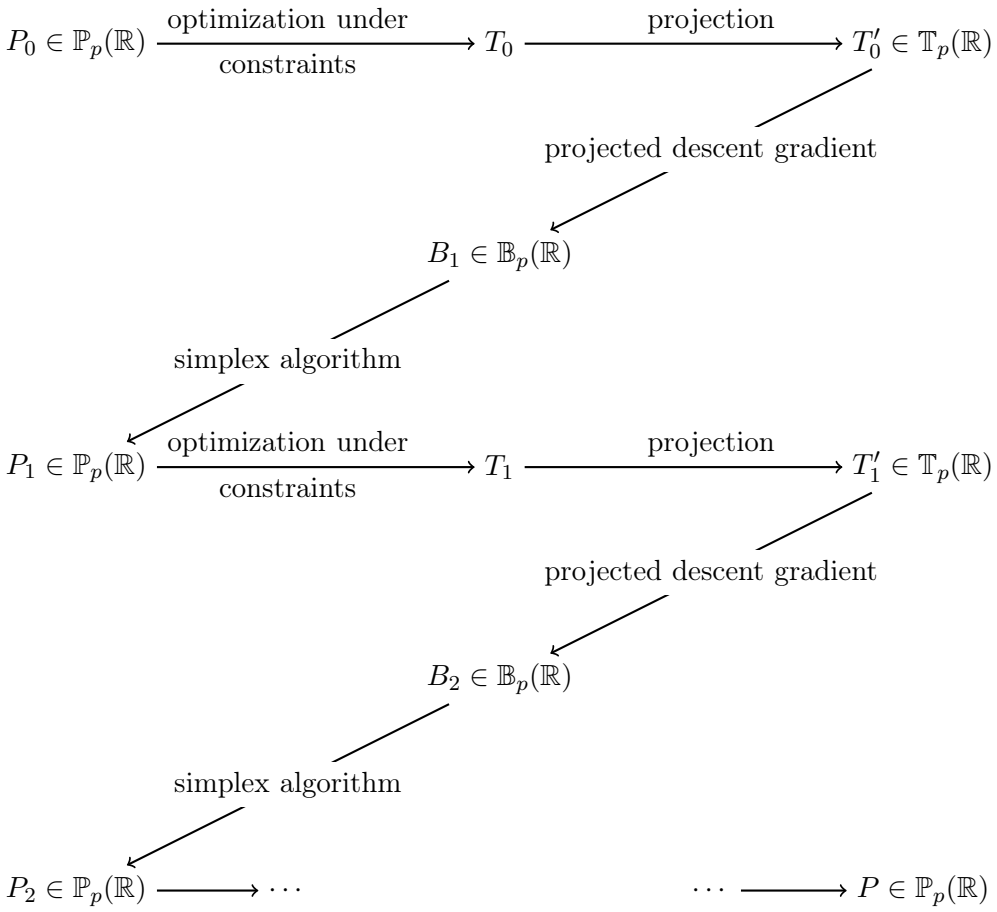


Figure IV.4: Alternating minimization devoted to solve (IV.8).

The set of strictly lower triangular matrices $\mathbb{T}_p(\mathbb{R})$

Proposition 5.2. *The set of stricly lower triangular matrices is a closed convex subspace of $\mathcal{M}_{p,p}(\mathbb{R})$.*

From Theorem 5.1, we then deduce that, for all $M \in \mathcal{M}_{p,p}(\mathbb{R})$, there exists a unique projection of M on $\mathbb{T}_p(\mathbb{R})$, given by Proposition 5.3 below:

Proposition 5.3. *Let $M \in \mathcal{M}_{p,p}(\mathbb{R})$ a $p \times p$ matrix. The projection of M on $\mathbb{T}_p(\mathbb{R})$ is the matrix $\left(\bar{M}_i^j\right)_{i,j}$, which elements are defined as:*

$$\bar{M}_i^j = \left(\mathrm{Proj}_{\mathbb{T}_p(\mathbb{R})}(M)\right)_i^j = \begin{cases} 0 & \text{if } i < j, \\ M_i^j & \text{otherwise.} \end{cases}$$

In other words, the projection of M on $\mathbb{T}_p(\mathbb{R})$ is given by vanishing all the upper coefficients of M:

$$\mathrm{Proj}_{\mathbb{T}_p(\mathbb{R})}(M) = \begin{pmatrix} 0 & \cdots & 0 \\ & \ddots & \vdots \\ M_i^j & & 0 \end{pmatrix}.$$

Proof. Let $\bar{M} \in \mathcal{M}_{p,p}(\mathbb{R})$ such that:

$$\bar{M}_i^j = \begin{cases} 0 & \text{if } i < j, \\ M_i^j & \text{otherwise.} \end{cases}$$

Check that $\bar{M}$ satisfies the characterization of a projection, given by Corrolary 5.1:

1. $\bar{M} \in \mathbb{T}_p(\mathbb{R})$,
2. let $T \in \mathbb{T}_p(\mathbb{R})$, then

$$T_i^j = \begin{cases} 0 & \text{if } i < j, \\ T_i^j & \text{otherwise.} \end{cases} \quad \text{and} \quad \left(M - \bar{M}\right)_i^j = \begin{cases} M_i^j & \text{if } i < j, \\ M_i^j - M_i^j = 0 & \text{otherwise.} \end{cases}$$

We thus deduce that:

$$\begin{aligned} \langle M - \bar{M}, T \rangle_F &= \sum_{i,j} (M - \bar{M})_i^j T_i^j \\ &= 0. \end{aligned}$$

Then, $M - \bar{M} \in \mathbb{T}_p(\mathbb{R})$, which ends the proof.

□

Procedure of optimization The objective function to minimize can be split into a sum of two functions: $g(T) = \frac{1}{n} \left\| X - XPT^tP \right\|_F^2$ and $h(T) = \lambda \|T\|_1$, where g is convex, differentiable and quadratic. As a consequence, g is L-Lipschitz differentiable and satisfies Proposition 5.1 for all $p \times p$ matrices T and U:

$$\forall T, U \in \mathcal{M}_{p,p}(\mathbb{R}), \quad g(T) \leq g(U) + \langle \nabla g(U), T - U \rangle_F + \frac{L}{2} \|T - U\|_F^2 .$$

In the spirit of [Wei08], a natural idea to minimize the function $f(., T) = g(T) + h(T)$ consists in defining a sequence $(T_k)_{k \geq 0}$ such that:

$$T_{k+1} = \underset{T}{\operatorname{argmin}} \left\{ g(T_k) + \langle \nabla g(T_k), T - T_k \rangle_F + \frac{L}{2} \|T - T_k\|_F^2 + h(T) \right\}, \tag{IV.38}$$

which ensures that the sequence $\left(g(T_k) + h(T_k)\right)_k$ decreases. Equation (IV.38) can also be written as:

$$\begin{aligned} T_{k+1} &= \underset{T}{\operatorname{argmin}} \left\{ \langle \nabla g(T_k), T - T_k \rangle_F + \frac{L}{2} \|T - T_k\|_F^2 + h(T) \right\} \\ &= \underset{T}{\operatorname{argmin}} \left\{ \langle \nabla g(T_k), T - T_k \rangle_F + \frac{L}{2} \|T - T_k\|_F^2 + \frac{L}{2} \left\| \frac{\nabla g(T_k)}{L} \right\|_F^2 + h(T) \right\} \\ &= \underset{T}{\operatorname{argmin}} \left\{ \frac{L}{2} \left\| T - \left(T_k - \frac{\nabla g(T_k)}{L} \right) \right\|_F^2 + h(T) \right\}. \end{aligned} \tag{IV.39}$$

Then, denote $T_k^0 = T_k - \frac{\nabla g(T_k)}{L}$. Equation (IV.39) becomes:

$$T_{k+1} = \underset{T}{\operatorname{argmin}} \left\{ \frac{L}{2} \left\| T - T_k^0 \right\|_F^2 + \lambda \|T\|_1 \right\}.$$

In an element-wise formulation, this writes:

$$\forall i,j \in [\![1,p]\!], \quad (T_{k+1})_i^j = \underset{T_i^j \in \mathbb{R}}{\operatorname{argmin}} \left\{ \frac{L}{2}\left(T_i^j - (T_k^0)_i^j\right)^2 + \lambda \left|T_i^j\right| \right\}. \tag{IV.40}$$

Lemma 5.1 below gives an explicit solution of Equation (IV.40).

Lemma 5.1. *Denote $\varphi(x) = (x-x_0)^2 + \frac{2\lambda}{L}|x|$. A solution of the optimization problem $\min_{x\in\mathbb{R}} \varphi(x)$ is given by:*

$$x = \operatorname{sign}(x_0)\max\left(0, |x_0| - \frac{\lambda}{L}\right). \tag{IV.41}$$

Proof. Using Theorem 5.2, to prove this result, we only need to show that $0 \in \partial\varphi(x)$ for $x = \operatorname{sign}(x_0)\max\left(0,|x_0| - \frac{\lambda}{L}\right)$. By definition of the subgradient, $0 \in \partial\varphi(x)$ if and only if, for all $y \in \mathbb{R}$,

$$(x-x_0)^2 + \frac{2\lambda}{L}|x| \leq (y-x_0)^2 + \frac{2\lambda}{L}|y|. \tag{IV.42}$$

Assume that $x_0 > \frac{\lambda}{L}$ $(x > 0)$. We then have $x = x_0 - \frac{\lambda}{L}$. Moreover, for $t > 0$,

$$\frac{\partial\varphi(t)}{\partial t}(x) = 0,$$

and x minimizes φ on $\mathbb{R}^+$. Equation (IV.42) then holds for all $y > 0$. Consider now $y < 0$. Since $x_0 > 0$, $\varphi(y) > \varphi(-y)$. We then deduce that Equation (IV.42) is satisfied for all $y \neq 0$. Consider finally that $y = 0$:

$$\begin{aligned}(-x_0)^2 - (x-x_0)^2 - \frac{2\lambda}{L}x &= x_0^2 - \left(\frac{\lambda}{L}\right)^2 - \frac{2\lambda}{L}\left(x_0 - \frac{\lambda}{L}\right) \\ &= \left(x_0 - \frac{\lambda}{L}\right)^2 \geq 0.\end{aligned}$$

Equation (IV.42) then holds for all $y \in \mathbb{R}$, which ends the proof of Lemma 5.1. □

Using Lemma 5.1, we rewrite Equation (IV.40) as:

$$(T_{k+1})_i^j = \operatorname{sign}((T_k^0)_i^j)\max\left(0, \left|(T_k^0)_i^j\right| - \frac{\lambda}{L}\right), \tag{IV.43}$$

with $T_k^0 = T_k - \frac{\nabla g(T_k)}{L}$, where the gradient of $g(T) = \frac{1}{n}\left\|X - XPT^tP\right\|_F^2$ is given by Proposition 5.4 below.

Proposition 5.4. *The gradient of $g(T) = \frac{1}{n}\left\|X - XPT^tP\right\|_F^2$ is defined as:*

$$\nabla g(T) = -\frac{2}{n}{}^t(XP)(X - XPT^tP)P.$$

Proof. Let $T, H \in \mathcal{M}_{p,p}(\mathbb{R})$ two squared matrices. We then have,

$$\begin{aligned} g(T+H) &= \frac{1}{n}\left\|X - XP(T+H)^tP\right\|_F^2 \\ &= \frac{1}{n}\left\|X - XPT^tP\right\|_F^2 - \frac{2}{n}\langle XPH^tP, X - XPT^tP\rangle_F + o(H) \\ &= g(T) - \frac{2}{n}\langle H, {}^t(XP)(X - XPT^tP)P\rangle_F + o(H), \end{aligned}$$

which implies:

$$g(T+H) - g(T) = -\frac{2}{n}\langle H, {}^t(XP)(X - XPT^tP)P\rangle_F + o(H^2).$$

The result follows by identification. □

Algorithm 8 recaps the minimization procedure of (IV.36), with projection on the space of constraints $\mathbb{T}_p(\mathbb{R})$, given an arbitray permutation matrix P.

Algorithm 8: Minimization over $T \in \mathbb{T}_p(\mathbb{R})$

Input: $P \in \mathbb{P}_p(\mathbb{R})$ a permutation matrix, $X \in \mathcal{M}_{n,p}(\mathbb{R})$ the design matrix, L the Lipschitz-differentiability constant of $g(T) = \frac{1}{n}\left\|X - XPT^tP\right\|_F^2$, λ the penalization parameter, k_{up} the maximal number of iterations and $\epsilon > 0$ the precision.

Initialisation: $T_0 \in \mathcal{M}_{p,p}(\mathbb{R})$ the null squared $p \times p$ matrix, $k = 0$ and $e = +\infty$.

for $k = 1$ *to* k_{up} **do**

  if $e > \epsilon$ **then**

    Compute $T_k^0 = T_k - \frac{\nabla f(T_k)}{L}$ with Proposition 5.4,

    Compute the current matrix $T_{k+1} = \left((T_{k+1})_i^j\right)_{i,j}$:

$$(T_{k+1})_i^j = \text{sign}\left((T_k^0)_i^j\right)\max\left(0, \left|(T_k^0)_i^j\right| - \frac{\lambda}{L}\right),$$

    Project T_{k+1} on $\mathbb{T}_p(\mathbb{R})$ using Proposition 5.3:

$$T_{k+1} \leftarrow \text{Proj}_{\mathbb{T}_p(\mathbb{R})}(T_{k+1}),$$

    Compute $e = \|T_{k+1} - T_k\|_F$,

    Increase k: $k \leftarrow k+1$.

  end

end

Output: $T_k \in \mathbb{T}_p(\mathbb{R})$ the unique solution of (IV.36).

