Apprentissage statistique en gestion de portefeuille

Ruocong Zhang

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2014-ENST-0049

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Signal et Image »

présentée et soutenue publiquement par

Ruocong ZHANG

le 23 septembre 2014

Apprentissage statistique en gestion de portefeuille

Directeur de thèse : **Stéphan CLÉMENÇON**
Co-encadrement de la thèse : **Nicolas VAYATIS**

Jury

M. Frédéric ABERGEL, Professeur, LMAS, École Centrale de Paris	Examinateur
M. Nicolas BASKIOTIS, Maître de Conférences, LIP6, Université Pierre et Marie Curie	Examinateur
M. Patrice BERTAIL, Professeur, MODAL'X UFR SEGMI, Université Paris X	Rapporteur
M. Balázs KÉGL, Directeur de Recherche, LAL, Université Paris Sud	Rapporteur
M. Yann AIT-MOKHTAR, Chef de la Recherche Quantitative, Exane BNP Paribas	Invité
M. Stéphan CLÉMENÇON, Professeur, LTCI, Télécom ParisTech	Directeur de thèse
M. Nicolas VAYATIS, Professeur, CMLA, ENS de Cachan	Directeur de thèse

TELECOM ParisTech
école de l'Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

THÈSE

Table des matières

Résumé

Les travaux présentés dans cette thèse portent sur la prédiction des rendements d'actifs financiers par des méthodes d'apprentissage statistique, motivée par le problème de sélection de titres en gestion de portefeuille. En particulier, nous nous intéressons à la prédiction du signe du rendement à un horizon donné. Nous développons une suite de méthodes traitant le problème depuis la construction d'une base de données d'entraînement jusqu'aux tests de performance. Les contributions principales de ces travaux sont d'une part la mise en œuvre d'un processus complet pourvoyant à toutes les étapes de la prédiction, d'autre part la proposition d'une méthode d'apprentissage multi-tâche pour résoudre simultanément les tâches de prédiction et les relations de dépendance entre elles.

Après avoir introduit le cadre de la gestion de portefeuille et les approches usuelles, nous exposons les enjeux de l'apprentissage statistique en gestion de portefeuille, ainsi que les objectifs opérationnels qui, selon nous, définissent une méthode pertinente pour l'application. Les travaux s'organisent autour de quatre thèmes.

Le traitement de la prédiction commence par la représentation des données de séries temporelles en descripteurs des actifs financiers. Nous présentons des méthodes de représentation susceptibles de produire des descripteurs robustes et interprétables afin de servir d'observations en entrée de la prédiction. La méthode retenue est l'approximation linéaire par morceaux, où la segmentation est construite par un arbre issu des algorithmes de type Coifman-Wickerhauser ou CART.

Une fois obtenue la description des actifs, nous traitons la prédiction du signe des rendements en classification binaire. Nous utilisons des arbres de classification construits par CART et agrégés par Random Forests. Afin de réduire le risque, nous proposons de conditionner la prédiction à des scores de confiance spécifiquement conçus à partir des arbres.

Nous nous intéressons ensuite à l'apprentissage multi-tâche. Nous proposons de considérer les tâches de prédiction comme les sommets d'un graphe, et de pénaliser le risque empirique par le laplacien du graphe. Cette formulation permet de résoudre conjointement les tâches et le graphe de dépendance entre elles.

Un protocole de test par fenêtres glissantes est enfin proposé pour évaluer la performance de la méthode dans les conditions d'utilisation des prédictions. Nous soulignons dans ce cadre l'importance du choix des paramètres d'apprentissage.

Ces méthodes ont abouti à l'émission de recommandations financières publiées en temps réel. Nous concluons par une discussion sur les performances en sélection de titres.

Abstract

The goal of this work is to predict the returns of financial assets with statistical learning methods. We are motivated by the problem of stock selection in portfolio management. In particular, we will focus on the prediction of the sign of future returns at a given horizon. For this purpose, the methods cover the full range of problems from the design of training data to performance tests. The main contributions are both the full process addressing every step and the multitask Laplacian learning formulation that jointly solves for prediction tasks and their dependence structure.

We first introduce the portfolio optimization framework and usually related approaches. We present the challenges for statistical learning and sound objectives for real-world application. The prediction process is made of four modules.

The first module begins with data representation. We need a method for time series representation in order to process raw price data into robust and interpretable features as inputs for the supervised learning problem. The method is chosen to be piecewise linear approximation. Trees, built by the CART or dyadic Coifman-Wickerhauser algorithm, segment time series into homogeneous periods.

The binary classification of returns then uses these features. We perform this learning task with a Random Forests procedure aggregating classification trees built by CART. We propose to add a reject option conditioned by tree-specific confidence scores, in order to reduce generalization risk.

Multitask learning is then explored in the third module. We see dependent prediction tasks as the vertices of a graph, then propose an empirical risk minimization framework with the graph Laplacian as a penalty. Our method jointly solves for the tasks and their graph of relations.

The last module defines the backtesting protocol corresponding to real conditions of use. Performance is assessed in this framework and the importance of parameter validation is stressed.

This work has found concrete application in real-time financial recommendation. We conclude with a discussion on real performance in stock selection.

Remerciements

À l'origine se trouvent les professeurs qui ont su me donner goût à l'apprentissage, en particulier mes directeurs de thèse, Stéphan Clémençon et Nicolas Vayatis. Ils m'ont guidé dans mes travaux, et j'ai eu plaisir à travailler avec eux sur l'application des méthodes scientifiques en un produit concret implanté dans les pratiques opérationnelles. Nicolas m'a initialement orienté de façon déterminante vers un stage intéressant qui s'est poursuivi en cette thèse, et m'a prodigué des conseils avisés en de maintes occasions. Stéphan est celui qui m'a suivi de plus près et encadré avec patience pour me permettre de faire aboutir mes travaux. Pour toutes ces raisons, mes premiers remerciements leur sont adressés. L'autre personne que je tiens à remercier dans le monde académique est Andréas Argyriou, avec qui j'ai pu développer et approfondir un sujet de recherche en particulier. Ses compétences et son expérience ont été déterminantes à l'aboutissement de notre collaboration sous forme de contribution dans des conférences.

Bien entendu, tout cela n'aurait pas été possible sans l'implication d'Exane BNP Paribas. Je suis reconnaissant à Yann Aït-Mokhtar, mon responsable dans l'équipe de Recherche Quantitative, pour avoir soutenu le projet d'application opérationnelle des recherches en apprentissage statistique depuis le début. Je tiens aussi à souligner la contribution importante de Patrick Nielsen Martínez. Par nos échanges réguliers et par son avis d'expert ayant initié le projet, il a considérablement enrichi la recherche par des problématiques pertinentes. Enfin, c'est avec mon ancien collègue Nicolas Bertrand que j'ai travaillé au quotidien. Au-delà de son implication essentielle dans l'application, j'ai largement bénéficié de ses compétences dans mon propre développement professionnel, et j'ai beaucoup appris grâce à lui.

Mes pensées vont aussi aux personnes qui ont contribué indirectement à ma thèse. Bien entendu, ma famille, qui m'a régulièrement exhorté à la terminer, parce que "Bac+10 est un peu excessif". Mais aussi des collègues avec qui je n'ai pas travaillé directement sur ce sujet.
Mes points de vue sur mon travail ont bénéficié des discussions au sein d'Exane, avec d'autres membres de l'équipe comme Christophe et Alexandre, ou des stagiaires comme Charlotte et Ismaïl qui ont travaillé sur des sujets connexes.
À Télécom ParisTech, j'ai reçu l'aide précieuse d'anciens doctorants et post-docs, tels que Nicolas M, Romaric, Sylvain R et Émilie C. Je me souviendrai aussi de moments partagés, en conférence, autour d'une pause thé ou en séminaire des doctorants, avec en particulier Alexandre, Amandine, Andrés, Antoine, Claire, Cristina, Émilie K, Éric, Nicolas S, Olivier et Onur.

Pour terminer, je remercie mes amis pour leur contribution ou leur soutien indéfectible pendant ces dernières années. Je pense notamment à Chen qui s'est reconverti en finance, à Emmanuelle pour son amitié fidèle, à Fei et Pierre qui m'ont encouragé en bien des aspects de la vie, à Gisela et Juwen pour leur foi en mes compétences, à Loïc qui m'a fourni des conditions favorables, à Robin qui a étoffé ma culture musicale et avec qui j'ai pu deviser sur la thèse – et la vie en général, et à Sylvie qui a toujours été disponible pour des discussions variées. Enfin, mes dernières pensées vont à Suqiong, dont la présence a rendu la fin de la thèse moins difficile.

Introduction

Les travaux présentés dans cette thèse ont pour objet l'application de méthodes d'apprentissage statistique à l'optimisation de portefeuille, et plus spécifiquement la prédiction sur les rendements de séries financières pour la sélection d'actifs. Pour chaque actif d'intérêt, notre objectif est de prédire son sens de variation futur à un horizon fixé à partir de l'observation d'un ensemble d'actifs prédicteurs.

Contexte

La sélection des actifs d'un portefeuille est une problématique partagée par un grand nombre de professionnels opérant sur les marchés financiers. Il s'agit de déterminer, parmi les actifs sur lesquels il est possible d'investir, la répartition du capital constituant le meilleur portefeuille possible. Le futur étant incertain, le rendement des actifs doit être considéré comme un vecteur aléatoire, et "meilleur" s'entend généralement en tenant compte à la fois du gain espéré et du risque pris. Cet objectif peut être synthétisé de façon simplifiée dans le cadre de la théorie moderne du portefeuille [Mar52], qui formule le problème de sélection comme une minimisation de la variance estimée du portefeuille sous contrainte de rendement espéré. Cependant, le problème d'optimisation suppose estimés l'espérance du vecteur de rendement des actifs et leur matrice de variance-covariance.

Une grande variété d'approches ont été proposées et pratiquées par les acteurs de marché pour réaliser cette estimation. On peut distinguer trois grands courants méthodologiques : l'analyse fondamentale, l'analyse technique et l'analyse quantitative.

L'analyse fondamentale repose sur l'identification d'une valeur fondamentale des entreprises au moyen d'une estimation des flux de trésorerie générés dans le futur, à partir de leurs données financières, leur stratégie de développement et une connaissance poussée des mécanismes économiques [QF14]. Le principe sous-jacent est la convergence future de leur prix de marché vers la valeur fondamentale.

À l'opposé, l'analyse technique repose uniquement sur l'observation et le traitement du prix des actifs, avec le postulat que le prix reflète toute l'information pertinente pour la prédiction de tendances futures. Les méthodes d'analyse technique ne reposent sur aucun principe théorique concernant les mécanismes économiques, mais se concentrent sur l'identification empirique de motifs ou d'indicateurs caractéristiques permettant de prédire l'évolution future des prix [Sew07].

Les approches d'analyse quantitative font intervenir des outils techniques et des cadres théoriques variés. Entre autres, la modélisation stochastique se place dans un cadre probabiliste rigoureux pour modéliser en temps continu les rendements des actifs au moyen de processus stochastiques. Cette discipline a connu un essor important pour le pricing de produits dérivés. Une autre catégorie d'approches étudie les séries financières sous forme de processus auto-régressifs, tenant compte de la dépendance temporelle entre les prix. Ce type de méthodes a notamment donné le modèle GARCH [Bol86], connu pour l'estimation de la volatilité. Les travaux récents en analyse quantitative tendent à privilégier l'analyse de données réelles afin d'identifier des faits stylisés.

Cette orientation peut s'expliquer par la performance de méthodes non-paramétriques adaptées à la grande dimension des données financières, tout en nécessitant très peu d'hypothèses sur la réalité physique attendue.

Les travaux de cette thèse sont appliqués au sein de l'équipe de Recherche Quantitative d'Exane BNP Paribas. Exane BNP Paribas est une entreprise de courtage en bourse, dont le rôle est l'intermédiation entre les investisseurs et les marchés financiers pour le passage d'ordres d'achat et vente d'actions. Une telle société est constituée de deux grands pôles d'activité : la recherche et l'exécution. La recherche consiste en l'émission de recommandations d'achat ou vente sur les actions à partir de méthodes d'analyse traditionnellement fondamentale. L'exécution reçoit et réalise les ordres passés par les clients, moyennant une commission. L'intérêt d'Exane BNP Paribas pour l'application de l'apprentissage statistique à la prédiction répond à la motivation de proposer à ses clients des recommandations performantes et différenciantes par leur nature objective et systématique.

Enjeux

La prédiction de tendances est un problème si difficile que la possibilité même de prédire est remise en cause par des hypothèses théoriques fortes telles que l'efficience des marchés [Fam70] ou l'absence d'opportunité d'arbitrage. Dans un marché où tous les acteurs sont rationnels, parfaitement informés et avec les mêmes objectifs, toute information est instantanément propagée sur les prix, qui reflètent alors parfaitement l'information, rendant impossible la prédiction des variations futures avec une espérance de gain strictement positive. En pratique, il est avéré que les opportunités d'arbitrage existent, mais sont identifiées et exploitées rapidement. Nous considérons qu'en pratique, il est possible de détecter des arbitrages complexes mettant du temps à être exploités et absorbés par les marchés. L'ambition de notre approche sera donc d'identifier efficacement ces arbitrages avec le moins possible d'hypothèses a priori.

Les difficultés du problème sont liées à la nature des données financières : les prix sont dépendants entre eux et forment une série temporelle. De plus, les variations de prix ne sont pas stationnaires. La grandeur d'intérêt, comme le rendement ou la volatilité, dépendent notamment du contexte économique. S'agissant d'arbitrages complexes, on peut en outre prévoir que la prédiction fera intervenir un grand nombre d'actifs présentant des dépendances complexes entre eux. Il existe de nombreuses approches de traitement des séries temporelles, reposant sur des modélisations de leur structure [Tsa05], une représentation par morceaux [KCHP04], harmonique [Mal00], ou encore l'analyse des valeurs singulières [GNZ01].

En supposant trouvée une bonne méthode de transformation des données brutes en une représentation pertinente des marchés, la non-stationnarité de cette représentation constitue un autre problème lorsque l'on veut y appliquer des algorithmes d'apprentissage statistique. En effet, le cadre classique de l'apprentissage statistique suppose que les données sont indépendantes et identiquement distribuées, afin de garantir des résultats théoriques essentiels tels que la consistance des algorithmes. Des travaux récents, dont [Yu94, KV04, LKS05, MR10a], étendent les résultats théoriques à des processus en général stationnaires sous des conditions de mélange. Plus rare, [SHS09] traite de processus non-stationnaires et montre la consistance des SVM sous certaines conditions. Bien qu'il y ait peu de résultats théoriques dans notre cadre précis, il semble faire sens.

Les méthodes explorées devront aussi répondre à des objectifs opérationnels. Le but est de

concevoir une approche objective portant le moins d'hypothèses possible, par exemple au moyen de méthodes non-paramétriques. Le principe est de permettre à l'utilisateur, expert des marchés financiers, de confronter ses propres conclusions aux résultats des modèles. Il pourra ainsi identifier ses hypothèses implicites et en vérifier la validité. Pour cela, une attention particulière sera accordée à la possibilité d'interpréter les résultats des modèles à tout niveau. Il ne s'agit pas d'une boîte noire en laquelle placer toute confiance, mais bien d'une approche donnant une meilleure compréhension des réalités sous-jacentes aux observations. Enfin, l'ambition est de proposer une approche performante en prédiction et capable de traiter une grande quantité de données de façon exhaustive.

Approche

Nous considérons que l'ensemble de méthodes à proposer n'est pas destiné à remplacer la prise de décision de l'utilisateur, mais à produire une prédiction que l'utilisateur pourra intégrer aux informations à sa disposition. En ce sens, nous n'avons pas pour ambition d'estimer l'espérance de rendement et la matrice de covariance de l'optimisation de portefeuille, mais plutôt de fournir un avis d'expert comme dans le cadre de Black-Litterman [BL92]. Cela dit, les étapes menant à prédiction forment un processus complet (figure 1).

Représentation des données Les données brutes doivent être transformées en données utilisables dans le problème d'apprentissage supervisé. Il s'agit d'une part de représenter les séries de prix pour obtenir une description en entrée, d'autre part de définir la réponse à prédire en sortie. Nous proposons d'utiliser une représentation linéaire par morceaux en segmentant les séries sur des fenêtres de taille fixe, par un algorithme de Coifman-Wickerhauser [CW92] dyadique ou par l'algorithme CART [BFOS84]. Nous supposons ainsi que les séries sont stationnaires par morceaux et qu'elles sont bien décrites par la pente et la variance des résidus sur chaque segment. Cette segmentation appliquée à chaque actif prédictif fournit un vecteur descripteur avec une interprétation directe.
La sortie sera quant à elle plus simple. Nous proposons de prédire le signe du rendement à un horizon fixé. Il s'agira donc d'un problème de classification binaire.

Apprentissage Nous proposons de construire des fonctions prédictives sous la forme d'arbres de classification. Plus précisément, nous utiliserons *Random Forests* [Bre01] pour agréger des arbres de classification appris par CART. Ces arbres permettent de partitionner l'espace d'entrée en régions de classes homogènes de façon adaptée à la taille effective des régions. L'interprétation en termes de règles de décision et de sélection de variables a motivé ce choix. L'analyse de l'importance des variables est la clé de l'interprétation économique des modèles. Nous modifions la méthode originale pour ajouter une option de rejet conditionnée par un score, ou indicateur de confiance, calculé à partir des arbres.

Test et calibration Les modèles de prédiction ne sont pas supposés entraînés et conservés indéfiniment, mais renouvelés selon les changements des marchés. Nous adoptons un protocole de test par fenêtres glissantes correspondant à l'utilisation pratique des prédictions. Il nous permet de voir que le choix des paramètres d'apprentissage est crucial, et de tirer quelques conclusions sur les choix de certains paramètres déterminants. En particulier, l'élagage des arbres est nécessaire dans les méthodes agrégées afin d'éviter le sur-apprentissage. Le suivi des backtests dans le temps permet de déduire les meilleurs jeux de paramètres et de les modifier dans l'étape d'apprentissage.

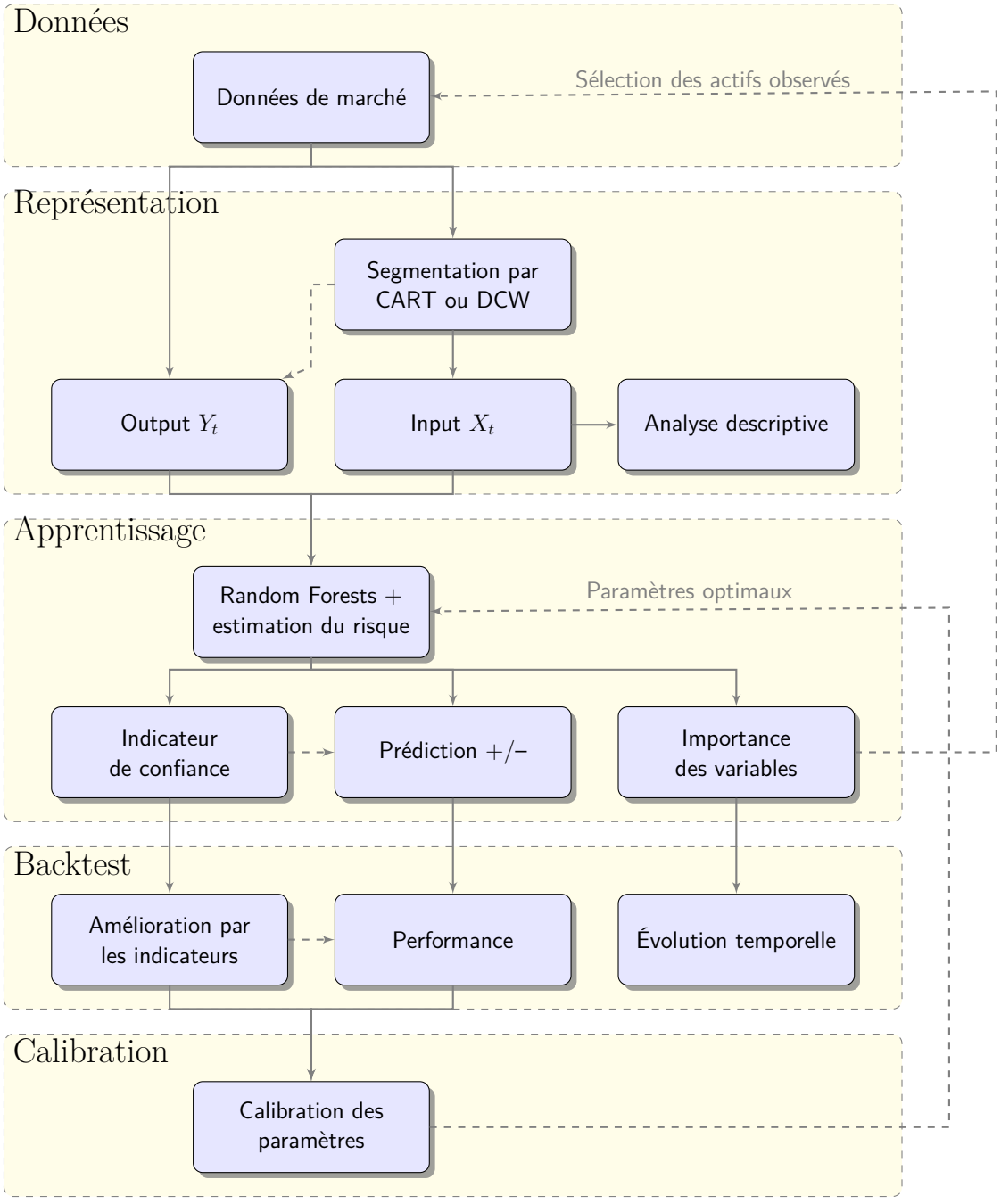


FIGURE 1 – Schéma fonctionnel des travaux

On peut distinguer trois points originaux.

Premièrement, une option de rejet vient modifier la règle de prédiction. Cette modification correspond à la réalité de l'utilisation des prédictions. Le score utilisé pour l'option de rejet doit refléter le risque de chaque prédiction. Il peut tout d'abord être calculé à partir de la similarité entre les nouvelles observations et les données d'entraînement, en supposant que des données très différentes seront potentiellement mal traitées par les arbres. La distance entre les données suit la structure de l'arbre. Ensuite, le risque peut aussi provenir de l'instabilité des résultats prédits face à des petites perturbations des observations.

Le deuxième point est l'exploration du domaine de l'apprentissage multi-tâche. En effet, la

prédiction simultanée des actifs cibles peut être considérée comme un problème unique à sortie multi-dimensionnelle. Ce genre de problème est traité de façon spécifique en tenant compte des dépendances entre les tâches de prédiction. Nous proposons, dans le cadre de la régression par minimisation du risque empirique régularisé, une formulation originale où les tâches appartiennent à un graphe dont le laplacien sert de terme de régularisation. Nous montrons, à l'issue de travaux menés avec Andreas Argyriou et Stéphan Clémençon, qu'il est possible de résoudre simultanément les tâches et l'estimation du graphe avec de bonnes performances. Même si cette approche n'a pas été intégrée à notre procédure de classification, nous sommes convaincus qu'une approche multi-tâche s'avèrerait pertinente.

Enfin, les méthodes proposées ont abouti à une application concrète sous la forme de recommandations financières publiées aux investisseurs. La mise en œuvre du processus de prédiction a permis de se confronter à des problématiques spécifiques, comme la contrainte d'efficacité temporelle ou la production de résultats interprétables.

Plan du manuscrit

Les travaux de cette thèse ont été menés dans le département Traitement du Signal et des Images (TSI) de Télécom ParisTech et au sein de l'équipe de Recherche Quantitative d'Exane BNP Paribas. Ce document détaille les thèmes de recherche et les étapes du processus de prédiction du signe des rendements futurs. La mise en œuvre de ces travaux a abouti à un produit de recommandations financières fondées sur l'apprentissage statistique.

Enjeux de l'apprentissage en gestion de portefeuille Le chapitre 1 présente le cadre de la gestion du portefeuille, qui motive initialement ces travaux. Les variations sur l'optimisation de base sont présentées, ainsi que différentes grandes catégories d'approches pour traiter le problème d'estimation pré-requis. Après avoir rappelé le cadre classique de l'apprentissage statistique, nous identifions les enjeux scientifiques et les objectifs opérationnels de la prédiction. Il s'agira de concevoir une procédure complète depuis la représentation des données jusqu'à la validation des paramètres, avec des problématiques spécifiques à chaque étape.

Représentation des données Le chapitre 2 est la première étape du processus de prédiction. Nous montrons les difficultés découlant de la nature de séries temporelles des données financières et la nécessité de représenter les données brutes de façon pertinente. Après un aperçu de différentes méthodes de traitement, nous choisissons de mettre en œuvre deux méthodes de segmentation par arbres afin de représenter les séries sous la forme de tendances linéaires par morceaux. Nous définissons enfin les entrées et sorties du problème d'apprentissage supervisé.

Apprentissage et prédiction Dans le chapitre 3, nous détaillons les algorithmes au cœur de la prédiction, deuxième étape du processus. La méthode retenue est une agrégation d'arbres de classification construits par l'algorithme CART. L'agrégation est effectuée par *Random Forests*, qui génère des arbres diversifiés par l'échantillonnage aléatoire des données et des variables. Nous proposons de modifier la règle de prédiction en faisant intervenir un score de confiance, fondé soit sur la similarité des données soit sur la stabilité des prédictions. Ce score détermine le degré de confiance accordé à chaque prédiction, permettant à l'utilisateur de rejeter la prédiction si elle est jugée risquée.

Apprentissage multi-tâche Bien que la méthode ne soit pas intégrée à notre processus, le chapitre 4 explore le domaine de l'apprentissage multi-tâche. Les approches multi-tâches sont pertinentes dans les cas où les tâches de prédiction sont dépendantes, ce qui est notre cas. Nous

proposons de considérer les tâches de prédiction comme les sommets d'un graphe, et d'utiliser le laplacien du graphe pour pénaliser le risque empirique. Sur ce principe, nous proposons une formulation originale qui donne lieu à un problème difficile, consistant à trouver simultanément le laplacien du graphe et les tâches de prédiction. Néanmoins, nous parvenons à résoudre le problème au moyen d'une optimisation alternée. Les expériences sur données simulées et réelles montrent une bonne performance.

Résultats expérimentaux Le chapitre 5 définit le protocole de test de la méthode de prédiction. Le test par fenêtres glissantes correspond à une utilisation réaliste des prédictions. Dans ce cadre, nous testons quelques paramètres d'apprentissage pour en déduire les choix à faire pour améliorer la qualité de la prédiction. Nous montrons aussi que les indicateurs de confiance proposés ont des qualités très différentes et que la stabilité est un score intéressant pour améliorer le modèle de base.

Recommandation financière : solution PRISMS Le chapitre 6 présente succinctement l'application des méthodes présentées sous la forme de recommandations financières publiées périodiquement. Les défis sont principalement l'interprétation des modèles et les contraintes d'implémentation. Le document joint en annexe B montre un exemple complet de communication.

Chapitre 1

Enjeux de l'apprentissage en gestion de portefeuille

La composition et la gestion des portefeuilles est un sujet essentiel pour de nombreux acteurs, sinon tous, des marchés financiers. Étant donné un capital détenu, il s'agit de savoir comment distribuer cette quantité entre les actifs disponibles pour réaliser un investissement. Chaque actif présente des caractéristiques différentes, et l'on souhaite constituer le meilleur portefeuille possible. La définition d'un critère d'optimalité du portefeuille est un problème à part entière. En effet, que signifie "meilleur" et quelles propriétés recherche-t-on ? L'investisseur peut vouloir en première approche obtenir le plus grand gain, mais le comportement futur des actifs est incertain, et le gain est au mieux envisageable en espérance. Ensuite, même si l'on maximise l'espérance de gain du portefeuille, l'investisseur est alors exposé au risque de perdre une part de la valeur initiale du portefeuille en cas de variations défavorables des actifs investis. Il peut, par conséquent, préférer réduire sa prise de risque en échange d'un gain espéré plus faible. Enfin, l'investisseur doit se conformer à certaines contraintes telles que les coûts de transaction et des limites de capital investi.

La "théorie moderne du portefeuille" pose un cadre d'optimisation sous contraintes (Markowitz, 1952 [Mar52]) répondant à ces questions pour un investissement statique sur une période unique, lorsque l'espérance et la covariance des rendements sont données. Outre l'optimisation, ces estimations d'espérance et de covariance sont des problèmes essentiels, difficiles à cause de la nature des données, et donnant lieu à de nombreux types d'approches.

Nous verrons dans ce chapitre le cadre classique de l'optimisation de portefeuille, ses adaptations réalistes, les problématiques et les approches usuelles de différents types d'acteurs de marché, afin de situer les enjeux scientifiques et les questions traitées dans la suite.

1.1 Cadre classique de l'optimisation de portefeuille

Dans la théorie classique de l'optimisation de portefeuille, les deux principes de maximisation du rendement et minimisation du risque sont indissociables. Nous verrons, dans le cadre de l'investissement sur une période unique, les définitions du portefeuille, les contraintes et les formulations générales de l'optimisation.

1.1.1 Définitions

Nous introduisons ici les notations dans le cadre d'un investissement statique sur une unique période de temps, où le portefeuille est constitué à l'instant initial et conserve sa composition

jusqu'à l'instant final. Le gérant décide de la composition du portefeuille au début d'une période prédéfinie selon sa fréquence d'investissement (par exemple une semaine, un mois, un an, etc.) d'après ses estimations sur les caractéristiques des actifs disponibles, et ne modifie pas la quantité détenue dans chaque actif jusqu'à la fin de la période. De façon externe à la gestion du portefeuille, la valeur de chaque actif varie dans le temps. Les variations de la valeur du portefeuille entre les instants initial et final sont donc entièrement déterminées par les actifs le constituant. Une grande partie de la littérature démarre de ce cadre élémentaire. Certains travaux traitent de périodes multiples, ou encore de la gestion dite "active" où, de façon réaliste, le gérant peut décider de modifier la composition selon les circonstances au cours de la période.

On considère K actifs de marché ayant chacun un prix dans le temps $S_t^1, \dots, S_t^K$. À l'instant initial $t = 0$, on compose avec le capital C un portefeuille $\mathcal{P}$ en investissant sur les K actifs avec des poids $w = (w^1, \dots, w^K)$, c'est-à-dire qu'on a

$$C = \sum_{k=1}^{K} w^k S_0^k \ ,$$

et que la valeur P_t du portefeuille dans le temps est :

$$P_t = \sum_{k=1}^{K} w^k S_t^k \ .$$

On utilisera de manière plus pratique le rendement R_T^k des actifs entre l'instant initial et l'instant final T, défini par :

$$R_T^k = \frac{S_T^k - S_0^k}{S_0^k} \ .$$

Dans toute la suite de cette partie, on adoptera les simplifications suivantes.

— Les raisonnements se font sur les rendements uniquement.

— Étant sur une unique période, on notera simplement $R = \left(R_T^1, \dots, R_T^K\right)$ le vecteur des rendements d'actifs, sans l'indice T.

— Le capital est unitaire ($C = 1$), et les poids représentent des fractions de capital avec $\sum w^k = 1$. Le rendement du portefeuille est donc $R_{\mathcal{P}} = \sum_{k=1}^{K} w^k R^k = w^\top R$.

1.1.2 Théorie moderne du portefeuille (Markowitz)

L'article *Portfolio Selection* de Harry Markowitz en 1952 [Mar52] est une référence majeure fondant la théorie moderne du portefeuille. On considère néanmoins que les origines remontent à la thèse de Louis Bachelier en 1900, intitulée "Théorie de la Spéculation" [Bac00], et la théorie a été considérablement étoffée et approfondie depuis l'article de Markowitz. Il pose le problème à période unique sous la forme d'une optimisation sous contraintes en considérant conjointement le rendement et le risque du portefeuille, représenté par la variance du rendement. Ce faisant, il rejette la suffisance de la maximisation seule du rendement espéré et montre la nécessité de la diversification, c'est-à-dire la répartition du capital entre des actifs suffisamment décorrélés. L'intuition de la diversification n'est toutefois pas nouvelle à ce moment-là, et se retrouvait déjà dans diverses littératures, comme le souligne Rubinstein dans sa rétrospective de 2002 sur Markowitz [Rub02]. Il cite par exemple Daniel Bernouilli qui, en 1738 à propos du paradoxe de Saint-Pétersbourg, suggérait que l'aversion au risque impliquait la recherche de la diversification ; ou encore, le personnage d'Antonio de la pièce *The Merchant of Venice* de Shakespeare se dit confiant grâce à la répartition de sa fortune sur différents placements.

Problème d'optimisation

On pose les hypothèses suivantes :

1. Efficience du marché [Fam70] : toute information est immédiatement disponible pour tous les acteurs de marché et reflétée par le prix des actifs de marché.
2. Aversion au risque : tout investisseur accorde une valeur au risque, et sera prêt à accepter une diminution du rendement espéré en échange d'un risque réduit.

On considère que les rendements des actifs sont des variables aléatoires dont on connaît (ou a estimé) les moments des deux premiers ordres du vecteur aléatoire R :

— espérance $\mu = \mathbb{E}[R]$,
— covariance $\Sigma = \mathrm{Cov}[R]$, représentant le risque.

Le rendement du portefeuille $R_{\mathcal{P}}$ est par conséquent une variable aléatoire

— d'espérance $\mu_{\mathcal{P}} = \sum_{k=1}^{K} w^k \mu^k = w^\top \mu$ (où μ^k est le k-ième élément de μ),
— de variance $\sigma_{\mathcal{P}}^2 = w^\top \Sigma w$. On appelle aussi volatilité du portefeuille l'écart-type par unité de temps.

Afin de s'assurer de bonnes propriétés, on pose les hypothèses supplémentaires suivantes :

— non-redondance : aucun rendement d'actif ne peut être obtenu par combinaison linéaire du rendement des autres actifs,
— tous les rendements d'actifs ont une variance strictement positive, ce qui signifie que tous les actifs sont risqués. Σ est donc définie positive.

On recherche le vecteur de poids w maximisant le rendement du portefeuille sous contrainte d'objectif de risque v :

$$w^\star = \underset{w}{\operatorname{argmax}}\ w^\top \mu \qquad \text{s.c.} \qquad w^\top \Sigma w = v,\ w^\top \mathbb{1} = 1 \tag{1.1}$$

ou de façon équivalente, minimisant le risque sous contrainte d'objectif de rendement r :

$$w^\star = \underset{w}{\operatorname{argmin}}\ w^\top \Sigma w \qquad \text{s.c.} \qquad w^\top \mu = r,\ w^\top \mathbb{1} = 1 \tag{1.2}$$

où $\mathbb{1}$ est le vecteur de K éléments 1. Un tel portefeuille de poids optimaux $w^\star$ est appelé **portefeuille efficient**. On appelle de plus **frontière efficiente** l'ensemble des optima $(\sigma_{\mathcal{P}}(w^\star), \mu_{\mathcal{P}}(w^\star))$, obtenus en faisant varier soit σ_0 dans 1.1, soit r dans 1.2. Cette frontière est une hyperbole dans le plan (écart-type, moyenne), ou (volatilité, rendement) comme illustré dans la figure 1.1.

On peut résoudre le problème d'optimisation 1.2 de façon analytique par la méthode des multiplicateurs de Lagrange. La solution unique est, comme montrée dans la preuve qui suit,

$$w^\star = \Sigma^{-1} \begin{pmatrix} \mu & \mathbb{1} \end{pmatrix} \left[\begin{pmatrix} \mu & \mathbb{1} \end{pmatrix}^\top \Sigma^{-1} \begin{pmatrix} \mu & \mathbb{1} \end{pmatrix} \right]^{-1} \begin{pmatrix} r \\ 1 \end{pmatrix} . \tag{1.3}$$

De façon similaire, le portefeuille de variance minimale sans contrainte de rendement espéré est caractérisé par les poids $w_{MV}^\star = \Sigma^{-1}\mathbb{1}/(\mathbb{1}^\top \Sigma^{-1}\mathbb{1})$ et la variance $\sigma_{MV}^2 = 1/(\mathbb{1}^\top \Sigma^{-1}\mathbb{1})$.

Démonstration. On pose $A = \begin{pmatrix} \mu & \mathbb{1} \end{pmatrix}^\top$ et $b = \begin{pmatrix} r & 1 \end{pmatrix}^\top$, de sorte que la contrainte se lit $Aw = b$. Soit $\lambda \in \mathbb{R} \times \mathbb{R}$ le vecteur de multiplicateurs de Lagrange. On pose le lagrangien :

$$L(w, \lambda) = \frac{1}{2} w^\top \Sigma w + \lambda^\top (Aw - b) .$$

L'annulation des dérivées de L au premier ordre donne le système d'équations :

$$\begin{cases} \frac{\partial L}{\partial w^\top} = \Sigma w - A^\top \lambda = 0 \\ \frac{\partial L}{\partial \lambda^\top} = Aw - b = 0 \end{cases}$$

qui s'écrit aussi

$$\begin{pmatrix} \Sigma & A^\top \\ A & \mathbf{0} \end{pmatrix} \begin{pmatrix} w \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ b \end{pmatrix} \tag{1.4}$$

Grâce aux hypothèses, Σ est inversible et A est de rang plein. On peut montrer que dans ces conditions, l'inverse de $\begin{pmatrix} \Sigma & A^\top \\ A & \mathbf{0} \end{pmatrix}$ existe et que la forme générale de l'inverse se calcule comme suit ([Gre07, p.824]) :

$$\begin{pmatrix} \Sigma & A^\top \\ A & \mathbf{0} \end{pmatrix}^{-1} = \begin{pmatrix} \Sigma^{-1}(I_K + A^\top F A \Sigma^{-1}) & -\Sigma^{-1} A^\top F \\ -F A \Sigma^{-1} & F \end{pmatrix} \tag{1.5}$$

où I_K est la matrice identité de taille K et $F = -(A\Sigma^{-1}A^\top)^{-1}$. On obtient alors la solution suivante en utilisant 1.5 dans 1.4 :

$$\begin{pmatrix} w \\ \lambda \end{pmatrix} = \begin{pmatrix} \Sigma^{-1}(I_K + A^\top F A \Sigma^{-1}) & -\Sigma^{-1} A^\top F \\ -F A \Sigma^{-1} & F \end{pmatrix} \begin{pmatrix} \mathbf{0} \\ b \end{pmatrix}$$

et en particulier $w = -\Sigma^{-1}A^\top F b = \Sigma^{-1}A^\top (A\Sigma^{-1}A^\top)^{-1} b$.

□

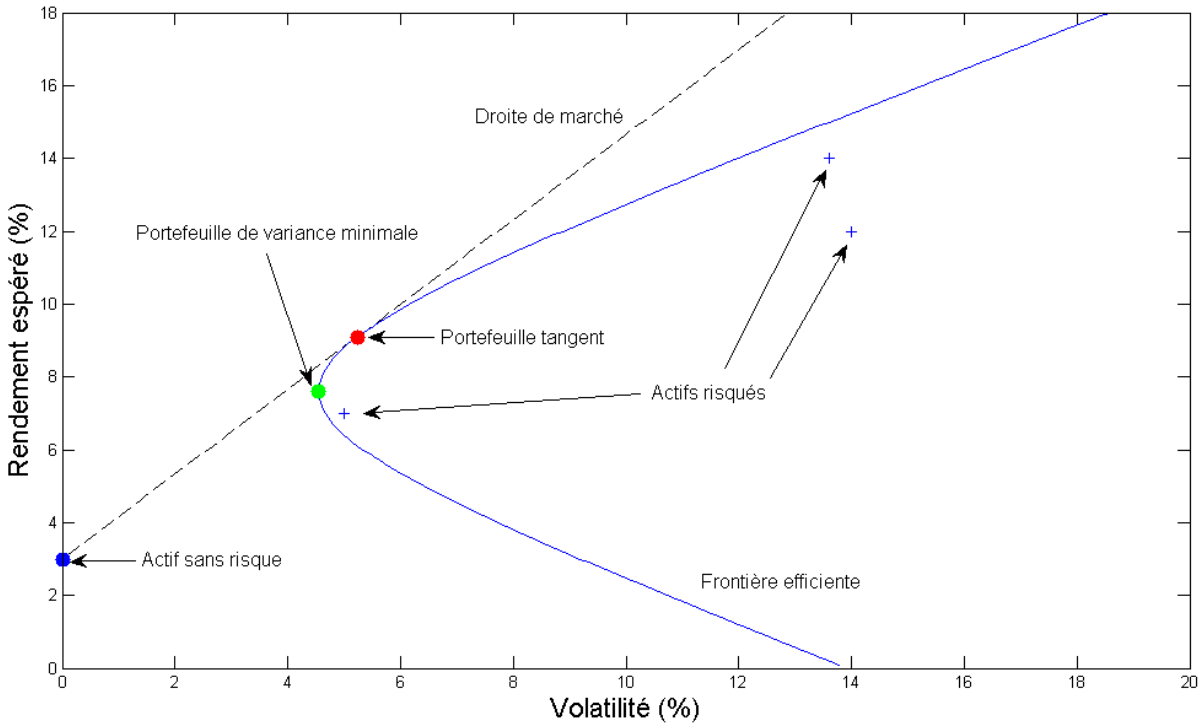


FIGURE 1.1 – Frontière efficiente, portefeuille tangent et droite de marché

Portefeuille tangent

On considère maintenant qu'il existe un actif supplémentaire sans risque, de rendement constant μ_0. L'actif sans risque est un concept théorique qui n'existe pas en pratique. On l'associe

souvent au rendement des bons du trésor de pays développés ayant un marché financier stable, car on considère peu probable qu'ils fassent défaut. Cependant, comme la crise souveraine récente l'a rappelé en Europe, la dette souveraine est loin d'être un actif sans risque.

L'investisseur répartit le capital entre les actifs risqués et cet actif sans risque en attribuant à celui-ci un poids w_0 dans le portefeuille, dont le rendement devient d'espérance $\mu_{\mathcal{P}} = w_0\mu_0 + w^\top \mu$ mais toujours de variance $\sigma^2_{\mathcal{P}} = w^\top \Sigma w$. Le problème d'optimisation 1.2 sur les actifs risqués devient :

$$w^\star = \operatorname{argmin} w^\top \Sigma w \qquad \text{s.c.} \qquad w_0\mu_0 + w^\top \mu = r,\ w_0 + w^\top \mathbb{1} = 1 \tag{1.6}$$

On peut montrer que les poids des actifs risqués s'écrivent alors :

$$w^\star = \frac{\Sigma^{-1}(\mu - \mu_0 \mathbb{1})}{\mathbb{1}^\top \Sigma^{-1}(\mu - \mu_0 \mathbb{1})} \ .$$

Ces poids ne dépendent alors plus de la contrainte de rendement r, mais du rendement μ_0 de l'actif sans risque.

On définit le portefeuille tangent dans le plan (écart-type, moyenne) comme l'intersection entre la frontière efficiente (actifs risqués) et sa tangente passant par le point $(0, \mu_0)$ représentant l'actif sans risque. Tout portefeuille efficient est donc une combinaison linéaire entre l'actif sans risque et le portefeuille tangent, et on appelle la droite passant par ces deux points la **droite de marché** (*Capital Market Line*), illustrée dans la figure 1.1. Tous les investisseurs ayant les mêmes estimations de μ et Σ auront donc des portefeuilles appartenant à cette même droite, répartissant leur capital entre le portefeuille tangent et l'actif sans risque. Il s'agit des théorèmes de *mutual funds* ou de séparation [Tob58] [Mer92, Chap. 2].

Capital Asset Pricing Model (CAPM)

Le CAPM, ou MEDAF (modèle d'évaluation des actifs financiers), est un corollaire direct de la droite de marché, issu des travaux de Sharpe [Sha64], Lintner [Lin65] et Mossin [Mos66]. On se place dans les hypothèses suivantes :

1. le marché est efficient,
2. il n'y a pas de coûts de transaction,
3. les investisseurs raisonnent tous dans le cadre de Markowitz, avec le même objectif d'optimisation statique,
4. les investisseurs disposent de la même information et des mêmes estimations pour les rendements.

Sous ces conditions, un modèle linéaire exprime l'excès de rendement des actifs risqués (par rapport au rendement sans risque) en fonction du rendement R_M d'un portefeuille de marché : pour tout $k \in \{1, \dots, K\}$,

$$R_k - \mu_0 = \beta_k (R_M - \mu_0) + \varepsilon_k$$

avec ε_k un bruit indépendant de R_M. En espérance, on écrit aussi, avec $\mu_M = \mathbb{E}(R_M)$:

$$\mu_k = \mu_0 + \beta_k \left(\mu_M - \mu_0\right) \ .$$

Ce modèle donne une explication des rendements d'un actif par rapport au marché à travers leur corrélation. Les β_k représentent le risque systématique, c'est-à-dire lié au marché et non-diversifié, tandis que les ε_i représentent le risque spécifique, propre aux actifs. Cette notion s'est largement répandue chez les opérationnels, qui raisonnent couramment en termes de "bêta" des actions par rapport à l'indice représentatif du pays ou du secteur d'activité, et interprètent la valeur implicitement comme un niveau de risque.

Entre autres, les travaux de Merton (c.f. [Mer92]) ont mis en évidence depuis les années 1970 l'insuffisance du CAPM ainsi formulé, du fait de sources de risque autres que le marché. L'*Arbitrage Pricing Theory* (APT) de Ross [Ros76] fournit des généralisations en exprimant le rendement espéré de l'actif risqué par un modèle factoriel, sous certaines hypothèses techniques :

$$\mathbb{E}(R^k - \mu_0) = \sum_{m=1}^{M} \beta_{k,m} \mathbb{E}(F_m - \mu_0)$$

où les F_m sont des facteurs de risque systématique, décorrélés deux à deux. La définition des facteurs n'est pas traitée à l'origine, mais constitue une problématique riche. Dans la littérature ont été proposés divers facteurs pour tenir compte de la croissance à long et court terme, des renversements de tendance, des fondamentaux des entreprises, et d'autres ajustements empiriques. Notamment, Fama et French ont publié une série d'articles (dont [FF93]) exhibant deux facteurs, calculés sur les prix de marché des entreprises, qui incorporent une grande partie de ces ajustements avec un R^2 de régression élevé. Depuis les années 90, des produits commerciaux, tels que Barra [1] et SunGard APT [2], fournissent une analyse du risque au moyen de facteurs calculés par analyse en composantes principales ou autres modèles factoriels. De nos jours, l'identification de facteurs de risque systématique est une préoccupation majeure courante pour les gérants de portefeuille et leurs analystes en risque.

1.1.3 Augmentation du problème

La formulation de base est épurée de beaucoup de problématiques réalistes de la gestion de portefeuille. Les éléments suivants sont des considérations courantes qui viennent augmenter le problème initial, souvent d'une manière non-triviale pour la résolution.

Variations sur l'objectif d'optimisation

Afin de considérer la rentabilité des gains par rapport au risque pris, on utilise couramment le **ratio de Sharpe** [Sha66], défini comme l'excès de rendement du portefeuille par rapport au taux sans risque divisé par l'écart-type du rendement :

$$SR(\mathcal{P}, r) = \frac{\mu_{\mathcal{P}} - r}{\sigma_{\mathcal{P}}} = \frac{w^\top \mu - \mu_0}{w^\top \Sigma w} \ .$$

On peut montrer [Cha09, p.297] que le portefeuille tangent est aussi le portefeuille maximisant le ratio de Sharpe.

Selon le type de gestion, la mesure "officielle" de la performance des portefeuilles, c'est-à-dire celle qui détermine la performance des gérants, et donc leur rémunération, peut être différente. Dans le cas des gestions dites classiques ou *long only*, dont l'objectif est de suivre un indice de marché $\mathcal{I}$ (par exemple le CAC 40, l'EURO STOXX 50, etc.) avec une sur-performance, on considère l'excès de rendement par rapport à l'indice, qui est lui-même une combinaison linéaire d'actions représentatives d'un pays ou d'une zone économique. La mesure est alors le **ratio d'information** :

$$IR(\mathcal{P}, \mathcal{I}) = \frac{\mathbb{E}[R_{\mathcal{P}} - R_{\mathcal{I}}]}{\sqrt{\mathrm{Var}[R_{\mathcal{P}} - R_{\mathcal{I}}]}} \ .$$

1. http://www.msci.com/products/barra.html
2. http://financialsystems.sungard.com/solutions/asset-management/apt

Enfin, une idée naturelle en économie consiste à maximiser directement une fonction d'utilité de l'investisseur. La forme générale du problème de maximisation d'utilité peut s'écrire :

$$w^\star = \text{argmax} \int_R U(w^\top R) dP(R) \ .$$

Markowitz propose dans son livre [Mar70] une utilité de la forme $U_\gamma(\mathcal{P}) = R_\mathcal{P} - \frac{\gamma}{2} R_\mathcal{P}^2$, où γ représente l'aversion au risque, qui reflète le niveau de compromis accepté par l'investisseur entre le rendement et le risque. On maximise l'espérance de cette utilité, qui est $\mathbb{E}[U_\gamma(\mathcal{P})] = \mu_\mathcal{P} - \frac{\gamma}{2}\sigma_\mathcal{P}^2$. En utilisant les notations de la partie 1.1.2, le problème d'optimisation est :

$$w^\star = \text{argmax}\, w^\top \mu - \frac{\gamma}{2} w^\top \Sigma w \qquad \text{s.c.} \qquad w^\top \mathbb{1} = 1 \ . \tag{1.7}$$

Il s'agit d'une maximisation du rendement espéré pénalisée par le risque, et de paramètre de régularisation γ. Cette formulation est en réalité équivalente à la formulation de Markowitz initiale 1.2. Pour s'en convaincre, il suffit d'écrire les lagrangiens de chaque équation et les conditions d'optimalité du premier ordre, qui sont identiques, à multiplicateurs de Lagrange près.

Bien entendu, de nombreuses formes d'utilité ont été proposées dans la littérature afin de tenir compte de différentes propriétés désirables, dont on peut trouver une synthèse dans [Meu05, Chap. 5].

Mesures de risque

La formulation de Markowitz utilise la variance des rendements pour représenter le risque, et c'est devenu une pratique courante pour beaucoup d'actifs de marché. Néanmoins, la variance ne rend pas compte de moments supérieurs de la distribution, et ce n'est pas nécessairement la mesure la plus pertinente, notamment si les rendements ne sont pas gaussiens.

Semi-variance La variance étant symétrique, les rendements positifs sont considérés risqués au même titre que les rendements négatifs, alors que l'aversion de l'investisseur porte souvent sur le risque de perte. La semi-variance a été initialement proposée par Markowitz [Mar70]. Pour une variable aléatoire X, on définit la semi-variance σ_{min}^2 par

$$\sigma_{\text{min}}^2 = \mathbb{E}\left[\min(X - \mathbb{E}[X], 0)^2\right] \ .$$

Il n'existe pas de solution analytique en remplaçant directement la variance par la semi-variance dans le problème 1.2. Néanmoins, les auteurs de [JMZ06] montrent l'existence d'une frontière efficiente, et des approximations [Est03] permettent de donner des solutions analytiques satisfaisantes.

Value at Risk (VaR) À la fin des années 1980, la banque JP Morgan développe *Value at Risk (VaR)* et fait connaître le concept par le rapport technique RiskMetrics [3] [JR96]. La VaR mesure le risque du portefeuille au moyen de quantiles de la distribution des rendements du portefeuille. Plus précisément, la VaR de niveau α d'un portefeuille est le quantile de niveau $1-\alpha$:

$$\text{VaR}(\mathcal{P}, \alpha) = \sup_q \{b \in \mathbb{R} : \mathbb{P}(R_\mathcal{P} \leq q) \leq 1 - \alpha\} \ .$$

Par exemple, la VaR de niveau 95% est le seuil de rendement tel qu'au plus 5% des rendements du portefeuille sont inférieurs à ce seuil, pour un horizon de temps donné. Autrement dit, il

3. http://www.msci.com/products/riskmetrics.html

s'agit d'un seuil de confiance tel que 95% des rendements du portefeuille sont supérieurs à cette valeur. La Value at Risk est largement employée comme mesure du risque des portefeuilles, parfois en remplacement de la variance. En particulier, les régulations bancaires comme les accords de Bâle préconisent l'utilisation de la VaR à 10 jours de niveau 99%, et requièrent des établissements bancaires des fonds propres égaux à k fois cette valeur, k étant un nombre fixé. La VaR est considérée plus pertinente que la variance car elle dépend de moments d'ordres supérieurs des rendements. Néanmoins, bien que qu'elle soit calculable sur les rendements historiques pour analyser le risque passé des portefeuilles, son estimation sur les rendements futurs pour l'optimisation de portefeuille est un problème plus difficile étudié dans beaucoup d'aspects (voir par exemple [BS01, FG02, GP05]).

VaR conditionnelle La sous-additivité est une propriété désirable pour les mesures de risque [Meu05, Chap. 5], car cohérente avec la diversification : le risque d'un portefeuille diversifié est au plus la somme des risques de ses constituants. Si on considère deux portefeuilles $\mathcal{P}_1$ et $\mathcal{P}_2$, la sous-additivité de la mesure de risque ρ est définie par $\rho(\mathcal{P}_1 + \mathcal{P}_2) \leq \rho(\mathcal{P}_1) + \rho(\mathcal{P}_2)$. La VaR n'est pas sous-additive (par exemple d'après [BS01]), et des portefeuilles peuvent afficher une VaR plus élevée alors qu'ils sont mieux diversifiés. Afin de compenser ce défaut, on utilise une grandeur liée, appelée VaR conditionnelle, définie comme l'espérance de rendement sous la VaR :

$$\mathrm{CVaR}(\mathcal{P}, \alpha) = \mathbb{E}\left[R_{\mathcal{P}} | R_{\mathcal{P}} < \mathrm{VaR}(\mathcal{P}, \alpha)\right] \ .$$

Cette mesure a notamment étudiée par Artzner [ADEH99], Rockafellar et Uryasev [RU00, Ury00].

Valeurs extrêmes L'estimation du risque est critique pendant les périodes de mouvements importants, comme les changements de modes économiques, les crises et les bulles. Dans ces périodes, la dépendance entre les actifs est significativement différente du comportement en marché stable, car les acteurs de marché eux-mêmes réagissent différemment. On observe entre autres que la corrélation entre les actifs augmente et que les rendements journaliers sont plus volatils. Statistiquement, on peut parler de valeurs extrêmes. Si le risque d'un portefeuille est estimé indistinctement sur tout l'historique, le risque de queue de distribution est sous-évalué par rapport à son importance, car les occurrences sont peu fréquentes. On peut par exemple caractériser la tendance d'une variable aléatoire X_1 à être extrême sachant qu'une seconde variable X_2 est extrême par le coefficient de dépendance extrême, défini comme la limite lorsque α tend vers 1 de la probabilité

$$\chi_{1,2}(\alpha) = \mathbb{P}\{F_1(X_1) > \alpha | F_2(X_2) > \alpha\}$$

avec F_1 et F_2 les fonctions de répartition respectives de X_1 et X_2.

