TELECOM SudParis

ANNÉE : 2013/2014

RAPPORT de « MISSION en ENTREPRISE »

présenté par

Buron Cédric

Mission effectuée du 16 septembre 2013 au 14 mars 2014 chez :

eFront

sujet de la mission :

Prédiction du comportement financier d'un fonds

Directeur de stage : Landrein Geordy

Conseillère de stage : Brunet Elisabeth

Travail effectué pour la société : e-Front

TELECOM SudParis

Stage de 3ème ANNÉE
2013/2014

Nom de l'étudiant : Cédric Buron
Option : Sans objet (double diplôme à la NCTU)
Entreprise : eFront
Dates : 16 septembre – 14 mars
Sujet : Prédiction du comportement financier d'un fonds

RÉSUMÉ :
Le stage avait pour objectif d'établir et d'implémenter un modèle de cash-flow forecasting au sein de la société eFront. Au cours du stage, différentes méthodes de data mining, d'apprentissage automatique, ainsi que des méthodes propres aux séries temporelles ont été utilisées. Le stage a été réalisé au sein d'une entreprise d'édition de logiciels pour la finance non cotée, dans l'équipe Java. Il nous a également permis de découvrir le fonctionnement d'une application web, puisque le produit auquel a été intégré notre module est une application web.
Au-delà des aspect théorico-technique, le stage nous a permis de rencontrer un domaine bien particulier, la finance non cotée. En effet, bien que dans une équipe de Recherche et Développement, la problématique du stage nécessitait de comprendre de nombreux aspect fonctionnels, et la connaissance métier acquise a été d'un grand secours tout au long du développement. D'un point de vue humain, nous avons eu l'occasion de vivre un moment de mutation très particulier et intense pour notre équipe en particulier.

ABSTRACT :
The aim of this internship was to establish ansd implement a cash flow forecasting model in the company eFront. During this internship, we used data mining and machine learning methods. We also used some specific time series techniques. We spent the internship in a software editor for private equity, in the Java team. During the internship, we have had the chance to discover how a web application works, as we integrated our module into a web application.
Beyond the theoretical and technical aspects, We could discover the world of private equity. Even working in the Research and Development team, our problematic required us to understand the functional aspects, and the knowledge we acquired helped us quite a lot during the whole time. From a human point of view, we have had the opportunity to see how a mutation looks like in such a big company, as our team in particular has lived many changes.

Remerciements

Avant toute chose, je souhaite remercier Geordy, qui a accepté de m'intégrer dans son équipe et m'a fait confiance tout au long de ce stage, acceptant de se fier à mes idées et à mes intuitions. Il m'a soutenu dans les épreuves rencontrées et s'est avéré être un interlocuteur attentif ainsi qu'un manager attentionné.

Merci aussi à Elisabeth Brunet, ma conseillère pédagogique, qui a précautionneusement relu ce rapport dans son ensemble, et m'a fourni de nombreux conseils, toujours avisés. Merci aussi pour le contact qu'elle m'a fourni quand j'étais en difficulté. Au-delà, elle m'a toujours remonté le moral quand j'en ai eu besoin, ce qui n'a pas de prix.

Merci à toute l'équipe Java. Frédéric en premier lieu, qui m'a introduit au sujet, et avec qui j'aurais beaucoup aimé travailler plus longtemps. C'est lui qui est notamment à l'origine de ce stage pour lequel il m'aurait sans doute encadré dans d'autres circonstances. Merci à Raphaël, qui a tout de suite porté intérêt à mon projet, et m'a aidé lors d'une lutte épique contre une matrice. Merci à Denis, compagnon du petit matin, dont l'aide sur les parties techniques a été primordiale. Tes petits gâteaux sont délicieux. Merci aussi à Samuel pour son aide sur des parties de ce rapport autant que sur des points de programmation.

Un immense merci à Cédric et Pascal, qui m'ont soutenu et aidé à comprendre les bases de l'informatique, mais qui ont surtout partagé au quotidien leurs vies et leurs projets, et ont su me redonner confiance en moi. Merci à Olivier, pour ses conseils avisés et son aide prodiguée avec une amabilité sans faille, ainsi que pour son sens de l'humour. Je doute que tu le saches mais la séance de code review que nous avons faite ensemble m'a redonné confiance en moi. Merci à Xavier pour le temps qu'il a pris pour me répondre. Merci aussi pour le code review qu'il a fait comme ça, simplement pour m'aider. Finalement, tes coups de gueule cachent assez mal une immense sympathie. Merci enfin à Nicolas qui avec François m'a rappelé quelques souvenir taïwanais, ce qui m'a fait chaud au cœur.

Merci aussi à Baptiste, qui malgré un emploi du temps toujours plein a su trouver le temps pour me répondre, me donner des conseils et m'aider. Merci à Simon, avec lequel j'ai pu discuter de tout et de rien, de l'entreprise mais aussi de jeux de société. Merci aussi à Benoît pour ses conseils avisés sur mon projet, mais aussi sur le reste. En espérant te croiser un jour à Lérat.

Merci à ceux qui, en dehors de l'entreprise, m'ont apporté leur soutien. Ma compagne, ma famille et mes amis, qui m'ont souvent encouragé. Un grand merci enfin à mon père qui a pris le temps de relire ce rapport et d'en corriger les fautes.

Table des matières

Résumé **i**

Remerciements **iii**

Introduction **1**

I Présentation de l'entreprise **3**

1 Les chiffres et informations clés 5

2 Contexte : fonds d'investissement et marché non coté 5

3 Fonctionnement d'un capital-investissement 7

II La mission **9**

1 Contexte de la mission **11**

1.1 Contextualisation et vocabulaire financier 11

1.1.1 Net Asset Value 11

1.1.2 Net Present Value 11

1.1.3 IRR 12

1.1.4 La J-curve 12

1.2 Problématique 13

1.3 Le logiciel d'eFront : Pevara 13

1.3.1 La J-Curve 13

1.3.2 Les données à prendre en compte 14

1.3.3 Obligations légales 14

2 Modélisation du comportement du fonds **17**

2.1 Régression et Autorégression 17

2.1.1 Définitions 17

2.1.2 Autorégression 18

2.1.3 Régression linéaire prenant en compte le domaine 18

2.1.4 Conclusion 21

2.2 Plus proches voisins 22

2.2.1 Définitions 22

2.2.2 Méthode de l'apprentissage 23

2.2.3 Distance entre fonds et analyses factorielles 23

2.2.4 Fonction de distance de séries 28

2.2.5 Conclusion 30

2.3 Classification non supervisée 30

2.3.1 Définitions 30

2.3.2 Apprentissage hiérarchique 30

2.3.3 Adaptation du modèle 32

2.3.4 Conclusion 33

2.4 Bilan : choisir un modèle 33

3 Implémentation **37**

3.1 Généralités 37

3.2 Modèle linéaire 37

3.2.1 Les données à prédire 37

3.2.2 Prise en compte de l'âge interne du fonds 38

3.2.3 Résolution de l'équation et calcul de Θ 38

3.3 k plus proches voisins 40

3.4 Analyses factorielles 40

3.4.1 Besoin 40

3.4.2 Implémentation 40

3.5 Edit Distance for Real sequences 41

3.5.1 Normalisation des données 41

3.5.2 Implémentation 41
3.6 Classification non supervisée 43
3.7 Intégration des modules 43
3.7.1 Paramètres des séries 43
3.7.2 Récupération des fonds 45
3.7.3 Revue de code 45

4 Évaluations **47**
4.1 Démarche 47
4.2 Régression utilisant les k-plus-proches-voisins 47
4.3 Régression utilisant la classification non supervisée. 48
4.4 Un élément de comparaison : le module CFF V2 49
4.4.1 La méthode de cash flow forecasting V2 49
4.4.2 Comparaison 49
4.5 Bilan et axes d'amélioration 50

5 La vie en entreprise **51**
5.1 Contexte 51
5.2 Une entreprise internationale 52
5.3 Le fonctionnement d'une équipe de développeurs 53
5.4 Déroulement du stage 54
5.5 Bonnes pratiques 54
5.6 Bilan 55

6 Conclusion et perspectives **57**

Annexes **A-3**

A Organigrammes et chiffres clés de l'entreprise **A-3**

B J-Curves tirées des données de Pevara **A-7**

C Les données des fonds, classées selon différents critères **A-9**

D Les licences et leurs droits **A-13**

E Modèles d'autorégression **A-15**

E.1 Apprentissage statistique et séries temporelles A-15

E.1.1 Les séries stationnaires A-15

E.1.2 Séries non stationnaires et Modèle ARIMA A-17

E.1.3 Neural AutoRegressive model (NAR) A-17

E.2 Support Vector Machines (SVM) pour la régression A-19

F Analyses factorielles **A-23**

F.1 L'Analyse en Composantes Principales A-23

F.1.1 Notations A-23

F.1.2 Matrice centrée réduite A-23

F.1.3 L'analyse A-24

F.1.4 Critère d'arrêt A-24

F.1.5 Poids A-24

F.2 L'Analyse Factorielle des Correspondances A-25

F.2.1 Métrique du Φ^2 . . . . . A-25

F.2.2 L'analyse A-28

F.3 L'analyse de correspondances multiples A-29

F.3.1 Les différentes représentations des données A-29

F.3.2 Métrique du Φ^2 . . . . . A-31

F.3.3 L'analyse A-33

G Distances de séries réelles **A-35**

G.1 Dynamic Time Warping A-35

G.2 Longest common subsequence A-35

H Apprentissage non supervisé **A-37**

H.1 *k-means* A-37

H.2 Cartes de Kohonen A-39

I Outils **A-41**

I.1 Maven A-41
I.2 Source Control Manager A-42
I.2.1 Principe A-42
I.2.2 Fonctionnement A-42
I.3 Spring A-43
I.4 WSDL, SOAP et Axis2 . . . . . A-43

Glossaire **A-45**

Acronymes **A-47**

Bibliographie **A-49**

Table des figures

1 Marché non coté 6

1.1 Un exemple de J-curve 12

2.1 Tableau de variables pour l'AFDM 24

2.2 Les défauts de la distance euclidienne 29

2.3 Exemple de dendrogramme 32

2.4 Fonctionnement de notre module, option 1 34

2.5 Fonctionnement de notre module, option 2 35

5.1 Évolution des effectifs de l'équipe Java présente à Paris 52

A.1 Organisation d'eFront A-3

A.2 Organisation d'eFront France A-4

A.3 Évolution du chiffre d'affaire d'eFront entre 2006 et 2012 . . . . . A-5

B.1 Une J-curve utilisant l'IRR A-7

B.2 Une J-curve extraite des données recueillies A-7

B.3 Une autre J-curve extraite des données recueillies A-8

B.4 Cette J-curve est issue de la moyenne des Cumulated cash flows sur les fonds démarrés en 1999 . . . . . A-8

E.1 Un exemple de courbe stationnaire de moyenne 5 . . . . . A-16

E.2 Un exemple de réseau neuronal A-18

E.3 Un exemple de perceptron à trois couches cachées et à sortie unique A-18

E.4 Exemple d'utilisation de SVM sur une J-Curve A-20

E.5 Approximation de SVM faussée par un ε trop grand A-20

G.1 Les défauts du dynamic Time Warping A-36

H.1 Exemple de *k-means* pour 3 classes A-38

I.1 Organisation d'un dossier géré par Maven A-41

I.2 Fonctionnement d'un gestionnaire de version A-43

I.3 Appels de librairies A-44

Liste des tableaux

2.1 AFDM : exemple 25

2.2 AFDM : tableau disjonctif 26

2.3 AFDM : matrice réduite 27

3.1 Adaptation des données qualitatives 44

4.1 Évaluation de la régression couplée aux k-plus-proches-voisins . . . 47

4.2 Temps d'exécution de chaque module de la régression linéaire multiple couplée aux k-plus-proches-voisins, en ms 48

4.3 Évaluation de la régression couplée à la classification non supervisée 48

4.4 Temps d'exécution de chaque module de la régression linéaire multiple couplée à la classification non supervisée, en ms 48

4.5 Comparaison de notre méthode et de CFF V2 49

C.1 Distribution des fonds A-11

D.1 Les différentes licences A-13

F.1 Tableau d'AFC : exemple A-25

F.2 Tableau d'AFC : fréquences A-26

F.3 Tableau d'AFC : fréquences lignes A-27

F.4 Tableau d'AFC : fréquences colonnes A-27

F.5 Tableau d'AFC : taux de liaison A-29

F.6 Tableau disjonctif complet : exemple A-30

F.7 Tableau de Burt : exemple A-32

Introduction

La finance. Ce mot nous fait immédiatement penser aux salles de marché, à la Bourse, aux actions. Pourtant, la finance des actions (finance de marché) est loin d'englober l'ensemble de la finance. Elle n'en représente même qu'une petite partie de la finance puisqu'elle en concerne que les entreprises suffisamment importantes pour êtreintroduites en bourse. Beaucoup d'autres sont rachetées par des fonds, qui les gèrent pour le compte de gros clients : banques, assurances, groupes internationaux. C'est ce que l'on appelle le capital-investissement.

Ces clients autant que les fonds ont besoin d'outils pour gérer leurs investissements et pour analyser leurs performances. Elles sont aidées pour cela par des entreprises comme eFront qui fournissent aux acteurs du capital-investissement des outils de suivi, de gestion et d'analyse. C'est dans ce cadre que s'est déroulé le stage. Les investisseurs veulent connître et prévoir des flux de trésorerie sortant et surtout entrant, et savoir si à terme l'investissement a des chances d'être rentable.

eFront possède de nombreuses données, mais pas de module permettant de prévoir la trésorerie à partir de ces informations. La mission a consisté établir un modèle en utilisant les informations présentes dans le logiciel, et de prévoir des flux de trésorerie, et plus précisément des flux de trésorerie entrants.

Traiter cette problématique nécessite des connaissances métier afin d'appréhender au mieux les données et leurs intéractions. Cela nécessite aussi des compétences en data mining et en statistique, afin de traiter effectivement les données. Enfin, cela demande des capacités de développement, afin que le modèle élaboré soit implémenté et opérationnel.

Notons que plusieurs approches ont d'ores et déjà été abordées pour ce problème. En utilisant les méthodes de Monte Carlo, en moyennant les fonds identiques à celui que nous cherchons à prévoir. Notre méthode est plus aboutie : elle inclut de nombreuses techniques de data mining et de régression. Nous donnons aussi des indications sur l'entreprise dans laquelle le stage a été réalisé, ainsi que notre ressenti de la vie dans l'entreprise.

La première partie présente eFront, l'entreprise dans laquelle a été réalisé le stage, son positionnement sur le marché ainsi que les chiffres qui le caractériesent.

La seconde partie sera quant à elle dédiée à la mission dans son ensemble. Le premier chapitre contextualise la mission, donnne des définitions de base et précise la problématique du stage. Le second chapitre décrit les méthodes que nous avons élaborées et les raisons qui nous y ont conduit. Le troisième chapitre détaille la phase

d'implémentation les difficultés rencontrées et les solutions apportées. Nous donnons dans les quatrième chapitre les évaluations de nos deux méthodes et les comparons à une méthode préexistant dans l'entreprise. La cinquième partie porte sur l'aspect plus humain de notre stage et de son déroulement au sein de l'équipe Java. Enfin, le sixième chapitre synthétise les différents aspects du stage et donne une ouverture sur de futures perspectives, tant de notre point de vue que de celui du code produit.

Le rapport contient également un certain nombre d'annexes, qui permettront de comprendre différents aspects technico-mathématiques du projet, les différentes solutions envisagées et les causes de leur abandon. Enfin, le rapport contient un glossaire, une liste d'acronymes ainsi qu'une bibliographie.

Première partie

Présentation de l'entreprise

1 Les chiffres et informations clés

eFront est une entreprise créée en 1999 par Olivier Dellenbach. Elle édite des logiciels pour la finance, en particulier dans le monde de la finance non cotée (fonds d'investissement). L'entreprise a commencé par créer une plateforme de développement.Le premier produit développé a été FrontInvest, qui permet tant aux investisseurs qu'aux gestionnaires des fonds d'investissement de gérer leurs investissements. Cependant, d'autres produits sont apparus, notamment des logiciels de gestion des risques, ou encore de suivi des investissements (Pevara).

eFront est présente dans le monde entier, en France, où elle est née, mais aussi en Chine, au Royaume-Uni, en Inde (où elle a récemment racheté une entreprise), au Canada, à Hong-Kong, à Dubaï... Elle compte ainsi quelque 260 employés, et plus de 300 clients à travers le monde (*cf.* Figure A.1).

La section Recherche et Développement (R&D) se trouve véritablement au cœur de l'entreprise, puisqu'elle développe les solutions vendues aux clients. L'entreprise possède plusieurs autres sections dont : une branche support (ressources humaines, Finance etc.), une section de consulting qui intègre les produits chez les clients, une section pré-ventes qui traduit les besoins des clients à la R&D, et une section entièrement dédiée à la gestion de risques (*cf.* Figure A.2).

Le chiffre d'affaire de l'entreprise est en hausse constante depuis 2006, comme le montre la Figure A.3. Les développeurs travaillent en .NET pour la plupart, mais le projet Pevara est écrit en Java.

2 Contexte : fonds d'investissement et marché non coté

On se représente souvent une entreprise comme une Société Anonyme (ou SA), cotée en bourse. Il existe pourtant un marché non coté, *i.e.* non présent en bourse. Sur ce marché, l'investissement se fait indirectement, en passant par des fonds d'investissement. Lorsqu'un investisseur souhaite investir dans le marché non coté, il n'investit en général pas directement dans l'entreprise, car l'investissement dans ces entreprises demande une expertise et un temps qu'a rarement l'investisseur. Il passe donc par un fonds, qui s'occupe de l'achat d'entreprises, de leur vente et éventuellement de leur gestion.

Dans ce milieu, on appelle **Limited Partners (LPs)** les investisseurs et **General Partners (GPs)** les fonds d'investissement

La Figure 1 donne une idée du fonctionnement de ce marché. Les flèches représentent les flux d'investissement.

Les fonds sont généralement gérés par des équipes assez restreintes. En revanche, les risques présentés par l'investissement dans les marchés non cotés implique que les investisseurs sont souvent de grandes entreprises, parfois des banques. Le contexte est donc le suivant :

- Les LPs sont souvent de grandes structures, des grandes entreprises ou groupes.

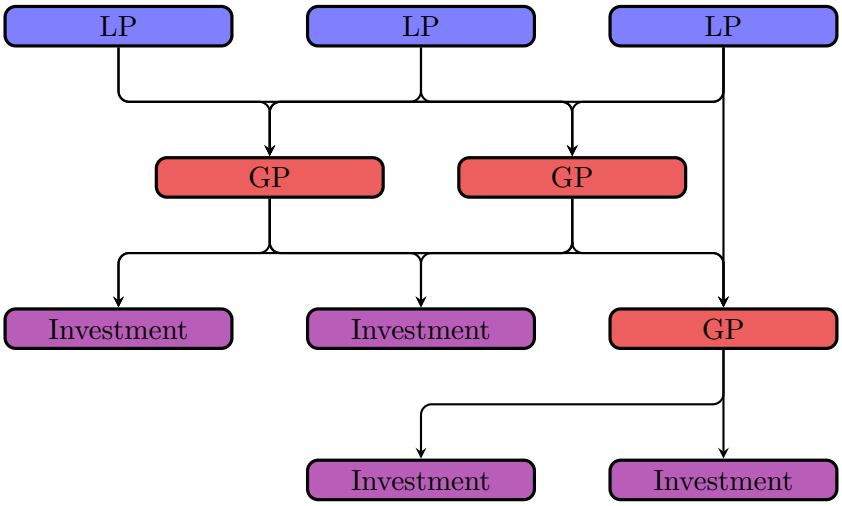


Figure 1 – Marché non coté

- Les GPs, au contraire, sont de petites structures, n'ayant en général pas plus de quelques dizaines d'employés.
- Les investments sont des entreprises diverses. Il peut s'agir de petites structures (start-ups) mais aussi de plus grandes entreprises (eFront en fait elle-même partie). Les différents cas amènent à différentes stratégies pour la gestion de ces entreprises par les fonds.

Ces marchés ne sont pas des marchés à court terme comme peuvent l'être les marchés cotés. Il s'agit d'investissements sur le long terme, parfois sept ou dix ans. La rentabilité n'est pas immédiate et suit un modèle particulier : la « J-curve », décrite dans la section jcurvesec de la partie suivante. Ils présentent un risque plus important que l'investissement coté mais ont souvent une espérance de gain plus élevée.

Les fonds d'investissement se divisent en plusieurs catégories :

- Le capital risque (ou *Venture Capital (VC)* en anglais) : il s'agit d'investissements risqués. Il peut s'agir d'investissements dans des start-ups ou bien dans des entreprises encore jeunes.
- Le *Real Estate* : il s'agit d'investissements immobiliers. Ces investissements sont très particuliers et comportent moins de risque.
- Le *Leverage BuyOut (LBO)* : ils s'agit de participer au rachat d'une entreprise, s'endettant pour augmenter ses capitaux propres.

Dans ce rapport, nous nous intéresserons particulièrement aux investissements de type Venture Capital et LBO car ce sont les plus présents dans nos bases.

Les plus grandes entreprises de fonds du monde sont :

1. Goldman Sachs Principal Investment Area
2. TPG Capital
3. Carlyle Group (client d'eFront)

Notons qu'il ne s'agit que de fonds américains.

3 Fonctionnement d'un capital-investissement

Un capital-investissement (*private equity* en anglais) se déroule en plusieurs temps :

1. Le fonds appelle des capitaux, ou plus précisément des promesses d'investissement des LP,
2. Le fonds recherche une opportunité d'achat,
3. Une fois l'opportunité trouvée, le fonds commence à investir l'argent des LP,
4. Les investissements se poursuivent,
5. Le fonds rémunère ses investisseurs grâce aux résultats des entreprises dans lesquelles il a investi,
6. Le fonds revend les entreprises et rembourse (selon son bénéfice) ses investisseurs.

Le fonds se rémunère principalement de deux façons : tout d'abord, il fait payer à ses investisseurs des frais lors de l'engagement de ceux-ci (première étape ci-dessus). De plus il investit lui aussi dans les entreprises dans lequel il engage les capitaux des LPs et est donc rémunéré de la même façon qu'eux.

Une fois l'accord signé, les LPs ont peu de pouvoirs. Souvent, ils ne peuvent pas influer sur les décisions d'investissement du GP, et ne peuvent se désengager qu'exceptionnellement (par exemple si la personne qui gère le fonds n'est plus en mesure de le faire) et lorsque le fonds est liquidé.

Afin que les GPs se sentent engagés dans leur fonds, ils constituent une partie de celui-ci. Cette partie est souvent modeste, car le GP est souvent, en regard du LP, une petite entreprise (1% du fonds en général). Il reçoit de plus une part significative des gains s'il l'investissement dépasse les attentes (typiquement, si l'investissement rapporte plus de 6% de plus que ce qui était attendu, le GP récolte 20% de ce supplément).

Une des particularités du marché est la quasi-absence de marché de seconde main. Les « personnes clés » qui gèrent les fonds ont une importance capitale.

Le milieu du *private equity* est en fait assez fermé, et très peu perméable aux nouveaux arrivants, qu'il s'agisse de GPs ou de LPs. en effet, en ce qui concerne les GPs, outre que le domaine exige un grand savoir faire, ils auront du mal à attirer des LPs fiables, et ainsi à créer un fonds pérenne. En ce qui concerne les LPs, les fonds les plus pérennes n'ont qu'un nombre réduit de LPs, qu'ils gardent en général d'un fonds sur l'autre. Il est donc difficile pour les nouveaux arrivants de trouver un GP connu, et pour le néophyte, seuls ceux-ci semblent assez sûrs pour constituer un investissement potentiel.

Finalement, au lieu de considérer les LPs et les GPs, il est plus raisonnable de considérer les duos LP–GP. En pratique, l'accord entre un LP et un GP se fait sur plusieurs fonds, appelés « follow-on », dans lesquels les LPs sont invités à réinvestir chez le même GP. Cela continue jusqu'à ce qu'une des deux parties souhaite arrêter. Pour les raisons citées ci-dessus, les LPs et GPs n'aiment pas changer de partenaires.

Deuxième partie

La mission

Chapitre 1

Contexte de la mission

Dans ce chapitre, nous décrivons l'état du projet tel qu'il était avant que nous arrivions ainsi que les prérequis à sa compréhension. Nous introduisons ici des notations et concepts plus orientés métier et intrinsèquement liés à notre misison.

1.1 Contextualisation et vocabulaire financier

1.1.1 Net Asset Value

La *Net Asset Value (NAV)* ou Valeur liquidative en français représente la valeur liquidative d'un fonds. Elle se calcule ainsi :

$$NAV = actifs - dettes - frais\_restants$$

Notons que dans ce rapport, nous considérons la NAV du point de vue du fonds (GP) et non de celui de l'investisseur (LP). Ainsi, nous ne divisons pas cette valeur par le nombre de parts.

1.1.2 Net Present Value

La *Net Present Value (NPV)* ou Valeur Actuelle Nette en français (VAN) est la valeur comptable de l'entreprise. Elle correspond au prix qu'on retirerait de l'entreprise si on la vendait à l'instant donné. Elle est formellement définie comme suit :

$$NPV(\tau, N) = \sum_{p=0}^{p=N} \left(\frac{Cashflows}{(1+\tau)^p} \right) - I + NAV(N)$$

où :

- $Cashflows$ représente l'ensemble des cash flows (positifs et négatifs),
- τ est le taux d'actualisation (il permet de prendre en compte de l'évolution de la valeur de l'argent, inflation et déflation),
- I est l'investissement,
- N est le nombre d'encaissements de cash flows.

1.1.3 IRR

L'*Intern Rate of Return (IRR)* ou encore Taux de retour interne en français (TRI) est un taux utilisé en finance afin de savoir si un projet de durée déterminé sera rentable. Il n'est pas unique. Les taux de rentabilité possibles après la n-ième opération sont les racines de l'équation suivante en τ :

$$NPV(\tau, N) = 0$$

avec les notations précédentes.

Il est possible d'obtenir un taux unique en utilisant le Modified IRR MIRR (ou en français TRIM), ayant la valeur suivante :

$$MIRR = \sqrt[n]{\frac{PososositiveCashlows}{NegativeCashflows}} - 1$$

Notons que le MIRR n'est pas un IRR, mais un autre indicateur qui en est indépendant.