5.2.2 Minimization over $P \in \mathbb{B}_p(\mathbb{R})$

In this section, we are interested in solving the relaxed problem obtained from (IV.37), with the constraint that P is a bistochastic matrix. Since the second part of the objective function does not depend on P, this problem can be rewritten as:

$$\min_{P \in \mathbb{B}_p(\mathbb{R})} \quad \frac{1}{n}\left\|X - XPT^tP\right\|_F^2. \tag{IV.44}$$

In a slight abuse of notation, denote f this new objective function: $f(P) = \frac{1}{n}\left\|X - XPT^tP\right\|_F^2$.

The set of bistochastic matrices $\mathbb{B}_p(\mathbb{R})$ The set of bistochastic matrices is defined as follows:

Definition 5.5. *Let $M \in \mathcal{M}_{p,p}(\mathbb{R})$. $M = (M_i^j)_{i,j}$ is a bistochastic matrix if*

1. $\forall i,j \in [\![1,p]\!]$, $M_i^j \geq 0$,
2. $\forall i,j \in [\![1,p]\!]$, $\sum_{i=1}^p M_i^j = \sum_{j=1}^p M_i^j = 1$.

We denote $\mathbb{B}_p(\mathbb{R})$ *the set of* $p \times p$ *bistochastic matrices.*

Let $U \in \mathbb{R}^p$ be the vector defined as: $U := \begin{pmatrix} 1 \\ | \\ 1 \end{pmatrix}$. Remark that the set of bistochastic matrices can be written as:

$$\mathbb{B}_p(\mathbb{R}) = \left\{ M \in \mathcal{M}_{p,p}(\mathbb{R}), \quad MU = {}^tMU = U \quad \text{and} \quad M_i^j \geq 0, \quad \text{for all} \quad i,j \in [\![1,p]\!]^2 \right\}.$$

Then, identifying $\mathcal{M}_{p,p}(\mathbb{R})$ with $\mathbb{R}^{p^2}$, we can identify $\mathbb{B}_p(\mathbb{R})$ with the convex polytope:

$$\mathbb{B}_p(\mathbb{R}) \sim \left\{ x \in \mathbb{R}^{p^2}, \quad Ax = b \quad \text{and} \quad x_i \geq 0, \quad \text{for all} \quad i \in [\![1,p^2]\!] \right\}, \tag{IV.45}$$

where $A = \begin{pmatrix} {}^tU & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & | \\ | & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & {}^tU \\ I_p & \cdots & \cdots & I_p \end{pmatrix} \in \mathcal{M}_{2p \times p^2}(\mathbb{R})$ and $b = U \in \mathbb{R}^p$.

Theorem 5.3, from Birkhoff [HJ85], makes the link between the permutation matrices and the bistochastic matrices:

Theorem 5.3 (Birkhoff [HJ85])**.** *The set of extreme points of the bistochastic matrices is the set of permutation matrices.*

For information only, remind that an extreme point of a convex set E is a point x of E which can not be written as a convex combination of elements of E (see Figure IV.5).

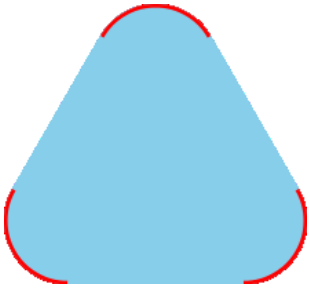


Figure IV.5: An example of a convex set (in blue) and its extreme points (in red).

Proof. A proof of this result is given by [Tak03] and consists in identifying the set of bistochastic matrices with the convex polyedron (IV.45), for which the set of extreme points is known. □

Since the objective function is convex and differentiable, the projected gradient descent algorithm, presented in Section 5.1, can be helpful to solve (IV.44). To this end, we have to know the projection of any matrix on the set $\mathbb{B}_p(\mathbb{R})$, which is the purpose of the next paragraph.

Projection on the set $\mathbb{B}_p(\mathbb{R})$ This paragraph is dedicated to finding an analytic expression of the projection of any matrix on the set $\mathbb{B}_p(\mathbb{R})$. This part relies on the works of [Tak03].

Remark first that $\mathbb{B}_p(\mathbb{R})$ can be written as an intersection of convex sets:

$$\mathbb{B}_p(\mathbb{R}) = \Lambda^+ \cap \mathcal{LC}_1, \tag{IV.46}$$

where

1. $\Lambda^+ = \left\{ M = \left(M_i^j\right)_{i,j} \in \mathcal{M}_{p,p}(\mathbb{R}), \quad \forall i,j \in [\![1,p]\!]^2, \quad M_i^j \geq 0 \right\}$ is a convex cone,
2. $\mathcal{LC}_1 = \left\{ M = \left(M_i^j\right)_{i,j} \in \mathcal{M}_{p,p}(\mathbb{R}), \quad \sum_{i=1}^p M_i^j = \sum_{j=1}^p M_i^j = 1 \right\}$ is an affine subspace.

To project any matrix on $\mathbb{B}_p(\mathbb{R})$, the idea consists in using an alternating projections algorithm, which only needs the knowledge of the projections on Λ^+ and $\mathcal{LC}_1$. This algorithm is also known as the von Neumann algorithm [vN50].

To simplify, let us consider a Hilbert space F and a convex set $E \subset F$ such that $E = A \cap B$, where A and B are two closed convex sets. Denote Proj_E, respectively Proj_A, Proj_B the projection on the set E, respectively A and B. The von Neumann algorithm consists in projecting alternately the point $x \in F$ to project, on A and B to obtain the projection of x on E (see Figure IV.6).

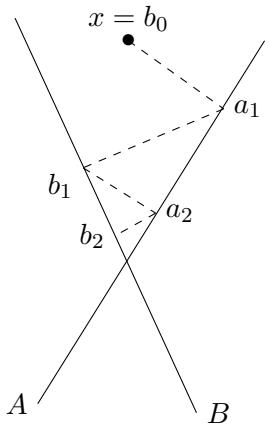


Figure IV.6: The von Neumann Algorithm for the projection of $x \in F$ on the intersection of two convex sets $A \cap B$.

Following the works of [vN50], if the sets A and B are subspaces, the von Neumann Algorithm converges to the projection of x on E. Assume now that one of the two sets is no longer a subspace. This is for example the case in Figure IV.7 where A is a cone. Given x as in Figure IV.7, the projection of x on $E = A \cap B$ is the right end of the segment $A \cap B$. However, the von Neumann Algorithm leads to a point that is stricly in this segment.

Hence, when one of the two spaces is not a subspace, the von Neumann algorithm doesn't always converge to a point of $A \cap B$. To overcome this difficulty, Dykstra [Dyk83] proposes a sligthly modification of this algorithm. Instead of projecting any point x alternately on A and on B, the Boyle-Dykstra algorithm creates two sequels $(p_k)_{k\geq 0}$ and $(q_k)_{k\geq 0}$, said of Dykstra, which correspond to the moving required to project x on both sets A and B. At each step k of the algorithm, the current point x_k is then defined as the projection on A, *resp.* B, of $x_{k-1} + p_{k-1}$, *resp.* $x_{k-1} + q_{k-1}$. The Boyle-Dykstra algorithm is given in Algorithm 9. For a geometrical

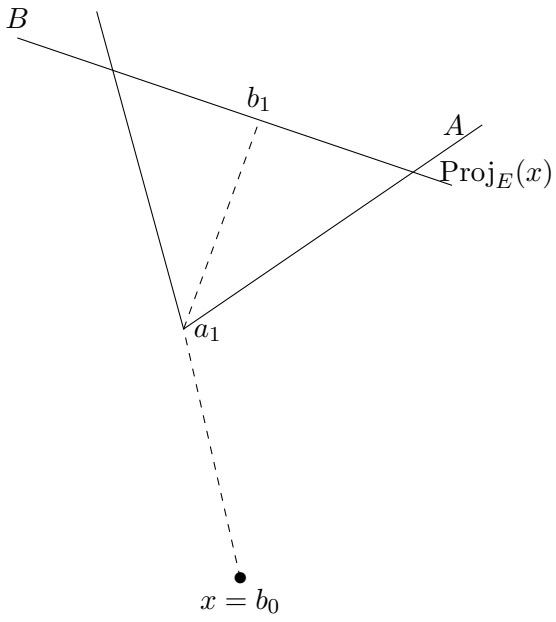


Figure IV.7: The von Neumann algorithm for the projection of $x \in F$ on the intersection of a convex cone A and a subspace B. It doesn't converge to a point of $A \cap B$.

explanation of Algorithm 9, see also Figure IV.8. If one of the two sets, say A, is an affine subspace, remark then that the computation of p_k is nonnecessary (see Figure IV.8).

Algorithm 9: The Boyle-Dykstra algorithm

Input: $x \in F$ the point to project, k_{up} the maximal number of iterations.
Initialization: $a_0 = 0$, $b_0 = x$, $p_0 = 0$ and $q_0 = 0$.
for $k = 1$ *to* k_{up} **do**
  Project on A: $a_k = \text{Proj}_A(b_{k-1} + p_{k-1})$,
  Compute $p_k = (b_{k-1} + p_{k-1}) - a_k$,
  Project on B: $b_k = \text{Proj}_B(a_k + q_{k-1})$,
  Compute $q_k = (a_k + q_{k-1}) - b_k$,
  Increase k: $k \leftarrow k + 1$.
end

The main result related to the convergence of Algorithm 9 is given by Bauschke and Borwein [BB94]:

Theorem 5.4 (Bauschke *et al.* [BB94])**.** *Let F be a Hilbert space, A, B be two closed convex subsets of F such that $A \cap B \neq \emptyset$ and $x \in F$. Consider the sequences of Dykstra, given by Algorithm 9. We then have:*

$$\|b_k - a_k\| \underset{k \to +\infty}{\longrightarrow} \quad \textit{and} \quad \|b_k - a_{k+1}\| \underset{k \to +\infty}{\longrightarrow} 0.$$

Moreover,

$$a_k, b_k \underset{k \to +\infty}{\longrightarrow} \text{Proj}_{A \cap B}(x).$$

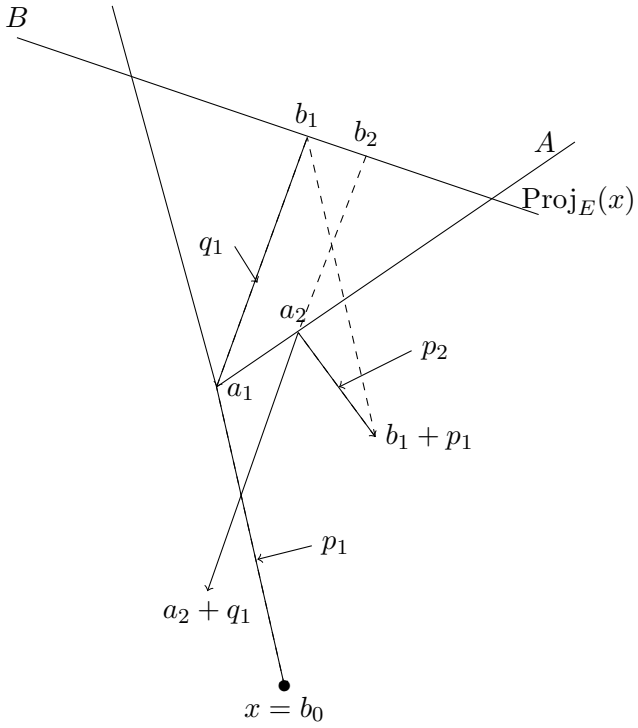


Figure IV.8: The Boyle-Dykstra algorithm for the projection on the intersection of a convex cone A and a subspace B.

Theorem 5.4 justifies the use of the Boyle-Dykstra algorithm to project on an intersection of two convex sets and provides a criterion to stop the algorithm.

The projections on Λ^+ and $\mathcal{LC}_1$ are given by Propositions 5.5 and 5.6 below.