L'étude du risque dans les extrêmes est un domaine riche en problématiques, et on peut se référer à [EKM97, McN99] pour de plus amples détails.

Contraintes

En pratique, la gestion des portefeuilles inclut des contraintes opérationnelles variées, différentes selon le type de gestion, la fréquence de rebalancement du portefeuille, et des caractéristiques propres à l'activité. De telles contraintes modifient significativement le problème d'optimisation et rendent rapidement impossible toute résolution analytique. Dans certains cas, les contraintes donnent lieu à des formulations solubles par des techniques numériques telles que la programmation quadratique, ou semi-définie. Ces contraintes et leur intégration au problème sont exposées par exemple dans [Meu05]. Cette partie fait un inventaire sans résolution des contraintes les plus courantes.

Vente à découvert L'interdiction de vente à découvert, courante pour les gérants de portefeuilles classiques, oblige les gérants à ne pouvoir vendre que les actifs détenus dans le portefeuille. Elle peut aussi être imposée par le régulateur lors d'événements particuliers. Cela se traduit par la positivité des poids du portefeuille : pour tout $k \in \{1, \dots, K\}$, $w^k \geq 0$. Cette contrainte a été souvent considérée, même si sa présence empêche la résolution analytique du problème d'optimisation.

Limites d'exposition L'exposition s'entend en montant investi et en risque actif. Afin de contrôler le capital total investi, la somme des poids peut être encadrée par deux valeurs $W_{\min}$ et $W_{\max}$: $W_{\min} \leq \sum_{k=1}^{K} w_k \leq W_{\max}$. De plus, on peut limiter la concentration du portefeuille sur quelques actifs en posant des bornes sur chacun sous la forme des vecteurs $w_{\min}$ et $w_{\max}$: $w_{\min} \leq w \leq w_{\max}$. De manière générale, un ensemble de N_c contraintes de ce type peut être résumé en une matrice Ω de taille $N_c \times K$ et des vecteurs A et B de taille N_c :

$$A \leq \Omega w \leq B \ .$$

La concentration du portefeuille est aussi représentée par sa dépendance à des facteurs macroéconomiques. Le gérant peut vouloir limiter ses corrélations estimées $\beta = \left(\beta_1, \dots, \beta_{N_f}\right)$ avec N_f facteurs expliquant les variations du marché, afin de ne pas s'exposer excessivement à des mouvements conjoncturels. De manière similaire, cette contrainte d'exposition peut s'écrire au moyen d'un vecteur majorant $\beta_{\max}$: $\beta \leq \beta_{\max}$.

Coûts de transaction et de détention Au-delà de l'investissement statique, le portefeuille est rebalancé périodiquement, c'est-à-dire que sa composition pourra être modifiée au cours du temps. En pratique, les coûts de transaction ont une influence majeure sur la performance des portefeuilles, l'amplitude des rebalancements et les actifs investis. On considère un rebalancement du portefeuille, passant des poids w aux poids w'. Le montant de transaction sur chaque actif est la variation de leur poids $\Delta w = w' - w$. L'achat et la vente d'actifs s'effectuent à un certain coût, différent selon l'actif, le sens (achat/vente), le marché (conjoncture économique, fiscalité, etc.) et le moyen (courtier, mode de passage d'ordre, etc.). Soit $\Delta w^+ = \min(w, 0)$ les montants en achat, $\Delta w^- = \max(-w, 0)$ les montants en vente (en valeur absolue), c^+ les coûts d'achat et c^- les coûts de vente. Le coût du rebalancement peut s'écrire : $c^+\Delta w^+ + c^-\Delta w^-$. Lorsque les coûts sont simples, il est équivalent de considérer le *turnover*, c'est-à-dire les mouvements absolus totaux du portefeuille, et le rapporter à la taille du portefeuille. Ainsi, une gestion de long terme à basse fréquence aura par exemple un turnover équivalent à une fois la taille du portefeuille par an, tandis qu'une gestion plus active pourra rebalancer plusieurs fois la taille du portefeuille par mois.

Outre la transaction, la détention d'un actif en portefeuille a un coût. C'est notamment le cas lorsqu'on emprunte des actifs afin de les vendre alors qu'ils ne sont pas déjà détenus dans le portefeuille, et ces actifs ont alors un poids négatif. S'agissant d'intérêts sur l'emprunt, ce coût dépend du temps de détention t. Si on note c^s ce coût par unité de temps, les coûts totaux à prendre en compte sont :

$$C(w, w') = c^+\Delta w^+ + c^-\Delta w^- + c^s t \min(-w', 0) \ .$$

Cette grandeur peut être intégrée en tant que pénalité dans l'optimisation, avec un paramètre contrôlant la proportion des coûts par rapport à la performance du portefeuille. En pratique, on peut ajuster les stratégies afin d'obtenir un rapport rendement sur coûts satisfaisant (par exemple un rendement brut égal à trois fois les coûts).

Impact de marché Dans la théorie classique, on considère que les prix de marché sont des données externes, indépendantes des décisions prises par les investisseurs. Cependant, les investisseurs font partie des marchés et influent directement sur les prix. Par exemple, si un gérant de portefeuille décide d'acheter une quantité importante des actions d'une entreprise, il modifie la disponibilité des actions et par conséquent provoque de façon probable une augmentation du prix de l'action. Si son achat ne peut s'effectuer en une seule transaction, son prix moyen d'achat sera plus élevé que le prix initial observé. Même s'il ne s'agit pas d'un coût à la transaction, cet impact est particulièrement important dès que le volume des transactions est significative par rapport à la liquidité des actifs (capital géré important), ou lorsque les gains attendus sont à la même échelle (trading algorithmique). La modélisation de l'impact de marché est complexe, et le passage d'ordres de transaction optimal est un sujet actif (voir [AC01, LLP11]).

1.1.4 Discussion

Le problème d'optimisation sous contraintes en soi n'était pas un concept mathématique nouveau, mais c'est l'application systématique de la méthode à la finance qui fut déterminante. En effet, le rendement optimal est inférieur au rendement maximal sans contrainte de risque, et paraît donc sous-optimal aux opérationnels recherchant uniquement le rendement en négligeant le risque. Cette formulation permet donc de définir une approche systématique et de raisonner dans le plan rendement-risque, ou en termes statistiques, moyenne-variance. La théorie moderne du portefeuille a eu une influence significative sur les pratiques de gestion, même si elle présente de nombreux défauts, notamment le manque de réalisme du modèle initial, qui est en réalité difficile à résoudre dès que l'on ajoute des contraintes courantes. On peut souligner les points suivants, pour la plupart signalés dans la partie 1.1.3 :

- Les coûts de transaction sont inévitables et non-négligeables. Le problème d'optimisation est significativement modifié dès que l'on en tient compte.
- La droite de marché et le CAPM supposent que tous les investisseurs ont la même information sur les marchés, les mêmes estimations des rendements et les mêmes objectifs d'optimisation. Or l'information n'est pas connue instantanément pour tous et dépend des conditions techniques. Cet aspect est particulièrement saillant dans le trading à haute fréquence, où la latence d'accès à l'information est critique. De même, il est évident que les estimations et les objectifs des acteurs de marchés sont propres à chaque acteur.
- Les prix sont considérées comme des variables exogènes indépendantes des portefeuilles constitués, mais les transactions influencent les prix. Cet effet devient significatif dès que le capital géré est important.
- Le raisonnement en moyenne-variance suffit si les rendements sont gaussiens, mais ce n'est pas le cas en général.

Enfin, la résolution du problème d'optimisation suppose connus ou estimés les rendements futurs en espérance et covariance. Cette méthode ne traite que de l'étape décisionnelle de la sélection de portefeuille, comme Markowitz le dit lui-même :

> *The process of selecting a portfolio may be divided into two stages. The first stage starts with observation and experience and ends with beliefs about the future performances of available securities. The second stage starts with the relevant beliefs about future performances and ends with the choice of portfolio. This paper is concerned with the second stage.*

L'estimation reste l'étape difficile et critique. Nous verrons dans la partie suivante différents types d'approches employés par les acteurs de marchés.

1.2 Approches usuelles

L'estimation des rendements futurs des actifs financiers est un sujet si vaste et si difficile qu'il existe une immense variété d'approches, sans solution standard prépondérante. De plus, le comportement du marché dépend lui-même des investisseurs, et donc de leur méthode de prise de décision. L'histoire des marchés financiers montre que l'adoption à grande échelle d'une méthode modifie le fonctionnement des marchés et crée des instruments, des risques et des phénomènes nouveaux, invalidant parfois les modèles utilisés auparavant. On distingue ici quelques catégories courantes d'approches, souvent associées à des types d'acteurs de marchés en particulier.

1.2.1 Analyse fondamentale des entreprises

En stratégie financière de l'entreprise (cf. [QF14]), on considère que les entreprises ont une valeur fondamentale, c'est-à-dire qu'il est possible de leur donner un prix à partir des données inhérentes à l'entreprise (chiffre d'affaires, bénéfices, etc.) et de l'estimation de leur valeur future. Si le prix courant de l'action est différent de cette valeur, on prédit que le prix de marché convergera vers la valeur fondamentale et on en déduit le sens de variation futur. Il s'agit du paradigme couramment employé par les analystes financiers émettant des recommandations d'achat et de vente sur les actions, où la valeur fondamentale est annoncée en tant que "prix cible". Il existe en réalité plusieurs méthodes de calcul de la valeur d'une entreprise, et elles ne convergent pas en général. Nous présentons ici la méthode des flux de trésorerie actualisés (DCF - *discounted cash flows*), qui est l'une des plus courantes chez les analystes financiers.

Méthode générale

La valeur d'entreprise (EV - *enterprise value*) correspond aux gains net de l'entreprise rapportés au coût pour l'actionnaire. Elle est essentiellement la série des flux de trésorerie disponible (FCF pour *free cash flows*) de chaque année à venir, actualisés à un taux $r \in [0, 1[$ reflétant la valeur du temps :

$$EV = \sum_{t=0}^{\infty} \frac{FCF_t}{(1+r)^t}$$

où FCF_t est le flux de l'année t. Pour une année d'exercice passée, dont on connaît les comptes de l'entreprise, le flux de trésorerie disponible correspond aux liquidités nettes effectivement dégagées par l'entreprise, et se calcule à partir des grandeurs comptables [QF14] :

$$\begin{aligned} \text{FCF} = \quad & \text{Excédent brut d'exploitation} \\ & \text{- Impôts sur le résultat d'exploitation} \\ & \text{- Variation du besoin en fonds de roulement} \\ & \text{- Investissements} \end{aligned}$$

Les flux futurs FCF_t sont extrapolés d'après le comportement récent de l'entreprise, sa stratégie de développement et la visibilité de son activité. Souvent, des estimations sont réalisées sur quelques années à venir, et la valeur est supposée constante ensuite.

Le taux d'actualisation r représente le coût moyen pondéré du capital, c'est-à-dire le coût implicite de la détention de l'action. De façon équivalente, c'est le rendement moyen attendu par les actionnaires. Soit E le capital propre de l'entreprise, D la valeur de marché de sa dette, k_E le coût des fonds propres, k_D le coût de la dette, et k_I le taux d'imposition. On écrit de façon simplifiée :

$$r = \frac{k_E E + k_D D(1 - k_I)}{E + D} \ .$$

L'élément le plus délicat est k_E, qui est le rendement de l'actif. D'après le CAPM de la partie 1.1.2, on pourrait écrire : $k_E = \mu_0 + \beta(\mu_M - \mu_0)$, avec un taux sans risque μ_0, un rendement de marché μ_M et une corrélation β avec le marché. Cependant, outre les limitations du CAPM, le calcul du coût du capital revient à une estimation d'espérance de rendement futur, et est donc un problème difficile. Entre autres, il faut considérer différentes sources de complexité :

— Comme les autres actifs, il faut tenir compte d'autres facteurs de risque. On résume ces sources par un terme de "prime de risque", dont l'estimation est étudiée dans la littérature (cf Damodaran [Dam]).
— Le coût du capital est propre à chaque marché. Il est donc beaucoup plus complexe dès que l'entreprise est exposée à différentes régions du monde.
— Il varie au cours du temps.

Le calcul de la valeur d'entreprise relève donc largement de l'expertise de chaque analyste.

Discussion

La valeur fondamentale repose non sur les prix mais sur les données comptables publiées par l'entreprise. L'information est donc par définition connue de façon postérieure et la principale source d'anticipation est l'initiative des analystes à réviser leurs estimations. Cette latence crée un certain paradoxe du fait que l'on a une incertitude non seulement sur les données futures, mais aussi sur les données comptables présentes et passées (avant publication). L'utilisation de l'estimation nécessite un cadre spécifique. autre que le problème d'optimisation 1.2. En effet, la recommandation donne un sens (achat/vente, voire nuances) et un prix cible, mais pas d'horizon, et elle est implicitement valable jusqu'au prochain changement de recommandation de la part de l'analyste. De plus, les recommandations des divers analystes sur une entreprise donnée ne sont ni périodiques, ni synchronisées.

La qualité de l'approche fondamentale vient du fait que les analyses menant à la valeur estimée apportent une compréhension concrète des entreprises et de leur prix. Les investisseurs les confrontent à leur propre expertise à la lumière des explications, plutôt qu'en déduire une estimation de rendement. La mesure de la performance prédictive des analystes peut s'avérer complexe, car il est parfois difficile de distinguer la prédiction et son influence sur les données. En effet, la compétence des émetteurs des recommandations est leur capacité à présenter un argumentaire convaincant et à entretenir la confiance des investisseurs dans leur avis. Dès lors, les analystes les plus suivis ont une influence significative sur le comportement de certains acteurs de marchés, ce qui augmente leur probabilité d'avoir raison.

1.2.2 Analyse technique

L'analyse technique a pour principe l'identification de motifs géométriques dans les séries temporelles de prix afin d'en prédire la tendance. Cette discipline repose sur l'hypothèse que les prix reflètent à tout instant toute l'information pertinente, et que leur observation suffit pour la prédiction (cf. [BBN08]). Elle ne repose sur aucun fondement théorique concernant l'économie ou les marchés financiers, et va même à l'encontre des hypothèses de marchés efficients et d'absence d'opportunité d'arbitrage. Elle est centrée sur la reconnaissance empirique de caractéristiques sur les prix et les volumes reflétant le comportement sous-jacent des acteurs de marché, et la ligne de tendance est le concept-clé. Il peut premièrement s'agir d'une transformation locale de la série temporelle, comme une approximation linéaire ou un lissage par moyennes mobiles, permettant d'extrapoler la direction des prix. Dans cette catégorie d'indicateurs, on trouve aussi des courbes majorantes et minorantes historiques, telles que des signaux d'achat/vente sont donnés lorsque le prix dépasse cet encadrement. Ce type de méthodes peut être vu comme un filtrage en traitement de signal ou des estimations statistiques sur les prix. Une deuxième

catégorie de méthodes étudie des motifs géométriques formés par l'agencement de lignes de tendances pour former des motifs géométriques. Ces motifs sont associés empiriquement à des mouvements caractéristiques. Il peut ainsi s'agir de droites passant par deux extrema locaux, marquant des niveaux de "résistance" (maxima) ou de "support" (minima), encadrant la courbe de prix, ou encore de motifs visuels tels que "tête et épaules" ou "tasse à anse". L'écartement des prix par rapport à ces motifs pourra entre autres être interprété comme un changement de tendance. Pour plus de détails avec des illustrations de ces techniques, on peut se référer à [Sew07].

La pratique de l'analyse technique a été souvent controversée du fait de la déconnexion avec la théorie économique et à cause de la subjectivité de l'interprétation, jusqu'à être comparée à l'astrologie ([Mal96]). La performance prédictive positive montrée par les études est nuancée par des travaux (cf [PI04]) mettant en lumière un manque de rigueur dans les protocoles de test. Néanmoins, l'analyse technique évolue depuis les années 2000, notamment dans [LMW00], vers un cadre formalisé s'apparentant à l'inférence statistique avec des méthodes telles que les ondelettes ou le lissage par noyaux pour décrire les prix. De même, ce cadre s'accompagne davantage d'algorithmes objectifs et de tests de performance rigoureux.

1.2.3 Modélisation stochastique

La modélisation stochastique a été l'une des disciplines les plus prisées des vingt dernières années en finance quantitative. Reposant sur un cadre mathématique théorique poussé, les méthodes sont appliquées à grande échelle dans les salles de marchés des banques d'investissement, de telle sorte que les prix de nombreux instruments financiers sont calculés au moyen de ces modèles. Sans poser le cadre rigoureux ni développer les points techniques, nous en rappelons les grandes lignes. On peut se référer à [Wil07, LL12] pour plus de détails.

Le principe est la modélisation du prix comme un processus stochastique en temps continu, à partir d'un mouvement brownien géométrique. Cette modélisation permet de calculer le prix des produits dérivés. Un produit dérivé est un produit financier dont la valeur (appelée *payoff*) contractuellement délivrée à l'acheteur à la date de maturité future dépend de la trajectoire de prix d'un actif, dit sous-jacent. Ce type d'instrument financier est initialement pensé pour couvrir le risque sur des actifs simples. Par exemple, si un investisseur veut se protéger contre la hausse du prix d'une action un an plus tard, il peut acheter une option d'achat (appelée *call*), lui permettant d'acheter un an plus tard, s'il le souhaite, l'action à un prix fixé en avance, appelé *strike*. La modélisation stochastique du prix de l'actif sous-jacent permet d'estimer l'espérance du payoff et de donner un prix au produit dérivé.

Hypothèses

On considère un actif de prix $S(t)$, de volatilité constante σ et de dérive constante μ (espérance de rendement par unité de temps). Les hypothèses sont les suivantes :

1. Le prix de l'actif suit une marche aléatoire log-normale. Autrement dit, les rendements logarithmiques de l'actif sont des réalisations indépendantes d'une même loi normale.
2. Il existe un actif sans risque de rendement connu et constant.
3. L'actif ne paie pas de dividendes.
4. Il n'y a pas de coûts de transaction.
5. Marchés complets : il est possible de répliquer continûment tout actif ou instrument financier au moyen d'un portefeuille d'actifs échangeables. Ce point est essentiel pour pouvoir couvrir le risque de tout portefeuille.

6. Absence d'opportunité d'arbitrage : il est impossible d'obtenir un gain d'espérance strictement positive à partir d'un capital initial nul. Cela inclut en particulier l'hypothèse d'efficience des marchés.

Modèle de Black-Scholes-Merton [Mer73, BS73]

Sous ces conditions, le prix de l'actif dans le temps est régi par l'équation aux dérivées partielles :

$$dS = \mu S dt + \sigma S dW$$

où W est un mouvement brownien. Le cadre théorique implique de plus que le prix des produits dérivés sous les hypothèses doit être l'espérance du payoff. Pour des payoffs simples (*call*, *put*, digitales), cette équation permet de calculer l'espérance sous forme analytique, au moyen du lemme d'Itô [Ito44]. Pour les payoffs plus complexes, le modèle permet de simuler les trajectoires de prix, qui peut alors être estimé par une méthode Monte-Carlo.

On calcule généralement la sensibilité du prix du produit dérivé par rapport au prix du sous-jacent ou aux paramètres du modèles, afin de couvrir dynamiquement la valeur du produit et assurer la livraison du payoff final en minimisant le coût de couverture.

Discussion

Certaines hypothèses du modèle de base peuvent être assouplies : il est notamment possible de tenir compte des dividendes et de considérer des volatilités non-constantes. Les hypothèses de complétude et de non-arbitrage sont en revanche essentielles, même si en réalité la couverture ne peut être qu'en temps discret et les arbitrages existent.

Le modèle a été très largement adopté par les banques d'investissement pour le calcul de prix de produits dérivés. En donnant un moyen de définir potentiellement un prix à tout produit financier, il a permis le développement et la création de produits dérivés. Le prix commercial des produits dérivés élaborés est généralement composé d'un *fair price* (prix calculé par le modèle), d'une marge de couverture (variation prix à une perturbation défavorable sur les paramètres) et d'une marge commerciale. En pratique, les paramètres sont estimés de façon spécifique à chaque banque et équipe à partir de modèles d'estimation. Chaque établissement développe ses propres variantes du modèle et sa propre implémentation de la résolution numérique, par éléments finis ou par Monte-Carlo pour les produits plus complexes. Il est à noter que même si les modèles théoriques sont bien établis, la validité de la méthode de résolution des prix peut varier : les prix des produits complexes sont plus difficiles à calculer et peuvent nécessiter des considérations supplémentaires (couverture des discontinuités du payoff, convergence de Monte-Carlo, etc.).

1.2.4 Analyse quantitative statistique

Il existe de nombreuses méthodes statistiques de traitement des données d'actifs financiers. Nous donnons ici un aperçu de deux types de méthodes fréquemment rencontrées pour l'estimation du risque.

Modèles à facteurs cachés

L'estimation de la matrice de covariance entre les actifs financiers est difficile lorsque le nombre d'actifs est élevé. Afin de réduire la dimension du problème, une méthode couramment utilisée est de considérer que tous les actifs sont générés par des facteurs, eux-mêmes calculés à partir des actifs. L'estimation se transpose ainsi aux facteurs, ce qui présente au moins deux avantages :

— L'estimation statistique dans l'espace transformé est plus efficace : typiquement, les facteurs peuvent être décorrélés ou indépendants.
— La représentation permet de sélectionner un nombre de facteurs réduit et résumer l'information.

En reprenant les notations de la partie 1.1, on suppose que le vecteur de rendement des D actifs observés s'écrit en fonction de M facteurs formant un vecteur F :

$$R = BF + \varepsilon \ ,$$

où B est une matrice $D \times M$ et ε un vecteur de bruits indépendants des facteurs. Le but est de résumer l'information des D actifs en trouvant un nombre réduit ($M << D$) de facteurs optimaux. Si F est connu, la matrice de covariance Σ de R s'exprime simplement en fonction de la matrice de covariance Γ_F de F et de celle du bruit, Γ_ε :

$$\Sigma = B\Gamma_F B^\top + \Gamma_\varepsilon \ .$$

Le problème d'estimation se réduit donc à F, sous des hypothèses convenables sur ε, typiquement la décorrélation entre les bruits.

L'analyse en composantes principales est l'une des méthodes les plus couramment utilisées, car elle permet de trouver les facteurs décorrélés optimaux. Néanmoins, la littérature sur ce type de problématique est riche à la fois en séparation de sources, en apprentissage non-supervisé et en statistiques. D'autres méthodes, comme l'analyse en composantes indépendantes ou la factorisation de matrices non-négatives, sont aussi considérées par les analystes de risque. Le choix de la méthode dépend des propriétés attendues sur les facteurs. Par exemple, une contrepartie de l'efficacité de beaucoup de modèles factoriels est la perte de l'interprétation des facteurs. Même si l'analyse est efficace au sens statistique, il peut parfois être judicieux de préférer des facteurs explicites ou des facteurs parcimonieux, afin de garder la possibilité de donner un sens économique à chaque facteur.

Modèle de volatilité GARCH

Les prix des actifs étant des fonctions du temps, on peut leur appliquer des méthodes de séries temporelles. On peut par exemple se référer pour les définitions et de plus amples détails à l'ouvrage [Tsa05], qui traite de l'étude de séries financières, ou au cours [MR10b], sur l'analyse des séries temporelles. En particulier, une façon de traiter la dépendance temporelle des prix d'un actif est de les représenter comme un processus auto-régressif. Sous cette forme, on tient compte de la dépendance temporelle en exprimant la valeur en t comme une combinaison linéaire des valeurs passées. Un processus $\{X_t\}$ est un processus auto-régressif à l'ordre p ($AR(p)$) si c'est un processus stationnaire au second ordre et qu'il est solution de l'équation de récurrence :

$$X_t = \sum_{i=1}^{p} \alpha_i X_{t-i} + Z_t$$

où les α_i sont des réels et Z_t est un bruit blanc de variance σ^2.

Ce cadre s'applique notamment à l'estimation de la variance des rendements d'un actif. Engle a tout d'abord proposé le modèle ARCH(p) (*AutoRegressive Conditional Heteroscedasticity* [Eng82]) :

$$\sigma_t^2 = \gamma\bar{\sigma}^2 + \sum_{i=1}^{p} \alpha_i u_{t-i}^2$$

où $\bar{\sigma}^2$ est une variance de long terme, les u_t sont les rendements de l'actif centrés autour de leur moyenne, et γ le poids donné à la variance de long terme. Avec la contrainte $\gamma + \sum_{i=1}^{p} \alpha_i = 1$, cette formulation exprime la variance comme une combinaison linéaire d'une variance de long terme et les p derniers rendements au carré. Les poids α_i sont typiquement décroissants en fonction de l'ancienneté, afin de pondérer davantage les observations récentes.

Le modèle GARCH(p,q) (*Generalized AutoRegressive Conditional Heteroscedasticity* [Bol86]) de Bollerslev étend cette formulation en ajoutant des termes antérieurs de variance :

$$\sigma_t^2 = \gamma\bar{\sigma}^2 + \sum_{i=1}^{p} \alpha_i u_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 \ .$$

Un tel modèle calibré sur des données historiques peut ainsi servir d'estimateur de la variance des rendements.

1.2.5 Optimisation de Black-Litterman

L'optimisation de portefeuille de Black-Litterman [BL92] a été proposée en 1990 pour limiter la sensibilité de l'allocation classique aux paramètres estimés (espérance et covariance des rendements). Cette méthode utilise en entrée des informations a priori, telles des avis d'expert, et modifie l'estimation initiale des paramètres vers ces informations. Nous utilisons ici les notations de [Meu05, Chap. 9].

Méthode

Supposons donnée une estimation "officielle" f_R de la distribution du vecteur aléatoire des rendements R. Afin de réduire le risque d'estimation, le gérant de portefeuille demande l'avis d'un expert dont il est capable d'évaluer la qualité. L'expert n'est pas supposé donner un avis sur tout le vecteur R, mais peut être spécialisé sur certains actifs, ou encore sur des combinaisons d'actifs ; de manière générale, on suppose que son domaine d'expertise est une fonction multivariée $g(R)$ des actifs. Il fournit donc son expertise sous forme d'un vecteur aléatoire V défini par sa densité de probabilité conditionnelle $f_{V|g(R)=g(x)}$, qu'on note plus simplement $f_{V|g(x)}$. Cette probabilité représente la confiance de l'avis de l'expert.
Nous nous intéressons plutôt à la distribution de R conditionnée à V. Par la règle de Bayes, on obtient :

$$f_{R|v}(x|v) = \frac{f_{V|g(x)}(v|x) f_R(x)}{\int f_{V|g(x)}(v|x) f_R(x) dx} \ .$$

Il suffit ensuite d'utiliser cette estimation dans le problème d'optimisation pour un avis d'expert donné.

Exemple

L'article original donne un cas particulier de cette méthode :
- L'estimation officielle est normale : $R \sim \mathcal{N}(\mu, \Sigma)$.
- Le domaine d'expertise est une combinaison linéaire de R : $g(x) = Px$. Cette expertise est donc représentée par la matrice de sélection P.
- La confiance est aussi normale : $V|Px \sim \mathcal{N}(Px, \Omega)$, où Ω représente le risque associé à l'opinion de l'expert.

Étant donné un avis v, on peut montrer [Meu05, Chap. 9] que la distribution résultante est normale : $R|v \sim \mathcal{N}(\mu', \Sigma')$. L'espérance est :

$$\mu'(v, \Omega) = \mu + \Sigma P^\top \left(P\Sigma P^\top + \Omega\right)^{-1} (v - P\mu)$$

et la matrice de covariance est :

$$\Sigma'(\Omega) = \Sigma + \Sigma P^\top \left(P\Sigma P^\top + \Omega\right)^{-1} P\Sigma \ .$$

1.3 Position des travaux

Nous proposons d'appliquer des méthodes d'apprentissage statistique à la prédiction de rendements pour la gestion de portefeuille.

1.3.1 Notions sur l'apprentissage statistique

L'apprentissage statistique traite de la construction automatique de modèles à partir de données observées, dans le but d'analyser le phénomène sous-jacent aux observations ou d'effectuer des prévisions. On distingue en général deux catégories : les problèmes d'apprentissage supervisés et non-supervisés.

Dans les problèmes supervisés, on dispose de données d'observation $\{X_1, \ldots, X_n\} \subset \mathcal{X}$, appelées entrées du problème, et de données $\{Y_1, \ldots, Y_n\} \subset \mathcal{Y}$ liées aux entrées, appelées réponses ou sorties du problème. L'on souhaite, pour toute entrée X nouvelle, prédire la sortie Y, et pour cela construire à partir des données un modèle associant une sortie à toute entrée. Par exemple dans le diagnostic médical, on observe les données cliniques des patients (poids, âge, résultats d'analyses, symptômes) et on souhaite en déduire s'ils sont atteints de certaines pathologies, grâce aux cas connus historiquement.

Les problème non-supervisés ne considèrent que des observations $\{X_1, \ldots, X_n\} \subset \mathcal{X}$, pour lesquelles l'on souhaite définir une sortie pertinente pour la structure des données. Il s'agit souvent du *clustering* des données, c'est-à-dire leur regroupement en sous-ensembles de données similaires pour un certain critère. Par exemple sur les marchés financiers, on peut vouloir distinguer des groupes d'entreprises similaires en termes de rendement, de risque ou de données financières.

On s'intéressera plus particulièrement ici à l'apprentissage supervisé, qui correspond aux problématiques de prédiction.

Cadre

On observe n échantillons de données d'apprentissage $\{(X_1, Y_1), \ldots, (X_n, Y_n)\} \subset \mathcal{X} \times \mathcal{Y}$, où $\mathcal{X}$ est l'espace d'entrée et $\mathcal{Y}$ est l'espace de sortie. Par exemple, $\mathcal{X}$ pourra être $\mathbb{R}^d$ $(d > 0)$, ou encore le produit d'ensembles finis d'attributs catégoriques (dans l'exemple de diagnostic : homme/femme, profession, malade/sain, etc.). $\mathcal{Y}$ pourra être un ensemble fini ou un sous-ensemble de $\mathbb{R}$. On suppose que les $Z_i = (X_i, Y_i)$ sont des réalisations indépendantes d'un même couple de variables aléatoires $Z = (X, Y)$ de loi P, inconnue a priori.

A partir de ces données, on cherche à contruire une fonction de prédiction f, qui est une fonction mesurable de $\mathcal{X}$ dans $\mathcal{Y}$. On note $\mathcal{F}(\mathcal{X}, \mathcal{Y})$ l'ensemble des fonctions de prédiction. On définit une fonction de perte $\ell : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$ mesurant le coût associé à la prédiction y' lorsque la vraie sortie est y. L'algorithme d'apprentissage peut être défini comme une fonction associant

aux données d'apprentissage la meilleure fonction de prédiction au sens de la perte ℓ dans $\mathcal{F}(\mathcal{X}, \mathcal{Y})$. Le choix de la fonction de perte détermine en grande partie le type de problème traité, et l'algorithme d'apprentissage est étroitement lié à l'ensemble $\mathcal{F}(\mathcal{X}, \mathcal{Y})$ dans lequel l'optimum est cherché. La régression et la classification sont des exemples courants :

— Régression aux moindres carrés : $\mathcal{Y} = \mathbb{R}$ et $\ell(y, y') = ||y' - y||_2^2$.
— Classification binaire : $\mathcal{Y} = \{0, 1\}$ et $\ell(y', y) = \mathbb{1}_{y' \neq y}$, où $\mathbb{1}$ est la fonction indicatrice, i.e. $\ell(y', y)$ vaut 1 si $y' \neq y$ et 0 sinon.

Risque et fonction cible

La perte quantifie la qualité de la fonction de prédiction f, donc l'objectif de l'algorithme d'apprentissage est de minimiser l'espérance de la perte sous la loi P. On appelle cette grandeur le risque de f, aussi appelée erreur de généralisation :

$$R(f) = \mathbb{E}\left[\ell\left(Y, f(X)\right)\right] \ . \tag{1.8}$$

On appelle risque optimal $R^\star$ l'infimum de R sur $\mathcal{F}(\mathcal{X}, \mathcal{Y})$. Lorsqu'il existe une fonction atteignant le risque optimal, on appelle *fonction cible* une telle fonction, notée $f^\star$.

Régression En régression aux moindres carrés, le risque est l'erreur quadratique moyenne : $R(f) = \mathbb{E}\left[||f(X) - Y||_2^2\right]$. Une fonction cible est :

$$\eta^\star(x) = \mathbb{E}(Y|X = x) = \int_{\mathcal{Y}} y \ dP(y|x) \ .$$

Classification En classification (binaire ou multi-classe), le risque est la probabilité d'erreur : $R(f) = \mathbb{E}\left[\mathbb{1}_{f(X) \neq Y}\right] = P\left(f(X) \neq Y\right)$. Les fonctions cibles $f^\star$ sont celles qui renvoient la classe la plus probable sachant X, appelées classifieurs de Bayes : $\forall x \in \mathcal{X},\ f^\star(x) \in \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\ P(Y = y|X = x)$.
Pour la classification binaire où $\mathcal{Y} = \{0, 1\}$, on a :

$$f^\star(x) = \begin{cases} 1 \text{ si } P(Y = 1|X = x) > \frac{1}{2} \\ 0 \text{ sinon.} \end{cases}$$

Consistance

Si on note $\hat{f}\left(Z_1, \ldots, Z_n\right)$ la fonction de prédiction produite par l'algorithme $\hat{f}$ pour les n échantillons de données, on définit le risque de l'algorithme par son risque moyen sous P, noté $\mathbb{E}R_n(\hat{f})$:

$$\mathbb{E}R_n(\hat{f}) = \mathbb{E}_{Z^n}\left[R\left(\hat{f}\left(Z_1, \ldots, Z_n\right)\right)\right] \ .$$

La propriété élémentaire désirée pour les algorithmes d'apprentissage est leur capacité à approcher le risque optimal lorsque la quantité de données augmente :

Définition 1 (Consistance)**.** *Un algorithme d'apprentissage est dit consistant par rapport à la loi P si et seulement si*

$$\mathbb{E}R_n(\hat{f}) \underset{n \to +\infty}{\longrightarrow} R^\star \ .$$

La consistance garantit le fait que la performance de la fonction de prédiction construite par l'algorithme converge vers la meilleure performance possible, même si cela n'implique pas une "bonne" performance dans l'absolu.

Validité pour des observations dépendantes

L'indépendance des observations est une hypothèse-clé pour la plupart des résultats théoriques essentiels sur les algorithmes d'apprentissage. Néanmoins, les données réelles sont souvent constituées d'observations dépendantes, par exemple lorsqu'elles se succèdent dans le temps ou sont proches dans l'espace. Pourtant, les algorithmes d'apprentissage sont aussi appliqués avec succès sur des cas réels ne vérifiant pas l'hypothèse d'indépendance.

On trouve des résultats de consistance dans le cas de processus dits β-*mixing* (voir [Yu94] pour la définition de différents types de processus dépendants), notamment [KV04, LKS05, MR10a]. La consistance des SVM a aussi été démontrée dans [SHS09] pour des processus vérifiant une loi faible des grands nombres.

1.3.2 Travaux en apprentissage statistique pour la finance

L'application des méthodes d'apprentissage statistique est un sujet d'intérêt depuis la fin des années 1980 au moins, et ce dans de nombreuses problématiques financières : gestion du risque, scoring de credit, prédiction de rendements, etc. Cependant, si l'identification de facteurs de risque est devenue une application standard en apprentissage non-supervisé (comme cité en partie 1.1.3), la prédiction des rendements ne donne lieu à aucune méthode unificatrice, ni même courante. La grande diversité des méthodes en témoigne, dans tous types d'algorithmes : chaînes de Markov [TSN89], SVM [HNW05], réseaux de neurones [KAYT90]. L'engouement pour ces derniers est toutefois plus visible, d'après les livres qui traitent de l'application des réseaux de neurones à la finance, par exemple [ST02]. Dans le passé, la mise en œuvre pratique des méthodes a pu être limitée, de l'aveu même de certains opérationnels, par la puissance de calcul nécessaire au traitement simultané de l'ensemble des actifs financiers pour la décision en temps réel.

La publication de l'ouvrage *The Elements of Statistical Learning* [HTF01] a contribué à la diffusion et à la mise en œuvre pratique des méthodes répandues d'apprentissage par les opérationnels en finance quantitative. Dans une présentation récente [Ron14], il apparaît toutefois que l'application directe des algorithmes à la gestion de portefeuille s'avère sinon décevante, du moins difficile. On peut expliquer cette difficulté par le fait que l'application ne se résume pas à l'apprentissage du modèle de prédiction. En effet, il s'agit d'un processus complet en plusieurs étapes, comprenant la représentation des données, le modèle prédictif, la validation des paramètres et l'évaluation des performances. Ces étapes forment un ensemble cohérent, et il peut arriver que la majeure partie de la performance repose sur le choix d'une représentation ou d'une métrique adaptée au problème (cf. [BM10]).

On observe depuis le début de la crise monétaire de 2008 un intérêt croissant des opérationnels pour les méthodes d'apprentissage statistique. En conséquence, les approches traitent de l'ensemble de la procédure de prédiction de façon plus complète, par exemple pour la prédiction séquentielle [Cha09] ou la représentation multi-échelle [Mah12].

1.3.3 Objectifs et enjeux

Nous cherchons à concevoir une procédure permettant de donner une prédiction sur les rendements futurs des actifs candidats au portefeuille, à partir de l'observation de l'ensemble des actifs d'intérêt. Nous ne chercherons pas nécessairement à estimer de façon exacte l'intégralité du vecteur de rendements et résoudre le problème d'optimisation, mais plutôt à produire des prédictions sur l'espérance ou le sens de variation, assimilables à des avis d'expert. Pour le gérant de portefeuille, l'outil de prédiction ne remplace pas son estimation, mais constitue une source d'information à prendre en compte dans sa décision.
Par conséquent, l'intégration théorique au problème d'optimisation semble plus pertinente avec

une procédure de type Black-Litterman (vue en partie 1.2.5), où l'avis d'expert de l'apprentissage statistique influence son estimation initiale, en supposant qu'il dispose par ailleurs d'une estimation personnelle (et inconnue) des paramètres du problème. La matrice de confiance accordée aux prédictions pourra être construite à partir d'indicateurs du risque de prédiction et des performances historiques des prédictions.

Afin de s'appliquer directement aux décisions concrètes des gérants de portefeuilles, les méthodes devront être performantes, proposer une solution aux problématiques scientifiques rencontrées et répondre aux objectifs opérationnels.

Détection d'arbitrages

Nous considérons que les marchés financiers ne sont en pratique pas parfaitement efficients, et que l'information ne se propage pas instantanément sur tous les actifs et tous les instruments. L'absence d'opportunité d'arbitrage signifie que les arbitrages, lorsqu'ils existent, sont immédiatement identifiés, exploités, et par conséquent invalidés. Nous considérons que l'absorption des arbitrages complexes faisant intervenir un grand nombre d'actifs différents n'est pas immédiat. Dans cette optique, il est possible de prédire les variations temporelles au moyen de méthodes d'apprentissage statistique utilisant des données représentatives du contexte économique et financier.

Méthode non-paramétrique

La théorie classique de l'optimisation de portefeuille utilise implicitement des modèles paramétriques (gaussiens) pour estimer les rendements. Une fois connus l'espérance et la covariance des rendements, la sélection des actifs devient une optimisation sous contraintes. Cependant, cette approche possède un certain nombre de défauts :

- Les modèles paramétriques sont rigides et peuvent être exposés à un risque de modèle important.
- Dès lors que les distributions ne sont pas gaussiennes, la description en espérance-variance ne suffit pas à caractériser les rendements.
- Les hypothèses implicites du choix des modèles peuvent ignorer des variables significatives.
- Les structures de dépendance entre les actifs varient dans le temps, et il est nécessaire d'adapter les modèles dynamiquement.

Nous proposons de mettre en œuvre des méthodes non-paramétriques fondées uniquement sur les données historiques, avec le moins d'a priori possible sur la distribution des rendements et les dépendances entre les actifs.

Objectifs opérationnels

La prise de décision dans la gestion de portefeuilles repose sur des méthodes d'estimation, mais surtout sur l'expertise des opérationnels. Cette expertise inclut des a priori correspondant à des modélisations implicites des mécanismes des marchés. En tant qu'application destinée à la prise de décision réelle, l'approche que nous proposons doit répondre à des objectifs opérationnels pertinents.

Exhaustivité Le nombre d'actifs et la complexité des dépendances entre eux est telle que la prédiction doit permettre de considérer l'ensemble des variables potentiellement pertinentes, afin de ne pas ignorer d'information utile. On peut considérer qu'il s'agit d'un problème en

grande dimension. Les méthodes devront donc tenir compte de cette contrainte et permettre de sélectionner les variables importantes.

Interprétabilité Afin de permettre des décisions éclairées, les méthodes ne doivent pas constituer une boîte noire. Par conséquent, chaque étape du processus de prédition doit fournir des résultats et modèles interprétables sur la réalité sous-jacente.

Objectivité Les acteurs de marché utilisent implicitement des hypothèses similaires, ainsi qu'une part de subjectivité commune. Avoir à disposition des résultats de prédiction issus de procédures objectives et agnostiques permet au gérant de les confronter avec ses propres conclusions, et d'identifier ses propres hypothèses implicites. Pour cela, il est important de proposer une approche portant le moins d'hypothèses externes possible.

Adaptativité Nous ne pensons pas qu'il soit possible d'entraîner un modèle une seule fois et de l'utiliser indéfiniment en ignorant les nouvelles données. Les méthodes employées doivent pouvoir s'adapter automatiquement au contexte de marché et aux changements significatifs, sans intervention ad hoc de l'opérationnel – le choix de l'intervention serait aussi subjectif.

1.3.4 Problèmes traités

L'ambition est de concevoir un processus d'ensemble autonome allant des données brutes au résultat de prédiction utilisable, tout en pouvant s'adapter au contexte de marché, contrôler le risque de modèle et se calibrer automatiquement. Chaque étape du processus porte ses propres difficultés et nécessite des méthodes de résolution dédiées. En conséquence, l'étendue des problématiques scientifiques traitées est large, dépassant les problématiques habituelles du problème de prédiction.

Représentation des données

L'information considérée est l'ensemble des prix historiques (journaliers) des actifs, de différentes catégories : actions, indices, taux d'intérêt, crédit, taux de change, matières premières, etc. Il est nécessaire de pré-traiter ces données brutes afin de constituer une représentation pertinente pour l'apprentissage supervisé. Nous proposons d'indexer les séries financières au moyen d'une approximation linéaire par morceaux, où les points de changement de tendance sont déterminés optimalement par segmentation optimale des séries en périodes homogènes.

Méthode de prédiction

Dans la représentation linéaire par morceaux, les périodes homogènes sont caractérisées par une tendance et la variance des résidus. Nous utilisons ces deux grandeurs de chaque actif pour constituer des vecteurs descripteurs du marché en entrée du problème d'apprentissage supervisé. La prédiction du niveau exact du rendement est un problème difficile, et nous chercherons principalement à prédire le signe du rendement, dans un cadre de classification au moyen d'arbres de décision agrégés en forêts aléatoires.

Prédiction multi-tâche

La prédiction des différents actifs consituant un portefeuille est un problème dont la sortie est multi-variée. Dans le cadre de l'apprentissage multi-tâche, nous proposons une formulation innovante permettant de résoudre simultanément les tâches de prédiction et les relations entre les tâches, lorsqu'elles forment un graphe pondéré.

Contrôle du risque de prédiction

Le risque associé aux prédictions individuelles est une problématique opérationnelle cruciale qui est relativement peu traitée dans le cadre de l'apprentissage supervisé. Nous proposons des indicateurs, idéalement liés à l'erreur de généralisation, rendant compte de la fiabilité des résultats de prédiction pour la décision réelle. Les solutions s'inspirent notamment de la détection d'anomalies et définissent une métrique adaptée au modèle d'apprentissage utilisé.

Évaluation de la performance

Nous établissons un protocole de test (*backtest*) séquentiel par fenêtres glissantes, où l'ensemble de test est postérieur aux données d'entraînement. Les mesures de performance permettent d'évaluer la qualité de l'algorithme dans le temps de façon réaliste vis-à-vis des conditions d'utilisation.

Automatisation de la sélection de paramètres

Les tests montrent que l'influence des paramètres sur la qualité prédictive est très variable. Nous proposons un processus de calibration dynamique des paramètres les plus importants, permettant de sélectionner automatiquement les paramètres d'apprentissage optimaux.

Chapitre 2

Représentation des données

Le prix de cotation instantané d'un actif financier est défini pendant les heures d'ouverture des marchés par le dernier prix de transaction sur l'actif. Dans une journée de marché, on distingue en particulier le prix d'ouverture, déterminé par la première transaction, le prix de clôture, correspondant à la dernière, et l'ensemble des prix observés dans la journée (*intraday*). La fréquence des transactions varie selon l'actif et l'activité des acteurs de marchés le concernant, mais pour des actifs couramment échangés tels que les actions de grandes entreprises, la durée entre deux transactions successives, et donc entre deux prix différents, peut être inférieure à une milliseconde. Il existe par conséquent une grande variété de fréquences d'investissement, selon l'échelle et la nature des variations sur lesquelles l'on souhaite investir. Si l'intérêt se porte sur la croissance à long terme d'une entreprise, l'horizon d'investissement sera de l'ordre de plusieurs années. S'il s'agit d'une gestion active de portefeuille, l'horizon considéré sera plutôt d'une semaine ou d'un mois. Si l'objectif est d'exploiter un grand nombre d'infimes opportunités d'arbitrage, l'horizon est inférieur à la seconde.

Nous nous plaçons dans la perspective de la gestion de portefeuille où les données d'intérêt sont les prix journaliers de clôture. Nous n'aborderons pas de données de fréquence supérieure, qui sont de nature sensiblement différente et nécessitent des traitements spécifiques.

Dans notre processus de prédiction, les données d'apprentissage ne sont pas fournies, mais doivent être construites à partir de données de marché. La première étape du processus consiste donc à transformer les données brutes en des descripteurs adaptés au problème de prédiction. Les difficultés sont nombreuses : bruit, non-stationnarité, dépendance temporelle et grande dimension. Nous proposons d'indexer les séries temporelles par des approximations linéaires par morceaux et une segmentation par arbres [CW92, BFOS84], afin de fournir une représentation pertinente du marché. Nous définissons ainsi les entrées et les sorties du problème d'apprentissage supervisé.

2.1 Nature des données

2.1.1 Catégories d'actifs

Les actifs dont les données de cotation sont disponibles sont nombreux et variés. Nous considèrerons diverses classes d'actifs aux caractéristiques variées dans le but de couvrir les comportements des marchés financiers de façon exhaustive. Nous nous intéresserons uniquement aux actifs liquides, c'est-à-dire couramment échangées en volumes importants, car ils sont représentatifs des marchés financiers dans l'ensemble, et nous exposent peu aux problèmes de données. Les cours de clôture de ces actifs sont supposés disponibles en historique dès qu'ils sont connus.

Actions Les actions sont les parts émises par les entreprises sur leurs fonds propres. Les actions sont échangées sur le marché réglementé, et la variation de leur prix reflète en général la fluctuation de la valeur attribuée par les investisseurs à l'entreprise. Nous nous intéresserons uniquement aux entreprises de capitalisation boursière suffisante : en Europe, nous nous limitons aux actions constituant l'indice STOXX Europe 600 [1].

Indices d'actions Les indices d'actions sont des combinaisons linéaires d'actions représentatives d'un marché ou d'un type d'entreprises donné. Leur définition, calcul et publication sont gérées par des entreprises financières. Les actions utilisées dans la composition d'un indice sont appelées constituants de l'indice. La composition et les poids des constituants peuvent varier au cours du temps selon les décisions périodiques ou exceptionnelles d'un comité d'experts. Par exemple, le CAC 40 [2], compilé par NYSE Euronext, est l'indice regroupant les actions des 40 entreprises les plus représentatives cotées sur le marché Euronext Paris. Elles sont sélectionnées sur des critères de capitalisation et de liquidité, de sorte que l'indice reflète l'économie française. Les indices peuvent représenter un pays (DAX 30, FTSE 100, S&P 500), une région économique (EURO STOXX 50, STOXX Europe 600), un secteur (STOXX Europe 600 Food & Beverage). Les indices d'actions sont particulièrement pertinents dans une vision d'ensemble des marchés financiers. Etant une moyenne pondérée, leurs rendements sont plus réguliers que les actions seules.

Matières premières Les données sur les matières premières sont les prix des contrats à terme portant sur l'échange de ces matières premières à une date future fixée. Les matières premières couramment échangées incluent par exemple le blé, le cuivre, le coton, l'or et le pétrole. Comme ce sont des actifs tangibles, leurs fluctuations peuvent refléter une économie plus "réelle" liée à la production et aux ressources. Ce sont aussi des indicateurs de risques géopolitiques, car leur production dépend du sol d'origine.

Devises Dans les théories économiques, le taux de change entre deux devises est lié aux flux de capitaux entre les zones économiques respectives. Les devises en circulation les plus courantes (dollar, euro, livre sterling, yen, etc.) sont échangées sur plusieurs marchés, et donc en continu, à la différence des actions qui ne s'échangent que pendant l'ouverture des marchés où elles sont cotées. Le cours de clôture sera défini dans notre base de données comme la valeur à l'instant de la clôture des marchés européens principaux.

Taux d'intérêt Les taux d'intérêt des emprunts souverains, à différentes maturités, reflètent le rendement annuel de la détention de tels emprunts. Même si ces taux sont souvent assimilés à des actifs sans risque, la modélisation de leurs variations est un sujet complexe [HW90]. Le taux des emprunts émis par un État, dit taux nominal, est souvent analysé en trois composantes porteuses d'informations différentes sur le pays :

- le taux réel, indiquant la croissance véritable de l'économie du pays,
- l'inflation, qui reflète l'augmentation des prix,
- la prime de risque spécifique au pays, qui peut être vue comme l'excès de rendement exigé par les investisseurs. Plus l'économie du pays est considérée risquée, plus le taux d'emprunt est élevé.

Il sera intéressant de considérer les taux d'emprunts de différentes maturités, car leurs variations reflètent des aspects différents de l'économie et leur perception par le marché. Une maturité

1. La définition et les données historiques sont disponibles sur le site de STOXX : http://www.stoxx.com/indices/index_information.html?symbol=SXXP.
2. https://indices.nyx.com/fr/products/indices/FR0003500008-XPAR

courte (< 1 an) reflète des problématiques de liquidité dans le futur proche, tandis que les taux longs (10 ans) sont liés à la solvabilité et au développement à long terme. Nous incluons aussi l'inflation *break-even*, qui est le taux d'inflation attendu par les acteurs de marché.

Dette souveraine Nous appréhendons la dette des pays par leur CDS (*Credit Default Swap*), qui sont les produits dérivés permettant de se protéger contre un éventuel défaut de remboursement. Plus précisément, la grandeur d'intérêt est le *spread* de CDS, soit la différence entre le prix d'achat et le prix de vente de la protection. Intuitivement, un écart élevé montre une forte demande, et donc une estimation élevée de la probabilité de défaut. Au contraire, un écart faible reflète la confiance des investisseurs sur la dette en question. Le spread de CDS est donc lié au risque estimé par les acteurs de marché à propos d'un pays, et la modélisation de la prime de risque (citée plus haut) à partir du spread de CDS est une problématique souvent rencontrée chez les opérationnels et dans la littérature [Dam].

Crédit d'entreprises De façon similaire, le spread de CDS sur les obligations des entreprises reflète le risque estimé [Mer74]. Nous considérons ce risque de manière agrégée grâce aux indices Markit iTraxx[3]. À l'instar des indices d'actions, ces indices de crédit sont calculés comme des combinaisons linéaires de spreads de CDS sélectionnés pour leur représentativité.

Volatilité implicite La volatilité implicite représente la volatilité estimée par les acteurs de marchés. Elle se déduit du prix des options couramment échangées sur des indices liquides, en supposant que ces prix sont issus d'un modèle de type Black-Scholes-Merton, vu en partie 1.2, utilisant la volatilité en tant que paramètre.

2.1.2 Difficultés posées

La nature des données financières soulève des difficultés pour l'apprentissage statistique, notamment en s'éloignant du cadre théorique classique d'observations indépendantes et identiquement distribuées.

Séries temporelles, non-stationnarité

Intuitivement, les prix d'un actif sont des signaux bruités successifs et dépendants dans le temps : ils forment une série temporelle, comme dans la figure 2.1. Les données brutes de prix sont donc très éloignées des cadres statistiques habituels de réalisations indépendantes et identiquement distribuées. La représentation des prix commence par la recherche d'invariants [Meu05], c'est-à-dire de grandeurs identiques en toute date du temps. Les variations des prix, c'est-à-dire les rendements journaliers, sont généralement utilisés comme des invariants, grâce à de bonnes propriétés constatées sur leur distribution [Tsa05, Meu05]. L'auto-corrélation est notamment moins triviale (figure 2.2). Cependant, il est aussi connu que la distribution des rendements n'est pas stationnaire [Ham89], ce qui est visible dans la figure 2.3. La non-stationnarité peut provenir de différentes sources :

- les événements ponctuels introduisant une rupture [KPSW92] : décisions politiques majeures, crises systémiques, faillite d'une entreprise de capitalisation boursière élevée, etc.
- l'introduction de nouveaux paradigmes influant sur le comportement des investisseurs ou d'instruments financiers modifiant le fonctionnement des marchés : modèles de pricing, utilisation des *Credit Default Swaps*, etc.

3. http://www.markit.com/Product/ITraxx

— changement de mode économique suivant le concept de cycles économiques : croissance, ralentissement, récession, reprise.

La non-stationnarité de la volatilité des actifs est connue et formalisée par les modèles de processus auto-régressifs de type GARCH [Eng82, Bol86]. Celle des rendements espérés a été étudiée plus récement par Fama et French [FF02], avec des conséquences sur les rendements observés. Cependant, nous considérons que la stationnarité des rendements dépend de l'échelle d'observation, et qu'il est possible d'identifier des périodes homogènes.