1.1.4 La J-curve

Généralités

Un investissement dans un capital-investissement suit une forme particulière : la J-curve. L'allure de cette courbe est donnée en Figure 1.1. Sur cette figure, on représente les cash flows cumulés ou l'IRR de l'ensemble des LPs.

Cette courbe peut être utilisée sur plusieurs types de donnés. Dans notre domaine, on obtient des J-curves sur deux types de données : les cash flows agrégés ou l'IRR (*cf.* section 1.1.3).

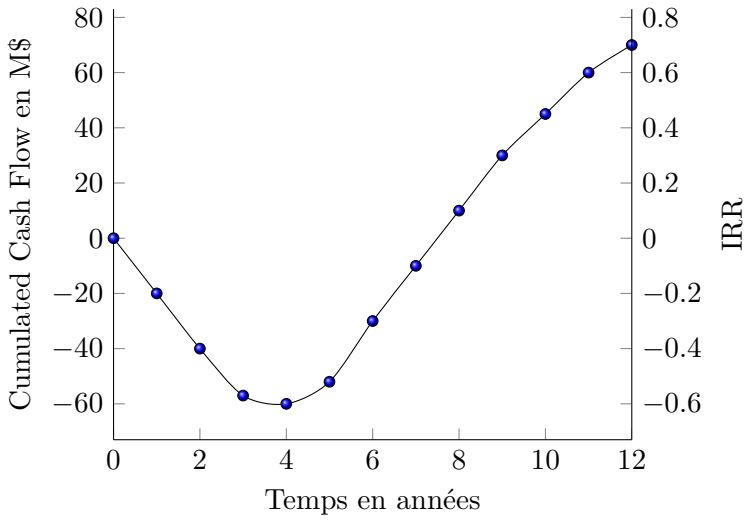


Figure 1.1 – Un exemple de J-curve

La forme de cette courbe s'explique ainsi : au début de l'investissement, le fonds appelle des capitaux et les investit immédiatement (cash flows négatifs pour les

LPs) dans les entreprises du fonds. Il ne rémunère donc pas les investisseurs. Après un certain temps, les investissements ont tous été faits. Il n'y a donc plus d'investissement de la part des LPs (plus de cash flows négatifs). Les distributions commencent, puisque les entreprises deviennent rentables (cash flows positifs). Cependant, la courbe finit par s'aplatir et le fonds finit par revendre les entreprises. Pour plus de précisions sur cette courbe, voir [1].

Modèle mathématique

Il n'existe pas véritablement de « modèle mathématique » pour la J-Curve. Nous avons donc essayé de réaliser une régression linéaire multiple à partir des données présentes dans Pevara. Cependant, comme les J-curves peuvent beaucoup différer d'un fonds à l'autre, il nous faudra diviser les J-curves en plusieurs catégories.

1.2 Problématique

Le but de la mission est d'établir un modèle de Cash flow Forecasting et de l'étalonner grâce aux données de Pevara. À partir de données nominales du fonds (sa localisation, sa devise etc ;) et éventuellement de ses *calls*, nous souhaitons prévoir tous ses cash flows. Notons que comme nous l'avons dit, les cash flows cumulés suivent une forme particulière, la J-Curve. Une fois le modèle construit, nous l'avons implémenté et intégré à Pevara.

En réalité, la mission est partie du constat d'un ingénieur avant vente, qui a fait remarquer qu'un modèle existait sur un autre produit d'eFront, FrontAnalytics (qui est intégré dans le produit phare d'eFront, FrontInvest). Ce produit permet à l'investisseur (LP ou GP) de remplir manuellement les différentes opérations, paramètres et attentes qu'il a afin de lui montrer les cash Flows à venir. Dans un souci d'efficacité, il serait plus facile de remplir le modèle plus ou moins automatiquement, plutôt que de le faire remplir intégralement par le client.

Comme nous l'avons expliqué plus haut, il est possible (et nécessaire dans l'établissement de notre modèle) de diviser et de catégoriser les fonds. Aucune division objective n'existait dans Pevara. Or, visualiser ces catégories et voir quels paramètres ont un impact sur la J-curve permettrait de comprendre un peu mieux les données présentes en base. Les problématiques de ce stage s'apparentent donc à du data mining.

1.3 Le logiciel d'eFront : Pevara

1.3.1 La J-Curve

Pevara nous permet d'observer de très nombreuses données recueillies par eFront. Cependant, il faut garder à l'esprit que les données ont déjà été « pré-traitées ». Par exemple, les NAVs sont ajoutées automatiquement si elles manquent.

La première observation est que les différentes « J-Curves » présentes sur Pevara ne ressemblent pas à des J-curves. Après observation des cash flows, nous remarquons que les NAVs sont présentes. Afin d'obtenir des données conformes (du moins en surface) à ce que nous attendons. Nous décidonsde faire abstraction du calcul fourni de l'IRR et de ne nous intéresser qu'aux Cash-Flows, en retirant les valuations (correspondant aux NAVs). En effet, celles-ci sont laissées à l'appréciation du gestionnaire de fonds, qui les remplit sans garantie de conformité ni même de cohérence avec la réalité. Si la NAV est manquante, elle est généré selon des règles qui ne peuvent reproduire la réalité. Or, comme on le peut le voir dans la section 1.1.3, la NAV intervient dans le calcul de l'IRR. Cela pourrait expliquer les écarts importants que nous observons. Nous donnons un exemple d'une telle J-Curve (qui n'en est pas une) sur la Figure B.1

Nous avons cependant créé nos propres courbes à partir des cash flows, affichées sur les Figures B.2, B.3 et B.4.

1.3.2 Les données à prendre en compte

Avant de rechercher une méthode d'apprentissage automatique, il nous faut étudier des données, pour savoir dans quels champs nous disposons d'assez de donnés pour faire une véritable analyse. Pour un fonds, il existe les champs suivants :

Size La taille du fonds, en devises locales.

Vintage L'anné où le fonds a été créé.

Industry L'industrie à laquelle appartient le fonds.

Geography La région du monde à laquelle appartient le fonds.

Strategy Il s'agit du type de fonds.

Status Il s'agit d'indiquer si le fonds est liquidé ou encore actif.

Currency La devise utilisée pour le fonds.

Or, il se trouve que ces paramètres ne sont pas forcéments assez représentés pour que nous puissions les prendre en compte dans notre analyse, comme le montrent les Tables C.1a, C.1b, C.1c, C.1d et C.1e.

En conséquence, nous avons décidé de nous intéresser à cinq facteurs. Le premier est le « vintage », ou l'année de création du fonds ; le second est le type de fonds, la « strategy », le troisième est la « currency », le quatrième est la « geography ». Enfin, le dernier est la taille du fonds. Les autres facteurs ne nous semblent pas présenter assez de données non nulles pour pouvoir constituer un vrai critère d'analyse.

1.3.3 Obligations légales

Les données présentes dans Pevara sont privées, fournies par les utilisateurs. Elles peuvent être utilisées pour créer un benchmark, mais certaines actions sont prohibées :

- sortir les données de Pevara,
- utiliser une donnée seule : les données sont utilisées en groupe, pour des benchmarks seulement (6 fonds au moins sont nécessaire),

- les fonds doivent rester anonymes.

De même, Pevara est un logiciel commercial, aussi est-il impératif de veiller aux licences des librairies que nous utiliserons, et éviter les librairies virales tels GPL, AGPL, et leur préférer des licences plus adaptées à un usage commercial (*cf.* Table D.1). Notons également qu'il y a deux types de données dans Pevara. Les données privées sont fournies par le client pour comparaison sur Pevara. Nul n'y a accès, à l'exception du client lui-même. Les données publiques sont récoltées *via* FrontInvest. Elles servent à créer des benchmarks, et passent par un processus d'anonymisation avant d'être envoyées à Pevara, depuis FrontInvest. Ainsi, ceux qui les manipulent ne peuvent pas savoir à quelle entreprise elles appartiennent.

Chapitre 2

Modélisation du comportement du fonds

Il existe plusieurs manière de prédire une série temporelle à partir de quelques valeurs et/ou de données nominales. Dans ce chapitre, nous exposons celles que nous avons envisagées, avant de conclure avec celle que nous avons finalement choisie et de donner nos raisons.

Nous avons commencé par nous intéresser aux méthodes de régression car ce sont celles qui sont habituellement proposées pour compléter une série temporelle comme la nôtre. Néanmoins, certaines de ces méthodes ne prennent en compte que le début de la série temporelle considérée et non les autres données, que nous avons en nombre dans notrre cas. Nous nous sommes donc ensuite interrogés sur l'adaptation de méthodes de classification non supervisée à notre problème, ce qui nous a paru plus prometteur.

2.1 Régression et Autorégression

2.1.1 Définitions

Commençons par définir ce que sont la régression et l'autorégression.

Définition 2.1.1 (Régression)
Une méthode de régression est une méthode permettant de trouver une relation (mathématique) entre plusieurs variables.

Pour deux variable x et y, une régression s'écrira $f(x, y) = 0$ où f est la fonction associée à cette régression.

Définition 2.1.2 (Autorégression)
Une méthode d'autorégression est une méthode de régression pour les séries temporelles. Dans cette méthode, on suppose que la valeur d'un terme de la série temporelle dépend des valeurs précédentes de la même série temporelle.

Dans notre cas, l'apprentissage se fait sur les fonctions de régression et d'autorégression et les paramètres de ces fonctions.

2.1.2 Autorégression

Nous avons étudié plusieurs méthode d'autorégression, qui sont décrites dans l'Annexe E. Cependant, ces modèles ne prennent en compte que le début de la série considérée, et non l'ensemble des séries que nous avons sur Pevara. Nous les laisserons donc de côtés et nous nous intéresserons à un modèle de régression linéaire. Dans ce modèle, nous utiliserons pour variables connues un ensemble de séries déjà connues, tirées de Pevara.

2.1.3 Régression linéaire prenant en compte le domaine

Principes

Les modèles présentés ci-dessus sont des modèles généraux, ne prenant pas en compte la connaissance que nous avons du domaine d'étude. En effet, d'une part nous savons quelle est la forme de la courbe ; d'autre part, nous notons que les distributions dépendent avant tout des investissement faits. On adopte donc un modèle linéaire. Ce modèle a été décrit dans [2] avant notre arrivée par Frédéric Paulin, qui était en charge du projet avant notre arrivée.

On appelle c_t l'appel de fonds fait au moment t, d_t la distribution faite au moment t et h_t la prédiction faite de d_t. On suppose alors que h_t ne dépend que des investissement faits précédemment et de paramètres correspondants. Ainsi,

$$h_t = \sum_{k=1}^{t} \theta_{tk} \cdot c_{1+t-k}$$

On associe de plus une matrice à chaque variable ci-dessus :

$$C = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}$$

$$D = \begin{pmatrix} d_1 \\ \vdots \\ d_n \end{pmatrix}$$

$$\Theta = \begin{pmatrix} \theta_{1,1} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ \theta_{n,n} & \dots & \dots & \theta_{n,1} \end{pmatrix}$$

et

$$H = \begin{pmatrix} h_1 \\ \vdots \\ h_n \end{pmatrix}$$

Remarquons que Θ est une matrice triangulaire inférieure. On obtient alors :

$$H = \Theta C$$

Nous faisons en sus une autre simplification. Nous supposons que $\theta_{i,t}$ ne dépend pas de t *i.e.* $\theta_{i,t} = \theta_i$, ce qui signifie que la contribution d'un investissement ne dépend que du temps le séparant de la distribution considérée. La matrice Θ devient alors :

$$\Theta = \begin{pmatrix} \theta_1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ \theta_n & \dots & \dots & \theta_1 \end{pmatrix}$$

et

$$h_t = t = \sum_{k=1}^{t} \theta_k \cdot c_{1+t-k}$$

et nous notons

$$\theta = \Big(\theta_1, \dots, \theta_i, \dots, \theta_n\Big)$$

Afin d'étalonner le modèle, d'obtenir la matrice Θ, nous l'apprenons sur un ensemble de m fonds. Nous notons ainsi d_t^f la distribution faite au temps t du fonds f, et h_t^f la prévision que nous en avons faite. Nous notons

$$\mathbf{h_t} = \begin{pmatrix} h_t^1 & \cdot & h_t^f & \dots & h_t^m \end{pmatrix} \text{ et } \mathbf{d_t} = \begin{pmatrix} d_t^1 & \cdot & d_t^f & \dots & d_t^m \end{pmatrix}$$

Nous écrivons donc une fonction de coût globale

$$J : (\theta_1, \dots, \theta_n) \mapsto \frac{1}{2m} \sum_{t=1}^{n} \|\mathbf{h_t} - \mathbf{d_t}\|^2$$

dont nous déduisons

$$J(\theta) = \frac{1}{2m} \sum_{t=1}^{n} \sum_{i=1}^{m} (h_t^f - d_t^f)^2$$

Résolution par descente du gradient

Nous allons maintenant minimiser cette fonction par une approche classique de descente du gradient, comme décrit ci-dessous :

Il faut cependant ajouter des contraintes. En effet, il est contre-intuitif d'obtenir un θ négatif. Cela signifierait en effet qu'un *call* entraîne une distribution négative. De plus, obtenir un θ_1 non nul serait étrange : cela signifierait qu'un *call* produit une distribution immédiate, ce qui est illogique.

Algorithme 1 Descente du gradient

1 : $\theta_k \leftarrow \theta_0$
2 : **repeat**
3 : $\quad \theta_k \leftarrow \theta_k - \left(\overrightarrow{\nabla J}\right)_k$ ▷ $\forall k \in [\![0, n]\!]$
4 : **until** $\dfrac{\partial J(\theta)}{\partial \theta_m} \leq \varepsilon$

Résolution par programmation linéaire

Il est possible d'utiliser de la programmation linéaire si on utilise la norme L_1. Le problème s'écrit alors

$$\sum_{f \in kPPV} \sum_{t \leq |f|} |\sum_{j=0}^{t} \theta_j \cdot c^f_{t-j+1} - d^f_t|$$

Ce problème est connu, il s'agit de la *Least Absolute Deviation (LAD)*. Ce problème peut se résoudre en utilisant la programmation linéaire, en ajoutant des variables u_i et en écrivant le problème :

$$\begin{aligned} \textit{minimiser :} \ & \sum_{t,f} u_{t,f} \\ \textit{sous contrainte :} \ & u_{t,f} \geq d^f_t - \sum_{j=0}^{t} \theta_j \cdot c^f_{t-j+1} \\ & u_{t,f} \geq -d^f_t + \sum_{j=0}^{t} \theta_j \cdot c^f_{t-j+1} \end{aligned}$$

Nous ajouterons de plus des contraintes afin que $\theta_i \geq 0$. Nous obtenons alors :

$$\begin{aligned} \textit{minimiser :} \ & \sum_{t,f} u_{t,f} \\ \textit{sous contrainte :} \ & u_{t,f} \geq d^f_t - \sum_{j=0}^{t} \theta_j \cdot c^f_{t-j+1} \\ & u_{t,f} \geq -d^f_t + \sum_{j=0}^{t} \theta_j \cdot c^f_{t-j+1} \\ & \theta_j \geq 0 \end{aligned}$$

Résolution par méthode des moindres carrés

Cependant, le meilleur estimateur n'est pas forcément celui-ci. En effet, ce problème peut être résolu en norme euclidienne $\|.\|_2$. On utilise dans ce cas une méthode appelée *NonNegative Least Squares (NNLS)* (ou NNLS). Il s'agit de l'approximation des moindres carrés, avec la contrainte que chaque paramètre soit positif ou nul. Cette méthode, est décrite dans [3] et reprise dans [4].

L'équation du problème s'écrit également

$$H = C'\Theta'$$

$$\text{avec } C' = \begin{pmatrix} c_1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ c_n & \dots & \dots & c_1 \end{pmatrix} \text{ et } \Theta' = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_n \end{pmatrix}$$

Nous cherchons à déduire les paramètres du modèle. Pour cela, nous souhaitons résoudre l'équation en Θ'

$$D = C'\Theta'$$

de façon optimale pour tous les fonds qui sont « proches » de notre fonds. Nous pouvons traduire ces différentes équations par une équation unique. Nous notons D_i, C'_i, les matrices de l'équation pour le $i^{ème}$ voisin.

Nous pouvons alors traduire l'ensemble de ces équations par :

$$\begin{pmatrix} D_1 \\ D_2 \\ \vdots \\ D_k \end{pmatrix} = \begin{pmatrix} C'_1 \\ C'_2 \\ \vdots \\ C'_k \end{pmatrix} \cdot \Theta'$$

Cette équation n'a pas de solution exacte (il y a plus d'équations que de variables) mais il est possible d'optimiser Θ' de façon à minimiser la distance euclidienne entre la matrice résultat et son modèle. Notons qu'afin d'alléger les notations, nous noterons

$$H \leftarrow \begin{pmatrix} D_1 \\ D_2 \\ \vdots \\ D_k \end{pmatrix} \text{ et } C' \leftarrow \begin{pmatrix} C'_1 \\ C'_2 \\ \vdots \\ C'_k \end{pmatrix}$$

Cette méthode fait partie de la famille des algorithmes *active set* au sens où elle garde un ensemble pour lequel les contraintes sont respectées en posant $x_i = 0$ (l'ensemble dit **actif**) et un autre ensemble, **passif** pour lesquels les contraintes sont $x_i > 0$. De plus amples explications sur la méthode *active set* sont données dans [5].

2.1.4 Conclusion

Nous choisissons donc pour modèle la régression linéaire prenant en compte le domaine. Cependant, dans cette régression, il y a deux inconnues. Les distributions, et les paramètres sur lesquels faire la rgression. Nous devons donc choisir un certains nombre de fonds ayant assez de valeurs pour faire notre régression. Ces fonds

doivent être « proches » du fonds que nous voulons prédire. Nous avons envisagé pour sélectionner ces fonds deux méthodes. La première est une sélection des k-plus-proches voisin. La seconde est de classifier les fonds. Une fois la classification non supervisée réalisée, nous plaçons le fonds à prédire dans la classe qui lui correspond le mieux, puis nous faisons la régression sur cette classe. Ces deux méthodes sont décrites dans les sections 2.2 et 2.3

2.2 Plus proches voisins

Une des façons de sélectionner des fonds proches de celui que nous souhaitons prévoir est d'utiliser les k-plus-proches-voisins, sur une distance entre fonds définie par nos soins.

2.2.1 Définitions

Il est possible d'utiliser les k-plus-proches-voisins dans le cas de la régression (voir [6] pour plus d'information). Pour ce faire, il faut :

1. définir une fonction de proximité de deux courbes,
2. nettoyer les données, afin de ne pas avoir à chercher un k trop grand,
3. sur l'ensemble d'entraînement, calculer les valeurs de chaque courbe en prenant en compte de manière identique les k-plus-proches-voisins, et mesurer l'écart entre notre courbe et la courbe réelle,
4. réitérer ce procédé pour plusieurs k, jusqu'à trouver le k optimal (apprentissage),
5. valider le modèle.

La principale problématique soulevée par ce modèle est de trouver une fonction de distance entre les fonds (*quelles sont les caractéristiques des fonds à prendre en compte ?* ou, pour coller aux notations de [6], *quelles sont les dimensions à conserver et quelle fonction distance peut-on utiliser ?*).

Notons qu'il est possible (et assez courant) d'améliorer cette méthode en pondérant les différents voisins selon leur distance à la requête, par exemple en utilisant l'interpolation de Shepard, décrite dans [7].

Définition 2.2.1
Interpolation de Shepard

Dans une interpolation de Shepard, la valeur interpolée $\mathbf{x}$ *à partir d'un ensemble de N valeurs* $(\mathbf{x}_i)_{i\leq N}$ *se définit par :*

$$\mathbf{x} = \frac{1}{\sum_{i=1}^{N} \frac{1}{d(\mathbf{x}, \mathbf{x}_i)}} \sum_{i=1}^{N} \frac{1}{d(\mathbf{x}, \mathbf{x}_i)} \cdot \mathbf{xi}$$

où $d(\mathbf{x}, \mathbf{x}_i)$ *est la distance entre* $\mathbf{x}$ *et* $\mathbf{x}_i$*.*

2.2.2 Méthode de l'apprentissage

Normalement, la méthode des k-plus-proches-voisins est une méthode dite **paresseuse**, c'est-à-dire qu'il n'y a pas à apprendre les paramètres du modèles avant de faire la classification (contrairement aux méthodes qui utilisent des surfaces séparatrices). Il est néanmoins nécessaire de déterminer k et de s'assurer que ce paramètre convient.

Pour vérifier que le k est acceptable, on utilise la cross-validation, plus particulièrement la leave-one-out cross-validation (LOOCV). Dans cette méthode, nous vérifions k pour chaque fonds, c'est à dire que pour chaque fonds, nous considérons l'ensemble des autres fonds, puis nous le tronquons (nous conservons uniquement les données d'entrée).

Nous appliquons alors notre méthode de bout en bout, et en retirons une prévision, que nous comparons au reste de la véritable série temporelle, via une distance Edit Distance for Real sequences (EDR) aux paramètres identiques à ceux dont nous nous servons dans notre programme (*cf. infra*). Nous avons aussi utilisé la distance euclidienne, afin de valider une seconde fois nos résultats.

Notons qu'avant toute chose, il nous faut déterminer la fonction de distance entre deux fonds, ce que nous permettra l'analyse factorielle des données mixtes.

2.2.3 Distance entre fonds et analyses factorielles

Le but d'une analyse factorielle est de réduire le nombre des dimensions sur lesquelles on travaille et de les décorréler. Il existe pour cela plusieurs méthodes, décrites succinctement dans l'Annexe F. Cette méthode nous permettra d'obtenir une représentation numérique des fonds, et ainsi de calculer une distance entre les fonds.

Le choix d'une analyse factorielle

L'analyse factorielle la plus connue est l'Analyse en Composantes Principales (ACP). Cependant, elle n'est pas l'analyse qui correspond le mieux à notre situation. En effet, l'ACP est utilisée sur des données situées dans des espaces vectoriels. Or, si la distance entre les séries temporelles ou la taille d'un fonds peuventt être considérées comme réels (appartenant donc à des espaces vectoriels), les autres données, telles la localisation, la devise etc. sont des énumérations, des données quantitatuives qui ne peuvent pas être vues comme faisant partie d'un espace vectoriel. Il existe en revanche d'autres analyses factorielles qui permettent le traitement de données qualitatives.

L'Analyse Factorielle des Correspondances (AFC) et l'Analyse des Correspondances Multiples (ACM) sont deux méthodes associées à l'analyse de donnés qualitatives de respectivement deux variables ou plus. Elles utilisent comme « mesures » les fréquences des différentes modalités de chaque variable qualitative. Ces analyses sont exposées en Annexe F et le lecteur aura soin de les lire avant de poursuivre sa lecture s'il n'a pas de notion à ce propos.

Dans la suite du chapitre, nous décrivons une méthode d'analyse qui permet l'analyse de données mixtes (qualitatives et quantitatives), et tient donc à la fois de l'ACP et de l'ACM.

Analyse Factorielle pour les Données Mixtes

Cette méthode est décrite dans un article de Jérôme Pagès, [8]. L'idée principale est de mélanger une ACP avec une ACM.

Représentation des données Nous mettons les données sous la forme d'un tableau, croisement entre un tableau d'ACP pour les données quantitatives et d'un tableau disjonctif complet pour les donnés qualitatives, comme montré en Figure 2.1.

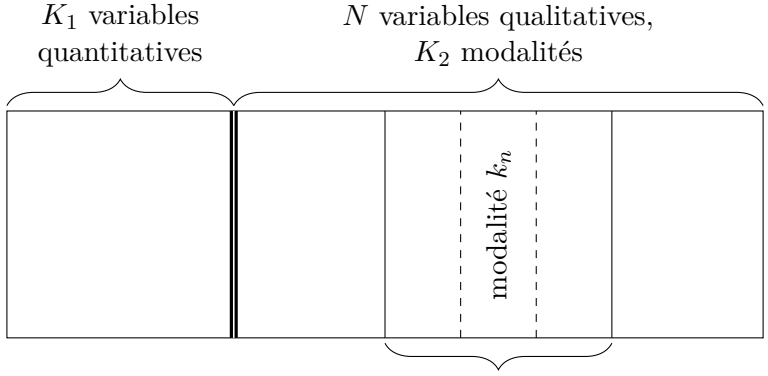


Figure 2.1 – Tableau de variables pour l'AFDM

L'article présente différentes manières équivalentes de réaliser l'Analyse Factorielle des Données Mixtes (AFDM), mais nous ne nous intéresserons qu'à la façon « ACP ». Avant de réaliser l'analyse, il nous faut « normaliser » les données.

Normalisation des données L'ACP que nous réaliserons ne sera pas normalisée (non réduite) car les variables qualitatives et quantitatives ne sont pas représentées de la même façon (représentation sous forme d'ACP pour les variables quantitatives et ACM pour les variables qualitatives). Ainsi, il nous faut préparer les donnés avant de réaliser l'analyse. Nous centrons et réduisons donc les données quantitatives.

Pour les données qualitatives, nous commençons par les centrer, et puis les réduire. Ordinairement, en ACM, les données centrées sont réduites en divisant chaque valeur par $N \cdot p_{k_n}$ avec p_{k_n} la fréquence de la modalité k_n, c'est à dire la moyenne de cette modalité. Ici, puisque nous comparons des données de type différents, nous nous contenterons de normaliser en divisant par p_{k_n}. Cependant, nous souhaitons que ce soient les données de la matrice variance-covariance qui soient pondérées ainsi. Nous divisons donc les valeurs par $\sqrt{p_{k_n}}$ (*cf.* F.1.5).

Analyse Cette normalisation nous permet d'obtenir les égalités suivantes (les explications sont données dans la seconde section de [8] :

1. pour une variable quantitative k et une variable centrée y, $\langle k, y\rangle = r^2(k, y)$, la corrélation entre k et y
2. pour une variable qualitative n et une variable centrée y, l'équation s'écrit $\sum_{k_n \in n} \langle k_n, y\rangle = \eta^2(n, y)$, le rapport de corrélation entre n et y

Nous cherchons à maximiser la corrélation entre les données et les nouveaux axes, et donc à trouver $\mathbf{v}$ de façon à maximiser $\sum_{k \in K_1} r^2(k, \mathbf{v}) + \sum_{n \in N} \eta^2(n, \mathbf{v})$. Cela revient à faire une ACP sur la matrice précédemment décrite.