Proposition 5.5 (Projection on Λ^+)**.** *Let $M = \left(M_i^j\right)_{i,j} \in \mathcal{M}_{p,p}(\mathbb{R})$, then, the projection of M on Λ^+ is the matrix M^+ defined as:*

$$\mathrm{Proj}_{\Lambda^+}(M) = M^+, \quad \textit{where} \quad \forall i,j, \quad \left(M^+\right)_i^j = \max\left(M_i^j, 0\right). \tag{IV.47}$$

Proof. Let $M = \left(M_i^j\right)_{i,j} \in \mathcal{M}_{p,p}(\mathbb{R})$ and M^+ the matrix defined by Equation (IV.47). Then, the following conditions are satisfied:

1. $M^+ \in \Lambda^+$,
2. let $N \in \Lambda^+$. On the one hand, all the elements of N are non-negative. On the other hand, by definition of M^+, the non-zero coefficients of $M - M^+$ are negative and correspond to the zero coefficients of M^+. We thus deduce:

$$\begin{aligned}
\langle M - M^+, N - M^+ \rangle_F &= \langle M - M^+, N \rangle_F - \langle M - M^+, M^+ \rangle_F \\
&= \sum_{i,j} \underbrace{\left(M - M^+\right)_i^j}_{\leq 0} \underbrace{N_i^j}_{\geq 0} - \underbrace{\sum_{i,j} \left(M - M^+\right)_i^j \left(M^+\right)_i^j}_{=0} \\
&\leq 0,
\end{aligned}$$

which ends the proof with Proposition 5.1. □

Proposition 5.6 (Projection on $\mathcal{LC}_1$)**.** *Let $M = \left(M_i^j\right)_{i,j} \in \mathcal{M}_{p,p}(\mathbb{R})$, then, the projection of M on $\mathcal{LC}_1$ is the matrix $\bar{M}$ defined as:*

$$\mathrm{Proj}_{\mathcal{LC}_1}(M) = \bar{M} = W_p M W_p + J_p, \quad \textit{where} \quad J_p = \frac{1}{p} U^t U \quad \textit{et} \quad W_p = I_p - J_p. \tag{IV.48}$$

Proof. The proof of this result is given by Takouda [Tak03]. To simplify the notations, the set $\mathcal{LC}_1$ can be written as:

$$\mathcal{LC}_1 = \left\{ M \in \mathcal{M}_{p,p}(\mathbb{R}), \quad MU = {}^tMU = U \right\}, \tag{IV.49}$$

where $U = \begin{pmatrix} 1 \\ | \\ 1 \end{pmatrix}$. Remark that $\mathcal{LC}_1$ is the affine subspace determined by I_p, the identity matrix of order p, and the direction $\mathcal{LC}_0$, where $\mathcal{LC}_0 = \left\{ M \in \mathcal{M}_{p,p}(\mathbb{R}), \quad MU = {}^tMU = 0 \right\}$:

$$\mathcal{LC}_1 = I_p + \mathcal{LC}_0. \tag{IV.50}$$

Let $M \in \mathcal{M}_{p,p}(\mathbb{R})$ and denote $\bar{M} = \mathrm{Proj}_{\mathcal{LC}_1}(M)$. Then, from Theorem 5.1, $\bar{M}$ satisfies:

$$\begin{cases} \bar{M} \in \mathcal{LC}_1 \\ M - \bar{M} \in \mathcal{LC}_1^{\perp}. \end{cases}$$

To find an analytic expression for $\bar{M}$, one of the objective consists in describing the set $\mathcal{LC}_1^{\perp}$.

Consider the function l given by:

$$\begin{array}{rccc} l: & \mathcal{M}_{p,p}(\mathbb{R}) & \longrightarrow & \mathbb{R}^p \times \mathbb{R}^p \\ & M & \longmapsto & (MU, {}^tMU). \end{array}$$

We then have $\mathcal{LC}_0 = \ker l$. We introduce a scalar product on $\mathbb{R}^p \times \mathbb{R}^p$, defined as the following way:

$$\forall x, y, z, t \in \mathbb{R}^p \times \mathbb{R}^p, \langle (x, y), (z, t) \rangle_{p \times p} = \langle x, z \rangle + \langle y, t \rangle.$$

Then, the adjoint operator l^* of l is defined as follows:

$$\forall (x, y) \in \mathbb{R}^p \times \mathbb{R}^p, \forall M \in \mathcal{M}_{p,p}(\mathbb{R}), \quad \langle l(M), (x, y) \rangle_{p \times p} = \langle M, l^*(x, y) \rangle. \tag{IV.51}$$

Using the definition of l, we have:

$$\begin{array}{rcl} \langle l(M), (x, y) \rangle_{p \times p} & = & \langle (MU, {}^tMU), (x, y) \rangle_{p \times p} \\ & = & \langle MU, x \rangle + \langle {}^tMU, y \rangle \\ & = & \langle M, x^t U + U^t y \rangle, \end{array}$$

which leads to the following definition of l^* from Equation (IV.51):

$$\begin{array}{rccc} l^*: & \mathbb{R}^p \times \mathbb{R}^p & \longrightarrow & \mathcal{M}_{p,p}(\mathbb{R}) \\ & (x, y) & \longmapsto & x^t U + U^t y. \end{array}$$

Using arguments of functional analysis and Equation (IV.50), we can then deduce that:

$$\begin{aligned}\mathcal{LC}_1^{\perp} = \mathcal{LC}_0^{\perp} = (\ker l)^{\perp} &= \operatorname{Im} l^* \\ &= \left\{M \in \mathcal{M}_{p,p}(\mathbb{R}), \quad M = x^t U + U^t y, \quad \text{where } x, y \in \mathbb{R}^p \times \mathbb{R}^p\right\}.\end{aligned}$$

$\bar{M}$ is thus defined as follows:

$$\begin{cases} \bar{M}U = {}^t\bar{M}U = U, \\ M - \bar{M} = x^t U + U^t y, \quad \text{where} \quad x, y \in \mathbb{R}^p \times \mathbb{R}^p. \end{cases} \tag{IV.52}$$

From the second equation of (IV.52), we deduce that:

$$\bar{M} = M - x^t U - U^t y,$$

and the first equation of (IV.52) then gives:

$$\begin{aligned} U &= \left(M - x^t U - U^t y\right) U \\ &= MU - px - U^t y U. \end{aligned} \tag{IV.53}$$

In the same way, we have:

$$\begin{aligned} U &= {}^t\big(M - x^t U - U^t y\big)U \\ &= {}^t MU - U^t x U - py, \end{aligned}$$

which directly implies

$$y = \frac{1}{p}\left({}^t MU - U^t x U - U\right). \tag{IV.54}$$

Including Equation (IV.54) in Equation (IV.53) yields:

$$\begin{aligned} U &= MU - px - \frac{1}{p}U^t\big({}^t MU - U^t x U - U\big)U \\ &= MU - px - \frac{1}{p}U\left({}^t UMU - p\,{}^t Ux - p\right) \\ &= MU - px - \frac{1}{p}U^t UMU + U^t Ux + U, \end{aligned}$$

which implies:

$$(pI_p - U^t U)x = (I_p - \frac{1}{p}U^t U)MU.$$

Denoting $W_p = I_p - \frac{1}{p}U^t U$, we then obtain:

$$pW_p x = W_p MU. \tag{IV.55}$$

Remark that the same computations for y yield to the following equation:

$$pW_p y = W_p\,{}^t MU. \tag{IV.56}$$

To solve Equation (IV.55), we aim first at finding a general solution of $pW_p x = 0$. Remark that pW_p is the matrix:

$$pW_p = \begin{pmatrix} p-1 & -1 & \cdots & -1 \\ -1 & p-1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & -1 \\ -1 & \cdots & -1 & p-1 \end{pmatrix}.$$

Then, using some elementary row and column operations, we can reduce the rank of matrix pW_p to the form:

$$\text{rank}(pW_p) = \text{rank}\begin{pmatrix} p-1 & -1 & \cdots & -1 \\ -p & p & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ -p & 0 & \cdots & p \end{pmatrix} = \text{rank}\begin{pmatrix} 0 & -1 & \cdots & -1 \\ \vdots & p & 0 & 0 \\ & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & p \end{pmatrix}.$$

It follows that $\text{rank}(pW_p) = p-1$ and $\dim(\ker(pW_p)) = 1$. Since $pW_pU = 0$, a general solution for $pW_px = 0$ is then $x = kU$, with $k \in \mathbb{R}$. A particular solution of Equation (IV.55) is $x_{part} = \frac{1}{p}MU$. We thus deduce that the general solutions for Equation (IV.55) are given by:

$$x = kU + \frac{1}{p}MU. \tag{IV.57}$$

Using the same arguments for y, the general solutions of Equation (IV.56) are given by:

$$y = k'U + \frac{1}{p}{}^tMU, \quad \text{where } k' \in \mathbb{R}. \tag{IV.58}$$

From Equations (IV.53), (IV.57) and (IV.58), we deduce that:

$$\begin{aligned} U &= MU - p\left(kU + \frac{1}{p}MU\right) - U^t\left(k'U + \frac{1}{p}{}^tMU\right)U \\ &= -pkU - pk'U - \frac{1}{p}U^tUMU \\ &= -p(k+k')U - J_pMU, \end{aligned}$$

with $J_p = \frac{1}{p}U^tU$. Thus:

$$(k+k')U = -\frac{1}{p}(I_p + J_pM)U. \tag{IV.59}$$

Moreover, $M - \bar{M} = x^tU + U^ty$. From Equations (IV.57) and (IV.58), we then have:

$$\begin{aligned} \bar{M} &= M - \left(kU + \frac{1}{p}MU\right){}^tU - U^t\left(k'U + \frac{1}{p}{}^tMU\right) \\ &= M - kU^tU - \frac{1}{p}MU^tU - k'U^tU - \frac{1}{p}U^tUM \\ &= M(I_p - J_p) - (k+k')U^tU - J_pM \\ &= M(I_p - J_p) + \frac{1}{p}(I_p + J_pM)U^tU - J_pM, \quad \text{using Equation (IV.59)}, \\ &= M(I_p - J_p) + (I_p + J_pM)J_p - J_pM \\ &= M(I_p - J_p) + J_p - J_pM(I_p - J_p) \\ &= (I_p - J_p)M(I_p - J_p) + J_p, \end{aligned}$$

which ends the proof with $W_p = I_p - J_p$. □

An algorithm devoted to project any matrix M on $\mathbb{B}_p(\mathbb{R})$ is then given by Algorithm 10. Remark that, since $\mathcal{LC}_1$ is an affine subspace, the projection on $\mathcal{LC}_1$ is linear and the corresponding sequence of Dykstra isn't useful.

Algorithm 10: Projection on $\mathbb{B}_p(\mathbb{R})$

Input: $M \in \mathcal{M}_{p,p}(\mathbb{R})$ the matrix to project, ϵ the precision, k_{up} the maximal number of iterations.

Initialization: $A_0 = 0$, $B_0 = M$, $Q_0 = 0$.

for $k = 1$ *to* k_{up} **do**

  while $\|A_{k-1} - B_{k-1}\| < \epsilon$ **do**

    Project on $\mathcal{LC}_1$:

$$A_k = \text{Proj}_{\mathcal{LC}_1}(B_{k-1}) = W_p B_k W_p + J_p,$$

    by Proposition 5.6,

    Project on Λ^+ :

$$B_k = \text{Proj}_{\Lambda^+}(A_k + Q_{k-1}) = (A_k + Q_{k-1})^+,$$

    by Proposition 5.5,

    Compute $Q_k = (A_k + Q_{k-1}) - B_k$,

    Increase k: $k \leftarrow k + 1$.

  end

end

Output: $\text{Proj}_{\mathbb{B}_p(\mathbb{R})}(M) = A_{k_{up}}$.

Procedure of minimization We finally solve (IV.44) using a projected descent algorithm, described in Section 5.1, where the gradient of the function f to minimize is given by Proposition 5.7 below:

Proposition 5.7. *The gradient of f in P is given by:*

$$\nabla f(P) = -2\left({}^tXX\left(I_p - PT^tP\right)P^tT + \left(I_p - P^tT^tP\right){}^tXXPT\right).$$

Proof. Let $P, H \in \mathcal{M}_{p,p}(\mathbb{R})$. We then have:

$$\begin{aligned}
f(P+H) &= \frac{1}{n}\left\|X - X(P+H)T^t(P+H)\right\|_F^2 \\
&= \frac{1}{n}\left\|X - XPT^tP\right\|_F^2 - \frac{2}{n}\langle XHT^tP + XPT^tH, X - XPT^tP\rangle_F + o(H) \\
&= f(P) - \frac{2}{n}\langle H, {}^tX(X - XPT^tP)^t(T^tP)\rangle_F - \frac{2}{n}\langle {}^t(X - XPT^tP)XPT, H\rangle_F + o(H) \\
&= f(P) - \frac{2}{n}\langle H, {}^tX(X - XPT^tP)P^tT + ({}^tX - P^tT^tP^tX)XPT\rangle_F + o(H).
\end{aligned}$$

Therefore,

$$f(P+H) - f(P) = -\frac{2}{n}\langle H, {}^tXX(I_p - PT^tP)P^tT + (I_p - P^tT^tP)^tXXPT\rangle_F + o(H).$$

And we deduce the result by identification. □

Algorithm (11) sums up the procedure of minimization of (IV.44).

Algorithm 11: Minimization over P

Input: $T \in \mathbb{T}_p(\mathbb{R})$ a stricly lower triangular matrix, X the $n \times p$ design matrix and k_{up} the maximl number of iterations.

Initialization: $P \in \mathbb{B}_p(\mathbb{R})$ a bistochastic matrix.

for $k = 1$ *to* k_{up} **do**

Compute $\nabla f(P)$ from Proposition 5.7,

Compute the current matrix P:

$$P \leftarrow P - \gamma_k \nabla f(P),$$

where $\gamma_k = \underset{\gamma>0}{\operatorname{argmin}} \{f(P - \gamma \nabla f(P)\}$,

Project P on $\mathbb{B}_p(\mathbb{R})$ with Algorithm 10:

$$P \leftarrow \mathrm{Proj}_{\mathbb{B}_p(\mathbb{R})}(P),$$

Increase k: $k \leftarrow k + 1$.

end

Output: $P_{k_{up}} \in \mathbb{B}_p(\mathbb{R})$ a solution of Problem (IV.44).