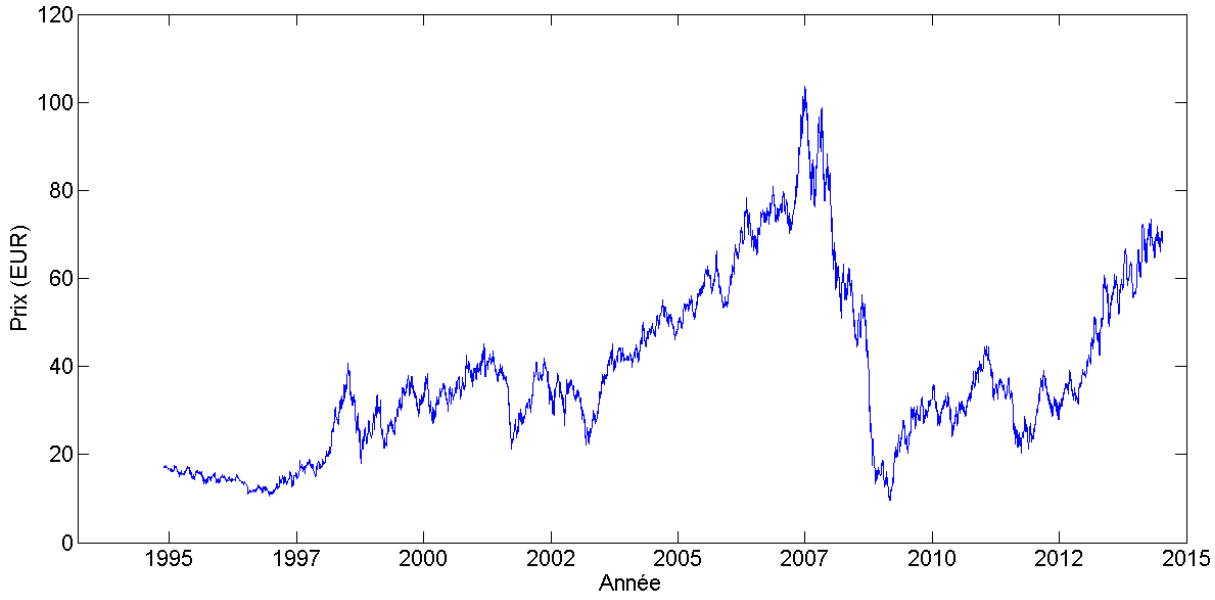


FIGURE 2.1 – Prix quotidien de l'action Renault

Inter-dépendance complexe

Les différents actifs sont dépendants entre eux, et la structure de dépendance varie selon le temps et l'échelle d'observation.

Corrélation entre actions De façon largement reconnue, les rendements d'actions similaires (par le secteur d'activité, la région géographique, etc.) sont corrélées à un certain degré, selon l'échelle de temps considérée. Considérons deux entreprises concurrentes, par exemple Renault et Peugeot pour le secteur automobile français. Les rendements de leurs actions suivent à long terme les variations du marché français, représenté par exemple par l'indice CAC 40, car les investisseurs tiennent compte de l'économie nationale dans leur valeur. De façon plus spécifique, faisant toutes deux partie du même secteur, leurs rendements sont potentiellement plus corrélés entre eux qu'avec des entreprises d'industries différentes. À court terme, imaginons que Renault annonce de meilleurs résultats que Peugeot. La valeur accordée par les investisseurs à son action peut en conséquence augmenter au détriment de celle de son concurrent, ce qui se traduit localement sur les rendements par une corrélation négative. Enfin, plaçons-nous en période de crise. La différence entre Renault et Peugeot importe alors peu, et les deux actions s'effondrent de la même manière avec le marché. Historiquement, on observe effectivement des corrélations très élevées entre les actions lors de périodes d'essor économique et de crises.

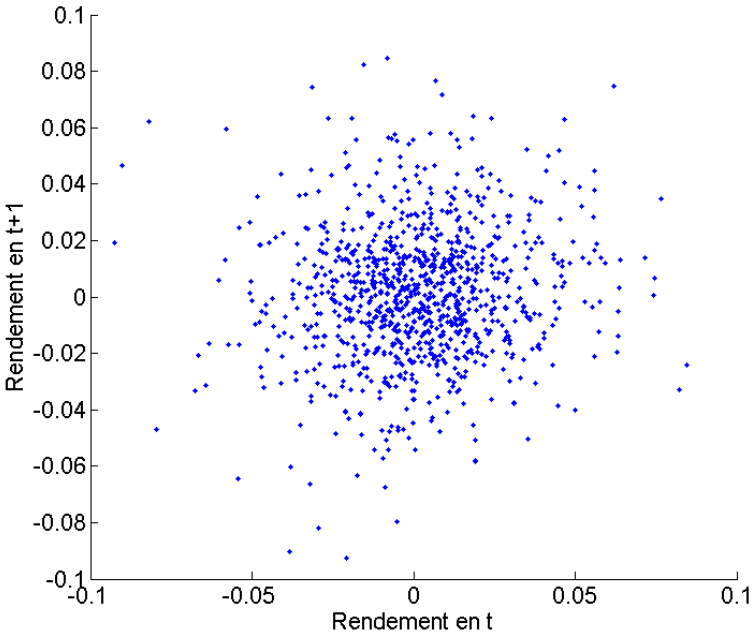


FIGURE 2.2 – Rendements journaliers de Renault depuis 2010. Les rendements successifs ne montrent pas de dépendance évidente.

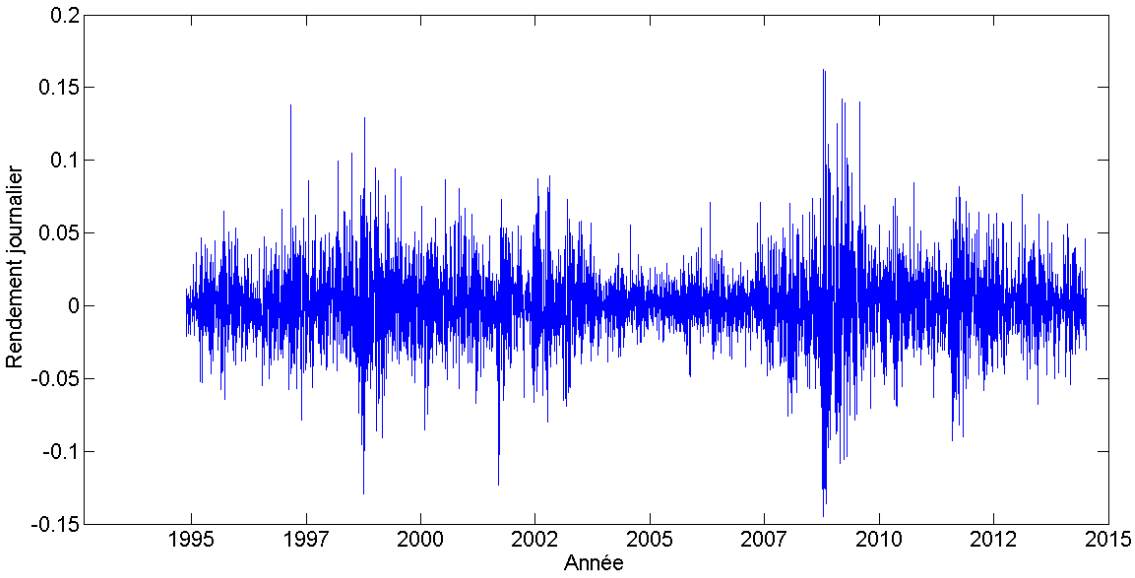


FIGURE 2.3 – Rendements journaliers de Renault dans le temps.

Un des inconvénients de l'omniprésence de corrélations parmi les actifs financiers est le manque de parcimonie de l'information. D'une part, comme l'erreur d'estimation des algorithmes est liée à la dimension des observations, l'idéal est de travailler sur un nombre minimal de variables tout en maximisant l'information portée par cet ensemble de variables. D'autre part (et par conséquent), il est plus intéressant de travailler dans une base dont les vecteurs de base sont décorrélés, ce qui revient à diagonaliser la matrice de covariance entre les actifs, suivant le théorème de Karhunen-Loève.

Relation structurelle Certains actifs sont liés de manière technique, soit à cause de mécanismes d'arbitrage quasi-instantanés, soit parce qu'il existe un modèle commun à la plupart des acteurs de marché. Prenons l'exemple des devises euro, dollar et yen. Sous l'hypothèse d'absence d'opportunité d'arbitrage, les taux de change T entre ces devises sont contraintes par une relation de Chasles, par exemple : $T_{euro/dollar} = T_{euro/yen}.T_{yen/dollar}$. La relation est d'autant plus forte que les actifs sont liquides, couramment traités sur la plupart des marchés financiers. Pour de tels actifs, tout écart infime est très rapidement détecté et exploité. La relation peut être plus faible pour d'autres types d'actifs liés non par cette contrainte mais par l'existence d'un modèle adopté par le marché. L'exemple limite fait partie des hypothèses du CAPM, qui suppose que tous les investisseurs ont le même modèle d'optimisation et les mêmes estimations. En pratique, il arrive qu'une partie significative des investisseurs suivent implicitement des modèles similaires, en s'intéressant par exemple aux mêmes facteurs de risques ou en calculant les prix de produits financiers à partir du même cadre de base. Typiquement, les prix des options simples sont calculés par le modèle de Black-Scholes-Merton, ce qui suggère que la volatilité implicite calculée à partir du prix de ces options reflète bien l'estimation de la volatilité par les acheteurs et vendeurs d'options.

Propagation de l'information Dans l'hypothèse de marché efficient, toute information se propage instantanément et se répercute sur tous les actifs ; les liens de cause à effet sont donc comprimés et invisibles. Or en pratique, on peut observer des délais entre les événements et les effets attendus sur les prix des actifs, au moins à cause de limitations techniques (accès à l'information, heures d'ouverture des marchés). Les actifs peuvent donc être dépendants de façon asynchrone avec un délai temporel.

Grande dimension

Le nombre d'échantillons de données est simple à prévoir : s'agissant de données journalières, nous disposerons d'environ 250 points par année d'historique. En revanche, la dimension des observations est plus difficile à contrôler. En se concentrant principalement sur l'Europe et après une sélection raisonnable, les données disponibles comptent à première vue 600 actions et une centaine des autres catégories d'actifs. Il est raisonnable d'éliminer les actions pour la description du marché, car elles sont individuellement à une échelle trop petite. 100 actifs de description paraissent en nombre abordable, mais il est peu probable que la représentation réduise chaque série temporelle en un scalaire. Il sera donc nécessaire de résumer l'information contenue dans les séries de façon efficace pour contrôler la dimension en entrée.

De plus, les données ne sont pas stationnaires, mais au mieux stationnaires par morceaux. Cela signifie que la quantité de données effectivement disponible doit être considérée "par morceau", ce qui divise les données autant qu'il y a de périodes identifiées. En tenant compte de cela, la tâche de prédiction peut être qualifiée de problème en grande dimension. Afin de dégager une analyse pratique des résultats, le modèle devra permettre de sélectionner les variables les plus importantes pour la prédiction.

2.2 Méthodes de représentation

Du point de vue de l'apprentissage statistique, on considère en général que la qualité de l'algorithme d'apprentissage a une priorité très supérieure au pré-traitement des observations, parce que la nature des données disponibles correspond déjà à une réalité pertinente (par exemple, les données cliniques d'un patient), ou encore grâce à la capacité des algorithmes à sélectionner les variables importantes et ainsi supporter une grande dimension en entrée. Néanmoins, comme

nous l'avons vu en partie 1.3.1, la théorie de l'apprentissage ne garantit la performance des algorithmes que dans le cadre iid, ou de dépendance faible dans les travaux récents. Il est donc nécessaire de résoudre en partie les difficultés montrées précédemment grâce à un traitement en amont. Outre les problématiques techniques, il est aussi essentiel que cette étape fournisse une description des séries temporelles interprétable en termes économiques pour les opérationnels.

La représentation des séries temporelles est un sujet riche en littérature dans de nombreux domaines comme l'analyse de données, le traitement de signal et l'apprentissage. La recherche d'une bonne représentation suit en général différents objectifs [KK03, KCHP04, KE07] :

— extraire l'information pertinente en éliminant le bruit,
— réaliser une bonne approximation du signal dans un but prédictif,
— segmenter le signal en périodes homogènes,
— réduire la dimensionnalité des données en la réduisant à un nombre abordable de descripteurs,
— produire des grandeurs adaptées à l'apprentissage, entre autres stationnaires.

Nous verrons ici quelques types d'approches courantes répondant à ces objectifs, auxquelles nous nous sommes intéressés dans une première démarche exploratoire.

2.2.1 Filtrage

Les méthodes de filtrage représentent le signal en le décomposant sur une famille de fonctions élémentaires prédéfinies, aussi appelée dictionnaire d'atomes. L'information du signal est ainsi résumée par un vecteur de coefficients associés à certains atomes, et peut être restitué en appliquant ces coefficients au même dictionnaire.

Analyse harmonique

Les méthodes d'analyse harmonique sont omniprésentes dans la compression de signaux comme les images et la musique. Un premier exemple est l'analyse de Fourier, où le dictionnaire est formé d'une base orthonormale de fonctions sinusoïdales caractérisées par la fréquence ω. Le coefficient de Fourier de fréquence ω d'un signal temporel y est sa convolution sur tout l'espace avec la sinusoïde correspondante :

$$\hat{f}(\omega) = \int y(t) e^{i\omega t} dt \ .$$

La compression (avec perte) proprement dite est effectuée en ne conservant qu'un nombre fini de coefficients de taille suffisante. Afin de tenir compte de la non-stationnarité du signal, la transformée de Fourier peut être effectuée sur des fenêtres contigues.

L'analyse par ondelettes [Dau88, Mey89, Mal00] généralise l'idée à des dictionnaires d'atomes localisés en temps et en fréquence. Les atomes s'obtiennent par translation et dilatation d'une fonction élémentaire $\psi \in L^2(\mathbb{R})$ de moyenne nulle, normalisée et centrée au voisinage de 0. La transformée de f en position u et échelle s s'écrit alors [Mal00] :

$$\langle y, \psi_{u,s} \rangle = \int_{-\infty}^{+\infty} y(t) \frac{1}{\sqrt{s}} \psi^\star \left(\frac{t-u}{s} \right) dt \ .$$

Cette transformation fournit une formulation générale permettant de s'adapter aux caractéristiques du signal par le choix de l'ondelette de base. De plus, elle est parcimonieuse (par rapport à l'analyse de Fourier par fenêtres) en rendant compte des perturbations locales, et donne une représentation multi-résolution. Des méthodes fondées sur les ondelettes ont été développées pour le filtrage de séries financières [Ram02], notamment à des fins prédictives [RSM02].

Singular Spectrum Analysis (SSA)

Singular Spectrum Analysis (cf. [GNZ01]) est une méthode répandue d'analyse des séries unidimensionnelles introduite dans les années 1980 [BP82, BK86]. Comparée aux autres méthodes de filtrage, elle se distingue par le fait que le dictionnaire n'est pas donné a priori, mais construit de façon non-paramétrique sur les données. La décomposition qui en résulte permet de représenter le signal à diverses résolutions, mais aussi d'extraire des composantes propres au signal, telles que la tendance, les composantes oscillatoires et le bruit. L'analyse d'un signal de longueur N s'effectue en deux étapes :

1. **Décomposition** : on construit la matrice de Hankel de taille $L \times (N - K + 1)$ de toutes les trajectoires de longueur L décalées, puis on effectue sa décomposition en valeurs singulières pour obtenir des matrices élémentaires associées aux valeurs singulières.
2. **Reconstruction** : on effectue des regroupements entre les matrices élémentaires afin d'obtenir un ensemble de signaux additifs (dont la somme est le signal originel) rangés par ordre de résolution.

Cette décomposition permet de débruiter le signal en ignorant les dernières composantes, d'extrapoler le signal et de détecter un changement dans la structure du signal. L'interprétation en tendance, oscillations et bruits n'est pas intrinsèque à l'analyse, mais des méthodes ont été proposées pour distinguer ces caractéristiques [GNZ01]. Enfin, la méthode n'a pas été généralisée jusqu'ici à une dimension quelconque, mais des extensions à deux dimensions ont été proposées [GU10].

2.2.2 Régularisation

La régularisation consiste à chercher une approximation du signal dans un espace fonctionnel $\mathcal{H}$ en respectant certaines bonnes propriétés, telles que la dérivabilité à un certain ordre. Il s'agit de minimiser un critère d'erreur entre le signal d'origine y et les fonctions $f \in \mathcal{H}$, avec une pénalité supplémentaire sur l'éloignement desdites propriétés. On se réfèrera au cadre de l'analyse de données fonctionnelles (cf. [Sil05]), où les méthodes de régularisation sont largement développées. On suppose disposer de données observées $\{(x_1, y_1), \ldots, (x_n, y_n)\} \in \mathcal{X} \times \mathcal{Y}$ suivant une fonction inconnue de $\mathcal{X}$ vers $\mathcal{Y}$. Formellement, le problème régularisé se pose comme suit :

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{n} L\left(y_i, f(x_i)\right) + \lambda J(f) \ ,$$

où L est une fonction de qualité de l'approximation, J est une fonction scalaire dans $\mathcal{H}$ et λ le paramètre de régularisation. $J(f)$ sera typiquement choisi pour être faible si f a les propriétés de régularité attendues et de valeur élevée si f s'en éloigne, et le paramètre λ quantifie l'importance donnée à cette régularité par rapport au critère d'erreur. La souplesse de la formulation donne lieu à une grande variété de méthodes et de considérations (cf. [Sil05, Chap. 5], [HTF09, Chap. 5], selon les choix sur l'espace fonctionnel, le critère d'erreur et la pénalité. Les méthodes de régularisation sont largement employées dans de nombreux domaines : traitement de signal, analyse de données fonctionnelles, apprentissage statistique et optimisation. Le lissage des fonctions par *splines* ou par noyaux en sont des exemples, et on en retrouve des applications en séries financières, par exemple dans [SL12]. En particulier, comme étudié dans [SS09], de nombreux travaux ont porté sur la prédiction de séries temporelles par SVM.

2.2.3 Approximation par morceaux

Sous notre hypothèse de stationnarité par périodes homogènes, il est naturel de considérer les méthodes d'approximation par morceaux, qui suscitent un intérêt certain dans l'analyse de séries

financières (voir par exemple [FCN06, FCLN08]). Ce type de méthode approche localement sur K segments distincts le signal $y(t)$ entre $t = 0$ et $t = T$. On suppose qu'il existe des points $0 = t_0 < t_1 < \ldots < t_K = T$, des bruits $\varepsilon_1, \ldots, \varepsilon_K$ et des fonctions déterministes $f_1, \ldots, f_K$, tels que pour $i \in \{1, \ldots, K\}$,

$$\forall t \in [t_{i-1}, t_i[,\ y(t) = f_i(t) + \varepsilon_i(t)\ .$$

Typiquement, les bruits sont centrés, et les fonctions f_i appartiennent à un espace fonctionnel paramétré, de sorte que chaque segment sera résumé par un petit nombre de paramètres. La fonction de base et les bruits peuvent être choisis selon le cadre physique et les propriétés attendues, et des contraintes peuvent être posées entre les segments, typiquement la continuité de la fonction résultante.

La difficulté de l'approximation par morceaux tient non pas à l'estimation sur chaque segment, mais à l'identification des segments [KCHP04]. En statistiques, le problème correspond à la détection des points de rupture (cf. [BN93]). On considère que les séries sont des concaténations de K segments générés chacun par un processus stochastique de distribution inconnue, et le problème consiste à identifier les points séparant chaque paire de segments consécutifs. Le problème est souvent traité avec des hypothèses sur le nombre de points de rupture, le type de distribution ou l'indépendance des données, mais une série de travaux récents [KR12, KR13, KR14] fournit des résultats non-paramétriques, pour un nombre de points de rupture inconnu, sur des données dépendantes entre elles.

Selon [KCHP04], on peut distinguer trois catégories de méthodes de segmentation :

— Fenêtre glissante : les segments sont construits de proche en proche en parcourant la série temporelle. À chaque nouveau point, un critère d'erreur détermine s'il doit être intégré au segment courant ou s'il s'agit d'un nouveau segment.

— *Top-down* : la série est initialement considérée comme un unique segment, puis divisée récursivement selon un critère.

— *Bottom-up* : la série est initialement morcelée en un nombre maximal de segments, qui sont ensuite fusionnés récursivement selon un critère.

2.2.4 Choix de la méthode

L'interprétabilité du résultat a motivé en grande partie une préférence pour les méthodes d'approximation par morceaux. D'une part, les approches par filtrage et par régularisation fournissent des outils puissants pour le traitement des séries, mais le choix des atomes et des critères d'erreur est en soi une question complexe. D'autre part, le lien entre la représentation et la réalité économique représentée peut s'avérer difficile à appréhender. L'approximation par morceaux fournit une représentation efficace et présente l'avantage déterminant que tous les aspects de la méthode sont interprétables en termes économiques :

— Il est généralement admis que les actifs financiers connaissent des périodes de stabilité et des changements de mode. La segmentation et les points de rupture correspondent à ces phénomènes de façon directe.

— Il est possible de choisir un modèle interprétable sur les segments ; par exemple, les approximations linéaire et exponentielle sont intuitives pour les actifs financiers. Cette modélisation permet de caractériser les périodes segmentées, par exemple en tendance et en variance des rendements.

2.3 Indexation des séries financières

Nous proposons de représenter les séries temporelles des actifs financiers par une estimation de tendances locales à la manière de la partie 2.2.3. Nous considérons donc le modèle suivant pour la série de prix $y(t)$ d'un actif donné : il existe une fonction f et un bruit centré ε tels que

$$y(t) = f(t) + \varepsilon(t),$$

en supposant que f est une fonction affine par morceaux, pas nécessairement continue, et que ε est stationnaire localement sur chaque morceau de f. Nous choisissons dans toute la suite une modélisation linéaire, mais le remplacement par une fonction exponentielle par morceaux, aussi courante en finance, ne présente pas de difficulté supplémentaire. Dans notre contexte, $f(t)$ représente la tendance linéaire de la série de prix, et le bruit $\varepsilon(t)$ reflète les variations des prix autour de cette tendance. En termes financiers, $f(t)$ modélise le rendement moyen et $\varepsilon(t)$ la volatilité de l'actif. On peut voir ce modèle comme une version modifiée des processus d'Ornstein-Uhlenbeck, définis par

$$dy(t) = -\theta(r - y(t))dt + \sigma dB(t)$$

où $B(t)$ est un mouvement brownien. Nous différons de ces processus principalement par le fait que nous supposons qu'il existe des changements de tendance dans $f(t)$ et que nous ne précisons pas a priori la distribution du bruit.

Comme montré précédemment, l'enjeu de la représentation par morceaux est l'identification efficace des périodes homogènes, soit de manière équivalente des points de rupture. Il s'agit de déterminer en chaque point s'il s'agit d'un changement de tendance ou seulement d'une fluctuation due à la volatilité. Notre procédure devra donc partitionner la série en un nombre K, inconnu en avance, de segments $I_k = [t_{k-1}, t_k[$, de façon adaptée au critère d'approximation linéaire par morceaux (ici l'erreur quadratique moyenne). Nous souhaitons par ailleurs contrôler la complexité de la segmentation, car une représentation tès fine donnerait certes une bonne approximation des données, mais ne fournirait pas une information pertinente sur la tendance et le bruit. Le critère d'adéquation de la représentation aux données doit donc être pénalisé par un critère de complexité, par exemple sur le nombre ou la taille des segments.

2.3.1 Formulation de l'approximation linéaire par morceaux

Supposons que l'on dispose de n points de données $\{y_0, \ldots, y_n\}$ du signal $y(t)$ sur l'intervalle de temps $[0, 1]$, et qu'il existe un certain nombre $K+1$ de points de rupture (incluant les bornes) $0 = t_0 < t_1 < \ldots < t_K = 1$ définissant les segments I_k. Le temps peut donc être indexé par j/n ($j \in \{0, \ldots, n\}$), et on note $\varepsilon_j = \varepsilon\left(\frac{j}{n}\right)$. On observe un signal bruité, linéaire par morceaux :

$$\forall j \in \{0, \ldots, n\},\ y_j = f\left(\frac{j}{n}\right) + \varepsilon_j\ .$$

La fonction cible $f^\star$ est entièrement définie par ses valeurs aux points de rupture : pour tout $k \in \{1, \ldots, K\}$,

$$\forall t \in [t_{k-1}, t_i[,\ f^\star(t) = f^\star(t_{k-1}) + \frac{f^\star(t_k)_- - f^\star(t_{k-1})}{t_k - t_{k-1}}$$

avec $f(t_i)_- = \lim_{\substack{t \to t_i \\ t < t_i}} f(t)$. Nous notons $\mathcal{F}$ l'ensemble des fonctions linéaires par morceaux, dans lequel nous chercherons les solutions.

La qualité d'une fonction estimée $\hat{f} \in \mathcal{F}$ se mesure par son erreur quadratique moyenne, ou risque quadratique :

$$R\left(\hat{f}\right) = \mathbb{E}_y\left[\int_0^1 \left(\hat{f}(t) - f(t)\right)^2 dt\right] .$$

Ce risque est estimé statistiquement sur les données par la moyenne des résidus au carré, ou risque empirique :

$$\hat{R}\left(\hat{f}\right) = \sum_{j=1}^{n} \left(y_j - \hat{f}(j/n)\right)^2 .$$

Afin de contrôler la complexité de la solution, nous ajoutons une pénalité proportionnelle au nombre K de segments dans $\hat{f}$. Le critère final est le risque empirique régularisé :

$$\hat{R}_{pen}\left(\hat{f}, \lambda\right) = \sum_{j=1}^{n} \left(y_j - \hat{f}(j/n)\right)^2 + \lambda K,$$

où $\lambda > 0$ est un paramètre de régularisation quantifiant le niveau de compromis entre l'approximation et la complexité.

Enfin, la fonction optimale est le minimiseur du risque empirique régularisé :

$$\hat{f} = \underset{f \in \mathcal{F}}{\operatorname{argmin}}\ \hat{R}_{pen}\left(\hat{f}, \lambda\right) .$$

Dans le cas où la segmentation est connue, le problème revient simplement à estimer la meilleure droite dans chaque intervalle I_i à partir des données. Par additivité du risque, il suffit de décomposer le risque empirique en K parties :

$$\hat{R}\left(\hat{f}\right) = \sum_{k=1}^{K} L(I_k) \text{ , avec } L(I_k) = \sum_{j:\ j/n \in I_k} \left(y_j - \hat{f}(j/n)\right)^2 .$$

La minimisation de chaque $L(I_k)$ est réalisée par une droite de régression aux moindres carrés. $\hat{f}$ est définie sur chaque I_k par

$$\forall t \in I_k, \hat{f}(t) = \alpha_k + \beta_k t .$$

Soit $|I_k|$ est le nombre de points de données dans l'intervalle I_k. On pose les grandeurs statistiques suivantes des données :

$$\begin{aligned}
\bar{t}_k &= \frac{1}{|I_k|} \sum_{j:\ j/n \in I_k} j/n , \\
\bar{y}_k &= \frac{1}{|I_k|} \sum_{j:\ j/n \in I_k} y_j , \\
\gamma_k &= \frac{1}{|I_k|} \sum_{j:\ j/n \in I_k} (j/n - \bar{t}_k)(y_j - \bar{y}_k) , \\
s_k^2 &= \frac{1}{|I_k|} \sum_{j:\ j/n \in I_k} (j/n - \bar{t}_k)^2 , \\
u_k^2 &= \frac{1}{|I_k|} \sum_{j:\ j/n \in I_k} (y_j - \bar{y}_k)^2 .
\end{aligned}$$

Les coefficients de régression sont :

$$\alpha_k = \bar{y}_k - \beta_k \bar{t}_k \qquad \text{et} \qquad \beta_k = \gamma_k / s_k^2$$

et l'erreur résultant de la régression s'écrit : $L^\star(I_k) = |I_k| \left(u_k^2 - \frac{\gamma_k^2}{s_k^2}\right)$.

2.3.2 Segmentation par arbres

Avec ce critère de régression aux moindres carrés, nous proposons de partitionner les séries au moyen d'une structure d'arbres binaires.

Top-down : CART

L'algorithme CART (*Classification and Regression Trees*) est une procédure célèbre de partitionnement récursif sous forme d'arbre binaire, introduite par Breiman, Friedman, Olshen et Stone [BFOS84]. Nous en présentons ici une utilisation non-supervisée, et décrirons la méthode plus en détail dans le chapitre 3, dans le cadre de l'apprentissage supervisé. Il s'agit d'un partitionnement *top-down*, car il considère initialement un unique segment, puis aboutit à une partition raffinée. Il est composé de deux étapes.

Construction de l'arbre (Figure 2.4)
La racine de l'arbre est définie comme l'intervalle complet $I_{0,1}$, caractérisé par l'erreur $L^\star(I_{0,1})$ issue de la régression aux moindres carrés sur cet intervalle. On cherche la meilleure division divisant $I_{0,1}$ en deux intervalles $I_{1,1}$ et $I_{1,2}$, qui sont les nœuds fils. Le critère de division est l'amélioration de l'erreur entre la racine et les nœuds fils, c'est-à-dire :

$$\Delta L(I_{0,1}, I_{1,1}, I_{1,2}) = L^\star(I_{0,1}) - L^\star(I_{1,1}) - L^\star(I_{1,2}) \ .$$

On choisit le point de rupture maximisant cette amélioration. À chaque itération de l'agorithme, il s'agit donc de calculer exhaustivement le critère ΔL pour chaque nœud terminal (intervalle non-divisé) et chaque division possible, puis sélectionner la division ayant le ΔL maximal. On notera $I_{j+1,2k-1}$ et $I_{j+1,2k}$ les nœuds fils de $I_{j,k}$. On obtient ainsi une suite de P divisions dans l'ordre décroissant d'amélioration avec leurs points de division $\{x_1, \ldots, x_P\}$ associés, qui ne sont généralement pas dans l'ordre chronologique.

Dans cette étape, on ne tient pas compte du paramètre de complexité λ, car il s'agit de construire l'arbre en entier jusqu'à la partition la plus fine. En pratique, on définit un critère d'arrêt supplémentaire, par exemple sur la taille minimale m des intervalles. Dans ce cas, on ne considère à chaque itération que les divisions dont les nœuds fils contiennent au moins m éléments. Cette condition d'arrêt permet notamment de réduire le temps d'exécution en ignorant d'emblée les divisions non-désirées.

Élagage des branches (Figure 2.5)
À partir de l'arbre complet $\mathcal{T}$, cette étape cherche le sous-arbre optimal $\mathcal{T}^\star$ minimisant le risque empirique régularisé

$$\hat{R}_{pen}\left(\mathcal{T}', \lambda\right) = \sum_{I \in \mathcal{L}(\mathcal{T}')} L(I) + \lambda |\mathcal{L}(\mathcal{T}')|$$

où $\mathcal{L}(\mathcal{T}')$ est l'ensemble des feuilles (nœuds terminaux) de l'arbre $\mathcal{T}'$ et $|\mathcal{L}(\mathcal{T}')|$ le nombre de feuilles. L'algorithme procède en calculant le nombre $K \leq P$ de divisions à conserver dans l'ordre décroissant d'amélioration d'erreur. On pose $\{\Delta L_1, \ldots, \Delta L_P\}$ les améliorations d'erreurs des divisions successives, et par convention $\Delta L_0 = 0$. On remarque de plus que chaque division ajoute une feuille dans l'arbre. La minimisation est alors équivalente à une optimisation très simple :

$$\min_{\substack{K \\ 0 \leq K \leq P}} \left[-\sum_{k=0}^{K} \Delta L_k + \lambda(K+1) \right] \ .$$

Les points de rupture (si $K > 0$) sont alors les $\{x_1, \ldots, x_K\}$ ré-arrangés dans l'ordre croissant de valeur.

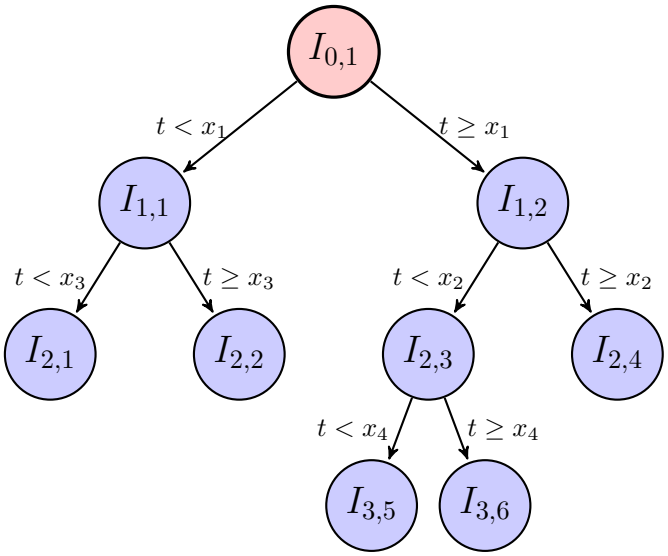


FIGURE 2.4 – Construction de l'arbre de partitionnement par CART avec 4 points de rupture x_1, x_2, x_3, x_4

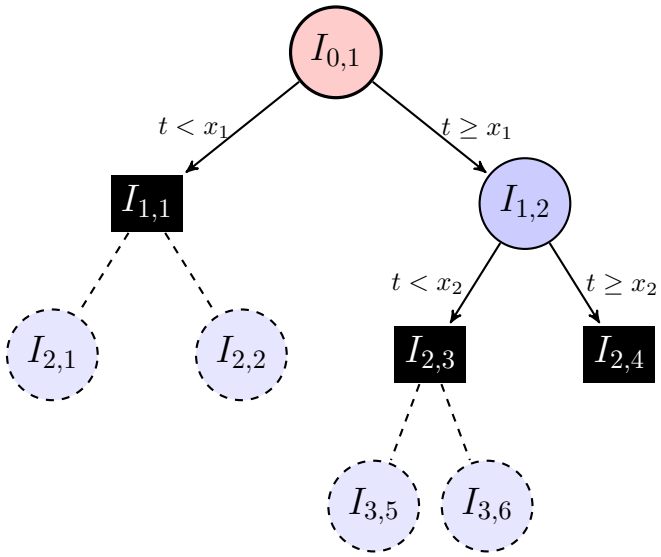


FIGURE 2.5 – Élagage par CART : partition formée de $I_{1,0}$, $I_{2,2}$ et $I_{2,3}$

La construction de l'arbre entier permet une grande flexibilité de l'élagage, qui peut être testé sans répéter la première étape. Un critère de complexité alternatif peut être un seuil minimal sur l'amélioration ΔL, à la manière des procédures d'optimisation.

Bottom-up : Coifman-Wickerhauser dyadique

Nous nous sommes aussi intéressés à une approche *bottom-up* dyadique issue de la méthode de Coifman-Wickerhauser [CW92], à laquelle nous nous réfèrerons en tant que "DCW". L'algorithme procède par fusion récursive de la partition la plus fine à une partition optimale.

En considérant initialement la partition la plus fine possible constituée de 2^Q intervalles de taille minimale m (typiquement $m = 2$), on fusionne les paires d'intervalles voisins $\{I_{Q,2q-1}, I_{Q,2q}\}$ en $I_{Q-1,q} = I_{Q,2q-1} \cup I_{Q,2q}$ si la fusion améliore le critère d'erreur pénalisé, c'est-à-dire si la gran-

deur suivante est strictement positive :

$$\Delta C\left(I_{Q,2q-1}, I_{Q,2q}, I_{Q-1,q}\right) = L^{\star}(I_{Q-1,q}) - L^{\star}(I_{Q,2q-1}) - L^{\star}(I_{Q,2q}) - \lambda \ .$$

Ce critère est différent de celui du ΔL précédemment utilisé pour CART du fait que la complexité est intégrée. Il faut noter que la fusion a lieu même si la régression sur l'intervalle union a une erreur supérieure – ce qui est en particulier inévitable si la partition démarre avec des intervalles de deux points – mais exige que la perte d'erreur totale soit compensée par le gain en complexité λ. L'autre différence majeure avec CART est le fait que l'algorithme ne recherche pas exhaustivement la séparation optimale, mais fusionne simultanément à chaque itération toutes les paires satisfaisant le critère. L'approche dyadique implique que les paires sont prédéfinies et qu'il n'est possible de fusionner que les intervalles voisins de même taille.

En termes d'arbres, la procédure revient à démarrer avec un arbre binaire équilibré dont les feuilles sont les intervalles consécutifs de taille minimale, puis fusionner itérativement à profondeur décroissante les feuilles en leur parent commun.

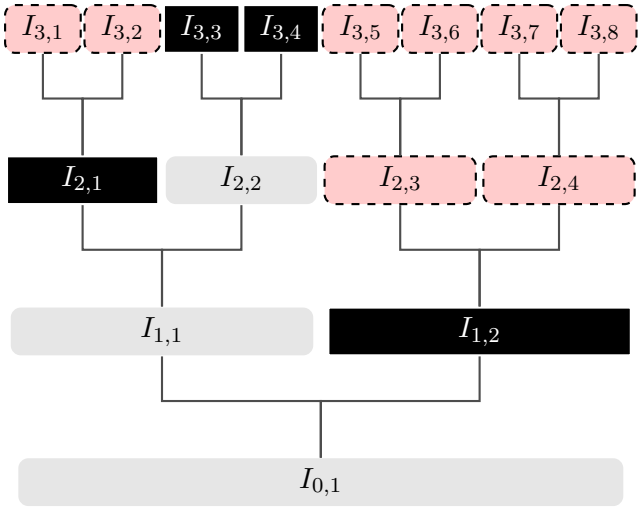


FIGURE 2.6 – Segmentation par Coifman-Wickerhauser dyadique : partition formée par $I_{2,1}$, $I_{3,3}$, $I_{3,4}$ et $I_{1,2}$

Comparaison

Chaque méthode présente ses avantages et inconvénients. Ces différences détermineront les cas d'utilisation et les qualités que l'on peut attendre de chacun.

Complexité Ces procédures sont très différentes en complexité : CART est beaucoup plus coûteuse que DCW. Considérons la régression sur un segment prédéfini comme une opération unitaire. DCW réalise au plus n opérations de fusion, soit une complexité de $\mathcal{O}(n)$, tandis que CART réalise n opérations de division d'intervalles par itération, et parcourt toute la profondeur de l'arbre, soit une complexité en $\mathcal{O}(n^2)$ pour l'étape de construction. Certes, il est possible d'optimiser l'implémentation en gardant en mémoire les résultats des itérations, mais cela ne change pas fondamentalement la complexité. De plus, CART réalise dès la première itération au moins autant d'opérations que DCW en tout.

Qualité de l'estimation La contrepartie majeure à la rapidité de DCW est sa limitation dans la structure représentable. L'algorithme dyadique produit des segmentations dont la taille

est mécaniquement une puissance de 2. Cette rigidité ne permet pas de respecter la structure interne de la série, et peut altérer la qualité de la segmentation par rapport au signal original. Non seulement l'approximation peut être moins bonne, mais des artefacts peuvent apparaître en exhibant des segments qui n'ont pas lieu d'être. Typiquement dans la figure 2.6, le résultat pourrait correspondre à seulement deux périodes homogènes en réalité, qui seraient $I_{2,1} \cup I_{3,3}$ et $I_{3,4} \cup I_{1,2}$.

Stabilité Chaque méthode présente une forme différente d'instabilité. Comme CART est un algorithme récursif optimisant précisément chaque division, la procédure est sensible aux valeurs des données, dont les changements sont susceptibles de modifier radicalement la structure de l'arbre. Cependant, CART est stable par rapport aux ajouts et suppressions de données aux extrémités, ce qui signifie notamment que l'on retrouve les mêmes points de rupture intermédiaires lorsque l'on applique la procédure à des fenêtres glissantes de données.
À l'inverse, DCW est robuste aux changements de valeurs car la fusion des intervalles ne dépend que des résidus, qui sont réguliers par rapport aux variations des données. La stabilité temporelle est en revanche un défaut pour l'interprétabilité des résultats, car la segmentation, et donc l'analyse économique, varie vraisemblablement lorsque l'on déplace la fenêtre d'observation. De plus, les descripteurs extraits par DCW peuvent différer sensiblement entre des dates proches.

Le choix de l'une ou l'autre méthode dépendra des contraintes : afin produire rapidement les segmentations dans le processus de prédiction, on pourra préférer DCW pour le gain en complexité, tandis que pour décrire finement le marché et son évolution, il faudra une segmentation performante par CART. Les jeux de paramètres utilisés dans chaque algorithme devront aussi être ajustés pour contrôler la complexité.

2.3.3 Input et output

La segmentation restitue les informations suivantes :
— les points de rupture $\{t_1, \dots, t_K - 1\}$ et les segments homogènes $\{I_1, \dots, I_K\}$ correspondants,
— la durée $\{\tau_1, \dots, \tau_K\}$ de chaque période homogène,
— la pente $\{\beta_1, \dots, \beta_K\}$ sur chaque segment,
— la variance des résidus $\{v_1, \dots, v_K\}$ sur chaque segment, estimée par les $L^\star(I_k)$,
— des moments d'ordre supérieur, si la taille des segments permet de les estimer.

Afin de travailler avec des grandeurs comparables entre actifs, il est d'usage de considérer pour les acifs financiers la tendance (rendement moyen) et la volatilité, définis comme la pente et l'écart-type rapportés au prix, par unité de temps :

$$r_k = \frac{1}{\tau_k}\frac{\beta_k}{\bar{y}_k} \qquad \text{et} \qquad \sigma_k = \frac{1}{\tau_k}\frac{\sqrt{v_k}}{\bar{y}_k} \ .$$

Nous appellerons de plus chaque segment un *régime*.

Vecteurs descripteurs - input

On considère, dans la vue globale, un ensemble de D actifs sélectionnés pour décrire le marché, une date t, et une fenêtre d'observation T sur les données, c'est-à-dire l'intervalle $[t - T + 1, t]$.

Pour chaque actif d, nous proposons d'effectuer la segmentation et de conserver uniquement les K' derniers segments afin de rendre compte du comportement récent et de contrôler la dimension finale des données. Nous retenons donc plusieurs vecteurs de taille K' :

— les tendances $r_t^d = (r_{K-K'+1}, \ldots, r_K)_d$,
— les volatilités $\sigma_t^d = (\sigma_{K-K'+1}, \ldots, \sigma_K)_d$,
— les durées $\tau_t^d = (\tau_{K-K'+1}, \ldots, \tau_K)_d$, que nous choisissons d'ignorer dans un premier temps.

Le vecteur descripteur final à la date t, qui servira comme observation en entrée du problème de prédiction, est la concaténation des ces vecteurs en un vecteur de taille $2DK'$:

$$X_t = \left(r_t^1, \ldots, r_t^D, \sigma_t^1, \ldots, \sigma_t^D\right) .$$

Afin de constituer une base données sur l'ensemble des dates disponibles, cette procédure d'extraction des descripteurs est répétée par fenêtre glissante.

On peut remarquer qu'il n'y a pas de garantie que la segmentation trouve plus d'une période, donc il est prudent de choisir $K' = 1$ pour l'apprentissage, à moins que l'algorithme d'apprentissage soit robuste aux données manquantes. Néanmoins, l'information sur le régime $K - 1$ peut être utile car elle rend compte du dernier changement de tendance.

En pratique, on pourra choisir en détail les composantes que l'on souhaite garder pour chaque actif, selon que l'on considère qu'elles font sens ou non en particulier.

Objectif à prédire - output

Nous nous intéressons à la prédiction des rendements d'un actif cible, tel qu'une action ou un indice. Étant donné le prix S_t, le rendement linéaire $R_{t,h}$ à l'horizon h est défini par :

$$R_{t,h} = \frac{S_{t+h}}{S_t} - 1 .$$

Nous considérons que la prédiction en valeur du rendement est un problème difficile, et nous nous concentrerons sur une sortie plus robuste, qui est la direction future de l'actif, soit le signe du rendement. Il s'agira donc d'un problème de classification avec $\mathcal{Y} = \{-1, 1\}$. Supposons que l'on dispose de segmentations en t et en $t + h$ pour cet actif, et donc des dernières tendances segmentées, notées $\tilde{r}_t$ et $\tilde{r}_{t+h}$. Plusieurs sorties Y_t sont possibles :

— le signe du rendement à l'horizon h : $Y_t = \text{sgn}(R_{t,h})$,
— le signe de la dernière tendance segmentée à l'horizon h : $Y_t = \text{sgn}(\tilde{r}_{t+h})$,
— le changement de tendance : $Y_t = \mathbb{1}\{\text{sgn}(\tilde{r}_{t+h}) \neq \text{sgn}(\tilde{r}_t)\}$.

Nous testerons en premier lieu nos méthodes sur le premier objectif, qui est le plus simple. Les autres objectifs semblent plus robustes du fait de la segmentation, mais le lien avec le prix courant (afin de prendre une décision d'investissement) est moins direct, ce qui implique des post-traitements sur le résultat de prédiction. Des raffinements multi-classes peuvent aussi être envisagés pour discrétiser les rendements en niveaux d'intensité. Cela correspond à une pratique courante dans les recommandations (forte hausse, hausse, neutre, baisse, forte baisse). Cependant, cette modification rend le problème plus difficile en morcelant les données, et aussi plus complexe car il faut choisir la bonne manière de définir les classes.

2.4 Exemples

2.4.1 Segmentation simple de séries

Les figures 2.7 et 2.8 illustrent les deux méthodes sur le cours du dollar contre l'euro sur deux années d'historique, et montrent les approximations linéaires trouvées, avec les résidus et la volatilité. Comme prévu, CART approche mieux la série avec des résidus plus faibles et des points de rupture mieux placés. Cependant, CART est beaucoup plus coûteux : la segmentation simple

de la figure 2.8 a nécessité plusieurs secondes alors que le temps d'exécution de DCW est de l'ordre du dixième de seconde. Pour générer les données d'observation de la prédiction, il faudra potentiellement segmenter un grand nombre de fenêtres de la série temporelle (250 par année de données) sur une centaine d'actifs. Le coût de CART (estimé sous ces conditions à plusieurs heures) est donc pénalisant dans le cadre de construction de modèle en temps réel. Néanmoins, il est possible en pratique d'alléger le coût marginal de la segmentation en sauvegardant les partitions. Les partitions déjà réalisées dans le passé sont donc chargées depuis le disque plutôt que d'être re-calculées, ce qui optimise grandement le coût de cette étape dans une utilisation fréquente. Dans ce cas, le coût de la segmentation est en majeure partie le temps de chargement des données enregistrées.

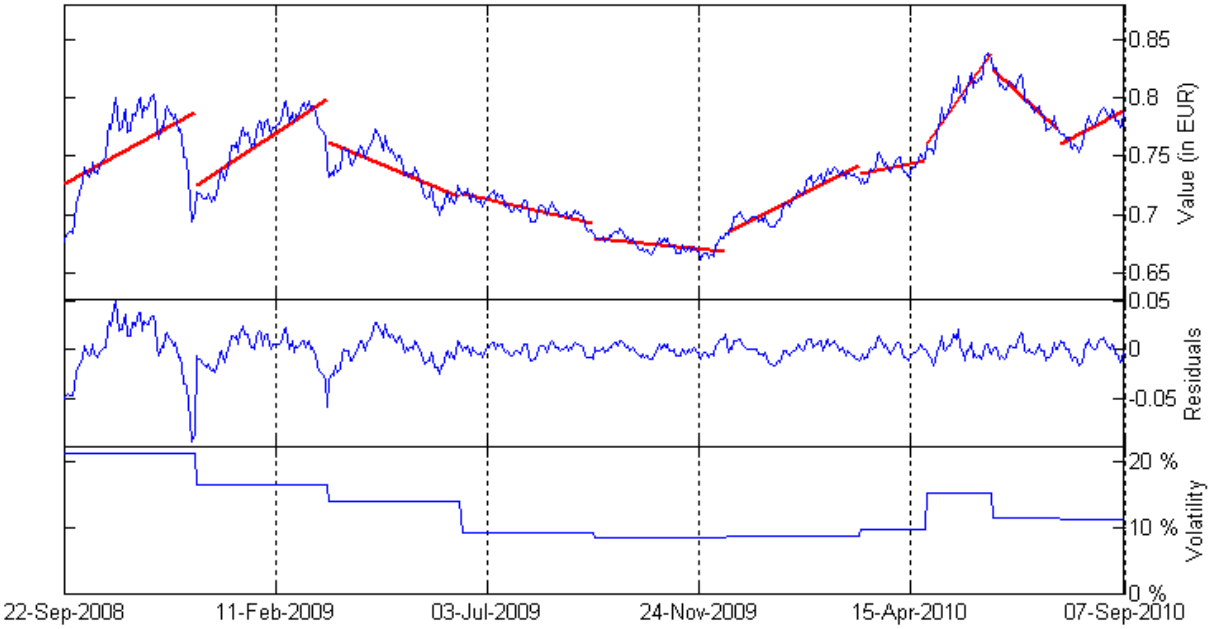


FIGURE 2.7 – Segmentation du taux de change dollar/euro avec DCW

2.4.2 Clustering en tendance et volatilité

La représentation en tendance et volatilité permet de donner une analyse descriptive des actifs en apprentissage non-supervisé, de façon indépendante de la prédiction. Une application directe est le clustering dans le plan volatilité/tendance. En choisissant une fenêtre d'observation, nous segmentons (une seule fois) les séries de prix des constituants d'un indice tel que l'EURO STOXX 50. Nous considérons que ces constituants représentent le marché des actions de la zone euro, et que leurs tendances et volatilités sont représentatives de différents régimes de marchés. Chaque segment de chaque action est un point dans le plan tendance/volatilité. Une méthode de clustering peut permettre de distinguer et caractériser les régimes. La figure 2.9 montre un clustering en trois régimes par K-Means des séries segmentées de 2000 à 2014.
Pour rappel, l'algorithme K-Means (cf. [HTF09, Chap. 14]) cherche itérativement un clustering des points en K groupes en initialisant K centroïdes de clusters aléatoirement, puis en répétant la procédure :

1. attribuer à chaque point le label de cluster correspondant au centroïde le plus proche,
2. mettre à jour les centroïdes comme les barycentres des points de chaque cluster,

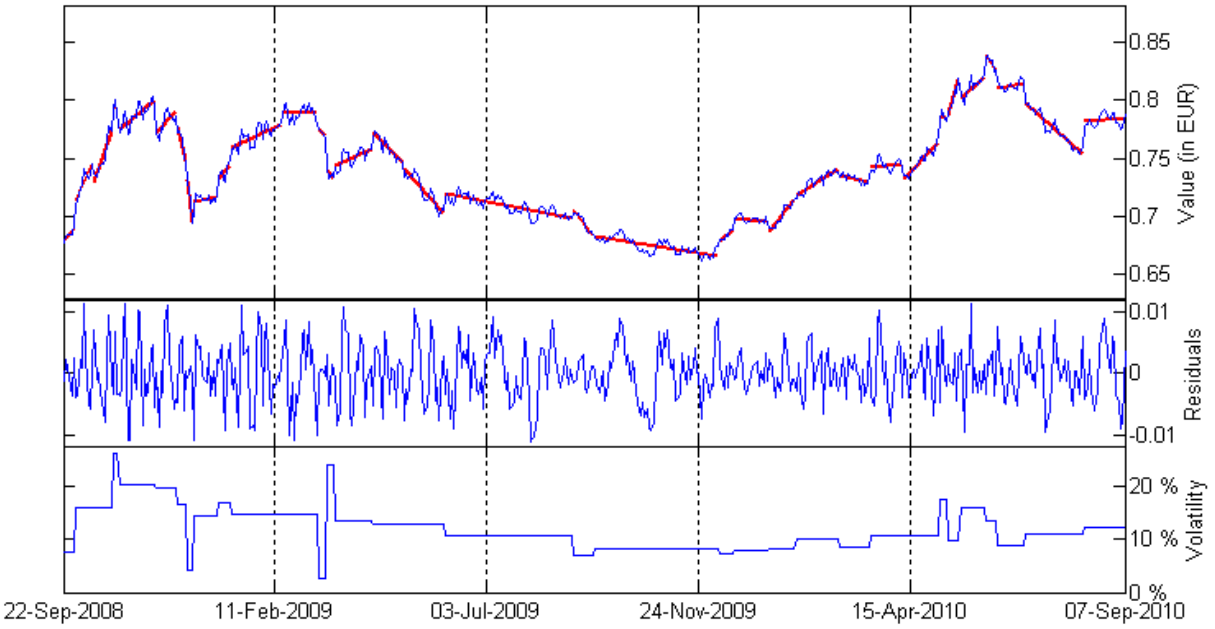


FIGURE 2.8 – Segmentation du taux de change dollar/euro avec CART

jusqu'à ce que les centroïdes et le clustering ne changent plus. L'algorithme trouve des optima locaux, donc il est en pratique répété pour sélectionner la meilleure solution.
La figure 2.9 montre un clustering assez proche de l'intuition : le cluster le plus peuplé correspond à un "ventre mou' de tendances et volatilités faibles observées pendant un régime de marché stable, tandis que les clusters extrêmes sont des régimes de crise (à gauche) ou de croissance forte (à droite). La méthode choisie illustre une application possible donnant une interprétation économique concrète de la représentation, mais elle n'est probablement pas la plus pertinente. Par exemple, d'autres méthodes de clustering peuvent être envisagées, comme le clustering spectral [vL07]. Par ailleurs, le clustering dans l'espace tendance/volatilité peut donner des problèmes intéressants – non-traités ici – comme la prédiction des régimes de marché.