Exemple Donnons maintenant un exemple. Supposons que nous avons 10 fonds (dont le fonds de requête et neuf fonds de référence), ayant pour attribut quantitatif la *distance* entre les série temporelle de call et pour attribut qualitatif le *millesime* (compris entre 2001 et 2005), la *strategie* (LBO, EVC ou LVC), la *localisation* (Europe ou USA) et la *devise* (EUR, USD ou GBP). La table initiale est donnée en Table 2.1. Nous donnons ensuite la table avec le tableau disjonctif complet des données qualitatives en Table 2.2. On calcule aussi la variance de la *distance*, qui nous donne 7.49. On en déduit la Table 2.3, qui donne la matrice réduite.

	distance	*localisation*	*devise*	*strategie*	*millesime*
req	0	Eur	EUR	LBO	2002
ref_1	2	Eur	EUR	EVC	2004
ref_2	3	USA	USD	LBO	2004
ref_3	4	Eur	EUR	LBO	2005
ref_4	7	USA	GBP	LVC	2002
ref_5	3	Eur	GBP	EVC	2001
ref_6	4	USA	USD	LVC	2003
ref_7	8	USA	GBP	LBO	2002
ref_8	5	USA	USD	LVC	2001
ref_9	1	USA	USD	EVC	2003

Table 2.1 – AFDM : exemple

Il ne reste plus qu'à y appliquer une ACP.

La fonction de distance

Afin de calculer la distance entre deux fonds nous avons utilisé la distance de Frobenius entre les vecteurs rendus par l'AFDM.

Définition 2.2.2 (Distance de Frobenius)
La distance induite par la norme de Frobenius entre deux matrices (dans notre cas deux vecteurs) V_1 et V_2 de même dimension est définie par :

$$\|V_1 - V_2\|_2 = (V_1 - V_2)^\top \cdot (V_1 - V_2)$$

	distance	*localisation*		*devise*			*strategie*			*millesime*				
		Eur	USA	EUR	USD	GBP	LBO	EVC	LVC	2001	2002	2003	2004	2005
req	0	1	0	1	0	0	1	0	0	0	1	0	0	0
ref_1	2	1	0	1	0	0	0	1	0	0	0	0	1	0
ref_2	3	0	1	0	1	0	1	0	0	0	0	0	1	0
ref_3	4	1	0	1	0	0	1	0	0	0	0	0	0	1
ref_4	7	0	1	0	0	1	0	0	1	0	1	0	0	0
ref_5	3	1	0	0	0	1	0	1	0	1	0	0	0	0
ref_6	4	0	1	0	1	0	0	0	1	0	0	1	0	0
ref_7	8	0	1	0	0	1	1	0	0	0	1	0	0	0
ref_8	5	0	1	0	1	0	0	0	1	1	0	0	0	0
ref_9	1	0	1	0	1	0	0	1	0	0	0	1	0	0
Moyenne	3.7	0.4	0.6	0.3	0.4	0.3	0.4	0.3	0.3	0.2	0.3	0.2	0.2	0.1

Table 2.2 – AFDM : tableau disjonctif

	distance	*localisation*		*devise*			*strategie*			*millesime*				
		Eur	USA	EUR	USD	GBP	LBO	EVC	LVC	2001	2002	2003	2004	2005
req	-0.49	0.95	-0.77	1.28	-0.63	-0.55	0.95	-0.55	-0.55	-0.45	1.28	-0.45	-0.45	-0.32
ref_1	-0.23	0.95	-0.77	1.28	-0.63	-0.55	-0.63	1.28	-0.55	-0.45	-0.55	-0.45	1.79	-0.32
ref_2	-0.09	-0.63	0.52	-0.55	0.95	-0.55	0.95	-0.55	-0.55	-0.45	-0.55	-0.45	1.79	-0.32
ref_3	0.04	0.95	-0.77	1.28	-0.63	-0.55	0.95	-0.55	-0.55	-0.45	-0.55	-0.45	-0.45	2.85
ref_4	0.44	-0.63	0.52	-0.55	-0.63	1.28	-0.63	-0.55	1.28	-0.45	1.28	-0.45	-0.45	-0.32
ref_5	-0.09	0.95	-0.77	-0.55	-0.63	1.28	-0.63	1.28	-0.55	1.79	-0.55	-0.45	-0.45	-0.32
ref_6	0.04	-0.63	0.52	-0.55	0.95	-0.55	-0.63	-0.55	1.28	-0.45	-0.55	1.79	-0.45	-0.32
ref_7	0.57	-0.63	0.52	-0.55	-0.63	1.28	0.95	-0.55	-0.55	-0.45	1.28	-0.45	-0.45	-0.32
ref_8	0.17	-0.63	0.52	-0.55	0.95	-0.55	-0.63	-0.55	1.28	1.79	-0.55	-0.45	-0.45	-0.32
ref_9	-0.36	-0.63	0.52	-0.55	0.95	-0.55	-0.63	1.28	-0.55	-0.45	-0.55	1.79	-0.45	-0.32

Table 2.3 – AFDM : matrice réduite

2.2.4 Fonction de distance de séries

Notons que notre méthode de détermination de distances entre fonds nécessite une fonction de distance entre les séries de *calls* (séries temporelles). Cette distance n'a pas encore été décrite.

Pour trouver une fonction de distance, nous avons eu l'opportunité de nous intéresser à plusieurs méthodes, particulièrement la distance euclidienne, utilisée pour la première fois pour les séries temporelles dans [9], Edit distance with Real Penalties (ERP), de [10], EDR tiré de [11]. Plusieurs papiers comparent ces méthodes, en particulier [11, 12].

Nous donnons dans l'Annexe G un rapide résumé des autres méthodes comparées, et nous expliquons ici pourquoi nous avons choisi EDR, mais pour plus de précisions, le lecteur se reportera aux articles sus-cités. Nous décrivons aussi la distance euclidienne dans cette section puisque nous l'avons utilisé pour valider notre modèle. Nous donnons enfin la distance ERP, qui a été utilisée pour la mise en place du clustering (cf. Section 2.3).

Distance euclidienne

Soient deux séries temporelles R et S, tels que : $R = \{(t_1, r_1), (t_2, r_2), \ldots, (t_n, r_n)\}$ et $S = \{(t_1, s_1), (t_2, s_2), \ldots, (t_n, s_n)\}$. Alors, la distance euclidienne entre R et S est :

$$Eucl(R, S) = \sqrt{\sum_{t=0}^{n} \|r_i - s_i\|^2}$$

Le problème principal de cette approche est sa robustesse au décalage des courbes, surtout pour ce qui est de l'axe temporel. En effet, s'il est possible de normaliser les courbes en valeurs, il est impossible de détecter par cette approche les décalages temporels. Un exemple de deux courbes « similaires » est donne en Figure 2.2.

Sur cet exemple, la distance de R à T est de 4, tandis que S, qui est simplement décalée temporellement de 2 par rapport à R, a une distance euclidienne plus grande, 8.

Edit distance with real penalties

Cette distance est métrique, ce qui signifie qu'elle respecte l'inégalité triangulaire. Elle est une adaptation de l'*edit distance* pour les séries réelles. Cependant, comme elle respecte l'inégalité triangulaire, elle est assez vulnérable au bruit (à l'instar de DTW). Pour deux séquences R, S avec $R = (t_i, r_i)_{i \in [\![1,n]\!]}$ et $S = (t_i, s_i)_{i \in [\![1,m]\!]}$ et g

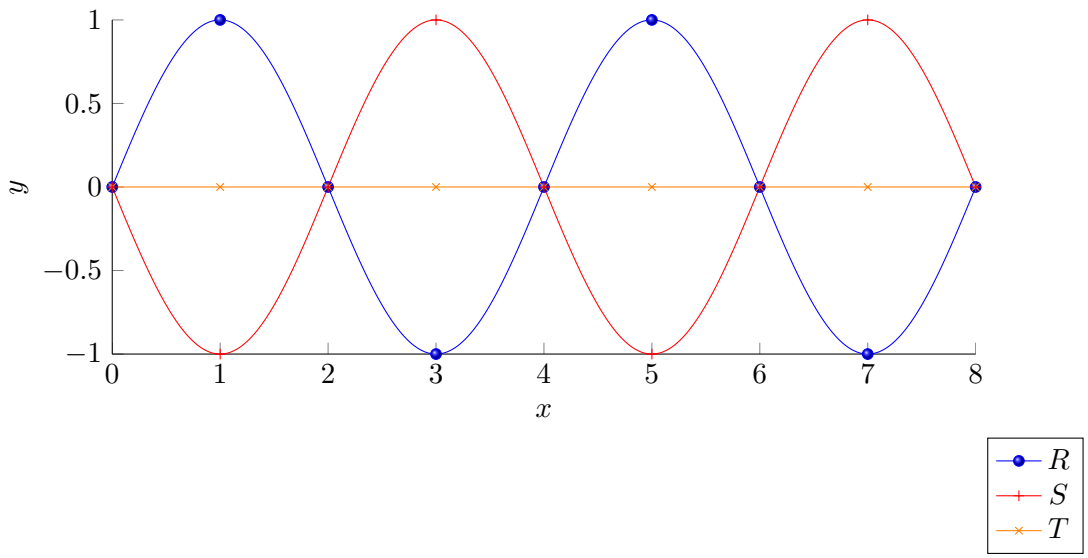


Figure 2.2 – Les défauts de la distance euclidienne

une valeur fixée, la distance ERP se définit récursivement par :

$$ERP(R,S) = \begin{cases} \sum_{i=0}^{n} r_i & \text{si } m = 0 \\ \sum_{i=0}^{m} s_i & \text{si } n = 0 \\ min\left\{ERP\left(R,(s_i)_{i\in[\![2,m]\!]}\right) + |g - s_1|,\right. & \\ \qquad ERP\left((r_i)_{i\in[\![2,n]\!]},S\right) + |g - r_1|, & \text{sinon} \\ \left. ERP\left((r_i)_{i\in[\![2,n]\!]},(s_i)_{i\in[\![2,m]\!]}\right) s_1 - r_1|\right\} & \end{cases}$$

Edit distance for real sequences

Cette distance, malgré son nom, est assez différente de la précédente. En effet, si ces deux méthodes partagent leurs origines (adaptation de l'*edit distance*), cette distance ne respecte pas l'inégalité triangulaire, et est très robuste au bruit. Elle est, selon les tests proposés dans [11, 12] la meilleure pour une grande partie des benchmarks sinon pour la majorité. Pour deux séquences R, S avec $R = (t_i, r_i)_{i\in[\![1,n]\!]}$ et $S = (t_i, s_i)_{i\in[\![1,m]\!]}$, la distance EDR se définit récursivement par :

$$EDR_\varepsilon(R,S) = \begin{cases} n & \text{si } m = 0 \\ m & \text{si } n = 0 \\ min\left\{EDR_\varepsilon\left((r_i)_{i\in[\![2,n]\!]},(s_i)_{i\in[\![2,m]\!]}\right) + \mathbb{1}_{\|s_1-r_1\|\leq\varepsilon},\right. & \\ 1 + EDR_\varepsilon\left((r_i)_{i\in[\![2,n]\!]},S\right), & \text{sinon} \\ \left. 1 + EDR_\varepsilon\left(R,(s_i)_{i\in[\![2,m]\!]}\right)\right\} & \end{cases}$$

où $\mathbb{1}_{\|s_1-r_1\|\leq\varepsilon} = \begin{cases} 1 \text{ si } \|s_1 - r_1\| \leq \varepsilon \\ 0 \text{ sinon} \end{cases}$

Notons qu'il est indiqué dans [11] que la valeur la plus efficace pour ε est $\frac{\sigma}{4}$ où σ est l'écart-type maximal des séries (la même que pour LCS).

2.2.5 Conclusion

Cette méthode comporte de nombreux avantages, elle est très paramétrable et permet une sélection très précise des fonds à utiliser pour la régression linéaire multiple, puisque le choix est fait en temps réel, selon le fonds à prédire. Comme on le verra, cela n'est pas le cas de la classification non supervisée.

2.3 Classification non supervisée

Le but de la classification non supervisée, ou *clustering* en anglais est de diviser l'ensemble des données en plusieurs ensembles, et plus particulièrement de le partitionner, de façon à ce que chaque élément de l'ensemble des données soit dans une et une seule partition.

2.3.1 Définitions

Afin d'avoir de plus amples explications sur le *clustering* en général, le lecteur pourra se reporter à [13]. Rappelons que les trois principaux types d'apprentissage non supervisé sont l'apprentissage par **k-moyennes**, les réseaux ou **cartes de Kohonen** et **l'apprentissage hiérarchique**. Ces trois apprentissages automatiques connaissent tous deux variantes, la classification non supervisée « dure » ou *hard clustering* et la classification non supervisée « floue », ou *fuzzy clustering*. La classification non supervisée dure implique qu'un objet appartient à une et une seule classe. Dans la classification non supervisée floue au contraire, un objet peut appartenir à plusieurs classes, avec une appartenance plus ou moins forte selon sa proximité avec chacune. Les définitions des deux premières méthodes sont données en annexe H.

2.3.2 Apprentissage hiérarchique

La classification hiérarchique n'établit pas un partitionnement de l'espace, mais un arbre de partitionnement appelé dendrogramme, dont la racine est constitué d'une unique classe, et dont les feuilles sont constituées de chaque élément de l'ensemble des données. Il existe deux façon de créer le dendrogramme.

La première est de commencer par considérer toutes les données comme une catégorie et de diviser cette catégorie en deux sous-catégories, puis de diviser ces deux sous-catégories en deux etc., jusqu'à ce que chaque classe ne contienne plus qu'un élement. La méthode est de subdiviser de façon à avoir deux ensembles aussi *éloignés* que possible. Cette méthode est dite **descendante**. La seconde est de considérer chaque élélement comme une catégorie et d'agréger les ensembles les plus *proches* par deux jusqu'à obtenir une seule catégorie. Cette méthode est dite **ascendante**.

Notons que ces deux approches nécessitent de définir une distance entre en-

sembles, que nous noterons par la suite Δ. Dans la pratique, il existe plusieurs distances possibles : la distance entre centres de gravité, la distance minimale entre les éléments de chaque ensemble, la méthode de Ward (*cf.* [14]). Idéalement, cette distance doit également avoir une propriété supplémentaire (encore que cette propriété n'est pas absolument nécessaire) : elle doit être une distance ultramétrique (nous donnons la définition de ce concept ci-dessous).

Définition 2.3.1 (Distance ultramétrique)
Pour un ensemble E, une distance ultrmétrique de E est une fonction $\Delta : E^2 \to \mathbb{R}^+$ telle que :

$$\forall (x, y) \in E^2 \Delta(x, y) = \Delta(y, x)$$

$$\forall (x, y) \in E^2, \Delta(x, y) = 0 \Leftrightarrow x = y$$

$$\forall (x, y, z) \in E^3, \Delta(x, z) \leq max(\Delta(x, y), \Delta(y, z))$$

Notons que dans notre cas, E n'est pas l'ensemble des données, mais l'ensemble des parties de l'ensemble des données. Le dernier point implique donc en particulier que la distance entre deux parties de l'ensemble des données est plus petite que la plus grande distance entre les deux sous-ensemble qui le composent. Nous donnons maintenant l'algorithme de classification hiérarchique ascendante, qui pourra aisément être tranformé en algorithme descendant.

Algorithme 2 Classification hiérarchique ascendante

```
1 : function HIERARCH(dat)▷ La fonction arbre(valeur, sag, sad) renvoie un arbre ayant
    pour valeur de nœud value et pour sous-arbres sag et sad.
2 :                                  ▷ Δ est une distance ultramétrique entre parties de dat.
3 :     E ← ∅
4 :     for all d ∈ dat do
5 :         E ← E ∪ arbre(0, d, ∅)
6 :     end for
7 :     while |E| > 1 do
8 :         (e1, e2) = argmin_{(e,e')∈E²}{Δ(e, e')}
9 :         a = arbre(Δ(e1, e2), e1, e2)
10 :        E ← E \ {e1, e2} ∪ {a}
11 :    end while
12 :    return E
13 : end function
```

Donnons maintenant un exemple à une dimension. Nous considérons 7 points de $\mathbb{R}$ ayant pour valeur : $a = 34; b = 37; c = 24; d = 26; e = 10; f = 14; g = 1$. Nous utilisions pour Δ le minimum de la distance entre les éléments. Nous obtenons alors, après agglomération, le dendrogramme Figure 2.3.

Une distance utilisée régulièrement est l'indice de Ward. Notons que cette distance n'est pas une ultramétrique.

Définition 2.3.2 (Indice de Ward)
Dans le cadre de la classification hiérarchique par agglomération, soient trois ensembles h_1, h_2 et h_3. On note $h_4 = h_1 \cup h_2$. Alors, la distance au sens de l'indice de

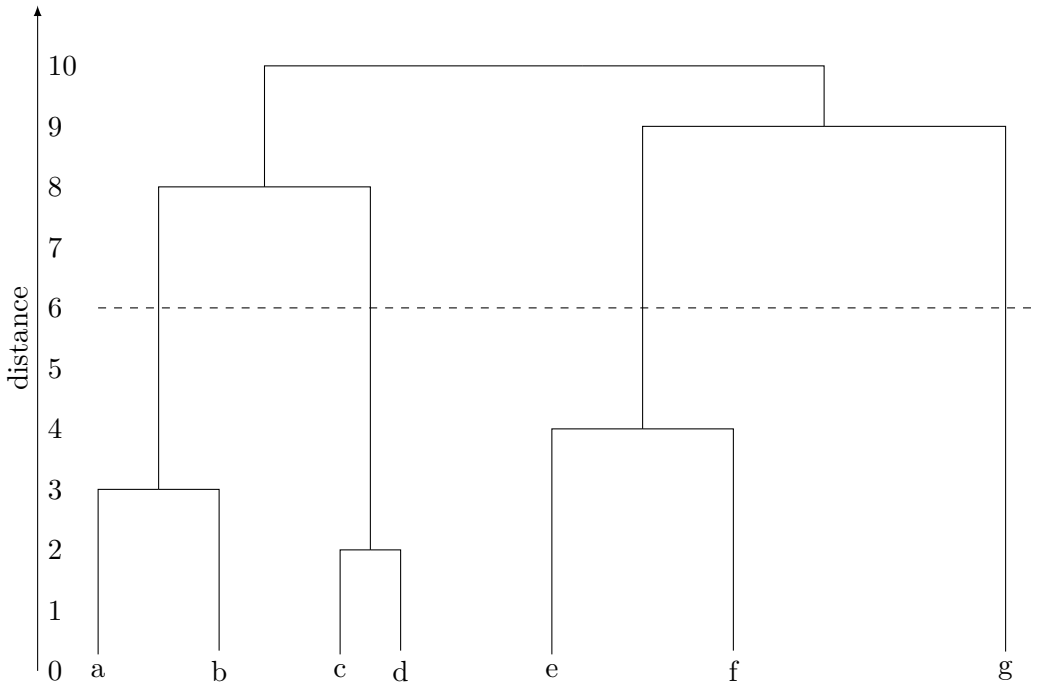


Figure 2.3 – Exemple de dendrogramme

Ward δ_W *entre* $h1$ *et* $h4$ *est défini par :*

$$\begin{aligned}\delta_W(h_1,h_4) =& \frac{|h_1|+|h_2|}{|h_1|+|h_2|+|h_3|}\delta_W(h_1,h_2) + \frac{|h_1|+|h_3|}{|h_1|+|h_2|+|h_3|}\delta_W(h_1,h_3) \\ &- \frac{|h_1|}{|h_1|+|h_2|+|h_3|}\delta_W(h_2,h_3)\end{aligned} \tag{2.1}$$

Une fois le dendrogramme obtenu, une coupe pour une distance quelconque donne une partition de l'ensemble des données. Par exemple sur notre exemple (Figure 2.3), nous réalisons une coupe à 6. Cette coupe nous donne 4 classes, une contenant $\{a, b\}$, une contenant $\{c, d\}$, une contenant $\{e, f\}$ et une contenant $\{g\}$. Le but est en fait de couper l'ensemble au moment où l'écart entre le premier nœud au-dessus de la coupe et le premier nœud au-dessous est important.

Il existe également des façons pour trouver automatiquement une coupe pertinente pour le dendrogramme. On peut notamment citer le *Dynamic Tree Cut*, explicité dans [15].

2.3.3 Adaptation du modèle

Notre problème a une particularité. Nos données sont en effet constituées de séries temporelles de longueur potentiellement différentes. Nous ne pouvons considérer ces données comme des vecteurs comprenant les différents *cash flows* au cours du temps ; cela sous-entendrait que les valeurs de chaque périodes ne seraient comparées qu'aux valeurs de la même période. Cela reviendrait donc à utiliser une distance euclidienne. De plus en ce qui concerne la longueur de ces vecteurs, elles peuvent différer d'un

fonds à l'autre. Cette difficulté peut cependant être surmontée en utilisant un âge interne.

Il faut donc concevoir un algorithme particulier, et prendre en compte les décalages temporels. Pour cela, nous avons décidé d'utiliser une classification hiérarchique avec la distance ERP comme distance entre 2 séries temporelles (parce qu'elle est métrique) et l'indice de Ward.

Par rapport à ses concurrentes, la classification hiérarchique a l'avantage d'aboutir à une solution globale, et d'éviter de calculer des moyennes qui peuvent avoir des effets pervers. Elle permet de plus d'obtenir un arbre et d'avoir une classification plus ou moins fine en fonction du niveau de la coupe. Nous avons décidé de ne pas utiliser le Dynamic Tree Cut, car il présentait des difficultés d'implémentation trop importantes.

2.3.4 Conclusion

Nous avons choisi de sélectionner la méthode de classification hiérarchique, qui nous a paru correspondre le mieucx à notre besoin, et à un nombre déléments assez important. Une fois cette méthode choisie, nous l'avons adaptée à notre problématique. Lors du choix de cette méthode, il nous a été demandé de créer un service supplémentaire pour pouvoir accéder à une classification non supervisée des fonds de Pevara depuis le logiciel.

2.4 Bilan : choisir un modèle

Voici la ligne que nous avons donc décidé de suivre : sélectionner les k-plus-proches-voisins, ce qui permettra de ne prendre en compte que les courbes les plus pertinentes. Ensuite, faire une régression linéaire multiple sur ces k-plus-proches-voisins afin d'obtenir les paramètres de FrontAnalytics.

Un résumé de notre module est donné en Figure 2.4.

Expliquons cette figure. Chaque numéro correspond à une étape :

0. Le client a une séquence de calls, ici représentée par une courbe ainsi que des paramètres qualitatifs.
1. Il saisit ces données sur FrontAnalytics.
2. Les données sont envoyées à Pevara
3. L'AFDM permet de réduire le nombre de dimensions.
4. Les données sont placés dans ce repère.
5. On ne garde que les k-plus-proches-voisins.
6. Nous déduisons les paramètres de FrontInvest par une régression linéaire multiple sur les k-plus-proches-voisins.

Comme nous l'avons déjà mentionné, seules les données nominales ou qualitatives seront alors connues, ainsi que les appels, connus en général par contrat. Notons que pour remplir la matrice de poids, nous rechercherons l'importance que nous donnons à chaque paramètre en analysant les résultats de la classification hiérarchique.

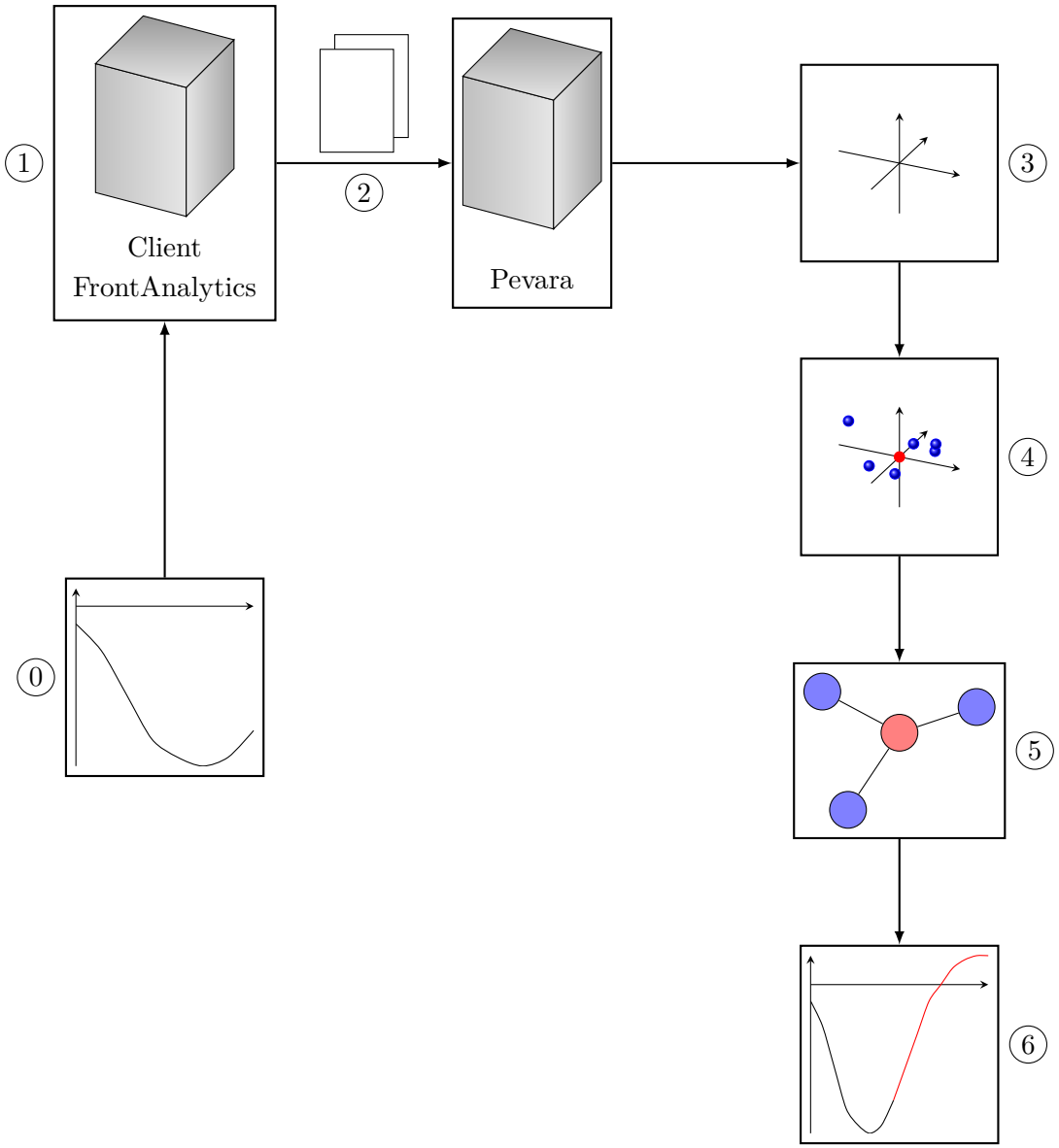


Figure 2.4 – Fonctionnement de notre module, option 1

En ce qui concerne la validation, nous prenons en compte que la distance EDR entre la courbe prévue et la courbe réelle. Nous avons également implémenté une fonction de distance euclidienne pour la comparaison.