5.2.3 Approximation by a permutation matrix

The last part of this work deals with the approximation of any bistochastic matrix by a permutation matrix. Given $B \in \mathbb{B}_p(\mathbb{R})$, this leads to find the permutation matrix $P \in \mathbb{P}_p(\mathbb{R})$ the closest (in terms of norm) to B:

$$\min_{P \in \mathbb{P}_p(\mathbb{R})} \|B - P\|_F \,. \tag{IV.60}$$

By developping the squared of the norm, we can easily show that the objective function to minimize can be written as:

$$\begin{aligned} \|B - P\|_F^2 &= \|B\|_F^2 + \|P\|_F^2 - 2\langle B, P\rangle_F \\ &= \|B\|_F^2 + p - 2\langle B, P\rangle_F. \end{aligned}$$

Problem of optimization (IV.60) then becomes:

$$\min_{P \in \mathbb{P}_p(\mathbb{R})} -2\langle B, P\rangle_F, \tag{IV.61}$$

where the function $-2\langle B, P\rangle_F$ to minimize is linear. The set of constraints $\mathbb{P}_p(\mathbb{R})$ is still non-convex, but corresponds to the set of extreme points of the bistochastic matrices, which can be associated to a convex polyedron by Equation (IV.45). To solve Problem (IV.8), we thus use the simplex algorithm, presented in Section 5.1, particularly adapted to find an extreme point solution of a linear problem.

Remark that the approximation of any bistochastic matrix B by a permutation matrix is a non-trivial problem. The closer to a vertex of the polyedron (IV.45) B is, the better the approximation is. On the contrary, when B is close to the center of the polyedron, this approximation may be rough.

In practice, the alternating minimization procedure presented here suffers from some negative points:

— efficience: the computational time is quite large to obtain a solution with sufficient precision,

— fiability: the projected descent gradient algorithm is sensitive to the initial point choice and can sometimes lead to a local minimum.

5.3 Procedure of optimization based on genetic algorithms

A second method devoted to solve (IV.60) could be to use global optimization algorithms, able to explore the set of permutation matrices. Among the discrete optimization methods that exist in the litterature, we focus on Genetic Algorithms.

5.3.1 Review of the genetic algorithms

Genetic algorithms (GA) are stochastic global search methods that have proven to be successful for many kinds of optimization problems like wire-routing, transportation problems, traveling salesman problem... (see for instance [Mic94]). They are able to search very large solution spaces efficiently by providing a small computational cost, mimicking the process of natural evolution. GA are also related with the evolution theory: the genes that passed down over the generations of a population are the most adapted to the needs of this population.

GA devoted to solve optimization problems have been introduced in the 60s by researchers of the University of Michigan, particularly Holland [Hol92]. At this time, the most important innovation was the creation of a crossover operator, associated to a mutation operator. Indeed, this operator, which combines genes of different individues of a population, improves the convergence of the population to an optimal point of the considered objective function. Genetic Algorithms were then popularized by Goldberg [Gol89] in 1989.

GA use a vocabulary derived from natural genetics. We thus talk about individuals (or genotypes, chromosomes) in a population. Chromosomes are made of units, called genes, arranged in linear succession. Genes are located at certain positions of the chromosome, called loci. Each individual represents a potential solution to a problem of optimization. Genetic Algorithms run on a population of individuals and correspond to a search through the space of potential solutions. Remark that this search requires balancing between two objectives: exploiting the best solutions and exploring the search space. To obtain this balance, Genetic Algorithms work with a population of candidate solutions and try to optimize the objective function by means of three natural principles: mutation, crossover and selection operators.

More precisely, the structure of a simple Genetic Algorithm is given in Algorithm 12. During iteration k, a Genetic Algorithm maintains a population of potential solutions $\text{Pop} = \{x_1^k, ..., x_N^k\}$. Each solution x_i^k is evaluated to give some measure of its fitness. Then, a new population (iteration $k+1$) is formed by selecting the fitest individuals. Some individuals of this new population undergo alterations by means of crossover and mutation to form new solutions. The step of crossover aims at combining the features of two parent chromosomes to form two new chromosomes close to its parents, by swapping segments of the parents. To add some variability in the population, the mutation operator arbitrarily alters one or more genes of a selected chromosome by a random change with the mutation rate.

Since the works of [Hol92], a large number of study has been dedicated to provide rigourous mathematical analyse of GAs. The first mathematical convergence results were obtained by [Cer96] and [Cer98], who constructed an asymptotic theory for GAs.

Algorithm 12: The structure of a Genetic Algorithm

Input: Pop an initial population.
Initialization: $t = 0$ and $\text{Pop}(0) := \text{Pop}$.
while *(**not** terminate-condition)* **do**
  $t \leftarrow t + 1$,
  Select $\text{Pop}(t)$ from $\text{Pop}(t-1)$,
  Alter $\text{Pop}(t)$,
  Evaluate $\text{Pop}(t)$.
end

5.3.2 Adaptation to Problem of optimization (IV.8)

The aim of this section is to develop a Genetic Algorithm to solve the problem of optimization (IV.8):

$$(\hat{P}, \hat{T}) = \underset{P \in \mathbb{P}_p(\mathbb{R}), T \in \mathbb{T}_p(\mathbb{R})}{\operatorname{argmin}} \left\{ \frac{1}{n} \left\| X - XPT^tP \right\|_F^2 + \lambda \|T\|_1 \right\}.$$

Following the works of [Mic94], one has to carefully define the genetic operators presented in Section 5.3.1. The objective is twofold: first, define the chromosomes of the population. These chromosomes have to resume all the information contained in a graph, while being the most minimal. Secondly, the crossover, selection and mutation operators must have a graphical sense.

Problem definition For any given $P \in \mathbb{P}_p(\mathbb{R})$, the penalized maximum likelihood estimate is:

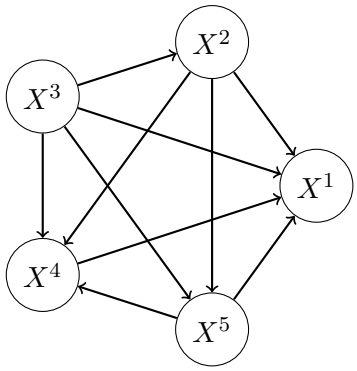
$$T^* = \underset{T \in \mathbb{T}_p(\mathbb{R})}{\operatorname{argmin}} \left\{ \frac{1}{n} \left\| X - XPT^tP \right\|_F^2 + \lambda \|T\|_1 \right\}. \tag{IV.62}$$

Hence, the optimization task (IV.8) comes down to exploring the $\mathbb{P}_p(\mathbb{R})$ space of permutation matrices in dimension p. As any $P \in \mathbb{P}_p(\mathbb{R})$ is uniquely defined by a permutation vector of $[\![1, p]\!]$ (see Example 7), the search space used is $\mathfrak{S}(p)$ (the set of permutations of $[\![1, p]\!]$), which is a more suited formulation for optimization.

Example 7. *Consider the permutation matrix ($p = 5$):*

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix},$$

which leads to the graph



Then, P is represented by the vector | 1 | 4 | 5 | 2 | 3 | *, looking at the ranks of non-values of P column by column. The nodes are ranked according to their number of parents, from large at the left (X^1) to small at the right (X^3).*

Note that our problem resembles the classical traveling salesman problem (TSP), which has been succesfully addressed by means of genetic algorithms [GGRVG85] [Dav91]. Identically to the TSP, we optimize over the space of permutations. This induces specific constraints for defining the crossover and mutation operators. Unlike the TSP however, the permutation here defines a hierarchy between nodes rather than a path, which makes the use of TSP-designed operators a poor solution.

Given the initial population, the Genetic Algorithm works as presented in Figure IV.9. The genetic operators (crossover, mutation and evaluation) only affect the permutations. Each step of the Genetic algorithm thus requires a step of evaluation to compute the best T^* associated to each permutation. The genetic operators we use are described in the next paragraphs.

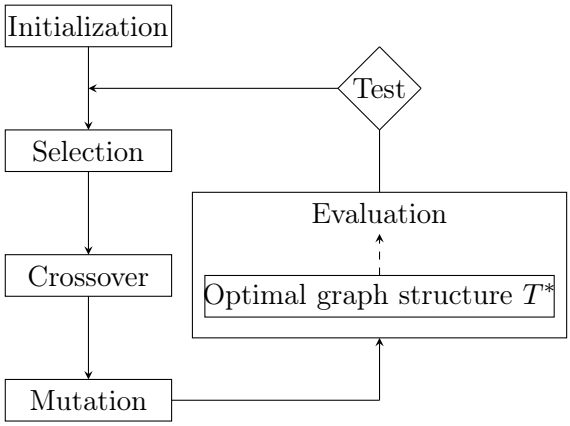


Figure IV.9: Scheme of our proposed Genetic Algorithm.

Fitness function Given a potential solution $p_i \in \mathfrak{S}_p$, the fitness function is defined as:

$$f(p_i) = \frac{1}{n} \left\| X - X P_i T_i^{*t} P_i \right\|_F^2 + \lambda \left\| T_i^* \right\|_1 ,$$

with P_i constructed from p_i as in Example 7 and T_i^* the solution of Equation (IV.62) with $P = P_i$. Hence, each evaluation of the fitness function requires running the sequential procedure described in Section 5.2.1.

Initialization The first step of the algorithm consists in generating an initial population of size N. Note that N must be chosen neither too small (to avoid early convergence) nor too large (to avoid waste of convergence ressources). Here, the initialization step amounts to generating N elements of $\mathfrak{S}_p$ and evaluating their fitness.

Crossover operator A crossover operator generates a new set of potential solutions (children) from existing solutions (parents). Crossover aims at achieving at the same time a good exploration of the search space (by mixing the characteristics of the parents) while conserving some of the characteristics of the parents (exploiting the best solutions). In a classical GA, a typical crossover consists in swapping two sections of the chromosomes of two parents to produce two children. Our crossover procedure must ensure that (a) the children are in $\mathfrak{S}_p$ and (b) the children inherit "as much as possible" of the characteristics of each parent.

Our crossover operator is defined as follows. Given two parents p_1 and p_2, a random set of crossover points are selected. We denote it Ω. It consists in a k-permutation of $[\![1, p]\!]$, with k uniformly drawn between 0 and p. A first child C_1 between p_1 and p_2 is then generated by:

1. swapping the crossover points of p_1 with those of p_2,
2. completing C_1 with the missing numbers in the order they appear in p_2.

A second child C_2, complementary of C_1, is created with the same procedure, replacing p_1 with p_2 (see Example 8, example of a crossover between two permutations).

Example 8. *Consider the two following parents:*

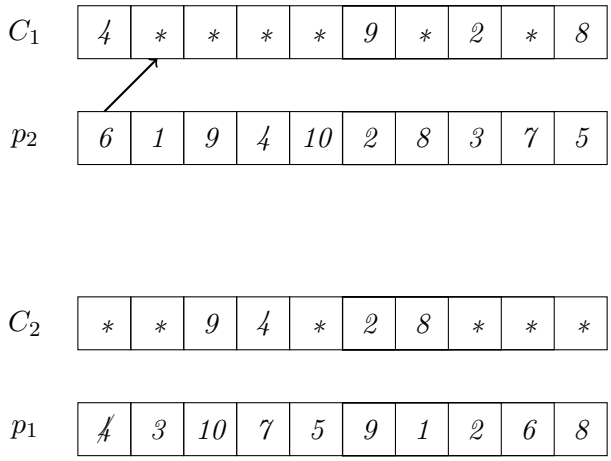
p_1	4	3	10	7	5	9	1	2	6	8
p_2	6	1	9	4	10	2	8	3	7	5

Assume that the crossver points randomly chosen are 4, 9, 2 *and* 8 *(in red above and below). Then, the two children C_1 and C_2 between p_1 and p_2 are:*

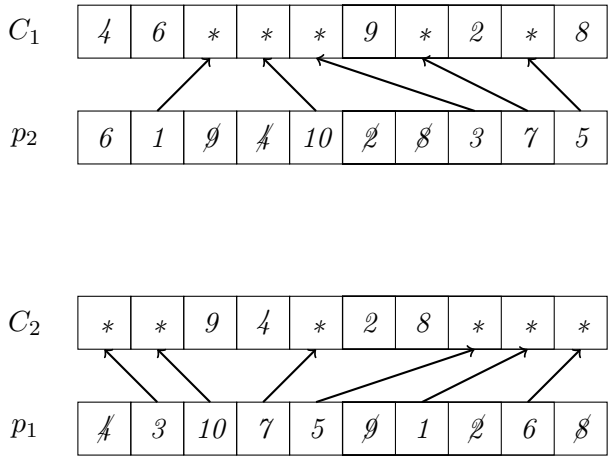
C_1	4	*	*	*	*	9	*	2	*	8
C_2	*	*	9	4	*	2	8	*	*	*

where "" represents a code that needs to be decided in the following step to make a entire individual:*

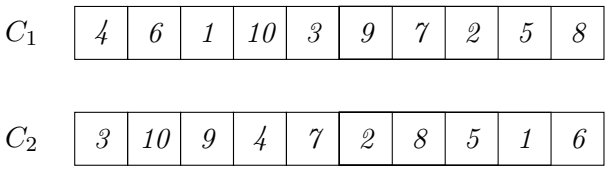
1. *Select the first code "6" in p_2 and compare with the decided codes in C_1. Since "6" doesn't appear in C_1, put "6" in the first possible position (shown in the arrow below). In the same way, select the first code "4" in p_1 and compare with the decided codes in C_2. Since "4" appears in C_2, give up "4".*



2. Repeat this operation to obtain the entire C_1 and C_2.



The finally got new individuals are:



From a graphical point of view, a crossover between two permutations p_1 and p_2, which encode two complete graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, constructs a new graph $\mathcal{G}$ composed of the subgraph of $\mathcal{G}_1$ induced by Ω and the subgraph of $\mathcal{G}_2$ induced by Ω^C, where Ω^C is the complementary set of Ω in $[\![1, p]\!]$ (for more details see Figure IV.10 below).

We ensure that the children graphs generated by crossovering two parents are in a sense closer to their parents. We remark two facts here: (a) the larger the number of crossover points, the more similar to their parents the children look, and (b) the more similar the parents, the more similar the children.

Choosing the parents Part of the crossover step consists in choosing the parents to generate the children. Here, we select N times a random pair of parents. To favour the best individuals, each parent is selected according to a *probability of crossover* that depends on its objective value. We thus define a selection process for crossover based on roulette wheel (for further details, see [Mic94]). We construct such a roulette wheel as follows:

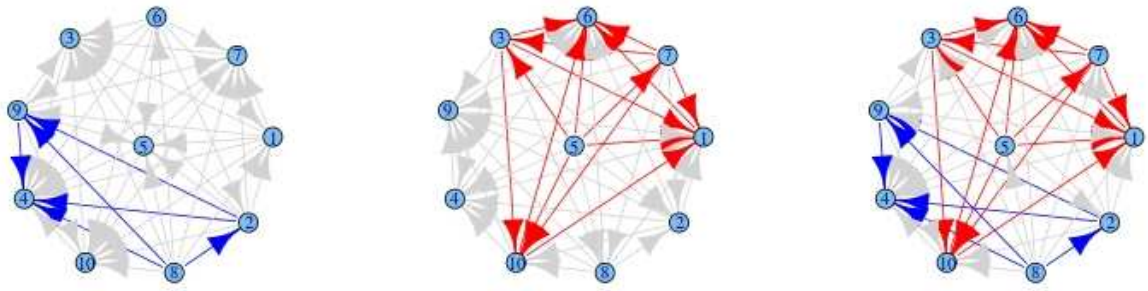


Figure IV.10: Graphical representation of crossover when the number of nodes is set to 10. The graph in blue, respectively red, is the first, respectively second, parent. One of the two children is the third represented graph.

— Compute the fitness value $f(p_i)$ for each chromosome p_i, $i = 1, ..., N$.
— Find the minimal and the maximal value of the fitness of the population:

$$f_{min} = \min_{1 \leq i \leq N} f(p_i) \quad \text{and} \quad f_{max} = \max_{1 \leq i \leq N} f(p_i).$$

— Compute a weight $\omega_i \in [0, 1]$ associated to each chromosome p_i:

$$\omega_i = \frac{f(p_i) - f_{min}}{f_{max} - f_{min}}.$$

— Given ϵ a factor of attenuation to be defined, calculate the attenuate weight $\bar{\omega}_i$ of a selection for each chromosome p_i:

$$\bar{\omega}_i = \omega_i(1 - \epsilon) + \epsilon.$$

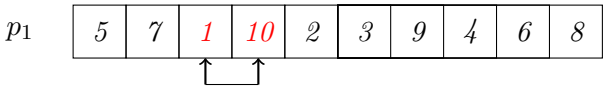
— Compute a cumulative weight q_i for each chromosome p_i:

$$q_i = \sum_{j=1}^{i} \bar{\omega}_j.$$

The selection process is based on spinning the roulette wheel N times. Each time, we select a single chromosome in the following way:
— Generate a random number r from the range $[0, \max_{1 \leq i \leq N} q_i]$.
— If $r < q_1$, then select the first chromosome p_1. Otherwise, select the i-th chromosome p_i $(2 \leq i \leq N)$ such that $q_{i-1} < r \leq q_i$.
Obviously, chromosomes that have the best fitness value, would be selected more than once.

Mutation Mutation operators are used to add some external variability into the population. It usually corresponds to the smallest possible change in an individual. We thus define it as an alteration of two neighbouring genes of any random chromosome. (see Example 9).

Example 9. *Consider the individual:*



and assume that the selected mutant genes are "1" and "10" (in red above and below). Then, the mutated individual is defined as:

M_1	5	7	10	1	2	3	9	4	6	8

Graphically, a mutation consists in switching the arrowhead of an edge between two nodes (see Figure IV.11) and thus corresponds to the smallest possible change for a graph.

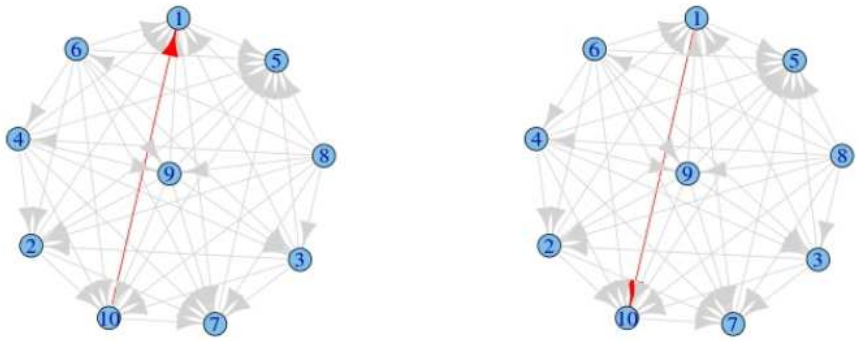


Figure IV.11: Graphical representation of mutation when the number of nodes is set to 10. The first graph represents the individual before mutation, the second, after mutation.

The mutation operator is applied with a probability p_m [Mic94]. The choice of p_m is known to critically affect the behavior and performance of GA. In fact, the mutation rate controls the exploration speed of a new area. Small p_m values are commonly adopted in GA, typically in the range $[0.001, 0.05]$ [DS90].

Selection Given a population of N parents and N children, obtained after crossover and mutation steps, the selection operator brings the population size back to N individuals. In the litterature, several selection operators exist, including elitist strategies (keeping the N best individuals among the parents and children) or non-elitist strategies (replacing all the parents by their children). Here, as most of the convergence of the algorithm is driven by the roulette wheel procedure, the selection operator is chosen as non-elitist, except for the worst child, which is replaced by the best parent.

Stopping criterion The last important point of the implementation of a Genetic Algorithm deals with the stopping criterion. The easiest termination condition consists in defining a maximal number of generations. However, to well-define the total number of generations, some characteristics of the function have to be known. For convenient purpose, the algorithm should end when the probability of a significant improvment becomes relatively weak.

We propose two stopping criteria based on different performance measures. The first one determines a convergence of the population, *i.e.* the heterogeneity of the population along the iterations of the algorithm, using the Shannon entropy.

The Shannon entropy H_j of each locus $j \in [\![1, p]\!]$ of the current population is defined as follows:

$$\forall j \in [\![1, p]\!], \quad H_j = -\sum_{i=1}^{p} p_i \log(p_i),$$

where p_i is the proportion of the i-th distinct genes in the population (see Example 10 below), and the population entropy is:

$$H = \sum_{j=1}^{N} H_j.$$

Example 10. *Consider the population (a row corresponds to an individual of this population) of length $N = 7$:*

1	*2*	*3*	*4*	*5*	*6*	*7*
1	*3*	*5*	*6*	*2*	*4*	*7*
6	*5*	*2*	*3*	*1*	*4*	*7*
5	*6*	*1*	*3*	*2*	*4*	*7*
1	*3*	*5*	*4*	*2*	*6*	*7*

Looking at column by column the population, its Shannon entropy is obtained by computing:

$$H_1 = \underbrace{-\frac{3}{5}\log\left(\frac{3}{5}\right)}_{\text{proportion of gene } 1} - \underbrace{\frac{1}{5}\log\left(\frac{1}{5}\right)}_{\text{proportion of gene } 5} - \underbrace{\frac{1}{5}\log\left(\frac{1}{5}\right)}_{\text{proportion of gene } 6},$$

$$H_2 = -\frac{1}{5}\log\left(\frac{1}{5}\right) - \frac{2}{5}\log\left(\frac{2}{5}\right) - \frac{1}{5}\log\left(\frac{1}{5}\right) - \frac{1}{5}\log\left(\frac{1}{5}\right),$$

and so on... Finally, the Shannon entropy of the population equals:

$$H = (0.9503 \quad 1.3322 \quad 1.3322 \quad 1.0549 \quad 0.9503 \quad 0.6730 \quad 0).$$

The rank position entropy measures local individual variability over the population: it is 0 when this position contains identical genes (i.e., all the potential solutions have the same rank value for node j), and is otherwise positive with maximal value when all genes are distinct. Entropy may be high at the beginning of the optimization procedure (exploration) and hopefully decreases towards zero with the convergence of the algorithm. Then, when the Shannon entropy is smaller than a given threshold, the Genetic Algorithm has to be stopped.

Although the Shannon entropy corresponds to our practical objective (convergence of the current population to a single optimal chromosome), a fitness-based criterion is necessary since the global optimum is certainly not unique in the case of sparse graphs, for which a large number of permutations may provide identical estimates (see for instance Example 11 below).

A second criterion we monitor is the evolution of the objective function on the population over the iterations ($\frac{1}{N}\sum_{i=1}^{N} f_(p_i)$): optimization is stopped when the average fitness of the current population does not change during several consecutive iterations.

Example 11. *Consider the star graph (a unique node regulates all the other nodes) composed of* 30 *nodes and represented in Figure IV.12.*

One can remark that inferring this network consists in finding the role of node 1*. More precisely, to recover the structure of the DAG, we aim at estimating the following permutation:*

*	*		⋯	*	*	1

to ensure that node 1 *has the largest number of children in the graph. The other genes of the chromosome don't matter since the estimation of matrix T in Equation (IV.62) will make the graph sparse. This example then shows that the Shannon entropy can not converge whereas the Genetic Algorithm can.*

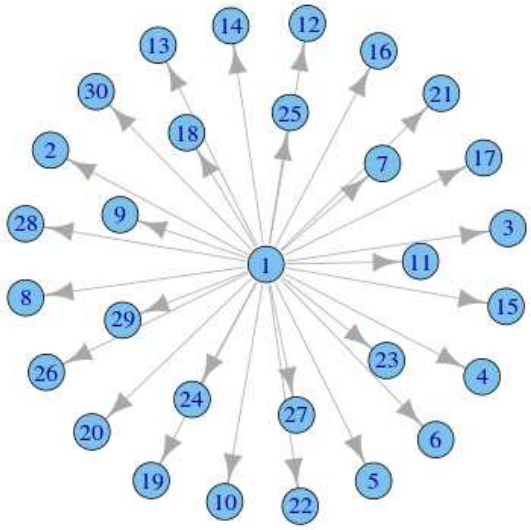


Figure IV.12: A star graph, used in Example 11.

6 Numerical applications

6.1 Parameters of the Genetic Algorithm

Three inherent parameters have to be defined to run the proposed Genetic Algorithm:
— the factor of attenuation of crossover ϵ is fixed to 0.5,
— the rate of mutation p_m is set to 0.01, a commonly used value in the literature,
— the size of the population N. Several researchs have investigated the size of the population for GA: a study on influence of the control parameters is for example presented in [SCED89]. [Mic94] proposes to use a GA with varying population size. In our simulation study, we choose as a rule-of thumb $N = 5p$, which was found as a good compromise on several experiments.

Since most of the simulated networks are sparse, the algorithm ends as soon as the average fitness value becomes smaller than a 1% threshold during at least 20 iterations. The maximal number of iterations is set to 2000.

6.2 Performance evaluation

6.2.1 Algorithms

We compare our GA to other inference methods. The two first methods used decompose the prediction of the network into p regression problems. In each of these regressions, a target variable is predicted from the others using Random Forests [Bre01] or Lasso regression [Tib96]. In all our simulation study, we use the Matlab implementation of GENIE3 [HTIWG10], based on Random Forests, and the Matlab implementation of the Lasso algorithm. GENIE3 was the best performer in the DREAM4 In Silico Multifactorial challenge.

We also compare the performances of GA with the Boost-Boost $\mathcal{D}$-correlation sum algorithm presented in Chapter II and denoted "Boost-Boost algorithm" thereafter. In the GRN inference settings, this algorithm aims at inferring the true DAG by iterative approximations (for more details on this procedure, see Chapter II).

6.2.2 Performance measurments

A classical performance measure for graph inference methods consists in comparing the inferred interactions with the known edges in the true graph $\mathcal{G}_0$ using precision versus recall curves. We denote TP, *resp.* FP, FN and TN, the true positive (correctly inferred) edges, *resp.* the false positive (inferred by mistake) edges, the false negative (missed) edges, and the true negative (correctly non-predicted) edges. Then, the recall, defined as $\frac{TP}{TP+FN}$, measures the power (or sufficiency) of reconstruction of non-zero elements of the true matrix G (or of the true network) for one method, whereas the precision, equal to $\frac{TP}{TP+FP}$, measures the accuracy of the reconstruction. The closer to 1 the precision and the recall the better. We also compute the area under the precision versus recall curve (AUPR) normalized between 0 and 1.

GENIE3 outputs a ranked list of regulatory interactions, which corresponds to the edges of the inferred graph. Edges are then successively introduced with decreasing confidence scores. Contrary to GENIE3, our proposed GA and Lasso are based on penalized optimization: it should find linear dependencies between the variables with a controlled level of parsimony (λ in Equation (IV.8) for GA). For λ varying from 0 (complete graph) to $+\infty$ (empty graph), they produce a list of edges, successively introduced in the model. These lists of edges define the precision versus recall curve.

6.3 Numerical results

6.3.1 Datasets

We consider simulated data from networks with different characteristics (number of nodes and edges) to assess the practical preformances of GA and compare it to competing approaches. We set the mean of the residual values to 0 and their standard deviations to 1. Non-zero parameters of G_0 are simulated according to independent Gaussian distribution $\mathcal{N}(0,1)$. In all our simulations, we always generate $n = 100$ observations. All experiments are replicated 50 times and results are averaged over these replicates. Table IV.1 summarizes the statistics of these networks.

Network	Number of variables p	Number of edges
Network 1-1 and 1-2	30	29
Network 2	100	99
Network 3	30	50

Table IV.1: The four networks used in our experiments. For more details on the differences between Network 1-1 and 1-2, see Figure IV.13 below. Network 2 is a star graph with 100 nodes: node 1 regulates all other nodes in the graph. Network 3 is a non-particular graph and is represented in Figure IV.14.

6.3.2 Results

In Figure IV.15, we present the behaviour of GA for Network 1-1 for a fixed penalization parameter λ. The first curve shows the convergence of the algorithm to the minimum of the objective function. The quantiles are larger in the first steps than at the end of the algorithm, which agree with the choice of the population size N (important mixing at the beginning of the algorithm). The second curve presents the evolution of the Shannon entropy along the iterations of GA. As explained in Example 11, the Shannon entropy doesn't converge to 0 since Network 1-1 is sparse, but one can remark the behaviour of a particular gene (in green). The current

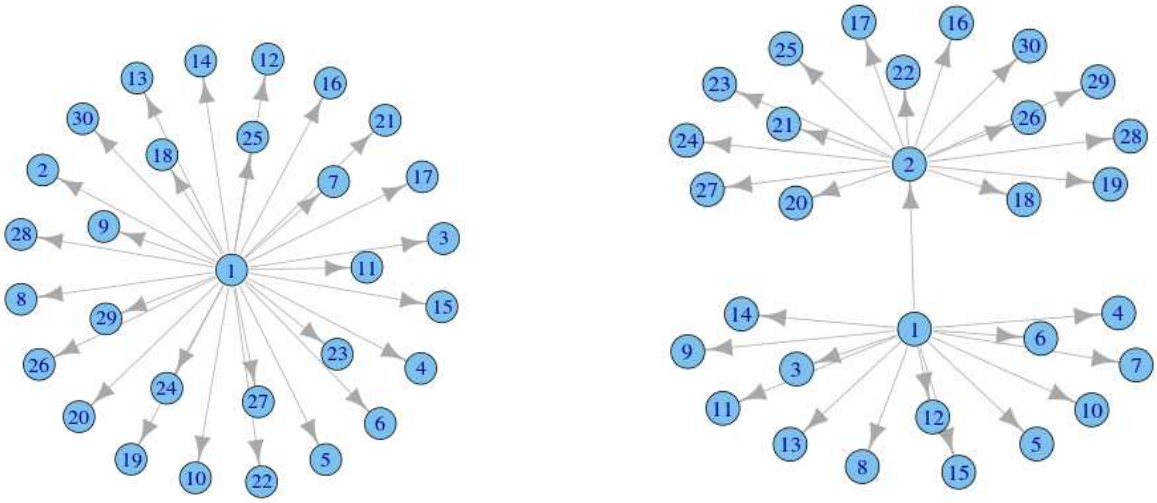


Figure IV.13: Networks 1-1 and 1-2 (see Table IV.1) used in our simulation study.

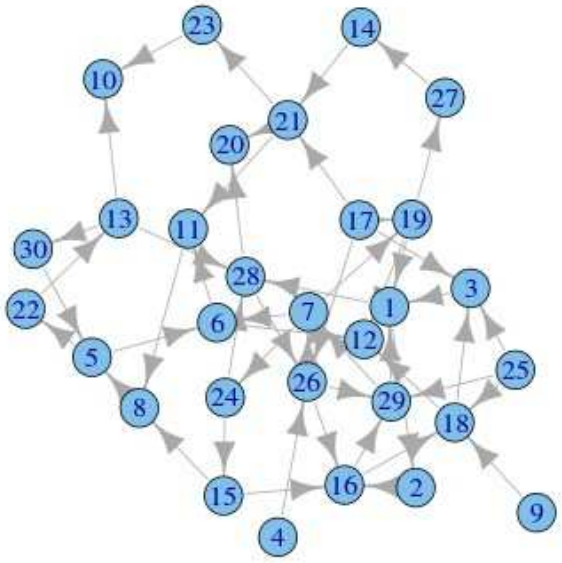


Figure IV.14: Network 3 (see Table IV.1) used in our simulation study.

population seems to converge to a chromosome with only one fixed gene. The last curve finally represents the evolution of node 1 in Network 1-1. After 20 iterations, node 1 is nearly always at the end (locus 30) of the population. As a consequence, GA converges to a graph for which the source vertex is node 1.

Figure IV.16 represent the precision-recall curves for GENIE3, lasso, Boost-Boost algorithm and GA on Networks 1-1 and 1-2. The results are clearly in favour of GA. In both networks, firstly predicted edges are far more accurate than that of other methods. The precision of the three method drops suddenly with a slow increases in recall above 50%. Adding new edges in the model, any of the three methods doesn't identify clearly reliable edges anymore and large number

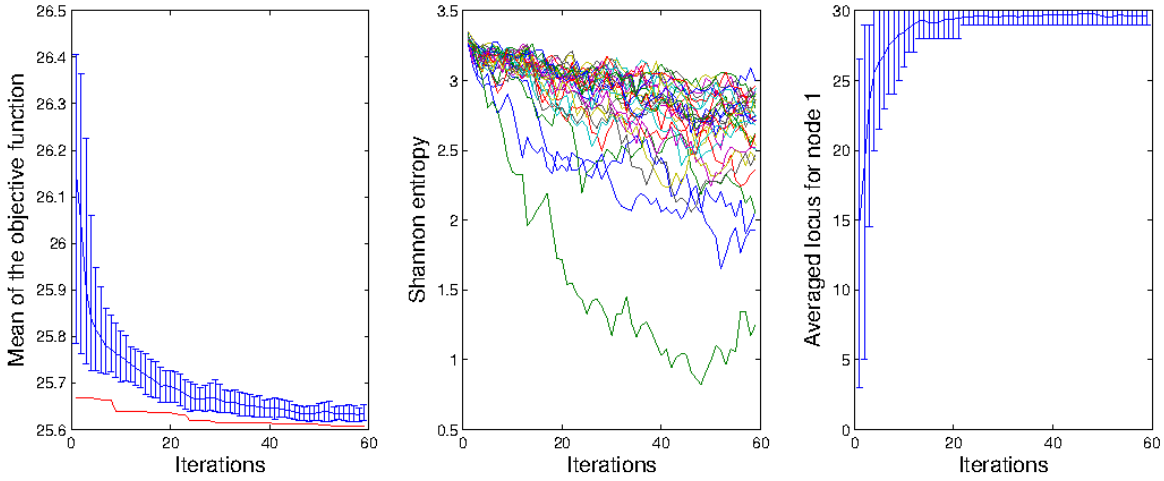


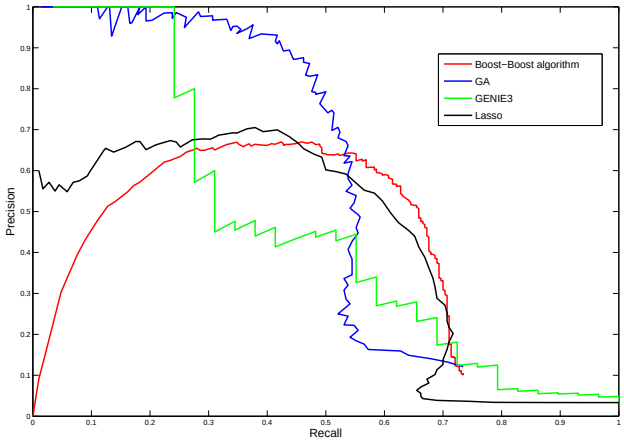
Figure IV.15: Results of GA for Network 1-1 with the penalization parameter λ set to 0.35. The first curve represents the mean of the objective function (in blue) along the iterations of the GA. The minimal value of the current population is represented in red. We also draw the quantiles. The second curve represents the evolution of the Shannon entropy. The last curve finally represents the averaged locus of node 1, the source vertex in Network 1-1.

of FP edges are produced. More precisely, it's more difficult for GA to obtain a recall equals to 1 than other methods since the number of inferred edges can't exceed $\frac{p(p-1)}{2}$ (the inferred graph is a DAG). Note the singular behaviour of Boost-Boost algorithm: the first edges added in the model are inferred by mistake.

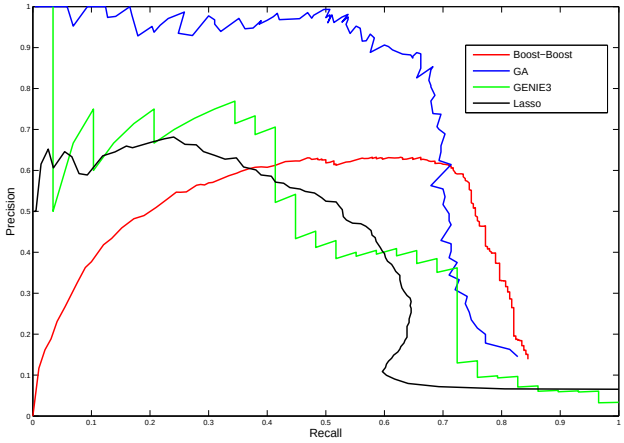
In addition, Table IV.2 summarizes the results of the three methods in terms of area under the precision-recall curve. Concerning Network 1 and 3, our algorithm outperforms the performance results compared to the state-of-the-art methods. When the number of nodes of the graph is large ($p = 100$), the AUPR of both GA and Boost-Boost algorithm is around 0.35, larger than the AUPR of GENIE3, but smaller than that of Lasso. However firstly predicted GA edges are still much more reliable (higher precision between 80 and 100 %). This high-quality of predictions is a feature which is often desirable, *e.g.* in Biology where testing new regulations is obtained from a time-consuming and expensive series of experiments.

Method	GA	Boost-Boost algorithm	GENIE3	Lasso
Network 1-1	**0.5223**	0.3947	0.4212	0.4312
Network 1-2	**0.6919**	0.4315	0.4136	0.3920
Network 2	0.3557	0.3596	0.2718	**0.4112**
Network 3	**0.4171**	0.2332	0.2887	0.3349

Table IV.2: Area under precision-recall curve for all networks and state-of-the-art methods.



(a) Network 1-1



(b) Network 1-2

Figure IV.16: Precision-recall curves for Networks 1-1 and 1-2 for the proposed Genetic Algorithm (in blue) compared with GENIE3 (in green), Lasso (in black) and Boost-Boost algorithm (in red).

Conclusion and perspectives

In this chapter, we propose an approach for network inference with Gaussian observational data using the ℓ_1-penalized log-likelihood. From a theoretical point of view, we provide bounds both in prediction and estimation under assumptions on the model, which unfortunately don't include the high-dimensional scenario. To compute the corresponding estimator, a solution of (IV.8) is obtained through a Genetic Algorithm. The numerical studies presented here are helpful to understand its functioning and aims at comparing it to other Gene Regulatory Networks methods on toy data. As a further work, it could be interesting to test our algorithm on real data. However, one of the main difficulties when using real data comes from the violation of the assumption *equal noise variances* on the model. This assumption is clearly restrictive, but necessary to obtain identifiability.

A major challenge is the network inference for any noise variances. Then, only the Markov equivalence class is identifiable: the inferred graph is undirected. In order to orient some of the edges of the learning graph, one solution consists in incorporating interventional data on the model. Interventional data are observations obtained from perturbations of the system. Then, the log-likelihood estimator we propose has to be adjusted to take into account these new observations. More precisely, in order to represent the effect of any intervention on variable X^i, we use the so-called *do*-operators [Pea00]. The distribution $\mathcal{L}(X^j|do(X^i = x))$, generated by an intervention on variable X^i, represents what would occur if treatment condition $X^i = x$ was enforced uniformly over the population via some intervention. When observational data are jointly modeled with interventional data, the linear Gaussian Structural Equation Model (IV.1) follows a multivariate Gaussian distribution too, which dimension is equal to the number of genes without intervention. The log-likelihood then becomes:

$$l(G) = \frac{1}{n}\sum_{k=1}^{n}\sum_{j\notin\mathcal{I}_k}\left((X - XG)_j^k\right)^2,$$

where $\mathcal{I}_k$ represents the set of genes with intervention for a given $k \in [\![1, n]\!]$ (for further details, see [RJN13]). The Genetic Algorithm then has also to be adapted to this particular data structure.

Synthèse des travaux

Les travaux présentés dans ce manuscrit ont fait l'objet des publications et présentations orales suivantes:

Publications

— [CCAGV14] M. Champion, C. Cierco-Ayrolles, S. Gadat et M. Vignes. Sparse regression and support recovery with $\mathbb{L}_2$-Boosting algorithms. *Journal of Statistical Planning and Inference*. A paraître, 2014.
— [CCGP14] M. Champion, G. Chastaing, S. Gadat et C. Prieur. $\mathbb{L}_2$-Boosting on generalized hoeffding decomposition for dependent variables - application to sensitivity analysis. *Statistica Sinica*. A paraître, 2014.
— M. Champion et V. Picheny. Estimation of sparse directed acyclic graphs: theoretical framework and Genetic Algorithms. En cours de rédaction.

Présentations orales

Congrès internationaux

— SIAM Conference on Uncertainty Quantification "$\mathbb{L}_2$-Boosting on Generalized Hoeffding Decomposition for Dependent Variables - Application to sensitivity Analysis", Savannah, USA, 31 mars 2014.
— NIPS Workshop Machine Learning for Computational Biology, poster "An $\mathbb{L}_2$-Boosting algorithm for sparse multivariate regression: application to gene network recovery", Sierra Nevada, Espagne, 17 décembre 2011.

Congrès nationaux

— 45ièmes Journées de Statistique de la SFDS "Résultats sur les algorithmes de $\mathbb{L}_2$-Boosting pour les régressions parcimonieuses", Toulouse, 27 mai 2013.

Séminaires, école d'été et groupes de travail

— Séminaire de mathématiques appliquées de l'Université de Nantes "Sparse regression and optimization in high-dimensional framework: application to Gene Regulatory Networks.", Nantes, 6 novembre 2014.
— Séminaire de statistique du GREMACQ de l'Université Toulouse 1 Capitole "Sparse regression and optimization in high-dimensional framework: application to Gene Regulatory Networks.", Toulouse, 7 octobre 2014.
— Séminaire de statistique de l'Université de Strasbourg "An hybrid convex/greedy algorithm for learning DAG", Strasbourg, 16 juin 2014.
— Colloque Math-Info de l'INRA, poster "An hybrid convex/greedy algorithm for learning DAG", Ecully, 20 mars 2014.

— Groupe de travail de statistiques de l'Université de Nancy "Modélisation et inférence de réseaux biologiques", Nancy, 13 décembre 2013.
— Journées NETBIO "Optimisation convexe pour l'apprentissage de réseaux de régulation de gènes", Paris, 12 septembre 2013.
— Workshop Statistique Mathématique et Applications "Convex optimization for learning Gene Regulatory Network", Fréjus, 3 septembre 2013.
— Séminaire MIA-T de l'INRA de Toulouse "Convex optimization for learning Gene Regulatory Network", Toulouse, 14 juin 2013.
— Séminaire de probabilités et statistique de Montpellier "Résultats sur les algorithmes de $\mathbb{L}_2$-Boosting pour les régressions sparses : cadre formel et extensions à la situation multivariée", Montpellier SupAgro-INRA, 4 février 2013.

Bibliography

[ADH09] S. Anjum, A. Doucet, and C. C. Holmes. A boosting approach to structure learning of graphs with and without prior knowledge. *Bioinformatics*, 25(22):2929–2936, 2009.

[Aka74] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, AC-19:716–723, 1974. System identification and time-series analysis.

[All71] D. M. Allen. Mean square error of prediction as a criterion for selecting variables. *Technometrics*, pages 469–475, 1971.

[AM02] C. Ambroise and G. J. McLachlan. Selection bias in gene extraction on the basis of mi- croarray gene-expression data. *Proceedings of the National Academy of Science*, 99:6562–6566, 2002.

[AMP97] S. A. Andersson, D. Madigan, and M. D. Perlman. A characterization of markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997.

[Bac08] F. Bach. Bolasso: model consistent lasso estimation through the bootstrap. In *Proceedings of the Twenty-fifth International Conference on Machine Learning*, pages 33–40, Helsinki, Finland, 2008. ACM.

[BB94] H. H. Bauschke and J. M. Borwein. Dykstra's alternating projection algorithm for two sets. *Journal of Approximation Theory*, 79(3):418–443, 1994.

[BCT11] J. D. Blanchard, C. Cartis, and J. Tanner. Compressed sensing: how sharp is the restricted isometry property? *SIAM Review*, 53(1):105–125, 2011.

[BdB07] M. Bansal and D. di Bernardo. Inference of gene networks from temporal gene expression profiles. *IET Systems Biology*, 1(5):306–312, 2007.

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

[BFOS84] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and regression trees*. Wadsworth Statistics/Probability Series. Wadsworth Advanced Books and Software, Belmont, CA, 1984.

[Bis06] C. M. Bishop. *Pattern recognition and machine learning*. Information Science and Statistics. Springer, New York, 2006.

[Bla09] G. Blatman. *Adaptive sparse polynomial chaos expansions for uncertainty propagation and sensitivity analysis*. PhD thesis, Université BLAISE PASCAL - Clermont II, 2009.

[BM07] L. Birgé and P. Massart. Minimal penalties for Gaussian model selection. *Probability Theory and Related Fields*, 138(1-2):33–73, 2007.

[Bre83] H. Brezis. *Analyse fonctionnelle*. Collection Mathématiques Appliquées pour la Maîtrise. Masson, Paris, 1983. Théorie et applications.

[Bre95] L. Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, 1995.

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

[BRT09] P. J. Bickel, Y. Ritov, and A. B. Tsybakov. Simultaneous analysis of lasso and Dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

[Büh06] P. Bühlmann. Boosting for high-dimensional linear models. *The Annals of Statistics*, 34(2):559–583, 2006.

[Büh13] P. Bühlmann. Causal statistical inference in high dimensions. *Mathematical Methods of Operations Research*, 77(3):357–370, 2013.

[Bul98] P. S. Bullen. *A dictionary of inequalities*, volume 97 of *Pitman Monographs and Surveys in Pure and Applied Mathematics*. Longman, Harlow, 1998.

[BY03] P. Bühlmann and B. Yu. Boosting with the L_2 loss: regression and classification. *Journal of the American Statistical Association*, 98(462):324–339, 2003.

[BY10] P. Bühlmann and B. Yu. Boosting. *Wiley Interdisciplinary Reviews: Computational Statistics 2*, pages 69–74, 2010.

[CBGB04] S. L. Carter, C. M. Brechbühler, M. Griffin, and A. T. Bond. Gene co-expression network topology provides a framework for molecular characterization of cellular state. *Bioinformatics*, 20:2242–2250, 2004.

[CCAGV14] M. Champion, C. Cierco-Ayrolles, S. Gadat, and M. Vignes. Sparse regression and support recovery with $\mathbb{L}_2$-boosting algorithms. *Journal of Statistical Planning and Inference*, To appear, 2014.

[CCGP14] M. Champion, G. Chastaing, S. Gadat, and C. Prieur. $\mathbb{L}_2$-boosting on generalized hoeffding decomposition for dependent variables - application to sensitivity analysis. *Statistica Sinica*, To appear, 2014.

[Cer96] R. Cerf. A new genetic algorithm. *The Annals of Applied Probability*, 6(3):778–817, 1996.

[Cer98] R. Cerf. Asymptotic convergence of genetic algorithms. *Advances in Applied Probability*, 30(2):521–550, 1998.

[CGP12] G. Chastaing, F. Gamboa, and C. Prieur. Generalized Hoeffding-Sobol decomposition for dependent variables—application to sensitivity analysis. *Electronic Journal of Statistics*, 6:2420–2448, 2012.

[CGP72] G. Chastaing, F. Gamboa, and C. Prieur. Generalized sobol sensitivity indices for dependent variables: Numerical methods, 2013, Available at `http://arxiv.org/abs/1303.4372`,.

[CH05] L. Cavalier and N. W. Hengartner. Adaptive estimation for inverse problems with noisy operators. *Inverse Problems*, 21(4):1345–1361, 2005.

[Chi02] D. M. Chickering. Optimal structure identification with greedy search. *Journal Machine Learning Research*, 3:507–554, 2002.

[CHM04] D. M. Chickering, D. Heckerman, and C. Meek. Large-sample learning of bayesian networks is np-hard. *Journal Machine Learning Research*, 5:1287–1330, 2004.

[CIBN05] D. G. Cacuci, M. Ionescu-Bujor, and I. M. Navon. *Sensitivity and uncertainty analysis. Vol. II.* Chapman & Hall/CRC, Boca Raton, FL, 2005. Applications to large-scale systems.

[CJ11] T. T. Cai and T. Jiang. Limiting laws of coherence of random matrices with applications to testing covariance structure and construction of compressed sensing matrices. *The Annals of Statistics*, 39(3):1496–1525, 2011.

[CLMM09] T. Crestaux, O. Le Maître, and J. M. Martinez. Polynomial chaos expansion for sensitivity analysis. *Reliability Engineering & System Safety*, 94(7):1161–1172, 2009.

[CLRS09] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.

[CT84] I. Csiszár and G. Tusnády. Information geometry and alternating minimization procedures. *Statistics & Decisions*, (suppl. 1):205–237, 1984. Recent results in estimation theory and related topics.

[CT05] E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[CT07] E. Candes and T. Tao. The Dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

[CW11] T. T. Cai and L. Wang. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information Theory*, 57(7):4680–4688, 2011.

[Dav91] L. Davis. *Handbook of genetic algorithms*, volume 115. Van Nostrand Reinhold New York, 1991.

[DET07] D. L. Donoho, M. Elad, and V. N. Temlyakov. On Lebesgue-type inequalities for greedy approximation. *Journal of Approximation Theory*, 147(2):185–195, 2007.

[DeV98] R. A. DeVore. Nonlinear approximation. *Acta Numerica*, 7:51–150, 1998.

[dlFBHM04] A. de la Fuente, N. Bing, I. Hoeschele, and P. Mendes. Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, 20:3565–3574, 2004.

[DLS00] P. D'Haeseleer, S. Liang, and R. Somogyi. Genetic network inference: From co-expression clustering to reverse engineering. *Bioinformatics*, 16:707–726, 2000.

[DRE] Dream project. Organizers: Columbia university and IBM. Available: http//wiki.c2b2.columbia.edu/dream/index.php/The_DREAM_Project.

[DS90] K. A. DeJong and W. M. Spears. An analysis of the interacting roles of population size and crossover in genetic algorithms. In *Proceedings of the First Workshop Parallel Problem Solving from Nature*, pages 38–47, Berlin, 1990. Springer-Verlag.

[DT96] R. A. DeVore and V. N. Temlyakov. Some remarks on greedy algorithms. *Advances in Computational Mathematics*, 5(2-3):173–187, 1996.

[DVWG09] S. Da Veiga, F. Wahl, and F. Gamboa. Local polynomial estimation for sensitivity analysis on models with correlated inputs. *Technometrics*, 51(4):452–463, 2009.

[Dyk83] R. L. Dykstra. An algorithm for restricted least squares regression. *Journal of the American Statistical Association*, 78(384):837–842, 1983.

[EHJT04] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

[ER10] Y. C. Eldar and H. Rauhut. Average case analysis of multichannel sparse recovery using convex relaxation. *IEEE Transactions on Information Theory*, 56(1):505–519, 2010.

[Faw06] T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[FHHT07] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[FLNP00a] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian networks to analyze expression data. In *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology*, RECOMB '00, pages 127–135, New York, NY, USA, 2000. ACM.

[FLNP00b] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian networks to analyze expression data. In *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology*, RECOMB '00, pages 127–135, New York, NY, USA, 2000. ACM.

[FNP99] N. Friedman, I. Nachman, and D. Pe'er. Learning bayesian network structure from massive datasets: The sparse candidate algorithm. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 206–215, Stockholm, Sweden, 1999.

[Fre90] Y. Freund. Boosting a weak learning algorithm by majority. In *Proceedings of the 3rd annual workshop on Computational Learning Theory*, pages 202–216, 1990.

[Fri01] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

[Fri04] N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303:799–805, 2004.

[FS81] J. H. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of the American Statistical Association*, 76:817–823, 1981.

[Fu98] W. J. Fu. Penalized regressions: the bridge versus the lasso. *Journal of computational and graphical statistics*, 7(3):397–416, 1998.

[Gad08] S. Gadat. Jump diffusion over feature space for object recognition. *SIAM Journal on Control and Optimization*, 47(2):904–935, 2008.

[GE03] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal Machine Learning Research*, 3:1157–1182, 2003.

[GGRVG85] J. Grefenstette, R. Gopal, B. Rosmaita, and D. Van Gucht. Genetic algorithms for the traveling salesman problem. In *Proceedings of the first International Conference on Genetic Algorithms and their Applications*, pages 160–168. Lawrence Erlbaum, New Jersey (160-168), 1985.

[Gha99] B. Ghattas. *Agrégation d'arbres de classification.* Documents de travail GREQAM. GREQAM, 1999.

[GN08] R. Gribonval and M. Nielsen. Beyond sparsity: recovering structured representations by l^1 minimization and greedy algorithms. *Advances in Computational Mathematics*, 28(1):23–41, 2008.

[Gol89] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.

[GWBV02] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

[HB12] A. Hauser and P. Bühlmann. Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 13:2409–2464, 2012.

[HC95] D. Heckerman and D Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

[HJ85] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, Cambridge, 1985.

[HK70] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[Hoc83] R. R. Hocking. Developments in linear regression methodology: 1959–1982. *Technometrics*, 25(3):219–249, 1983.

[Hoe48] W. Hoeffding. A class of statistics with asymptotically normal distribution. *Annals of Mathematical Statistics*, 19:293–325, 1948.

[Hol92] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1992.

[Hoo07] G. Hooker. Generalized functional ANOVA diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics*, 16(3):709–732, 2007.

[HTF09] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer Series in Statistics. Springer, New York, second edition, 2009. Data mining, inference, and prediction.

[HTIWG10] V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts. Inferring regulatory networks from expression data using tree-based methods. *PLoS ONE*, 5(9):e12776, 2010.

[Hua98] J. Z. Huang. Projection estimation in multiple regression with application to functional ANOVA models. *The Annals of Statistics*, 26(1):242–272, 1998.

[HUL93] J. B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms. II*, volume 306 of *Fundamental Principles of Mathematical Sciences*. Springer-Verlag, Berlin, 1993.

[JLD06] J. Jacques, C. Lavergne, and N. Devictor. Sensitivity analysis in presence of model uncertainty and correlated inputs. *Reliability Engineering and System Safety*, (91):1126–1134, 2006.

[KB07] M. Kalisch and P. Bühlmann. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research*, 8:613–636, 2007.

[KF09] D. Koller and N. Friedman. *Probabilistic graphical models*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2009. Principles and techniques.

[Kir84] S. Kirkpatrick. Optimization by simulated annealing: quantitative studies. *Journal of Statistical Physics*, 34(5-6):975–986, 1984.

[KT51] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley, USA, 1951. ACM.

[Lau96] S. L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

[LB06] R. W. Lutz and P. Bühlmann. Boosting for high-multivariate responses in high-dimensional linear regression. *Statistica Sinica*, 16(2):471–494, 2006.

[LBD+10] S. Lèbre, J. Becq, F. Devaux, M. PH. Stumpf, and G. Lelandais. Statistical inference of the time-varying structure of gene-regulation networks. *BMC Systems Biology*, 4(130), 2010.

[LJSB02] S. Lacoste-Julien, M. W. Schmidt, and F. Bach. A simpler approach to obtaining an o(1/t) convergence rate for the projected stochastic subgradient method, 2012, Available at `http://arxiv.org/abs/1212.2002`,.

[LM07] H. Liu and H. Motoda. *Computational Methods of Feature Selection (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series)*. Chapman & Hall/CRC, 2007.

[LPvdGT11] K. Lounici, M. Pontil, S. van de Geer, and A. B. Tsybakov. Oracle inequalities and optimal inference under group sparsity. *The Annals of Statistics*, 39(4):2164–2204, 2011.

[LR10] G. Li and H. Rabitz. D-morph regression: application to modeling with unknown parameters more than observation data. *Journal of mathematical chemistry*, 48(4):1010–1035, 2010.

[LRY+10] G. Li, H. Rabitz, P. E. Yelvington, O. O. Oluwole, F. Bacon, C. E. Kolb, and J. Schoendorf. Global sensitivity analysis with independent and/or correlated inputs. *Journal of Physical Chemistry A*, 114:6022–6032, 2010.

[LW94] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[Mal73] C. L. Mallows. Some comments on c_p. *Technometrics*, 15(4):661–675, 1973.

[Mas07] P. Massart. *Concentration inequalities and model selection*, volume 1896 of *Lecture Notes in Mathematics*. Springer, Berlin, 2007. Lectures from the 33rd Summer School on Probability Theory held in Saint-Flour, July 6–23, 2003.

[MB06] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.

[MDA04] G. J. McLachlan, K. A. Do, and C. Ambroise. *Analyzing microarray gene expression data*. Wiley, 2004.

[Mic94] Z. Michalewicz. *Genetic algorithms + data structures = evolution programs*. Springer-Verlag, Berlin, second edition, 1994.

[MRY07] N. Meinshausen, G. Rocha, and B. Yu. A tale of three cousins: Lasso, L_2Boosting and Dantzig. Discussion: "The Dantzig selector: statistical estimation when p is much larger than n". *The Annals of Statistics*, 35(6):2373–2384, 2007.

[MSMF09] D. Marbach, T. Schaffter, C. Mattiussi, and D. Floreano. Generating realistic in silico gene networks for performance assesment of reverse engineering methods. *Journal of Computational Biology*, 16(2):229–239, 2009.

[MT12] T. Mara and S. Tarantola. Variance-based sensitivity analysis of computer models with dependent inputs. *Reliability Engineering and System Safety*, 107:115–121, 2012.

[MZ93] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[MZKS13] M. Munoz Zuniga, S. Kucherenko, and N. Shah. Metamodelling with independent and dependent inputs. *Computer Physics Communications*, 184(6):1570–1580, 2013.

[Nes04] Y. Nesterov. *Introductory lectures on convex optimization*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, Boston, MA, 2004.

[NY83] A. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization*. A Wiley-Interscience Publication. John Wiley & Sons, Inc., New York, 1983.

[OM12] C. J. Oates and S. Mukherjee. Network inference and biological dynamics. *The Annals of Applied Statistics*, 6(3):1209–1235, 2012.

[OPT99] M. R. Osborne, B. Presnell, and B. A. Turlach. On the lasso and its dual. *Journal of Computational and Graphical Statistics*, 9:319–337, 1999.

[OWJ11] G. Obozinski, M. J. Wainwright, and M. I. Jordan. Support union recovery in high-dimensional multivariate regression. *The Annals of Statistics*, 39(1):1–47, 2011.

[Pap94] C. M. Papadimitriou. *Computational complexity*. Addison-Wesley, Reading, Massachusetts, 1994.

[PB14] J. Peters and P. Bühlmann. Identifiability of gaussian structural equation models with equal error variances. *Biometrika*, (101):219–228, 2014.

[Pea78] J. Pearl. On the connection between the complexity and credibility of inferred models. *International Journal of General Systems*, 4:255–264, 1978.

[Pea00] J. Pearl. *Causality*. Cambridge University Press, Cambridge, 2000. Models, reasoning, and inference.

[PMJS11] J. Peters, J. M. Mooij, D. Janzing, and B. Schölkopf. Identifiability of causal graphs using functional models. In *27th Conference on Uncertainty in Artificial Intelligence (UAI 2011)*, pages 589–598, Corvallis, Oregon, 2011. AUAI Press.

[Pol87] B. T. Polyak. *Introduction to optimization*. Translations Series in Mathematics and Engineering. Optimization Software Inc. Publications Division, New York, 1987.

[PP12] K. B. Petersen and M. S. Pedersen. The matrix cookbook, nov 2012. Version 20121115.

[PS82] C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Prentice-Hall Inc., Englewood Cliffs, N.J., 1982.

[PSHdlF11] A. Pinna, N. Soranzo, I. Hoeschele, and A. de la Fuente. Simulating system genetics data with sysgensim. *Bioinformatics*, 27(17):2459–2462, 2011.

[PZB+10] J. Peng, J. Zhu, A. Bergamaschi, W. Han, D. Y. Noh, J. R. Pollack, and P. Wang. Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer. *The Annals of Applied Statistics*, 4(1):53–77, 2010.

[RASS99] H. Rabitz, O.F. Aliş, J. Shorter, and K. Shim. Efficient input-output model representations. *Computer Physics Communications*, 117(1):11–20, 1999.

[Rid99] G. Ridgeway. *Generalization of boosting algorithms and applications of Bayesian inference for massive datasets*. PhD thesis, University of Washington, 1999.

[RJFD10] A. Rau, F. Jaffrézic, J. L. Foulley, and R. W. Doerge. An empirical Bayesian method for estimating biological networks from temporal microarray data. *Statistical Applications in Genetics and Molecular Biology*, 9:Art. 9, 28, 2010.

[RJN13] A. Rau, F. Jaffrézic, and G. Nuel. Joint estimation of causal effects from observational and intervention gene expression data. *BMC Systems Biology*, 7:111, 2013.

[RKA06] J. J. Rodríguez, L. I. Kuncheva, and C. J. Alonso. Rotation forest: A new classifier ensemble method. *IEEE Transaction Pattern Analysis and Machine Intelligence*, 28(10):1619–1630, 2006.

[RT11] P. Rigollet and A. Tsybakov. Exponential screening and optimal rates of sparse estimation. *The Annals of Statistics*, 39(2):731–771, 2011.

[SAB12] M. Solnon, S. Arlot, and F. Bach. Multi-task regression using minimal penalties. *Journal of Machine Learning Research*, 13:2773–2812, 2012.

[SCED89] J. D. Schaffer, R. Caruana, L. J. Eshelman, and R. Das. A study of control parameters affecting online performance of genetic algorithms for function optimization. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 51–60, San Mateo, CA, 1989. Morgan Kaufman Publishers.

[Sch78] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[Sch90] R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[Sch99] R. E. Schapire. Theoretical views of boosting. In *Computational learning theory (Nordkirchen, 1999)*, volume 1572 of *Lecture Notes in Computer Sciences*, pages 1–10. Springer, Berlin, 1999.

[SCS00] A. Saltelli, K. Chan, and E. M. Scott. *Sensitivity Analysis*. Wiley, West Sussex, 2000.

[SDLC93] D. J. Spiegelhalter, A. P. Dawid, S. L. Lauritzen, and R. G. Cowell. Bayesian analysis in expert systems. *Statistical Science*, 8(3):219–283, 1993.

[SF96] R. E. Schapire and Y. Freund. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156, San Francisco, 1996. Morgan Kaufman.

[SG91] P. Spirtes and C. Glymour. A fast algorithm for discovering sparse causal graphs. *Social Science Computer Review*, 9:62–72, 1991.

[SGS00] P. Spirtes, C. Glymour, and R. Scheines. *Causation, prediction, and search*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, second edition, 2000.

[SM06] T. Silander and T. Myllym"aki. A simple approach for finding the globally optimal bayesian network structure. In *Proceedings of the Twenty-second Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 445–452, 2006.

[SM10] A. Shojaie and G. Michailidis. Penalized likelihood methods for estimation of sparse high-dimensional directed acyclic graphs. *Biometrika*, 97(3):519–538, 2010.

[SMC07] G. Stolovitzky, D. Monroe, and A. Califano. *Dialogue on Reverse-Engineering Assessment and Methods: The DREAM of High-Throughput Pathway Inference*. Number 1115 in Annals of the New York Academy of Sciences. Stolovitzky G and Califano Editions, 2007.

[SMF11] T. Schaffter, D. Marbach, and D. Floreano. Genenetweaver: In silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, 27(16):2263–70, 2011.

[Sob67] I. M. Sobol′. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7, 1967.

[Sob93] I. M. Sobol′. Sensitivity estimates for nonlinear mathematical models. *Mathematical Modeling and Computational Experiment*, 1(4):407–414, 1993.

[Sob01] I. M. Sobol′. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55(1-3):271–280, 2001.

[ST07] T. Similä and J. Tikka. Input selection and shrinkage in multiresponse linear regression. *Computational Statistics & Data Analysis*, 52(1):406–422, 2007.

[Sto94] C. J. Stone. The use of polynomial splines and their tensor products in multivariate function estimation. *The Annals of Statistics*, 22(1):118–184, 1994.

[Tak03] P. M. Takouda. *Problèmes d'approximation matricielle linéaires coniques : Approches par projections et via Optimisation sous contraintes de semidéfinie positivité.* PhD thesis, Université Paul Sabatier - Toulouse III, 2003.

[Tem00] V. N. Temlyakov. Weak greedy algorithms. *Advances in Computational Mathematics*, 12(2-3):213–227, 2000.

[Temv1] V. N. Temlyakov. Greedy approximation in convex optimization, 2012, Available at http://arxiv.org/abs/1206.0392v1.

[Tho73] R. Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42(3):563–585, 1973.

[Tib96] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288, 1996.

[Tro04] J. A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

[Tro12] J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

[TZ11] V. N. Temlyakov and P. Zheltov. On performance of greedy algorithms. *Journal of Approximation Theory*, 163(9):1134–1145, 2011.

[Vap95] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag ,Inc., New York, 1995.

[Vap98] V. N. Vapnik. *Statistical learning theory.* Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons, Inc., New York, 1998.

[vdGB13] S. van de Geer and P. Bühlmann. ℓ_0-penalized maximum likelihood for sparse directed acyclic graphs. *The Annals of Statistics*, 41(2):536–567, 2013.

[vdGBZ11] S. van de Geer, P. Bühlmann, and S. Zhou. The adaptive and the thresholded lasso for potentially misspecified models (and a lower bound for the lasso). *Electronic Journal of Statistics*, 5:688–749, 2011.

[Ver12a] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. In *Compressed sensing*, pages 210–268. Cambridge University Press, 2012.

[Ver12b] N. Verzelen. Minimax risks for sparse regressions: ultra-high dimensional phenomenons. *Electronic Journal of Statistics*, 6:38–90, 2012.

[vM13] R. von Mises. Mechanik der festen körper im plastisch deformablen zustand. *Göttin. Nachr. Math. Phys.*, 1:582–592, 1913.

[VMV+12] J. Vandel, B. Mangin, M. Vignes, D. Leroux, O. Loudet, M. L. Martin-Magniette, and S. de Givry. Inférence de réseaux de régulation de gènes au travers de scores étendus dans les réseaux bayésiens. *Revue d'Intelligence Artificielle*, pages 679–708, 2012.

[vN50] J. von Neumann. *Functional Operators. II. The Geometry of Orthogonal Spaces.* Annals of Mathematics Studies, no. 22. Princeton University Press, 1950.

[Vov90] V. Vovk. Aggregating strategies. In *Proceedings of the 3rd annual workshop on Computational Learning Theory*, pages 372–383, 1990.

[VP91] T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the 6th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 1991.

[VVA+11] M. Vignes, J. Vandel, D. Allouche, N. Ramadan-Alban, C. Cierco-Ayrolles, T. Schiex, B. Mangin, and S. de Givry. Gene regulatory network reconstruction using Bayesian networks, the Dantzig selector, the lasso and their meta-analysis. *PLoS ONE*, 6(12), 2011.

[Wai09] M. J. Wainwright. Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting. *IEEE Transactions on Information Theory*, 55(12):5728–5741, 2009.

[Wei08] P. Weiss. *Algorithmes rapides d'optimisation convexe. Application à la reconstruction d'images et à la détection de changements.* PhD thesis, INRIA Sophia Antipolis, 2008.

[WF03] X. Wang and K. T. Fang. The effective dimension and quasi-Monte Carlo integration. *Journal of Complexity*, 19(2):101–124, 2003.

[YL06] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, 68(1):49–67, 2006.

[ZH05] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, 67(2):301–320, 2005.

[Zha09] T. Zhang. On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research*, 10:555–568, 2009.

[Zha11] T. Zhang. Adaptive forward-backward greedy algorithm for learning sparse representations. *IEEE Transactions on Information Theory*, 57(7):4689–4708, 2011.

[ZY06] P. Zhao and B. Yu. On model selection consistency of Lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.