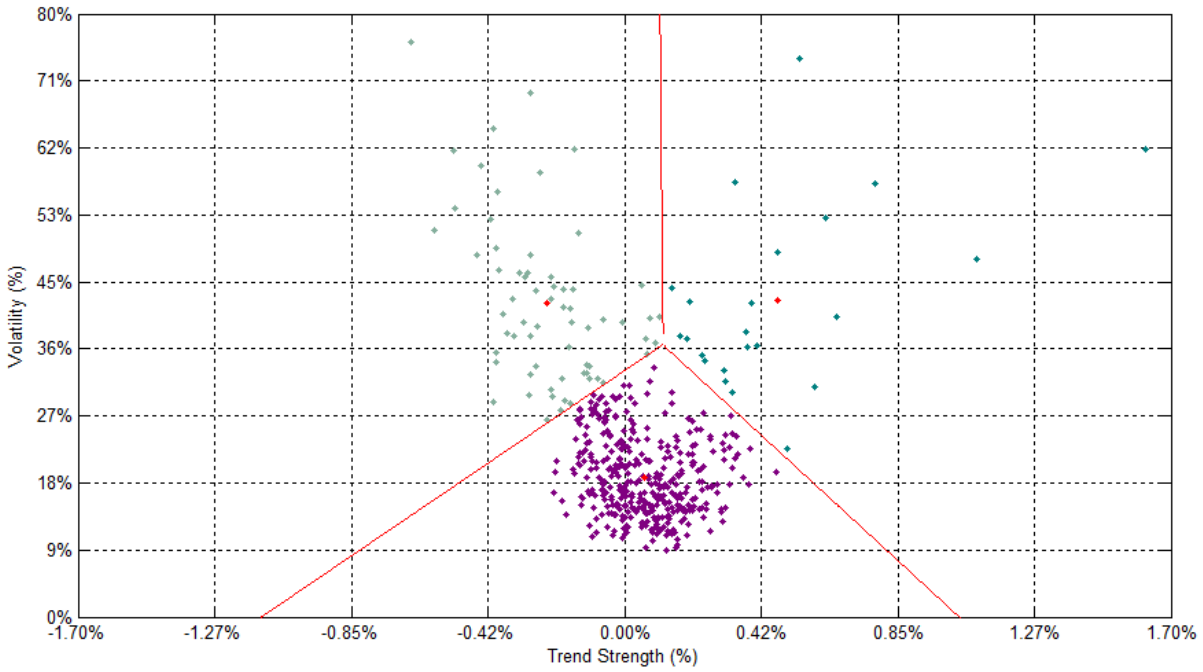


FIGURE 2.9 – Clustering dans le plan tendance/volatilité avec K-Means

Chapitre 3

Apprentissage et prédiction

La base de données construite dans le chapitre 2 fournit les entrées et sorties du problème de classification du signe des rendements. Parmi les nombreuses familles d'algorithmes, nous avons choisi d'explorer et mettre en œuvre une méthode d'arbres de décision par l'algorithme CART [BFOS84], agrégés par Random Forests [Bre01]. La procédure est adaptée au problème de prédiction car les arbres permettent de traiter les données en grande dimension en sélectionnant les variables macro-économiques d'intérêt. Le partitionnement de l'espace par les arbres de classification permet de rendre compte de relations complexes entre l'observation actuelle et la direction future, en déterminant automatiquement les régions de classes homogènes et en adaptant localement la finesse de la séparation entre les cas de baisse et de hausse des actifs. L'algorithme de construction segmente donc l'espace en régions de valeurs d'observation caractéristiques d'une même direction future pour l'actif cible. De plus, l'interprétation des règles de décision est essentielle afin de tirer des conclusions économiques et les confronter à l'expertise humaine. Random Forests agrège les prédictions d'un ensemble d'arbres entraînés par CART sur des plages différentes d'historique et des variables aléatoirement échantillonnées, d'où la notion de forêts aléatoires. L'agrégation des prédictions permet de rendre les résultats plus stables avec un meilleur pouvoir de généralisation.

L'analyse de la structure des arbres permet ensuite de restituer l'importance de chaque variable dans la prédiction et ainsi d'identifier les variables sur lesquelles le gérant de portefeuille doit se concentrer. Ces variables sont celles qui permettent de distinguer au mieux les baisses des hausses de l'actif cible. Outre l'analyse statique, l'évolution de l'importance dans le temps dans des arbres successifs donne une information dynamique de l'évolution des discriminants du rendement de l'actif.

Le problème du risque de prédiction se pose rapidement dès que l'on utilise les résultats des modèles pour prendre des décisions. En termes opérationnels, il s'agit de quantifier la fiabilité d'une prédiction dans un contexte donné ; autrement dit de manière plus formelle, estimer le risque de la prédiction sachant les observations, le résultat prédit et la structure du modèle. En disposant d'une telle estimation, la procédure de décision peut être modifiée pour rejeter la prédiction et s'abstenir de l'utiliser si le niveau de risque est trop élevé. Cette estimation donne un indicateur de la confiance que le décideur peut placer dans chaque résultat. Il s'agit d'une problématique opérationnelle importante, car d'une part, il n'est pas toujours nécessaire de décider dans un sens ou dans l'autre, d'autre part, il peut être plus coûteux de suivre la prédiction que de l'ignorer. Nous définissons deux types d'indicateurs : un indicateur de confiance mesurant la similarité entre les nouvelles observations et les observations d'entraînement, et un indicateur de stabilité quantifiant la variabilité de la prédiction lorsque l'observation est soumise à des perturbations. Nous mesurons la qualité de ces indicateurs par l'amélioration de la précision selon le seuil d'acceptation des prédictions.

3.1 Algorithme CART pour les arbres de décision

Les arbres de décision ont connu un large succès dans de nombreuses applications de classification [HTF09, Chap. 9], grâce à plusieurs avantages. Ce sont des méthodes non-paramétriques, qui n'utilisent pas de d'hypothèse a priori sur la distribution des variables à classer. Leur structure permet de produire des fonctions de prédiction globalement non-linéaires. Ils admettent aussi bien des données numériques que catégoriques, et peuvent fonctionner avec des données manquantes. Enfin, ils se traduisent en une succession de règles de décision et sont faciles à interpréter.

Un arbre binaire $\mathcal{T}$ est un graphe orienté acyclique (voir par exemple [Wes01]). Il est défini par ses n sommets $V = \{1, \dots, n\}$, appelés nœuds, et ses arêtes orientées $E \subset V \times V$, appelées branches. Lorsque deux nœuds de l'arbre sont reliés par une arête, le nœud d'origine de l'arête est appelé nœud père et le nœud destinataire est appelé nœud fils. Tous les nœuds ont un unique père, sauf un, qu'on appelle racine, n'ayant pas de nœud père. Chaque nœud p de l'arbre peut avoir soit exactement deux fils, auquel cas on parle de nœud interne, soit aucun, auquel cas on parle de nœud terminal ou feuille. On note de plus $\mathcal{L}(\mathcal{T})$ l'ensemble des feuilles de l'arbre $\mathcal{T}$.

Pour rappel, on suppose dans toute la suite que l'on dispose d'une base de données constituée d'un ensemble d'observations $\{X_1, \dots, X_T\} \subset \mathcal{X}$ et d'un ensemble de réponses $\{Y_1, \dots, Y_T\} \subset \mathcal{Y}$, où $\mathcal{X}$ et $\mathcal{Y}$ sont respectivement les espaces d'entrée et de sortie du problème d'apprentissage. Dans notre classification des signes du rendement, $\mathcal{X} = \mathbb{R}^D$ $(D > 0)$ et $\mathcal{Y} = \{-1, 1\}$ (que l'on représentera $\{-, +\}$ dans les schémas). Les $Z_i = (X_i, Y_i)$ sont supposées être des réalisations d'une variable aléatoire $Z = (X, Y)$ de loi inconnue. On notera de plus x_d $(d \in \{1, \dots, D\})$ la d-ième variable d'observation. Notre but est d'entraîner au moyen de la base de données une fonction de prédiction associant les sorties Y aux observations X.

3.1.1 Classification par arbres binaires

Les arbres de classification permettent de représenter l'espace d'entrée comme une partition formée de sous-espaces sur lesquels la réponse est constante. Étant donné un tel arbre, la prédiction de la réponse d'une nouvelle observation est très directe : il suffit de situer l'observation parmi les partitions et de retourner la classe associée. La question essentielle est : comment choisir la partition à partir des données ? Idéalement, l'algorithme doit déterminer automatiquement les régions étendues de classe homogène et les découpages plus fins aux endroits où la séparation entre les classes est plus complexe. Une méthode efficace consiste à utiliser les arbres binaires.

Définition

En classification, l'arbre de décision $\mathcal{T}$ partitionne l'espace d'entrée en sous-espaces et affecte une prédiction dans $\mathcal{Y}$ à chaque sous-espace. On appelle partition d'un ensemble $\mathcal{E}$ un ensemble fini de $n_{\mathcal{E}}$ sous-ensembles disjoints $(\mathcal{E}_i)_{i \in \{1,\dots,n_{\mathcal{E}}\}}$ dont l'union est $\mathcal{E}$:

$$\mathcal{E} = \bigsqcup_{i=1}^{n_{\mathcal{E}}} \mathcal{E}_i \ .$$

Les nœuds de l'arbre représentent chacun un sous-espace de $\mathcal{X}$ (sans former une partition), et on note $\mathcal{X}_i$ le sous-espace représenté par le nœud i. La racine représente $\mathcal{X}$. Le partitionnement est hiérarchique dans le sens où chaque nœud interne se divise en deux nœuds fils dont les sous-espaces représentés forment une partition de l'espace représenté par leur père. La partition de $\mathcal{X}$ par l'arbre $\mathcal{T}$ est définie par les feuilles de $\mathcal{T}$:

$$\mathcal{X} = \bigsqcup_{i \in \mathcal{L}(\mathcal{T})} \mathcal{X}_i \ .$$

Plus généralement, le partitionnement hiérarchique implique que les feuilles de tout sous-arbre $\mathcal{T}'$ contenant la racine de $\mathcal{T}$ forment une partition de $\mathcal{X}$.
La fonction de prédiction $f_{\mathcal{T}}$ associe à chaque feuille i de $\mathcal{T}$ une réponse $f_{\mathcal{T}}(\mathcal{X}_i) \in \mathcal{Y}$.

Le plus souvent, chaque division de nœud interne i représente une *règle de décision* univariée (d, α_i), c'est-à-dire une percolation vers l'un des nœuds fils selon la valeur relative de la co-variable x_d ($d \in \{1, \ldots, D\}$) par rapport à un seuil $\alpha_i \in \mathbb{R}$. On peut voir les règles de décision de ce type comme des hyperplans. On note $\mathcal{X}_{x_d<\alpha}$ et $\mathcal{X}_{x_d\geq\alpha}$ les deux demi-espaces séparés par l'hyperplan d'équation $x_d = \alpha$. La division d'un nœud i en nœuds fils a et b signifie que les sous-espaces représentés par a et b sont respectivement

$$\mathcal{X}_a = \mathcal{X}_i \cap \mathcal{X}_{x_d<\alpha} \qquad \text{et} \qquad \mathcal{X}_b = \mathcal{X}_i \cap \mathcal{X}_{x_d\geq\alpha} \ .$$

Les figures 3.1 et 3.2 donnent un exemple d'arbre de classification binaire et son partitionnement induit pour 2 variables et 4 règles de décision.

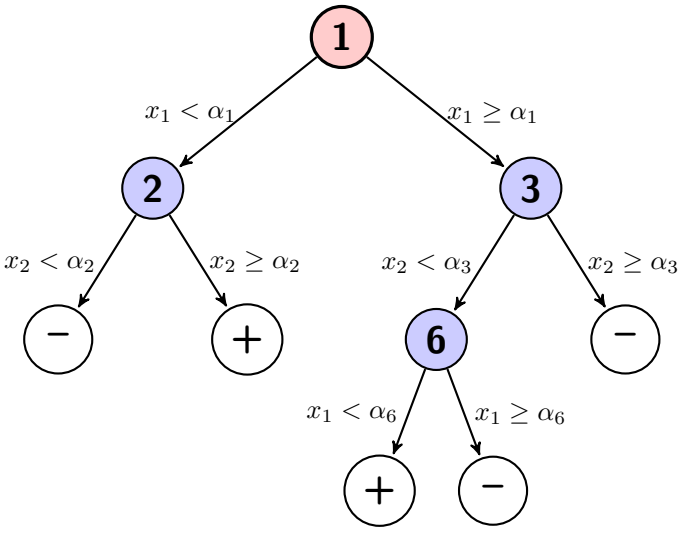


FIGURE 3.1 – Arbre de décision binaire avec $D = 2$

Étant donné un arbre de classification et une nouvelle observation $X' \in \mathcal{X}$, la prédiction Y' associée est obtenue par percolation dans l'arbre, en testant successivement à chaque nœud la co-variable correspondante et en suivant la branche indiquée par la règle de décision. Le label de la feuille d'arrivée donne la classe prédite.

Algorithmes de construction

La construction d'un arbre de décision optimal est en théorie un problème difficile. Il a été démontré que la recherche d'un arbre de décision minimal consistant avec les données d'entraînement était un problème NP-difficile [HJLT96]. Trouver l'arbre minimal équivalent d'un arbre de décision donné est aussi NP-difficile [ZB00]. La construction d'un arbre de décision optimal pour le nombre de tests nécessaires à la classification d'un observation nouvelle est quant à elle un problème NP-complet [HR76].

L'apprentissage d'arbres de classification fait donc appel à des heuristiques gloutonnes construisant l'arbre nœud par nœud. Comme nous l'avons vu pour la segmentation, la construction d'un arbre peut généralement être fait par divisions récursives de la racine aux feuilles (*top-down*, aussi appelée stratégie *divide and conquer*) ou par fusion des feuilles d'un arbre maximal (*bottom-up*).

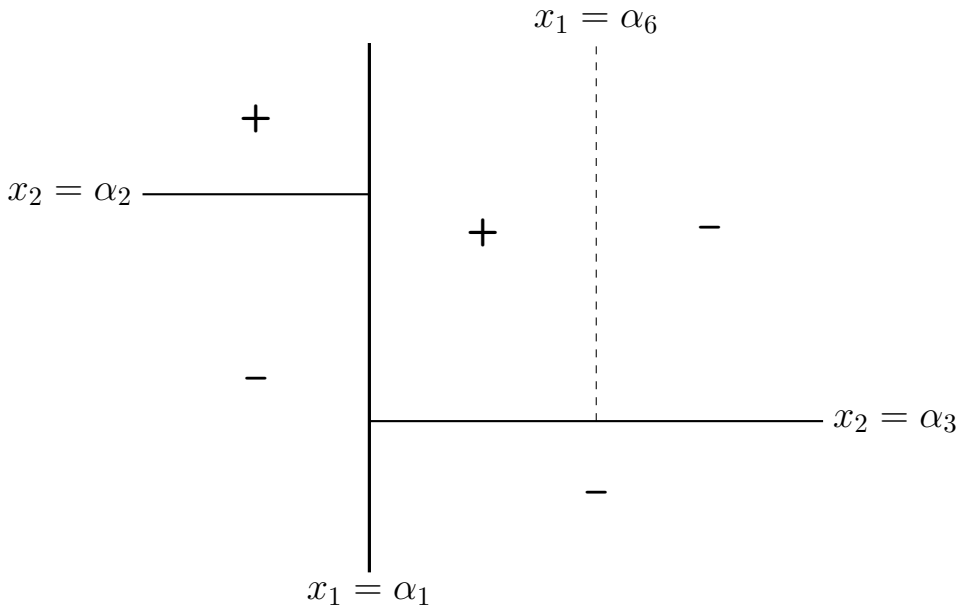


FIGURE 3.2 – Partitionnement de l'espace d'entrée par l'arbre de la figure 3.1

La première approche est largement dominante pour les arbres de classification. À chaque itération, un critère est calculé en chaque nœud à partir des données d'entraînement, et les nœuds sélectionnés sont divisés en deux nœuds fils, jusqu'à la vérification d'un critère d'arrêt.

Nous nous intéressons plus particulièrement au fonctionnement correspondant aux algorithmes les plus connus, qui sont C4.5 de Quinlan [Qui93] et CART de Breiman, Friedman, Olshen et Stone [BFOS84]. Ces algorithmes procèdent en deux étapes : une construction de l'arbre puis un élagage pour réduire la complexité.

Construction La construction démarre à la racine et divise récursivement les nœuds de l'arbre. Les règles de décision sont univariées, signifiant que la division dépend de la valeur d'une seule variable d'observation à chaque nœud. L'algorithme d'apprentissage recherche à chaque itération le meilleur nœud, la meilleure variable et le meilleur seuil de division dans cette dimension. Étant donné l'arbre intermédiaire $\mathcal{T}'$ construit à une itération donnée, la division est décidée par un critère g, qui reflète l'amélioration éventuelle de la performance empirique en classification. Il s'agit de maximiser une fonction faisant intervenir les données d'entraînement de la feuille i divisible, que l'on note $\tilde{Z}_i = (\tilde{X}_i, \tilde{Y}_i) = \{(X_j, Y_j) : X_j \in \mathcal{X}_i\}$, la dimension d dans laquelle l'espace est divisé, et le seuil α :

$$\max_{i \in \mathcal{L}(\mathcal{T}'),\ d \in \{1,\dots,D\},\ \alpha \in \mathbb{R}} \psi\left(\tilde{Z}_i, d, \alpha\right) .$$

Ce critère peut être de nombreuses sortes selon la fonction jugée appropriée pour optimiser la qualité prédictive [RM05] : gain d'information (C4.5), indice de Gini (CART), vraisemblance, distance, etc. La littérature semble montrer que le choix entre les différents critères n'influe pas considérablement sur la performance dans la plupart des cas, tant que le critère spécialise des nœuds vers des données de même classe. La sélection de la division optimale est naturelle, mais dans la littérature cette optimisation n'est pas nécessairement effectuée à la fois sur le nœud, la variable et le seuil. Par exemple, la division peut être dyadique [Don97, BSR04] ou aléatoire [Cis14]. Ne pas optimiser le seuil permet notamment de simplifier la division et donne des résultats théoriques tels que la consistance.

La construction s'arrête lorsque le critère d'arrêt est vérifié. Cette condition peut généralement être :

- Toutes les réponses du nœud sont de la même classe.
- La taille du nœud est en-dessous d'une taille limite m, ou toute division donnerait des fils de taille inférieure à m.
- La profondeur maximale est atteinte.
- Le critère d'amélioration est inférieur à un seuil $g_{\min}$.

Le label de prédiction des feuilles est typiquement la classe majoritaire :

$$\forall i \in \mathcal{L}(\mathcal{T}),\ f_{\mathcal{T}}(i) = \underset{c\in\mathcal{Y}}{\operatorname{argmax}} \sum_{y\in\tilde{Y}_i} \mathbb{1}_{y=c}\ .$$

Élagage La construction crée volontairement un arbre complet, ramifié, et potentiellement sensible au bruit dans les données, mais avec des critères d'arrêt simples. Afin d'éviter le sur-apprentissage par une partition trop fine, la phase d'élagage vient réduire la complexité de l'arbre en supprimant les sous-arbres non-pertinents. Cette phase tend à améliorer sa qualité de généralisation de l'arbre et sa clarté.
Supposons estimé (pendant la construction par exemple) le risque empirique $\hat{R}(\mathcal{T}')$ de tout sous-arbre $\mathcal{T}'$. L'élagage équivaut à trouver le sous-arbre optimal de l'arbre complet $\mathcal{T}$ pour le risque empirique pénalisé :

$$\min_{\mathcal{T}'} \hat{R}(\mathcal{T}') + \lambda\Omega(\mathcal{T}')$$

où $\Omega(\mathcal{T}')$ pénalise la complexité du sous-arbre et λ est un paramètre de régularisation quantifiant le niveau de compromis entre le risque empirique et la complexité. De façon heuristique, cette minimisation s'effectue en supprimant de bas en haut les feuilles de l'arbre complet, et en modifiant les nouveaux nœuds terminaux en leur affectant un label de classification. Diverses méthodes d'élagage ont été étudiées dans la littérature. On peut notamment citer :

- *Error-Based Pruning*, utilisé dans C4.5 [Qui93], qui estime un taux d'erreur fondé sur un intervalle de confiance statistique,
- l'élagage par remplacement des nœuds internes par la classe la plus fréquente sur un jeu de données distinct des données d'entraînement [Qui87],
- *Minimum Description Length Pruning* [QR89], mesurant la taille d'un arbre de décision par le nombre de bits nécessaires à son encodage,
- l'élagage des feuilles par ordre de gain en impureté, utilisé par CART, que nous détaillerons dans la partie suivante.

À cause de la construction hiérarchique des arbres, ceux-ci sont généralement sensibles aux valeurs des données d'entraînement. En effet, des changements de données faibles en norme euclidienne peuvent mener à des arbres très différents, car les modifications d'optima à tout nœud proche de la racine se répercutent sur une grande partie de l'arbre. L'instabilité des arbres est un problème pour l'interprétation des résultats. Certains travaux se sont penchés sur la construction d'arbres plus stables, notamment au moyen de l'élagage (notamment [Qui87]), mais on résout généralement le problème en combinant plusieurs arbres par une méthode d'agrégation de classifieurs faibles [Bre96], comme nous le verrons dans la partie 3.2.

3.1.2 Algorithme CART

L'algorithme CART (*Classification and Regression Trees* [BFOS84]) utilise un critère fondé sur l'impureté des nœuds de l'arbre. L'impureté d'un nœud mesure le degré de diversité des classes de sortie des données du nœud. La phase de construction a pour but la diminution de l'impureté.

Impureté

On définit formellement l'impureté comme suit. Pour tout nœud i, on note $n(i) = |\tilde{Z}_i|$ le nombre d'échantillons de données dans le nœud, et $p_c(i) = \frac{1}{n(i)} \sum_{y \in \tilde{Y}_i} \mathbb{1}_{y=c}$ la fréquence de la classe $c \in \mathcal{Y}$ dans le nœud.

Étant donnée une variable aléatoire y binaire dans $\{-1, 1\}$, distribuée selon la loi $\Pi = (\pi_{-1}, \pi_1)$, une mesure d'impureté est une fonction $\phi : [0,1]^2 \to \mathbb{R}_+$ satisfaisant les conditions :

— ϕ admet un minimum aux points $(0,1)$ et $(1,0)$,

— ϕ admet un maximum au point $\left(\frac{1}{2}, \frac{1}{2}\right)$ d'équiprobabilité,

— ϕ est strictement concave, symétrique dans chaque dimension et différentiable.

L'impureté du nœud i est estimée par l'impureté de sa distribution empirique des classes : $\phi\left(p_{-1}(i), p_1(i)\right)$. L'impureté est maximale lorsque le nœud contient autant de données dans chaque classe, et le nœud est totalement pur s'il ne contient que des données de la même classe. CART utilise l'indice de Gini, historiquement utilisé pour mesurer l'inégalité des revenus dans un pays :

$$\phi\left(p_{-1}(i), p_1(i)\right) = p_{-1}(i) \cdot p_1(i) = p_1(i)(1 - p_1(i)) \ .$$

D'autres fonctions d'impureté existent, comme :

— l'erreur de classification $\phi\left(p_{-1}(i), p_1(i)\right) = 1 - \max(p_{-1}(i), p_1(i))$,

— l'entropie $\phi\left(p_{-1}(i), p_1(i)\right) = -\left[p_{-1}(i) \log(p_{-1}(i)) + p_1(i) \log(p_1(i)\right]$, utilisé dans C4.5 [Qui93].

La figure 3.3 donne un aperçu comparatif de ces choix.

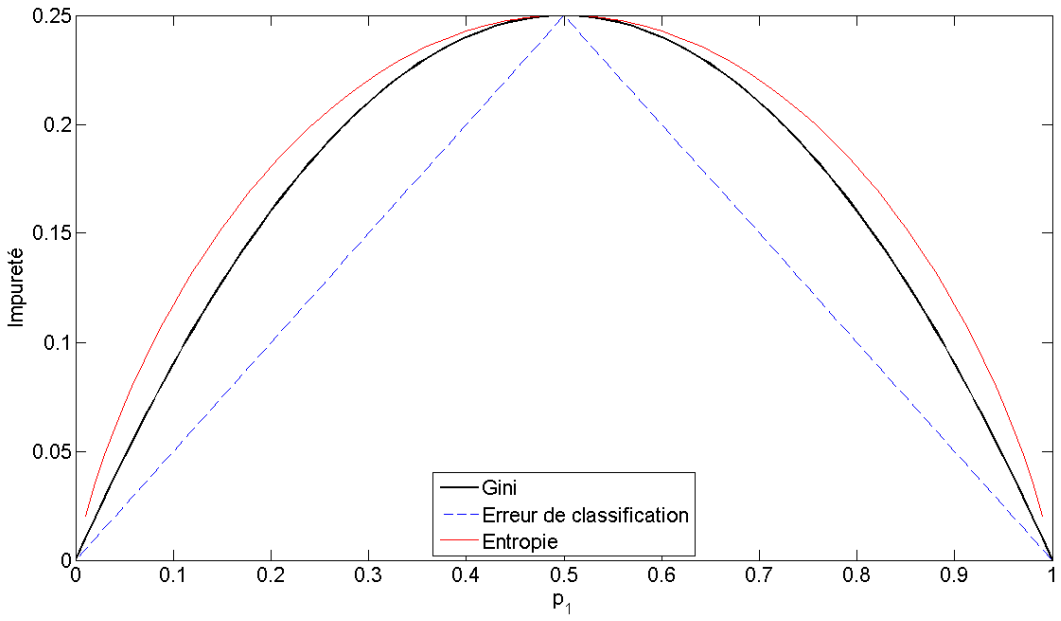


FIGURE 3.3 – Mesures d'impureté (mises à l'échelle de l'indice de Gini)

Nous notons plus spécifiquement dans la suite $G(i)$ l'indice de Gini du nœud i, et la fonction d'impureté est précisément l'indice normalisé par la taille du nœud :

$$g(i) = \frac{n(i)}{T} G(i) \ .$$

On étend naturellement cette impureté à tout arbre $\mathcal{T}$ comme l'impureté totale des feuilles :

$$g(\mathcal{T}) = \sum_{i \in \mathcal{L}(\mathcal{T})} g(i) \ ,$$

que l'algorithme de construction tend à minimiser.

Division par réduction d'impureté

Supposons un arbre $\mathcal{T}'$ construit jusqu'à une itération de l'algorithme. On considère pour chaque feuille i de $\mathcal{T}'$ sa division éventuelle en deux nœuds fils a et b, selon la variable d et avec un seuil α. Les nœuds a et b représentent respectivement les sous-espaces de $\mathcal{X}_i$ de part et d'autre de l'hyperplan défini dans $\mathcal{X}$ par l'équation $x_d = \alpha$. Le critère caractérisant la division est la décroissance de l'impureté :

$$\Delta g(i, d, \alpha) = g(i) - g(a) - g(b)$$

en sous-entendant que la définition de a et b découle de d et α. La division est réalisée au paramètre (nœud, variable, seuil) maximisant ce critère parmi les feuilles :

$$(i^\star, d^\star, \alpha^\star) = \underset{i \in \mathcal{L}(\mathcal{T}'),\ d \in \{1,\dots,D\},\ \alpha \in \mathbb{R}}{\operatorname{argmax}} \Delta g(i, d, \alpha) \ .$$

$\mathcal{T}'$ est augmenté des fils de $i^\star$ en un arbre $\mathcal{T}''$. On affecte à chaque fils un label de prédiction correspondant à la classe majoritaire du nœud fils. L'algorithme s'arrête si l'arbre est suffisamment ramifié (en nombre ou taille de feuilles) ou s'il n'y a plus d'amélioration possible, soit $\Delta g(i^\star, d^\star, \alpha^\star) \leq 0$.

Élagage

La procédure de construction fournit les informations suivantes :
- l'arbre final $\mathcal{T}$ après P divisions, et la suite des sous-arbres $(\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_P)$ construits successivement au cours des divisions,
- les suites de nœuds divisés $(p_1, \dots, p_P)$, les dimensions de division $(d_1, \dots, d_P)$ et les seuils $(\alpha_1, \dots, \alpha_P)$,
- la suite des critères de division optimale $(\Delta g_1, \dots, \Delta g_P)$, **dans l'ordre décroissant**.

La phase d'élagage cherche le sous-arbre optimal $\mathcal{T}^\star$ minimisant l'impureté totale pénalisée par la taille :

$$g_\lambda(\mathcal{T}') = \sum_{i \in \mathcal{L}(\mathcal{T}')} g(i) + \lambda |\mathcal{L}(\mathcal{T}')|$$

L'algorithme procède en calculant le nombre $K \leq P$ de divisions à conserver dans l'ordre décroissant d'amélioration d'impureté.

Comme pour la version non-supervisée de CART (partie 2.3), on remarque que chaque division ajoute une feuille dans l'arbre et on note par convention $\Delta g_0 = 0$. La minimisation est alors équivalente à une optimisation très simple :

$$K^\star = \underset{\substack{K \\ 0 \leq K \leq P}}{\operatorname{argmin}} \left[-\sum_{k=0}^{K} \Delta g_k + \lambda K \right]$$

et $\mathcal{T}^\star = \mathcal{T}_{K^\star}$. L'élagage peut aussi être effectué au moyen d'autres critères à la place de la réduction d'impureté, comme l'erreur d'entraînement ou l'erreur de validation sur un ensemble non utilisé.

Consistance

La consistance est une question courante pour tout algorithme d'apprentissage. Cependant, il existe peu de résultats théoriques sur la consistance de CART. Des inégalités oracle ont démontrées pour CART en version dyadique pour la régression par Donoho [Don97] en le liant avec la recherche de la meilleure base orthonormée. Le partitionnement dyadique consiste à

Algorithme 1 CART

Paramètres : $\lambda > 0, m > 0, M > 0$

Phase 1 : Construction

Initialisation : $\mathcal{T}_0 = (1, \emptyset)$, $k = 0$

repeat

Notation : $\mathcal{T}_k = (V_k, E_k)$.

[1] Calculer $(i^\star, d^\star, \alpha^\star) = \underset{i \in \mathcal{L}(\mathcal{T}'),\ d \in \{1,\dots,D\},\ \alpha \in \mathbb{R}}{\operatorname{argmax}} \Delta g(i, d, \alpha)$.

[2] Définir $\Delta g_k = \Delta g(i^\star, d^\star, \alpha^\star)$.

[3] Diviser le nœud $i^\star$ en deux nœuds fils a et b :

$$\mathcal{X}_a = \mathcal{X}_{i^\star} \cap \mathcal{X}_{x^{d^\star} < \alpha}$$
$$\mathcal{X}_b = \mathcal{X}_{i^\star} \cap \mathcal{X}_{x^{d^\star} \geq \alpha}$$

[4] Associer aux nouvelles feuilles les labels

$$y(a) = \underset{c \in \mathcal{Y}}{\operatorname{argmax}} \sum_{y \in \tilde{Y}_a} \mathbb{1}_{y=c}$$
$$y(b) = \underset{c \in \mathcal{Y}}{\operatorname{argmax}} \sum_{y \in \tilde{Y}_b} \mathbb{1}_{y=c}$$

[5] Créer l'arbre de la nouvelle itération : $\mathcal{T}_{k+1} = (V_k \cup \{a, b\}, E_k \cup \{(i^\star, a), (i^\star, b)\})$.

[6] $k \leftarrow k + 1$

until $|\mathcal{L}(\mathcal{T}_k)| > M$ **or** $\max_{i \in \mathcal{L}(\mathcal{T}_k)} |\tilde{Z}_i| \leq m$ **or** $\Delta g(i^\star, d^\star, \alpha^\star) \leq 0$

return $P = k - 1$, $(\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_P)$, $(\Delta g_1, \dots, \Delta g_P)$

Phase 2 : Élagage

[1] Calculer $K^\star = \underset{\substack{K \\ 0 \leq K \leq P}}{\operatorname{argmin}} \left[-\sum_{k=0}^{K} \Delta g_k + \lambda K \right]$.

[2] $\mathcal{T}^\star = \mathcal{T}_{K^\star}$.

return $\mathcal{T}^\star$.

diviser les nœuds en deux sous-espaces de taille égale, si on considère l'espace borné par les coordonnées extrêmales des données d'entraînement. D'autres résultats de consistance ont été établis pour la phase d'élagage [Nob02, GN03], et pour des algorithmes de construction d'arbres dyadiques de classification [BSR04].

3.1.3 Importance des variables

L'algorithme sélectionne les variables d'intérêt en même temps qu'il construit la fonction de prédiction. La structure d'arbre permet de calculer l'importance des variables dans le modèle. Dans la prédiction du signe du rendement des actifs, l'importance des variables est essentielle pour interpréter le modèle et comprendre quels sont les actifs observés (indices, taux, etc.) et les composantes (tendance, volatilité, durée) déterminantes dans la prédiction. L'utilisateur pourra ainsi porter une attention particulière aux variables explicatives les plus importantes car celles-ci sont les plus influentes sur le sens futur de l'actif prédit, tandis qu'il pourra ignorer les autres.

Dans CART, l'importance d'une variable d'entrée x_d dans l'arbre $\mathcal{T}^\star$ est définie comme la

somme des réductions d'impureté dans les nœuds où la variable est utilisée pour la division :

$$I_{\mathcal{T}^\star}(d) = \sum_{k=1}^{K^\star} \Delta g_k \ \mathbb{1}_{d_k=d} \ .$$

Les variables d'importances les plus élevées sont donc celles qui contribuent le plus à la discrimination entre les cas de hausse et de baisse de l'actif cible.

Afin de classer les variables par importance relative et de façon comparable entre deux arbres, on pourra aussi considérer le rang normalisé de la variable. En supposant un classement par ordre croissant de $I_{\mathcal{T}^\star}$, on définit le rang normalisé de la variable d de rang k_d par

$$\tilde{I}_{\mathcal{T}^\star}(d) = \frac{k_d}{D} \ .$$

3.2 Méthodes d'agrégation

Les arbres de classification sont par nature instables [Bre96, Die00] à cause de leur construction hiérarchique. En effet, des perturbations faibles sur les données d'entraînement peuvent mener à des fonctions de prédiction très différentes. Un moyen efficace de réduire cette variance est d'agréger les modèles. Les méthodes d'agrégation combinent un ensemble de fonctions de classification [Die00], appelées classifieurs faibles, afin de produire un classifieur plus performant. L'avantage des classifieurs faibles a été montrée et explorée dans les années 1990, notamment dans [Sch90]. L'idée est d'utiliser des fonctions de prédiction moins riches, mais plus simples, donc plus faciles et rapides à entraîner. Il s'agit d'accepter une moins bonne approximation de la fonction cible en échange d'une meilleure qualité d'estimation et une plus grande efficacité unitaire. La qualité d'approximation est ensuite obtenue en combinant des fonctions variées. L'agrégation permet notamment d'enrichir le pouvoir de généralisation par la diversité des classifieurs faibles ou de réduire la variance des modèles simples. Nous nous intéresserons ici aux méthodes prisées que sont le boosting [Fre95, FS97, SS99], le bagging (pour *Bootstrap Aggregating*, [Bre96]) et les forêts aléatoires (*Random Forests*, [Bre01]). Nous utiliserons cette dernière méthode dans notre procédure de prédiction.

3.2.1 Boosting

Le boosting a été introduit par Freund et Schapire dans [Fre95, FS97] initialement en tant qu'algorithme AdaBoost. AdaBoost construit itérativement des fonctions de classification en pondérant les observations dans le risque empirique et en augmentant à chaque itération le poids des échantillons mal prédits. Les fonctions sont ensuite agrégées par une somme pondérée, où les poids sont liés à l'erreur de chaque fonction.

Formellement, soit $\mathcal{F}$ l'ensemble des classifieurs faibles à agréger, et les données d'entraînement $\{Z_1, \dots, Z_T\} = \{(X_1, Y_1), \dots, (X_T, Y_T)\} \subset \mathcal{X} \times \mathcal{Y}$ en reprenant les notations de la partie 3.1. On attribue initialement une distribution ψ_1 uniforme sur les T échantillons de données : $\forall t \in \{1, \dots, T\},\ \psi_1(t) = \frac{1}{T}$.
À l'itération i de l'algorithme, la distribution est ψ_i et on entraîne une fonction de prédiction f_i minimisant le risque empirique pondéré, défini pour toute fonction $f \in \mathcal{F}$:

$$\hat{R}_i(f) = \sum_{t=1}^{T} \psi_i(t) \mathbb{1}_{f(X_t) \neq Y_t} \ .$$

On définit un poids w_i associé à cette itération à partir du risque empirique de f_i :

$$w_i = \frac{1}{2} \ln \left(\frac{1 - \hat{R}_i(f_i)}{\hat{R}_i(f_i)} \right) .$$

Ce poids sert à modifier la distribution comme suit :

$$\forall t \in \{1, \dots, T\}, \ \psi_{i+1}(t) = \frac{\exp(-w_i Y_t f_i(X_t))}{\sum_{t=1}^{T} \exp(-w_i Y_t f_i(X_t))} \psi_i(t) .$$

L'algorithme est arrêté au bout de $i_{\max}$ itérations, et la fonction de prédiction agrégée est :

$$f^\star = \sum_{i=1}^{i_{\max}} w_i f_i .$$

De nombreuses méthodes de boosting ont été proposées (dont les nombreuses versions spécialisées AdaBoost.[], GentleBoost [FHT00], MedBoost [Ké03], RankBoost [FISS03]), différant entre autres par :

— la forme du risque empirique selon l'objectif de prédiction en sortie (régression, classification, ranking, multi-classe, etc.)
— la pondération des échantillons, c'est-à-dire la modification de la distribution ψ,
— la pondération des prédicteurs faibles.

Le boosting a vite bénéficié de nombreuses études théoriques dans la littérature (voir [Sch99, Vay06]). On peut notamment voir AdaBoost comme une descente de gradient pour la minimisation du risque empirique [FHT00], où on utilise un substitut de la perte de la forme $\exp(-yf(x))$. On peut trouver des résultats de consistance dans [Bre00, MMZ02, LV04, LKS05, BT07].

Dans notre méthode de prédiction, on peut utiliser les arbres de classification de petite taille comme classifieurs faibles d'AdaBoost. Une manière proposée dans [HTF09, Chap. 10] est d'utiliser des arbres de même taille en choisissant la taille optimale par validation croisée. L'importance des variables du classifieur "boosté" est directement donnée par une somme pondérée sur les différents arbres.

3.2.2 Bagging

Le bagging, pour *Booststrap Aggregating*, a été introduit par Breiman [Bre96] pour améliorer la performance d'algorithmes de prédiction en agrégeant des fonctions entraînées sur des échantillons de *bootstrap*, c'est-à-dire des échantillonnages avec ou sans remise des données d'entraînement. Contrairement au boosting, il s'agit d'une méthode non-déterministe à cause du tirage aléatoire de données.

À chaque itération i, on tire un sous-ensemble (ou un échantillonnage avec remise) de n données d'entraînement, indexé par $\mathcal{I}_i = \{i_1, \dots, i_n\}$. On entraîne une fonction de prédiction f_i minimisant le risque empirique sur cet échantillon, défini pour toute fonction $f \in \mathcal{F}$ par :

$$\hat{R}_i(f) = \sum_{i \in \mathcal{I}_i} \mathbb{1}_{f(X_i) \neq Y_i} .$$

Au bout de $i_{\max}$ itérations, la fonction de prédiction agrégée est la fonction de prédiction moyenne :

$$f^\star = \frac{1}{i_{\max}} \sum_{i=1}^{i_{\max}} f_i .$$

Dans le cas de la classification, il s'agit du vote à la majorité.

Il a été montré que le bagging permettait de réduire la variance des algorithmes instables [Bre96, FH99], comme c'est le cas pour les arbres de classification. De plus, des résultats de consistance ont été trouvés pour le bagging appliqué à l'algorithme des plus proches voisins [HS05, BD10].

L'application du bagging aux arbres est directe : il suffit d'appliquer CART à des sous-échantillons de données. Néanmoins, même si les itérations (parallèles) du bagging sont simples, la procédure est plus complexe qu'AdaBoost car il faut choisir le nombre et la taille des échantillons. Un grand nombre d'échantillons permet une meilleure réduction de la variance, mais augmente linéairement le temps de calcul. Des échantillons de petite taille impliquent une grande diversité des arbres, mais l'estimation est de moins bonne qualité. La calibration de ces paramètres permet de choisir le bon compromis entre la perte en qualité d'estimation (à cause d'échantillons plus petits) et la diversité des arbres. En revanche, les données mises de côté (*out-of-bag*) permettent de mieux estimer l'erreur de généralisation des arbres individuels, en les utilisant comme données de validation.

3.2.3 Random Forests

La méthode Random Forests a été introduite par Breiman dans [Bre01] comme une méthode spécialisée dans l'agrégation d'arbres. Elle étend le bagging avec une sélection aléatoire de variables [Die00], en restreignant à chaque division les variables possibles à un sous-ensemble aléatoirement tiré parmi les D observées. Ce tirage aléatoire permet à la fois de diversifier les arbres et d'équilibrer l'utilisation des variables dans l'ensemble. Dans l'algorithme CART original, si certaines variables donnent lieu à des réductions d'impureté proches, la division choisira toujours la variable optimale et il est probable que les autres variables quasi-optimales n'apparaissent plus dans la suite des divisions. Les forêts aléatoires sont des procédures complètes et efficaces pour l'agrégation des arbres de régression ou de classification : ils génèrent une grande diversité d'arbres et utilisent les échantillons out-of-bag pour estimer l'erreur de généralisation des arbres individuels. De plus, cette erreur estimée est une mesure alternative de l'importance des variables.
Des résultats de convergence ont été montrés pour les forêts aléatoires, tout d'abord dans le cas de la régression de quantile [Mei06], puis plus récemment en liant le partitionnement par arbres avec les algorithmes de plus proches voisins [BDL08, BD10, Bia12].

Algorithme 2 Random Forests

Paramètres : $\lambda > 0$, $m > 0$, $M > 0$, $n_t > 0$, T', $n_v \in \{1, \dots, D\}$
Échantillonnage : tirer aléatoirement n_t échantillons $Z'_1, \dots, Z'_{n_t}$ de taille T' des données d'entraînement.
Génération des arbres : appliquer CART avec les paramètres (m, M, λ) à chacun des $Z'_1, \dots, Z'_{n_t}$ en tirant aléatoirement n_v variables à chaque division, et obtenir n_t arbres $f_1, \dots, f_{n_t}$.
Agrégation : définir la fonction de prédiction f telle que $\forall x \in \mathcal{X},\ f(x) = \operatorname{sgn}\left(\sum_{i=1}^{n_t} f_i(x)\right)$.
return f

Nous avons choisi d'implémenter cette méthode pour l'efficacité et les possibilités d'analyse servant à l'interprétation des modèles. En pratique, l'ensemble de la procédure de prédiction nécessite le calibrage de plusieurs paramètres : complexité des arbres, proportion de données de bootstrap, nombre d'arbres et nombre de variables tirées à chaque division. Les expériences de la partie 5 donnent un aperçu de l'effet de ces paramètres.

3.3 Prédiction conditionnée au risque estimé

L'utilisation pratique des modèles prédictifs soulève rapidement la question suivante : étant données les nouvelles observations des marchés financiers et les prédictions associées, comment distinguer les prédictions fiables des prédictions risquées ? S'il s'agissait d'un expert humain (comme un analyste financier), on parlerait couramment de "degré de conviction" ou de "confiance". Pour le décideur, le modèle d'apprentissage est un expert parmi d'autres. Suivre une recommandation erronée du modèle a un coût, qui peut être plus élevé que de simplement ignorer le résultat. Donner un degré de confiance aux prédictions permet donc de réduire le risque supporté par l'utilisateur vis-à-vis des résultats. Nous cherchons donc dans cette partie à améliorer la performance et l'utilité pratique de la procédure d'apprentissage en donnant de tels mesures à chaque prédiction. À cet effet, nous proposons deux types d'indicateurs reflétant deux sources différentes de risque : la similarité des observations et la stabilité des prédictions.

3.3.1 Classification avec option de rejet

Nous pouvons relier cette démarche au problème général de classification avec option de rejet [HW06]. En classification binaire, on envisage dans ce cadre la possibilité d'ignorer le résultat de la fonction de prédiction, selon la valeur d'un certain score. Cette possibilité est intéressante à la fois pour les qualités théoriques que l'utilité réaliste.

Typiquement, dans le cadre d'une classification binaire par prédicteur *plug-in*, on réalise une estimation $\hat{\eta}(x)$ de la probabilité $\eta(x)$ de la sortie 1 sachant l'observation x. La fonction de prédiction f est simplement le prédicteur de Bayes dans lequel on remplace la probabilité par l'estimateur :

$$f(x) = \begin{cases} 1 \text{ si } \hat{\eta}(x) \geq \frac{1}{2} \\ -1 \text{ si } \hat{\eta}(x) < \frac{1}{2} \end{cases}$$

Intuitivement, si $\eta(x)$ est proche de $\frac{1}{2}$, prédire l'une ou l'autre classe s'approche d'une décision aléatoire. La solution proposée est d'ajouter une troisième décision possible, consistant à rejeter la prédiction dans un voisinage de type $\left[\frac{1}{2} - d, \frac{1}{2} + d\right]$ avec $0 < d < \frac{1}{2}$:

$$f(x) = \begin{cases} 1 & \text{si } \hat{\eta}(x) \geq \frac{1}{2} + d \\ -1 & \text{si } \hat{\eta}(x) < \frac{1}{2} - d \\ R & \text{si } \frac{1}{2} - d \leq \hat{\eta}(x) < \frac{1}{2} + d \end{cases}$$

et compter directement un coût $c < 1$ (inférieur au coût de l'erreur) en cas de rejet. Il a été montré dans [AT05] qu'il était possible d'obtenir des convergences très rapides sous certaines conditions sur la distribution de η au voisinage de $\frac{1}{2}$. D'autres travaux [HW06, BW08, YW10] ont aussi montré des résultats positifs de vitesse de convergence pour différentes fonctions de perte, notamment en utilisant des substituts des fonctions habituelles afin d'obtenir des formulations abordables.

La classification avec rejet correspond à une réalité pratique dans les applications (par exemple [HD08]). Par exemple, dans le cas du diagnostic médical, classifier un patient comme "malade" a des conséquences sur la suite, et un coût important en cas d'erreur. En cas de doute (risque de prédiction élevé), le rejet du résultat consiste à réaliser d'autres tests, avec un coût externe connu, plutôt que de décider absolument sur la base du premier modèle.

Dans notre situation, la classification par forêts aléatoires avec option de rejet mènerait à des formulations théoriques complexes que nous n'étudierons pas ici. En revanche, nous proposons des mesures qui paraissent pertinentes pour indiquer la fiabilité des prédictions.

3.3.2 Similarité des observations

Une première idée est de comparer les nouvelles observations avec les observations utilisées pour l'entraînement. Si les nouvelles observations sont similaires à la base d'entraînement, on considère que la fonction de prédiction généralisera correctement. Au contraire, si par exemple les nouvelles observations appartiennent à des régions de l'espace d'entrée peu représentées, la généralisation sur ces données peut être risquée. On peut voir cette approche comme une détection d'anomalies, où les données d'entraînement sont considérées "normales" tandis que les nouvelles observations trop différentes doivent être identifiées comme "anormales". La littérature en détection d'anomalies est vaste. D'après [CBK09, MS03a, MS03b], on peut distinguer plusieurs catégories d'approches.

— Détection par apprentissage supervisé : ayant à disposition des labels "normal" et "anormal" sur une base d'entraînement, le problème revient à une classification.
— Analyse des plus proches voisins : l'anomalie est détectée par la distance de l'observation aux points les plus proches, ou par la densité des points au voisinage.
— Clustering des données : les anomalies appartiennent à des petits clusters ou sont éloignés des centres de clusters.
— Vraisemblance statistique : l'anomalie est identifiée d'après la distribution des données sous-jacentes, estimée soit avec des hypothèses sur la forme de la distribution, soit par des méthodes non-paramétriques.

Après avoir comparé plusieurs familles d'approches, nous avons retenu une méthode caractérisant les distances typiques entre les données d'entraînement. Nous reprenons dans la suite les notations de la partie 3.1.

Score de confiance

Afin de caractériser les données d'entraînement dans l'espace d'entrée $\mathcal{X}$, nous construisons un histogramme des distances typiques. Pour chaque point d'entraînement X_t, on calcule sa distance médiane aux k plus proches voisins. Ensuite, on définit un histogramme des distances médianes associées à tous les points, où les hauteurs de l'histogramme représentent les proportions de points dans les intervalles de distance correspondants. L'histogramme résume la topologie des observations en des distances typiques, car les valeurs élevées de l'histogramme montrent des régions densément représentées. La mesure de la similitude d'un nouveau point est directe : il suffit de calculer sa distance médiane avec ses k plus proches voisins et de reporter la valeur correspondante dans l'histogramme. Enfin, le score est obtenu en normalisant cette valeur par la valeur maximale de l'histogramme.

La méthode initiale utilise naturellement la distance euclidienne, mais il s'avère qu'une métrique propre aux fonctions de classification utilisées est plus adaptée.

Distance par arbres

Nous construisons une mesure de distance des données propre à la structure de l'arbre de décision produit par CART. Les observations étant classées dans l'arbre de façon récursive en suivant les différentes branches, la proximité des points dans l'arbre est plus pertinent car reflète la distance en termes de règle de décision. Naturellement, la distance sera construite à partir du critère utilisé par CART, à savoir l'impureté.

Soit $\mathcal{T}$ un arbre de classification binaire à K nœuds $\mathcal{N}_\mathcal{T} = \{1, \dots, K\}$, dans l'ordre croissant de profondeur à partir de la racine (nœud 1). L'arbre est entraîné sur T points. Chaque nœud i est caractérisé par les $n(i)$ points d'entraînement classés dans le nœud et son impureté

$g(i) = \frac{n(i)}{T} G(n)$, où $G(n)$ est l'indice de Gini. Les distances au sein de l'arbre seront construites à partir du plus court chemin $\mathcal{P}(a, b)$ entre deux nœuds a and b, en considérant ici l'arbre comme un graphe non-orienté. $\mathcal{P}(a, b)$ est l'ensemble minimal de nœuds formant un chemin entre a et b dans l'arbre. Sous cette définition, $\mathcal{P}(1, i)$ est simplement le chemin suivi par percolation dans l'arbre pour classer les observations dans le nœud i.

Pour rappel, lors de la division d'un nœud, CART maximise la réduction d'impureté entre ce nœud et ses deux fils. Si on note p le nœud père et $\{a, b\}$ ses nœuds fils, il y a une diminution d'impureté pondérée :

$$\Delta g(p) = g(p) - g(a) - g(b) = \frac{1}{T} \left[n(p)G(p) - n(a)G(a) - n(b)G(b) \right] > 0 \tag{3.1}$$

avec $n(a) + n(b) = n(p)$. Cette variation Δg peut être vue comme une distance entre les deux fils, car plus la perte d'impureté est grande, plus les nœuds fils sont différenciés.

Afin de trouver une extension de cette notion à toute paire de nœuds a and b, on définit leur plus proche racine commune p, qui est aussi le nœud le plus haut placé sur le plus court chemin entre eux :

$$p = \max \left(\mathcal{P}(1, a) \cap \mathcal{P}(1, b) \right) = \min \mathcal{P}(a, b)$$

Aller de a à b par ce chemin revient donc à remonter l'arbre jusqu'à p et redescendre jusqu'à b. Du point de vue d'un graphe pondéré, on peut considérer que le parcours de chaque branche entre deux nœuds i et j induit une distance de $|g(i) - g(j)|$, et ainsi définir cette différence en impureté pondérée g en tant que poids d'arête. La figure 3.4 illustre cette idée.

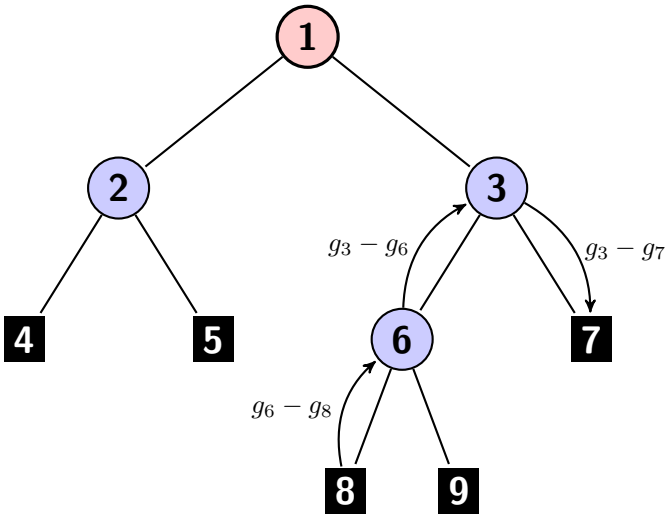


FIGURE 3.4 – Arbre à 9 nœuds et distance entre les feuilles 7 et 8.

En sommant les poids sur le chemin $\mathcal{P}(a, b)$, on aboutit à une distance $D(a, b)$ entre a and b comme suit :

$$D(a, b) = 2g(p) - g(a) - g(b) \tag{3.2}$$

D possède les propriétés d'une distance :
— Positivité : $\forall a, b \in \mathcal{N}_\mathcal{T}$, pour $p = \min \mathcal{P}(a, b)$, $g(p) \geq g(a)$ et $g(p) \geq g(b)$.
— Séparation : $\forall a \in \mathcal{N}_\mathcal{T}, D(a, a) = 0$.
— Symétrie : $\forall a, b \in \mathcal{N}_\mathcal{T}, D(a, b) = D(b, a)$ découlant de la définition.

— Inégalité triangulaire : $\forall a, b, c \in \mathcal{N}_\mathcal{T}, D(a,c) \leq D(a,b) + D(b,c)$. Cette propriété est immédiate dans un arbre car l'intervention d'un troisième point de l'arbre fait parcourir au moins autant de branches : $\mathcal{P}(a,c) \subseteq \mathcal{P}(a,b) \cup \mathcal{P}(b,c)$.

Cette distance dépend non seulement de la décroissance d'impureté, mais aussi de la valeur de l'impureté de la racine commune. Par conséquent, plus les nœuds sont différenciés tôt dans l'arbre, plus ils sont éloignés.

Variation

La distance définie ci-dessus 3.2 majore la distance 3.1 naturellement définie par la structure de l'arbre. Plus précisément, si a et b sont des nœuds fils directs de p, la distance entre eux est majorée de $g(p)$: $D(a,b) = \Delta g(p) + g(p)$.

On peut définir une distance plus proche de 3.1 en décomposant comme suit :

$$\Delta g(p) = \left[\frac{n(a)}{n(p)} g(p) - g(a)\right] + \left[\frac{n(b)}{n(p)} g(p) - g(b)\right] = \frac{n(a)}{T}\left[G(p) - G(a)\right] + \frac{n(b)}{T}\left[G(p) - G(b)\right]$$

Cependant, les deux termes ainsi décomposés ne sont pas nécessairement positifs : le gain en impureté étant sur l'ensemble des deux nœuds fils, il est possible que la division augmente l'impureté G de l'un en faveur d'une réduction plus importante d'impureté sur l'autre. Le poids associé à chaque branche de p à a sera donc plutôt $|\frac{n(a)}{n(p)} g(p) - g(a)|$ afin d'assurer la positivité de la distance.