Nous avons également mis au point une seconde méthode, qui utilisera la classification hiérarchique. Le résumé en est donné en Figure 2.5.

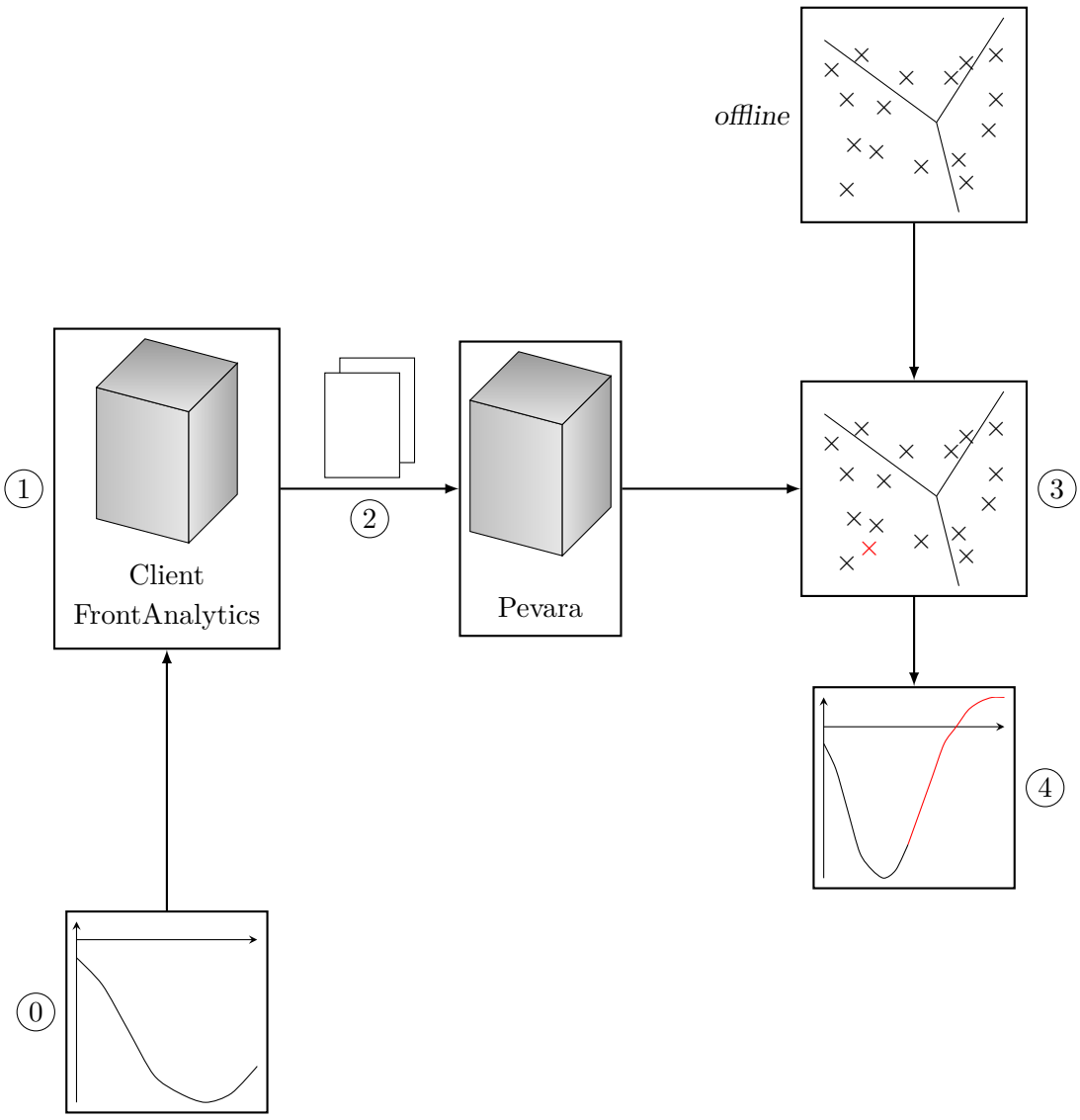


Figure 2.5 – Fonctionnement de notre module, option 2

Donnons une description de cette méthode. Elle est identique à la méthode précédente, à cela près que les étapes 3 à 5 sont remplacées par l'étape 3. Les clusters sont calculés *offline*, à chaque mise à jour de la base de Pevara. Une fois ces clusters calculés, un fonds peut être placé, de façon entière ou floue à un ou plusieurs clusters. Nous procédons par la suite à la même régression linéaire multiple que dans la première méthode.

Notons que cette méthode est plus rapide que la précédente, puisque les véritables calculs sont fait lors de la mise à jour de la base de Pevara, que l'on peut considérer comme *offline*.

Chapitre 3

Implémentation

Ce chapitre contient des précisions un peu plus techniques sur la façon dont l'implémentation se déroule. Elle contient notamment quelques références techniques aux librairies utilisées.

3.1 Généralités

En ce qui concerne la façon de travailler, nous avons travaillé « en marge » du projet. Le module que nous avons produit n'a pas tout de suite été utilisé pour la production. Cependant, il se trouve que même pour la production, le module n'a pas posé de problème, puisque les données ne sont pas très importantes. Ainsi, l'ensemble a été directement implémenté sur Pevara et en pur Java, afin d'éviter les erreurs de *Virtual Machine*.

Nous avons porté notre projet sur Maven. Maven est un logiciel libre d'Apache et implémente une norme établie par cette même compagnie Project Object Model (POM). La dernière version de Maven est Maven2 ; c'est la version la plus utilisée, et c'est aussi celle utilisée à eFront. Plus d'informations sont données en Annexe I.1. Notons que notre projet a été intégré dans Pevara, qui utilise Spring pour l'injection de référence. Plus d'informations peuvent être trouvées dans l'Annexe I.3.

Notre projet a été intégré à Pevara en tant que web service. Ce web service peut être appelé depuis un autre programme. Le principe et les détails techniques sont exposés dans l'Annexe I.4

3.2 Modèle linéaire

3.2.1 Les données à prédire

Les calls sont, selon [1] connus à l'avance par contrat entre le LP et le GP. Nous décidons donc de ne prévoir que les distributions. En prenant en compte le modèle linéaire que nous avions décrit, il ne nous reste plus que les paramètres à prévoir.

3.2.2 Prise en compte de l'âge interne du fonds

Nous avons investigué la méthode donnée dans [1, 16], qui donne l'âge relatif (ou interne) du fonds comme étant la moyenne de deux âges relatifs, le *drawdown age* et le *repayment age*. Devant la multitude des problèmes posés par cette méthode et en particulier l'impossibilité, malgré nos recherche de donner une approximation de la NAV à un instant donné, nous avons décidé d'adopter une approche linéaire. Ainsi, nous avons observé sur les fonds liquidés que le centre de gravité des calls d'un fonds est situé en général à quelque 15% de la durée de vie totale du fonds. Notons que nous avons opté pour le centre de gravité car il nous semblait être un meilleur indicateur que par exemple le dernier call, qui peut en fait être tardif et ne représenter qu'un allongement pour un fonds particulier ou une opération mineure.

3.2.3 Résolution de l'équation et calcul de Θ

Rappelons que nous cherchons à déduire des k-plus-proches-voisins les paramètres du modèle. Pour cela, nous souhaitons résoudre l'éqution en $Theta'$

$$D = C'\Theta'$$

de façon optimale pour tous ces voisins, cette équation pouvant se traduire par :

$$\begin{pmatrix} D_1 \\ D_2 \\ \vdots \\ D_k \end{pmatrix} = \begin{pmatrix} C'_1 \\ C'_2 \\ \vdots \\ C'_k \end{pmatrix} \cdot \Theta'$$

Notons

$$D \leftarrow \begin{pmatrix} D_1 \\ D_2 \\ \vdots \\ D_k \end{pmatrix} \text{ et } C' \leftarrow \begin{pmatrix} C'_1 \\ C'_2 \\ \vdots \\ C'_k \end{pmatrix}$$

Pour plus de précision, le lecteur se reportera à la section 2.1.3.

Programmation linéaire

Une librairie purement Java permet de résoudre ce genre de problèmes, Apache Commons Maths. Cette librairie a de plus pour avantage d'être purement Java. Après avoir implémenté une classe qui traduit le problème de régression linéaire multiple en programmation linéaire, nous avons implémenté une classe test (J-Unit), qui réalise un test unitaire sur un problème de régression. Le test réussit, mais une fois le programme testé sur les véritables données de notre problème, une erreur `aucune solution realisable` apparaît et nous empêche d'utiliser cette solution.

Non-Negative Least Squares

Il nous a hélas été impossible de trouver une librairie qui permette de résoudre ce problème et dont la licence permette de l'inclure dans une application commerciale (la seule librairie que nous ayons trouvée est sous licence GPL). Nous avons posté une question sur Stack Overflow, mais sans succès. Nous avons donc décidé d'implémenter l'algorithme nous-même.

Nous donnons l'algorithme de résolution ci-dessous. Les notations de l'algorithme correspondent aux notions suivantes : $\mathbf{C}$ la matrice des calls (C'), telle que définie ci-dessus, le vecteur $\mathbf{d}$, le vecteur des distributions, et le vecteur $\mathbf{x}$, vecteur que nous devons déterminer par notre régression.

Certaines notations supplémentaires doivent être définies. La matrice $\mathbf{A^P}$ est une matrice carrée ayant $|P|$ colonnes, et qui a pour valeurs les vecteurs colonne de $\mathbf{A}$ ayant pour indices les entiers contenus dans P. De même, le vecteur $\mathbf{s^P}$ est le vecteur de dimension $|P|$ ayant pour valeurs celles de $\mathbf{s}$, mais uniquement pour les indices appartenant à P. Enfin, la fonction `Update` retire de R les entiers $j : x_j = 0$, ajoute à P ces mêmes indices et réciproquement pour les indices pour lesquels $x_j \neq 0$.

Algorithme 3 Non-negative Least Square

function NNLS($\mathbf{C}, \mathbf{d}, tolerance$)
  $P \leftarrow \emptyset$
  $R \leftarrow \{1, 2 \ldots, m\}$
  $\mathbf{x} \leftarrow \mathbf{0}$
  $\mathbf{w} \leftarrow \mathbf{C}^\top(\mathbf{d} - \mathbf{C}\mathbf{x})$
  while $R \neq \emptyset \wedge max_{i \in R}\{w_i\} > 0$ **do**
    $j \leftarrow argmax_{i \in R}\{w_i\}$
    $P \leftarrow P \cup \{j\}$
    $R \leftarrow R \setminus \{j\}$
    $\mathbf{s^P} \leftarrow \left[\left(\mathbf{C^P}\right)^\top \mathbf{C^P}\right]^{-1} \left(\mathbf{C^P}\right)^\top \mathbf{d}$
    while $min\left(\mathbf{s^P}\right) \leq 0$ **do**
      $\alpha \leftarrow -min_{i \in P : \mathbf{x_i} < \mathbf{0}}\{\frac{x_i}{d_i - x_i}\}$
      $\mathbf{x} \leftarrow \mathbf{x} + \alpha\mathbf{s} - \mathbf{x}$
      Update(P)
      Update(R)
      $\mathbf{s^P} \leftarrow \left[\left(\mathbf{C^P}\right)^\top \mathbf{C^P}\right]^{-1} \left(\mathbf{C^P}\right)^\top \mathbf{d}$
      $\mathbf{s^R} \leftarrow \mathbf{0}$
    end while
    $\mathbf{x} \leftarrow \mathbf{s}$
    $\mathbf{w} \leftarrow \mathbf{C}^\top(\mathbf{d} - \mathbf{C}\mathbf{x})$
  end while
  return x
end function

3.3 k plus proches voisins

Pour coder les k-plus-proches-voisins, et afin de préparer au mieux l'intégration, nous avons décidé d'utiliser pour paramètre d'entrée une matrice du format de celle qui résulte de l'AFDM. Dans l'espace construit par l'AFDM, nous utilisons la distance induite norme de Frobenius, comme nous l'avons décrit dans le chapitre précédent.

3.4 Analyses factorielles

Dans cette section, nous décrivons la manière dont nous avons décidé d'implémenter notre Analyse en Composantes Principales et l'Analyse Factorielle des Données Mixtes.

3.4.1 Besoin

Décrivons tout d'abord nos besoins. Comme nous l'avons mentionné précédemment, dans l'Annexe F.1.5 il est nécessaire de pouvoir donner plus ou moins de poids à une variable selon sa fiabilité. Nous avons cherché des librairies correspondant à ce besoin, qui intègrent donc une matrice de poids, mais qui correspondent aussi aux restrictions légales de licences (*cf.* 1.3.3). Il nous a malheureusement été impossible de trouver aucune librairie qui corresponde à ces besoins. De plus, nous cherchons en priorité une librairie purement Java (non native d'un autre langage) afin d'éviter les problèmes liés à la transcription de types sans Virtual Machine (Java est « safe », ce qui n'est pas le cas d'un langage qui n'utilise pas la VM). Après une comparaison des benchmarks faite sur https ://code.google.com, nous décidons de choisir la librairie EJML (Efficient Java Matrix Library) (*cf.* La page de réfrence) pour la manipulation des matrices.

3.4.2 Implémentation

En ce qui concerne l'ACP, nous avons inclus les deux matrices de poids et la projection des données initiales sur les axes. Nous n'avons cependant pas inclus les outils d'analyse classique. Nous l'avons testée sur un exemple trouvé sur Internet, sur le site de l'Institut de Mathématiques de Luminy. Pour ce qui est de l'AFDM, nous avons ajouté quelques méthodes pour le traitement des données qualitatives, mais nous avons globalement repris les méthodes de l'ACP.

Pour l'AFDM, nous avons largement utilisé le code de l'ACP, à une méthode supplémentaire, permettant d'ajuster les variables qualitatives près. Notons que contrairement à ce qui est proposé dans [8], nous avons décidé de faire l'ajustement des données en amont, centralisation comprise et de ne laisser que l'analyse à notre classe ACP.

3.5 Edit Distance for Real sequences

3.5.1 Normalisation des données

Les courbes ont été « normalisées » afin de pouvoir les comparer plus facilement avec l'EDR. Nous les avons normalisées en temps (une valeur par semestre) et en valeur (nous avons réduit les données de façon que $\sum calls = 1$). Il s'agit de cash flows cumulés, ainsi, pour calculer les cash flows d'un trimestre, nous avons simplement récupéré la valeur la plus proche dans le passé.

3.5.2 Implémentation

Lorsque nous avons implémenté l'EDR, nous sommes arrivés à un problème. En effet, l'exécution d'un test sur des données de tailles modestes (moins de 10 valeurs dans chaque série temporelle) prend plusieurs secondes, et réalise de l'ordre de 10^6 appels récursifs. Cela est du au fait que la structure de l'EDR permet une approche de programmation dynamique (pour plus d'information sur la programmation dynamique, voir [17], chapitre 15). L'idée principale de la programmation dynamique est de « garder en mémoire » les solutions des différents sous-problèmes afin de gagner du temps d'exécution et de ne pas les recalculer à chaque fois. Cependant, il nous faut procéder par étapes. [17] explique les quatre étapes à suivre pour construire un algorithme de programmation dynamique :

1. Définir la structure du problème, décrire sa structure et montrer qu'il consiste à optimiser plusieurs sous-problèmes.
2. Exhiber une version récursive naïve du problème.
3. A partir des points précédents, exhiber un algorithme récursif du problème, mais qui utilise une table afin de ne pas recalculer les mêmes valeurs plusieurs fois.
4. *(Facultatif)* Transformer l'algorithme récursif en algorithme itératif (approche bottom-up).

Pour ce qui est de définir la structure du problème, la définition même de l'EDR nous permet d'y répondre. En effet

$$EDR_\varepsilon(R,S) = \begin{cases} n & \text{si } m = 0 \\ m & \text{si } n = 0 \\ min\left\{EDR_\varepsilon\left((r_i)_{i\in[\![2,n]\!]}, (s_i)_{i\in[\![2,m]\!]}\right) + \mathbb{1}_{\|s_1-r_1\|\leq\varepsilon}, \right. & \\ 1 + EDR_\varepsilon\left((r_i)_{i\in[\![2,n]\!]}, S\right), & \text{sinon} \\ \left. 1 + EDR_\varepsilon\left(R, (s_i)_{i\in[\![2,m]\!]}\right)\right\} & \end{cases}$$

où $\mathbb{1}_{\|s_1-r_1\|\leq\varepsilon} = \begin{cases} 1 \text{ si } \|s_1 - r_1\| \leq \varepsilon \\ 0 \text{ sinon} \end{cases}$. Cela signifie que calculer l'EDR de deux séries temporelles revient à calculer les EDR des sous séries correspondantes entre elles et avec chacune des séries.

La seconde étape est elle aussi évidente si l'on prend en considération la définition de l'EDR.

Il reste à déterminer l'algorithme de programmation dynamique : étant donné le faible volume de données, de l'ordre de $|R| \cdot |S|$, on négligera la dernière étape, et on conservera la mémoïsation (version récursive remplissant un tableau au fur et à mesure). L'algorithme est donné dans l'Algorithme 4.

Algorithme 4 Calcul de l'EDR

```
1 : Int table[ ][ ]
2 : function EDR(Series R, Series S, double ε)                ▷ Compute EDR(R, S)
3 :     if |R| = 0  or |S| = 0 then
4 :         return Σ_{i|r_i≠0} r_i + Σ_{i|s_i≠0} s_i
5 :     end if
6 :     for i ≤ |R|, j ≤ |S| do
7 :         table[i][j] = −1
8 :     end for
9 :     if |R[0] − S[0]| ≤ ε then
10 :        dist ← 0
11 :    else
12 :        dist ← 1
13 :    end if
14 :    return min{EDR(1, 1, R, S, ε) + dist,
                EDR(0, 1, R, S, ε) + 1,
                EDR(1, 0, R, S, ε) + 1}
15 : end function
16 : function RecEDR(Int k, Int l, Series R, Series S, double ε)
                                  ▷ Compute EDR^sub ((r_i)_{i≥k},(s_i)_{i≥l})
17 :    if |R| < k  or |S| < l then
18 :        table[k][l] = 0
19 :        return Σ_{i>k|r_i≠0} r_i + Σ_{i>l|s_i≠0} s_i
20 :    end if
21 :    Int dist ← 0
22 :    if table[k][l]= −1 then
23 :        if |R[k] − S[l]| ≤ ε then
24 :            dist ← 0
25 :        else
26 :            dist ← 1
27 :        end if
28 :        table[k][l]← min{EDR(k + 1, l + 1, R, S, ε) + dist,
                EDR(k, l + 1, R, S, ε) + 1,
                EDR(k + 1, l, R, S, ε) + 1}
29 :    end if
30 :    return table[k][l]
31 : end function
```

3.6 Classification non supervisée

Avant l'implémentation, nous avons testé les différentes distances possibles, ultramétriques ou non. Pour ces tests, nous avons utilisé R. Cela nous a de plus permis de visualiser simplement les dendrogrammes. Nous avons ensuite comparé les différents dendrogrammes et avons choisi celui qui distribuaot le mieux les classes et les fonds.

3.7 Intégration des modules

Maintenant que nous avons construit toutes les briques nécessaires à la création de notre programme, il nous faut penser à la façon de les coordonner afin de le construire. Cela va nous poser de nouvelles problématiques.

3.7.1 Paramètres des séries

La première problématique est de définir quels paramètres nous allons garder et comment nous allons les représenter. Nous avons sélectionné la devise, la localisation, le millésime et la stratégie qui sont à la fois présents dans FrontInvest et renseignés dans Pevara.

Pour le millésime, nous le gardons tel quel, mais nous rassemblons les millésimes les moins pourvus (ainsi, nous rassemblons les fonds `avant 1992`, puis les fonds `1993-1994`, les fonds `1995-1996` et les fonds `2009-2010`, les autres sont laissés tels quels).

Pour les stratégies, nous rassemblons `Buyout Large` et `Mega Cap` en une seule catégorie, Les `Funds of funds` dans une unique catégorie, Le `Venture Debt` est ajouté au `Venture Capital`. Les catégories `Natural Resources`, `Oil Gas`, `Other`, `Private Equity`, `Real Estate` et `Timberland` sont agrégées en `Others`.

La répartition des localisations ne nous laisse pas beaucoup de choix (la répartition étant très inégale). Nous mettons de côté `l'Europe de l'Ouest et du Nord`, qui ont beaucoup d'indices proches (IDH etc.) à savoir `Europe du Nord`, `Danemark`, `Espagne`, `Suede`, `Grande Bretagne`, `Pays-Bas`, `Italie`, `Norvege`, `Allemagne`, `Finlande`, `Europe de l'Ouest` et `Suisse`. Nous laissons de côté la `France`, qui a assez de fonds pour être considérée comme un élément à part entière. Nous agrégeons les `Etats-Unis` et le `Canada` et nous mettons toutes les autres dans une catégorie `Autres`.

Enfin, nous répartissons les devises en quatre parties : `USD`, `EUR`, `GBP` et `Autres`.

Nous obtenons alors les tableaux donnés dans la Table 3.1. Le but est d'avoir au moins quatre catégories, et que chaque catégorie ait au moins 80 données. Nous essayons aussi d'agréger les données logiquement. Comme nous le voyons, il arrive que les contraintes que nous nous sommes fixées ne soient pas respectées.

Notons que nous réservons un sort particulier aux données classées dans `Autres`. En effet, classer les données dans `Autres` donne une signification à cette catégorie

Buyout	182
Buyout Large Cap	258
Buyout Medium Cap	312
Buyout Small Cap	417
Distressed Debt	121
Fund Of Funds	74
Mezzanine Capital	134
Autres	93
Venture Capital	114
Venture Capital Balanced	231
Venture Capital Early Stage	418
Venture Capital Late Stage	212

(a) Adaptation des stratégies

1979-1992	185
1993-1994	101
1995-1996	86
1997	104
1998	118
1999	139
2000	200
2001	141
2002	80
2003	113
2004	134
2005	204
2006	270
2007	297
2008	251
2009-2010	143

(b) Adaptation des millésimes

USD	1973
EUR	480
GBP	60
Autres	53

(c) Adaptation des localisations

Europe	444
Amerique du Nord	1701
Autres	286
France	135

(d) Adaptation des devises

Table 3.1 – Adaptation des données qualitatives

qui ne devrait pas en avoir. Par exemple, nous classerons un fonds asiatique proche d'un fonds africain alors que potentiellement, les fonds de ces deux continents n'ont aucun point en commun. Notons qu'il en est de même pour les fonds dont le millésime est antérieur à 1992 : quoique ce critère soit plus objectif, un fonds de 1980 n'a potentiellement aucun rapport avec un fonds de 1992.

3.7.2 Récupération des fonds

Nous avons aussi chargé les données de Pevara. Pour cela, nous ne sommes pas passés par la base de donnée dans un premier temps (celle-ci nécessite en effet beaucoup de mise en place). Nous avons tout d'abord utilisé des fichiers contenant les bases dont nous avions besoin sous la forme de TSV (*Tabulation Standard Value*), qui représentent les données sous forme de texte séparé par des tabulations. Notons que nous n'avons eu aucune modification à faire sur la base, ce qui nous a permis d'utiliser ce type de format. Dans un second temps, lors de l'intégration de notre module, nous avons mis en place une connexion à la base de données.

3.7.3 Revue de code

Lors des quelques séances de revue de code que nous avons eues, nous avons modifié plusieurs parties de notre programme et avons ainsi abouti à de meilleurs temps de calcul.

Une des séquences les plus importants de notre programme est le chargement des fonds. Dans un premier temps, nous chargions ces fonds à chaque test, ce qui les allongeait énormément les temps de calcul. Nous avons donc décidé de les garder en mémoire pour une même batterie de test et avons ainsi pu gagner énormément de temps.

Les différentes sessions de revue de code nous ont également permis d'utiliser les bon design patterns, et de rendre notre code plus clair et plus souple. Nous avons également écrit des tests unitaires pour chaque fonction en utilisant le framework JUnit.

Chapitre 4

Évaluations

4.1 Démarche

Nous avons implémenté les deux versions possibles du module, que nous avons testées chacune avec différents paramètres. Cela nous a permis de comparer les meilleures versions de nos deux programmes, de choisir le meilleur et de le comparer à un modèle préexistant dans l'entreprise, qui a été implémenté par Olivier Dellenbach (le PDG). Nous donnons plus de précisions dans la troisième section de ce chapitre. Pour chaque évaluation, nous utilisons trois types de distance : la distance EDR, la distance ERP et la distance euclidienne.