Cette variante $\tilde{D}$ est donc :

$$\tilde{D}(a,b) = \sum_{i,j \in \mathcal{P}(a,b),\ i \text{ père de } j} \left| \frac{n(i)}{T} g(i) - g(j) \right| \tag{3.3}$$

3.3.3 Stabilité des prédictions

Le risque des prédictions peut aussi être représenté par l'instabilité du résultat. En effet, si une prédiction peut être modifiée de façon significative par de faibles perturbations sur les données, il peut s'agir de sur-apprentissage ou de mauvaise estimation de la fonction de prédiction. Dans le cas de la sélection de titres pour le portefeuille, il est utile de pouvoir ignorer un signal d'achat lorsqu'il pourrait facilement devenir un signal de vente dès que quelques variables macro-économiques changent légèrement. La mesure de la stabilité des prédictions vise à distinguer les prédictions stables de celles qui sont sensibles aux petites perturbations. La notion de stabilité ne se réfère donc pas ici à la sensibilité de l'algorithme d'apprentissage aux données d'entraînement, comme on l'entend dans la littérature (cf. [BE02]), mais à la sensibilité des prédictions d'un modèle aux données de test.

Intuitivement, l'idée de stabilité des prédictions pour un arbre peut être appréhendée par des problématiques similaires à l'idée de départ des SVM. L'arbre partitionne l'espace d'entrée par des hyperplans. Si une observation se trouve au centre d'un sous-espace constituant la partition, on peut considérer qu'elle est bien représentée par le nœud en question, et par conséquent par le label de prédiction qui lui sera associé. En revanche, si l'observation est proche d'une borne du sous-espace dans une certaine direction, une petite perturbation au sens de la norme euclidienne peut aboutir à un nœud très différent dans l'arbre, et potentiellement à une prédiction contraire. Cette discontinuité peut donc être source de l'instabilité étudiée ici, et la position relative de l'observation dans le sous-espace peut permettre de quantifier cette instabilité. Afin de mettre en œuvre cette idée, nous avons choisi de tester empiriquement les réponses des forêts aléatoires en ajoutant un bruit aux nouvelles observations, plutôt que de calculer précisément la distance aux bornes des nœuds, car cela peut s'avérer coûteux en termes de calculs sur tous les arbres.

Définition

Nous proposons de quantifier la stabilité de façon empirique à la sortie des fonctions de prédiction. Pour cela, nous ajoutons un bruit aux observations et mesurons la similarité entre la prédiction initiale et les prédictions sur les observations bruitées. Formellement, on suppose qu'on observe l'entrée X et qu'on prédit $\hat{Y} = f(X)$ avec la fonction de prédiction f. On définit la stabilité S de f en X de la manière suivante :

$$S(f, X, \delta, \varepsilon) = E\left[\delta\left(f(\tilde{X}), \hat{Y}\right)\right] \ ,$$

où $\tilde{X} = X + \epsilon$ avec ε un vecteur aléatoire de bruit et δ une fonction dans $[0, 1]$ et mesurant la distance entre deux prédictions.

Mise en œuvre

Le bruit ε peut être, par exemple, un vecteur gaussien sous $\mathcal{N}(0, \Sigma)$, où Σ est la matrice de covariance entre les rendements à horizon H des variables en entrée. Ce paramètre d'horizon définit l'amplitude du bruit en le liant au nombre de jours de rendement à considérer. De façon plus fine, on pourrait aussi définir un bruit respectant à la fois la covariance et les distributions marginales des rendements.

Dans notre cas, le modèle prédictif, qui est une forêt aléatoire, est trop complexe pour une étude analytique donnant une forme explicite pour S. Nous estimons donc S en générant M échantillons de vecteurs de bruit $(\epsilon_i)_{i=1,\dots,M}$ pour les ajouter à X et obtenir les observations perturbées $\tilde{X}_i = X + \epsilon_i$. Enfin, les prédictions "perturbées" sont $\tilde{Y}_i = f\left(\tilde{X}_i\right)$. La stabilité est ensuite estimée sur ces échantillons :

$$\hat{S} = \frac{1}{M}\sum_{i=1}^{M} \delta\left(\tilde{Y}_i, \hat{Y}\right) \ ,$$

où $\delta(a, b)$ vaut 1 si $a = b$ et 0 sinon.

Dans le cas de la classification des sens de variation, il s'agit d'effectuer la prédiction sur les échantillons perturbés et de mesurer la fréquence des prédictions identiques à la "vraie" prédiction.

Dans le cas de modèles agrégés comme les forêts aléatoires, la stabilité peut servir de critère de pondération des arbres individuels. En mesurant la stabilité sur chaque arbre au lieu de la forêt, le modèle prédictif peut être modifié en vote pondéré par la stabilité. Cependant, cette pondération risque de donner un poids excessif aux arbres à un seul nœud, puisque de tel arbres effectuent toujours la même prédiction et ont une stabilité maximale.

3.3.4 Procédure de recommandation avec rejet

La procédure de recommandation avec rejet se fera donc en plusieurs étapes :

1. construire les données d'entraînement $\{Z_1, \dots, Z_T\} = \{(X_1, Y_1), \dots, (X_T, Y_T)\}$,
2. apprendre une forêt aléatoire f à partir des Z_t,
3. définir un indicateur de confiance (similarité ou stabilité) $c_f : \mathcal{X} \times \mathcal{Y} \longrightarrow [0, 1]$,
4. pour toute nouvelle observation x, prédire la réponse $f(x)$ et le score $s = f(x) \cdot c_f(x, f(x))$,
5. émettre un signal d'achat si $s > s_{\min}$, émettre un signal de vente si $s < -s_{\min}$, et s'abstenir sinon.

3.4 Exemple d'arbre

Afin d'illustrer les résultats de l'algorithme CART, nous présentons ici un exemple concret de prédiction du signe du rendement à horizon 25 jours de l'EURO STOXX 50. Le modèle est entraîné sur 210 jours d'observations jusqu'au 4 janvier 2010 avec 88 variables d'entrée.

CART donne l'arbre de la figure 3.5. L'arbre présenté a été élagué de deux divisions. La partition induite par les deux premières divisions se représente dans le plan de la figure 3.6. Les variables représentées sont rendues explicites dans le tableau 3.1.

Code	Nom	Composante	Importance ($\times 10^{-3}$)
x43	Zone euro - Inflation break-even 10 ans	Tendance	6.72
x27	Allemagne - CDS souverain 5 ans	Tendance	8.39
x9	Espagne - Taux d'intérêt 10 ans	Tendance	4.67
x32	Taux de change CHF/EUR	Volatilité	3.53
x25	CAC 40	Tendance	2.50

TABLE 3.1 – Premières variables de division pour l'arbre 3.5

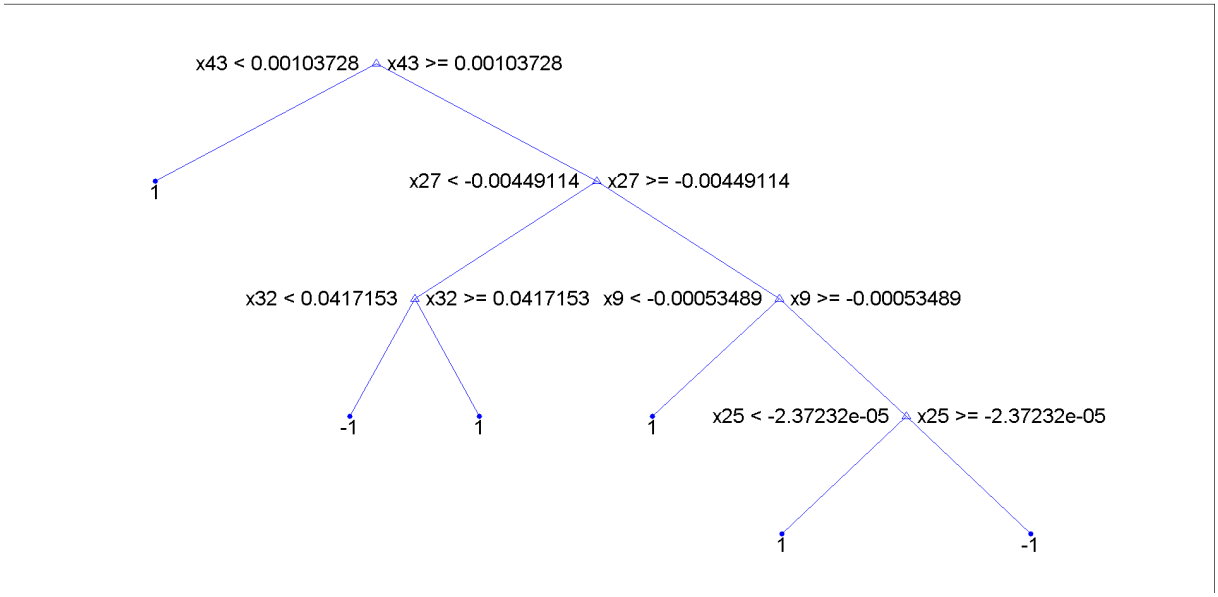


FIGURE 3.5 – Arbre construit par CART pour la prédiction de l'EURO STOXX 50

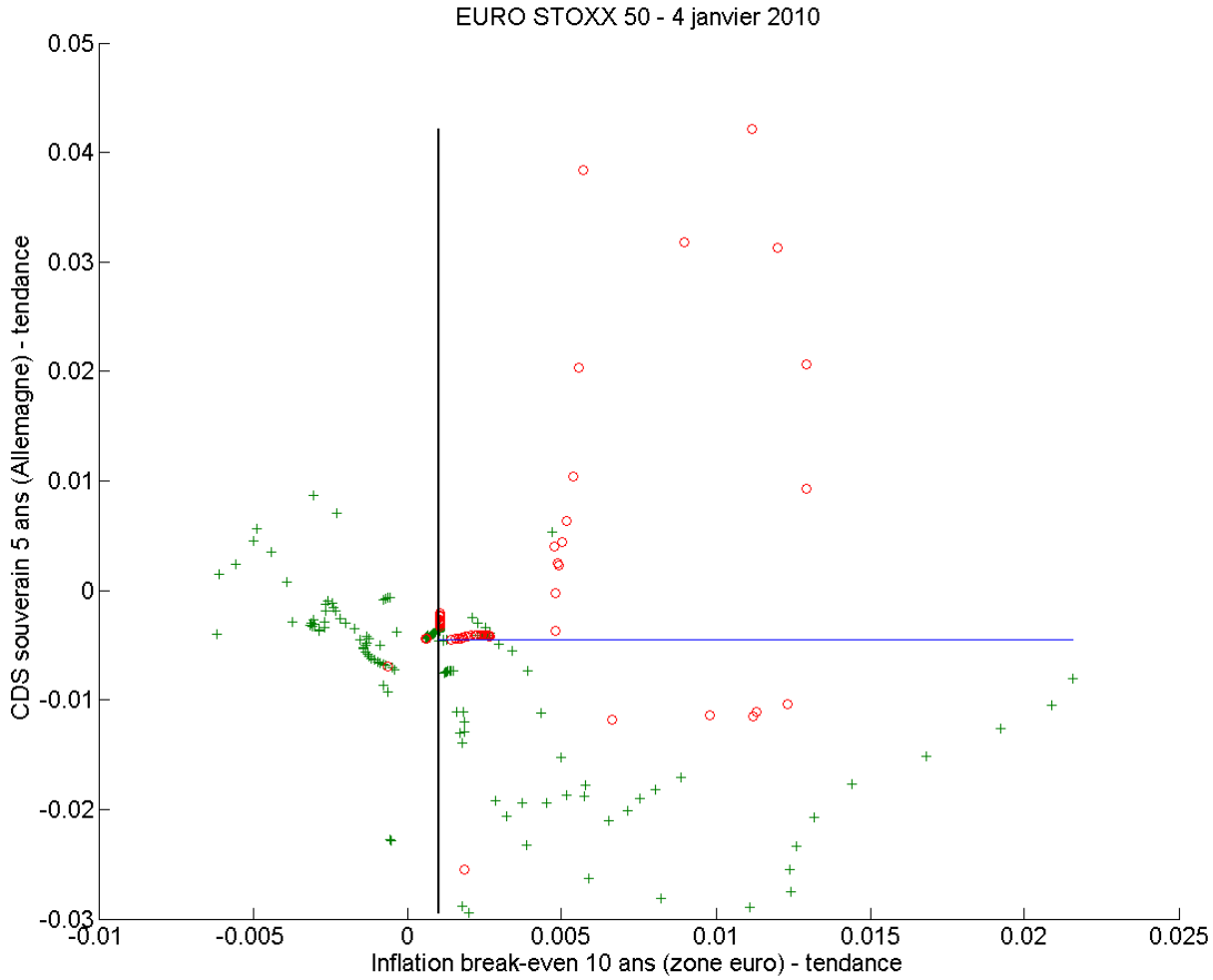


FIGURE 3.6 – Partition induite pour la prédiction de l'EURO STOXX 50

Chapitre 4

Apprentissage multi-tâche

La prédiction du rendement des actifs constituant le portefeuille est un problème multivarié, plutôt qu'un ensemble de problèmes indépendants. En effet, les actifs sont dépendants entre eux, et l'objectif est de constituer un portefeuille unique à partir de ceux-ci. L'intérêt d'une approche multivariée est double : d'une part il paraît intéressant d'exploiter la dépendance afin de parvenir à des performances supérieures, d'autre part on peut obtenir des résultats plus cohérents avec cette dépendance.

Dans cette optique, nous entrons dans le domaine de l'apprentissage multi-tâche, et proposons une approche adaptée à la structure de dépendance recherchée dans un cadre linéaire. Nous proposons de considérer les tâches comme les sommets d'un graphe, et leurs dépendances mutuelles comme les poids des arêtes. Nous formulons le problème d'apprentissage comme une minimisation du risque empirique dont la régularisation utilise le laplacien du graphe des tâches de prédiction, en modifiant la proposition initiale de [EMP05]. Cette formulation est innovante en plusieurs aspects. La structure en graphe permet de rendre compte de dépendances plus riches entre les tâches. Surtout, la structure de dépendance n'est pas supposée connue, mais apprise en même temps que les fonctions de prédiction : il s'agit d'un problème par nature plus difficile. Ces travaux ont fait l'objet de contributions, dont on peut trouver un exemple en annexe A.

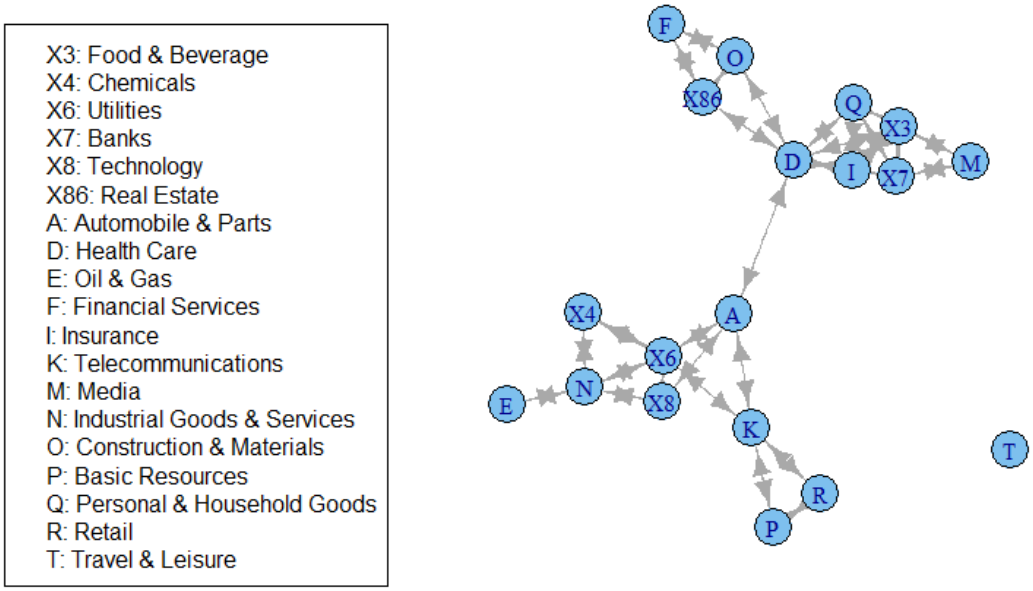


FIGURE 4.1 – Exemple de graphe entre tâches de prédiction financière : indices sectoriels.

4.1 Cadre et état de l'art

L'apprentissage multi-tâche est un sujet actif dans la littérature récente. Les problèmes multi-tâches portent sur la résolution simultanée de plusieurs tâches de classification ou de régression, afin d'améliorer la performance par rapport à l'apprentissage de chaque tâche séparément. Ce type d'approche est notamment efficace dans les cas où les données sont peu nombreuses, en faisant profiter à chaque modèle de l'information contenue dans les autres tâches, dans le cas où celles-ci sont suffisamment dépendantes.

L'aspect multi-tâche est naturel dès que le problème est constitué de plusieurs objets à prédire au même niveau. La classification en catégories d'un texte, la recommandation de produits à des utilisateurs et la prédiction d'actifs financiers ont en commun le traitement d'un ensemble de tâches de prédiction potentiellement liées. Cette relation peut consister par exemple en des données d'observation partagées, des variables explicatives communes, ou encore une dépendance entre les labels à prédire. En effet, certaines catégories de texte apparaissent souvent ensemble, les utilisateurs peuvent partager des similarités de goût ou d'intérêt, et les actifs financiers sont généralement corrélés plus ou moins fortement. Il devient donc intéressant de considérer le problème d'ensemble avec un objectif multivarié plutôt que plusieurs objectifs différents, afin de tirer profit de ces relations.

Les approches multi-tâches sont aussi variées que les hypothèses possibles sur la structure de dépendance. Néanmoins, un cadre théorique général a été énoncé [Bax00, BDS03, Mau06]. On considèrera le cadre de base suivant dans la suite du chapitre. Soit $\mathcal{X}$ l'espace d'entrée et $\mathcal{Y}$ l'espace de sortie, par exemple $\mathcal{X} \subseteq \mathbb{R}^d$ et $\mathcal{Y} \subseteq \mathbb{R}$. Les tâches sont représentées par n fonctions f_ℓ $(\ell = 1, \dots, n)$, à entraîner sur les données $\{(x_{li}, y_{li}) : i = 1, \dots, m_\ell,\ \ell = 1, \dots, n\} \subseteq \mathcal{X} \times \mathcal{Y}$. Ces tâches peuvent être vues comme des réalisations de variables aléatoires de loi jointe inconnue, cette loi étant à l'origine du biais liant les tâches entre elles.

4.1.1 Régression multivariée

Historiquement, le problème de régression *ridge* multivariée a été posé dès 1980 par Brown et Zidek [BZ80]. Dans le cadre classique, le modèle est $y = x\beta + \varepsilon$ où $y \in \mathbb{R}$ est la sortie à prédire, $x \in \mathbb{R}^d$ est l'observation en entrée, $\beta \in \mathbb{R}^d$ est le vecteur de coefficients de régression et ε est un bruit aléatoire d'espérance nulle. On considère n tâches de régression de ce type, partageant les mêmes observations et dont les bruits **ne sont pas nécessairement indépendants**. On suppose disposer de N points de données pour chaque tâche, ce qui se résume à une matrice d'observations X commune de taille $N \times d$ et d'une matrice de réponses Y de taille $N \times n$. On veut estimer la matrice de coefficients de régression $\hat{\beta}$ de taille $d \times n$. En notant M^ℓ la colonne ℓ d'une matrice M, la résolution séparée donne les estimateurs :

$$\hat{\beta}^\ell = \left(X^\top X + \gamma I_d\right)^{-1} X^\top Y^\ell$$

où γ est le paramètre de régularisation et I_d la matrice identité de taille $d \times d$.

Les auteurs de [BZ80] proposent de tenir compte de la covariance entre les Y^ℓ en calculant l'estimateur précédent dans un problème de taille dn :

$$\tilde{\beta} = \left(X^\top X \otimes I_n + I_d \otimes \Gamma\right)^{-1} \left(X^\top X \otimes I_n\right) \hat{\beta} \tag{4.1}$$

où $\otimes$ représente le produit de Kronecker, Γ est une matrice définie positive $n \times n$ de régularisation, et les $\tilde{\beta}$ et $\hat{\beta}$ sont les formes vectorielles des matrices de coefficients de régression. Le choix

de Γ d'après les données, étudié en détail dans l'article, détermine le biais multivarié qui permet d'adapter le problème aux cas d'application.

Les problèmes multi-tâches ont été étudiés de façon plus active à partir du milieu des années 90, notamment dans les travaux de thèse de Caruana [Car97] axés sur les réseaux de neurones, et dans un article de Breiman et Friedman [BF97] sur la régression en réponse multivariée. Les auteurs de ce dernier montrent que des réponses corrélées peuvent être mieux prédites en tenant compte des autres prédictions. Si on dispose, pour les n réponses y_ℓ (centrées), de régressions aux moindres carrés $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_n)$ entraînées séparément, ils proposent de trouver des prédicteurs plus précis $\tilde{y}_\ell$ par une combinaison linéaire :

$$\forall \ell \in \{1, \ldots, n\},\ \tilde{y}_\ell = \sum_{k=1}^{n} b_{\ell k} \hat{y}_k$$

où les $b_{\ell k}$ sont des réels. En notant B la matrice des $b_{\ell k}$, cela s'écrit aussi $\tilde{y} = B\hat{y}$.

Les auteurs proposent d'estimer B par analyse canonique des corrélations (*CCA - Canonical Correlation Analysis*). Sans entrer dans les détails, étant donnés deux vecteurs aléatoires U et V, la CCA recherche les directions u et v, appelées coordonnées canoniques de U et V, qui maximisent la corrélation entre les vecteurs $u^\top U$ et $v^\top V$. L'analyse donne jusqu'à $\min(\dim(x), \dim(y))$ paires successives de directions (u_k, v_k) de sorte que la corrélation $c_k = \mathrm{corr}(u_k^\top U, v_k^\top V)$ soit maximale sous la contrainte que les vecteurs u_k et v_k soient décorrélés des précédents. Pour de plus amples descriptions et études, on peut se référer par exemple à [HSST03].

Dans le cas étudié, la CCA est effectuée entre le vecteur de réponse y et les observations x La matrice B optimale est estimée sous la forme $B = T^{-1}DT$, où T est la matrice composée des vecteurs (lignes) de coordonnées canoniques pour y, et D est une matrice diagonale dont les coefficients sont issus des corrélations canoniques c_k. Cela revient élégamment à une régression presque classique dans un espace transformé : effectuer un changement de base par la matrice T, réaliser la régression aux moindres carrés classique sur les y transformés, redimensionner les régressions $\hat{y}$ par D, puis retransformer le résultat vers l'espace de départ.

4.1.2 Régularisation

Comme dans beaucoup de méthodes d'apprentissage par minimisation du risque empirique, les approches par régularisation sont fréquemment utilisées pour le multi-tâche. En particulier dans le cas linéaire, chaque tâche peut être représentée par une fonction de prédiction $f_\ell : x \mapsto w_\ell^\top x$, ce qui équivaut à un vecteur w_ℓ de coefficients de régression. Dans ce cas, l'apprentissage régularisé prend la forme d'un problème d'optimisation :

$$\min \left\{ \sum_{\ell=1}^{n} \sum_{j=1}^{m} E(w_\ell^\top x_{j\ell}, y_{j\ell}) + \gamma\, \Omega(W) : W \in \mathbf{M}_{d,n} \right\}, \tag{4.2}$$

où $E : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ est une fonction de perte convexe, Ω est un terme de pénalité, $\mathbf{M}_{d,n}$ est l'ensemble des matrices réelles de taille $d \times n$ et W est la matrice composée des w_ℓ en colonnes. Le premier terme de perte assure l'adéquation de la solution aux données tandis que la pénalité Ω favorise une certaine structure de dépendance entre les tâches. Le paramètre de régularisation γ (positif, à calibrer) détermine le niveau de compromis entre la bonne adéquation aux données et le biais sur la relation entre les tâches. Le choix de la pénalité Ω donne donc lieu à une certaine diversité de méthodes multi-tâches.

Une approche courante consiste à utiliser des pénalités de norme L_2 entre les tâches ou des combinaisons de tâches, comme montré en détail dans [EMP05, JBV09]. Dans ce type d'approche, on pénalise la distance entre tâches, et celles-ci sont par conséquent poussées à être similaires. Par exemple dans [EMP05], cette similarité se présente sous la forme de tâches regroupées autour d'un centre, avec la pénalité suivante et un paramètre supplémentaire ρ :

$$\Omega(W) = \sum_{\ell=1}^{n} \|w_\ell\|^2 + \rho \sum_{\ell=1}^{n} \|w_\ell - \frac{1}{n}\sum_{q=1}^{n} w_q\|^2 \ .$$

D'autres travaux [LPTvdG09, AEP08, KX10, ZCY12] considèrent des pénalités combinant des normes L_1 et L_2 afin de privilégier la parcimonie :

$$\Omega(W) = \|W\|_{2,1} = \sum_{i=1}^{d} \|w^i\|_2$$

où w^i est une ligne de w, correspondant à une dimension donnée pour toutes les tâches. Les auteurs montrent que la formulation multi-tâche a une performance théorique au moins aussi bonne que le Lasso.

Dans une autre catégorie importante de méthodes, l'objectif est d'apprendre des fonctions – et de façon équivalente, des vecteurs w_ℓ – dans des sous-ensembles ou des variétés de $\mathbb{R}^d$ de dimension faible. Ces méthodes (par exemple [AEP08, SRJ05, ADG10, AZ05]) reposent généralement sur une pénalisation par la norme trace $||W||_*$, qui est par définition la somme des valeurs singulières de la matrice W. Le problème d'optimisation devient :

$$\min \left\{ \sum_{\ell=1}^{n} \sum_{j=1}^{m_\ell} E(w_\ell^\top x_{j\ell}, y_{j\ell}) + \gamma \, ||W||_* \right\}. \tag{4.3}$$

Ce type de pénalisation favorise les solutions de rang faible, car la norme trace est connue pour être le substitut convexe le plus proche du rang matriciel (cf. [FHB01]).

4.1.3 Autres approches

Outre le cadre de régularisation, il existe une variété d'approches plus spécifiques aux algorithmes de base que l'on adapte au cadre multi-tâche. Il s'agit souvent de modifier une étape de fonctionnement de l'algorithme ou le critère d'apprentissage afin de tenir compte de tâches multiples. Elles se sont fondées entre autres sur le boosting [CSV+11], les réseaux de neurones [Car97], les arbres de décision [KVS07, WZCG08, FCT] et les méthodes bayésiennes hiérarchiques [BH03].

4.2 Régularisation par le laplacien du graphe

La plupart des méthodes couramment rencontrées reposent sur l'hypothèse implicite que toutes les tâches sont liées d'une certaine manière. Cependant, dans beaucoup de cas réels, on ne sait pas d'avance si les tâches doivent effectivement être toutes liées entre elles, et il est raisonnable de supposer que les liens entre les tâches peuvent être d'intensités diverses, sans information a priori sur ces intensités. En particulier, on peut considérer que les tâches se regroupent en clusters, avec des relations fortes au sein des groupes et faibles entre groupes différents. Il peut être important de considérer à la fois les relations intra et inter-groupes. Cette perspective, plus réaliste, rend toutefois le problème plus complexe, faute d'hypothèse fiable sur le regroupement pertinent en général. De plus, un pré-traitement visant à regrouper les tâches,

tels qu'un clustering par k-means ou clustering spectral, s'avère souvent insuffisant par manque d'information adéquate sur le critère de regroupement.

L'application présente de prédiction des actifs financiers est un bon exemple de cette difficulté. Les tâches sont dépendantes entre elles avec une structure sous-jacente inconnue a priori, qui varie au cours du temps, ce qui rend inapproprié le clustering des tâches par une grande quantité de données historiques. Les systèmes de recommandation et l'apprentissage de préférences consituent d'autres exemples frappants, où on doit apprendre les préférences de nombreux consommateurs dont les comportements peuvent être regroupés en clusters d'intérêts et goûts similaires au sein des groupes, mais très différents d'un groupe à un autre.

Nous proposons de considérer les tâches dans un graphe définissant leurs relations, et d'utiliser le laplacien du graphe dans une approche multi-tâche régularisé, avec pour base la formulation proposée (mais non résolue) dans [EMP05]. En modifiant la formulation de cette approche, nous proposerons dans la partie 4.3 une nouvelle méthode pour apprendre simultanément les tâches et le graphe de relations entre elles.

4.2.1 Apprentissage des relations entre tâches : travaux de référence

Jusqu'ici, seule une proportion réduite de travaux traitent de tâches regroupées en clusters, dans un contexte de régularisation. Par exemple dans [AMP08, KGS11], les tâches sont apprises dans des sous-espaces multiples par une variation des approches de faible rang (norme trace). L'approche recherche plusieurs sous-espaces de dimension faible pour apprendre simultanément les tâches avec ces sous-espaces. Dans [BH03], le clustering des tâches est traité dans un cadre bayésien hiérarchique avec des mélanges de distributions. Une approche très différente [KD12] est inspirée de la norme trace et son expression en termes de factorisation de matrices. Essentiellement, l'un des facteurs de W est vu comme une matrice encodant le regroupement des tâches et est pénalisé en parcimonie L_1. En pénalisant les facteurs de W, cette méthode favorise des solutions de rang faible.

L'apprentissage simultané des tâches et leurs clusters a aussi été le sujet de [JBV09], où on trouve une relaxation convexe du problème de regroupement des tâches dans un cadre de régularisation L_2. Les auteurs considèrent des pénalités combinant trois termes : une pénalité L_2 sur la moyenne des tâches, une mesure de la variance intra-cluster - similaire à une distance des tâches aux centroïdes de cluster - et une mesure de la variance inter-clusters à partir des centroïdes. La régularisation est une fonction de la matrice binaire encodant l'appartenance aux clusters et il s'agit d'un problème d'optimisation non-convexe. Les auteurs proposent une relaxation convexe de la pénalité, menant à la forme alternative

$$\operatorname{tr}(\Pi W \Sigma_c^{-1} W^\top \Pi) \quad \text{s.c.} \quad \Sigma_c \succeq 0, \alpha I_n \preceq \Sigma \preceq \beta I_n, \operatorname{tr}\Sigma = \gamma, \tag{4.4}$$

où tr est la trace matricielle, Σ_c et Σ sont en relation affine, Π une matrice de projection fixée et α, β, γ des constantes positives liées aux paramètres de régularisation.

4.2.2 Notions sur les graphes

L'approche que nous proposons dans la suite se fonde sur une représentation des relations entre tâches sous forme de graphe. Nous introduisons ici quelques notions utiles sur les graphes que nous utiliserons. Pour des détails complets, nous nous référons au tutoriel de von Luxburg sur le clustering spectral [vL07].

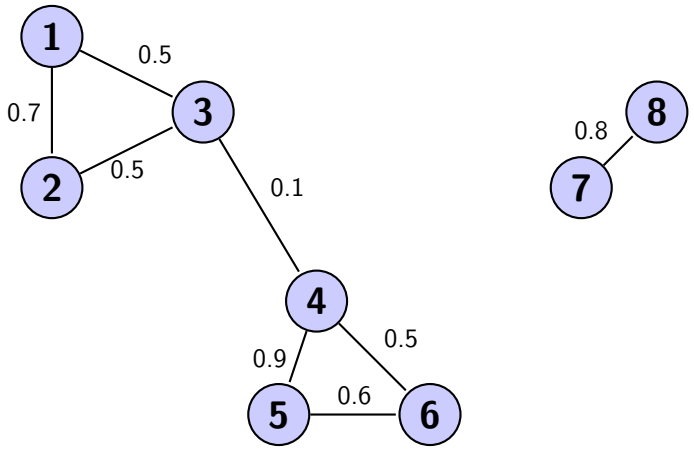


FIGURE 4.2 – Exemple de graphe non-orienté pondéré

Un graphe non-orienté $G = (V, E)$ est un ensemble de sommets $V = \{v_1, \ldots, v_n\}$ dont certains sont reliés par des arêtes $E = \{e_{ij} : i, j \in \{1, \ldots, n\}$, i et j reliés$\}$. On considère un graphe pondéré, où chaque arête e_{ij} est associé à un poids $a_{ij} \geq 0$ caractérisant le degré de relation entre les sommets reliés par l'arête.

On définit la matrice d'adjacence A de taille $n \times n$ de coefficients $A_{ij} = a_{ij}$, et la matrice degrés D diagonale de taille $n \times n$ de coefficients $D_{ii} = \sum_{j=1}^{n} a_{ij}$. Le laplacien de G peut être défini comme la matrice L de taille $n \times n$ calculée par la différence entre les degrés et l'adjacence : $L = D - A$. Dans certains cas, on peut aussi considérer un laplacien normalisé $L_{sym} = D^{1/2} L D^{-1/2}$. Les propriétés qui nous intéressent sont les suivantes sur le spectre de L.

Propriété 1. *La matrice laplacienne L possède les propriétés suivantes :*

1. *L est symétrique semi-définie positive.*
2. *La plus petite valeur propre de L est 0. Le vecteur propre associé est le vecteur constant dont les éléments sont 1.*
3. *L possède n valeurs propres réelles positives $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.*

Propriété 2.

Le nombre de composantes connexes de L est donné par la multiplicité de la valeur propre 0.

De plus, il est intéressant d'observer les différences entre les valeurs propres successives de L : la présence de clusters dans le graphe est caractérisée par un saut dans le spectre, et la position de ce saut indique le nombre optimal de clusters. Dans l'exemple de la figure 4.2, le graphe possède $n = 8$ sommets formant deux composantes connexes, groupes distincts non-reliés. Ses matrices caractéristiques sont les suivantes.

Adjacence A

$$\begin{pmatrix} & 0.7 & 0.5 & & & & & \\ 0.7 & & 0.5 & & & & & \\ 0.5 & 0.5 & & 0.1 & & & & \\ & & 0.1 & & 0.9 & 0.5 & & \\ & & & 0.9 & & 0.6 & & \\ & & & 0.5 & 0.6 & & & \\ & & & & & & & 0.8 \\ & & & & & & 0.8 & \end{pmatrix}$$

Degré D

$$\begin{pmatrix} 1.2 & & & & & & & \\ & 1.2 & & & & & & \\ & & 1.1 & & & & & \\ & & & 1.5 & & & & \\ & & & & 1.5 & & & \\ & & & & & 1.1 & & \\ & & & & & & 0.8 & \\ & & & & & & & 0.8 \end{pmatrix}$$

$$\text{Laplacien : } L = D - A = \begin{pmatrix} 1.2 & -0.7 & -0.5 & & & & & \\ -0.7 & 1.2 & -0.5 & & & & & \\ -0.5 & -0.5 & 1.1 & -0.1 & & & & \\ & & -0.1 & 1.5 & -0.9 & -0.5 & & \\ & & & -0.9 & 1.5 & -0.6 & & \\ & & & -0.5 & -0.6 & 1.1 & & \\ & & & & & & 0.8 & -0.8 \\ & & & & & & -0.8 & 0.8 \end{pmatrix}$$

Le spectre du laplacien montre deux valeurs propres nulles correspondant aux deux composantes connexes. Le saut entre la troisième et la quatrième valeur propre indique un nombre optimal de clusters égal à trois, ce qui correspond aux sommets $\{1, 2, 3\}$, $\{4, 5, 6\}$ et $\{7, 8\}$.

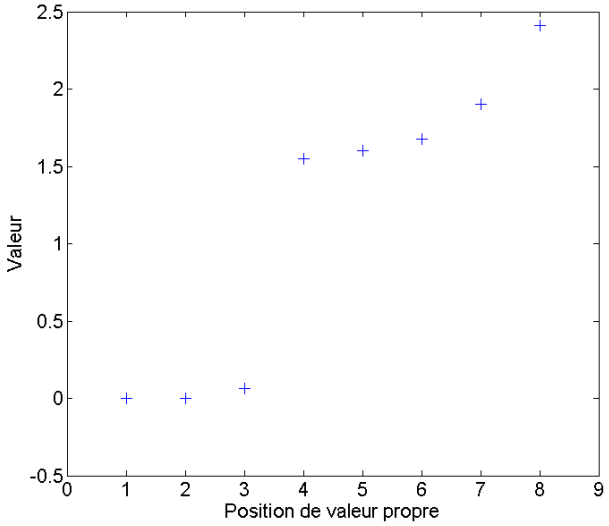


FIGURE 4.3 – Spectre du graphe 4.2 : 2 composantes connexes, 3 clusters

4.2.3 Pénalisation par le laplacien : formulation initiale

Nous proposons de représenter l'ensemble des tâches sous la forme d'un graphe caractérisé par son laplacien, et d'utiliser celui-ci dans la régularisation.

Un cadre général a été proposé dans [EMP05] pour l'apprentissage multi-tâche par régularisation au moyen du laplacien de graphe, dans un RKHS. On y énonce des problèmes de régularisation dans l'espace conjoint des tâches avec une pénalité quadratique de la forme $w^\top Q w$, où w est la forme vectorielle de W et Q est une matrice positive définie de taille $dn \times dn$ définissant les relations entre les tâches deux à deux. Le problème d'apprentissage peut ensuite être reformulé comme un problème de régularisation dans un RKHS de fonctions définies sur l'espace augmenté $\mathcal{X} \times \mathbb{N}_n$, où $\mathbb{N}_n$ est l'ensemble $\{1, \ldots, n\}$.

Par exemple, si $Q = \Delta \otimes I_d$, où Δ est une matrice diagonale positive de taille $n \times n$ et $\otimes$ est le produit de Kronecker, on retrouve le cas particulier de l'apprentissage indépendant, et les coefficients diagonaux de Δ sont les paramètres de régularisation. Ou encore, un cas non-trivial est donné par la méthode générale étudiée dans [EMP05], qui pénalise les distances L_2 entre les tâches w_ℓ et leur moyenne :

$$\sum_{\ell=1}^{n} \|w_\ell\|^2 + \rho \sum_{\ell=1}^{n} \|w_\ell - \frac{1}{n} \sum_{q=1}^{n} w_q\|^2 \,. \tag{4.5}$$

Cette formulation correspond à une hypothèse naturelle dans beaucoup d'applications, à savoir que toutes les tâches sont proches les unes des autres en distance L_2.

Cependant, lorsque les tâches forment plusieurs groupes distincts, les régularisations comme 4.5 ne sont plus appropriées. Les tâches de groupes distincts sont faiblement dépendantes, tandis que cette pénalité favorise une relation forte entre toutes. Les travaux de [Sol13] démontrent de surcroît que dans certains cas, la performance de l'ensemble des tâches peut être significativement altérée par l'existence un petit nombre de tâches ne vérifiant pas l'hypothèse. Pour tenir compte d'une structure plus flexible, les auteurs de [EMP05] définissent le graphe des tâches à travers une matrice de poids A, à coefficients positifs, déterminant les liens entre les n tâches. On obtient ainsi le laplacien des tâches L comme étant le laplacien calculé à partir de A :

$$L = D - A \,, \tag{4.6}$$

où $D = \mathrm{Diag}(d)$ est la matrice diagonale de degrés des sommets : $d_i = \sum_{j=1}^{n} A_{ij}$. Une méthode multi-tâche est ensuite proposée dans [EMP05] en faisant intervenir ce laplacien. Bien que non-implémentée, cette méthode pénalise les tâches avec le laplacien comme suit :

$$\min \left\{ \sum_{\ell=1}^{n} \sum_{j=1}^{m_\ell} E(w_\ell^\top x_{j\ell}, y_{j\ell}) + \gamma \sum_{\ell,q=1}^{n} w_\ell^\top w_q L_{\ell q} : \; w_\ell \in \mathbb{R}^d \; \forall \ell \in \mathbb{N}_n \right\} . \tag{4.7}$$

Cette régularisation se justifie par le fait que la pénalité est égale à $\sum_{\ell,q=1}^{n} \|w_\ell - w_q\|^2 A_{\ell q}$ et que cela favorise par conséquent la similtude des paires de tâches (w_ℓ, w_q) avec des poids $A_{\ell q}$ élevés.

La pénalité est aussi égale à $w^\top (L \otimes I_d) w$, où $w = (w_1^\top \dots w_n^\top)^\top$ est la forme vectorisée de W et I_d est la matrice identité $d \times d$. Cette forme quadratique est positive semi-définie, et non définie, donc elle n'est pas directement équivalente à une régularisation dans un RKHS. [EMP05] propose de résoudre ce problème en restreignant w à l'image de $L \otimes I_d$, ce qui donne une formulation équivalente dans un RKHS avec le noyau multi-tâche $K((x,\ell),(t,q)) = L_{lq}^{+} x^\top t$, pour tout $x, t \in \mathbb{R}^d$, $\ell, q \in \mathbb{N}_n$.

4.3 Apprentissage multi-tâche par laplacien

Nous considérons qu'en pratique, il est nécessaire d'optimiser W sur tout l'espace, car la restriction précédente ignore des informations cruciales contenues dans le noyau du laplacien. Afin de s'en convaincre, on peut considérer par exemple un graphe de deux tâches avec des poids positifs, ce qui correspond à la pénalité $\|w_1 - w_2\|^2$. La contrainte de [EMP05] à l'image implique $w_1 + w_2 = 0$, car le vecteur constant appartient au noyau de L pour tout laplacien. Or cette contrainte contredit l'information d'un poids élevé entre les deux tâches et d'une distance faible entre w_1 et w_2. De manière plus générale, tout graphe avec k composantes connexes possède k valeurs propres nulles, et on trouvera donc k contraintes linéairement indépendantes sur w_ℓ. Si les tâches au sein de chaque cluster étaient proches les unes des autres (qui est notre but), alors les contraintes les forceraient à être proches du vecteur nul. En pratique, imposer les k contraintes de [EMP05] restreint les tâches dans un sous-espace de dimensions $n - k$. Ce sous-espace dépend uniquement de la topologie du graphe et peut être incohérente avec les tâches ayant effectivement généré les données. Ces contraintes peuvent par conséquent empêcher l'apprentissage des tâches correctes, comme montré par les expériences.

Nous proposons une nouvelle méthode pour apprendre simultanément les tâches et le graphe de relations entre elles. Nous reprenons le cadre de régularisation par laplacien de graphe énoncé

dans [EMP05], et montrons comment l'adapter à notre objectif. Nous formulons ensuite notre méthode d'apprentissage en tant que problème d'optimisation dans un RKHS (*reproducing kernel Hilbert space* - espace de Hilbert à noyau reproduisant), où le noyaux lui-même doit être appris. Le problème est analogue à l'apprentissage parmi un ensemble infini de noyaux, mais l'ensemble ici n'est pas convexe. Malgré cela, il est possible de résoudre notre problème au moyen d'un algorithme d'optimisation alternée.

4.3.1 Régularisation par le laplacien augmenté

L'objectif final est d'apprendre simultanément les n tâches et le graphe des tâches représenté par le laplacien. Nous reprenons l'approche (4.7) proposée par [EMP05], mais en obtenant la formulation en RKHS par une voie différente. Nous proposons d'ajouter une perturbation du laplacien avec une constante ε fixée :

$$\min\left\{\sum_{\ell=1}^{n}\sum_{j=1}^{m_\ell} E(w_\ell^\top x_{j\ell}, y_{j\ell}) + \gamma\left(\sum_{\ell,q=1}^{n} w_\ell^\top w_q L_{\ell q} + \varepsilon\sum_{\ell=1}^{n}\|w_\ell\|^2\right) :\ w_\ell \in \mathbb{R}^d\ \forall \ell \in \mathbb{N}_n\right\}. \tag{4.8}$$

Cette formulation augmente la pénalité $\sum_{\ell,q=1}^{n}\|w_\ell - w_q\|^2 A_{\ell q}$ d'un terme de perturbation sur les tâches. Lorsque ε est faible, le terme lié au graphe domine et les similarités entre les tâches correspond aux poids. Ainsi, la formulation (4.8) permet de refléter l'intuition sur le laplacien des tâches dans une forme positive définie bien commode. En pratique, le paramètre ε sera fixé à une valeur faible.

La pénalité peut aussi être formulée en $w^\top((L+\varepsilon I_n)\otimes I_d)w$, ou encore $\mathrm{tr}((L+\varepsilon I_n)W^\top W)$. Le noyau multi-tâche est alors égal à :

$$K((x,\ell),(t,q)) = (L+\varepsilon I_n)^{-1}_{lq}\, x^\top t\ , \tag{4.9}$$

pour tout $x, t \in \mathbb{R}^d$, $\ell, q \in \mathbb{N}_n$.

Au lieu d'un noyau linéaire, tout noyau reproduisant scalaire G peut être utilisé pour les inputs [CMPY08] :

$$K((x,\ell),(t,q)) = (L+\varepsilon I_n)^{-1}_{lq}\, G(x,t)\ , \tag{4.10}$$

où x, t sont dans un espace d'entrée $\mathcal{X}$. Soit $\mathcal{H}_K$ le RKHS associé à ce noyau.

Il apparaît d'après la forme $\mathrm{tr}((L+\varepsilon I_n)W^\top W)$ qu'un laplacien diagonal par blocs (moyennant des permutations de lignes et colonnes) tiendrait compte des corrélations entre tâches du même groupe tout en ignorant les corrélations entre groupes. Cette intuition s'applique de même aux blocs extra-diagonaux de coefficients faibles entre clusters différents. Par conséquent, pour un laplacien donné (estimé ou connu), des méthodes de clustering permettraient de retrouver le regroupement des tâches.

À partir des données $x_{j\ell} \in \mathcal{X}$, $y_{j\ell} \in \mathcal{Y}$, avec $\ell \in \mathbb{N}_n$, $j \in \mathbb{N}_{m_\ell}$, l'entraînement peut être réalisé en résolvant le problème d'optimisation

$$\min_{f\in\mathcal{H}_K}\left\{\sum_{\ell=1}^{n}\sum_{j=1}^{m_\ell} E(f(x_{j\ell},\ell), y_{j\ell}) + \gamma\,\|f\|_{\mathcal{H}_K}\right\} \tag{4.11}$$

où $E : \mathbb{R}\times\mathcal{Y}\to\mathbb{R}$ est une fonction de perte convexe choisie et $\gamma > 0$ est un paramètre de régularisation à calibrer, par exemple, par validation croisée.

Le problème (4.11) admet une solution unique, qui par le théorème de représentation [KW70], peut être écrite sous la forme

$$\forall t \in \mathcal{X}, \forall q \in \mathbb{N}_n,\ f(t,q) = \sum_{\ell=1}^{n} \sum_{j=1}^{m_\ell} c_{j\ell} K((x_{j\ell}, \ell), (t,q)) \tag{4.12}$$

pour $c \in \mathbb{R}^M$, où $M = \sum_{\ell=1}^{n} m_\ell$. En remplaçant cette formule dans (4.11), on obtient le problème d'optimisation convexe sur les M variables :

$$\min \left\{ E_y(K_{\mathbf{x}} c) + \gamma\, c^\top K_{\mathbf{x}} c \ :\ c \in \mathbb{R}^M \right\} \tag{4.13}$$

où $K_{\mathbf{x}}$ est la matrice de noyau des données d'entraînement, et $E_y : \mathbb{R}^M \to \mathbb{R}$ est la fonction convexe (paramétrée par les données de sortie) définie par $E_y(z) = \sum_{\ell=1}^{n} \sum_{j=1}^{m_\ell} E(z_{j\ell}, y_{j\ell})$, $\forall\, z \in \mathbb{R}^M$. Si le graphe des relations entre tâches (et donc $K_{\mathbf{x}}$) est donné, alors résoudre (4.13) pour c - par exemple, par des méthodes de type SVM, régression ridge, etc. - puis utiliser (4.12) donne les fonctions pour chacune des n tâches.

On remarque que la fonction prédictive pour la tâche q ne dépend que de la colonne q de $(L + \varepsilon I_n)^{-1}$. En réalité, les valeurs de cette colonne influent sur la contribution des autres tâches dans la fonction prédictive. On peut aussi remarquer que dans le cas où les données d'observation sont communes à toutes les tâches, comme dans nombre d'applications, la matrice de noyau peut s'écrire

$$K_{\mathbf{x}} = (L + \varepsilon I_n)^{-1} \otimes G_{\mathbf{x}}, \tag{4.14}$$

où $G_{\mathbf{x}}$ est la matrice de noyau pour le noyau G appliqué aux données. Dans le cas général où les tâches peuvent posséder des échantillons de données distincts, chaque bloc (ℓ, q) de $K_{\mathbf{x}}$ est égal au produit de $(L + \varepsilon I_n)^{-1}_{lq}$ avec le bloc (ℓ, q) de $G_{\mathbf{x}}$. Dans toute la suite, nous définissons la matrice

$$Z = (L + \varepsilon I_n)^{-1}. \tag{4.15}$$

Nous notons aussi $T_{\mathbf{x}}$ l'application linéaire qui associe à la matrice Z de taille $n \times n$ la matrice $M \times M$ dont les blocs sont les produits de $Z_{\ell q}$ avec les blocs (ℓ, q) de $G_{\mathbf{x}}$.

4.3.2 Apprentissage simultané des tâches et du graphe

Pour rappel, notre objectif est d'apprendre les n tâches sans connaître la matrice L au départ. Nous voulons donc apprendre les tâches et le graphe au cours de la même procédure. Nous montrons que cela est réalisable via la méthode d'apprentissage des noyaux infiniment paramétrés, comme dans [AMP05]. Au lieu d'apprendre le laplacien de graphe directement, on optimise la fonction objectif (4.13) avec un noyau K appartenant à un ensemble $\mathcal{K}$:

$$\min \left\{ E_y(K_{\mathbf{x}} c) + \gamma\, c^\top K_{\mathbf{x}} c \ :\ c \in \mathbb{R}^M, K_{\mathbf{x}} \in \mathcal{K} \right\}\ . \tag{4.16}$$

L'objectif dans (4.16) est convexe, car la fonctionnelle

$$K \mapsto \min \left\{ E_y(K_{\mathbf{x}} c) + \gamma\, c^\top K_{\mathbf{x}} c \ :\ c \in \mathbb{R}^M \right\}$$

est convexe [AMP05, Lem. 2].

L'apprentissage conjoint de la fonction et du noyau a été largement utilisé et justifié par des bornes en théorie de l'apprentissage. On peut se référer, par exemple, à [LCB+04, RBCG08, SBD06, YC09].

Afin de s'assurer du fait que K est bien un noyau valide provenant d'un laplacien comme en (4.10), nous choisissons

$$\mathcal{K} = \left\{ T_{\mathbf{x}}(Z) : 0 \prec Z \preceq \tfrac{1}{\varepsilon} I_n, (Z^{-1})_{\text{off}} \leq 0, Z\mathbf{1}_n = \tfrac{1}{\varepsilon}\mathbf{1}_n \right\} \tag{4.17}$$

où on note $(Z^{-1})_{\text{off}}$ les coefficient extra-diagonaux de l'inverse de la matrice Z, et $\mathbf{1}_n$ le vecteur de n 1. Les contraintes assurent l'obtention d'un Z de la forme 4.15, provenant d'un laplacien ayant par définition des lignes sommant à zéro et dont les coefficients extra-diagonaux sont négatifs ou nuls. Cependant, l'ensemble $\mathcal{K}$ n'est pas convexe, à cause de la contrainte sur $(Z^{-1})_{\text{off}}$.

Outre les contraintes de validité, nous voulons encourager un faible nombre de clusters dans le graphe des tâches, afin d'obtenir des graphes plus simples et structurés. Dans le laplacien d'un graphe, la présence de clusters se manifeste sous la forme d'un saut dans le spectre, et la position du saut - donc le nombre de valeurs propres faibles - indique le nombre optimal de clusters. Sachant cela, la régularisation doit favoriser les laplaciens avec peu de valeurs propres faibles, ce qui implique un rang faible pour Z. Dans cette optique, une pénalité pertinente est la norme trace $\|Z\|_* := \sum_{i=1}^n \lambda_i(Z)$, où $\lambda_i(Z)$ indique la i-ème valeur propre de Z. Cette norme est connue pour être le substitut convexe optimal du rang [FHB01], et est dans ce cas égale à la trace de Z.

Ainsi, étant donné un noyau G, nous résolvons le problème

$$\min\Big\{ E_y(K_{\mathbf{x}}c) + \gamma\, c^\top K_{\mathbf{x}} c + \alpha \operatorname{tr} Z : \\ c \in \mathbb{R}^M, K_{\mathbf{x}} = T_{\mathbf{x}}(Z), 0 \prec Z \preceq \tfrac{1}{\varepsilon} I_n, (Z^{-1})_{\text{off}} \leq 0, Z\mathbf{1}_n = \tfrac{1}{\varepsilon}\mathbf{1}_n \Big\}, \tag{4.18}$$

où α est le deuxième paramètre de régularisation.

En appliquant les contraintes au problème primal (4.8), et en définissant la variable

$$Q = L + \varepsilon I_n \ , \tag{4.19}$$

on obtient le problème d'optimisation suivant :

$$\min\left\{ \sum_{\ell=1}^{n} \sum_{j=1}^{m_\ell} E(w_\ell^\top x_{j\ell}, y_{j\ell}) + \gamma \operatorname{tr}(WQW^\top) + \alpha \operatorname{tr}(Q^{-1}) : \right. \\ \left. W \in \mathbf{M}_{d,n},\ Q \succeq \varepsilon I_n,\ Q_{\text{off}} \leq 0, Q\mathbf{1}_n = \varepsilon \mathbf{1}_n \right\}. \tag{4.20}$$

La fonction objectif de ce problème est non-convexe tandis que l'ensemble admissible est convexe.

4.3.3 Résolution du problème d'optimisation

Même si le problème (4.20) n'est pas conjointement convexe en W et Q, il est convexe en l'une des variables dès que l'autre est fixée. Nous pouvons alors utiliser cette propriété pour obtenir un algorithme minimisation alternativement W et Q (algorithme 3).

L'étape 1 consiste à résoudre une méthode à noyaux telle que les SVM ou la régression à noyaux en dn variables. Nous avons utilisé des méthodes standard dans l'espace augmenté. Dans le cas particulier où les données d'observation sont communes, nous avons considéré une résolution en équation de Lyapunov, plus efficace.

Algorithme 3 Apprentissage conjoint des tâches et du graphe de tâches.