4.2 Régression utilisant les k-plus-proches-voisins

Pour les k-plus-proches-voisins, nous avons testé plusieurs valeurs de k allant de 10 à 25, comme le montre la Table 4.1. Nous donnons également les temps d'exécution de chaque partie du programme en Table 4.2.

k	EDR	ERP	Euclidienne
10	77.5	$2.85 \cdot 10^8$	$6.93 \cdot 10^8$
15	68.9	$2.77 \cdot 10^8$	$6.93 \cdot 10^8$
20	70.3	$2.79 \cdot 10^8$	$7.00 \cdot 10^8$
25	71.1	$2.72 \cdot 10^8$	$7.13 \cdot 10^8$

Table 4.1 – Évaluation de la régression couplée aux k-plus-proches-voisins

Nous pouvons faire plusieurs remarques à propos de ces résultats. Tout d'abord, on voit que les distances sont relativement corrélées les unes aux autres, malgré queslsues dissimilarités pour $k = 25$. Ensuite, en ce qui concerne les temps, on remarque que mis à part les temps de régression, ils sont à peu près constants, ce qui s'explique par leur relative petitesse.

k\Module	EDR	AFDM	k-p-p-v	Régression
10	2790	415	3	2349
15	2618	390	3	3768
20	2600	387	3	5346
25	2760	419	3	7589

Table 4.2 – Temps d'exécution de chaque module de la régression linéaire multiple couplée aux k-plus-proches-voisins, en ms

4.3 Régression utilisant la classification non supervisée.

Pour la classification hiérarchique, nous avons utilisé plusieurs niveaux de granularité, et un nombre différent de classes à chaque fois. Les évaluatons sont données en table 4.3. Les temps d'exécution sont donnés en Table 4.4

Hauteur de coupe	Nombre de classes	EDR	ERP	Euclidienne
0.04	8	83	$3.31 \cdot 10^8$	$7.51 \cdot 10^8$
0.01	16	78	$2.95 \cdot 10^8$	$7.15 \cdot 10^8$
0.0058	20	77	$2.92 \cdot 10^8$	$6.95 \cdot 10^8$

Table 4.3 – Évaluation de la régression couplée à la classification non supervisée

Module\Hauteur de coupe	0.04	0.01	0.0058
Régression	36361	33376	30876
Clustering	683628	718011	709054

Table 4.4 – Temps d'exécution de chaque module de la régression linéaire multiple couplée à la classification non supervisée, en ms

Comme on peut le voir : d'une part, le programme obtient de meilleurs résultats au fur et à mesure que la classification devient plus fine, ce qui correspond à nos attentes (plus la classification est fine, et plus la prévision faite sera juste). D'autre part, les temps de calcul diminuent. Cela est dû au fait que nous avons imposé à notre régression de prendre au moins 100 fonds de façon « floue », c'est-à-dire de pondérer les différents fonds en fonction de leur proximité en termes de vintage, localisation etc. avec le fonds à prévoir. Ainsi, plus les classes sont importantes, plus on dépassera la centaine de fonds et plus la régression prendra du temps.

On remarque également que les performances restent inférieures à celles des k-plus proches-voisins. Enfin, notons que pour une hauteur de coupe de 0.04, la plus grande classe contient plus de 1000 fonds, pour une hauteur de 0.01, elle en contient à peu près 400 et pour une hauteur de 0.0058, elle en contient quelque 300.

4.4 Un élément de comparaison : le module CFF V2

Suite à une discussion avec le Preésident Directeur Général, Olivier Dellenbach, nous avons décidé de comparer notre méthode avec une méthode implémentée par ses soins, nommée ici cash flow forecasting V2. Nous allons tout d'abord expliquer la méthode d'Olivier Dellenbach.

4.4.1 La méthode de cash flow forecasting V2

La méthode est assez simple. L'idée principale de cette méthode est de moyenner les séries de calls et les séries de distributions sur les séries de la même stratégie après avoir normalisé en temps et en valeur. Les deux stratégies prises en compte sont `Venture Capital` et `Buyout`.

Notons que l'un des avantages de cette méthode est qu'elle n'a pas besoin de courbe d'appel, et qu'elle la prévoit. Notre méthode au contraire nécessite une courbe d'appel.

4.4.2 Comparaison

Afin de comparer les deux méthodes, nous utilisons le modèle de cash flow forecasting V2.0 pour inférer les calls dans notre modèle. Cela nous permet de les mettre sur un pied d'égalité et de leur donner les mêmes données en entrée. Nous en vaons profité pour vérifier la validité de notre méthode sur une autre table. Le résultat apparaît sur la Tabble 4.5. Comme on peut le constater, notre méthode overperforme assez largement la « CFF V2 ». Nous avons fait notre comparaison sur la distance ERP, qui est la plus facilement représentable.

k	Notre méthode	CFF V2
10	$4.57 \cdot 10^7$	$4.71 \cdot 10^7$
15	$4.14 \cdot 10^7$	$4.71 \cdot 10^7$
20	$4.11 \cdot 10^7$	$4.71 \cdot 10^7$
25	$3.92 \cdot 10^7$	$4.71 \cdot 10^7$
30	$3.85 \cdot 10^7$	$4.71 \cdot 10^7$
35	$3.91 \cdot 10^7$	$4.71 \cdot 10^7$
40	$3.91 \cdot 10^7$	$4.71 \cdot 10^7$

Table 4.5 – Comparaison de notre méthode et de CFF V2

Notons que nous constatons que la meilleure configuration pour notre méthode est de prendre un k égal à 30, ce qui diffère de ce que nous avions constaté dans le cadre des tests faits sur Pevara. De plus, les résultats sont globalement moins bons (et ce pour tous les k) que ceux obenus sur Pevara. Cela peut s'expliquer par le fait que nous prenons en compte une courbe d'appels inférée par une moyenne sur les

fonds, ce qui peut bruiter les résultats.

4.5 Bilan et axes d'amélioration

Comme on peut le voir, la démarche adoptée a porté ses fruits. Cependant, il reste plusieurs axes d'amélioration qui eussent pu être suivis. Tout d'abord, du point de vue de la théorie et en ce qui concerne la régression, nous avons vu que nous avions utilisé une simplification très forte, qui peut amener à une performance moindre. S'abstraire de cette régression amènerait à un temps de calcul bien supérieur (le nombre de paramètre est mis au carré), mais permettrait sans doute une meilleure précision.

Un second axe est l'optimisation au sein de la régression. Comme nous l'avons dit, nous avons utilisé l'algorithme NNLS, mais le module de FrontInvest demande uniquement six paramètres. Nous avons donc essayé d'ajouter cette contrainte à l'algorithme NNLS, mais sans succès. En effet, dans ce cas, celui-ci risque de ne plus finir. Même dans le cas où il finit, il ne donne pas de meilleurs résultats qu'une réduction effectuée *a posteriori*, une fois les paramètres obtenus.

Ce problème peut être résolu de deux façons. Tout d'abord, on pourrait utiliser la norme L_1. Comme on l'a vu, cela reviendrait à utiliser de la programmation dynamique. Cela nécessiterait de trouver une librairie native qui soit capable de résoudre un problème comprenant des booléens pour variables. Une seconde solution serait d'utiliser le livre de Hanson et Lawson, [3] pour trouver un algorithme de résolution des moindres carrés contraints, qui permettent de ne garder que six paramètres non nuls. Comme pour la simplification précédente, cela impliquerait sans doute un algorithme ayant une complexité supérieure et donc un temps de calcul plus important, mais de meilleurs résultats.

Chapitre 5

La vie en entreprise

Ce chapitre décrira l'environnement de travail dans lequel j'ai évolué, ainsi que la façon dont s'est déroulé le stage d'un point de vue humain.

5.1 Contexte

Mon stage s'est déroulé dans un contexte très particulier. Ceci pour plusieurs raisons.

Tout d'abord, j'ai changé de maître de stage pendant son déroulement. En effet, après deux semaines, le programmeur qui devait encadrer mon projet (Frédéric Paulin) est parti. Son manager (et le mien par la même occasion), Geordy Landrein, est donc devenu mon maître de stage. Le fait qu'il a amené deux nouvelles particularités. Tout d'abord, il n'avait pas de connaissances théoriques sur le sujet, et que son emploi du temps chargé (il est responsable de l'équipe de Recherche et Développement Java) ne lui a pas laissé beaucoup de temps pour se former. Néanmoins et au fur et à mesure que je lui fournissais mes rapports, il les a lus et a pu me suivre. Seconde difficulté, il vit à Londres d'où il travaille, ne passant que deux jours par quinzaine à Paris. Cela a bien entendu compliqué la communication, et nous avons aussi dû utiliser régulièrement des moyens de communication tels Skype ou plus simplement la téléphonie. Hélas, ces moyens ne sont pas toujours de bonne qualité, et les communications ont parfois été un peu difficiles.

Par ailleurs, Frédéric n'a pas été le seul à quitter l'entreprise. Denis, cadre expérimenté qui m'a souvent aidé sur les sujets techniques, a lui aussi quitté l'entreprise lors de mon stage. Mes autres collègues travaillant sur le même problème que moi sont partis à Londres, où ils ont rejoint Geordy. Tant et si bien que je me suis retrouvé seul pour mon projet. J'étais cependant entouré par trois autres développeurs de l'équipe Java, travaillant sur un autre projet. Parmi ceux-ci il y avait deux prestataires, dont un n'a pas renouvelé son contrat et nous a quitté début février. Un heureux évènement en a amené un autre à prendre un congé paternité. Je me suis donc retrouvé seul avec le dernier prestataire. Afin de donner une meilleure idée de la fluctuation des effectifs, un graphique en est donné en Figure 5.1

Enfin, le 8 octobre, l'entreprise a racheté AnalytX, une entreprise dans le même

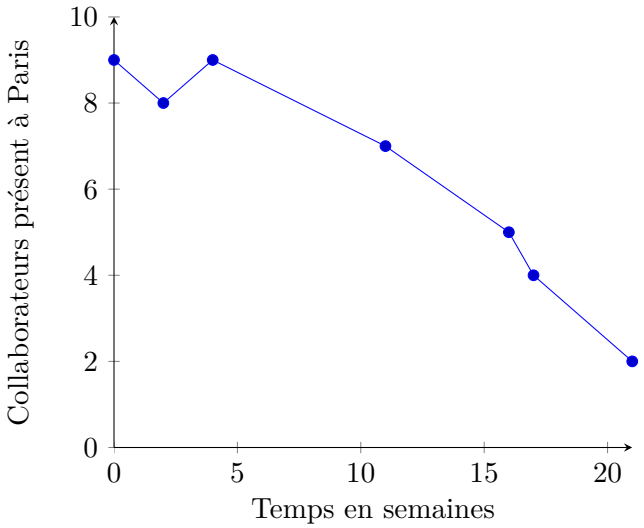


Figure 5.1 – Évolution des effectifs de l'équipe Java présente à Paris

secteur, qui possédait un pôle de développement à Belgrade. Ce rachat a été assorti d'un vaste plan de recrutement local, ce qui a eu de nombreuses conséquences sur mon travail. D'abord, cela a alourdi l'emploi du temps de Geordy, qui était chargé du recrutement sur place. Ensuite, pour encadrer les nouvelles recrues, les développeurs de mon équipe ont été à leur tour dépéchés sur place.

5.2 Une entreprise internationale

Comme je l'ai expliqué dans la première partie, eFront est une entreprise de dimension mondiale qui compte ses clients parmi les fonds les plus grands au monde. L'entreprise possède des bureaux dans le monde entier : Londres, New York, Belgrade... J'ai effectué la totalité de mon stage à Paris sans être amené à travailler avec des développeurs localisés à l'étranger, en revanche j'ai dû composer avec un maître de stage situé à Londres où l'ensemble de l'équipe l'a progressivement rejoint. Cet éloignement a singulièrement compliqué ma tâche Un peu en raison du décalage horaire, même s'il n'est que d'une heure entre Paris et Londres, mais surtout en raison de mon isolement géographique peu propice au travail d'équipe. Aussi performants qu'ils soient, des outils de communication comme Skype ou le partage de bureau sont loin de remplacer le contact physique,et les échanges sont moins spontanés et moins efficaces que la présence d'une personne physique à ses côtés. Cela a d'autant plus d'impact que l'employé a besoin de ses collègues. Cette sensation d'isolement a été particulièrement forte lorsqu'il m'a fallu intégrer mes modules au logiciel de l'entreprise.

Mon stage m'a également permis de mesurer à quel point la hiérarchie est fortement sollicitée dans une entreprise de grande dimension. Amenés à cumuler les responsabilités, certains cadres sont très contraints par le temps au point d'en devenir difficilement joignables. J'ai été d'autant plus confronté à ce problème que mon projet était en quelque sorte en marge de ce qui était réalisé dans l'entreprise. À certaines périodes, il m'a même été difficile d'obtenir de mes collaborateurs des rendez-

vous dont j'avais pourtant grand besoin. Je n'avais que peu ou pas d'échéances, à l'exception de celles que je me suis imposées et de l'échéance finale, où le projet devait être terminé. Cela explique pourquoi dans certaines périodes en dépit de mon insistance, j'ai regretté de ne pouvoir rencontrer certains collaborateurs. Mon stage est de plus arrivé à un moment de métamorphose particulièrement fort pour l'entreprise, ce qui a encore accentué cet effet.

5.3 Le fonctionnement d'une équipe de développeurs

L'équipe Java est divisée en deux « sous-équipes ». L'une s'occupe de Pevara, le produit sur lequel j'ai travaillé et comprend trois personnes : Geordy, qui le gérait, Xavier, l'architecte fonctionnel, et Olivier, qui était ingénieur de développement. L'autre sous-équipe s'occupe d'un projet pour un grand client. Deux autres collègues travaillaient pour les deux équipes à la fois. L'un se chargeait de sa mise en production, l'autre agissait en tant que *senior developper* et apportait sa contribution à l'équipe Java dans son intégralité. Les moyens humains consacrés au projet étaient plus conséquents que ceux de notre sous-équipe : manager à plein temps, développeurs et deux prestataires.

Puisque ce projet n'était développé que pour un seul client, Un dialogue constant s'était naturellement instauré avec celui-ci, dialogue auquel était partie prenante l'équipe située en Inde qui avait contribué à ce projet. Cela m'a permis de découvrir plusieurs choses. Tout d'abord, le rapport de force qui peut s'établir entre le fournisseur et son client. Le client a, comme on peut s'en douter, un certain pouvoir, une certaine influence sur le fournisseur (c'est lui qui paie). Cependant, le fournisseur n'accepte pas toutes ses demandes, et a lui aussi un certain pouvoir sur son client. Il faut notamment que le cahier des charges soit précis et que le client s'engage à fournir les données nécessaires au bon déroulement du projet.

Cela a aussi été l'occasion de constater que d'une équipe ou d'un projet à l'autre, les méthodes de travail changent, parfois même de façon significative. Deux équipes ont souvent des approches, des habitudes et des compétences différents. Gérer l'idiosyncrasie de chacun fait aussi partie du rôle de Manager.

Enfin, la fréquentation de ces équipes m'a permis de constater qu'un équipe a besoin de renouveau. Ce nouveau souffle nécessaire peut notamment être amené par l'arrivée de nouveaux acteurs. L'équipe, par exemple, a vécu assez peu de changements avant l'arrivée des prestatires (qui sont arrivés en même temps que moi). Ils ont permis de mettre à jour ce que l'on pourrait appeler le capital de connaissances techniques de l'équipe. Ils avaient une bonne connaissance de certains sujets, des dernières innovations techniques pour de nombreuses problématiques. Cela a permis de faire évoluer et le projet sur lequel ils ont travaillé et les connaissances générales de l'équipe. L'importance de la dynamique d'une équipe est une leçon dont je me souviendrai. La nécessité de telles impulsions est d'autant plus importante lorsqu'il s'agit de domaines en constante évolution, comme c'est le cas pour l'informatique.

5.4 Déroulement du stage

Pour les raisons citées ci-dessus, le déroulement du stage a été inhabituel. Afin de tenir mon responsable au courant de mes avancées, j'ai rédigé un rapport par semaine pendant seize semaines. Après ces seize semaines, l'aspect théorique de mon projet ayant été mené à bien, il ne m'a pas paru nécessaire de poursuivre, d'autant que la version finale de ce rapport était alors en cours d'élaboration.

Pour ce qui est de mon projet, je découperais mon stage en trois parties. Dans la première partie qui a duré à peu près un mois et demi, j'ai beaucoup travaillé sur la théorie. J'ai étudié le problème qui m'était donné et j'ai constitué un modèle. J'ai ensuite isolément implémenté ce modèle sur ma machine, ce qui m'a pris deux mois. Enfin, dans un troisième temps, j'ai intégré mon projet à Pevara, ce qui m'a pris un mois et demi.

Ces trois étapes ont été instructives sur différents points. La première m'a permis d'approfondir mes connaissances théoriques sur le *machine learning* et les statistiques. La seconde m'a permis de me mettre à niveau en Java, où ma seule expérience se résumait à quelques cours théoriques. Enfin, la troisième partie du stage m'a permis de rencontrer de nouveaux concepts et de nouvelles technologies.

Lors de ce stage, j'ai régulièrement posé des questions à mes collègues, et j'ai eu la chance de toujours obtenir des réponses. C'est un point très positif de mon stage. J'ai eu la chance de rencontrer des gens que j'estime hors du commun, et cela tout au long de mon stage. Hors du commun pour plusieurs raisons. Tout d'abord, d'un point de vue technique. Les collègues qui m'entouraient connaissent de nombreuses technologies, et sont capables de résoudre des problèmes de tous types rapidement. Ils ont aussi eu beaucoup de bienveillance à mon égard et ont souvent pris le temps de répondre à mes questions de jeune développeur junior, peu initié aux différentes technologies utilisées dans l'entreprise.

5.5 Bonnes pratiques

Je vais décrire dans cette section un aspect très important pour un développeur ; il s'agit d'un point que j'ai compris lors de ce projet. En programmation, il existe de nombreuses façon de résoudre un problème. Certaines étapes pourraientent même être considérées par certains comme « dispensables ». Ainsi, il est théoriquement possible de se passer de tests unitaires... mais cela a généralement de très lourdes conséquences par la suite, puisque lors d'une modification ultérieure du code, rien ne prouve que ladite modification n'a pas cassé le code.

Ce sont toutes ces habitudes que l'on rassemble sous le nom de « bonnes pratiques ». J'ai pu lors de mon stage constater leurs bénéfices, ainsi que les dégâts qu'engendrent leur non respect. La réécriture d'une partie de mon propre code ou encore celle du second projet de l'équipe Java en utilisant ces pratiques ont permis d'améliorer significativement leurs performances.

En sus des réflexes que m'ont enseignés mes collègues lors de mes sessions de revue de code (encore une bonne pratique), on m'a conseillé la lecture de Effective

Java ([18]), qui m'a beaucoup aidé, m'a appris de nouveaux concepts Java, mais aussi d'autres « bonnes pratiques » et des pièges à éviter. Un autre exemple de bonne pratique est la documentation. J'ai fait l'expérience de devoir lire de grandes quantités de code non commenté ni documenté et force m'a été de constater que cela est extrêmement difficile. Une dernière bonne pratique, indiquée dans l'*item* 47 d'Effective Java ([18]), est l'utilisation de librairies et *frameworks* standards. Cela permet de trouver plus facilement la documentation relative à la librairie. C'est également gage d'une certaine qualité assurée par la communauté des utilisateurs.

5.6 Bilan

Le but de ce stage était de créer un modèle de cash-flow forecasting, de l'intégrer à Pevara. Cela a été réussi, et même au-delà des espérances puisque j'ai pu implémenter deux services, opérationnels et utilisables. Ce stage était aussi l'occasion de m'améliorer en Java, et de découvrir de nouvelles technologies ; sur ce point aussi c'est une réussite. J'ai eu l'occasion d'observer la vie en entreprise, et de voir quels risques présentent l'isolement ou l'obsolescence technique. J'y ai rencontré des gens tous différents, mais qui m'ont tous beaucoup apporté. J'y ai compris, finalement, ce que j'attendais d'un emploi de développeur.

Du point de vue de mon projet, celui-ci est fini. Certes, il s'agit d'une version qui pourrait être améliorée, mais elle est déjà fonctionnelle. Il reste à implémenter l'interface graphique qui permettra son utilisation depuis FrontInvest.

Chapitre 6

Conclusion et perspectives

Ce stage a été une réussite sur plusieurs plans. D'un point de vue humain, il a été l'occasion de rencontrer des développeurs de talent, de travailler au sein d'une équipe efficace. Cela m'a aussi permis de voir une entreprise en mutation et m'a obligé à m'adapter à un changement important au sein d'une équipe. Cela a enfin été l'occasion de travailler avec des collaborateurs à l'étranger, dont mon maître de stage et manager, ce qui a été pour moi une expérience assez insolite.

D'un point de vue méthodologique, le stage m'a permis d'aborder deux aspects. Mon sujet m'a amené, comme en atteste ce rapport, à utiliser les méthodes classiques de recherche. Néanmoins, cette recherche n'est pas restée uniquement théorique et a abouti à un produit fini et exploitable, et cela a impliqué que ce que je produisait devait être utilisable ensuite en prodution. Cela signifie aussi que les données utilisées étaient de véritables données, donc nécessairement bruitées, et non un jeu de données propres élaboré intentionnellement à l'usage exclusif de ma problématique.

L'avantage méthododlogique a donc été double. J'ai dû poursuivre l'effort que j'avais amorcé lors de la rédaction de mon mémoire de Master, et respecter une méthode scientifique rigoureuse. D'un autre côté, j'ai dû me plier aux exigences de l'entreprise, utiliser des librairies créées par d'autres que moi, faire un code parfois moins efficace mais plus lisible afin qu'il puisse être repris par quelqu'un d'autre. Ces concessions faites par la recherche à l'entreprise m'ont permis de voir mon projet comme plus « réaliste » et cela m'a enthousiasmé au point que je recherche une thèse CIFRE pour la suite.

Enfin, d'un point de vue technique, ce stage m'a permis de m'améliorer en Java, où j'ai dû apprendre à programmer de façon pratique et efficace. Cela m'a aussi permis de rencontrer différentes librairies et *frameworks* qui m'ont été indispensables au cours du stage, et le resteront sans aucun doute pour longtemps. Cela m'a permis de rencontrer de nouveaux concepts, et si certains resteront cantonnés à cette expérience, d'autres figurent désormais en tête de mon CV.

Je terminerai ce rapport en disant que la problématique du stage a été traitée conformément au besoin, et qu'un service a bel et bien été ajouté au produit de l'entreprise. Il reste bien sûr quelques axes d'amélioration, mais le succès de cette expérience m'a persuadé de poursuivre dans cette voie et à rechercher une thèse CIFRE, fonctionnant sur le même principe que ce stage.