Paramètres : $\gamma, \alpha, \varepsilon$
Initialisation : Choisir $W^0 \in \mathbf{M}_{d,n}$
for $k = 1, 2, \ldots$ **do**
[1.] $W \leftarrow \operatorname{argmin}\Big\{\sum_{\ell=1}^{n}\sum_{j=1}^{m_\ell} E(w_\ell^\top x_{j\ell}, y_{j\ell}) + \gamma\ \operatorname{tr}(WQW^\top) : W \in \mathbf{M}_{d,n}\Big\}$
[2.] $Q \leftarrow \operatorname{argmin}\Big\{\operatorname{tr}(WQW^\top) + \alpha \operatorname{tr}(Q^{-1}) : Q \succeq \varepsilon I_n\ ,\ Q_{\text{off}} \leq 0, Q\mathbf{1}_n = \varepsilon \mathbf{1}_n\Big\}$
end for

L'étape 2 est un programme semi-défini, étant donné que les contraintes de type $\operatorname{tr}(Q^{-1}) \leq \beta$ peuvent être ré-écrites en $\operatorname{tr} R \leq \beta, \left(\begin{smallmatrix} R & I_n \\ I_n & Q \end{smallmatrix}\right) \succeq 0$. Le nombre de variables dépend uniquement de n, et pour un petit nombre de tâches (au plus 100), on peut le résoudre par des méthodes de points intérieurs et des bibliothèques comme SDPT3 ou SeDuMi. Toutefois, ces méthodes passaient mal à la dimension et nous avons dû développer notre propre méthode afin de traiter des n plus grands (algorithme 4). Cette méthode repose sur des projections positives semi-définies, et donc de décompositions en éléments propres. Elle a pu être calculée de façon exacte jusqu'à quelques milliers de tâches, et peut être étendue au-delà au moyen d'approximations de rang faible.

Algorithme 4 Séparation parallèle de Douglas-Rachford pour PSD à l'étape 2 de l'algorithme 3.

Paramètres : α, ε
Initialisation : Choisir $X_1 = X_2 = X_3 \succeq \varepsilon I_n$
for $k = 1, 2, \ldots$ **do**
[1.] $P \leftarrow \frac{1}{3}(X_1 + X_2 + X_3)$
[2.] Calculer la décomposition en éléments propres $X_1 = U \operatorname{Diag}(\lambda) U^\top$
[3.] $\mu_i \leftarrow \operatorname{argmin}\{\frac{1}{2}(\lambda_i - e)^2 + \alpha e^{-1} : e \geq \varepsilon\}\ \forall i \in \mathbb{N}_n$
[4.] $Y_1 \leftarrow U \operatorname{Diag}(\mu) U^\top$
[5.] $(Y_2)_{ij} \leftarrow \begin{cases} \min\{(X_2)_{ij}, 0\} & \text{if } i \neq j \\ (X_2)_{ij} & \text{if } i = j \end{cases} \quad \forall i, j \in \mathbb{N}_n$
[6.] $Y_3 \leftarrow B - BE - EB + (\frac{1}{n}\sum_{i,j=1}^{n} B_{ij} + \varepsilon)E$, where $B = X_3 - W^\top W$, $E = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$
[7.] $K \leftarrow \frac{1}{3}(Y_1 + Y_2 + Y_3)$
[8.] $X_1 \leftarrow X_1 + 2K - P - Y_1$
[9.] $X_2 \leftarrow X_2 + 2K - P - Y_2$
[10.] $X_3 = X_3 + 2K - P - Y_3$
end for
return $Q \leftarrow Y_3$

L'algorithme 4 repose sur le principe des algorithmes de séparation parallèle de type Douglas-Rachford (cf. [BC11, Prop. 27.8]). Les algorithmes de type Douglas-Rachford nécessitent le calcul d'*opérateurs de proximité*, qui étendent le concept de projection. On peut se référer par exemple à [BC11]. L'opérateur de proximité $\operatorname{prox}_f(x)$ d'une fonction semi-continue à gauche convexe $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ en $x \in \mathbb{R}^d$ est le minimiseur unique

$$\operatorname{prox}_f(x) = \operatorname{argmin}\left\{\tfrac{1}{2}\|y - x\|^2 + f(y) : y \in \mathbb{R}^d\right\}. \tag{4.21}$$

La première opération de proximité (étapes 2-4 de l'algorithme 4) est celle de la fonction

$$f_1(Q) = \begin{cases} \alpha \operatorname{tr}(Q^{-1}) & \text{if } Q \succeq \varepsilon I_n \\ +\infty & \text{sinon} \end{cases} . \tag{4.22}$$

Puisque f_1 est une fonction spectrale (une fonction sur les valeurs propres de Q seulement), l'opération de proximité revient à l'opération de proximité sur son spectre, par l'inégalité sur la trace de von Neumann [Mir75]. Cela se sépare à son tour en problèmes d'optimisation univariés, qui peuvent être résolus avec des équations cubiques [BMAP12, Lem. 4.1].
La deuxième opération de proximité (étape 5) est la projection sur l'ensemble $\{Q \in \mathbf{S}_n : Q_{\text{off}} \leq 0\}$, où on note $\mathbf{S}_n$ l'espace des matrices symétriques $n \times n$.
Enfin, la troisième opération de proximité (étape 6) est celle de la fonction

$$f_2(Q) = \begin{cases} \operatorname{tr}(WQW^\top) & \text{si } Q \in \mathbf{S}_n \text{ et } Q\mathbf{1}_n = \varepsilon \mathbf{1}_n \\ +\infty & \text{sinon} \end{cases} . \tag{4.23}$$

Cet opérateur de proximité est facile à trouver avec des multiplicateurs de Lagrange.

4.4 Résultats expérimentaux

Afin de vérifier la validité pratique de notre méthode multi-tâche par le laplacien ("Graph Laplacian MT"), nous avons effectué des tests en régression aux moindres carrés sur des données simulées et réelles. Un premier test sur un cas artificiel, généré par deux clusters de deux tâches chacun, valide le comportement attendu de la méthode, à savoir l'amélioration par rapport à l'apprentissage indépendant et la performance relative à des méthodes de référence. Le second test sur des données réelles de recommandation de films montre que l'on restitue le graphe des tâches tout en obtenant des prédicteurs efficaces. Nous avons comparé notre méthode avec deux bases de référence et trois approches de l'état de l'art :

- **[Independent learning]** La régression régularisée indépendante, où les tâches sont apprises séparément.
- **[True Laplacian]** L'apprentissage multi-tâche par noyaux 4.13 où le vrai laplacien est connu au lieu d'être estimé, dans le cas des données simulées. Cela constitue une "vérité terrain" que l'on s'attend à retrouver par l'apprentissage. Cette référence est
- **[Kernel with graph Laplacian]** L'approche multi-tâche proposée dans [EMP05] où le graphe est fixé comme étant celui que l'on retrouve par notre méthode. Pour rappel, cette approche n'apprend pas le laplacien, mais les résultats illustrent les défauts discutés dans la partie 4.2.3.
- **[Trace norm]** La régularisation par norme trace, qui est considérée comme état de l'art pour beaucoup de problèmes multi-tâches [SRJ05, AEP08]. Nous avons utilisé la bibliothèque SLEP (code disponible sur `http://www.public.asu.edu/~jye02/Software/SLEP`) décrite dans [LJY09].
- **[Clustered MT]** La méthode multi-tâche clusterisée formulée dans [JBV09], discuté en partie 4.2.3 et dont le code est disponible sur `http://cbio.ensmp.fr/~ljacob/`.

4.4.1 Données simulées

Nous avons validé notre méthode sur un jeu de données simulées consistant en deux groupes de deux tâches chacun, et dont le laplacien de graphe sous-jacent est :

$$0.85 \cdot \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

La méthode de génération de données suit en grande partie le protocole de [JBV09].

Chaque tâche t du cluster c est un vecteur w_t de coefficients de régression dans $\mathbb{R}^d$ ($d = 30$), généré en $w_t = w_c + \tilde{w}_t$, où w_c est le centre du cluster et $\tilde{w}_t$ un vecteur spécifique à la tâche t ayant les mêmes entrées non-nulles que w_c. Chaque élément non-nul de $\tilde{w}_t$ est tiré d'une loi $\mathcal{N}(0, \sigma_c^2)$, avec $\sigma_c^2 = 16$. Les centres de clusters w_c sont orthogonaux dans les $d-2$ premières dimensions : le premier cluster a $(d-2)/2$ éléments tirés de $\mathcal{N}(0, \sigma_r^2)$ ($\sigma_r^2 = 900$), et le second cluster a les $(d-2)/2$ autres éléments, générés de la même façon. Les deux dernières dimensions sont des bruits de loi $\mathcal{N}(0, \sigma_c^2)$ tirés pour chaque tâche.

Les observations X sont des vecteurs dans $\mathbb{R}^d$, générés selon une distribution uniforme sur $[0, 1]$, et la sortie correspondante Y_t pour la tâche t est $Y_t = w_t^\top X + \epsilon_y$, où ϵ_y est un bruit gaussien suivant $\mathcal{N}(0, \sigma_n^2)$ ($\sigma_n^2 = 150$). Les tâches partagent les mêmes observations en entrée.

Nous entraînons d'abord chaque méthode sur un ensemble d'entraînement de taille m, pour calibrer les paramètres sur un ensemble de validation de 500 points, avant de tester la méthode avec les paramètres optimaux sur un ensemble de test de 2000 échantillons. L'erreur quadratique moyenne sur l'ensemble des tâches – les sorties ont la même amplitude – sur l'ensemble de test est la mesure d'erreur pour chaque méthode. La figure 4.4 montre les résultats expérimentaux pour ce protocole, en moyenne sur 50 jeux de données générés ainsi, pour différentes tailles de données d'entraînement.

Les résultats nous mènent à trois conclusions principales :

— Si on exclut la taille d'entraînement la plus petite (40 points), la meilleure erreur correspond bien à la référence "True Laplacian", comme attendu. Lorsque le graphe est connu, la régularisation par le laplacien est pertinente est donne les meilleurs résultats.

— Notre approche restitue correctement les clusters et donne de meilleures performances jusqu'à 80 points d'entraînement.

— L'avantage du multi-tâche d'estompe lorsque la taille des données d'entraînement devient suffisamment large, car la sur-performance par rapport à l'apprentissage indépendant se réduit. L'une des qualités attendues du multi-tâche étant de pallier le manque de données, cet effet est cohérent avec l'intuition.

4.4.2 Recommandation de films

Notre approche a aussi montré son efficacité sur les données de MovieLens, un jeu de données souvent utilisé pour l'apprentissage multi-tâche et les systèmes de recommandation.

Les données MovieLens 100k

Les données MovieLens proviennent du site web de MovieLens (`movielens.umn.edu`) et sont disponibles sur `http://www.grouplens.org/node/73`. Le jeu de données "MovieLens 100k" contient les notes données à 1682 films par 943 utilisateurs. Chaque utilisateur a donné des notes entre 1 et 5 à un certain nombre de films de son choix. Les films sont décrits par leur titre, leur date de sortie et leur appartenance à 19 genres différents. On dispose d'informations sur les utilisateurs telles que l'âge, le sexe, la profession et le code postal de résidence (États-Unis).

Nous considérons la fonction de notation de chaque utilisateur comme une tâche de prédiction. Pour un film donné, il s'agit de prédire les notes que les utilisateurs donneront sachant les caractéristiques du film. Chaque film est caractérisé par le vecteur descripteur $(t, g_1, \ldots, g_d)$ où t est la date de sortie et g_i vaut 1 si le film appartient au genre i, et 0 sinon. Pour chaque tâche l, nous voulons régresser le vecteur de sortie Y_l de taille m_l sur la matrice descriptive X_l

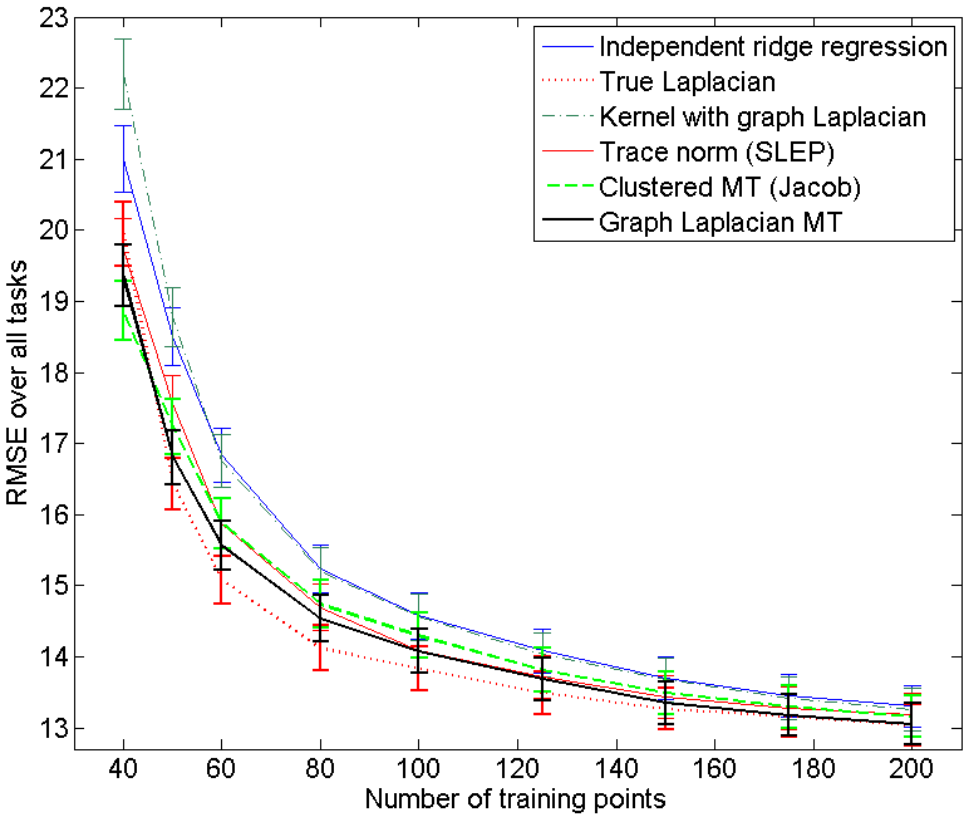


FIGURE 4.4 – Erreur test sur données simulées

de taille $m_l \times (d+1)$. Dans ce cadre, nous n'utilisons pas les informations supplémentaires sur les utilisateurs.

Nous avons utilisé les jeux de données d'entraînement et de test fournis, où 10 échantillons sont réservés au test pour chaque utilisateur et le reste est disponible pour l'entraînement et la validation. Les paramètres d'apprentissage ont été optimisés par validation croisée d'ordre 3 sur l'ensemble d'entraînement. Les tailles d'échantillons sont très variables selon les tâches, allant de 20 à 737 notes données, avec une médiane à 65. Dans la plupart des cas, il y a peu de données d'entraînement par tâche, et on peut donc s'attendre à une sur-performance des méthodes multi-tâches par rapport à l'apprentissage séparé.

Résultats

Le tableau 4.1 montre les erreurs de test, les erreurs de validation et les erreurs standard de la validation. Dans la validation, nous avons exclu les tâches possédant moins de 30 observations afin d'éviter les tâches avec trop peu de données et ainsi mieux estimer l'erreur. Ces cas mis de côté représentent environ un quart des tâches. On observe effectivement une meilleure adéquation de l'erreur de validation à l'erreur de test en excluant ces tâches, sans toutefois changer les paramètres optimaux. L'erreur standard de validation est donc plus pertinente pour représenter la variabilité de l'erreur en prédiction.

D'après le tableau 4.1, notre méthode montre la meilleure erreur de test. Il peut paraître étonnant que la méthode de multi-tâche clusterisé [JBV09] donne une erreur supérieure à l'apprentissage indépendant, alors que la formulation non-convexe est démontrée comme étant au

TABLE 4.1 – Erreur quadratique de test, erreur de validation et erreur standard (validation croisée) pour MovieLens 100k.

Méthode	Test	Validation	Erreur standard
Independent ridge	1.0896	1.0907	0.0027
Trace norm	1.0776	1.0875	0.0018
Clustered MT	1.0903	1.0900	0.0029
Graph Laplacian MT	1.0725	1.0695	0.0029

moins aussi efficace. Cela provient du fait que la relaxation convexe (4.4) n'est pas majorée en erreur par la régularisation en norme de Frobenius, à cause de la matrice de projection Π.

Il est intéressant de noter que notre méthode donne de meilleurs résultats que la norme trace, qui est une référence dans les problèmes de recommandation. Cela montre la prépondérance des clusters dans les données comme MovieLens.

Lorsqu'on étudie le laplacien obtenu, on distingue des clusters intéressants parmi les utilisateurs. Le spectre montre un nombre optimal de clusters autour de 5, et quelques groupes d'utilisateurs se distinguent par des caractéristiques particulières. Par exemple dans la figure 4.5, on distingue un groupe d'utilisateurs avec un fort coefficient négatif sur l'année de sortie des films, ce qui suggère un groupe d'amateurs de films anciens. Ou encore, on remarque un autre groupe constitué majoritairement d'étudiants. Enfin, les dimensions les plus différenciantes sont l'année de sortie et des genres comme "Action", "Comedy", "Drama" et "Thriller".

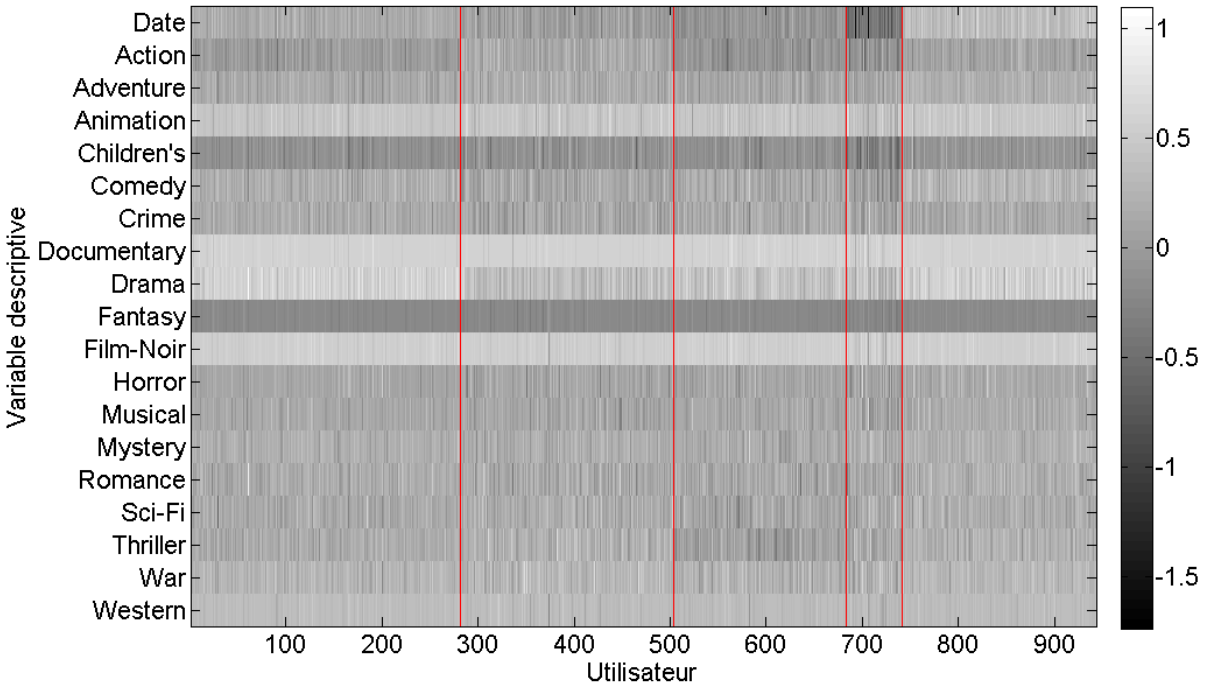


FIGURE 4.5 – MovieLens 100k : coefficients de régression obtenus, clustering des utilisateurs en 5 groupes.

4.4.3 Indices financiers

Comme nous souhaitons appliquer l'apprentissage multi-tâche aux données financières, nous testons ici un problème de régression. Les données d'entraînement sont construites par la méthode de la partie 2.3 avec 50 actifs, dont des indices d'actions, des taux d'intérêt, des matières premières, etc. En utilisant uniquement les tendances et volatilités du dernier régime, l'espace d'entrée est de dimension $d = 50$. L'historique des données s'étend de 2000 à 2012. Les sorties à prédire sont les excès de rendements (par rapport au marché) futurs d'indices sectoriels représentant les secteurs d'activité en Europe [1]. Les secteurs peuvent se regrouper en clusters, car certains secteurs sont similaires en termes de risque. Par exemple, le secteur bancaire est souvent corrélé au secteur de l'assurance ; le secteur alimentaire est souvent corrélé au secteur de la santé, étant tous deux des secteurs dits "défensifs", c'est-à-dire peu sensibles aux variations du marché. Ces faits font de ce problème un candidat pertinent pour l'apprentissage multi-tâche, et nous nous attendons à une meilleure performance de prédiction.

Résultats

Nous adoptons le protocole de test par fenêtres glissantes de 250 jours d'entraînement suivis de 50 jours de test, décrit ultérieurement en partie 5.1. Nous utiliserons les $k_{validation}$ premières fenêtres pour valider les paramètres et les suivantes pour le test de l'algorithme.

Nous avons testé un ensemble de 7 secteurs : Food & Beverage, Health Care, Personal & Household Goods, Retail, Financial Services, Banks, and Insurance. D'un point de vue statistique et financier, les quatre premiers peuvent former un cluster et les trois autres sont souvent dans un autre cluster. Toutefois, la dépendance entre eux varie au cours du temps, donc effectuer un clustering en amont sur un historique large n'est pas pertinent. Le tableau 4.2 compare les performances entre notre méthode, la régression indépendante et la méthode de multi-tâche clusterisé. La validation a été effectuée sur toutes les fenêtres se terminant au plus tard en janvier 2009. La figure 4.6 montre en outre que les clusters sont restitués de façon plus claire par notre méthode.

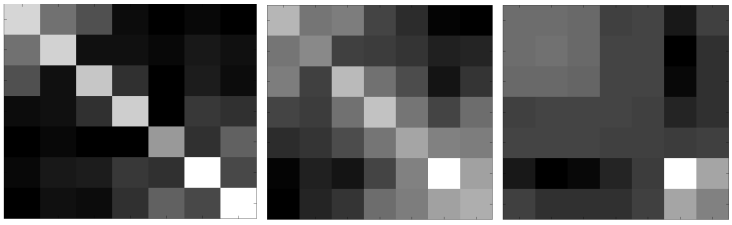


FIGURE 4.6 – Laplacien (en valeurs absolues) à retrouver sur la dernière fenêtre (gauche) ; $W^\top W$ restitué par Graph Laplacian MT (milieu) et Clustered MT (droite).

4.4.4 Discussion

L'application de notre méthode à la prédiction des signes du rendement n'est malheureusement pas immédiate. Il faut trouver une fonction de perte qui soit à la fois adaptée au problème et qui admette une méthode de résolution. Le tableau d'erreurs 4.2 montre en réalité qu'il ne suffit pas de transposer le problème de classification en régression en utilisant le rendement au

1. comme définis dans la classification ICB : http://www.icbenchmark.com au niveau Supersecteur. Les indices sont définis et maintenus par STOXX : http://www.stoxx.com/indices/types/sector.html.

TABLE 4.2 – Erreur de prédiction (MSE) et erreur standard pour les 7 indices sectoriels

Méthode	Independent ridge	Clustered MT	Graph Laplacian MT
Test (SE) $\times 10^{-3}$	2.03 (0.6552)	1.97 (0.6014)	1.85 (0.5084)

lieu de son signe, car les erreurs sont en réalité très grandes par rapport aux réponses.

Même si les exemples ci-dessus ne sont pas de très grandes dimensions, il peut s'agir d'un point d'entrée intéressant pour traiter des données de grande échelle. Le problème initial est difficile car il nécessite l'optimisation conjointe des tâches et du graphe de relations. Il est non-convexe et sa complexité dépend à la fois de la dimension des observations, de la taille des échantillons et du nombre de tâches. La méthode proposée contourne ces difficultés en séparant le problème en deux sous-problèmes sensibles à des dimensions différentes. L'étape de régression dépend de la taille des données et de la dimension des observations, tandis que l'étape de résolution du laplacien dépend essentiellement du nombre de tâches. En séparant les complexités, l'optimisation alternée peut faire usage de méthodes employées dans les problèmes à grande échelle. Par exemple, on peut envisager une distribution du problème sur des sous-échantillons des données afin de trouver une première approximation du graphe. Une fois ce graphe approché, on peut résoudre les tâches en se concentrant sur les tâches en relation, et ainsi traiter le problème par sous-ensembles plus petits de tâches.

Chapitre 5

Résultats expérimentaux

Les méthodes proposées dans les chapitres précédents forment un processus allant des données brutes à un résultat de prédiction accompagné d'une mesure de confiance. Toutefois, le processus ne s'arrête pas là. D'une part, la qualité de la prédiction dépend de paramètres à tous niveaux. La validation de ces paramètres peut être une tâche coûteuse, surtout que les paramètres optimaux peuvent dépendre de l'objectif de prédiction et du contexte. D'autre part, nous ne supposons pas que les modèles seront valides pour une durée de temps illimitée après entraînement. Il est donc nécessaire de suivre les résultats de chaque modèle et d'étudier la performance statistique avec un protocole rigoureux et des mesures pertinentes. Le ré-entraînement des modèles est essentiel lorsque des informations nouvelles apparaissent ou lorsque les paramètres ne sont plus adaptés.

Par conséquent, le processus global peut être résumé par la figure 5.1. Nous proposons dans un premier temps un protocole de test par fenêtres glissantes, rendant compte de l'utilisation pratique dans le temps. La performance, reflétée par la précision et le rendement des prédictions, peut être vue à la fois dans le temps et en fonction du nombre de jours d'utilisation des modèles. Nous discuterons notamment de la qualité des indicateurs de confiance proposés.

Dans un second temps, nous étudierons la sensibilité des performances aux paramètres. Il y a notamment une grande variété de comportements, et certains paramètres peuvent mener à du sur-apprentissage. Le choix des paramètres est donc critique pour assurer la qualité d'ensemble de la démarche. Si certains paramètres sont intrinsèques à la qualité statistique des algorithmes, d'autres ne peuvent pas être optimisés de façon universelle à tout contexte. La calibration pourra être dynamique en maintenant l'historique détaillé des performances pour différents jeux de paramètres.

5.1 Protocole de test et mesures de performance

La nature chronologique des données requiert un protocole de test adéquat. Les méthodes correspondant au cadre iid optimisent souvent les paramètres par validation croisée, ou bien testent la performance en généralisation en réservant des données aléatoirement choisies. Dans notre situation, ce genre de procédure ne correspond pas à une utilisation possible des prédictions. Lorsque l'on les utilise, c'est nécessairement pour prédire le futur. De plus, l'erreur a de grandes chances d'être sous-estimée en un point compris dans l'intervalle de temps des données d'entraînement.

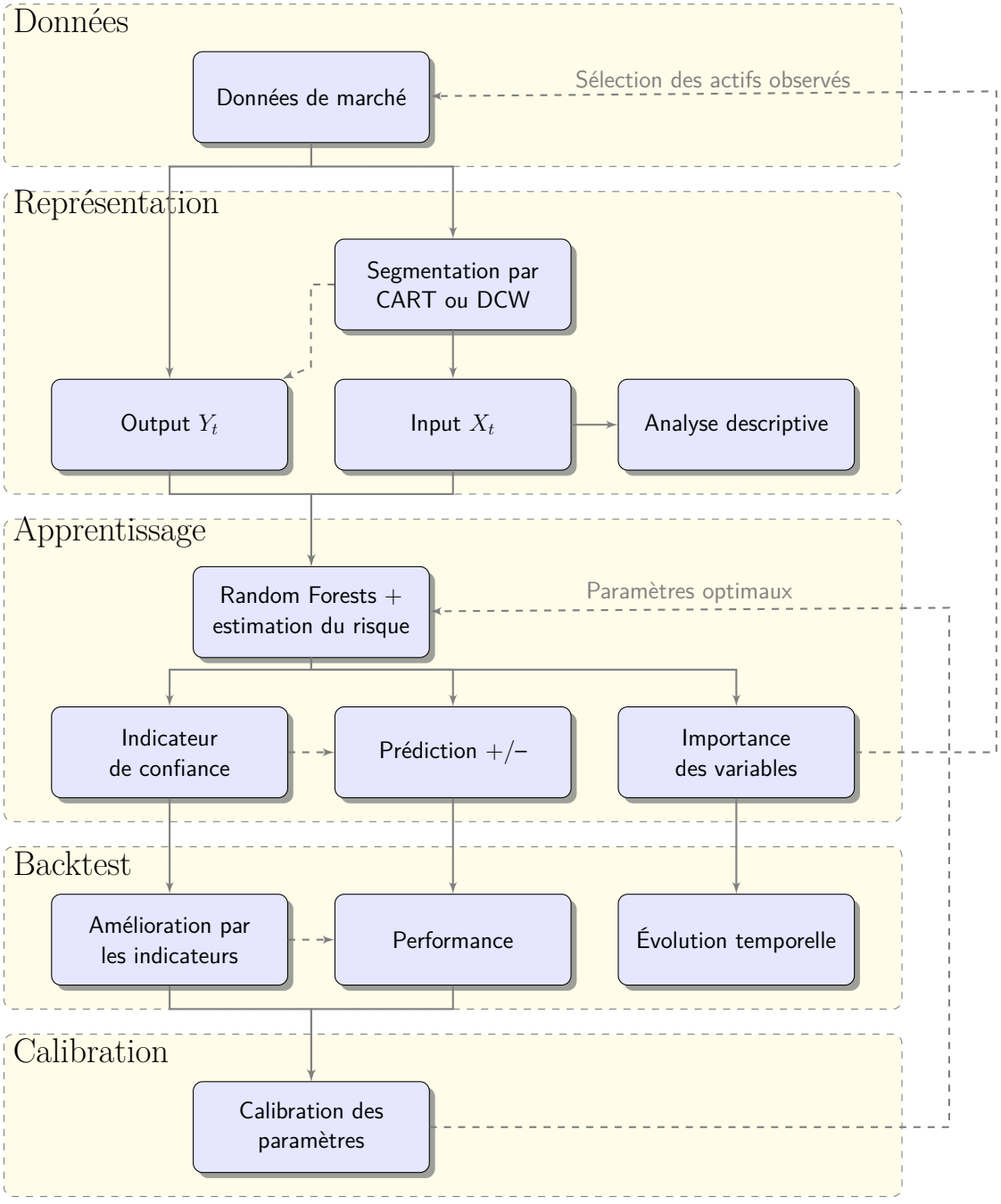


FIGURE 5.1 – Schéma fonctionnel des travaux

5.1.1 Backtest par fenêtres glissantes

Dans la mise en œuvre réaliste de la procédure de prédiction, l'utilisateur entraîne à la date t un modèle prédictif sur une fenêtre d'entraînement des T dernières données historiques disponibles. Le modèle construit est utilisé sur un certain nombre τ de jours à partir de la date suivante, inconnue de l'historique d'entraînement. Le test de performances historiques, ou *backtest*, doit correspondre à cette utilisation concrète pour estimer correctement la qualité effective de la méthode. On utilise dans toutes les expériences un protocole par fenêtres glissantes (figure 5.2) :

— un nouveau modèle est entraîné chaque jour sur les T dernières observations, puis utilisé pour prédire τ jours consécutifs à partir du jour suivant l'entraînement,
— la prédiction à la date t consiste à prédire le signe du rendement entre t et $t+h$, où h est un horizon fixé, à partir de l'observation X_t des D variables prédictives à la date t,
— la règle de décision suivant la prédiction $f(X_t)$ est, si $f(X_t) = 1$, d'acheter l'actif cible en t et de le revendre en $t+h$, et si $f(X_t) = -1$, de vendre à découvert l'actif cible en t et de le racheter en $t+h$.

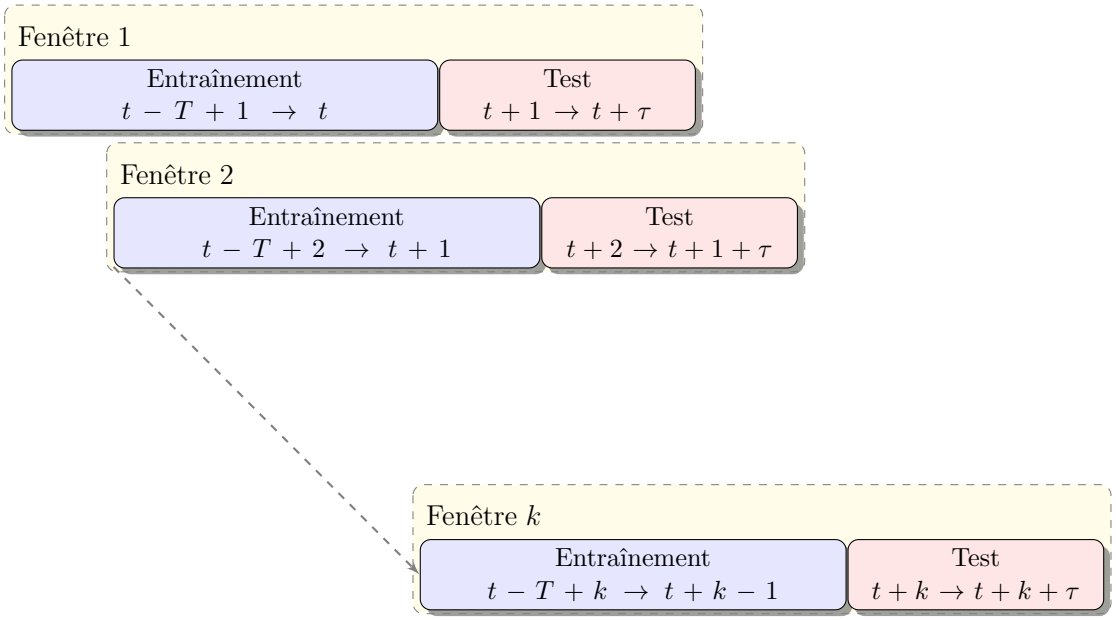


FIGURE 5.2 – Test par fenêtres glissantes

Ce protocole garantit un calcul de performances simples : la sortie est $Y_t = \operatorname{sgn} R_{t,h}$, la précision d'un modèle se mesure bien par l'adéquation entre les signes du rendement prédit et réalisé $(L(Y_t, f(X_t)) = \mathbb{1}_{Y_t = f(X_t)})$, et le gain est le rendement réalisé par la prédiction, $f(X_t)R_{t,h}$. Toutefois, il est à noter que ce protocole est en toute rigueur irréalisable en pratique : on suppose être capable de construire le modèle prédictif dès que les données de la date t sont disponibles, et de prendre une décision et exécuter instantanément une transaction. Cela est impossible, car d'une part la prédiction n'est pas instantanée, d'autre part les données sont disponibles par définition à la clôture du marché, soit au moment où on ne peut plus exécuter de transaction. En quelque sorte, le processus de décision ne vérifie pas l'hypothèse d'efficience des marchés. En toute rigueur, il faudrait supposer que la prise de position s'effectue pendant le jour suivant, voire à l'ouverture. Cependant, cela rend l'évaluation des performances complexe, car il s'agirait d'évaluer la qualité de l'algorithme d'apprentissage avec des données de nature différente de ses données d'entraînement. Heureusement, le protocole n'est pas si déraisonnable si l'on considère que les prix obtenus peu avant la clôture sont proches du prix de clôture. Dans ce cas, il s'agirait de construire le modèle et prendre position juste avant la clôture.

Dans la suite, sauf mention contraire, nous prendrons l'exemple de la prédiction de l'indice EURO STOXX 50 à un horizon de 25 jours (environ un mois). L'indice EURO STOXX 50 est un indice constitué des 50 entreprises les plus représentatives de la zone euro. Il représente bien les marchés de la zone euro, qui sont particulièrement intéressants à étudier à cause du contexte des dernières années. La période de test va de juin 2008 à juin 2014, soit sur 6 ans. La représentation

choisie est DCW, avec le même paramètre pour toutes les expériences.

5.1.2 Classement des variables

Le backtest fournit un classement des variables pour chaque modèle, et par conséquent à chaque date. La figure 5.3 montre les 35 variables de poids (réduction d'impureté) non nul sur l'ensemble du backtest.

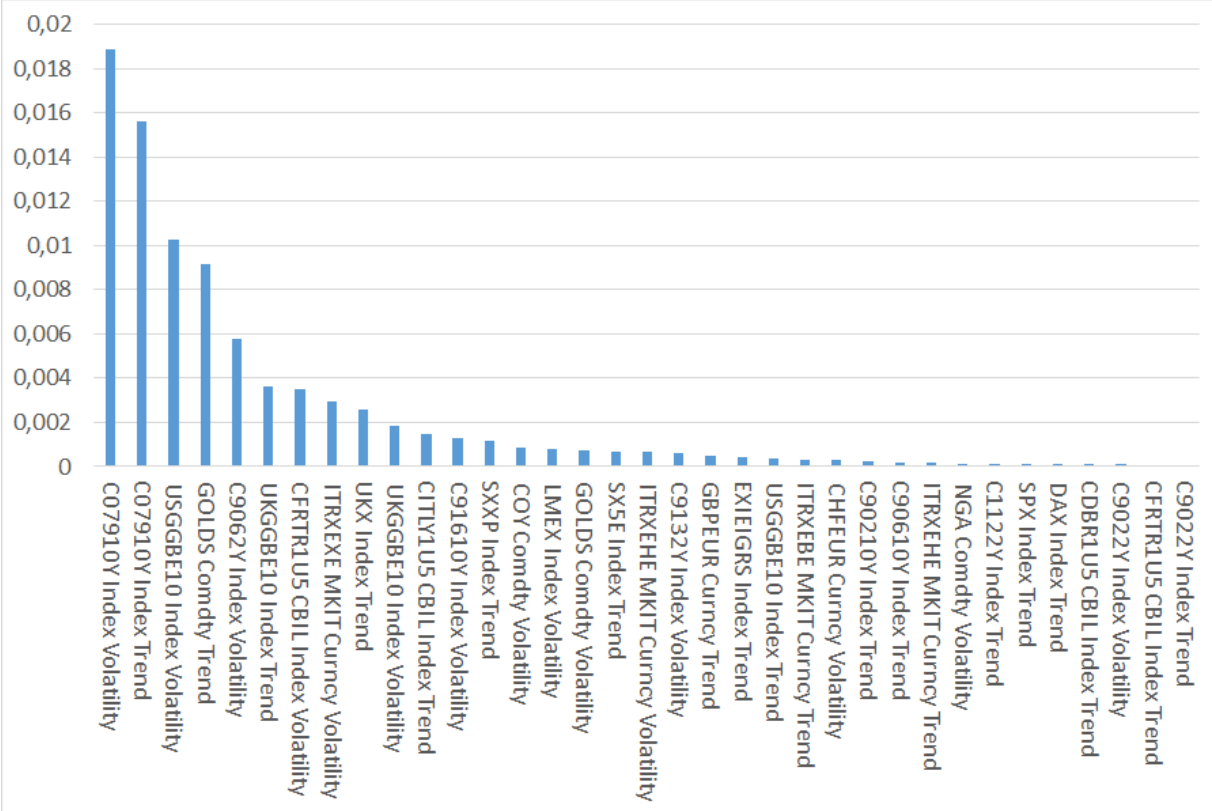


FIGURE 5.3 – Réduction d'impureté moyenne des variables significatives

Cependant, une analyse statistique des poids ne suffit pas, car l'importance des variables change dans le temps. Typiquement, l'importance de la deuxième variable (les taux 10 ans américains, en tendance) est localisée dans le temps, comme le montre la figure 5.4. Cette localisation temporelle reflète souvent des événements ou un contexte économique particulier.

Dans certains cas, il peut être intéressant de suivre l'évolution temporelle du rang de chaque variable. Une augmentation rapide du rang d'une variable jusque-là muette dans la prédiction peut être un indicateur de changement économique. Le classement des variables peut aussi permettre de mesurer la similarité entre deux dates par la corrélation des rangs (τ de Kendall [Ken38]). Dans l'exemple de la figure 5.5, les périodes larges de corrélation élevée sont vues comme des périodes de stabilité, tandis que les discontinuités marquent des changements de mode de marché.

5.1.3 Mesures de performance

Le but du backtest n'est pas de mesurer la performance de modèles en particulier, mais bien de la procédure dans son ensemble, par répétition jour après jour de l'apprentissage et prédic-

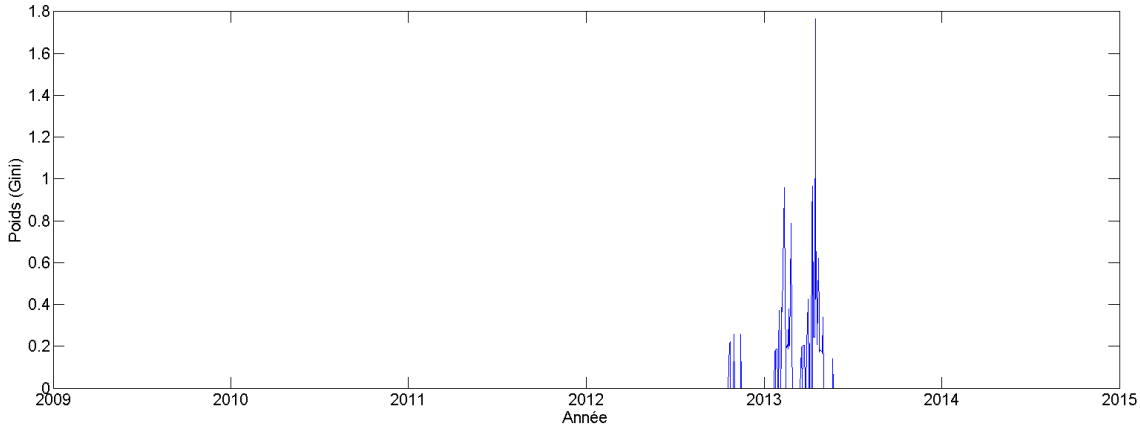


FIGURE 5.4 – Poids des taux 10 ans américains dans la prédiction de l'EURO STOXX 50

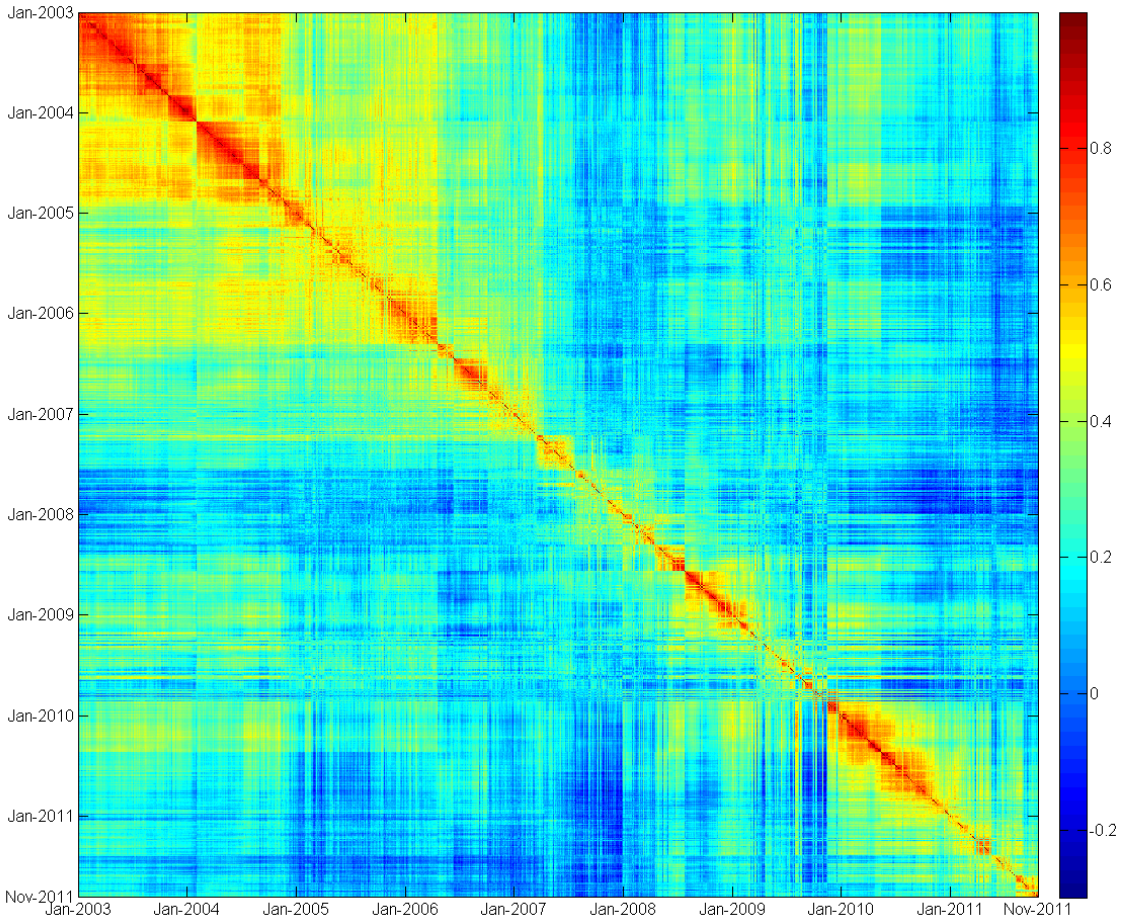


FIGURE 5.5 – Corrélation des rangs des variables prédictives pour l'indice STOXX Europe 600

tion avec les mêmes paramètres. L'analyse de la performance s'effectue sur deux dimensions : le modèle f_t, indexé par sa date de construction t, et le nombre de jours de test suivant la construction, Δt. Comme la dernière donnée disponible à la date t est le prix en t, la dernière sortie d'entraînement est $R_{t-h,h}$. Par conséquent, le premier rendement de test possible est $R_{t-h+1,h}$, dont la prédiction est $f_t(X_{t-h+1})$. en tenant compte de cela, nous nous intéressons aux mesures

suivantes pour $t \in \{1, \dots, t_{\max}\}$ et $\Delta t \in \{1, \dots, \Delta t_{\max}\}$:

— perte en classification $H(t, \Delta t) = \mathbb{1}_{f_t(X_{t-h+\Delta t})=Y_{t-h+\Delta t}}$,

— rendement réalisé $P(t, \Delta t) = f_t(X_{t-h+\Delta t}) R_{X_{t-h+\Delta t},h}$.

Néanmoins, étant donné qu'il s'agit de classification, l'algorithme n'est pas supposé maximiser le gain et celui-ci ne montre pas la performance statistique de la procédure. Les tests se concentreront donc sur la précision dans un premier temps, avant de faire intervenir le rendement effectif lors de l'application en temps réel.

Afin de prendre en compte des déséquilibres de performance entre les hausses et les baisses, on pourra aussi considérer la matrice de confusion, constituée des vrais positifs (VP), vrais négatifs (VN), faux positifs (FP) et faux négatifs (FN) :

	Prédit +	Prédit –
Réalisé +	VP	FN
Réalisé –	FP	VN

Carte de performances

La carte de précision moyenne (figure 5.6) est une visualisation directe de H pour la prédiction du signe du rendement de l'EURO STOXX 50, pour un certain jeu de paramètres. Le point de coordonnées $(t, \Delta t)$ montre la précision en classification du modèle t depuis sa construction, soit

$$\tilde{H}(t, \Delta t) = \sum_{i=1}^{\Delta t} H(t, i) \ .$$

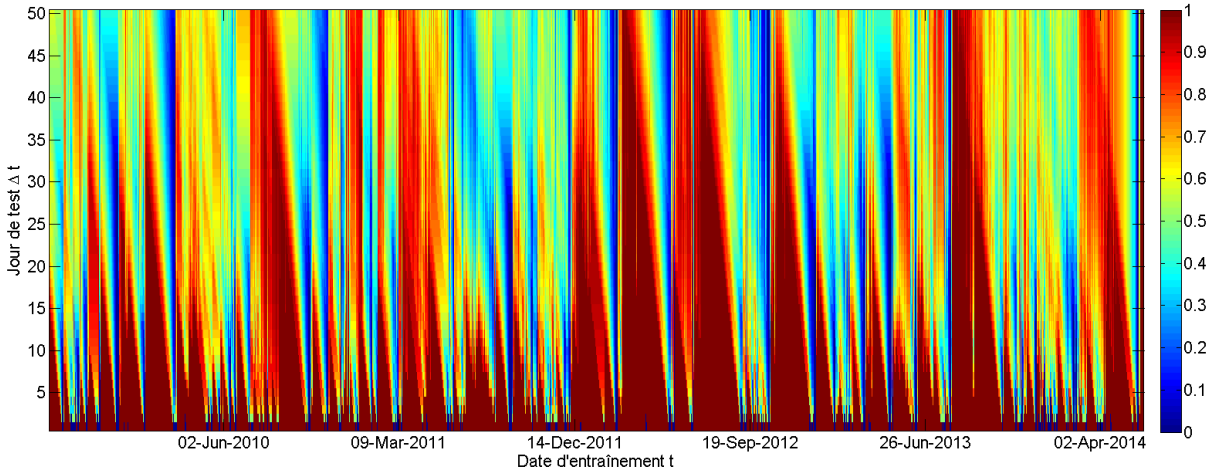


FIGURE 5.6 – Carte de précision moyenne pour l'EURO STOXX 50

L'évolution verticale est le suivi de la précision moyenne à modèle fixé. On peut noter que les droites d'équation $y = t - x$ représentent tous les points acquis à la date t. En effet, à la date t, les données au-delà de la droite d'équation $y = t - x$ sont postérieures à l'observation.

D'après ce graphique, on vérifie l'intérêt d'un suivi des performances dans le temps :

— la précision dépend du contexte : par exemple, la période de décembre 2011 à septembre 2012 bénéficie d'une précision plus élevée,

— la précision dépend de la durée d'utilisation d'un même modèle : elle semble plus élevée pour les dates proches de la date d'entraînement, et s'estompe lorsque le modèle n'est pas mis à jour.

Le second point peut avoir une interprétation économique : les arbitrages identifiés par l'apprentissage sont rapidement exploités et invalidés par les acteurs de marché. Cependant, nous verrons immédiatement dans la partie suivante que cette précision décroissante s'explique autrement.

Précision par durée de test

Agrégeons la précision selon l'axe temporel sur les N dates, afin de représenter la précision moyenne par durée de test, soit :

$$< H >_t (\Delta t) = \frac{1}{N} \sum_{t=1}^{N} H(t, \Delta t) \ .$$

On obtient la figure 5.7.

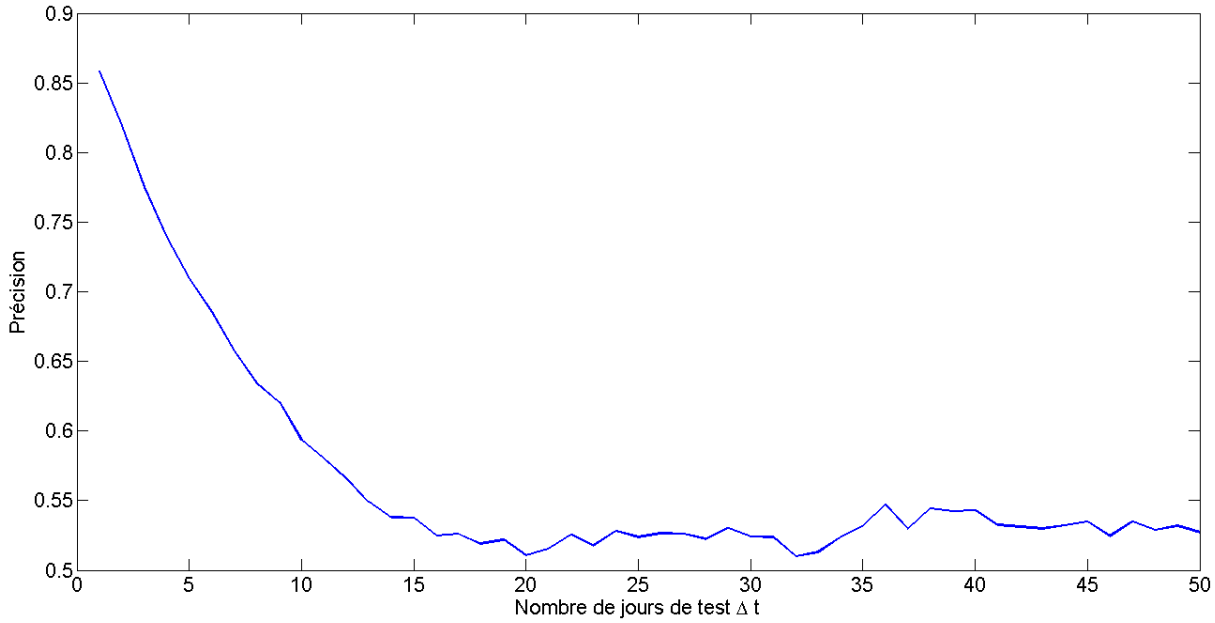


FIGURE 5.7 – Précision par durée de test

La valeur de 85% pour $\Delta t = 1$ est en réalité très proche de la précision d'entraînement. Étant donné que l'horizon est de 25 jours, les 24 premiers points ne peuvent pas être totalement considérés comme des points de test. Pour $\Delta t < h$, même si le rendement $R_{t-h+\Delta t,h}$ n'est effectivement pas connu, une partie de l'information est déjà contenue dans dans les prix entre $t-h+\Delta t$ et t ! Cela explique pourquoi la performance décroît de la performance d'entraînement vers une performance de généralisation qui semble stable ensuite. Intuitivement, prédire le signe du rendement entre $t-h+\Delta t$ et $t+\Delta t$ sachant les prix jusqu'à t est effectivement un problème plus facile. De plus, ces points de test ne sont pas pertinents car on ne peut pas les utiliser dans une prise de décision concrète. Les points de test antérieurs à l'horizon, n'étant ni des points d'entraînement ni des points de test, seront donc exclus des calculs de performance. Toutefois, la courbe de précision par durée de test est un outil intéressant pour visualiser le sur-apprentissage – qui est flagrant dans l'exemple – par la convexité de la courbe.

Précision dans le temps

La réduction de la carte de précision dans l'autre dimension donne une précision moyenne dans le temps, représentée dans la figure 5.8, avec un lissage par moyenne sur une fenêtre glissante

de 25 jours, afin de conserver la lisibilité.

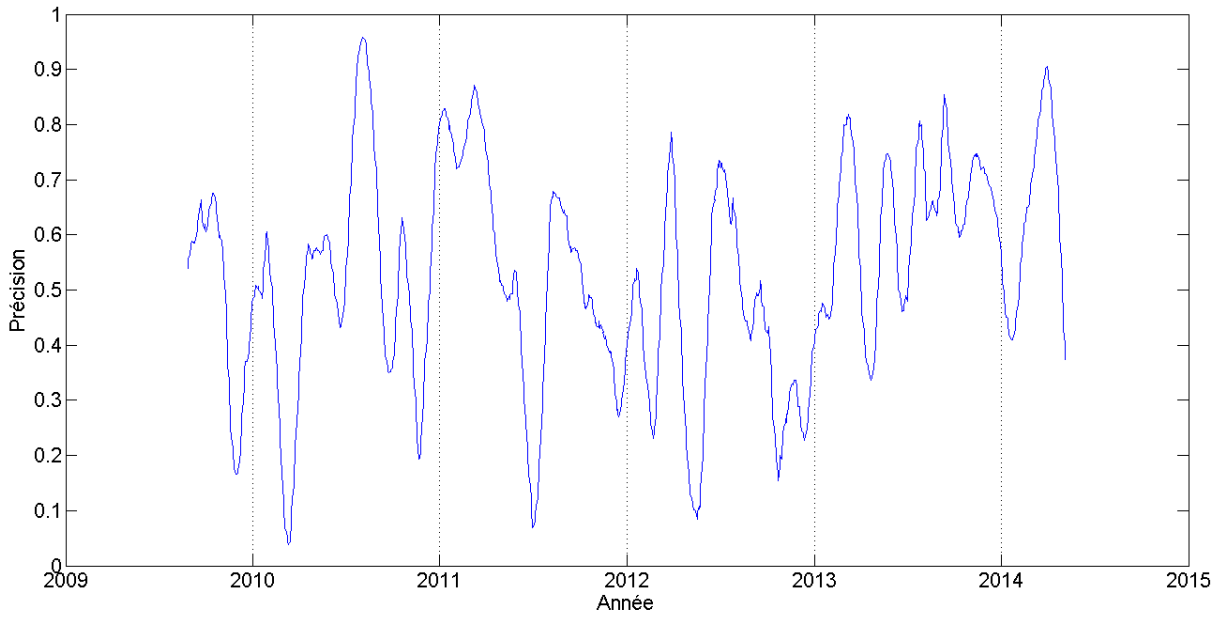


FIGURE 5.8 – Précision par date d'entraînement

La confrontation de ce comportement avec l'expertise financière permet de voir que les chutes de précision correspondent aux périodes de changement de tendance, et que les améliorations de précision au lieu pendant des périodes relativement stables. On peut interpréter cela comme un manque de robustesse de la méthode (avec les paramètres) face aux changements importants dans le comportement des actifs. Cela peut provenir en partie du sur-apprentissage constaté plus haut. Les taux de faux positifs ($\frac{FP}{VP+FP}$) et de faux négatifs ($\frac{FN}{VN+FN}$) montrés en figure 5.9 suggèrent des biais vers les classes correspondant aux tendances récentes.

5.2 Calibration de paramètres

Comme nous l'avons vu dans l'exemple illustratif des performances de la méthode, les modèles entraînés peuvent être pénalisés par du sur-apprentissage, les rendant moins robustes aux variations de contexte. Afin d'avoir une idée plus précise de l'influence des paramètres d'apprentissage sur la performance, nous avons réalisé des backtests sur une série de jeux de paramètres.

5.2.1 Paramètres testés

Nous avons testé les paramètres suivants de l'algorithme d'apprentissage :

— Nombre d'arbres dans la forêt aléatoire : 3, 11, 51, 101. Ce paramètre influence la convergence et la stabilité des fonctions prédictives. Le choix de la taille de la forêt permet de sélectionner le bon compromis entre la stabilité et le coût en calculs.

— Taux d'échantillonnage des données pour le bootstrap : 100%, 70%, 30%. Ce taux reflète la diversité des arbres de la forêt.

— Proportion de feuilles à élaguer (à la place du paramètre λ) : 0%, 10%, 50%, 80%. Nous vérifions la nécessité d'élaguer les arbres pour éviter le sur-apprentissage.

— Taille de la base d'apprentissage (jours) : 65, 130, 250, 380. La question de la quantité d'historique à conserver est intéressante pour l'interprétation en tant que longueur de

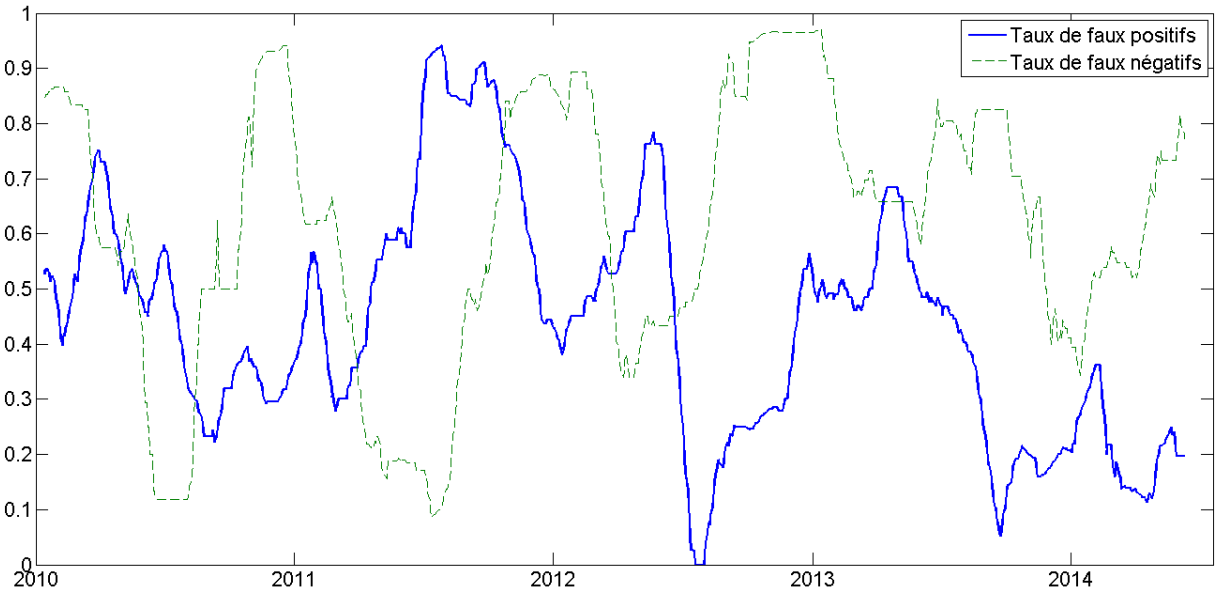


FIGURE 5.9 – Taux de faux positifs et de faux négatifs

mémoire de l'information passée à conserver pour expliquer le présent.
Parmi les autres paramètres de la méthode, nous ne testons pas ici le nombre de variables à utiliser à chaque division, ni le choix de la représentation.

5.2.2 Résultats

Les résultats sont synthétisés en première approche sous forme de courbe de précision par durée de test (figure 5.10). Ce graphique montre une grande amplitude de performances (15%) selon le choix des paramètres.

Sur-apprentissage

On distingue rapidement deux types de courbes : celles qui ont une forte convexité avec une décroissance importante de précision sur les données intermédiaires entre entraînement et test ($\Delta t < 25$), et celles qui sont proches de constantes. Les premières sont caractéristiques du sur-apprentissage. La figure 5.11 met en évidence l'influence prépondérante de la proportion de feuilles élaguées. Les courbes ayant la même valeur de ce paramètre ont des formes très similaires.

D'un point de vue d'apprentissage statistique, on peut définir un critère de sur-apprentissage y comme étant la différence entre la précision à $\Delta t = 1$ et la précision à $\Delta t = 25$. Les observations x sont les valeurs des quatre paramètres testés. L'arbre de régression construit par CART sur ces données (x en entrée et y en sortie) montre aussi que le taux d'élagage est la variable discriminante (figure 5.12).

En conclusion, l'élagage des arbres dans Random Forests est nécessaire pour éviter le sur-apprentissage.

Performance

Nous nous intéressons bien entendu aux choix de paramètres permettant d'améliorer la performance dans l'ensemble. Malheureusement, la réponse n'est pas aussi évidente que pour le

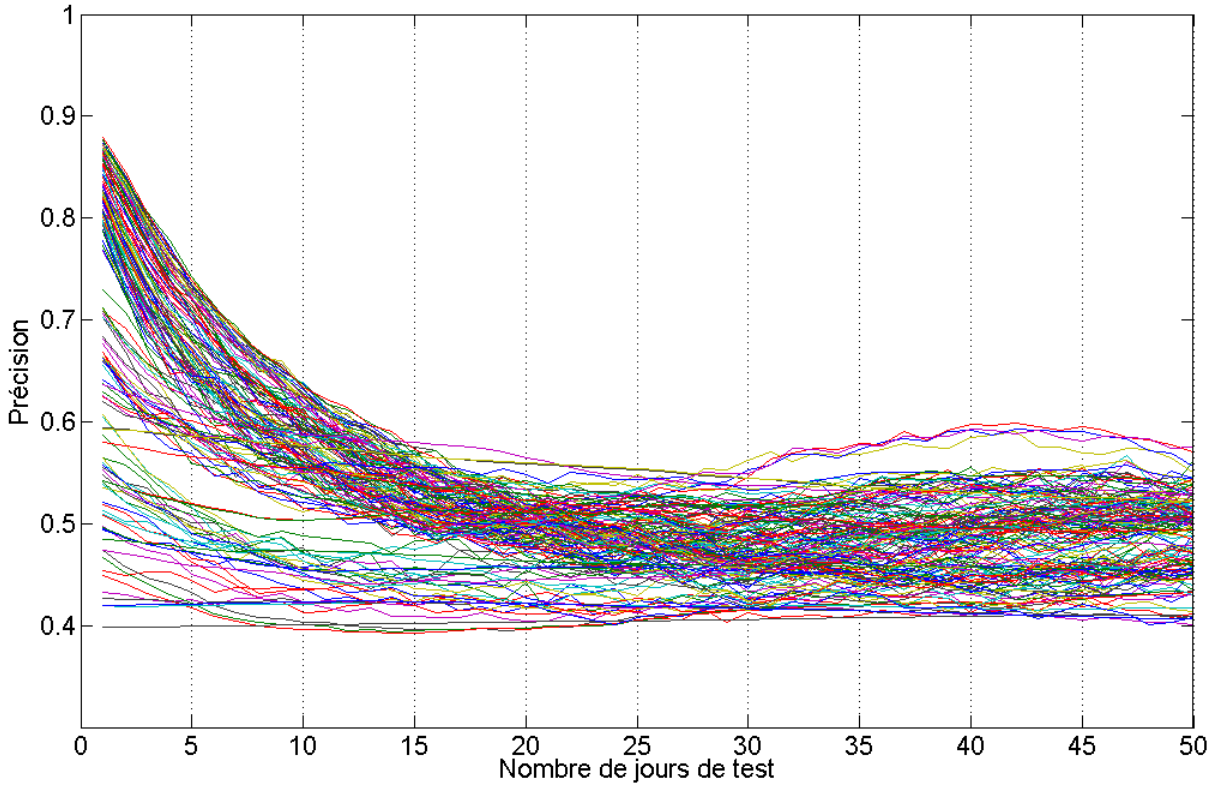


FIGURE 5.10 – Influence des paramètres sur la performance de la procédure de prédiction

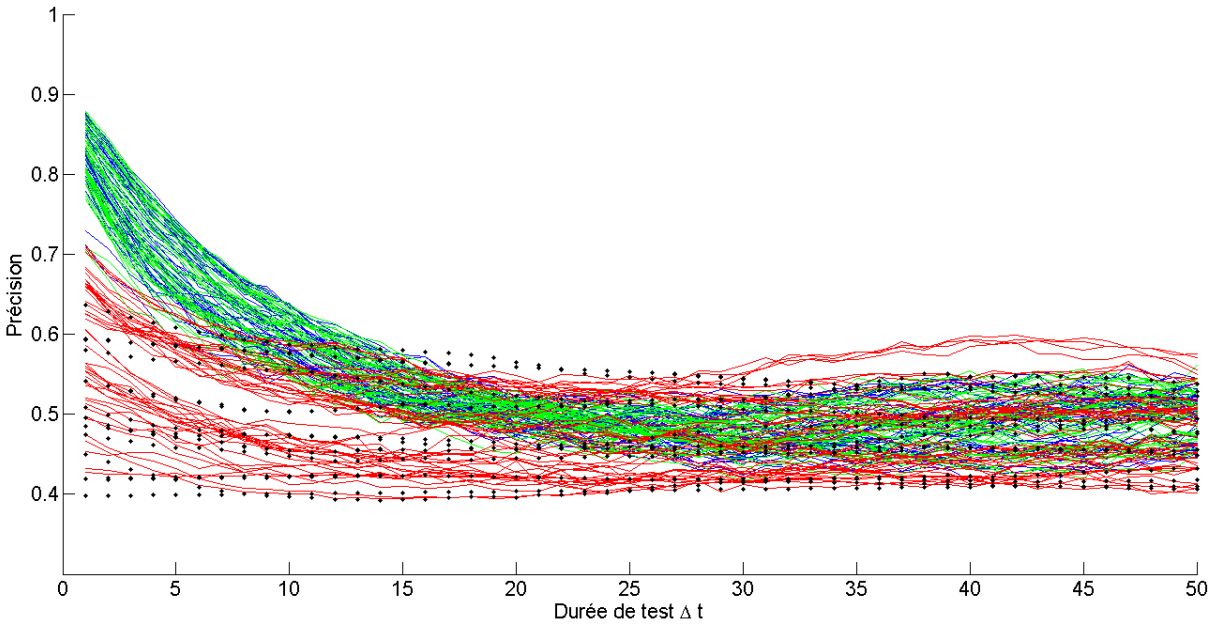


FIGURE 5.11 – Précisions des quatre niveaux d'élagage : 0% (bleu), 10% (vert), 50% (rouge), 80% (points noirs)

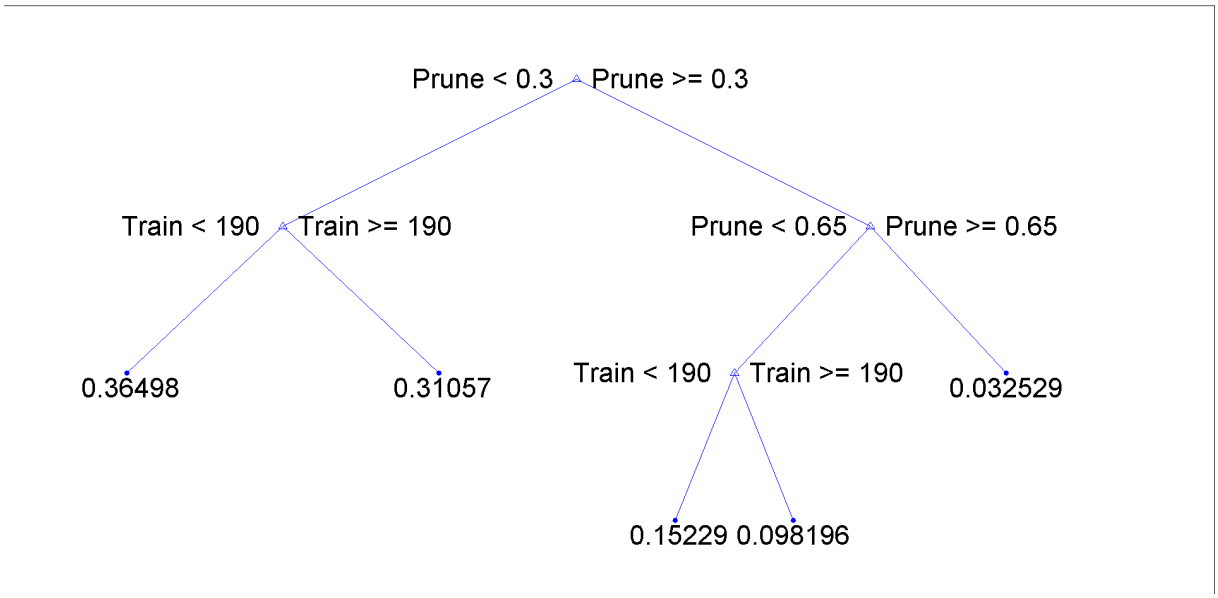


FIGURE 5.12 – Arbre de régression (CART) pour le critère de sur-apprentissage : le taux d'élagage ("Prune") apparaît prédominant

sur-apprentissage, mais nous pouvons de façon similaire construire un arbre de régression sur les valeurs de paramètres en entrée et, par exemple, la précision à 25 jours en sortie. L'arbre de la figure 5.13 obtenu par CART semble montrer que les deux paramètres les plus importants sont le taux de bootstrap ("Brate") et le taux d'élagage.

On peut effectivement vérifier que les courbes ayant la meilleure précision de test ont un taux de bootstrap de 100%. Cela peut paraître paradoxal étant donné que ce taux règle la diversité des arbres, qui contribue à améliorer la qualité de la forêt. Il est possible que le facteur caché ici soit plutôt la quantité de données disponibles, car un teux d'échantillonnage plus faible réduit aussi la quantité de données d'entraînement pour chaque arbre.

5.2.3 Sélection dynamique des paramètres

Des expériences plus complètes, avec des backtests sur plusieurs actifs, montrent que certains paramètres de notre procédure ne peuvent pas être optimisés, car leur effet dépend du contexte et de l'actif cible. C'est notamment le cas de la taille de la base d'apprentissage. Dans le cadre classique, prendre le plus long historique possible permet d'obtenir la meilleure qualité d'estimation. Pour les actifs financiers, prendre trop d'historique n'est pas nécessairement pertinent, car le fonctionnement des marchés évolue au cours du temps. Ainsi, certaines structures de dépendance passées peuvent être invalides aujourd'hui, et les intégrer aux données d'entraînement peut au contraire nuire à la capacité prédictive récente.

Dans la recherche d'optimalité des paramètres, ces paramètres peuvent être calibrés en maintenant des backtests sur plusieurs jeux de paramètres. L'analyse de leurs performances historiques permet de choisir dynamiquement les valeurs optimales pour un certain critère.

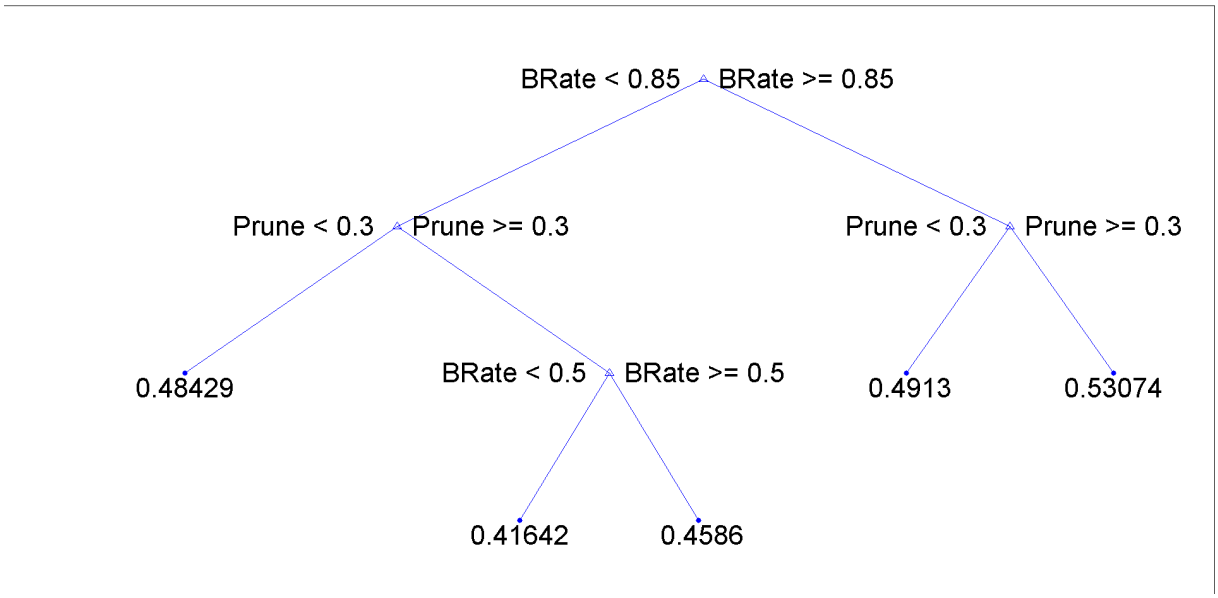


FIGURE 5.13 – Arbre de régression (CART) pour la précision d'apprentissage

5.3 Qualité des indicateurs de confiance

Nous étudions dans cette partie la qualité des indicateurs de confiance proposés en partie 3.3. Pour ce faire, nous appliquons la procédure d'apprentissage avec option de rejet tour à tour avec l'indicateur de similarité et celui de stabilité, avec différents seuils de rejet $0 = s_1 < s_2 < \ldots < s_n = 1$. Pour rappel, la règle décision consiste à suivre la prédiction initiale si l'indicateur de confiance dépasse le seuil, et de s'abstenir sinon. On suppose pour cette comparaison que l'abstention n'a pas de coût. Étant donné un indicateur g choisi, la performance de l'indicateur au seuil s_i est la différence entre la précision de la prédiction sachant que le niveau de confiance est suffisant et la précision d'origine sans indicateur.

Les résultats sont calculés à partir des backtests d'environ 40 indices d'actions de 2008 à 2014.

La figure 5.14 montre cette amélioration de la performance pour chaque indicateur. La stabilité apparaît clairement comme le meilleur indicateur, étant croissant en fonction du seuil, tandis que l'indicateur de similarité induit une amélioration plus faible et non monotone.

De plus, il convient de vérifier le taux d'acceptation (ou taux de détection) des indicateurs, c'est-à-dire la proportion de prédictions qui ont un niveau de confiance suffisant. Les deux indicateurs n'ont pas la même forme (figure 5.15). L'acceptation de la stabilité est concave et est au minimum 0.2, ce qui veut dire que même au seuil le plus exigeant de confiance, il reste 20% de prédictions à suivre. En revanche, la courbe d'acceptation de la similarité est convexe et décroît rapidement.

Enfin, les indicateurs de confiance sont des scores de classification permettant de reconnaître une "bonne prédiction" (confiance au-dessus du seuil) des "mauvaises prédictions" (confiance insuffisante). Une mesure courante dans ce cas est la courbe ROC (*Receiver Operating Characteristic* [GS66]), qui est graphiquement la proportion de vrais positifs en fonction de la proportion de faux positifs (figure 5.16). Encore une fois, la stabilité apparaît comme un meilleur score, avec une aire sous la courbe (AUC) supérieure.

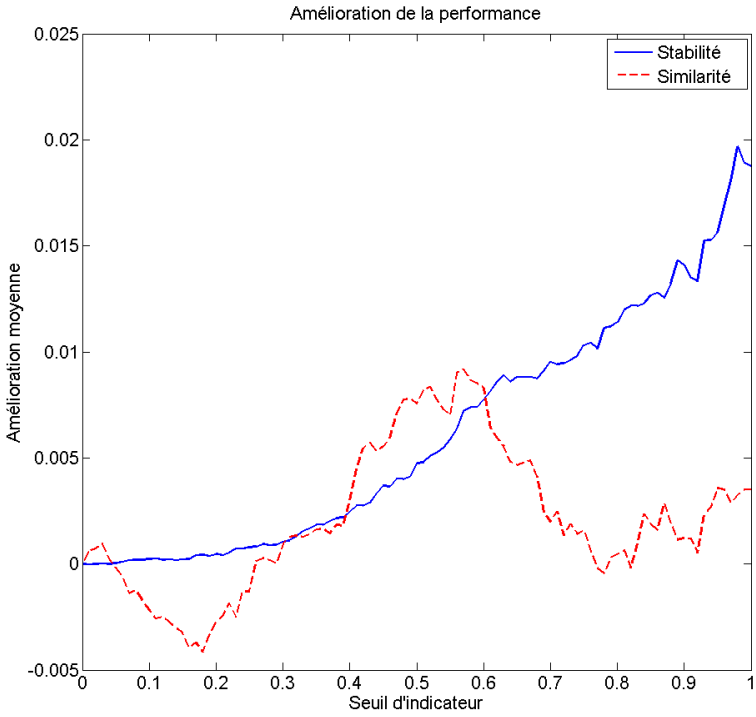


FIGURE 5.14 – Amélioration de la performance par les indicateurs en fonction du seuil de confiance

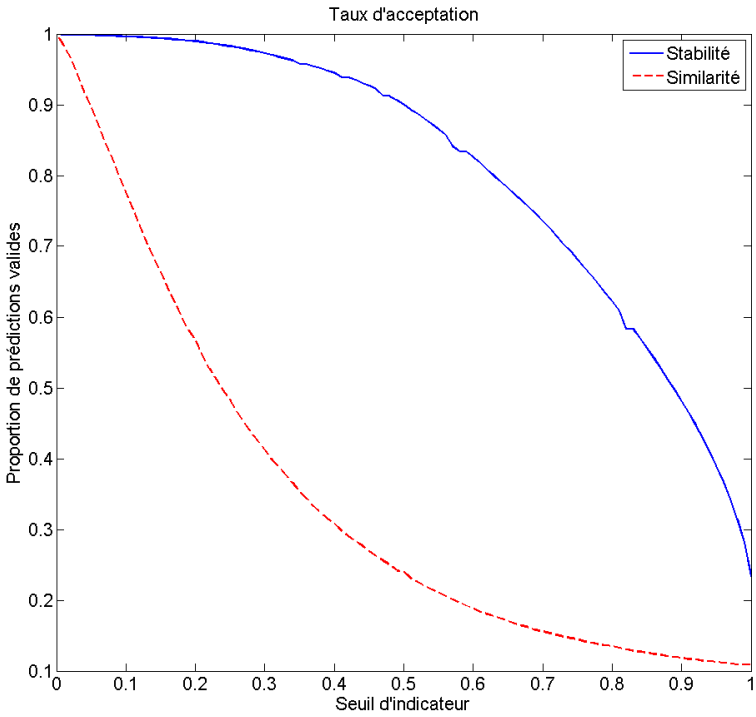


FIGURE 5.15 – Taux d'acceptation en fonction du seuil de confiance

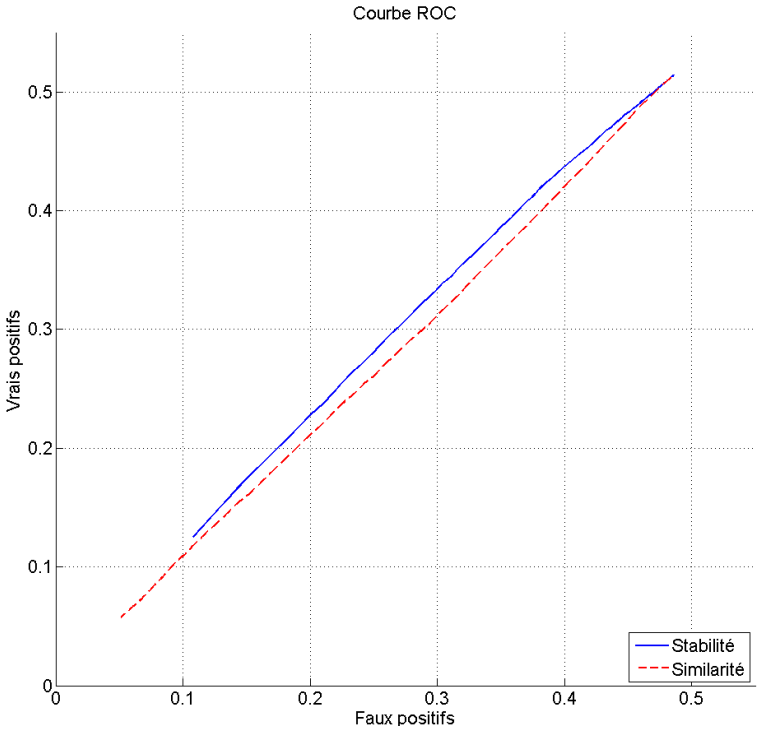


FIGURE 5.16 – Courbe ROC des indicateurs de similarité et de stabilité

Chapitre 6

Recommandation financière : solution PRISMS

Les chapitres précédents ont détaillé les différentes étapes formant le processus qui permet de produire et tester des modèles prédictifs pour le sens de variation de tout actif cible à partir d'un ensemble d'actifs prédicteurs. L'auto-suffisance de cette démarche complète a permis de mettre en œuvre toutes les méthodes de manière opérationnelle au sein de la Recherche Quantitative d'Exane BNP Paribas. L'application des recherches est complète, dans le sens où les applications s'étendent d'une librairie dédiée de fonctions MATLAB aux présentations commerciales des interprétations économiques du modèle.

L'application de haut niveau se concrétise en particulier sous la forme d'un "produit" de recommandation financière, nommé *Portfolio Risk Identification and Selection Methods based on Statistical learning* (PRISMS). Il s'agit à la fois de l'ensemble des outils produisant les prédictions et de la publication périodique de recommandations aux investisseurs sur certains actifs. Nous verrons dans ce chapitre le passage des méthodes décrites précédemment à un processus concret de recommandation financière, les problématiques pratiques et les solutions employées. Un exemple de recommandation publiée périodiquement se trouve en annexe B.

6.1 Interface graphique

Une première manière de mettre en application les outils et algorithmes aux utilisateurs qui y sont initiés est d'y donner accès via une interface (figure 6.1). Cela n'est possible que parce que nous avons développé un ensemble cohérent et auto-suffisant de méthodes de traitement des données pour la prédiction. L'organisation des modules de l'interface suit exactement la même logique que le processus de prédiction : gestion des données, représentation, apprentissage, prédiction, backtesting et calibration. Chaque module restitue des données sous forme visuelle, avec une interprétation directe des résultats.

Les défis principaux sont l'autonomie et la visualisation. L'utilisateur doit pouvoir effectuer toutes les opérations de calcul et de maintenance depuis l'interface, sans toucher au bas niveau des outils. Représenter les données pertinentes de façon compréhensible devient rapidement une tâche compliquée, car le nombre de dimensions des données dépasse facilement 2. Typiquement, une série de backtests avec plusieurs jeux de paramètres sur un ensemble d'actifs est une base de données de dimension 5.

FIGURE 6.1 – Aperçu de l'interface graphique de PRISMS

6.2 Recommandations financières

Le processus de recommandation financière consiste à transformer pas-à-pas les résultats des algorithmes, ou des méthodes issues d'une expertise particulière, en un message clair de recommandation d'achat ou de vente motivée par une interprétation des résultats en termes économiques.

La recommandation répond aux questions suivantes :

1. Que faut-il faire ? Acheter ? Vendre ?
2. Pourquoi devrait-on le faire ?
3. Comment la recommandation se situe-t-elle par rapport au contexte actuel ?
4. Quelle qualité peut-on attendre des recommandations ?

La première question définit la recommandation : il s'agit avant tout de fixer les actifs ciblés et l'objectif de prédiction. Suivre les prédictions des forêts aléatoires avec option de rejet revient à fixer un ensemble d'actifs d'intérêt, un horizon de prédiction, et d'émettre une recommandation seulement sur ceux dont la prédiction a une confiance suffisante. La question "pourquoi" est particulièrement difficile dans le cas d'algorithmes d'apprentissage, car il est nécessaire de construire une explication économique. Elle peut s'effectuer par le lien au contexte : le poids des

variables fournit (parfois) une interprétation directe. L'identification des variables les plus importantes et de leur signification permet de donner un sens à la structure du modèle. De plus, on associe aux recommandations des indicateurs de confiance dont les variations s'expliquent par le contexte. L'interprétation économique est essentielle pour convaincre l'investisseur de la validité de la recommandation. La qualité potentielle des recommandations peut se montrer grâce à un *track-record*, soit un suivi des performances réalisées, soit un backtest valide.

Une fois émises, les recommandations sont suivies en temps réel de façon indépendante du processus d'apprentissage, et une base de données statistiques sur la performance de chaque recommandation est maintenue. Cette base vise à reproduire les performances réelles d'un investisseur ayant suivi la recommandation. Elle est à distinguer des historiques de backtests car elle a une plus grande valeur en tant que résultat de test pur. Les performances réalisées sont publiées à chaque nouvelle recommandation.

En parallèle, une base de backtests est constituée et mise à jour périodiquement (au minimum avant chaque publication) afin de justifier la validité historique de la méthode et de suivre les performances pour différents jeux de paramètres. On choisit à chaque prédiction le modèle ayant le meilleur jeu de paramètres d'après la base de tests.

En résumé, l'émission d'une recommandation s'effectue par ces étapes :

1. Évaluer la performance des recommandations précédentes.
2. Rendre compte du track-record effectif.
3. Mettre à jour la base de backtests.
4. Construire le modèle prédictif par le jeu de paramètres optimal.
5. Produire les prédictions sur les actifs fixés au départ, les indicateurs de confiance, les poids des variables et leur évolution dans le temps.
6. Lier les indicateurs et les variables d'intérêt au contexte économique.
7. Donner un sens économique aux prédictions.
8. Publier les recommandations d'achat ou de vente, avec les arguments précédents.

6.3 Défis

Les défis posés par ce processus sont de deux types. Tout d'abord, l'interprétation économique du modèle prédictif, bien que nécessaire, n'est pas directe. Les modèles d'apprentissage que nous utilisons sont conçus pour optimiser la performance et non pour donner une explication de l'espace d'entrée. Cette difficulté a poussé notamment à développer des indicateurs permettant de lier le comportement des prédictions au contexte, en mesurant par exemple la similarité des données ou la différence de rang entre des variables prédictives de natures différentes.

L'autre difficulté est la contrainte de temps : le délai entre l'information (prix de clôture) et la recommandation doit être le plus court possible pour ne pas perdre de performance. Or le nombre d'actifs suivis en même temps ne permet pas toujours de mettre l'information à jour sous forme de backtests et modèles en temps raisonnable. Typiquement, la mise à jour des tests de 200 actions peut nécessiter une quinzaine d'heures de traitement, ce qui veut dire que la recommandation est systématiquement retardée d'un jour par rapport à la date d'entraînement, à moins de faire les calculs le week-end. Ce retard se répercute comme une baisse de précision. Ces contraintes ont donné lieu à une adaptation de la segmentation des séries et du backtesting

afin de les rendre incrémentaux. Les résultats de l'un et l'autre sont systématiquement enregistrés dans une base, et chaque incrément n'effectue les calculs que sur le nombre minimum de fenêtres de données pour mettre la base à jour. L'efficacité passe aussi en grande partie par la parallélisation des backtests et par la souplesse des processus. Les résultats sont enregistrés au fur et à mesure et la reprise est transparente en cas d'interruption.

À la fin, le processus est entièrement automatisé de la mise à jour des données de marché au tableau de prédictions, en effectuant les opérations suivantes :

1. Mettre à jour les données de marché.
2. Segmenter les nouvelles fenêtres de données et mettre à jour la base de partitions.
3. Réaliser les backtests (avec plusieurs jeux de paramètres) sur les dates non-couvertes par la base de calibration existante.
4. Ajouter les nouveaux backtests à la base de calibration.
5. Produire des données de résumé des backtests pour retenir les informations essentielles.
6. Choisir le meilleur jeu de paramètres.
7. Construire les modèles prédictifs avec les paramètres optimaux.
8. Calculer les prédictions, les rangs des variables et les indicateurs de confiance et les restituer dans un format utilisable directement.

6.4 Performance opérationnelle

La publication des recommandations de PRISMS a officiellement commencé en janvier 2012. Elle concerne la prédiction du marché (STOXX Europe 600) et des 19 indices sectoriels au niveau Supersecteurs (selon la classification ICB [1]), en prix relatif par rapport au marché (signe de l'excès de rendement). Les recommandations sont publiées mensuellement, avec un horizon d'un mois. Les performances sont dans l'ensemble positives :

— marché : 60% de précision avec un rendement moyen de 11.6% par an,
— indices sectoriels relatifs au marché : 55% de précision avec un rendement moyen de 1.6% par an.

Le produit a connu des évolutions et des versions supplémentaires, comme la prédiction sur des actions avec un test de performance par constitution de portefeuille.

1. http://www.icbenchmark.com

Conclusion et perspectives

Nous avons proposé et mis en œuvre un ensemble de méthodes formant un processus complet pour la prédiction du signe des rendements d'actifs financiers. Les méthodes de chaque étape traitent de nombreuses problématiques posées principalement par la nature des données, qui est de prime abord très éloignée du cadre classique de l'apprentissage statistique. Néanmoins, même sous ces conditions, les procédures construisent étape par étape une solution performante. L'aboutissement à une application concrète officielle est une preuve supplémentaire de pertinence, même si l'interprétation économique des modèles peuvent parfois sembler capillotractées.

Les méthodes scientifiques ont aussi bénéficié, dans l'autre sens, des problématiques soulevées par les opérationnels. C'est le cas du protocole de backtesting, qui est naturel en finance, et encore plus de la prédiction avec option de rejet, qui est l'une des premières préoccupations énoncées après l'obtention du modèle de prédiction.

Les résultats expérimentaux peuvent sembler faibles, avec 55% de précision, comparé aux résultats de problèmes de classification plus courants. Néanmoins, le fait que les performances sont similaires dans l'application concrète en temps réel montre que le protocole de test est juste et estime fidèlement l'erreur de généralisation. Il faut aussi reconnaître que l'hypothèse d'absence d'opportunité d'arbitrage des marchés financiers n'est pas une simple commodité théorique, mais correspond à la réalité dans une certaine mesure. Identifier des arbitrages n'est pas supposé être une tâche facile, et une précision de 55% est suffisante en finance quantitative, **si** elle est **stable et contrôlée**.

L'implémentation directe des recommandations en tant que portefeuille semble cependant plus difficile, pour plusieurs raisons. Tout d'abord, la prédiction porte sur le signe du rendement et ne dit rien sur la valeur du rendement. Or les risques sont typiquement asymétriques : les baisses inattendues sont de plus grande amplitude que les hausses non désirées. Il est possible de discrétiser le rendement en plusieurs classes, mais il s'avère que cela morcèle considérablement les données et augmente la dimension du problème. Le deuxième argument est la présence de coûts de transaction. La détection de phénomènes transitoires comme les opportunités d'arbitrage doit permettre un rebalancement fréquent du portefeuille, mais le gain est alors réduit par les coûts. Enfin, la performance d'un portefeuille provient seulement en partie des estimations de départ, car la réalité n'est pas un problème à période unique. La stratégie de gestion a donc une grande influence sur la performance, et les signaux d'achat/vente prédits nécessitent une stratégie adéquate pour les utiliser.

Toutefois, on peut considérer qu'il y a une certaine marge de manœuvre dans les paramètres à ajuster. Les résultats montrés ne sont pas issus d'une validation poussée des paramètres, et le choix des prédicteurs est une source importante d'amélioration. Leur optimisation dynamique peut bénéficier de procédures adaptées de sélection de variables.

Enfin, les avantages de l'apprentissage multi-tâche ne sont pas à négliger. L'approche que nous avons proposée n'a pas certes été intégrée et testée dans le processus de prédiction, mais

la vision multi-tâche de la prédiction des rendements financiers est certainement pertinente et à développer. Il s'agit de trouver une fonction d'erreur adéquate pour le problème de classification et qui admette une méthode de résolution (typiquement, un théorème de réprésentation). Une autre façon d'appliquer le multi-tâche à notre méthode est de concevoir une extension multi-tâche des arbres ou de Random Forests.

Annexe A
Learning the Graph of Relations Among Multiple Tasks

L'article qui suit a été publié au workshop "New Learning Frameworks and Models for Big Data" de la conférence ICML 2014. Ces travaux ont donné lieu à des présentations dans plusieurs conférences :

- 7th International Conference on Computational and Financial Econometrics (CFE 2013 : http://cfenetwork.org/CFE2013/) en décembre 2013 à Londres,
- ICML 2014 workshop on New Learning Frameworks and Models for Big Data (http://ama.liglab.fr/nlfmbd_icml14/) en juin 2014 à Beijing,
- Conférence d'Apprentissage Automatique (CAp'2014 : http://cap2014.sciencesconf.org/) en juillet 2014 à Saint-Étienne.

Learning the Graph of Relations Among Multiple Tasks

Andréas Argyriou ANDREAS.ARGYRIOU@ECP.FR
Center for Learning and Visual Computing, Ecole Centrale Paris, Châtenay-Malabry, France

Stéphan Clémençon STEPHAN.CLEMENCON@TELECOM-PARISTECH.FR
Department of Signal and Image Processing (TSI), Telecom ParisTech, Paris, France

Ruocong Zhang RUOCONG.ZHANG@TELECOM-PARISTECH.FR
Department of Signal and Image Processing (TSI), Telecom ParisTech, Paris, France

Abstract

We propose *multitask Laplacian learning*, a new method for jointly learning clusters of closely related tasks. Unlike standard multitask methodologies, the graph of relations among the tasks is not assumed to be known a priori, but is learned by the multitask Laplacian algorithm. The algorithm builds on kernel based methods and exploits an optimization approach for learning a continuously parameterized kernel. It involves solving a semidefinite program of a particular type, for which we develop an algorithm based on Douglas-Rachford splitting methods. Multitask Laplacian learning can find application in many cases in which tasks are related with each other to *varying degrees*, some strongly, others weakly. Our experiments highlight such cases in which multitask Laplacian learning outperforms independent learning of tasks and state of the art multitask learning methods. In addition, they demonstrate that our algorithm partitions the tasks into clusters each of which contains well correlated tasks. Our method may be suitable for large scale and big data approaches, as it divides the initial complexity into sub-problems that depend on different dimensions.

1. Introduction

In recent years, *multitask learning* has been an active and growing area of interest in machine learning. The goal in such problems is to jointly learn several regression or classification tasks and in this way enhance statistical performance, compared to learning the tasks independently. The advantages of multitask learning are especially pronounced in situations lacking in sufficient samples per task. By "borrowing strength" from other tasks, it may be possible to learn better models for each task, provided that there are sufficiently strong *relations among the tasks*.

There has been a wide variety of multitask learning approaches mainly due to the large range of possible ways in which tasks may be related. On the other side, a more generic and broadly applicable learning theoretic treatment was developed early on (Baxter, 2000; Ben-David & Schuller, 2003; Maurer, 2006). Consider an input set $\mathcal{X}$ and an output set $\mathcal{Y}$ and for simplicity that $\mathcal{X} \subseteq \mathbb{R}^d$, $\mathcal{Y} \subseteq \mathbb{R}$. Tasks can be viewed as n functions f_ℓ, $\ell = 1, \ldots, n$, to be learned from given data $\{(x_{\ell i}, y_{\ell i}) : i = 1, \ldots, m_l, \ell = 1, \ldots, n\} \subseteq \mathcal{X} \times \mathcal{Y}$. These tasks may be viewed as drawn from an unknown joint distribution of tasks, which is the source of the bias that relates the tasks.

As with many machine learning problems, regularization based approaches have been applied to multitask learning as well. In particular, each task may be represented as a linear predictive function $x \mapsto w_\ell^\top x$, or equivalently as a vector w_ℓ of regression parameters.[1] Thus in a regularization based setting the learning algorithm may be phrased as an optimization problem of the form

$$\min \left\{ \sum_{\ell=1}^{n} \sum_{j=1}^{m} E(w_\ell^\top x_{j\ell}, y_{j\ell}) + \gamma\, \Omega(W) : W \in \mathbf{M}_{d,n} \right\},$$

where $\mathbb{E} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a convex loss function, Ω is a penalty term, $\mathbf{M}_{d,n}$ denotes the set of real $d \times n$ matrices and W is the matrix with columns w_ℓ. The error term should favor a good fit of the data whereas the regularizer Ω should favor certain types of relations among the

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

[1] Some, but not all, multitask methods can be kernelized – see (Argyriou et al., 2009) for conditions.

tasks. The positive regularization parameter γ (which can be tuned with techniques like cross validation) determines the trade off between fitting the data well and enforcing the bias of task relatedness. Clearly, different choices of the penalty Ω may give rise to quite different multitask methods.

One popular approach has been to use ℓ_2 type penalties on differences or various combinations of tasks, as studied in detail in (Evgeniou et al., 2005; Jacob et al., 2009). In such approaches, it is common to penalize distances between tasks and hence tasks are biased towards being similar to each other. Other approaches proposed in the past have relied on boosting techniques (Chapelle et al., 2011), neural networks (Caruana, 1997), hierarchical Bayesian methods (Bakker & Heskes, 2003) etc. Another class of methods aims to learn tasks (that is, the columns of W) which lie in low-dimensional subspaces or manifolds embedded in $\mathbb{R}^d$ (Agarwal et al., 2010; Ando & Zhang, 2005; Argyriou et al., 2008a; Srebro et al., 2005). Many of these methods use regularizers which involve the *trace norm*, which is defined as the sum of the singular values of W.

Most of the above methods rely on implicit assumptions about *all* tasks being related in a specific way. However, in many applications it is not known a priori whether all tasks are strongly related with each other. Or, in other applications it is reasonable to expect that tasks relate to each other in *varying degrees* without knowing much about the strengths of these relations. In particular, it is common that tasks *cluster in a few groups*, with weak task relations across groups, but strong task relations within each group. Such characteristics are particularly relevant in big data problems, in which one would focus resources on separate small groups of tasks, rather than processing all the tasks equally. The main complication with such situations is that the appropriate clustering may not be known a priori. In addition, it may be important to account both for the strong intra-cluster relations and the weak inter-cluster ones. In general, using a method such as k-means or spectral clustering for preprocessing the tasks is not satisfactory, since usually there is insufficient prior information to obtain good clusters of tasks.

Financial asset prediction is one example of an application in which the tasks are dependent with an unknown underlying structure. This structure may change over time, so that clustering large historical data would generally be unsuitable for prediction. Learning in a multitask framework would enable more consistent predictions, but also a better risk assessment of the predicted assets. Another example is *recommender systems* and *preference learning*, in which the preferences of multiple consumers are to be learned. It is known from marketing research that consumers cluster into groups of similar behavior (based on demographics, geography etc.) whereas they share weaker preference behavior across groups.

Until now, there has been limited work, in the context of regularization, on learning of tasks that follow clustered distributions. For example, in (Jacob et al., 2009), a convex relaxation of a task clustering problem within the ℓ_2 regularization framework has been proposed. In (Argyriou et al., 2008b; Kang et al., 2011), a clustering variation on the low rank approach is used to learn multiple subspaces on which the tasks lie. Also in (Bakker & Heskes, 2003), the tasks clustering problem has been addressed in a hierarchical Bayesian framework with mixtures of distributions. In (Kumar & Daumé III, 2012), a matrix factorization approach that penalizes matrix factors of W with ℓ_1 and ℓ_2 norms has been proposed.

In this paper, we propose a new method for simultaneously learning the tasks and the relations among them. We start from a Laplacian regularization framework by (Evgeniou et al., 2005), demonstrate its drawbacks and show how it can be suitably modified for our purposes. We then formulate our learning method as an optimization problem in a reproducing kernel Hilbert space, in which the kernel also needs to be learned. We thus derive a problem analogous to that of learning from an infinite set of kernels, with the difference that in our case the feasibility set is not convex. Despite this, we show how an alternating minimization algorithm can be used to compute good estimates of the tasks and of the graph of task relations. In addition, we propose an algorithm based on Douglas-Rachford optimization methods for solving certain semidefinite programs, which occur as subproblems. Finally we report experiments which highlight that a) our method is competitive and often outperforms state of the art multitask methods; b) our method recovers a good clustering of the tasks and their relations.

2. Learning the Tasks' Graph Laplacian

In our multitask learning framework, we account for the dependence structure between tasks representing them in a graph. Including the graph Laplacian as the relevant information in the optimization problem has been studied when the graph is known. We propose a joint formulation for learning both the tasks and the graph, as well as an alternating algorithm to solve the optimization problem.

2.1. Background

A general framework for multitask learning has been proposed in (Evgeniou et al., 2005), based on regularization in reproducing kernel Hilbert spaces (RKHS). This framework consists of regularization problems in the joint space of tasks, with a positive definite quadratic penalty $w^\top E w$,

where w is the column-wise vectorization of W and E is a $dn \times dn$ positive definite matrix. The intuition for E is that it describes relations between pairs of tasks. The authors of (Evgeniou et al., 2005) show that such multitask learning formulations can be rephrased as regularization problems in an RKHS of functions on the augmented input-task space $\mathcal{X} \times \mathbb{N}_n$, where $\mathbb{N}_n$ denotes the set $\{1, \ldots, n\}$.

For example, the case $E = \Delta \otimes I_d$, where Δ is an $n \times n$ positive diagonal matrix and $\otimes$ denotes the Kronecker product, expresses lack of any relations among the tasks and corresponds to learning the tasks *independently*, with the diagonal entries of D as regularization parameters. As another example, the main methodology studied in (Evgeniou et al., 2005) penalizes the squared ℓ_2 distances of the tasks $\{w_\ell, \ell \in \mathbb{N}_n\}$ *from their average*,

$$\sum_{\ell=1}^{n} \|w_\ell\|^2 + \rho \sum_{\ell=1}^{n} \|w_\ell - \frac{1}{n} \sum_{q=1}^{n} w_q\|^2 \,. \qquad (1)$$

This formulation corresponds to an assumption natural in many applications, namely, that all tasks are close to each other in ℓ_2 distance.

However, in many cases in which tasks cluster in two or more groups (like the examples of Section 1), regularizers like (1) are not appropriate. The reason is that tasks *across* different groups are *weakly* related, whereas the above type of penalty biases them towards being strongly related. This has motivated the authors of (Evgeniou et al., 2005) to define the *graph of tasks* as a weight matrix A with nonnegative entries which encodes the relatedness among the n tasks. Consequently, the *tasks Laplacian* L is the graph Laplacian matrix obtained from A,

$$L = D - A \,,$$

where $D = \mathrm{Diag}(d)$ is the diagonal matrix formed by the degrees of the vertices $d_i = \sum_{j=1}^{n} A_{ij}$. Thus in (Evgeniou et al., 2005) a multitask methodology involving the tasks Laplacian is proposed (but not implemented or further studied). This methodology involves penalizing the tasks with the Laplacian quadratic form as follows

$$\min \Bigg\{ \sum_{\ell=1}^{n} \sum_{j=1}^{m_\ell} E(w_\ell^\top x_{j\ell}, y_{j\ell}) + \gamma \sum_{\ell,q=1}^{n} w_\ell^\top w_q L_{\ell q} : w_\ell \in \mathbb{R}^d \; \forall \ell \in \mathbb{N}_n \Bigg\} \,. \qquad (2)$$

One rationale behind this regularization is that this penalty equals $\sum_{\ell,q=1}^{n} \|w_\ell - w_q\|^2 A_{\ell q}$ and hence it favors those pairs of tasks (w_ℓ, w_q) with large weights $A_{\ell q}$ to be similar.

The above penalty also equals $w^\top (L \otimes I_d) w$, where $w = (w_1^\top \ldots w_n^\top)^\top$ is the columnwise vectorization of W and I_d the $d \times d$ identity matrix. This quadratic form is positive semidefinite but not positive definite, so it is not directly equivalent to regularization in an RKHS. To address this issue, (Evgeniou et al., 2005) suggests restricting w on the range of $L \otimes I_d$, which yields an equivalent formulation in an RKHS with the multitask kernel $K((x,\ell),(t,q)) = L_{lq}^{+}\, x^\top t$, for every $x, t \in \mathbb{R}^d$, $\ell, q \in \mathbb{N}_n$.

We claim, however, that in multitask practice it will be necessary to optimize W over the entire space. The reason is that the above restriction from (Evgeniou et al., 2005) discards crucial information about task relations which is contained in the null space of the Laplacian. To see this, consider, as a simple example, a graph of two tasks with a positive weight, which corresponds to the penalty $\|w_1 - w_2\|^2$. The range constraint from (Evgeniou et al., 2005) enforces $w_1 + w_2 = 0$, since the constant vector belongs to the null space of L for any Laplacian. At the same time, this constraint contradicts the information, encoded by a large weight between the two tasks, that the distance between w_1 and w_2 is small. More generally, any graph with k connected components has k zero eigenvalues and hence, there will be k linearly independent constraints on the w_ℓ. If tasks within each cluster were close to each other (which is our objective) then the constraints would imply that all tasks were close to zero. In practice, imposing the k constraints of (Evgeniou et al., 2005) restricts the tasks to lie on an $n - k$ dimensional subspace. This subspace depends solely on the graph topology and may be completely inconsistent to the actual tasks that have generated the data. Hence these constraints may prevent the method from learning the correct tasks, as we further demonstrate with realistic examples in Section 3.

The problem of learning the task clusters and the tasks simultaneously has also been the topic of (Jacob et al., 2009). These authors consider penalties which are combinations of three terms: an ℓ_2 penalty on the *average* of the tasks, a measure of *within-cluster* variance similar to (1) and a measure of *between-cluster* variance which uses the cluster means. The regularization problem is a function of the matrix of ones and zeros encoding the clusters and is a nonconvex optimization problem. To deal with this in a tractable way, the authors propose a convex relaxation of the penalty, which leads to the alternative regularization

$$\begin{aligned} &\mathrm{tr}(\Pi W \Sigma_c^{-1} W^\top \Pi) \\ &\text{subject to } \Sigma_c \succeq 0, \alpha I_n \preceq \Sigma \preceq \beta I_n, \mathrm{tr}\,\Sigma = \gamma, \end{aligned} \qquad (3)$$

where tr is the matrix trace, Σ_c and Σ relate in an affine way, Π is a fixed projection matrix and α, β, γ are positive constants depending on the regularization parameters.

A very different approach (Kumar & Daumé III, 2012) has been inspired by the trace norm and its formula in terms of matrix factorization. The idea is that one of the matrix fac-

tors of W can be viewed as encoding the task grouping and is penalized with an ℓ_1 sparsity penalty. At the same time, this method will favor low rank solutions since it penalizes the factors of W. Another approach extending trace norm regularization (Argyriou et al., 2008b; Kang et al., 2011) assumes that tasks lie not on one but on multiple low dimensional subspaces and simultaneously learns the tasks and these subspaces.

2.2. Learning the Tasks Given the Graph

The question we will address is how to learn simultaneously the n tasks and the graph of tasks via its Laplacian. We will build on the proposal (2) of (Evgeniou et al., 2005), but will follow a different path towards obtaing an RKHS formulation. We propose a small perturbation of the Laplacian penalty with a fixed constant ε, namely,

$$\min\Bigg\{\sum_{\ell=1}^{n}\sum_{j=1}^{m_\ell} E(w_\ell^\top x_{j\ell}, y_{j\ell}) + \gamma\left(\sum_{\ell,q=1}^{n} w_\ell^\top w_q L_{\ell q} + \varepsilon\sum_{\ell=1}^{n}\|w_\ell\|^2\right) : \; w_\ell \in \mathbb{R}^d \; \forall \ell \in \mathbb{N}_n\Bigg\}. \quad (4)$$

This penalty equals the sum of $\sum_{\ell,q=1}^{n}\|w_\ell - w_q\|^2 A_{\ell q}$ plus the perturbation term on the tasks. When ε is small, the graph term dominates the penalty and hence one should expect strong tasks similarities to conform to large weights and the opposite. Thus, formulation (4) reflects the intuition about the tasks Laplacian and at the same time has a convenient positive definite form. Throughout the paper ε will be set to a fixed but very small value.