Annexes

Annexe A

Organigrammes et chiffres clés de l'entreprise

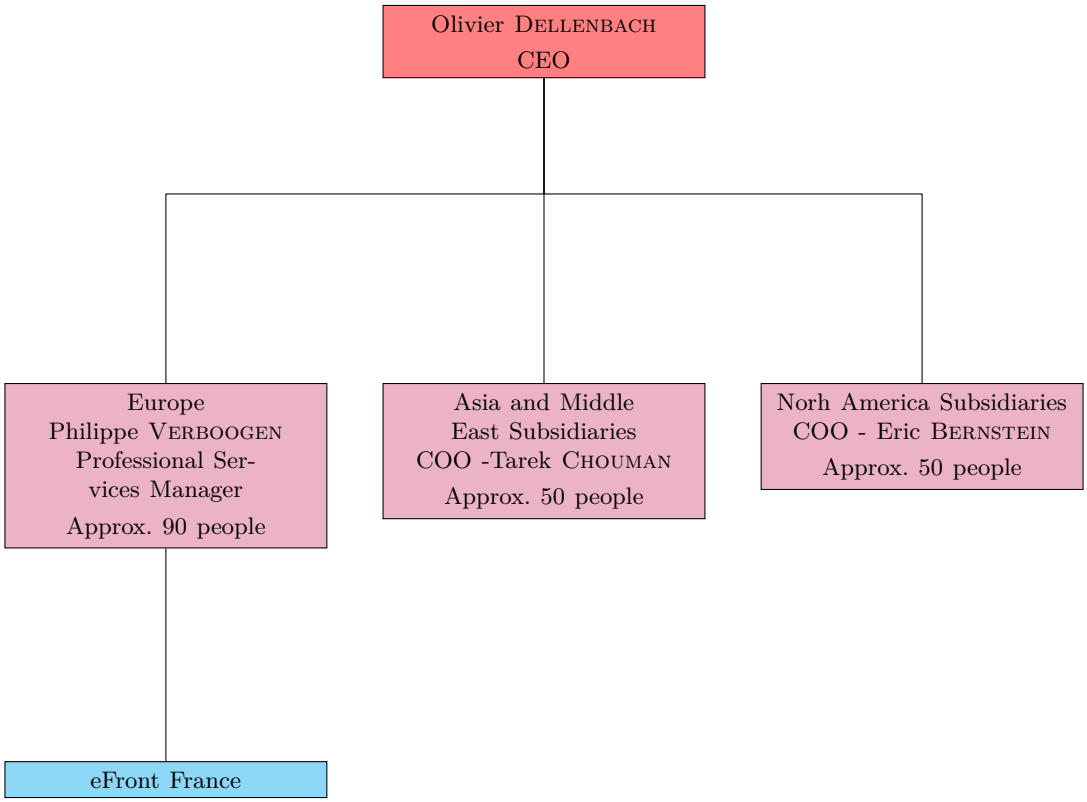


Figure A.1 – Organisation d'eFront

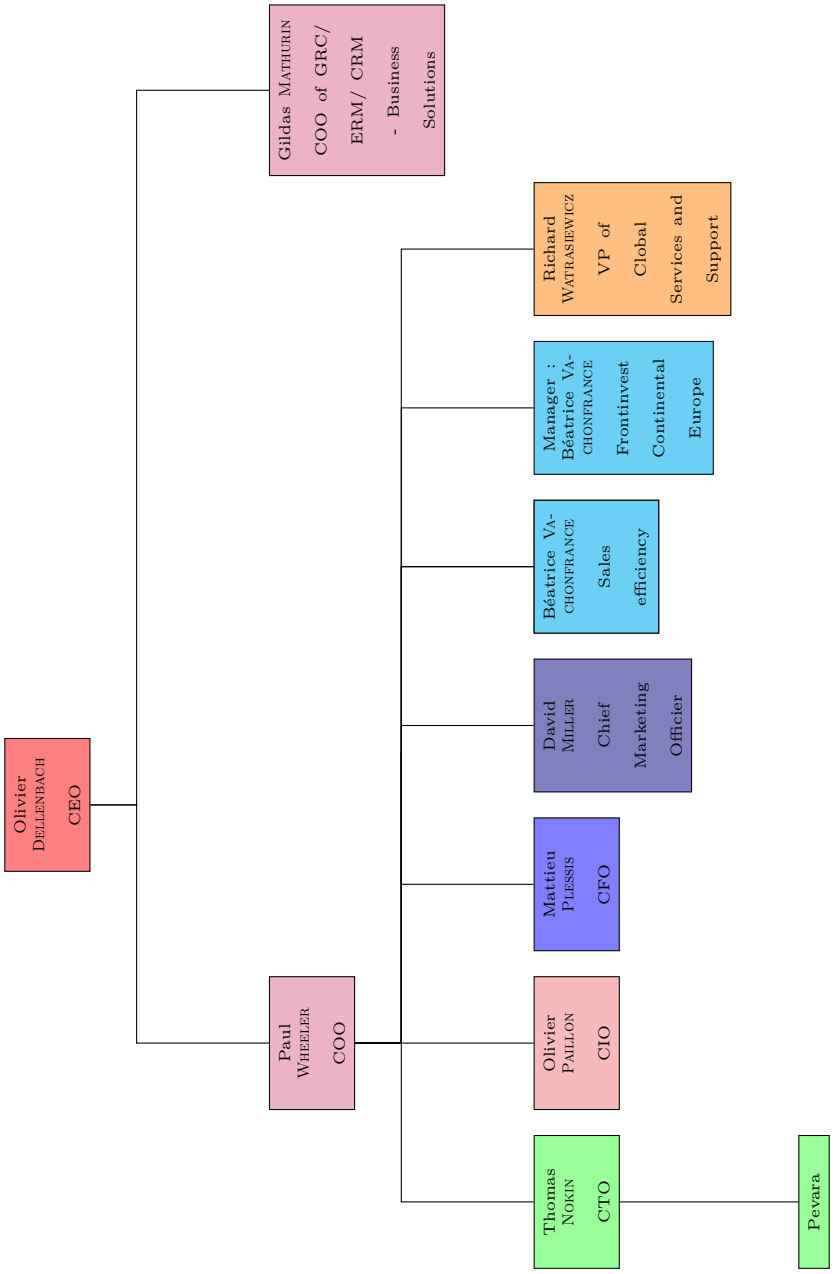


Figure A.2 – Organisation d'eFront France

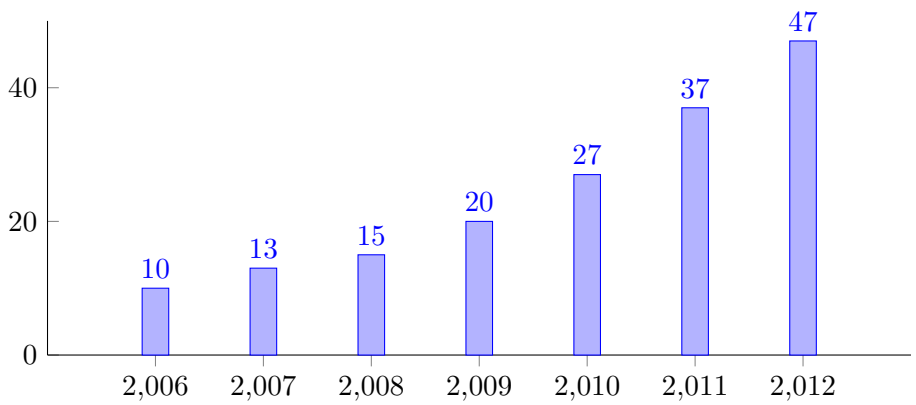


Figure A.3 – Évolution du chiffre d'affaire d'eFront entre 2006 et 2012

Annexe B

J-Curves tirées des données de Pevara

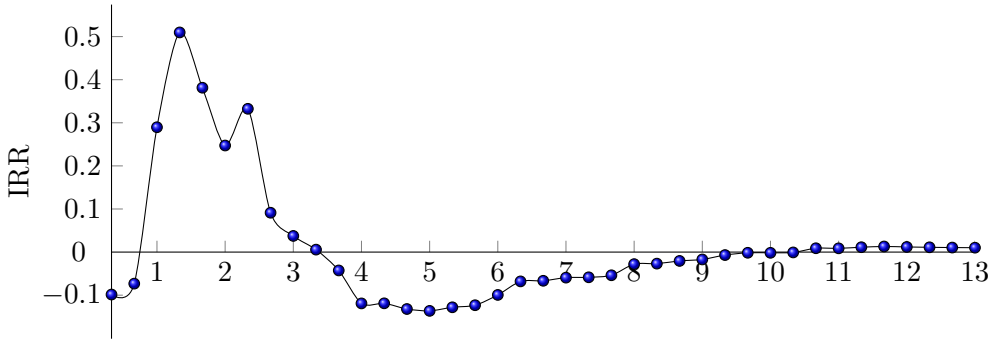


Figure B.1 – Une J-curve utilisant l'IRR

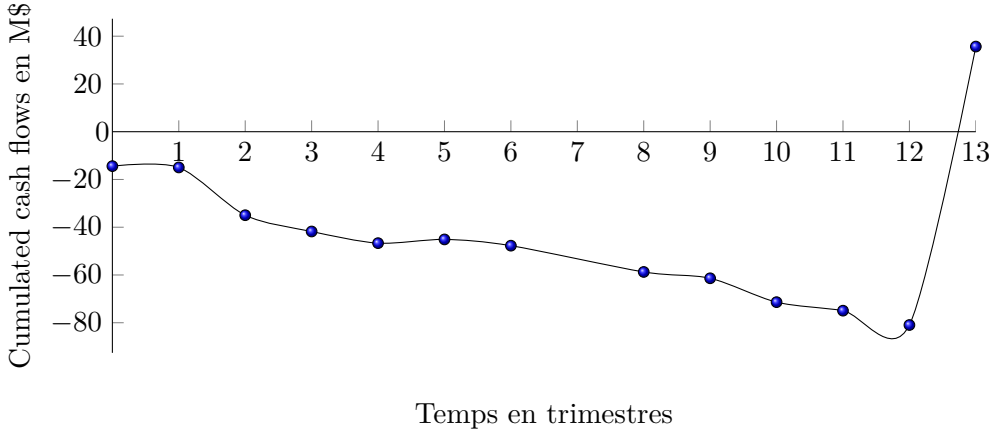


Figure B.2 – Une J-curve extraite des données recueillies

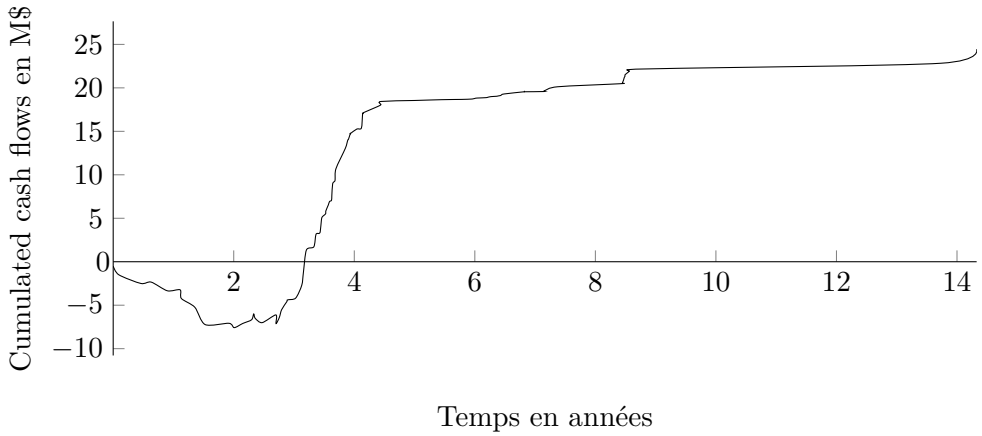


Figure B.3 – Une autre J-curve extraite des données recueillies

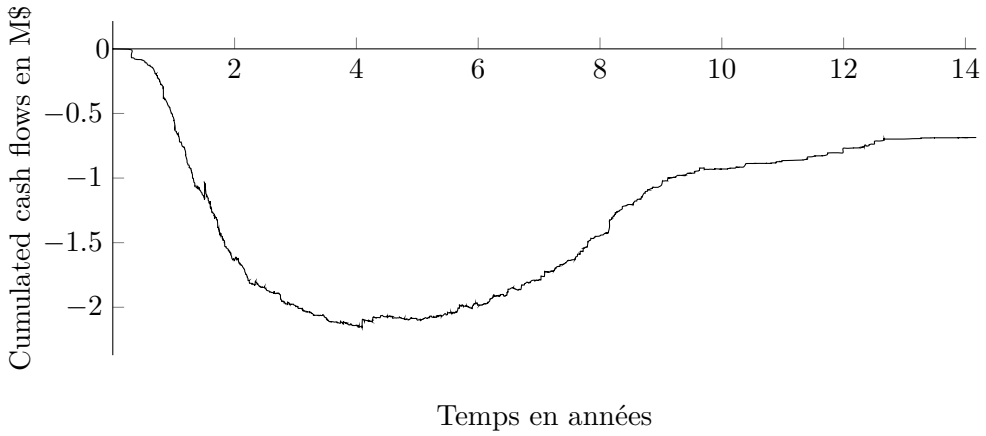


Figure B.4 – Cette J-curve est issue de la moyenne des Cumulated cash flows sur les fonds démarrés en 1999

Annexe C

Les données des fonds, classées selon différents critères

Balanced	295
Communications	9
Consumer	16
Financial	5
Healthcare	2
Industrial	8
Life Sciences	13
Manufacturing	1
Media	2
Oil Gas	1
Technology	24
Transportation	3
Null	2187

(a) Number of funds by Industry

Buyout	182
Buyout Large Cap	190
Buyout Medium Cap	312
Buyout Mega Cap	68
Buyout Small Cap	417
Distressed Debt	121
Fund Of Funds	44
Fund Of Funds Primary	9
Fund Of Funds Secondaries	21
Infrastructure	24
Mezzanine Capital	134
Natural Resources	5
Oil Gas	13
Other	1
Private Equity	21
Real Estate	24
Timberland	5
Venture Capital	110
Venture Capital Balanced	231
Venture Capital Early Stage	418
Venture Capital Late Stage	212
Venture Capital Venture Debt	4

(b) Number of funds by Strategy

Vintage	Number
1979	1
1980	5
1981	3
1982	1
1983	8
1984	11
1985	9
1986	13
1987	15
1988	21
1989	26
1990	31
1991	14
1992	27
1993	43
1994	58
1995	41
1996	45
1997	104
1998	118
1999	139
2000	200
2001	141
2002	80
2003	113
2004	134
2005	204
2006	270
2007	297
2008	251
2009	83
2010	60

(c) Number of funds by Vintage

AUD	8
CAD	6
CHF	1
DKK	5
EUR	480
GBP	60
ILS	5
JPY	18
KRW	1
SEK	9
USD	1973

(d) Number of funds by Currency

Active	1873
Liquidated	693

(e) Number of funds by Status

Europe	92
Japon	12
Danemark	5
Afrique	2
Espagne	12
Australie	8
Maroc	1
Thailande	1
France	135
Bresil	4
Russie	2
Suède	12
Israël	58
Grande Bretagne	52
Afrique du sud	3
Mexique	3
USA	1692
Pays-Bas	7
Italie	14
Norvège	2
Turquie	2
Europe de l'est	13
Inde	11
Allemagne	21
Monténégro	18
Finlande	12
Chine	35
Asie	71
Autres (Moyen orient/Afrique)	8
Amérique du Sud	7
Reste du monde	15
Canada	9
Corée	2
Hongrie	1
Europe de l'Ouest	214
Pologne	8
Suisse	1
Hong-Kong	1

(f) Number of funds by Geography

Table C.1 – Distribution des fonds

Annexe D

Les licences et leurs droits

Nom	Inclusion de la license	Virale	Droits réservés explicitement	Indication des changements	Sources ouvertes
GPL	×	✓	×	✓	✓
LGPL	×	✓ [1]	×	✓	✓ [2]
MIT	×	×	×	×	×
BSD	×	×	×	×	×
Modified BSD	×	×	✓	×	×
Apache	✓	×	✓	✓	×

Table D.1 – Les différentes licences

Toutes ces licences sont libres. Une license virale applique sa license à tout programme utilisant une source sous cette license. Elle est donc peu recommandée pour un logiciel commercial. Toutes ces licenses exigent l'inclusion des copyright, mais pas le texte de la licence, à l'exception de la licence Apache-2.0. Toutes les licenses sous-tendent que les droits sont réservés, mais seules certaines le spécifient explicitement. Les licences libres permettent de modifier le code fourni. Néanmoins quelques unes imposent que les modifications soient explicitement spécifiés.

1. possibilité d'utiliser une LGPL dans un code sans rendre celui-ci LGPL, mais toute modification du code LGPL reste LGPL
2. uniquement de la partie LGPL

Annexe E

Modèles d'autorégression

E.1 Apprentissage statistique et séries temporelles

Dans cette section, nous décrivons un modèle assez classique d'autorégression appelé ARMA/ARIMA. Nous décrivons succinctement ce modèle ci-dessous. Notons que toutes les références nécessaires à la description du modèle se trouvent dans le chapitre 7 de [19]. Ces modèles utilisent des méta-paramètres (p, d, q) et les paramètres appris sur la courbe $((\phi_i)$ et $(\theta_i))$ .Le modèle ARMA est en fait le composé de deux autres modèles, le modèle AR et le modèle MA.

E.1.1 Les séries stationnaires

Avant de définir les modèles que nous avons envisagés, nous devons donner la définition d'une caractéristique importante à propos des séries que nous analysons, celle de la stationnarité. La stationnarité (de second ordre) est définie en Définition E.1.1.

Définition E.1.1 (Stationnarité)
Une série temporelle $(X_i)_{i\in[\![0,n]\!]}$ est dite stationnaire de second ordre si et seulement si :

- $E[X_t] = \mu, \forall i \in [\![0, n]\!]$
- $Var[X_t] = \sigma^2, \forall i \in [\![0, n]\!]$
- $Cov[X_t, X_{t-k}] = f(k) = \rho_k, \forall t \in [\![0, n]\!], \forall k \in [\![1, t]\!]$

Une courbe stationnaire est donc simplement un ensemble de points distribués selon une même loi autour d'une droite horizontale. La Figure E.1 en donne un exemple.

Le modèle AR

Dans cette sous-section, nous décrivons le modèle AR, qui signifie *AutoRegression*. Il dépend d'un méta paramètre p, qui représente la profondeur historique de l'observation.

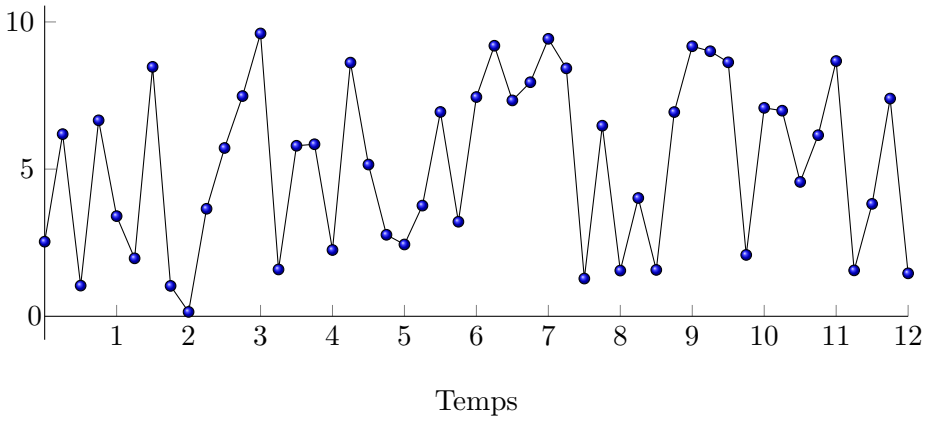


Figure E.1 – Un exemple de courbe stationnaire de moyenne 5

Le modèle AR ne peut être appliqué que sur des séries temporelles (*i.e.* des fonctions discrètes du temps) **stationnaires**.

On définit alors le modèle AR comme :

$$AR_p(X_t) = C + \sum_{k=1}^{p} \phi_k \cdot X_{t-k} + \varepsilon_t$$

où les ϕ_i sont les paramètres du modèle, C est une constante qui permet d'ajuster la moyenne de la prédiction, et ε_t est un bruit blanc généré aléatoirement. On comprend que $AR_p(X_t)$ est en fait une approximation de X_t et que dans ce modèle, l'apprentissage interviendra pour les paramètres ϕ_k. De plus, il est montré dans [20] que :

$$(X_t)\ stationnaire \Leftrightarrow \phi_i \leq 1, \forall i \in [\![1, p]\!]$$

Modèle MA

Le modèle *Moving Average* (moyenne mobile en français) est un modèle permettant de prévoir un terme d'une série temporelle à partir d'une moyenne pondérée des termes précédents, où les poids seront les paramètres appris ; ce modèle dépend d'un méta paramètre q, qui représente la profondeur historique des erreurs prises en compte.

$$MA_q(X_t) = C - \sum_{k=1}^{q} \theta_k \cdot \varepsilon_{t-k} + \varepsilon_t$$

où C est une constante correspondant à la moyenne des X_t, les $(\varepsilon_i)_{i \neq t}$ sont les bruits observés sur les X_k correspondants, ε_i est un bruit blanc généré aléatoirement et les θ_k les paramètres du modèle.

À partir de ces deux modèles, on peut définir le modèle ARMA.

Modèle ARMA

Le modèle ARMA est ainsi la somme de ces deux modèles, et s'écrit :

$$ARMA_{p,q}(X_t) = C + \sum_{k=1}^{p} \phi_k \cdot X_{t-k} + \sum_{k=1}^{q} \theta_k \cdot \varepsilon_{t-k} + \varepsilon_t$$

avec les paramètres précédents. C permet d'ajuster la moyenne de la prédiction.

Notons cependant que ce modèle n'est utilisable que pour les séries stationnaires. Dans le cas de séries non stationnaires, on utilisera un autre modèle, le modèle ARIMA.

E.1.2 Séries non stationnaires et Modèle ARIMA

Si le processus n'est pas stationnaire, il est possible de le « différencier » afin de le rendre stationnaire. Nous introduisons pour cela **l'opérateur de dérivation** B.

Définition E.1.2 (Opérateur de dérivation)
Soit une série temporelle $(X_i)_{i\in[\![0,n]\!]}$. *L'opérateur de dérivation* B *est défini par :*

$$BX_t = X_{t-1}$$

Le modèle ARIMA est identique au modèle ARMA, mais pour lequel on considère la variable initiale (X_i) comme une intégrale de la série à étudier, d'où le nom de ARIMA (pour AutoRegressive Integrated Moving Average). On note ainsi $Y_t = X_t - X_{t-1}$, *i.e.* $Y_t = (1-B)X_t$. Le modèle ARIMA prend pour meta-paramètre d, le nombre de différentiations utilisées pour rendre la courbe stationnaire. Le modèle $ARIMA(p,d,q)$ est alors :

$$(1 - \sum_{k=1}^{p} \phi_k \cdot B^k)(1-B)^d X_t = C + (1 + \sum_{k=1}^{q} \theta_k \cdot B^k)\varepsilon_t$$

Notons que sous cette notation compliquée se cache en fait un modèle assez proche du modèle ARMA. En effet, en prenant pour convention $Y_t = (1-B)^d X_t$, on obtient

$$Y_t = C + \sum_{k=1}^{p} \phi_k \cdot Y_{t-k} + \sum_{k=1}^{q} \theta_k \cdot \varepsilon_{t-k} + \varepsilon_t$$

c'est à dire le modèle ARMA, aux conditions de stationnarité près.

E.1.3 Neural AutoRegressive model (NAR)

Dans le modèle NAR, nous utilisons AR, conjointement avec un réseau de neurones. Un exemple d'une telle adaptation est donnée en [21].

L'idée est de prédire X_t en construisant un perceptron multi-couche prenant en entrée les $(X_{t-i})_{i\in[\![1,p]\!]}$. Nous allons tout d'abord décrire les réseaux de neurones.

Réseaux de neurones et perceptrons

Un réseau de neurones est un modèle de calcul dont le fonctionnement s'inspire du système nerveux biologique. Il s'agit d'un graphe orienté dont chaque nœud représente un neurone et dont chaque arête est une synapse. Les données en entrée (dans notre cas, il s'agit des $(X_{t-i})_{i\in[\![1,p]\!]}$) sont représentées comme des nœuds n'ayant que des arêtes sortantes, le résultat sort par une des synapses. Un exemple de réseau de neurones est donné en Figure E.2. Notez que nous ne considérons ici que les réseaux à sortie unique.

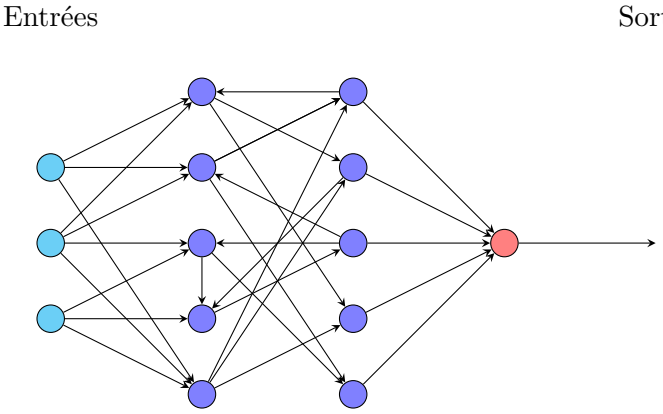


Figure E.2 – Un exemple de réseau neuronal

Chaque neurone calcule une fonction de ces entrées et en renvoie le résultat en sortie.

Un sous-ensemble de ces réseaux est appelé perceptron à multicouche. Un perceptron à multi-couche est composé de plusieurs **couches**, *i.e.* neurones non reliés entre eux. Les couches sont consécutives et ne reviennent jamais vers le passé, c'est à dire qu'il n'y a aucune arête de la couche k vers la couche $k - i_{i\leq k}$. La Figure E.3 donne un tel réseau.

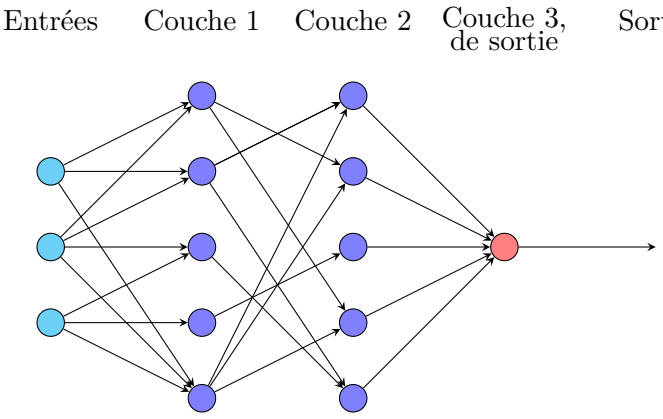


Figure E.3 – Un exemple de perceptron à trois couches cachées et à sortie unique

Réseaux de neurones pour l'autorégression

Dans un modèle NAR (Neuronal AutoRegressive), [21] exprime Y_t en fonction des (Y_{t-i}) :

$$Y_t = \alpha_0 + \sum_{j=1}^{K} \alpha_j \cdot \zeta \left(\sum_{i=1}^{p} \beta_{ij} Y_{t-i} + \beta_{0j} \right) + \varepsilon_t$$

où les β_{ij} sont les poids des connexions entre l'entrée i et la couche j, α_j est le poids de la couche j sur la sortie de la dernière couche à la sortie, $\beta_{0,j}$ est une entrée associée à la couche j, α_0 est une constante associée à la sortie et ζ est la fonction sigmoïde tangente hyperbolique.

E.2 Support Vector Machines (SVM) pour la régression

Pour cette méthode, nous nous sommes appuyés sur [22, 23]. Notons que là encore et comme dans tous les modèles autorégressifs, nous ne nous appuyons que sur une seule série, ce qui signifie que nous ne prenons pas en compte l'ensemble des courbes dont nous disposons, ni notre connaissance métier. L'idée de cette méthode est de faire une régression linéaire sur un autre espace (appelé espace de redescription), plus grand.

Nous transposons les points dans l'espace de redescription et y trouvons une régression linéaire associée grâce aux points connus. Puis, nous apprenons les paramètres de cette régression linéaire. Ensuite, nous prédisons la valeur suivante en réadaptant de ce modèle linéaire dans l'espace d'entrée. Notons Φ la fonction qui permet de passser de notre espace d'entrée à notre espace de redescription ($\mathbf{Y} = \overrightarrow{\Phi(\mathbf{X})}$), supposé de dimension p, et l'espace d'entrée de dimension $t-1$ (nous cherchons à prédire X_t). Écrivons l'équation de l'hyperplan de régression linéaire dans l'espace

$$\sum_{i=0}^{p} \mathbf{w} \cdot \mathbf{Y}^\top + b = 0$$

Une particularité de ce modèle est sa fonction de coût. Elle se définit ainsi :

$$L_\varepsilon(X_t, \hat{X}_t) = \begin{cases} |(X_t) - \hat{X}_t| - \varepsilon & \text{si } |(X_t) - \hat{X}_t| \geq \varepsilon \\ 0 & \text{sinon} \end{cases}$$

Cependant, ce n'est pas cette fonction qu'il faut minimiser. En effet, le risque à minimiser est :

$$R(C) = C \frac{1}{t-1} \sum_{i=1}^{p} L_\varepsilon(X_i, \hat{X}_i) + \frac{1}{2} \|\mathbf{w}\|$$

Cela signifie que seuls les points éloignés de la courbe sont pris en compte, et que l'irréductible bruit n'augmente pas le coût associé de la courbe. Cela a cependant des effets pervers si le bruit observé est très inférieur au bruit anticipé. La Figure E.4

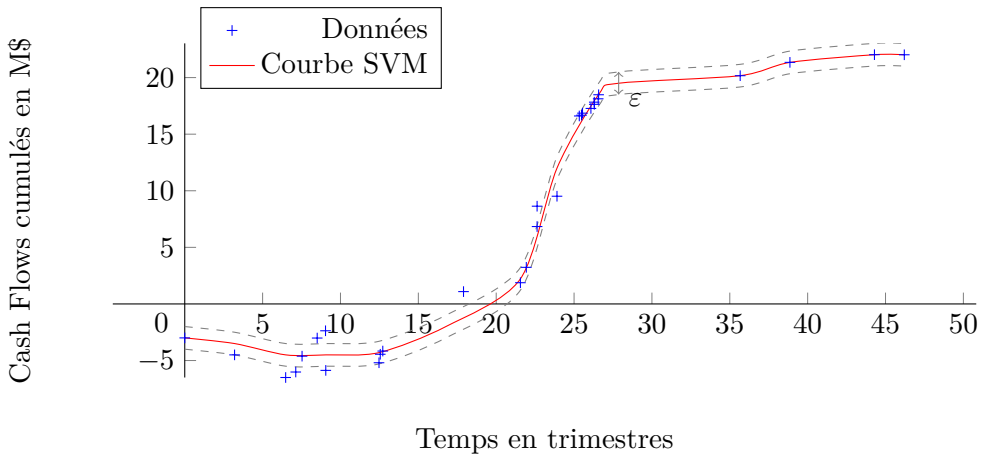


Figure E.4 – Exemple d'utilisation de SVM sur une J-Curve

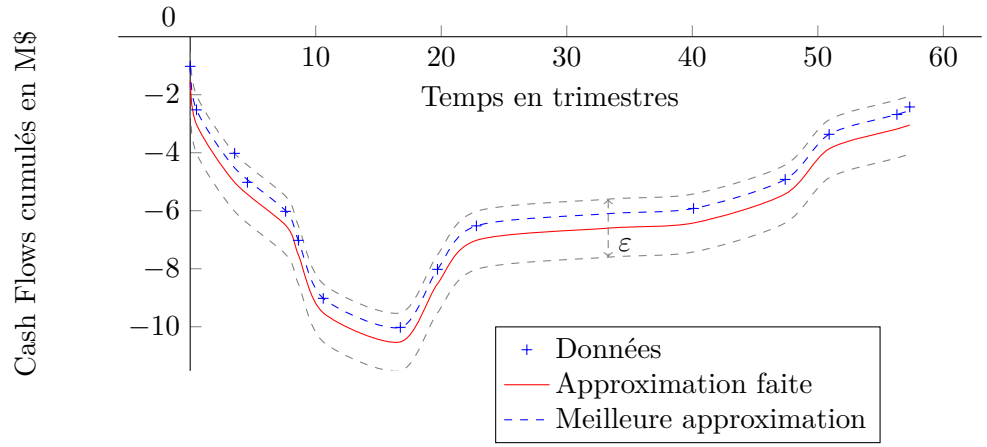


Figure E.5 – Approximation de SVM faussée par un ε trop grand

donne un exemple de la méthode suivie, tandis que la Figure E.5 donne un exemple des problèmes entraînés par l'utilisation d'un ε inadapté.

Sur cette figure, nous constatons que tous les points se trouvent à l'intérieur de la marge $[-\varepsilon, \varepsilon]$. Cependant, la courbe bleue correspond mieux à nos attentes, et il aurait été judicieux de la choisir au lieu de la courbe rouge. Nous aurions obtenu ce résultat en choisissant un ε plus petit, car alors tous les points auraient été en dehors de la marge pour la courbe rouge, mais dans la marge pour la courbe bleue.

Annexe F

Analyses factorielles

F.1 L'Analyse en Composantes Principales

Dans cette section, comme dans la précédente, nous décrivons très brièvement la théorie statistique dont est issue l'ACP. Il s'agit davantage d'un descriptif pratique de la méthode que nous utiliserons qu'une explication de ce qu'est l'ACP et de ses théories sous-jacentes. Pour plus de précision, le lecteur se reportera au livre de Jolliffe, accessible en ligne : [24].

F.1.1 Notations

Le principe de l'ACP est de projeter un ensemble de n individus ayant p « caractéristiques » sur un sous-espace de $\mathbb{R}^p$ de dimension $k, k \leq p$. On dispose d'une matrice

$$\mathbf{X} = \begin{pmatrix} x_1^1 & \dots & x_1^p \\ \vdots & x_i^j & \vdots \\ x_n^1 & \dots & x_n^p \end{pmatrix}$$

où x_i^j est la variable j de l'individu i.

On note $\mathbf{X_i}$ un vecteur ligne de $\mathbf{X}$ (*i.e.* un individu) et $\mathbf{X^j}$ un vecteur colonne de $\mathbf{X}$ c'est à dire une des variable caractérisant ces individus.

Pour un vecteur $\mathbf{v}$, on note $\bar{v}$ la moyenne de $\mathbf{v}$.

F.1.2 Matrice centrée réduite

Les différentes variables ne sont pas indexées de la même façon. Il est possible qu'une variable ait des valeurs comprises entre 0 et 100, alors qu'une autre a ses valeurs comprises entre -0.5 et 1. Pour autant, ces deux variables ont la même importance dans notre analyse. Afin de pallier cela, nous « normalisons » la matrice.

On appelle matrice centrée réduite déduite de $\mathbf{X}$ la matrice $\widetilde{\mathbf{X}} = (\widetilde{x}_i^j)$ où

$$\widetilde{x}_i^j = \frac{x_i^j - \bar{x}^j}{s_j}$$

et s_j est la variance de la colonne j de la matrice $\mathbf{X}$.

F.1.3 L'analyse

Nous notons alors $\mathbf{V}$ la matrice de corrélation :

$$\mathbf{V} = \frac{1}{n}\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}}$$

Notons $(\mathbf{e_k})_{k\in 1,p}$ les vecteurs propres et λ_k leurs valeurs associées. Alors, les vecteurs $\mathbf{v_k}$ suivent les directions des axes principaux de l'ACP. Ces axes sont ceux qui expliqunnt le mieux les valeurs de $\widetilde{\mathbf{X}}$ et λ_k est l'inertie expliquée par l'axe $\mathbf{e_k}$.

F.1.4 Critère d'arrêt

Notre but est de réduire le nombre d'axes présents au départ sur nos données. On déterminera le nombre d'axes que l'on conservera selon le critère de Kaiser, *i.e.* on conserve uniquement les axes dont la valeur propre est supérieure à la moyenne.

F.1.5 Poids

Il est parfois préférable de donner plus de poids à certains individus. Dans ce cas, on utilise une matrice de poids $\mathbf{D}$, de la forme :

$$\mathbf{D} = \begin{pmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & w_n \end{pmatrix}$$

avec w_i le poids associé à l'individu i, $\sum_{i=0}^{n} w_i = 1$. Dans ce cas, la matrice de corrélation s'écrit $\mathbf{V} = \widetilde{\mathbf{X}}^\top\mathbf{D}\widetilde{\mathbf{X}}$.

Il est possible (et même sûr dans notre cas) que certaines informations aient plus d'importance que d'autres. Ainsi, la distance d'une séquence à la séquence de référence (celle que l'on veut prédire) aura forcément plus d'importance qu'un critère anticipé par le client, et qui restera purement hypothétique. Dans ce cas, nous utilisons une matrice de poids non canonique $\mathbf{D'}$ de la forme :

$$\mathbf{D'} = \begin{pmatrix} \sqrt{w'_1}\cdot p & 0 & \dots & 0 \\ 0 & \sqrt{w'_2}\cdot p & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \sqrt{w'_p}\cdot p \end{pmatrix}$$

avec w_i' le poids associé à la variable i, $\sum_{i=0}^{n} w_i' = 1$. Nous notons alors la nouvelle matrice $\mathbf{Y} = \mathbf{D} \cdot \mathbf{X}$. Notons que $\mathbf{Y}$ n'est pas réduite. Dans ce cas, la matrice variance-covariance s'écrit $\mathbf{V} = \mathbf{Y}^\top \mathbf{D}\mathbf{Y}$.

F.2 L'Analyse Factorielle des Correspondances

Cette méthode permet d'analyser des individus ayant deux variables qualitatives. Plus de précisions sur les différents types d'analyse se trouvent dans [25].

Représentation des données

Les données sont placées dans un tableau qui représente le nombre d'individus en fonction des différentes modalités. Prenons un exemple. Supposons que nos courbes ont deux paramètres qualitatifs, le millésime et la stratégie. Le *millesime* est l'année du fonds, comprise entre 2000 et 2009 et la *strategie*, pouvant prendre les valeurs Buyout Large Cap (LBO), Buyout Medium Cap (MBO), Buyout Small Cap (SBO), Venture Capital Early Stage (EVC) et Venture Capital Late Stage (LVC). Les données portent sur 400 individus. Nous représentons alors les données sous la forme d'un tableau, fourni en Table F.1

	EVC	LVC	SBO	MBO	LBO	Total
2000	2	3	4	7	10	26
2001	5	8	5	9	8	35
2002	8	5	7	14	3	37
2003	10	7	8	15	7	47
2004	12	7	7	12	6	44
2005	10	9	8	10	7	44
2006	7	15	9	14	8	53
2007	7	17	3	13	9	49
2008	8	10	4	12	5	39
2009	5	1	3	12	5	26
Total	74	82	58	118	68	400

Table F.1 – Tableau d'AFC : exemple

F.2.1 Métrique du Φ^2

On appelle n le nombre de modalités de la première variable et m le nombre de modalités de la seconde. Avant de définir la métrique du ϕ^2, nous avons besoin de quelques définitions :

Définition F.2.1 (Fréquence)
Le tableau des fréquences présente la fréquence des individus au lieu de leurs effectifs. La fréquence $f_{i,j}$ de la paire de modalités i, j où i est une modalité ligne et j une modalité colonne est définie par

$$f_{i,j} = \frac{Effectif_{i,j}}{Effectif\ total}$$

Le tableau des fréquences est donné en Table F.2.

	EVC	LVC	SBO	MBO	LBO	Total
2000	0.5	0.75	1	1.75	2.5	6.5
2001	1.25	2	1.25	2.25	2	8.75
2002	2	1.25	1.75	3.5	0.75	9.5
2003	2.5	1.75	2	3.75	1.75	11.75
2004	3	1.75	1.75	3	1.5	11
2005	2.5	2.25	2	2.5	1.75	11
2006	1.75	3.75	2.25	3.5	2	13.25
2007	1.75	4.25	0.75	3.25	2.25	12.25
2008	2	2.5	1	3	1.25	9.75
2009	1.25	0.25	0.75	3	1.25	6.5
Total	18.5	20.5	14.5	29.5	17	100

Table F.2 – Tableau d'AFC : fréquences

Définition F.2.2 (Fréquence ligne)
Le tableau des fréquences lignes présente la fréquence des individus par rapport à la fréquence de leur ligne. La fréquence ligne $fl_{i,j}$ de la paire de modalités i, j où i est une modalité ligne et j une modalité colonne est définie par

$$fl_{i,j} = \frac{f_{i,j}}{f_{i,\cdot}}$$

où $f_{i,\cdot} = \sum_{j=0}^{n} f_{i,j}$.

Le tableau des fréquences lignes est donné en Table F.3.

Définition F.2.3 (Fréquence colonne)
Le tableau des fréquences colonnes présente la fréquence des individus par rapport à la fréquence de leur ligne. La fréquence $fc_{i,j}$ de la paire de modalités i, j où i est une modalité ligne et j une modalité colonne est définie par

$$f_{i,j} = \frac{f_{i,j}}{f_{\cdot,j}}$$

où $f_{\cdot,j} = \sum_{j=0}^{n} f_{i,j}$.

	EVC	LVC	SBO	MBO	LBO	Total
2000	7.69	11.54	15.38	26.92	38.46	100
2001	14.29	22.86	14.29	25.71	22.86	100
2002	21.62	13.51	18.92	37.84	8.11	100
2003	21.28	14.89	17.02	31.91	14.89	100
2004	27.27	15.91	15.91	27.27	13.64	100
2005	22.73	20.45	18.18	22.73	15.91	100
2006	13.21	28.30	16.98	26.42	15.09	100
2007	14.29	34.69	6.12	26.53	18.37	100
2008	20.51	25.64	10.26	30.77	12.82	100
2009	19.23	3.85	11.54	46.15	19.23	100

Table F.3 – Tableau d'AFC : fréquences lignes

	EVC	LVC	SBO	MBO	LBO
2000	2.70	3.66	6.90	5.93	14.71
2001	6.76	9.76	8.62	7.63	11.76
2002	10.81	6.10	12.07	11.86	4.41
2003	13.51	8.54	13.79	12.71	10.29
2004	16.22	8.54	12.07	10.17	8.82
2005	13.51	10.98	13.79	8.47	10.29
2006	9.46	18.29	15.52	11.86	11.76
2007	9.46	20.73	5.17	11.02	13.24
2008	10.81	12.20	6.90	10.17	7.35
2009	6.76	1.22	5.17	10.17	7.35
Total	100	100	100	100	100

Table F.4 – Tableau d'AFC : fréquences colonnes

Le tableau des fréquences colonnes est donné en Table F.4.

On peut alors définir la métrique du Φ^2 :

Définition F.2.4 (Métrique du Φ^2)
La distance du Φ^2 entre deux lignes i_1, i_2 est définie par :

$$d_{\Phi^2} = \sqrt{\sum_{j=0}^{m} \frac{fl_{i_1,j} - fl_{i_2,j}}{f_{\cdot,j}}}$$

La distance du Φ^2 entre deux colonnes j_1, j_2 est définie par :

$$d_{\Phi^2} = \sqrt{\sum_{i=0}^{n} \frac{fc_{i,j_1} - fc_{i,j_2}}{f_{i,\cdot}}}$$

Notons que cette métrique est utilisée parce qu'elle permet de faire abstraction des poids des différentes colonnes (respectivement lignes). Elle permet de plus de regrouper certaines modalités sans changer le reste du tableau. Il existe une autre mesure, dite du χ^2, qui utilise les effectifs au lieu des fréquences.

On définit également le taux de liaison, qui représente la variation d'une paire de modalité ligne-colonne par rapport à la situation d'indépendance des deux variables :

Définition F.2.5 (Taux de liaison)
Le taux de liaison $t_{i,j}$ d'une modalité ligne i et d'une modalité colonne j est donné par :

$$t_{i,j} = \frac{f_{i,j} - f_{i,\cdot} \cdot f_{\cdot,j}}{f_{i,\cdot} \cdot f_{\cdot,j}}$$

Les taux de liaison de notre série sont donnés dans la Table F.5.

Nous pouvons en déduire le coefficient Φ^2 :

Définition F.2.6 (Coefficient Φ^2)
Le coefficient Φ^2 est défini par :

$$\Phi^2 = \sum_{(i,j)\in:[\![0,n]\!]\times[\![0,m]\!]} f_{i,\cdot} f_{\cdot,j} t_{i,j}^2$$

Le coefficient du Φ^2 est la variance du taux de liaison, il sert à définir si les données sont plus ou moins indépendantes entre elles. Il est ainsi de 0 si les variables sont indépendantes.

F.2.2 L'analyse

Il s'agit en fait d'une double ACP, selon les modalités lignes et colonnes de données particulières, les $\frac{f_{i,j}}{\sqrt{f_{i\cdot} f_{\cdot,j}}}$. Les données sont ensuite projetées sur les axes

	EVC	LVC	SBO	MBO	LBO
2000	1.41	0.78	-0.06	0.10	-0.56
2001	0.30	-0.10	0.01	0.15	-0.26
2002	-0.14	0.52	-0.23	-0.22	1.10
2003	-0.13	0.38	-0.15	-0.08	0.14
2004	-0.32	0.29	-0.09	0.08	0.25
2005	-0.19	0	-0.20	0.30	0.07
2006	0.40	-0.28	-0.15	0.12	0.13
2007	0.30	-0.41	1.37	0.11	-0.07
2008	-0.10	-0.20	0.41	-0.04	0.33
2009	-0.04	4.33	0.26	-0.36	-0.12

Table F.5 – Tableau d'AFC : taux de liaison

obtenus, et on fait les mêmes traitements que dans une ACP classique. Notons une particularité cependant : la première valeur propre de chaque ACP égale à 1, et n'est en général pas prise en compte. La somme des autres valeurs propres obtenues est égale au coefficient Φ^2.

F.3 L'analyse de correspondances multiples

Cette analyse permet d'analyser des individus ayant plus de deux variables qualitatives, et n'ayant aucune variable quantitative.

F.3.1 Les différentes représentations des données

Les données sont habituellement représentées sous deux formes : les tableaux disjonctifs complets et les tableaux de Burt. Les tableaux disjonctifs complets sont une représentation des données de façon binaire, pour chaque individu-ligne sur chaque modalité colonne.

Prenons un exemple. Admettons que nos séries possèdent quatre caractéristiques : la *localisation* (Eur ou USA), la *devise* (EUR, USD ou GBP pour la Livre Sterling), le *millesime*(2001 à 2005) et la *strategie* (Early Venture Capital (EV), Late Venture Capital (LV) ou Leverage Buy-Out (LBO)), et admettons que nous avons cinq individus i_1, i_2, i_3, i_4 et i_5 tels que $i_1 = (Eur, EUR, 2005, EVC)$, $i_2 = (Eur, GBP, 2003, LVC)$, $i_3 = (USA, USD, 2005, EVC)$, $i_4 = (USA, GBP, 2002, LBO)$ et $i_5 = (Eur, EUR, 2001, LBO)$. Le tableau disjonctif complet de ces données est donné en Table F.6.

Une autre façon de représenter les données est le tableau de Burt, qui représente

	localisation		*devise*			*millesime*					*strategie*		
	Eur	USA	EUR	USD	GBP	2001	2002	2003	2004	2005	EVC	LVC	LBO
i_1	1		1							1	1		
i_2	1				1			1				1	
i_3		1		1						1	1		
i_4		1			1		1						1
i_5	1		1			1							1

Table F.6 – Tableau disjonctif complet : exemple

les donnés les unes en fonction des autres. On peut par exemple y lire le nombre de fonds des USA qui datent de 2001 etc. La Table F.7 représente l'exemple précédent sous la forme de tableau de Burt.

F.3.2 Métrique du Φ^2

Les définitions sont légèrement différentes de celles de l'AFC, puisqu'il y a plus de variables. Nous supposons qu'il y a N variables qualitatives définies par K_n modalités , où $n \leq N$ est la n-ième modalité. On note également $\sum_{k \leq K} K_k = K$ Commençons par définir le taux de liaison dans le cadre d'un tableau de Burt :

Définition F.3.1 (Taux de liaison)
Le taux de liaison d'une cellule est défini par

$$t_{k,k} = \frac{\textit{Nombre d'individus ayant choisi la modalité } k}{\textit{Nombre d'individus n'ayant ps choisi la modalité } k} = \frac{1 - f_k}{f_k}$$

On définit aussi le profil ligne moyen :

Définition F.3.2 (Profil ligne moyen)
Le profil ligne moyen $(\bar{L}_k)_{k \leq K}$ est défini par :

$$\bar{L}_k = \frac{f_k}{N}$$

où f_k est la fréquence de la modalité k.

La notion de profil ligne moyen correspond en fait à l'individu moyen. Notons que cette définition est valable tant pour le tableau disjonctif complet que pour le tableau de Burt.

Nous pouvons dès lors définir la distance du Φ^2

Définition F.3.3 (distance du Φ^2)
La distance du Φ^2 entre deux individus i_1 et i_2 s'écrit :

$$d_{\Phi^2}(i_1, i_2) = \sqrt{\frac{1}{N} \sum_{k \in \{i_1 \setminus i_2\} \cup \{i_2 \setminus i_1\}} \frac{1}{f_k}}$$

où k est une modalité faisant partie d'un des patrons sans faire partie de l'autre.

La distance du Φ^2 entre deux modalités k et k' s'écrit :

$$d_{\Phi^2}(k, k') = \sqrt{\frac{Eff(k) + Eff(k') - 2 \cdot Eff(k \cap k')}{\frac{Eff(k) \cdot Eff(k')}{Eff_{tot}}}}$$

où $Eff(k)$ représente l'effectif de la modalité k et Eff_{tot} est l'effectif total.

Cette définition, bien que quelque peu différente cell données par l'AFC (la représentation des données est différente) a la même signification. Il en est de même pour le coefficient du Φ^2 :

		localisation		*devise*			*millesime*					*strategie*		
		Eur	USA	EUR	USD	GBP	2001	2002	2003	2004	2005	EVC	LVC	LBO
loc.	Eur	3	0	2	0	1	1	0	1	0	1	1	1	1
	USA	0	2	0	1	1	0	1	0	0	1	1	0	1
dev.	EUR	2	0	2	0	0	1	0	0	0	1	1	0	1
	USD	0	1	0	1	0	0	0	0	0	1	1	0	0
	GBP	1	1	0	0	2	0	1	1	0	0	0	1	1
milles.	2001	1	0	1	0	0	1	0	0	0	0	0	0	1
	2002	0	1	0	1	0	0	0	0	0	1	1	0	0
	2003	1	0	0	0	1	0	0	1	0	0	0	1	0
	2004	0	0	0	0	0	0	0	0	0	0	0	0	0
	2005	1	1	1	1	0	0	0	0	0	2	2	0	0
str.	EVC	1	1	1	1	0	0	0	0	0	2	2	0	0
	LVC	0	1	0	0	1	0	0	1	0	0	0	1	0
	LBO	1	1	1	0	1	1	1	0	0	0	0	0	2

Table F.7 – Tableau de Burt : exemple

Définition F.3.4 (Coefficient du Φ^2)
Dans le cadre de l'ACM, le coefficient du Φ^2 *se définit par :*

$$\Phi^2 = \frac{K - N}{N}$$

F.3.3 L'analyse

L'analyse s'effectue sur le tableau de Burt ou sur le tableau disjonctif complet, à la façon d'une ACP. Notons que les valeurs obtenues ne seront pas les mêmes. Notons qu'il s'agit de faire en fait une « AFC multiple ». Ainsi, au lieu de les normaliser de façon classique, nous les centrons puis les divisons par le nomber de variables, multiplié par le nombre total d'individus. En fait, les valeurs propres obtenues avec le tableau de Burt seront le carré de celles obtenues par le tableau disjonctif complet. Notons qu'il est possible de faire des modifications, décrites dans [26] et[27].

Avant de clore ce chapitre et d'expliquer l'analyse mixte que nous utilisons, nous donnons une dernière définition – celle du rapport de corrélation – nécessaire à l'explication de l'AFDM.

Commençons par définir la contribution d'une variable n à un axe $\mathbf{v}$, obtenu lors de l'ACM. Elle est identique à celle de l'ACP (il s'agit du produit scalaire normé de l'axe et des coordonnées de la variable dans le nouveau repère). On peut en déduire le rapport de corrélation d'un axe par rapport à une variable.

Définition F.3.5 (Rapport de corrélation d'un axe par rapport à une variable)
Le rapport de corrélation d'un axe $\mathbf{v}$ *par rapport à une variable n s'écrit :*

$$\eta^2_{\mathbf{v}}(n) = C_{\mathbf{v}}(n) \times \lambda_{\mathbf{v}} \times N$$

où $C_{\mathbf{v}}(n)$ *est la contribution de la variable n à l'axe* $\mathbf{v}$*, et lambda*$_{\mathbf{v}}$ *est la valeur propre de l'axe* $\mathbf{v}$*.*

Nous pouvons maintenant définir la méthode d'analyse pour les données mixtes.

Annexe G

Distances de séries réelles

G.1 Dynamic Time Warping

Le *dynamic time warping* (DTW), qui est donnée dans [28] permet de remédier au principal problème de la distance euclidienne. En effet, dans cette approche, la distance est calculée non directement entre une séquence et une autre, mais entre les sous-séquences de ces deux courbes. Pour deux séquences R, S avec $R = (t_i, r_i)_{i\in[\![1,n]\!]}$ et $S = (t_i, s_i)_{i\in[\![1,m]\!]}$, la distance DTW se définit récursivement par :

$$DTW(R,S) = \begin{cases} 0 & \text{si } n = m = 0 \\ \infty & \text{si } \begin{cases} m \cdot n = 0 \\ m \neq 0 \text{ ou } n \neq 0 \end{cases} \\ \begin{array}{r} DTW(R,S) = Eucl(r_1, s_1) + \\ min\left\{DTW\left(R, (s_i)_{i\in[\![2,m]\!]}\right)\right. \\ DTW\left((r_i)_{i\in[\![2,n]\!]}, S\right), \\ \left. DTW\left((r_i)_{i\in[\![2,n]\!]}, (s_i)_{i\in[\![2,m]\!]}\right)\right\} \end{array} & \text{sinon} \end{cases}$$

Le principal problème de cette mesure est sa résistance (ou plutôt son absence de résistance) au bruit. En effet, comme on peut le voir sur la Figure G.1, la distance peut être élevée, même en cas de bruit sur quelques valeurs seulement de la séquence. Sur la figure, la séquence S est comparée aux séquences T et R. La séquence rouge est strictement identique à la séquence bleue, à trois valeurs « aberrantes » près. Pourtant, $DTW(R,S) \approx 138.5$ et $DTW(R,T) \approx 53.5$

G.2 Longest common subsequence

Cette mesure de similarité, définie dans [29] est différente de la précédente, en cela qu'elle ne respecte pas l'inégalité triangulaire (définie par $D(R,S) + D(S,W) \geq$

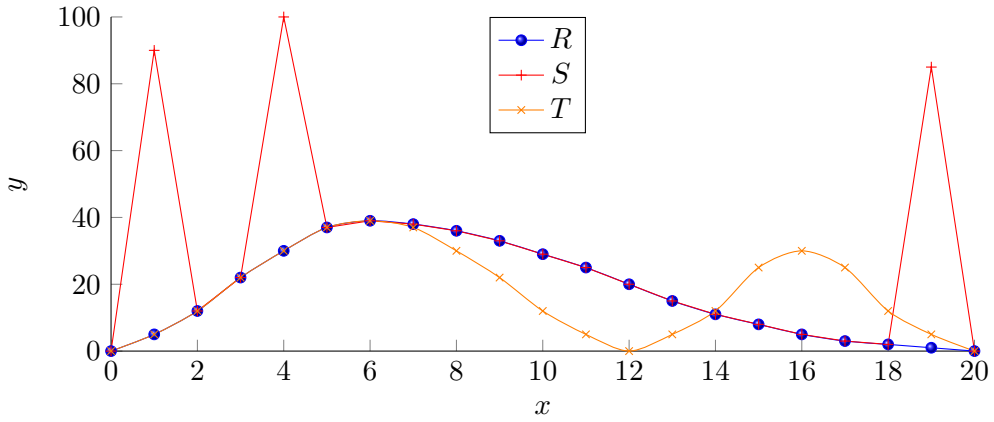


Figure G.1 – Les défauts du dynamic Time Warping

$D(R, W)$. Instinctivement, deux courbes sont similaires si elles ne sont pas éloignées la plupart du temps. C'est ce que cherche à représenter LCS. Pour deux séquences R, S avec $R = (t_i, r_i)_{i \in [\![1,n]\!]}$ et $S = (t_i, s_i)_{i \in [\![1,m]\!]}$, la distance DWT se définit récursivement par :

$$LCS_\varepsilon(R,S) = \begin{cases} 0 & \text{si } n \cdot m = 0 \\ LCS_\varepsilon\left((r_i)_{i\in[\![2,n]\!]}, (s_i)_{i\in[\![2,m]\!]}\right) + 1 & \text{si } \|r_1 - s_1\| \leq \varepsilon \\ max\left\{LCS_\varepsilon\left(R, (s_i)_{i\in[\![2,m]\!]}\right), \right. & \\ \qquad \left. LCS_\varepsilon\left((r_i)_{i\in[\![2,n]\!]}, S\right)\right\} & \text{sinon} \end{cases}$$

Cette méthode est robuste au bruit, à condition de choisir un bon ε.

Annexe H

Apprentissage non supervisé

Nous donnons ci-après une brève explication de chaque méthode. Nous avons eu recours à un article traitant de la classification non supervisée en général, [13] et plus particulièrement un article traitant du *clustering* pour les séries temporelles, [30], qui nous a permis de nous focaliser davantage sur les problématiques de notre sujet. Nous commencerons cependant par une description générale des méthodes d'apprentissage non supervisé. La plus simple à expliquer est la méthode des *k-means*.

H.1 *k-means*

Cette méthode consiste à placer les objets selon leur moyenne, puis à créer des classes (et donc des surfaces séparatrices) autour. L'algorithme figure en Algorithme 5.

Plusieurs remarques sont à faire sur cet algorithme. Tout d'abord, il faut être en mesure de calculer des moyennes et des distances sur l'ensemble auquel appartiennent les données. D'autre part, pour ce qui est de l'initialisation (lignes 3 à 6), il est possible d'initialiser avec n'importe quelles données. Nous donnons un exemple de l'algorithme sur la Figure H.1. Notons que nous n'avons représenté que les deux premières étapes de l'algorithme, mais que celui-ci convergera dès la troisième. Les centres de gravité seront changés, mais les classes contiendront les mêmes points.

Un des inconvénients de cette méthode est qu'elle est en fait une méthode de descente du gradient, ce qui signifie qu'elle s'arrête à un minimum local, et non au minimum global. Dans notre exemple, il s'agit du minimum global, mais cela n'est pas vrai dans le cas général.

Pour obtenir une méthode floue, on notera l'appartenance de chaque point comme une fraction, en fonction des classes voisines et de la distance à chaque centre de gravité.

Algorithme 5 *k-means*

```
1 : function kMeans(k, dat)                    ▷ k est un paramètre de l'algorithme
2 :                                            ▷ dat l'ensemble des données
3 :     for i ∈ ⟦0, k⟧ do
4 :         S[i] ← {dat[i]}
5 :         S[i].mean ← dat[i]
6 :     end for
7 :     while ∃i ∈ ⟦0, k⟧ : mean[i] changed do
8 :         for all s ∈ S do
9 :             s = ∅
10 :        end for
11 :        for all d ∈ dat do
12 :            s = argmin_{x∈S}{‖x.mean − d‖}
13 :            s ← s ∪ {d}
14 :        end for
15 :        for all s ∈ S do
16 :            s.mean = 1/|s| Σ_{d∈s} d
17 :        end for
18 :    end while
19 :    return S
20 : end function
```

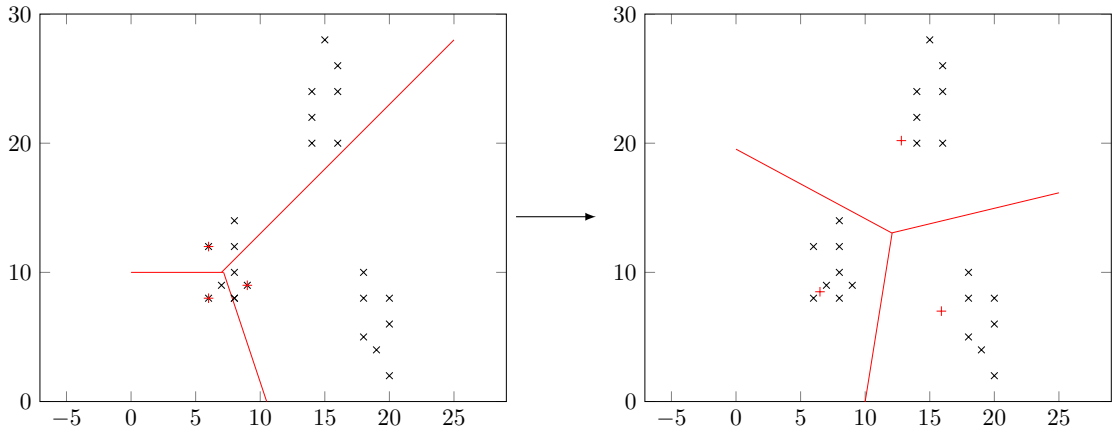


Figure H.1 – Exemple de *k-means* pour 3 classes

H.2 Cartes de Kohonen

Cette méthode utilise les réseaux de neurones. Nous considérons un réseau de neurones ; il s'agit d'une grille à m dimensions (avec $m \in \{1, 2\}$ en pratique). Prenons par exemple une grille de dimension 2, que nous appellerons $\mu_{i,j}, (i, j) \in [\![1, n]\!]^2$. Chacun de ces points va être placé de façon aléatoire sur l'espace des données. Nous allons ensuite introduire les données, et bouger le point de la grille le plus proche de chaque donnée (μ_{i_0,j_0}) pour le rapprocher de cette donnée. Nous approcherons aussi les voisins de ce point (*e.g.* les $\mu_{i,j} : max_{i,j}\{i - i_0, j - j_0\} = 1$). Nous obtenons à la fin de l'algorithme une grille de centres de gravités.

Nous donnons ci-dessous l'algorithme de la classification par cartes de Kohonen. Une figure est disponible au chapitre 18 de [31], p.602. Notons qu'une fois l'algorithme terminé, les points de la grille représentent les données, et sont répartis selon la concentration des points. On peut ainsi avoir une bonne idée de la densité des données en chaque endroit de l'espace.

Algorithme 6 Cartes de Kohonen

```
1 : function KOHONEN(n, dat, η, η', N)        ▷ n est un paramètre de l'algorithme
2 :                                            ▷ dat l'ensemble des données
3 :                                            ▷ V est un voisinage.
4 :                                            ▷ N est le nombre de fois où.
5 :     for (i, j) ∈ [[0, n]]^2 do
6 :         μ_{i,j} = alea()
7 :     end for
8 :     for i ∈ [[1, N]] do
9 :         for all d ∈ dat do
10 :            (i_0, j_0) ← argmin_{i,j}{||μ_{i,j} − d||}
11 :            μ_{i_0,j_0} ← μ_{i_0,j_0} + η(d − μ_{i_0,j_0})
12 :            for all μ_{i,j} ∈ V(μ_{i_0,j_0}) do
13 :                μ_{i,j} ← η'(d − μ_{i_0,j_0})
14 :            end for
15 :        end for
16 :    end for
17 :    return (μ_{i,j})
18 : end function
```

Annexe I

Outils

I.1 Maven

Un projet géré par Maven s'organise comme le montre la figure I.1. Les fichiers contenu dans les dossiers `java` contiennent les fichiers source, permettant de crér les classes et tests nécessaires à l'éxécution. Quant aux fichiers contenus dans `resources`, ils contiennent les autres fichiers (données, logos etc.) nécessaires au projet. Le fichier `POM.xml` contient les informations nécessaires à Maven pour gérer le projet.

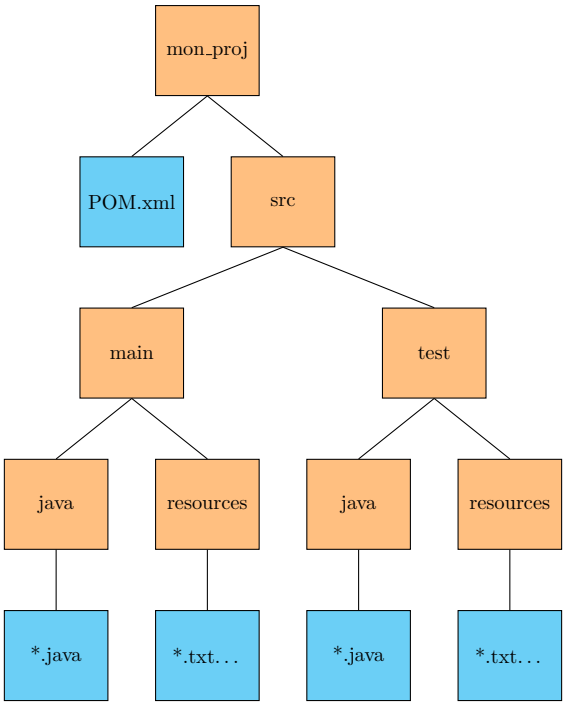


Figure I.1 – Organisation d'un dossier géré par Maven

Le fichier `POM.xml` contient au minimum les informations suivantes :

- le groupe dans lequel est inclus le projet,
- le mode de packaging utilisé (`war` ou `jar`),

- le nom du projet,
- la version du projet.

Le fichier `POM.xml` ressemble à cela :

```
<?xml version="1.0"?>
-<project xsi:schemaLocation=
"http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://maven.apache.org/POM/4.0.0">
  <modelVersion>4.0.0</modelVersion>
  <groupId>group.id</groupId>
  <artifactId>my_proj</artifactId>
  <packaging>jar or war</packaging>
  <version>1.0</version>
  <name>my_proj</name>
  <url>http://maven.apache.org</url>
  -<dependencies>
    -<dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

Il est possible de transformer un projet Maven en projet eclipse grâce à la commande `mvn eclipse:eclipse` appelée depuis le dossier où se trouve `POM.xml`.

I.2 Source Control Manager

I.2.1 Principe

Nous avons égalemnt décidé d'utiliser le gestionnaire de version (ou *source control manager* en anglais) Git. Cet outil permet de synchroniser le travail réalisé sur un dépôt distant et ainsi de se mettre à l'abri de la perte des données. Il permet aussi de travailler à plusieurs, puisque plusieurs personnes peuvent accéder simultanément au même dépôt. Au cours de notre satge, le dépôt Git a été supprimé. Nous avons donc migré vers le dépôt Subversion de l'entreprise, où nous avons également intégré notre projet à Pevara.

I.2.2 Fonctionnement

Le fonctionnement sommaire d'un gestionnaire de version est donné sur la Figure I.2. Un utilisateur fait des modifications sur le projet, et le *push* sur le dépôt.

Lorsqu'un autre utilisateur souhaite modifier le projet. Il *pull* le projet (le télécharge sur sa machine) puis, peut le modifier et faire un *push* et ainsi de suite.

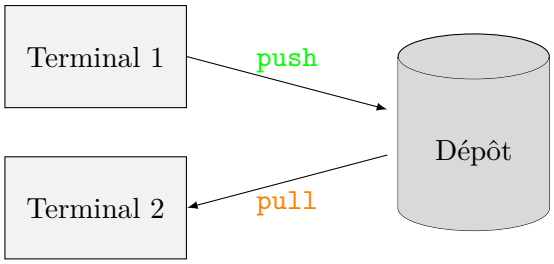


Figure I.2 – Fonctionnement d'un gestionnaire de version

Git fournit d'autres fonctions avancées. Il est ainsi possible de revenir à une version antérieure du projet, de préaprer les changements à *push* en faisant un *commit*. Un autre avantage de Git est qu'il garde l'intégralité des précédentes versions, et permet donc un retour à toutes les versions antérieure.s

Plus d'informations sont disponibles sur le site officiel de Git

I.3 Spring

Spring est un framework d'inversion de contrôle. Dans la représentation « naturelle » d'un programme, le programme appelle une librairie *cf.* Figure I.3a. Il appelle alors une méthode ou une classe dont il connaît l'implémentation concrète.

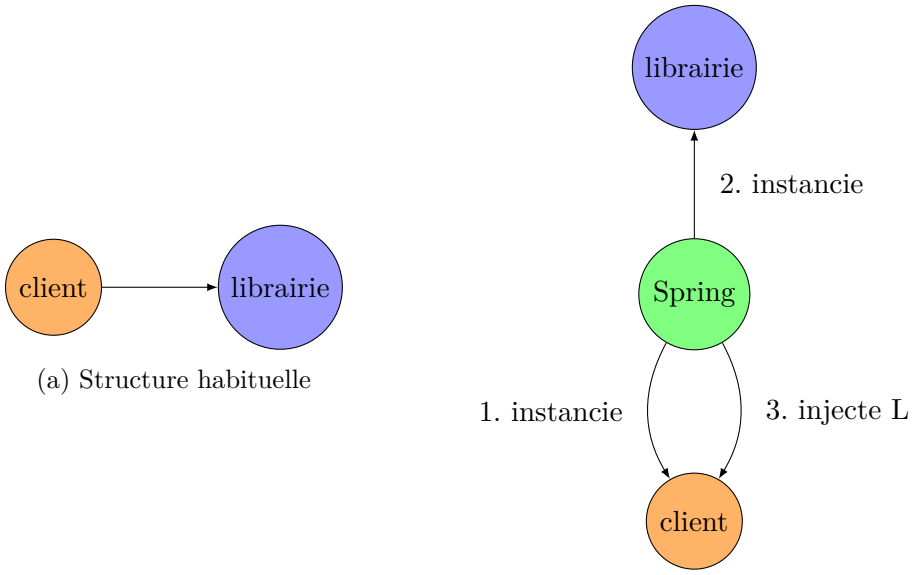
Dans l'inversion de contrôle, le client n'est pas implémenté en premier lieu. Une librairie tierce (le framework d'inversion de contrôle) est appelée d'abord. Il instancie le client, puis les librairies nécessaires. Aucun lien n'existe donc entre les clients et la librairie. Il injecte alors l'instance de la librairie dans le client (injection de dépendance). Notons que, qu'il s'agisse du client ou de la librairie, aucune instance ne connaît le framework d'inversion de contrôle. Ce concept est représenté sur la Figure I.3b.

L'avantage d'une telle pratique est de décorréler la librairie et le client. Avec Spring, l'implémentation de la librairie est décidée lors de l'exécution, de façon dynamique et non statique, comme cela est le cas dans l'approche décrite en Figure I.3a

Il reste à déterminer comment indiquer au framework d'inversion de contrôle qu'une interface est implémentée. Une solution est d'annoter (grâce à une annotation Java) les méthodes appelées par Spring et les variables qui s'en servent. Une autre façon de faire est de référencer ces méthodes et classes dans un fichier (au format XML). Ces deux méthodes sont présentes sur Spring, mais l'habitude de l'entreprise étant plutôt la seconde, c'est celle que nous avons choisie.

I.4 WSDL, SOAP et Axis2

La question de la communication avec le web service se pose également. Il existe pour cela plusieurs protocoles. Le WSDL (ou Web Service Description Language)



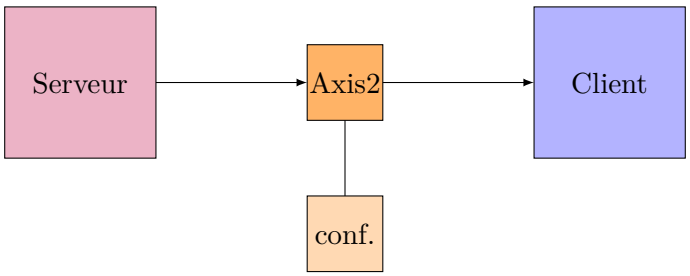
(a) Structure habituelle

(b) Structure inversion de contrôle

Figure I.3 – Appels de librairies

permet de décrire de nombreuses modalités du web service. Parmi ces modalités se trouve la façon de représenter les données et les différentes fonctions. La principale idée est d'envelopper le contenu de ce qui est envoyé / reçu dans le protocole, sous des noms normés, afin que différents langages puissent y accéder. La génération de tels messages ne se fait pas manuellement, mais *via* un outil, nommé Axis2. Cet outil (créé par Apache) permet la génération des messages SOAP, par le serveur depuis Java, qui seront envoyés à FrontInvest (en C#).

Notons au passage que c'est pour cette raison que nous utilisons le WSDL, qui nous empêche notamment d'utiliser une architecture REST. Celle-ci est égaleent utilisée dans l'entreprise et présente de nombreux avantages dont notamment celui de séparer clairement client et serveur mais elle est incompatible avec WSDL car c'est un langage typé, alors que l'architecture REST implique que les ressources ne sont pas spécialement organisées.



Glossaire

Analyse des Correspondances Multiples Analyse factorielle traitant des données qualitatives. 23, A-47

Analyse en Composantes Principales Analyse factorielle traitant des données quantitatives. 23, 40, A-47

Analyse factorielle Ou analyse de données. Famille de méthodes statistique permettant projeter et de visualiser des données multidimensionnelles. 23, A-45

Analyse Factorielle des Correspondances Analyse factorielle traitant deux variables qualitatives. 23, A-47

Analyse Factorielle des Données Mixtes Analyse factorielle traitant à la fois des données qualitatives et quantitatives. 24, 40, A-47

Apprentissage automatique Ou machine learning en anglais. Ensemble de méthodes qui permettent à une machine d'apprendre un certain nombre de règles à partir de données.. 14, 30, A-45

Apprentissage non supervisé Apprentissage automatique dont les données de test ne sont pas classifiées.. 30

Autorégression Méthode de régression pour les séries temporelles dans laquelle la valeur d'une valeur dépend des précédentes.. 17, 18

Capital risque Investissement dans une entreprise ayant pour but son développement. 6, A-46

Capital-investissement Activité financière consistant à investir dans des fonds, *i.e.* dans des marchés non cotés. 1, 7, 12

Classification hiérarchique Méthode de Classification particulière. 30, 31, 33, 35, 48

Classification non supervisée Méthode statistique consistant à regrouper les données en classes cohérentes. xiii, 17, 22, 30, 33, 48

Data mining Ou, en français exploration ou fouille de données. Ensemble de méthodes ayant pour but d'extraire de l'information d'une grande quantité de données.. 1, 13

Descente du gradient Méthode d'optimisation approchée consistant à rechercher comment améliorer la solution localement.. 19

Edit Distance for Real sequences Fonction de distance entre séries temporelles non métrique. Elle est la plus robuste au bruit et aux décalages temporels. 23, A-47

Edit distance with Real Penalties Fonction de distance entre séries temporelles métrique. Elle est robuste aux décalages temporels mais pas au bruit.. 28, A-47

General Partner Propriétaire et gestionnaire d'un fonds.. 5, A-47

Intern Rate of Return Taux de rentabilité interne. Indicateur financier servant de taux de rentabilité. 12, A-47

J-curve Courbe représentatn les cash-flows cumulés ou l'Intern Rate of Return d'un fonds. Ainsi nommée en raison de sa forme.. 12–14

k-plus-proches-voisins Méthode d'apprentissage consistant à prendre en compte les k échantillons les plus proches de la requête.. 22, 23, 33, 38, 40, 47

Least Absolute Deviation Problème classique de programmation linéaire permettant d'obtenir une régression linéaire multiple avec des paramètres positifs.. 20, A-47

Leverage BuyOut Fonds recourant à l'endettement comme effet de levier en vue d'acheter une entreprise.. 6, A-47

Limited Partner Personne physique ou morale investissant dans un fonds.. 5, A-47

Moindres carrés Méthode d'optimisation consistant à minimiser l'erreur quadratique.. 20, A-46

Net Asset Value Valeur liquidative en français. Valeur que l'on retirerait d'un fonds si on le vendait.. 11, A-47

Net Present Value Valeur actuelle nette en français. Indicateur financier indiquant si l'investissement peut réaliser les objectifs attendus des apporteurs de capitaux.. 11, A-47

NonNegative Least Squares Méthode des moindres carrés intégrant une contrainte de positivité des paramètres obtenus.. 20, A-47

Programmation dynamique Technique algorithmique permettant d'optimiser certains algorithmes.. 41, 42

Programmation linéaire Méthode permettant de résoudre des problèmes d'optimisation sous contrainte.. 20, 38, A-46

Real Estate Fonds investissant dans l'immobilier.. 6

Régression Méthode permettant de trouver une relation (mathématique) entre plusieurs variables.. 1, 17

Régression linéaire Méthode de rgression dans laquelle la relation entre les variables est linéaire.. 18, 21

Régression linéaire multiple Méthode de rgression dans laquelle la relation entre les variables est définie par une matrice.. xiii, 13, 30, 33, 35, 38, 48, A-46

Série temporelle Ou série chronologique. Suite de valeurs représentant l'évolution d'une valeur au cours du temps.. 17, 23, 25, 28, 32, 33, 41

Venture Capital *cf.* capital risque.. 6, A-47

Acronymes

ACM Analyse des Correspondances Multiples. 23, 24
ACP Analyse en Composantes Principales. 23–25, 40
AFC Analyse Factorielle des Correspondances. 23
AFDM Analyse Factorielle des Données Mixtes. 24, 40

EDR Edit Distance for Real sequences. 23, 28, 29, 35, 41, 42, 47
ERP Edit distance with Real Penalties. 28, 29, 33, 47

GP General Partner. 5–7, 11, 13, 37

IRR Intern Rate of Return. 12, 14

LAD Least Absolute Deviation. 20
LBO Leverage BuyOut. 6
LP Limited Partner. 5, 7, 11–13, 37

NAV Net Asset Value. 11, 13, 14, 38
NNLS NonNegative Least Squares. 20
NPV Net Present Value. 11

VC Venture Capital. 6

Bibliographie

[1] Thomas Meyer et Pierre-Yves Mathonet : *Beyond the J Curve : Managing a Portfolio of Venture Capital and Private Equity Funds*, volume 566. Wiley. com, 2011.

[2] Frédéric Paulin : Learning parameters for a cashflow forecasting model. Rapport interne d'eFront, 2013.

[3] Charles L. Lawson et Richard J. Hanson : *Solving least squares problems*, volume 15 de *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995. Revised reprint of the 1974 original.

[4] Rasmus Bro et Sijmen De Jong : A fast non-negativity-constrained least squares algorithm. *Journal of chemometrics*, 11(5) :393–401, 1997.

[5] Philip E Gill, Walter Murray et Margaret H Wright : Practical optimization. 1981.

[6] Thomas Laloë : Une approche de type k-plus proches voisins pour la régression fonctionnelle. **In** *41èmes Journées de Statistique, SFdS, Bordeaux*, 2009.

[7] Donald Shepard : A two-dimensional interpolation function for irregularly-spaced data. **In** *Proceedings of the 1968 23rd ACM national conference*, pages 517–524. ACM, 1968.

[8] J Pagès : Analyse factorielle de données mixtes. *Revue de statistique appliquée*, 52(4) :93–111, 2004.

[9] Rakesh Agrawal, Christos Faloutsos et Arun N. Swami : Efficient similarity search in sequence databases. **In** *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*, FODO '93, pages 69–84, London, UK, UK, 1993. Springer-Verlag.

[10] Lei Chen et Raymond Ng : On the marriage of lp-norms and edit distance. **In** *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 792–803. VLDB Endowment, 2004.

[11] Lei Chen, M Tamer Özsu et Vincent Oria : Robust and fast similarity search for moving object trajectories. **In** *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 491–502. ACM, 2005.

[12] Xiaoyue Wang, Abdullah Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann et Eamonn Keogh : Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2) :275–309, 2013.

[13] Rui Xu, Donald Wunsch **et al.** : Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3) :645–678, 2005.

[14] Joe H Ward Jr : Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301) :236–244, 1963.

[15] Peter Langfelder, Bin Zhang et Steve Horvath : Defining clusters from a hierarchical cluster tree : the dynamic tree cut package for r. *Bioinformatics*, 24(5) :719–720, 2008.

[16] Tom Weidig et Thomas Meyer : Modelling venture capital funds. *Risk magazine*, 2003.

[17] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest et Clifford Stein : *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd édition, 2009.

[18] Joshua Bloch : *Effective java*. Addison-Wesley Professional, 2008.

[19] Spyros Makridakis, Steven C Wheelwright et Rob J Hyndman : *Forecasting methods and applications*. Wiley. com, 2008.

[20] Jonathan D Cryer et Kung-Sik Chan : *Time series analysis with applications in R*. Springer, 2008.

[21] Joseph Rynkiewicz, Marie Cottrell, Morgan Mangeas et Jian-Feng Yao : Modèles de réseaux de neurones pour l'analyse des séries temporelles ou la régression : Estimation, identification, méthode d'élagage ssm. *Revue d'intelligence artificielle*, 15(3-4) :317–332, 2001.

[22] Lijuan Cao et Francis Eng Hock Tay : Support vector machine with adaptive parameters in financial time series forecasting. *Neural Networks, IEEE Transactions on*, 14(6) :1506–1518, 2003.

[23] Lijuan Cao et Francis Eng Hock Tay : Application of support vector machines in financial time series forecasting. *Omega*, 29(4) :309–317, 2001.

[24] Ian Jolliffe : *Principal component analysis*. Wiley Online Library, 2005.

[25] Brigitte Escofier et Jérôme Pagès : *Analyses factorielles simples et multiples : objectifs, méthodes et interprétation*. Dunod, 2008.

[26] Jean P Benzecri : Sur le calcul des taux d'inertie dans l'analyse d'un questionnaire, addendum et erratum à [bin. mult.]. *Cahiers de l'Analyse des Données*, 4(3) :377–378, 1979.

[27] Michael Greenacre : *Correspondence analysis in practice*. Chapman and Hall/CRC, 2010.

[28] Donald J. Berndt et James Clifford : Using dynamic time warping to find patterns in time series. **In** Usama M. Fayyad et Ramasamy Uthurusamy, éditeurs : *KDD Workshop*, pages 359–370. AAAI Press, 1994.

[29] Michail Vlachos, George Kollios et Dimitrios Gunopulos : Discovering similar multidimensional trajectories. **In** *Data Engineering, 2002. Proceedings. 18th International Conference on*, pages 673–684. IEEE, 2002.

[30] T Warren Liao : Clustering of time series data–a survey. *Pattern Recognition*, 38(11) :1857–1874, 2005.

[31] Antoine Cornuéjols et Laurent Miclet : *Apprentissage artificiel : concepts et algorithmes*. Eyrolles, 2011.