The above penalty can also be written as $w^\top((L+\varepsilon I_n)\otimes I_d)w$ or $\mathrm{tr}((L+\varepsilon I_n)W^\top W)$. Thus, the corresponding multitask kernel equals

$$K((x,\ell),(t,q)) = (L+\varepsilon I_n)^{-1}_{lq}\, x^\top t\ ,$$

for every $x, t \in \mathbb{R}^d$, $\ell, q \in \mathbb{N}_n$. Instead of the linear kernel, any scalar reproducing kernel G may be used for the inputs (Caponnetto et al., 2008) :

$$K((x,\ell),(t,q)) = (L+\varepsilon I_n)^{-1}_{lq}\, G(x,t)\ , \quad (5)$$

where x, t belong to a generic input space $\mathcal{X}$. Let us call $\mathcal{H}_K$ the RKHS associated with this kernel.

It is also clear from the form $\mathrm{tr}((L+\varepsilon I_n)W^\top W)$ that a block diagonal (after simultaneous permutation of its rows and columns) Laplacian would penalize correlations of tasks within each group while ignoring any correlations across different groups. This intuition extends also to weak inter-cluster blocks of the Laplacian. Therefore, if a known or learned tasks Laplacian is given then any clustering method can be used to yield the clustering of the tasks.

Given training data $x_{j\ell} \in \mathcal{X}$, $y_{j\ell} \in \mathcal{Y}$, for $\ell \in \mathbb{N}_n$, $j \in \mathbb{N}_{m_\ell}$, training can be performed by solving the regularization problem

$$\min_{f\in\mathcal{H}_K}\left\{\sum_{\ell=1}^{n}\sum_{j=1}^{m_\ell} E(f(x_{j\ell},\ell), y_{j\ell}) + \gamma\,\|f\|_{\mathcal{H}_K}\right\} \quad (6)$$

where $E : \mathbb{R}\times\mathcal{Y}\to\mathbb{R}$ is a prescribed convex loss function and $\gamma > 0$ is a regularization parameter to be tuned, for example, with cross validation.

Problem (6) has a unique solution, which, by the representer theorem (Kimeldorf & Wahba, 1970), can be written in the form

$$f(t,q) = \sum_{\ell=1}^{n}\sum_{j=1}^{m_\ell} c_{j\ell} K((x_{j\ell},\ell),(t,q)) \quad (7)$$

$\forall\, t \in \mathcal{X}, q \in \mathbb{N}_n$ and for some $c \in \mathbb{R}^M$, where $M = \sum_{\ell=1}^{n} m_\ell$. Substituting this formula in (6) yields a convex optimization problem in M variables

$$\min\left\{E_y(K_{\mathbf{x}}c) + \gamma\, c^\top K_{\mathbf{x}} c \; : \; c\in\mathbb{R}^M\right\} \quad (8)$$

where $K_{\mathbf{x}}$ denotes the kernel matrix of all the input-task training data, and $E_y : \mathbb{R}^M \to \mathbb{R}$ is the convex function (parameterized by the output data) defined as $E_y(z) = \sum_{\ell=1}^{n}\sum_{j=1}^{m_\ell} E(z_{j\ell}, y_{j\ell}), \forall\, z \in \mathbb{R}^M$. If the task relatedness graph (and hence $K_{\mathbf{x}}$) are given, then solving for c in (8) (with a method such as SVM, ridge regression etc. in the dual) and using formula (7) yields the functions for each of the n tasks.

Note that the predictive function for the q-th task depends only on the q-th column of $(L+\varepsilon I_n)^{-1}$ and not on the other columns. In fact, the entries of this column weight the contribution of the other tasks in the predictive function. Also note that, if the input samples are the same for all tasks (which is the case for many applications), the kernel matrix can be expressed as

$$K_{\mathbf{x}} = (L+\varepsilon I_n)^{-1}\otimes G_{\mathbf{x}},$$

where $G_{\mathbf{x}}$ is the kernel matrix for the input kernel G on the data. In general (when input samples may differ across tasks), every (ℓ, q) block of $K_{\mathbf{x}}$ equals the product of $(L+\varepsilon I_n)^{-1}_{lq}$ with the (ℓ, q) block of $G_{\mathbf{x}}$. In the following, we introduce the matrix

$$Z = (L+\varepsilon I_n)^{-1}.$$

We also use $T_{\mathbf{x}}$ to denote the linear mapping that maps an $n\times n$ matrix Z to the $M\times M$ matrix with blocks the products of $Z_{\ell q}$ with the (ℓ, q) blocks of $G_{\mathbf{x}}$.

2.3. Learning the Tasks and the Graph Jointly

Our aim is to learn the n tasks without knowing the task relatedness matrix L exactly. That is, we aim to learn the task functions and the graph Laplacian simultaneously. We argue that this can be done via the methodology of learning infinitely parameterized kernels as in (Argyriou et al., 2005). That is, instead of learning the graph Laplacian directly, we optimize the objective function (8) with the kernel K allowed to belong to a set $\mathcal{K}$,

$$\min\left\{E_y(K_{\mathbf{x}}c) + \gamma\, c^\top K_{\mathbf{x}} c \;:\; c \in \mathbb{R}^M, K_{\mathbf{x}} \in \mathcal{K}\right\} \;. \quad (9)$$

The objective in (9) is convex, since the functional $K \mapsto \min\left\{E_y(K_{\mathbf{x}}c) + \gamma\, c^\top K_{\mathbf{x}} c \;:\; c \in \mathbb{R}^M\right\}$ is convex (Argyriou et al., 2005, Lem. 2). The approach of jointly learning the function and the kernel has been extensively used and justified with learning-theoretic bounds – see, for example, (Lanckriet et al., 2004; Rakotomamonjy et al., 2008; Srebro & Ben-David, 2006; Ying & Campbell, 2009).

To ensure that K is indeed a valid kernel of the form (5) coming from a graph Laplacian, we choose

$$\mathcal{K} = \left\{ T_{\mathbf{x}}(Z) : 0 \prec Z \preceq \tfrac{1}{\varepsilon} I_n, (Z^{-1})_{\text{off}} \leq 0, Z\mathbf{1}_n = \tfrac{1}{\varepsilon}\mathbf{1}_n \right\}$$

where A_{off} denotes the off diagonal entries of a matrix A and $\mathbf{1}_n$ denotes the vector of n ones. However, this set $\mathcal{K}$ is not convex, due to the constraint on $(Z^{-1})_{\text{off}}$.

In addition, we wish to encourage *few clusters* in the tasks graph so that, whenever possible, simpler and more structured graphs are preferred. Since we wish to have few clusters, the regularization should favor Laplacians with few small eigenvalues, which in turn implies a low effective rank for Z. To achieve the above objective, an appropriate penalty is the trace norm $\|Z\|_* := \sum_{i=1}^n \lambda_i(Z)$, where $\lambda_i(Z)$ denotes the i-th eigenvalue of matrix Z. This norm is known to be the tightest convex surrogate to the rank (Fazel et al., 2001) and in this case equals the trace of Z.

Thus, given a fixed input kernel G, we solve the problem

$$\min\Big\{E_y(K_{\mathbf{x}}c) + \gamma\, c^\top K_{\mathbf{x}} c + \alpha \operatorname{tr} Z : c \in \mathbb{R}^M,$$
$$K_{\mathbf{x}} = T_{\mathbf{x}}(Z), 0 \prec Z \preceq \tfrac{1}{\varepsilon} I_n, (Z^{-1})_{\text{off}} \leq 0, Z\mathbf{1}_n = \tfrac{1}{\varepsilon}\mathbf{1}_n\Big\},$$

where α is a second regularization parameter.

Applying the constraints back to the primal problem (4) and introducing the variable

$$Q = L + \varepsilon I_n \;,$$

we obtain the optimization problem

$$\min\Bigg\{\sum_{\ell=1}^n \sum_{j=1}^{m_\ell} E(w_\ell^\top x_{j\ell}, y_{j\ell}) + \gamma \operatorname{tr}(WQW^\top) + \alpha \operatorname{tr}(Q^{-1})$$
$$: W \in \mathbf{M}_{d,n},\; Q \succeq \varepsilon I_n,\; Q_{\text{off}} \leq 0, Q\mathbf{1}_n = \varepsilon \mathbf{1}_n \Bigg\} \;. \quad (10)$$

The objective function of this problem is non-convex whereas the feasibility set is convex.

Algorithm 1 Learning the tasks and the tasks Laplacian jointly.

Initialization: Select $W^0 \in \mathbf{M}_{d,n}$
for $k = 1, 2, \ldots$ **do**
1. $W \leftarrow \operatorname{argmin}\Big\{\sum_{\ell=1}^n \sum_{j=1}^{m_\ell} E(w_\ell^\top x_{j\ell}, y_{j\ell}) + \gamma \operatorname{tr}(WQW^\top) : W \in \mathbf{M}_{d,n}\Big\}$
2. $Q \leftarrow \operatorname{argmin}\Big\{\operatorname{tr}(WQW^\top) + \alpha \operatorname{tr}(Q^{-1}) : Q \succeq \varepsilon I_n \;,\; Q_{\text{off}} \leq 0, Q\mathbf{1}_n = \varepsilon \mathbf{1}_n\Big\}$

end for

2.4. Optimization

Even though problem (10) is not convex jointly in W and Q, it is convex in one of the two matrix variables when the other remains fixed. Thus we may exploit this property to obtain an algorithm alternating between minimization of W and minimization of Q (Algorithm 1).

Step 1 requires solving a kernel based method such as support vector machines or kernel ridge regression in dn variables. We solved this step using standard methods in the augmented space. In cases in which the input sample was common to all tasks, we used a Lyapunov solver, which was more efficient. Step 2 is a semidefinite program, since a constraint of the form $\operatorname{tr}(Q^{-1}) \leq \beta$ can be rewritten as $\operatorname{tr} R \leq \beta, \left(\begin{smallmatrix} R & I_n \\ I_n & Q \end{smallmatrix}\right) \succeq 0$. The number of variables depends only on n and, up to about 100 tasks, this SDP can be solved using interior point methods and packages such as SDPT3 or Sedumi. However, interior point methods do not scale well to larger n and we had to develop a custom-made first-order method in order to handle larger numbers of tasks (Algorithm 2). Our method relies on positive semidefinite projections and hence on eigendecompositions (step 2). These can be computed exactly for a few thousand tasks and can be extended beyond that with low rank approximations combined with Lanczos or power methods.

Algorithm 2 is based on a parallel splitting algorithm of Douglas-Rachford type from (Bauschke & Combettes, 2011, Prop. 27.8). Douglas-Rachford algorithms require

Algorithm 2 Douglas-Rachford parallel splitting algorithm for SDP in step 2 of Algorithm 1.

Initialization: Select $X_1 = X_2 = X_3 \succeq \varepsilon I_n$
for $k = 1, 2, \ldots$ **do**
1. $P \leftarrow \frac{1}{3}(X_1 + X_2 + X_3)$
2. Compute eigendecomposition $X_1 = U \operatorname{Diag}(\lambda) U^\top$
3. $\mu_i \leftarrow \operatorname{argmin}\{\frac{1}{2}(\lambda_i - e)^2 + \alpha e^{-1} : e \geq \varepsilon\}\ \forall i \in \mathbb{N}_n$
4. $Y_1 \leftarrow U \operatorname{Diag}(\mu) U^\top$
5. $(Y_2)_{ij} \leftarrow \begin{cases} \min\{(X_2)_{ij}, 0\} & \text{if } i \neq j \\ (X_2)_{ij} & \text{if } i = j \end{cases} \quad \forall i, j \in \mathbb{N}_n$
6. $Y_3 \leftarrow B - BE - EB + (\frac{1}{n}\sum_{i,j=1}^n B_{ij} + \varepsilon)E$, where $B = X_3 - W^\top W$, $E = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$
7. $K \leftarrow \frac{1}{3}(Y_1 + Y_2 + Y_3)$
8. $X_1 \leftarrow X_1 + 2K - P - Y_1$
9. $X_2 \leftarrow X_2 + 2K - P - Y_2$
10. $X_3 = X_3 + 2K - P - Y_3$

end for
return $Q \leftarrow Y_3$

computation of *proximity operators*, which extend the concept of projections – see, for example, (Bauschke & Combettes, 2011). The proximity operator $\operatorname{prox}_f(x)$ of a lower semicontinuous, convex function $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ at $x \in \mathbb{R}^d$ is the unique minimizer

$$\operatorname{prox}_f(x) = \operatorname{argmin}\left\{\tfrac{1}{2}\|y - x\|^2 + f(y) : y \in \mathbb{R}^d\right\}.$$

The first proximity operation (steps 2-4 in Algorithm 2) is that of the function

$$f_1(Q) = \begin{cases} \alpha \operatorname{tr}(Q^{-1}) & \text{if } Q \succeq \varepsilon I_n \\ +\infty & \text{otherwise} \end{cases}.$$

Since f_1 is a *spectral* function (a function of only the eigenvalues of Q), the proximity operation reduces to the proximity operation on its spectrum, by von Neumann's trace inequality (Mirsky, 1975). This in turn separates into univariate optimization problems, which can be solved with cubic equations (Baldassarre et al., 2012, Lem. 4.1).

The second proximity operation (step 5) is the projection on the set $\{Q \in \mathbf{S}_n : Q_{\text{off}} \leq 0\}$, where $\mathbf{S}_n$ denotes the space of $n \times n$ symmetric matrices. Finally, the third proximity operation (step 6) is that of the function

$$f_2(Q) = \begin{cases} \operatorname{tr}(WQW^\top) & \text{if } Q \in \mathbf{S}_n \text{ and } Q\mathbf{1}_n = \varepsilon\mathbf{1}_n \\ +\infty & \text{otherwise} \end{cases}.$$

This proximity operator is easy to derive with Lagrange multipliers.

2.5. Complexity decoupling

Our approach results in transforming the initial problem to a formulation that is suitable for large scale methods. The initial problem is difficult because it simultaneously optimizes the tasks and their graph of relations. It is overall non-convex and its complexity depends on both feature dimensionality, training size, and number of tasks. The proposed method alleviates these difficulties by dividing into two convex problems that are sensitive to different dimensions. The complexity of the regression step depends on the training size and the feature dimensionality, while the graph Laplacian step is mainly sensitive to the number of tasks. By decoupling the complexities in such a way, the alternated optimization can make efficient use of big data methods. For example, one could first look for a fair approximation of the graph in a Map/Reduce fashion by distributing the problem on small parts of the data set. With this Laplacian approximation (which accounts for regression efficiency), one can then solve clusters of tasks separately, thus reducing the number of tasks in each subproblem.

This work firstly aimed at validating the efficiency and suitability of our formulation, so large scale implementations are not discussed in more details here. An efficient multitask formulation is critical for such applications. In the next section, we apply our approach to simulated and real data sets. Even though they are not from proper big data problems, the out-performance of our approach compared with state of the art methods validate the optimization problem formulation and the proposed algorithm.

3. Experiments

We applied our proposed multitask Laplacian learning method ("MT Laplacian") with squared loss on synthetic data, where the task clustering is known, and on a real dataset. We compared our method with two benchmarks and three state of the art approaches:

- The independent ridge regression, where the tasks are learnt separately, serving as a baseline.
- The "true Laplacian" case, where we directly use the expected graph Laplacian instead of learning it (applicable only to synthetic data). Considering that the clusters have the same dependence among their tasks, and that all the tasks are generated with the same distributions, the "ground truth" graph Laplacian of our approach is expected to be proportional to this.
- The multitask learning approach proposed in (Evgeniou et al., 2005) for a known graph, where we use the Laplacian found by our method ("kernel with graph Laplacian"). Recall that (Evgeniou et al., 2005) *do not learn* the Laplacian, but we report this result to demonstrate the drawbacks discussed in Section 2.1.
- Trace norm regularization, which is a state of the

art method for multi-task problems (Argyriou et al., 2008a; Srebro et al., 2005). We used the SLEP implementation from (Liu et al., 2009).

- The clustered multitask method ("Clustered MT")[2] formulated in (Jacob et al., 2009), discussed in section 2.1.

3.1. Simulated Data

We tested our method on a small artificial dataset with two clusters of two tasks, generated as in (Jacob et al., 2009), the true graph Laplacian being $\begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$. Each task t in the cluster c is a vector of regression coefficients in $\mathbb{R}^d$ ($d = 30$), generated as $w_t = w_c + \tilde{w}_t$, where w_c is the cluster center and $\tilde{w}_t$ a task-specific vector having exactly the same non-zero features as w_c, drawn from $\mathcal{N}(0, \sigma_c^2)$, with $\sigma_c^2 = 16$. The cluster centers w_c are orthogonal in the first $d-2$ dimensions: the first cluster has $(d-2)/2$ features drawn from $\mathcal{N}(0, \sigma_r^2)$ ($\sigma_r^2 = 900$), and the second cluster has the other $(d-2)/2$, generated in the same way. The remaining two features are drawn from $\mathcal{N}(0, \sigma_c^2)$ for each task.

The inputs X are vectors in $\mathbb{R}^d$, randomly generated from a uniform distribution on $[0, 1]$, and the corresponding output Y_t for task t is calculated as $Y_t = w_t^\top X + \epsilon_y$, ϵ_y being normal noise from $\mathcal{N}(0, \sigma_n^2)$ ($\sigma_n^2 = 150$). The tasks share the same input.

We first train each method on a training set of size m, then find the best parameters on a validation set of 500 points, and finally test the optimal parameters on a test set of 2000 samples. The average test RMSE across all tasks - since the task outputs have the same amplitude - and all test sets is the global error measure for each method. Results for this experiment are shown in Figure 1, on average for 50 such data sets.

Apart from the smallest training size (40 points), the best RMSE is achieved by the "True Laplacian" benchmark. When the true graph structure is known, the Laplacian-penalized regression formulation finds significantly better solutions. This demonstrates the interest of recovering the graph Laplacian. Our approach outperforms the other comparable methods until 80 training points. The recovered graph Laplacian is also closer (in Frobenius norm) to the true Laplacian. When the training set becomes larger, there is less benefit from multitask learning compared with independent ridge regression.

[2]The code comes from http://cbio.ensmp.fr/~ljacob/.

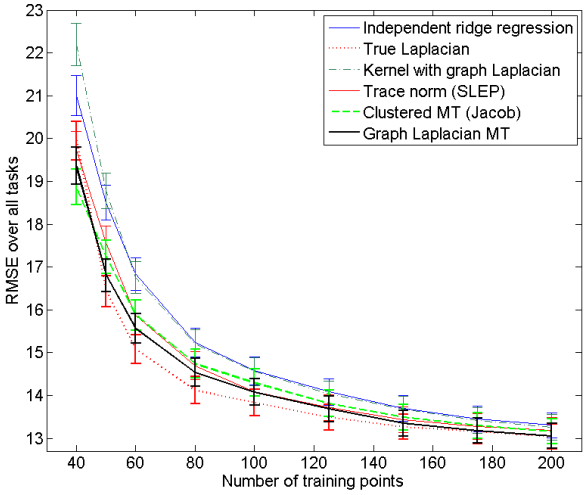


Figure 1. RMSE of each tested method on simulated data

3.2. Real Data: Movie Ratings

Our approach also proved to be effective on MovieLens data, a well known data set used for multitask learning and recommender systems.

3.2.1. The MovieLens 100k Data Set

The MovieLens 100k dataset[3] contains ratings of 1682 films by 943 users. The ratings are numbers among $\{1, 2, 3, 4, 5\}$, the films are described by their title, date, and categories they belong to, among $d = 19$ different genres. Information such as age, gender, occupation and zip code are provided about users.

To apply the different multitask methods on this data set, we consider each user as a task. We want to predict the rating a user gives to a film, based on the film's features. For each film, the feature vector is $(t, g_1, \ldots, g_d)$ where t is the release date and g_i is 1 if the film belongs to genre number i, and 0 otherwise. Each film can belong to several genres. Formally, for each task l, we want to regress the output vector Y_l of size m_l on its feature matrix X_l of size $m_l \times (d+1)$.

We used the training/test arrangements provided with this data set (MovieLens 100k set "a"), where 10 points were singled out as test data for each user, the rest being used for training and validation. The parameters for each method were tuned by 3-fold cross validation on the training subset. Tasks have very diverse data sizes, ranging from 20 to 737 ratings in total, with a median at 65. The learning problem has on average very few data observations per

[3]Full datasets are available at http://www.grouplens.org/node/73. These datasets were initially collected from the MovieLens website (movielens.umn.edu).

Table 1. Test error (MSE), validation error (MSE) and standard error (SE) for the MovieLens 100k dataset. Standard errors apply to the 3-fold cross-validation splits.

Method	Test	Validation	SE
Independent ridge	1.0896	1.0907	0.0027
Trace norm	1.0776	1.0875	0.0018
Clustered MT	1.0903	1.0900	0.0029
MT Laplacian	1.0725	1.0695	0.0029

task, thus we expect multitask learning approaches to outperform independent learning.

3.2.2. Results

Table 1 shows test errors, validation errors and validation standard errors for this dataset. Validation performance excluded tasks with less than 30 training points, to avoid tasks with too few samples and better estimate the overall error. By taking them out, validation proved to be much closer to test on all tasks. This process did not change the validated parameters compared with the full set. Since the error is well estimated, the standard error on cross validation splits is relevant in representing error variability.

Our method outperforms independent learning and the other approaches. Surprisingly, clustered multitask learning does not improve over independent learning on this problem. This is possible sometimes even though the non-convex formulation of (Jacob et al., 2009) subsumes independent learning. The reason is that the convex relaxation (3) does not subsume Frobenius regularization, due to the presense of the projection matrix Π. It is also interesting that MT Laplacian also outperforms trace norm regularization, which is a standard benchmark in recommendation problems. This fact may indicate a large significance of task clustering in problems like Movielens.

The graph Laplacian obtained with our method recovers meaningful clusters among users. The spectrum shows an optimal number of clusters around 5, and one can find groups that differ strongly from each other by their task coefficients. For instance, the year of issue and genres such as "Action", "Comedy", "Drama" and "Thriller" appear as the most differentiating features.

The next steps would include extending the experiments to larger MovieLens sets (1M and 10M).

4. Conclusion

We have proposed a novel multitask learning method for learning tasks which exhibit clustered graphs of relations. The method is based on regularization with a penalty that involves the Laplacian of the tasks graph. Our multitask formulation is unique in allowing for learning both the tasks and the Laplacian within the same optimization problem. We have also presented a conceptually simple alternating minimization algorithm which proved efficient for solving this problem up to a few thousand tasks. We have proposed a first-order convex optimization algorithm for solving a semidefinite program appearing as a subproblem of our method. Finally, we reported state of the art results showing performance improvement on both simulated and real datasets, simultaneously recovering well the graph of task relations.

References

Agarwal, A., Daumé III, H., and Gerber, S. Learning multiple tasks using manifold regularization. *Advances in Neural Information Processing Systems*, 23:46–54, 2010.

Ando, R. K. and Zhang, T. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

Argyriou, A., Micchelli, C. A., and Pontil, M. Learning convex combinations of continuously parameterized basic kernels. In *Proceedings of the Eighteenth Conference on Learning Theory*, pp. 338–352, 2005.

Argyriou, A., Evgeniou, T., and Pontil, M. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008a.

Argyriou, A., Maurer, A., and Pontil, M. An algorithm for transfer learning in a heterogeneous environment. In *European Conference on Machine Learning*, 2008b. To appear.

Argyriou, A., Micchelli, C. A., and Pontil, M. When is there a representer theorem? Vector versus matrix regularizers. *Journal of Machine Learning Research*, 10: 2507–2529, 2009.

Bakker, B. and Heskes, T. Task clustering and gating for bayesian multi–task learning. *Journal of Machine Learning Research*, 4:83–99, 2003.

Baldassarre, L., Morales, J., Argyriou, A., and Pontil, M. A general framework for structured sparsity via proximal optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 82–90, 2012.

Bauschke, H. H. and Combettes, P. L. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics. Springer, 2011.

Baxter, J. A model for inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

Ben-David, S. and Schuller, R. Exploiting task relatedness for multiple task learning. In *Proceedings of the Sixteenth Annual Conference on Learning Theory*, volume 2777, pp. 567–580, 2003.

Caponnetto, A., Micchelli, C. A., Pontil, M., and Ying, Y. Universal multi-task kernels. *The Journal of Machine Learning Research*, 9:1615–1646, 2008.

Caruana, R. Multi–task learning. *Machine Learning*, 28: 41–75, 1997.

Chapelle, O., Shivaswamy, P., Vadrevu, S., Weinberger, K., Zhang, Y., and Tseng, B. Boosted multi-task learning. *Machine learning*, 85(1-2):149–173, 2011.

Evgeniou, T., Micchelli, C. A., and Pontil, M. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

Fazel, M., Hindi, H., and Boyd, S. P. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings, American Control Conference*, volume 6, pp. 4734–4739, 2001.

Jacob, L., Bach, F., and Vert, J.-P. Clustered multi-task learning: a convex formulation. In *Advances in Neural Information Processing Systems 21*, pp. 745–752. MIT Press, 2009.

Kang, Z., Grauman, K., and Sha, F. Learning with whom to share in multi-task feature learning. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 521–528, 2011.

Kimeldorf, G.S. and Wahba, G. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41 (2):495–502, 1970.

Kumar, A. and Daumé III, H. Learning task grouping and overlap in multi-task learning. In *Proceedings of the 29th International Conference on Machine Learning*, pp. 1383–1390, 2012.

Lanckriet, G. R. G., Cristianini, N., Bartlett, P., Ghaoui, L. El, and Jordan, M. I. Learning the kernel matrix with semi-definite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

Liu, J., Ji, S., and Ye, J. *SLEP: Sparse Learning with Efficient Projections*. Arizona State University, 2009. URL `www.public.asu.edu/~jye02/Software/SLEP`.

Maurer, A. The Rademacher complexity of linear transformation classes. In *Proceedings of the 19th Annual Conference on Learning Theory (COLT)*, volume 4005 of *LNAI*, pp. 65–78. Springer, 2006.

Mirsky, L. A trace inequality of John von Neumann. *Monatshefte f ur Mathematik*, 79(4):303–306, 1975.

Rakotomamonjy, A., Bach, F., Canu, S., and Grandvalet, Y. SimpleMKL. *Journal of Machine Learning Research*, 9: 2491–2521, 2008.

Srebro, N. and Ben-David, S. Learning bounds for support vector machines with learned kernels. In *Proceedings of the Nineteenth Conference on Learning Theory*, volume 4005 of *LNAI*, pp. 169–183. Springer, 2006.

Srebro, N., Rennie, J. D. M., and Jaakkola, T. S. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems 17*, pp. 1329–1336. MIT Press, 2005.

Ying, Y. and Campbell, C. Generalization bounds for learning the kernel. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.

Annexe B
Recommandations PRISMS

Les pages qui suivent montrent un exemple de recommandations concrètes issues de la méthode présentée dans cette thèse. Elles sont publiées sous le nom de produit PRISMS, par la Recherche Quantitative d'Exane BNP Paribas. Il s'agit l'édition de juin 2013 de la série mensuelle "Market Risk report" , dont la publication a démarré en janvier 2012.

Le document en annexe a été exclu de la version publique de la thèse pour des raisons de droits de reproduction.

Bibliographie

[AC01] Robert Almgren and Neil Chriss. Optimal execution of portfolio transactions. *Journal of Risk*, 3 :5–40, 2001.

[ADEH99] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measure of risk. *Mathematical Finance*, 9(3) :203–228, 1999.

[ADG10] A. Agarwal, H. Daumé III, and S. Gerber. Learning multiple tasks using manifold regularization. *Advances in Neural Information Processing Systems*, 23 :46–54, 2010.

[AEP08] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3) :243–272, 2008.

[AMP05] A. Argyriou, C. A. Micchelli, and M. Pontil. Learning convex combinations of continuously parameterized basic kernels. In *Proceedings of the Eighteenth Conference on Learning Theory*, pages 338–352, 2005.

[AMP08] A. Argyriou, A. Maurer, and M. Pontil. An algorithm for transfer learning in a heterogeneous environment. In *European Conference on Machine Learning*, 2008. To appear.

[AT05] Jean-Yves Audibert and Alexandre B. Tsybakov. Fast learning rates for plug-in classifiers under the margin condition. 2005.

[AZ05] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6 :1817–1853, 2005.

[Bac00] Louis Bachelier. *Théorie de la Spéculation*. PhD thesis, École Normale Supérieure, 1900.

[Bax00] J. Baxter. A model for inductive bias learning. *Journal of Artificial Intelligence Research*, 12 :149–198, 2000.

[BBN08] T. Béchu, E. Bertrand, and J. Nebenzahl. *L'analyse technique*. Economica, 2008.

[BC11] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics. Springer, 2011.

[BD10] Gérard Biau and Luc Devroye. On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification. *Journal of Multivariate Analysis*, 101 :2499–2518, 2010.

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

[BDS03] S. Ben-David and R. Schuller. Exploiting task relatedness for multiple task learning. In *Proceedings of the Sixteenth Annual Conference on Learning Theory*, volume 2777, pages 567–580, 2003.

[BE02] Olivier Bousquet and André Elisseeff. Stability and Generalization. *Journal of Machine Learning Research*, 2 :499–526, 2002.

[BF97] L. Breiman and J. H. Friedman. Predicting multivariate responses in multiple linear regression. *Journal of the Royal Statistical Society, Series B*, 59(1) :3–54, 1997.

[BFOS84] Leo Breiman, Jerome H. Friedman, Robert A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Chapman & Hall, New York, 1984.

[BH03] B. Bakker and T. Heskes. Task clustering and gating for bayesian multi–task learning. *Journal of Machine Learning Research*, 4 :83–99, 2003.

[Bia12] Gérard Biau. Analysis of a random forests model. *Journal of Machine Learning Research*, 13 :1063–1095, 2012.

[BK86] D.S. Broomhead and Gregory P. King. Extracting qualitative dynamics from experimental data. *Physica D : Nonlinear Phenomena*, 20(2-3) :217–236, 1986.

[BL92] Fischer Black and Robert Litterman. Global portfolio optimization. *Financial Analysts Journal*, 1992.

[BM10] Joan Bruna and Stéphane Mallat. Classification with scattering operators. *IEEE CVPR Conference*, 2010.

[BMAP12] L. Baldassarre, J. Morales, A. Argyriou, and M. Pontil. A general framework for structured sparsity via proximal optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 82–90, 2012.

[BN93] Michèle Basseville and Igor V. Nikiforov. *Detection of abrupt changes : theory and application*. Prentice Hall, 1993.

[Bol86] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3) :307–327, 1986.

[BP82] M. Bertero and E.R. Pike. Resolution in diffraction-limited imaging, a singular value analysis. *Optica Acta : International Journal of Optics*, 29(6) :727–746, 1982.

[Bre96] Leo Breiman. Bagging Predictors. *Machine Learning*, 140 :123–140, 1996.

[Bre00] Leo Breiman. Some infinity theory for predictor ensembles. Technical report, University of Berkeley, 2000.

[Bre01] Leo Breiman. Random forests. In *Machine Learning*, pages 5–32, 2001.

[BS73] Fischer Black and Myron Scholes. The pricing of ooption and corporate liabilities. *Journal of Political Economy*, 81(3), 1973.

[BS01] S. Basak and A. Shapiro. Value-at-risk-based risk management : optimal policies and asset prices. *The Review of Financial Studies*, 14(2) :371–405, 2001.

[BSR04] Gilles Blanchard, Christin Schäfer, and Yves Rozenholc. *Learning Theory*, chapter Oracle Bounds and Exact Algorithm for Dyadic Classification Trees, pages 378–392. Springer Verlag Berlin Heidelberg, 2004.

[BT07] Peter L Bartlett and Mikhail Traskin. AdaBoost is Consistent. *Journal of Machine Learning Research*, 8 :2347–2368, 2007.

[BW08] Peter L Bartlett and Marten H. Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9 :1823–1840, 2008.

[BZ80] P. J. Brown and J. V. Zidek. Adaptive Multivariate Ridge Regression. *The Annals of Statistics*, 8(1) :64–74, January 1980.

[Car97] R. Caruana. Multi–task learning. *Machine Learning*, 28 :41–75, 1997.

[CBK09] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection : A survey. *ACM Computing Surveys (CSUR)*, 41(3), 2009.

[Cha09] Nicolas Chapados. *Sequential Machine Learning Approach for Portfolio Management*. PhD thesis, Université de Montréal, 2009.

[Cis14] Saïp Ciss. *Forêts uniformément aléatoires et détection des irrégularités aux cotisations sociales*. Phd thesis, Université Paris Ouest, 2014.

[CMPY08] A. Caponnetto, C. A. Micchelli, M. Pontil, and Y. Ying. Universal multi-task kernels. *The Journal of Machine Learning Research*, 9 :1615–1646, 2008.

[CSV+11] O. Chapelle, P. Shivaswamy, S. Vadrevu, K. Weinberger, Y. Zhang, and B. Tseng. Boosted multi-task learning. *Machine learning*, 85(1-2) :149–173, 2011.

[CW92] Ronald R. Coifman and Mladen Victor Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 1992.

[Dam] Aswath Damodaran. Equity risk premiums (erp) : Determinants, estimation and implications - the 2013 edition.

[Dau88] Ingrid Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41 :909–996, 1988.

[Die00] Thomas G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, pages 1–15, 2000.

[Don97] David L. Donoho. Cart and best-ortho-basis : a connection. *The Annals of Statistics*, 25(5) :1870–1911, 1997.

[EKM97] Paul Embrechts, Claudia Klüppelberg, and Thomas Mikosch. *Modelling extremal events : for insurance and finance*. Springer, 1997.

[EMP05] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6 :615–637, 2005.

[Eng82] Robert F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica : Journal of the econometric society*, pages 987–1007, 1982.

[Est03] Javier Estrada. Mean-semivariance behavior : A note. *Finance Letters*, 1 :9–14, 2003.

[Fam70] Eugene F. Fama. Efficient capital markets : A review of theory and empirical work. *The Journal of Finance*, 1970.

[FCLN08] Tak-Chung Fu, Fu-Lai Chung, Robert Luk, and Chak-Man Ng. Representing financial time series based on data point importance. *Engineering Applications of Artificial Intelligence*, 21 :277–300, 2008.

[FCN06] Tak-Chung Fu, Fu-Lai Chung, and Chak-Man Ng. Financial time series segmentation based on specialized binary tree representation. *Conference on Data Mining*, 2006.

[FCT] Jean Baptiste Faddoul, Boris Chidlovskii, and Fabien Torre. Learning Multiple Tasks with Boosted Decision Trees.

[FF93] Eugene F. Fama and Kenneth R. French. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33 :3–56, 1993.

[FF02] Eugene F. Fama and Kenneth R. French. The equity premium. *The Journal of Finance*, 57(2) :637–659, 2002.

[FG02] Laurent Favre and José-Antonio Galeano. Mean-modified value-at-risk optimization with hedge funds. *The Journal of Alternative Investments*, 5(2) :21–25, 2002.

[FH99] Jerome H. Friedman and Peter Hall. On bagging and nonlinear estimation. Technical report, 1999.

[FHB01] M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings, American Control Conference*, volume 6, pages 4734–4739, 2001.

[FHT00] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression : a statistical view of boosting. *The Annals of Statistics*, 28(2) :337–407, 2000.

[FISS03] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *The Journal of Machine Learning Research*, 4 :933–969, 2003.

[Fre95] Yoav Freund. Boosting a Weak Learning Algorithm by Majority. *Information and Computation*, 121(2) :256–285, September 1995.

[FS97] Yoav Freund and Robert E. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1) :119–139, August 1997.

[GN03] Servane Gey and Élodie Nedelec. *Nonlinear Estimation and Classification*, chapter Risk Bounds for CART Regression Trees, pages 369–379. Springer, 2003.

[GNZ01] Nina Golyandina, Vladimir Nekrutkin, and Anatoly Zhiglavsky. *Analysis of Time Series Structure - SSA and Related Techniques*. Chapman & Hall, 2001.

[GP05] Alexei A. Gaivoronski and Georg Pflug. Value-at-risk in portfolio optimization : Properties and computational approach. *Journal of Risk*, 2005.

[Gre07] William H. Greene. *Econometric Analysis*. Prentice Hall, 2007.

[GS66] David Martin Green and John A. Swets. *Signal detection theory and psychophysics*. Wiley, 1966.

[GU10] Nina Golyandina and K.D. Usevich. 2d-extension of singular spectrum analysis : algorithm and elements of theory. *Matrix Methods : Theory, Algorithms, Applications World Scientific*, pages 449–473, 2010.

[Ham89] James D. Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica : Journal of the econometric society*, pages 357–384, 1989.

[HD08] Blaise Hanczar and Edward R. Dougherty. Classification with reject option in gene expression data. *Bioinformatics*, 24(17) :1889–1895, 2008.

[HJLT96] Thomas Hancock, Tao Jiang, Ming Li, and John Tromp. Lower bound on learning decision lists and trees. *Informational Computing*, 126(2) :114–122, 1996.

[HNW05] Wei Huang, Yoshiteru Nakamori, and Shou-Yang Wang. Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, 32 :2513–2522, 2005.

[HR76] Laurent Hyafil and Ronald L. Rivest. Constructing optimal binary decision trees is np-complete. *Information Processing Letters*, 5(1), 1976.

[HS05] Peter Hall and Richard J. Samworth. Properties of bagged nearest neighbour classifiers. *Journal of the Royal Statistical Society, Series B*, 67(3) :363–379, 2005.

[HSST03] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis ; An overview with application to learning methods. Technical report, 2003.

[HTF01] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[HTF09] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning : Data Mining, Inference, and Prediction - Second Edition*. Springer, 2009.

[HW90] John Hull and Alan White. Pricing interest-rate-derivative securities. *Review of financial studies*, 3(4) :573–592, 1990.

[HW06] Radu Herbei and Marten H. Wegkamp. Classification with reject option. *Canadian Journal of Statistics*, 34 :709–721, 2006.

[Ito44] Kiyoshi Ito. Stochastic integral. *Proceedings of the Imperial Academy*, 20(8) :519–524, 1944.

[JBV09] L. Jacob, F. Bach, and J.-P. Vert. Clustered multi-task learning : a convex formulation. In *Advances in Neural Information Processing Systems 21*, pages 745–752. MIT Press, 2009.

[JMZ06] Hanqing Jin, Harry Markowitz, and Xun Yu Zhou. A note on semivariance. *Mathematical Finance*, 16 :53–61, 2006.

[JR96] JPMorgan and Reuters. *RiskMetrics - Technical Document*. 1996.

[Ké03] Balazs Kégl. Robust regression by boosting the median. In *16th Conference on Computational Learning Theory*, pages 258–272, 2003.

[KAYT90] T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka. Stock market prediction system with modular neural networks. *IJCNN International Joint Conference on Neural Networks*, 1 :1–6, 1990.

[KCHP04] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. Segmenting time series : A survey and novel approach. *Data mining in time series database*, 57 :1–22, 2004.

[KD12] A. Kumar and H. Daumé III. Learning task grouping and overlap in multi-task learning. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1383–1390, 2012.

[KE07] Leonidas Karamitopoulos and Georgios Evangelidis. Current trends in time series representation. In *Panhellenic Conference in Informatics (PCI)*, 2007.

[Ken38] M.G. Kendall. A new measure of rank correlation. *Biometrika*, 30, 1938.

[KGS11] Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *Proceedings of the 28th International Conference on Machine Learning*, pages 521–528, 2011.

[KK03] Eamonn Keogh and Shruti Kasetty. On the need for time series data mining benchmarks : a survey and empirical demonstration. *Data Mining and knowledge discovery*, 7(4) :349–371, 2003.

[KPSW92] Robert King, Charles I. Plosser, James H. Stock, and Mark Watson. Stochastic trends and economic fluctuations. 1992.

[KR12] Azadeh Khaleghi and Daniil Ryabko. Locating changes in highly dependent data with unknown number of change points. *Advances in Neural Information Processing Systems*, 25, 2012.

[KR13] Azadeh Khaleghi and Daniil Ryabko. Nonparametric multiple change point estimation in highly dependent time series. In *Proc. 24th International Conf. on Algorithmic Learning Theory (ALT'13)*, 2013.

[KR14] Azadeh Khaleghi and Daniil Ryabko. Asymptotically consistent estimation of the number of change points in highly dependent time series. In *ICML, JMLR W&CP*, volume 32, pages 539–547, 2014.

[KV04] R.L. Karandikar and M. Vidyasagar. Probably approximately correct learning with beta mixing input sequences. 2004.

[KVS07] Dragi Kocev, Celine Vens, and Jan Struyf. Ensembles of Multi-Objective Decision Trees. pages 624–631, 2007.

[KW70] G.S. Kimeldorf and G. Wahba. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41(2) :495–502, 1970.

[KX10] Seyoung Kim and Eric P Xing. Tree-Guided Group Lasso for Multi-Task Regression with Structured Sparsity. pages 1–14, 2010.

[LCB+04] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semi-definite programming. *Journal of Machine Learning Research*, 5 :27–72, 2004.

[Lin65] John Lintner. The valuation of risk assets and the selection of risky investments in stock protfolios and capital budgets. *The review of economics and statistics*, pages 13–37, 1965.

[LJY09] J. Liu, S. Ji, and J. Ye. *SLEP : Sparse Learning with Efficient Projections*. Arizona State University, 2009.

[LKS05] Aurélie C. Lozano, Sanjeev R. Kulkarni, and Robert E. Schapire. Convergence and Consistency of Regularized Boosting Algorithms with Stationary β -Mixing Observations. *Advances in Neural Information Processing Systems*, pages 819–826, 2005.

[LL12] Damien Lamberton and Bernard Lapeyre. *Introduction au Calcul Stochastique Appliqué à la Finance*. Ellipses, 2012.

[LLP11] Sophie Laruelle, Charles-Albert Lehalle, and Gilles Pagès. Optimal split of orders across liquidity pools : a stochastic algorithm approach. *SIAM Journal of Financial Mathematics*, 2(1) :1042–1076, 2011.

[LMW00] Andrew W. Lo, Harry Mamaysky, and Jiang Wang. Foundations of technical analysis : Computational algorithms, statistical inference, and empirical implementation. *The Journal of Finance*, 55(4) :1705–1765, 2000.

[LPTvdG09] K. Lounici, M. Pontil, A.B Tsybakov, and S. van de Geer. Taking advantage of sparsity in multi-task learning. In *Proc. of the 22nd Annual Conference on Learning Theory (COLT)*, 2009.

[LV04] Gábor Lugosi and Nicolas Vayatis. On the bayes-risk consistency of regularized boosting methods. *The Annals of Statistics*, 32(1) :30–55, 2004.

[Mah12] Nicolas Mahler. *Machine Learning Methods for Discrete Multi-scale Flows : Application to Finance*. PhD thesis, École Normale Supérieure de Cachan, 2012.

[Mal96] Burton G. Malkiel. *A Random Walk Down Wall Street*. 1996.

[Mal00] Stéphane Mallat. *Une exploration des signaux en ondelettes*. Les Éditions de l'École Polytechnique, 2000.

[Mar52] Harry M. Markowitz. Portfolio selection harry markowitz. *The Journal of Finance*, 7(1) :77–91, 1952.

[Mar70] Harry M. Markowitz. *Portfolio Selection - Efficient Diversification of Investment*. John Wiley & Sons, 1959 - 1970.

[Mau06] A. Maurer. The Rademacher complexity of linear transformation classes. In *Proceedings of the 19th Annual Conference on Learning Theory (COLT)*, volume 4005 of *LNAI*, pages 65–78. Springer, 2006.

[McN99] A. J. McNeil. Extreme value theory for risk managers. Departement Mathematik ETH Zentrum., 1999.

[Mei06] Nicolai Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7 :983–999, 2006.

[Mer73] Robert C. Merton. Theory of rational option pricing. *The Bell Journal of Economics and Management Science*, 4(1), 1973.

[Mer74] Robert C. Merton. On the pricing of corporate debt : The risk structure of interest rates. *The Journal of Finance*, 29 :449–470, 1974.

[Mer92] Robert C. Merton. *Continuous-Time Finance*. Wiley-Blackwell, 1992.

[Meu05] Attilio Meucci. *Risk and Asset Allocation*. Springer, 2005.

[Mey89] Yves Meyer. *Orthonormal wavelets*, pages 21–37. Springer-Verlag, 1989.

[Mir75] L. Mirsky. A trace inequality of John von Neumann. *Monatshefte f ur Mathematik*, 79(4) :303–306, 1975.

[MMZ02] Shie Mannor, Ron Meir, and Tong Zhang. The Consistency of Greedy Algorithms for Classification. *Computational Learning Theory*, 2375 :319–333, 2002.

[Mos66] Jan Mossin. Equilibrium in a capital asset market. *Econometrica : Journal of the econometric society*, pages 768–783, 1966.

[MR10a] Mehryar Mohri and Afshin Rostamizadeh. Stability bounds for stationary ϕ-mixing and β-mixing processes. *Journal of Machine Learning Research*, 2010.

[MR10b] Éric Moulines and François Roueff. *Analyse des Séries Temporelles et Applications*. Télécom ParisTech, 2010.

[MS03a] Markos Markou and Sameer Singh. Novelty detection : a review - part 1 : statistical approaches. *Signal Processing*, 83 :2481–2497, 2003.

[MS03b] Markos Markou and Sameer Singh. Novelty detection : a reviw - part 2 : neural networks based approaches. *Signal Processing*, 83 :2499–2521, 2003.

[Nob02] Andrew B. Nobel. Analysis of a complexity based pruning scheme. *IEEE Transactions on Information Theory*, 48 :2362–2368, 2002.

[PI04] Cheol-Ho Park and Scott H. Irwin. The proftability of technical analysis : A review. Technical report, AgMAS, 2004.

[QF14] Pascal Quiry and Yann Le Fur. *Finance d'entreprise*. Dalloz, 2014.

[QR89] John Ross Quinlan and Ronald L. Rivest. Inferring descision trees using the minimum description length principle. *Information and Computation*, 80(227-248), 1989.

[Qui87] John Ross Quinlan. Simplifying decision trees. *International Journal of Man-Machine Studies*, 27 :221–234, 1987.

[Qui93] John Ross Quinlan. *C4.5 : Programs for Machine Learning*. Morgan Kaufmann, 1993.

[Ram02] James B. Ramsey. Wavelets in economics and finance : Past and future. *Studies in Nonlinear Dynamics & Econometrics*, 6(3), 2002.

[RBCG08] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9 :2491–2521, 2008.

[RM05] Lior Rokach and Oded Maimon. Top-Down Induction of Decision Trees Classifiers—A Survey. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 35(4) :476–487, November 2005.

[Ron14] Thierry Roncalli. Big data in asset management. In *7th Financial Risks International Forum : Big Data in Finance and Insurance*, 2014.

[Ros76] Stephen A. Ross. The arbitrage theory of capital asset pricing. *Journal of Economic Theory*, 13 :341–360, 1976.

[RSM02] O. Renaud, J.-L. Starck, and F. Murtagh. Wavelet-based forecasting of short and long memory time series. Technical report, Université de Genève, 2002.

[RU00] R Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2 :21–42, 2000.

[Rub02] Mark Rubinstein. Markowitz's "portfolio selection" : A fifty-year retrospective. *The Journal of Finance*, LVII, 2002.

[SBD06] N. Srebro and S. Ben-David. Learning bounds for support vector machines with learned kernels. In *Proceedings of the Nineteenth Conference on Learning Theory*, volume 4005 of *LNAI*, pages 169–183. Springer, 2006.

[Sch90] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2) :197–227, June 1990.

[Sch99] Robert E. Schapire. Theoretical Views of Boosting and Applications. In Osamu Watanabe and Takashi Yokomori, editors, *Proceedings of the 10th International Conference on Algorithmic Learning Theory*, volume 1720 of *Lecture Notes in Computer Science*, pages 13–25, Berlin, Heidelberg, May 1999. Springer Berlin Heidelberg.

[Sew07] Martin Sewell. Technical analysis. Technical report, University College London, 2007.

[Sha64] William F. Sharpe. Capital asset prices : a theory of market equilibrium under conditions of risk. *The Journal of Finance*, 19(3) :425–442, 1964.

[Sha66] William F. Sharpe. Mutual fund performance. *The Journal of Business*, 39(1) :119–138, 1966.

[SHS09] Ingo Steinwart, Don Hush, and Clint Scovel. Learning from dependent observations. *Journal of Multivariate Analysis*, 100(1) :175–194, 2009.

[Sil05] B.W. Silverman. *Functional Data Analysis*. Springer, 2005.

[SL12] Yan Sun and Xiaodong Lin. Regularization for stationary multivariate time series. *Quantitative Finance*, 12(4) :573–586, 2012.

[Sol13] Matthieu Solnon. *Apprentissage statistique multi-tâches*. Phd thesis, Université Pierre et Marie Curie, 2013.

[SRJ05] N. Srebro, J. D. M. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems 17*, pages 1329–1336. MIT Press, 2005.

[SS99] Robert E. Schapire and Yoram Singer. Improved Boosting Algorithms Using Confidence-rated Predictions. *Machine Learning*, 336 :297–336, 1999.

[SS09] Nicholas I. Sapankevych and Ravi Sankar. Time series prediction using support vector machines : A survey. *IEEE Computational Intelligence Magazine*, 4(2) :24–38, 2009.

[ST02] J. Shadbolt and J.G. Taylor. *Neural Networks and the Financial Markets : Predicting, Combining, and Portfolio Optimisation*. Springer, 2002.

[Tob58] James Tobin. Liquidity preference as a behavior towards risk. *Review of Economic Studies*, 67 :65–86, 1958.

[Tsa05] Ruey S. Tsay. *Analysis of Financial Time Series*. Wiley, 2005.

[TSN89] Christopher M. Turner, Richard Startz, and Charles R. Nelson. A markov model of heteroskedasticity, risk, and learning the stock market. *Journal of Financial Economics*, 25(1) :3–22, 1989.

[Ury00] Stanislav Uryasev. Conditional value-at-risk : optimization algorithms and applications. *IEEE CIFER*, pages 49–57, 2000.

[Vay06] Nicolas Vayatis. *Approches statistiques en apprentissage : boosting et ranking*. Hdr, Université Pierre et Marie Curie (Paris VI), 2006.

[vL07] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4) :395–416, August 2007.

[Wes01] Douglas B. West. *Introduction to Graph Theory*. Prentice Hall, 2001.

[Wil07] Paul Wilmott. *Paul Wilmott Introduces Quantitative Finance*. John Wiley & Sons, 2007.

[WZCG08] Qing Wang, Liang Zhang, Mingmin Chi, and Jiankui Guo. MTForest : Ensemble Decision Trees based on Multi-Task Learning. pages 122–126. IOS Press, 2008.

[YC09] Y. Ying and C. Campbell. Generalization bounds for learning the kernel. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.

[Yu94] Bin Yu. Rates of convergence for empirical processes of stationary mixing sequences. *The Annals of Probability*, 22(1) :94–116, 1994.

[YW10] Ming Yuan and Marten H. Wegkamp. Classification methods with reject option based on convex risk minimization. *Journal of Machine Learning Research*, 11 :111–130, 2010.

[ZB00] Hans Zantema and Hans L. Bodlander. Finding small equivalent decision trees is hard. *International Journal of Foundations of Computer Science*, 2000.

[ZCY12] JiaYu Zhou, JianHui Chen, and JiePing Ye. *Multi-task Learning via Structural Regularization*. Arizona State University, 2012.

Apprentissage statistique en gestion de portefeuile

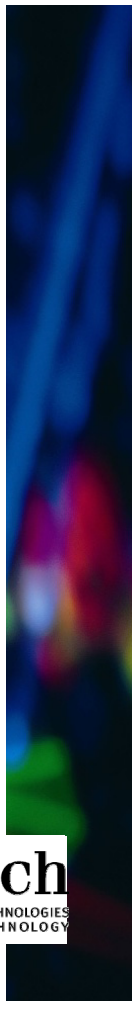
Ruocong ZHANG

RESUME : La prédiction des rendements d'actifs financiers est une problématique essentielle en optimisation de portefeuille. La nature de séries temporelles des données financières implique des difficultés telles que la dépendance temporelle, la non-stationnarité, la dimensionnalité élevée et une dépendance complexe entre les actifs. Notre but est de prédire le signe des rendements par une méthode objective, adaptative, non-paramétrique et produisant des résultats interprétables. Nous concevons à cette fin un ensemble de méthodes d'apprentissage statistique couvrant toutes les étapes, de la représentation des données au test de performance et à la validation des paramètres.
Les séries financières sont d'abord indexées au moyen d'une approximation linéaire par morceaux reposant sur une segmentation par arbres. Cette représentation fournit les descripteurs en entrée de la classification binaire, que nous traitons par l'algorithme de Random Forests. Nous adjoignions à la prédiction une option de rejet permettant d'ignorer le résultat si le niveau de confiance est insuffisant. Afin de tenir compte des dépendances entre les tâches de prédiction sur différents actifs, nous proposons alternativement une approche multi-tâche faisant intervenir un graphe dont ces tâches sont les sommets. Le laplacien du graphe, qui reflète la structure de dépendance, est utilisé pour régulariser la minimisation du risque empirique. Notre approche originale apprend simultanément les tâches et le graphe. Enfin, un protocole de test séquentiel rend compte de la qualité de notre méthode en utilisation pratique.
Ce processus de prédiction montre de bonnes performances en application réelle.

MOTS-CLEFS : apprentissage statistique, gestion de portefeuille, apprentissage multi-tâche, classification, arbres

ABSTRACT : Asset return prediction is a key problem in portfolio optimization. Financial data are time series with inherent difficulties such as time dependence, non-stationarity, and dependence among high-dimensional features. Our goal is to predict the sign of asset returns with an objective, adaptive and non-parametric method that provides interpretable results. We design a suite of statistical learning methods for this purpose, addressing all steps from data representation to performance testing and parameter validation.
Financial time series are first indexed by piecewise linear approximation based on tree-induced segmentation. This representation is then used as the input of a binary classification problem, solved by a Random Forests algorithm. We enable a reject option to the outputs, so that one can choose whether to follow or ignore the predictions based on their confidence. As an alternative, we also propose a multitask learning approach, considering the prediction tasks on different assets as vertices in a graph accounting for the dependence among them. This formulation is original because we use the graph Laplacian to regularize the empirical risk minimization and jointly solve for both the tasks and the graph of relations. Last, the sequential test protocol reliably reflects generalization quality in real uses.
This complete process proves suitable for real-world application with good realized performance.

KEY-WORDS : statistical learning, portfolio management, multitask learning, classification, trees



TELECOM ParisTech

EDiTE

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY