Apprentissage statistique sur données longitudinales de grande taille et applications au design des jeux vidéo

Thibault Allart

▶ **To cite this version:**

Thibault Allart. Apprentissage statistique sur données longitudinales de grande taille et applications au design des jeux vidéo. Traitement du signal et de l'image [eess.SP]. Conservatoire national des arts et metiers - CNAM, 2017. Français. NNT : 2017CNAM1136 . tel-01683136

HAL Id: tel-01683136
https://theses.hal.science/tel-01683136

Submitted on 12 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

le cnam

École Doctorale Informatique, Télécommunications et Électronique

Centre d'études et de recherche en informatique et communications

THÈSE DE DOCTORAT

présentée par : **Thibault ALLART**

soutenue le : **28 novembre 2017**

pour obtenir le grade de : **Docteur du Conservatoire National des Arts et Métiers**

Discipline / Spécialité : **Informatique**

Apprentissage statistique sur données longitudinales de grande taille et applications au design des jeux vidéo

THÈSE DIRIGÉE PAR

M. NATKIN Stéphane	*Professeur des Universités, CNAM, Cédric*
Mme. GUILLOUX Agathe	*Professeur des Universités, UPMC, LSTA*

ENCADRÉE PAR

M. PIERFITTE Michel	*Directeur de recherche, Ubisoft*
M. LEVIEUX Guillaume	*Maître de Conférence, CNAM, Cédric*

RAPPORTEURS

M. CANU Stéphane	*Professeur des Universités, INSA Rouen, Litis*
M. BUCHE Cédric	*Maître de Conférence HDR, ENIB, STICC*

EXAMINATEURS

Mme. LIMPACH Odile	*Professeur des Universités, Cologne Game Lab*
M. LATOUCHE Aurélien	*Professeur des Universités, CNAM, Cédric, Président du Jury*

"The cake is a lie."
Anonymous - Portal

Remerciements

Cette thèse a été financée par Ubisoft et réalisée en collaboration avec le laboratoire de statistiques théoriques et appliquées (LSTA) de l'Université Pierre et Marie Curie (UPMC) ainsi que le centre d'études et de recherche en informatique et communications (CEDRIC) du conservatoire national des arts et métiers (CNAM). Ce partage du temps, entre les milieux industriel et académique, est rendu possible grâce à la convention industrielle de formation par la recherche, mis en place par l'État français et encadré par l'agence nationale de recherche technologique (ANRT). Je tiens à remercier chaque membre de ces institutions, qui ont contribué à la réalisation de ce projet.

Sur un plan plus personnel, je tiens à remercier Michel Pierfitte pour son engagement dans ce projet. A travers ses nombreux conseils et son goût prononcé pour la recherche il a su me transmettre sa passion pour la science. Mes plus vifs remerciements vont à Agathe Guilloux qui m'a apporté la rigueur mathématique nécessaire au bon déroulement de la thèse ainsi qu'une aide précieuse dans les moments où le temps vient à manquer. Je remercie Stéphane Natkin pour ses enseignements sur la conception des jeux vidéo au travers de l'École nationale du jeu et des médias interactifs numériques (ENJMIN) qui m'ont permis de mieux comprendre les différentes mécaniques à l'œuvre dans la réalisation d'un jeu vidéo. Enfin, un grand merci à Guillaume Levieux pour son encadrement et son aide tout au long de la thèse.

Durant ces trois années, j'ai eu la chance de collaborer et d'échanger de nombreuses idées avec Mokhtar Z. Alaya. J'espère que ces discussions pourront continuer après la thèse. Sur le même registre, les échanges répétés avec Soufiane Khiat m'ont permis d'entrevoir de nouveaux champs d'application aux modèles proposés dans la thèse.

Enfin, ces trois années de thèse n'auraient pas été aussi agréables sans la bonne humeur de mes collègues du CNAM : Stéphanie, Thomas, Simon, Lubna, Zahen, Maria, Clément et Delphine, d'Ubisoft : Franck, Émilie, Marc, Yves, Sylvain, Julien, Arnaud, Mathilde et Adam, et les anciens ou actuels de l'UPMC : Erwan, Roxane, Simon, Félix, Yohann, Matthieu, Le Quyen, Qiming, Emilie et en particulier Sarah avec qui j'ai eu plaisir à collaborer.

Merci à Fabien, alias Tuc pour sa relecture du manuscrit.

Pour conclure, je remercie mes parents et ma famille qui m'ont soutenu durant ces années d'études. Cette aventure n'aurait certainement pas été menée à bien, sans le soutien journalier de ma femme Lisa. Cette thèse lui est dédiée.

Résumé

Cette thèse s'intéresse à l'analyse des données longitudinales, potentiellement grandes selon les trois axes suivants : nombre d'individus, fréquence d'observation et nombre de covariables. A partir de ces données, éventuellement censurées, nous considérons comme facteur d'étude le temps d'apparition d'un ou plusieurs évènements. Nous cherchons, dans des classes de modèles à coefficients dépendant du temps, à estimer l'intensité d'apparition des événements. Cependant, les estimateurs actuels ne permettent pas de traiter efficacement un grand nombre d'observations et/ou un grand nombre de covariables. Nous proposons un nouvel estimateur défini via la vraisemblance complète de Cox et une pénalisation permettant à la fois la sélection de variables et de forcer, quand c'est possible, les coefficients à être constants. Nous introduisons des algorithmes d'optimisation proximaux, permettant d'estimer les coefficients du modèle de manière efficace. L'implémentation de ces méthodes en C++ et dans le package R coxtv permet d'analyser des jeux de données de taille supérieure à la mémoire vive. Nous utilisons pour cela un streaming du flux de données et des méthodes d'apprentissage *online*, telles que la descente de gradient stochastique proximale aux pas adaptatifs. Nous illustrons les performances du modèle sur des simulations en nous comparant aux méthodes existantes. Enfin, nous nous intéressons à la problématique du design des jeux vidéo. Nous montrons que l'application directe de ce modèle, sur les grands jeux de données dont dispose l'industrie du jeu vidéo, permet de mettre en évidence des leviers d'amélioration du design des jeux étudiés. Nous nous intéressons d'abord à l'analyse des composantes bas niveau, telles que les choix d'équipement faits par les joueurs au fil du temps et montrons que le modèle permet de quantifier l'effet de chacun de ces éléments de jeu, offrant ainsi aux designers des leviers d'amélioration directs du design. Enfin, nous montrons que le modèle permet de dégager des enseignements plus généraux sur le design tels que l'influence de la difficulté sur la motivation des joueurs.

Mots clés : Apprentissage statistique, optimisation, jeux vidéo

Abstract

This thesis focuses on possibly large longitudinal time to event data along the following three axes : number of individuals, observation frequency and number of covariates. From this data eventually censored, we consider as study factor the time of occurrence of one or multiple events. We seek in the class of time-varying coefficients to estimate the intensity of the events. However existing estimators did not allow to deal with a huge number of observations and/or variables. We introduce a new estimator based on Cox complete likelihood and a penalty allowing both variables selections and coefficients to be constant over time when it is required. We introduce proximal optimization algorithms to efficiently fit model coefficients. Those methods have been implemented in a C++ library and in a R package coxtv, allowing us to analyse datasets bigger than RAM ; using data streaming and online learning algorithms such as proximal stochastic gradient descent with adaptive learning rates. We illustrate performances on simulations and benchmark with existing models. Finally, we investigate the issue of video game design. We show that using our model on large datasets available in video game industry allows us to bring to light ways of improving the design of studied games. First we have a look at low level covariates, such as equipment choices through time and show that this model allows us to quantify the effect of each game elements, giving to designers ways to improve the game design. Finally, we show that the model can be used to extract more general design recommendations such as difficulty influence on player motivations.

Keywords : Statistical learning, optimization, video games

Table des matières

Résumé **7**

Introduction **21**

I Etat de l'art **25**

1 Apprentissage Statistique **27**

1.1 Généralités 27

1.2 Compromis biais-variance 28

1.3 Approches pénalisées 29

1.3.1 Réduction du nombre de variables 29

1.3.2 Réduction du nombre de variations 31

1.3.3 Groupes de variables 32

1.3.4 Sélection de support 32

1.4 Synthèse 33

2 Optimisation **35**

2.1 Descente de gradient 35

2.1.1 Pas de descente et constante de Lipschitz 36

2.1.2 Approximation du second ordre : Newton Raphson 36

2.1.3 Recherche linéaire 37

2.1.4 Condition de Wolfe 38

2.2 Méthodes proximales 39

2.2.1 Exemple : ISTA pour le problème des moindres carrés 39

2.2.2 Interprétation 40

2.2.3 Recherche linéaire proximale . . . 41
2.3 Méthode du gradient accéléré . . . 41
2.3.1 Accélération . . . 41
2.3.2 Accélération proximale . . . 42
2.3.3 Recherche linéaire accélérée proximale . . . 42
2.4 Descente de gradient stochastique . . . 43
2.4.1 Algorithme du gradient stochastique . . . 43
2.4.2 AdaGrad . . . 44
2.4.3 AdaDelta . . . 45
2.4.4 ADAM . . . 45
2.5 Synthèse . . . 46

3 Analyse de survie et processus de comptage 47
3.1 Definitions . . . 47
3.2 Vraisemblance d'un processus de comptage . . . 48
3.3 Vraisemblance du modèle de Cox . . . 49
3.4 Vraisemblance partielle de Cox . . . 50
3.5 Estimation des paramètres de la vraisemblance partielle de Cox . . . 51
3.6 Synthèse . . . 53

4 Théories du Game design 55
4.1 Mécaniques, dynamique et esthétiques . . . 56
4.2 Les théories de la motivation . . . 57
4.2.1 Le Fun . . . 57
4.2.2 La théorie du flow . . . 57
4.2.3 L'auto détermination . . . 58
4.3 Les modèles du joueur . . . 59
4.3.1 Bartle . . . 59
4.3.2 Yee . . . 60
4.4 Synthèse . . . 60

II Contributions 61

5 Problématique 63

5.1 Descente de gradient stochastique sur la vraisemblance partielle de Cox . . . 63
5.2 L'espace mémoire nécessaire aux algorithmes concurrents . . . 64
5.3 Analyse des données longitudinales dans les jeux vidéo . . . 64
5.4 Approche formelle des modèles motivationnels . . . 65
5.5 Conclusion . . . 66

6 Modèles de Cox et d'Aalen dépendant du temps et en grande dimension 67
6.1 Introduction . . . 67
6.1.1 Cadre et modèles . . . 69
6.1.2 Estimateurs pénalisés constant par morceaux . . . 70
6.2 Procédure d'estimation . . . 71
6.2.1 Modèles et hypothèses . . . 71
6.2.2 Estimation . . . 71
6.3 Algorithmes . . . 73
6.4 Conclusion . . . 75

7 La librairie CoxTV 77
7.1 Modèle de Cox dépendant du temps . . . 77
7.1.1 Coefficients constants par morceaux . . . 78
7.1.2 Données longitudinales . . . 79
7.1.3 Vraisemblance discrète . . . 79
7.2 Implémentation . . . 81
7.3 Exemples d'utilisation du package . . . 82
7.4 Estimation avec support . . . 83
7.5 Expérimentation numérique . . . 84
7.5.1 Données simulées . . . 84
7.5.2 Données réelles . . . 87
7.6 Conclusion . . . 87

8 Influence du design sur le temps de jeu des joueurs 89
8.1 Introduction . . . 89
8.2 Contribution et résultat principal . . . 90
8.3 État de l'art . . . 91
8.4 Méthode . . . 92

8.4.1 Analyse du temps de jeu . . . 93
8.4.2 Comparaison aux autres jeux . . . 94
8.4.3 Régression de Cox . . . 96
8.4.4 Données dépendantes du temps . . . 96
8.4.5 Modélisation des données dépendantes du temps . . . 97
8.5 Jeux de données . . . 98
8.5.1 Far Cry 4 . . . 98
8.5.2 Description des données . . . 99
8.5.3 Visualisation des informations dépendantes du temps . . . 100
8.6 Analyse . . . 101
8.6.1 Rétention et taux de départ . . . 101
8.6.2 Comparaison . . . 102
8.6.3 Le modèle de Cox dépendant du temps . . . 103
8.7 Conclusion . . . 103

9 Influence de la difficulté au cours du temps sur la motivation 107
9.1 Introduction . . . 107
9.2 La difficulté dans les jeux vidéo . . . 108
9.2.1 Estimation de la difficulté . . . 110
9.3 Analyse de survie . . . 111
9.3.1 Rétention . . . 111
9.3.2 Estimation des paramètres du modèle de Cox . . . 112
9.3.3 Les coefficients et variables dépendants du temps . . . 113
9.4 Données et jeux . . . 114
9.5 Résultats . . . 115
9.5.1 Difficulté estimée . . . 115
9.5.2 Lien avec la rétention . . . 117
9.6 Conclusion . . . 120

Conclusion **123**

Bibliographie **124**

Glossaire **135**

Liste des tableaux

7.1 Résultats en MSE pour $n = 100$. CoxTV est la méthode pénalisé que nous avons introduit, Support correspond à sa version non pénalisée avec restriction au support déterminé par CoxTV, H&H est la méthode de Honda et Hardle et Timereg correspond aux résultats de la librairie timereg. 85

7.2 Résultats en MSE pour $n = 1000$. CoxTV est la méthode pénalisé que nous avons introduit, Support correspond à sa version non pénalisée avec restriction au support déterminé par CoxTV, H&H est la méthode de Honda et Hardle et Timereg correspond aux résultats de la librairie timereg. 86

7.3 Résultats en sélection de variables pour $n = 100$. CoxTV est la méthode pénalisé que nous avons introduit et H&H est la méthode de Honda et Hardle. 86

7.4 Résultats en sélection de variables pour $n = 1000$ de la méthode CoxTV que nous avons introduit. 86

7.5 Temps d'estimation en minutes pour les méthodes CoxTV, sa version support, Honda et Hardle et timereg. 86

Table des figures

1.1 Compromis biais variance 28
1.2 Lasso 30
1.3 Ridge 30

2.1 Oscillation et non convergence 37
2.2 Condition d'Armijo. Source [1] 38
2.4 Illustration de la descente de gradient complète en rouge et stochastique en rose. Source [2] 44

4.1 théorie du flow 58
4.2 Classification des joueurs de Bartle 59

7.1 Représentation des données temporelles en 3D. Individus et covariables selon les axes x et y et fréquence d'observation selon l'axe z 80
7.2 Intervalles communs à $X_{i,j}$ et β_j. 80
7.3 Illustration de la pénalité par variation totale. Coefficients non pénalisés (orange) et pénalisés (rouge). Valeur de la pénalité (vert). 84
7.4 Coefficients de régression cumulés estimés sur les données PBC par CoxTV(rouge), la version support (bleu) et `timereg` (rose). 87

8.1 Survie horaire de quelques jeux disponibles sur Steam. 95
8.2 Exemple de données longitudinales pour un joueur de Far Cry 4. La courbe bleu represente le nombre de compétences débloquées par le joueur. La courbe rouge correspond à l'argent (virtuel) dont dispose le joueur. Chaque point vert illustre l'acquisition d'une nouvelle arme par le joueur. 97
8.3 Densité au cours du temps du nombre d'ennemis tués avec l'arc. 100
8.4 Rétention (en noir) et taux de départ (en rouge) des joueurs de Far Cry 4. . 101
8.5 Survie horaire de Far Cry 4 comparée à d'autres jeux. 102

8.6 Coefficients dépendant du temps (en bleu) estimés à partir des données sur l'utilisation des armes de Far Cry 4. Les lignes pointillés représentent les intercalles de confiance à 95%. La ligne constante à zero sert de référence pour l'effet nul. 104

9.1 Estimation de la difficulté pour un joueur de *Rayman Legends* 111

9.2 Illustration de la pénalité par variation totale. Coefficients non pénalisés (orange) et pénalisés (rouge). Valeur de la pénalité (vert). 114

9.3 Difficulté estimée de *Rayman Legends* et *Tom Clancy's The Division* : Médiane, quantiles 0.25 et 0.75 (tirets) et 0.05 et 0.95 (pointilles) 116

9.4 Coefficients dépendant du temps pour *Rayman Legend* 117

9.5 Coefficients dépendant du temps pour *Tom Clancy's The Division* 119

Introduction

Contexte

L'interconnexion de pratiquement tous les dispositifs avec lesquels nous interagissons, la capacité à surveiller nos comportements via des objets connectés, cet univers où tous nos comportements sont traqués par « l'intelligence ambiante » est pour demain, si ce n'est aujourd'hui. Cette capacité à générer et concentrer des données croit avec le nombre d'utilisateurs des dispositifs concernés et donne lieu à ce que l'on nomme communément « les données massives » (Big Data) et les fameux « algorithmes » qui les exploitent. Un des principaux objectifs de cette collecte est de prédire le comportement des utilisateurs dans des domaines allant de la santé jusqu'à la surveillance en passant, bien entendu, par le marketing. D'un point de vue scientifique ces prédictions se heurtent à différentes difficultés, dont deux d'entre elles font l'objet de cette thèse. La première d'entre elle relève de l'optimisation des performances. Il s'agit d'adapter certaines méthodes d'analyse statistiques pour le traitement de volumes de données qui n'avaient jusque là pas été envisagés. Le second problème vise à faire en sorte que les résultats fournis par les algorithmes statistique soient interprétables. Prévoir que l'habitant d'une maison intelligente va avoir un malaise est intéressant, mais ce qui l'est encore plus c'est de savoir pourquoi. Il s'agit ici d'interpréter la prédiction, pour être plus efficace ou pour améliorer la conception d'un dispositif. Cette thèse aborde ces deux types de problèmes dans le cadre de la conception des jeux vidéos.

La conception d'un jeu vidéo est un processus complexe, puisque les designers ne créent pas directement l'expérience vécue par les joueurs, mais mettent en place les mécaniques de jeu qui vont engendrer l'expérience utilisateur. De cette complexité est né le besoin, tant industriel qu'académique, de mieux comprendre l'influence entre les éléments de design d'un jeu vidéo et l'activité des joueurs.

Des méthodes existent pour s'assurer que les intentions de design soient correctement traduites en jeu, telles que le rational game design mis en place chez Ubisoft. Cependant, les processus à l'œuvre dans un jeu vidéo sont complexes et la modification d'un élément du design peut avoir des conséquences inattendues sur l'ensemble du jeu et changer profondément le comportement des joueurs. Pour évaluer les conséquences de ces changements, les designers n'ont d'autre choix que de tester et faire tester le contenu qu'ils ont créé.

Les playtests, études qualitatives où les joueurs sont invités à jouer une sous-partie où l'ensemble d'un jeu, sont régulièrement mis en place tout au long de la conception d'un jeu. Durant ces tests, le comportement des joueurs est observé et ils sont invités à donner leur avis sur de nombreux aspects du jeu. Ces tests permettent, dès la phase de production du jeu, d'avoir un retour qualitatif des utilisateurs et de modifier le contenu en conséquence. Cependant ces tests sont coûteux et le nombre de participants excède rarement la centaine. Dans ces conditions et compte tenu de la complexité des jeux, il est souvent difficile d'obtenir des résultats quantitatifs.

La connectivité toujours croissante de nos appareils de jeu (consoles, pc, tablettes, smartphones, ...) permet de récupérer des informations sur le comportement de millions de joueurs. Leur analyse nous permettrait non seulement de mieux comprendre le comportement des joueurs, mais aussi de déterminer quels sont les composants du jeu à modifier pour améliorer leur satisfaction et leur temps de jeu. Cependant l'analyse de données aussi complexes et volumineuses s'accompagne de nouvelles problématiques.

Actuellement, les éditeurs de jeux vidéo observent l'état de santé de leurs titres à travers des mesures de l'activité des joueurs. Ces indicateurs sont comparés d'un jeu à l'autre et la prédiction de leur évolution au long terme offre une vision sur l'activité future des titres. Préalablement à la thèse, nous avons mis en place une méthode statistique permettant de prédire l'activité et la monétisation des joueurs dans les jeux de type free-to-play. Bien que la connaissance anticipée de ces indicateurs ait de nombreuses applications business, elle offre peu d'informations aux designers sur les leviers d'amélioration de leur jeu. C'est pour pallier à cette problématique que nous nous sommes engagés dans ce sujet de recherche.

Problématique

Les données relatives au comportement des joueurs sont très riches, puisqu'il est possible de conserver une trace de toutes leurs activités dans le jeu. Cependant, les modèles de classification ou de régression les plus simples ne permettent pas d'analyser des données longitudinales (dépendant du temps). C'est pourquoi, certains modèles déployés dans l'industrie se focalisent sur l'analyse des caractéristiques du joueur au moment où celui ci arrête de jouer. Cependant, en omettant l'aspect longitudinal des données et en se concentrant sur une photographie de leurs caractéristiques au moment où ils quittent le jeu, plutôt que d'analyser l'ensemble de leur trajectoire temporelle, ces études prennent le risque de tirer de fausses conclusions. Notre modèle doit donc être capable de prendre en compte tout l'historique des joueurs, ce qui nous conduit aux questions suivantes.

— Existe-t-il un modèle d'analyse des temps d'évènements à partir de données longitudinales, dont les résultats sont facilement interprétables et capable à la fois de traiter de grands jeux de données et un grand nombre de variables ?
— Dispose-t-on d'une implémentation de ce modèle, pouvant traiter des jeux de données de grande taille en temps raisonnable ?

Une fois ces problématiques résolues, nous pourrons traiter les deux questions suivantes :

— Peut-on extraire de l'analyse de l'activité des joueurs des recommandations de design spécifiques au jeu étudié ?
— Est-il possible d'extraire des recommandations de design plus larges, sur des variables partagées entre plusieurs jeux ?

Plan et contribution

L'objectif de cette thèse est de proposer une méthode d'analyse des données longitudinales, permettant d'extraire des recommandations sur le design des jeux vidéo, à partir des traces de l'activité des joueurs. Le modèle proposé pour extraire ces informations n'est pas spécifique aux jeux vidéo. Aussi, nous avons séparé les contributions mathématiques et algorithmiques de leurs applications.

Le chapitre 1 présente le cadre de l'apprentissage statistique. Nous abordons en particulier les méthodes de sélection de variable par approches pénalisées, utiles à la compréhension du modèle proposé.

Dans le contexte des données de grande taille, le choix d'une méthode d'apprentissage statistique est indissociable de la méthode d'optimisation employée pour procéder à l'estimation du modèle. Le chapitre 2 présente les méthodes d'optimisation déployées pour estimer les paramètres du modèle proposé. Nous nous attardons principalement sur les méthodes proximales, permettant d'estimer les paramètres des modèles d'apprentissage statistique pénalisés présentés dans le chapitre précédent.

L'un des objectifs de la thèse est de mieux comprendre les facteurs qui, dans le design du jeu, poussent le joueur à partir ou au contraire à continuer la partie. Nous nous intéressons donc à la modélisation du temps de réalisation d'un ou plusieurs évènements, tels que le départ du joueur. Parmi les modèles d'apprentissage statistique, ceux de l'analyse de survie et des processus de comptage sont voués à l'étude de tels phénomènes. Nous en faisons l'introduction dans le chapitre 3.

Le chapitre 4 introduit quelques-unes des théories du game design. Une attention particulière est portée aux modèles motivationnels, qui cherchent à établir les causes de l'intérêt des joueurs pour les jeux vidéo.

Les quatre principales problématiques de la thèse sont rappelées dans le chapitre 5.

Le chapitre 6 introduit les modèles de Cox et d'Aalen dépendant du temps dans le cadre des données en grande dimension. Nous détaillons les estimateurs constants par morceaux proposés, ainsi que la pénalité proposée. Enfin, nous décrivons les algorithmes utilisés pour l'estimation du modèle de Cox dépendant du temps. L'implémentation du modèle proposé est présentée au chapitre 7. Elle permet l'analyse de grands jeux de données sous la forme d'une librairie C++ et du package R *coxtv*. Nous détaillons quelques unes de ses caractéristiques et illustrons ses performances sur des simulations et un jeu de donnée réel.

Les chapitres suivants se consacrent à l'application du modèle à des jeux de données issus du jeu vidéo. Plus précisément, ils visent à mieux comprendre quels sont les éléments qui, dans le design d'un jeu vidéo, contribuent à l'attrait des joueurs pour ce dernier. Tout d'abord, dans le chapitre 8 nous montrons comment ce modèle nous a permis d'estimer le lien entre des éléments de design spécifiques à un jeu, tels que le choix de l'équipement fait par le joueur, et le temps de jeu des joueurs. Enfin dans le chapitre 9, nous explorons l'influence de la difficulté sur la motivation des joueurs, à travers l'analyse de deux jeux vidéo de style différent.

Le lecteur intéressé uniquement par la partie mathématique de la thèse et ses applications à d'autres domaines que le jeu vidéo peut commencer sa lecture par les chapitres 1, 2, 3, 5, 6 et 7. A contrario, le lecteur intéressé principalement par les applications aux jeux vidéo, pourra entamer sa lecture par les chapitres 4, 5, 8 et 9.

Première partie

Etat de l'art

Chapitre 1

Apprentissage Statistique

Dans cette section, nous présentons succinctement le cadre de l'apprentissage statistique supervisé, carrefour des statistiques, de l'informatique et de l'optimisation. On trouvera une présentation détaillée des concepts développés ici dans Hastie et al. (2001) et Vapnik (1995).

1.1 Généralités

Considérons un couple (X, Y) de variables aléatoires à valeurs dans $\mathcal{X} \times \mathcal{Y}$ de loi jointe $\mathbb{P}_{(X,Y)}$ inconnue. L'objectif de l'apprentissage supervisé est de trouver, au sein de la classe $\mathcal{F}$, une fonction $\hat{f} : \mathcal{X} \to \mathcal{Y}$, telle que $\hat{f}(X)$ soit proche (en un certain sens) de Y.

Soit $L : \mathcal{Y} \times \mathcal{X} \to \mathbb{R}^+$ une fonction de coût mesurant l'écart entre la prédiction $f(X)$ et la valeur de Y. Le risque associé à f est défini comme l'espérance de cette fonction de coût :

$$R(f) = \mathbb{E}[L(Y, f(X)]. \tag{1.1}$$

En d'autres termes, le risque mesure l'écart moyen entre Y et sa prédiction $f(X)$. Notons $f^\star$ et $f^\star_{\mathcal{F}}$ les fonctions qui respectivement minimisent le risque sur l'ensemble des fonctions et sur les fonctions appartenant à la classe $\mathcal{F}$:

$$f^\star \in \operatorname*{argmin}_f R(f), \tag{1.2}$$

$$f^\star_{\mathcal{F}} \in \operatorname*{argmin}_{f \in \mathcal{F}} R(f). \tag{1.3}$$

En pratique, comme la loi $\mathbb{P}_{(X,Y)}$ est inconnue, on ne peut pas calculer directement $f^\star$, ni même $f^\star_{\mathcal{F}}$. Cependant, on dispose d'un échantillon $((X_1, Y_1), \ldots, (X_n, Y_n))$ de n observations indépendantes et identiquement distribuées issues de la loi $\mathbb{P}_{(X,Y)}$. On définit alors l'estimateur $\hat{f}$ de $f^\star$ comme la fonction de $\mathcal{F}$ qui minimise le risque empirique $\hat{R}(f)$ sur cet échantillon :

$$\hat{f} \in \underset{f \in \mathcal{F}}{\operatorname{argmin}}\, \hat{R}(f) = \underset{f \in \mathcal{F}}{\operatorname{argmin}}\, \frac{1}{n} \sum_{i=1}^{n} L(Y_i, f(X_i)). \tag{1.4}$$

1.2 Compromis biais-variance

Les performances d'un estimateur $\hat{f}$ basé sur la minimisation du risque empirique sont très sensibles au choix de la classe $\mathcal{F}$. L'excès de risque s'écrit et se décompose comme suit :

$$R(\hat{f}) - R(f^\star) = \Big(R(\hat{f}) - R(f^\star_{\mathcal{F}})\Big) + \Big(R(f^\star_{\mathcal{F}}) - R(f^\star)\Big). \tag{1.5}$$

Le premier terme, dit de variance, mesure l'erreur d'estimation dans la classe $\mathcal{F}$. Le second terme, dit de biais, mesure l'erreur commise par la restriction à la classe $\mathcal{F}$. On parle aussi d'erreur d'approximation pour désigner le biais. Plus la complexité de $\mathcal{F}$ augmente, plus le biais diminue et plus la variance augmente (voir fig.1.1). La difficulté est donc d'identifier de ce qui est générique au problème d'apprentissage en négligeant ce qui est spécifique à l'échantillon donné. Dans le cas où l'estimateur présente un fort biais et une faible variance, on parle de sous apprentissage. Au contraire, si l'estimateur présente un faible biais, mais une forte variance, on parle de sur-apprentissage.

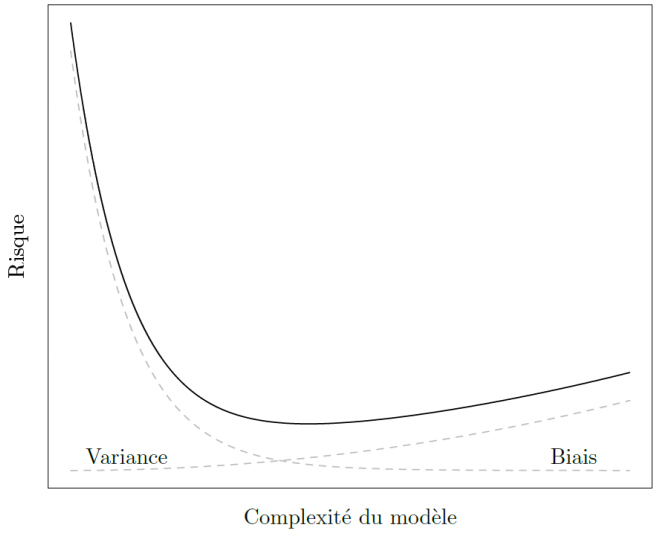

FIGURE 1.1 – Compromis biais variance

1.3 Approches pénalisées

1.3.1 Réduction du nombre de variables

La recherche du meilleur estimateur $\hat{f}$ au sens du compromis biais-variance est un problème fondamental en statistique. Une solution possible consiste à pénaliser le risque empirique de manière à contraindre la complexité de l'estimateur :

$$\hat{f} \in \underset{f \in \mathcal{F}}{\operatorname{argmin}} \left\{ \hat{R}(f) + \operatorname{pen}(f) \right\}. \tag{1.6}$$

Par la suite nous considérons le cas paramétrique où f dépend d'une fonction connue et d'un vecteur de paramètres $\beta \in \mathbb{R}^p$ inconnus (par exemple $f(X) = X\beta$ en régression linéaire). La solution du risque empirique pénalisé s'écrit :

$$\hat{\beta} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \hat{R}(\beta) + \operatorname{pen}(\beta) \right\}. \tag{1.7}$$

Les approches pénalisés ont été introduites dans le but d'améliorer la qualité du modèle, en réduisant par exemple le nombre de variables. Notons que dans le cas où chaque variable n'est liée qu'à un coefficient, la réduction du nombre de variables est équivalente à la réduction du nombre de coefficients. Idéalement, on voudrait donc pouvoir contrôler le nombre de coefficients non nuls du modèle. C'est ce que réalise la pénalité ℓ_0 (ou sélection complète), pour laquelle on définit l'estimateur pénalisé

$$\hat{\beta}_\lambda^{\ell_0} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \hat{R}(\beta) + \lambda \|\beta\|_0 \right\}, \text{ pour } \lambda > 0 \tag{1.8}$$
$$\text{où } \|\beta\|_0 = \#\left\{j, \beta_j \neq 0\right\}.$$

La pénalisation ℓ_0 inclut par exemple les critères d'information AIC (Akaike Information Criterion Akaike (1974)), BIC (Bayesian Information Criterion Schwarz et al. (1978)) et le Cp de Mallows Mallows (1973). Cependant la résolution de (1.8) est un problème non polynomial difficile, rapidement insoluble quand p devient grand. Pour pallier ce défaut, on peut remplacer la pénalité ℓ_0 par sa relaxation convexe : la norme ℓ_1. Cette idée a été introduite par Tibshirani (1996). On définit alors l'estimateur Lasso comme

$$\hat{\beta}_\lambda^{\ell_1} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \hat{R}(\beta) + \lambda \|\beta\|_1 \right\}, \text{ pour } \lambda > 0 \tag{1.9}$$
$$\text{où } \|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|.$$

Pour comparaison, nous rappellons ici la pénalisation ℓ_2 ou Ridge (voir Tikhonov (1963)

et Hoerl & Kennard (1970)) pour laquelle l'estimateur est défini par

$$\hat{\beta}_\lambda^{\ell_2} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \hat{R}(\beta) + \lambda \|\beta\|_2^2 \right\}, \text{ pour } \lambda > 0 \tag{1.10}$$

$$\text{où } \|\beta\|_2^2 = \sum_{j=1}^{p} \beta_j^2.$$

Les graphiques 1.2 et 1.3 permettent de visualiser les boules unité de normes ℓ_1 et ℓ_2 et les solutions respectives du Lasso et du Ridge en rouge. Sur cet exemple, on constate que le coefficient β_1 solution du Lasso est contraint à zero. Ce n'est pas le cas de la solution avec pénalité ℓ_2 pour laquelle les deux coefficients sont non nuls. Plus généralement, le Lasso introduit de la sparcité, alors que ce n'est pas le cas du Ridge.

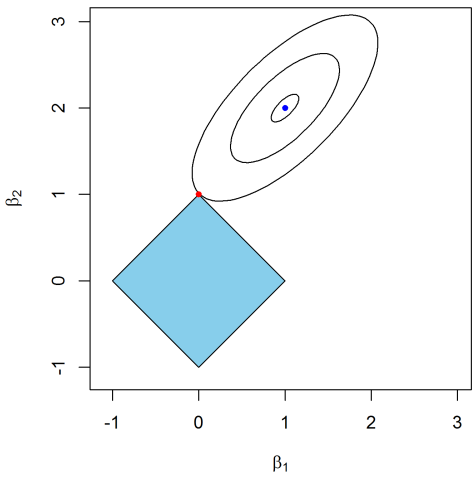

FIGURE 1.2 – Lasso

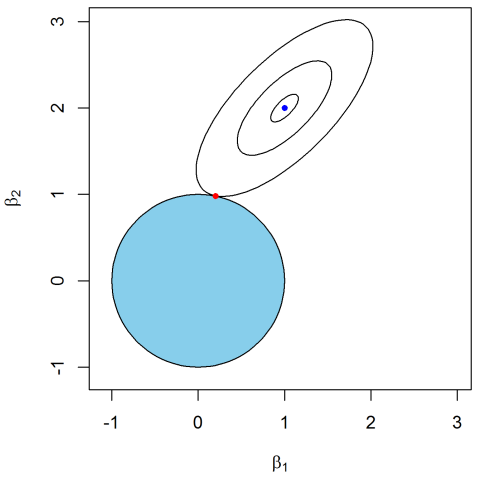

FIGURE 1.3 – Ridge

Le Lasso présente toutefois un problème d'identifiabilité. Dans le cas où certaines variables sont fortement corrélées entre elles, on aimerait que l'ensemble de ces variables soient sélectionnées. Cependant, le Lasso à tendance à ne garder qu'une partie de ces variables. Pour illustrer ce phénomène, prenons le cas où $\hat{\beta}_\lambda^{\ell_1}$ est le vecteur de taille p solution du lasso. Si la pénalité n'a pas réduit tous les coefficients à zero, il existe un j tel que le coefficient β_j associé à la variable x_j soit non nul. Introduisons une nouvelle variable $x'_j = x_j$, copie de x_j. Par propriété de la norme ℓ_1, il en découle que pour tout $\alpha \in [0, 1]$, le vecteur $(\beta_1, \cdots, \beta_{j-1}, \alpha\beta_j, (1-\alpha)\beta_j, \beta_{j+1}, \beta_p)$ est solution du Lasso. Donc la solution du lasso n'est plus unique et l'algorithme d'optimisation peut très bien choisir de ne conserver que la variable x_j où réciproquement la variable x'_j. Dans le cas du Ridge, ce problème ne ce présente pas, puisque la pénalité ℓ_2 tend à répartir de manière équivalente le coefficient β_j entre l'ancienne et la nouvelle variable, voir Hastie et al. (2015). Pour effectuer une sélection de variables tout en conservant les variables corrélés il est possible de combiner

les pénalité ℓ_1 et ℓ_2. Cette régularisation porte le nom d'Elastic-net et fut introduite par Zou & Hastie (2005). L'estimateur Elactic-net est défini comme

$$\hat{\beta}^{\ell_1+\ell_2}_{\lambda_1,\lambda_2} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \hat{R}(\beta) + \lambda_1\|\beta\|_1 + \lambda_2\|\beta\|_2 \right\}, \text{ pour } \lambda_1, \lambda_2 > 0. \tag{1.11}$$

1.3.2 Réduction du nombre de variations

Les pénalités qui précèdent visent à réduire le nombre de variables du modèle en forçant certains coefficients à zero. Si l'on se place du point de vue du problème d'optimisation, ce qui compte ce n'est pas le nombre de variables du modèle, mais le nombre de coefficients. Plus ce dernier est grand, plus le problème d'optimisation est difficile à résoudre. Nous présentons ici des pénalités qui visent à réduire le nombre de coefficients en les regroupant en un unique coefficient. Une des idées consiste à réduire le nombre de variations successives d'un signal ordonné.

Soit $\beta = (\beta_1, \cdots, \beta_p)$ un vecteur de paramètres de longueur p. Pour obtenir une solution avec un minimum de variations entre les coefficients, on peut utiliser une pénalité comptant le nombre de variations entre β_1 et β_p. On définit l'estimateur ainsi pénalisé comme

$$\hat{\beta}^{V0}_{\lambda} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \hat{R}(\beta) + \lambda\|\beta\|_{V0} \right\}, \text{ pour } \lambda > 0 \tag{1.12}$$

$$\text{où } \|\beta\|_{V0} = \#\left\{j, \beta_{j+1} \neq \beta_j\right\}. \tag{1.13}$$

Cependant, tout comme pour la pénalité ℓ_0, cette pénalité n'est pas convexe. On introduit donc la relaxation convexe, notée variation totale (TV), en définissant

$$\hat{\beta}^{\mathrm{TV}}_{\lambda} \in \underset{\beta\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \hat{R}(\beta) + \lambda\|\beta\|_{\mathrm{TV}} \right\}, \text{ pour } \lambda > 0 \tag{1.14}$$

$$\text{où } \|\beta\|_{\mathrm{TV}} = \sum_{j=2}^{p} |\beta_j - \beta_{j-1}|. \tag{1.15}$$

Remarquons que la variation totale (TV) dépend de l'ordre des coefficients. Dans le cas où, comme dans la suite de cette thèse, les coefficients dépendent du temps, cet ordre est naturel. Cependant, pour certaines applications cet ordre n'est pas nécessaire. Nous explorons le cas d'une pénalité de ce type dans la partie Travaux en cours.

Si en plus d'un faible nombre de variation, on souhaite que la solution contienne peu de variables, on peut combiner la pénalité TV avec le Lasso. Cette combinaison porte le nom de Fused-Lasso (Tibshirani et al. (2005)).

$$\|\beta\|_{\mathrm{FL}} = \sum_{j=1}^{p} |\beta_j| + \sum_{j=2}^{p} |\beta_j - \beta_{j-1}|. \tag{1.16}$$

1.3.3 Groupes de variables

Il est fréquent que des coefficients aient une structure de groupe. C'est le cas par exemple des coefficients d'une variable catégorielle. Les solutions du lasso sur ce genre de variable risquent d'exclure une partie des facteurs et d'en conserver l'autre. Hors il est souvent préférable de conserver entièrement toutes les modalités ou au contraire de toutes les exclure. C'est pour cette raison que fut introduit de Group-Lasso (voir Yuan & Lin (2006)).

Soit β un vecteur constitué de p groupes de L_j coefficients, où chaque groupe est potentiellement constitué d'un nombre différent de coefficients. On note $\beta_{j,l}$ le l-ième coefficient du groupe j. Le Group-Lasso est alors définit de la manière suivante :

$$\hat{\beta}_{\lambda}^{\text{group-Lasso}} \in \underset{\beta \in \mathbb{R}^{\sum L_j}}{\operatorname{argmin}} \left\{ \hat{R}(\beta) + \lambda \sum_{j=1}^{p} \|\beta_j\|_2 \right\}, \text{ pour } \lambda > 0 \tag{1.17}$$

$$\text{où } \|\beta_j\|_2 = \left(\sum_{l=1}^{L_j} \beta_{j,l}^2 \right)^{1/2}. \tag{1.18}$$

Par la suite, nous serons amenés à utiliser une pénalité group, définie comme ci dessous

$$\|\beta\|_{g\,\text{FL}} = \sum_{j=1}^{p} \left(|\beta_{j,1}| + \sum_{l=2}^{L_j} |\beta_{j,l} - \beta_{j,l-1}| \right). \tag{1.19}$$

1.3.4 Sélection de support

L'estimateur obtenu par les approches pénalisées est biaisé. Par exemple, le lasso tend à pousser les coefficients vers zéro. En pratique, pour pallier ce phénomène certains auteurs proposent d'utiliser le lasso comme une méthode de sélection de variable et d'estimer une nouvelle fois le modèle non pénalisé uniquement sur les variables retenues par le lasso.

De manière plus générale, les approches pénalisés ont pour objectif de sélectionner le support des coefficient et non pas de déterminer la valeur optimale de ces derniers. Dans ce cadre, la procédure d'estimation est la suivante :

1. Calculer les coefficients pénalisés

$$\hat{\beta} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \hat{R}(\beta) + \text{pen}(\beta) \right\}. \tag{1.20}$$

2. En déduire le support des coefficients

$$S = \text{support}_{\text{pen}}(\beta). \tag{1.21}$$

3. Faire l'estimation (non pénalisée) sur le support

$$\tilde{\beta} \in \underset{\beta \in S}{\operatorname{argmin}}\, \hat{R}(\beta). \tag{1.22}$$

Cet estimateur n'en reste pas moins biaisé au sens de (1.5), à cause de la restriction au support qui n'est pas nécessairement le support optimal. Cependant, on observe en pratique que les coefficients obtenus sont moins biaisés. Cette méthode est donc couramment utilisée, bien qu'il n'existe pas, à notre connaissance, de preuves mathématiques de leur supériorité face aux estimateurs pénalisés.

1.4 Synthèse

Dans ce chapitre, nous avons vu que le choix d'un modèle et de ses paramètres repose sur la minimisation du risque empirique. Les approches pénalisées visent à contrôler la complexité d'un modèle en cherchant un compromis entre biais et variance, via l'ajout d'une pénalité. Nous avons aussi proposé d'utiliser une pénalité pour sélectionner le support des paramètre, avant d'effectuer une nouvelle estimation non pénalisé de ces derniers. L'estimation des paramètres de ces modèles constitue un enjeu majeur de l'apprentissage statistique. Nous abordons, dans le chapitre 2, des méthodes d'optimisation permettant d'estimer les paramètres de ces modèles lorsque les données sont de grande taille. Enfin, dans le chapitre 3, nous introduisons un modèle d'apprentissage statistique particulièrement adapté à l'analyse des temps d'évènements : l'analyse de survie.

Chapitre 2

Optimisation

Les estimateurs que nous avons vu dans le chapitre précédent sont obtenus par la minimisation d'une fonction, exprimée comme la somme d'une perte empirique et d'une pénalité. Nous présentons ici quelques méthodes qui permettent de trouver efficacement un minimum à ces fonctions. La recherche du minimum d'une fonction quelconque est un problème difficile et, d'une manière générale, il n'existe pas d'algorithme garantissant la convergence dans un tel cas. Les estimateurs que nous présentons dans cette thèse, comme beaucoup d'estimateurs de l'apprentissage statistique, reposent sur des fonctions convexes, pour lesquelles il existe des algorithmes efficaces.

Avec l'augmentation des volumes de données à analyser, l'optimisation prend une part de plus en plus importante dans le choix des algorithmes à utiliser. Si bien que l'on ne s'intéresse plus uniquement aux erreurs de biais et de variance, mais que l'on inclut désormais l'erreur d'optimisation (Bousquet & Bottou (2008), Roux et al. (2012)). Nous présentons dans ce chapitre quelques méthodes d'optimisation adaptées aux grands jeux de données et renvoyons le lecteur vers Boyd & Vandenberghe (2004) pour une présentation plus détaillée de l'optimisation convexe.

2.1 Descente de gradient

La méthode la plus ancienne (Cauchy (1847)) pour résoudre un problème d'optimisation de manière itérative est la descente de gradient. Soit f une fonction de $\mathbb{R}^n$ dans $\mathbb{R}$ que l'on cherche à minimiser. Notons $\nabla f(x)$ le gradient de f au point x et μ^k le pas de descente à l'itération k. Alors l'algorithme de descente de gradient est donné par l'algorithme 1.

Le choix de la suite des pas de descente (μ^k) est d'une importance cruciale pour la convergence de l'algorithme et nous présentons ici quelques unes des méthodes couramment employées.

Algorithme 1 Descente de gradient

Entrées : x^0 un point de départ, (μ^k) une suite de pas de descente
Sortie : x^k
  while on n'a pas convergé **do**
    $x^{k+1} = x^k - \mu^k \nabla f(x^k)$
  end while

2.1.1 Pas de descente et constante de Lipschitz

Le gradient fournit la direction de descente, mais ne nous informe pas sur la longueur du pas à effectuer dans cette direction. Des pas trop grands risquent de conduire à la divergence de l'algorithme, alors que des pas trop petits nécessiteraient un temps très grand pour converger. Si l'on considère le cas simple où le pas de descente est constant à chaque itération, et que le gradient de la fonction possède de bonnes propriétés, alors il est possible de calculer un pas de descente optimal.

Rappelons qu'une fonction f est dite L-lipschitzienne si pour tous points x et y de son espace de départ, on a

$$|f(x) - f(y)| \leq L|x - y|. \tag{2.1}$$

Si le gradient de la fonction à minimiser est L-Lipschitzien, alors on peut montrer que pour toute constante $\mu \leq \frac{1}{L}$, la descente de gradient converge. Parmi les pas constants d'une itération à l'autre, le cas $\mu = \frac{1}{L}$ est celui offrant la convergence la plus rapide. Malheureusement, les fonctions que nous allons considérer par la suite ne présentent pas la propriété d'avoir un gradient lipschitzien. Nous étudions donc, dans la suite de ce chapitre, d'autres méthodes pour choisir la suite des pas de descente.

2.1.2 Approximation du second ordre : Newton Raphson

Supposons qu'au point courant x^k, la fonction soit deux fois dérivable. Alors, un développement de Taylor d'ordre 2 nous donne

$$f(x) \simeq f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{1}{2}(x - x^k)^T \nabla^2 f(x^k)(x^{k+1} - x^k) \tag{2.2}$$

où $\nabla^2 f(x^k)$ est la matrice hessienne de f en x^k.

Le minimum de cette approximation quadratique peut être utilisé à chaque itération de la descente de gradient.

$$x^{k+1} = \underset{x}{\operatorname{argmin}}\, f(x) = x^k - (\nabla^2 f(x^k))^{-1} \nabla f(x^k). \tag{2.3}$$

Cet algorithme porte le nom de Newton-Raphson et constitue une alternative à la descente de gradient où le pas de descente est remplacé par l'inverse de la matrice hessienne au point x^k.

Cependant, l'inversion de la matrice hessienne prend un temps considérable lorsque le nombre d'observations et de variables est grand. D'autant que ce calcul intervient à chaque itération de la descente de gradient.

Dans la pratique, pour les problèmes de grande dimension, il est possible d'approximer $\nabla^2 f(x^k)$ par une matrice diagonale, en utilisant par exemple la diagonale de la hessienne. Une autre méthode consiste à effectuer une recherche linéaire sur le pas de descente.

2.1.3 Recherche linéaire

A priori on pourrait se contenter d'accepter n'importe quel pas de descente tant que celui ci entraine une décroissance de la fonction, i.e. $f(x^{k+1}) = x^k - \mu^k \nabla f(x^k) < f(x^k)$. Cependant un pas de descente infinitésimal entrainerait une toute petite variation de la fonction et nécessiterait un très grand nombre d'itérations pour converger. Par ailleurs, un pas de descente "trop grand" peut conduire à des oscillations de la fonction et donc aussi à une convergence très lente. Enfin, notons qu'une séquence de pas de descente mal choisie peut bloquer l'algorithme, sans que celui ci ne converge, voir figure 2.1. Il est donc nécessaire d'introduire des conditions supplémentaires pour faciliter la convergence de l'algorithme.

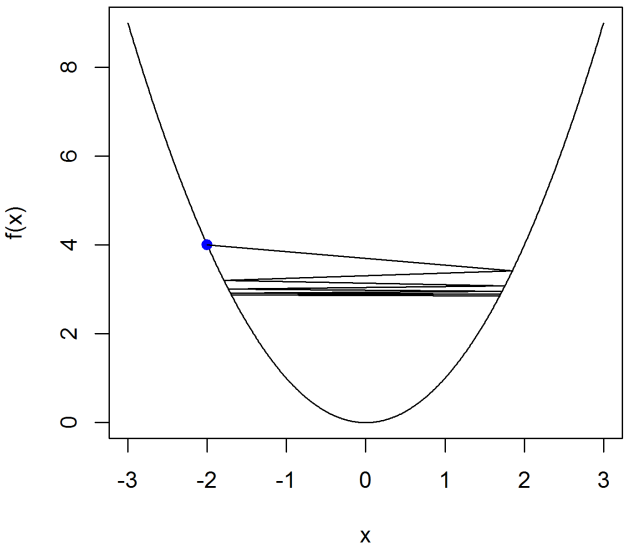

FIGURE 2.1 – Oscillation et non convergence

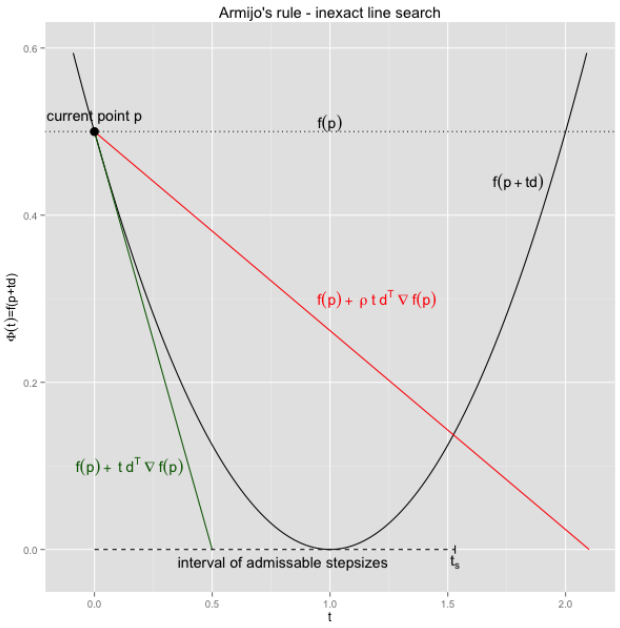

FIGURE 2.2 – Condition d'Armijo. Source [1]

2.1.4 Condition de Wolfe

Dans le but d'éviter les oscillations autour le l'optimum, Armijo propose la condition suivante, sur le pas de descente, afin d'éviter qu'il ne soit trop grand.

$$f(x - \mu \nabla f(x)) \leq f(x) - c_1 \mu ||\nabla f(x)||_2^2. \tag{2.4}$$

L'algorithme de recherche linéaire, visant à déterminer le pas de descente à utiliser à chaque itération de la descente de gradient est donc le suivant.

Algorithme 2 Recherche linéaire

Entrées : $0 < \alpha < 1$, μ_{init}, $0 < c_1 < 1$
Sortie : μ

$\mu = \mu_{init}$
while $f(x - \mu \nabla f(x)) > f(x) - \mu c_1 ||\nabla f(x)||_2^2$ **do**
  $\mu = \alpha \mu$.
end while

La figure 2.2 illustre l'effet de l'algorithme : la valeur du pas de descente ne peut pas dépasser t_s.

Si l'évaluation de f prend du temps, mieux vaut choisir un coefficient de réduction α assez petit pour qu'il n'y ait pas trop d'évaluations de la fonction dans la recherche linéaire.

1. https://blog.codecentric.de/files/2015/03/ArmijoRule.png

2.2 Méthodes proximales

Dans le chapitre sur l'apprentissage statistique, nous avons vu que les méthodes modernes reposent sur des approches pénalisées dans lesquelles la fonction à minimiser est la somme de deux fonctions, l'une différentiable et l'autre non.

Cette fonction à minimiser prend la forme

$$f(x) = g(x) + h(x) \tag{2.5}$$

où :

— g est convexe et différentiable

— h est convexe, mais pas nécessairement différentiable

Pour minimiser ces fonctions, on utilise quand c'est possible l'opérateur proximal défini comme suit :

$$\text{prox}_{\mu}^{h}(x) = \underset{z}{\text{argmin}} \frac{1}{2\mu}||x - z||_2^2 + h(z). \tag{2.6}$$

La descente de gradient proximale s'écrit alors :

Algorithme 3 Descente de gradient proximale

Entrées : $x^{(0)}$ et (μ^k) une suite de pas de descente
Sortie : x^k
  while on n'a pas convergé **do**
    $x^{(k+1)} = \text{prox}_{\mu_k}^{h}\left(x^k - \mu_k \nabla g(x^k)\right)$
  end while

Remarque : On peut aussi écrire la descente de gradient proximale sous la forme d'une descente de gradient classique en utilisant les notations suivantes

$$x^{(k+1)} = x^k - \mu_k \cdot G_{\mu_k}(x^k) \tag{2.7}$$

où G_μ est le gradient généralisé de f :

$$G_\mu(x) = \frac{x - \text{prox}_{\mu}^{h}\left(x - \mu \nabla g(x)\right)}{\mu}. \tag{2.8}$$

2.2.1 Exemple : ISTA pour le problème des moindres carrés

Un exemple d'application de l'algorithme de descente de gradient proximal est l'algorithme ISTA (Iterative Soft-Thresholding Algorithm Beck & Teboulle (2009)), que nous détaillons ici pour le problème des moindres carrés pénalisés par le Lasso.

Soient $y \in \mathbb{R}^n$ et $X \in \mathbb{R}^{n \times p}$. On cherche à résoudre le problème des moindres carrés avec pénalisation Lasso :

$$\begin{aligned} f(\beta) = & \quad g(\beta) \quad + \quad h(\beta) \\ & = \frac{1}{2}||y - X\beta||_2^2 + \lambda||\beta||_1. \end{aligned} \tag{2.9}$$

Rappelons que $\nabla g(\beta) = -X^T(y - X\beta))$. L'opérateur proximal de h : $h(\beta) = \lambda||\beta||_1$ s'écrit :

$$\begin{aligned} \text{prox}_{\lambda,\mu}^{L1}(\beta) &= \underset{z}{\text{argmin}} \frac{1}{2\mu}||\beta - z||_2^2 + \lambda||z||_1 \\ &= \underset{z}{\text{argmin}} \frac{1}{2}||\beta - z||_2^2 + \mu\lambda||z||_1 \\ &= S_{\mu\lambda}(\beta) \end{aligned} \tag{2.10}$$

où $S_{\mu\lambda}(\beta)$ est l'opérateur de soft-thresholding (Donoho (1995)), défini par :

$$S_{\mu\lambda}(\beta) = \begin{cases} \beta - \mu\lambda & \text{si } \beta > \mu\lambda \\ 0 & \text{si } -\mu\lambda < \beta < \mu\lambda \\ \beta + \mu\lambda & \text{si } \beta < -\mu\lambda \end{cases}. \tag{2.11}$$

Ici nous avons volontairement mis en évidence la dépendance au pas de descente μ. Une mise a jour de l'algorithme de gradient proximal est donc :

$$\beta^{k+1} = S_{\mu\lambda}\left(\beta^k + \mu X^T(y - X\beta^k)\right). \tag{2.12}$$

2.2.2 Interprétation

Une manière intuitive d'interpréter la descente de gradient proximale est de la voir comme une projection sur l'espace des contraintes. La recherche du minimum de l'équation (2.9) est équivalente à la recherche du minimum de l'équation suivante

$$\frac{1}{2}||y - X\beta||_2^2 \text{ , tel que } \|\beta\|_1 \leq \alpha. \tag{2.13}$$

La descente de gradient proximale consiste à projeter la solution obtenue par descente de gradient sur le point appartenant aux contraintes le plus proche, voir figure 2.3.

2. http://www.stat.cmu.edu/~ryantibs/convexopt/

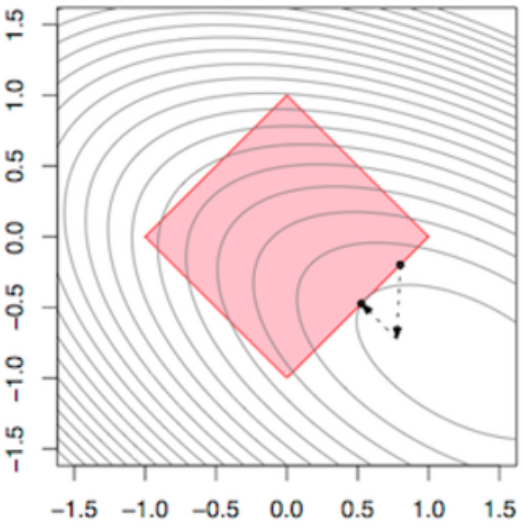

FIGURE 2.3 – Descente de gradient proximale. Source [2]

2.2.3 Recherche linéaire proximale

Dans le cas d'une descente de gradient proximale, la condition à respecter sur le pas de descente lors de la recherche linéaire devient :

$$g(x - \mu G_\mu(x)) \leq g(x) - \mu \nabla g(x)^T G_\mu(x) + \frac{\mu}{2} ||G_\mu(x)||_2^2. \tag{2.14}$$

Remarquons que $G_\mu(x)$ doit être recalculé à chaque étape de la recherche linéaire et donc que l'opérateur proximal est appelé à chaque fois.

2.3 Méthode du gradient accéléré

La vitesse de convergence de la descente de gradient, pour les fonctions dont le gradient est Lipschitz, est de l'ordre de $1/k$. Ce qui signifie qu'en doublant le nombre d'itérations de la descente de gradient, on réduit la distance entre le point d'origine et le minimum d'un facteur deux. Or pour des fonctions suffisamment régulières l'ordre de convergence des méthodes d'ordre un (qui utilisent uniquement l'information du gradient et non de la hessienne) est de $1/k^2$. Dans ce cas, doubler le nombre d'itérations réduit la distance au minimum par quatre. Pour atteindre de telle vitesse de convergence, Nesterov (2007) propose une descente de gradient accélérée.

2.3.1 Accélération

La descente de gradient ne conserve aucune information sur les itérations précédentes. Or la connaissance des directions de descente utilisées lors des itérations précédentes pour-

rait fournir une information permettant d'accélérer la vitesse de convergence de l'algorithme. C'est pourquoi Nesterov (2007) propose de combiner le gradient courant avec celui des itérations précédentes. Plusieurs versions existent quand aux poids utilisés pour combiner les différents gradients. Nous utilisons ici la version de Su et al. (2014).

Algorithme 4 Méthode du gradient accélérée

Entrées : $x^{(0)}$ et (μ^k) une suite de pas de descente
Sortie : x^k
  $x^1 = x^0$
  while on n'a pas convergé **do**
    $v = x^k + \frac{k-1}{k+2}\left(x^k - x^{(k-1)}\right)$
    $x^{(k+1)} = v - \mu_k \nabla g(v)$
  end while

L'idée est de faire une moyenne mobile des itérations précédentes de sorte à garder une certaine inertie dans les directions précédemment parcourues. On peut aussi voir cela comme le déplacement d'une balle sur la surface de la fonction à minimiser. Au départ la balle a une vitesse nulle. A chaque instant, son accélération est donnée par le gradient de la fonction. Avec la descente de gradient classique, entre chaque itération la vitesse de la balle est remise à zéro. Alors que la descente de gradient accélérée impose à la balle de conserver une partie de la vitesse acquise lors des itérations précédentes.

2.3.2 Accélération proximale

Beck & Teboulle (2009) ont adapté la méthode du gradient accéléré au cas proximal.

Algorithme 5 Méthode du gradient proximal accéléré

Entrées : $x^{(0)}$ et (μ^k) une suite de pas de descente
Sortie : x^k
  $x^1 = x^0$
  while on n'a pas convergé **do**
    $v = x^k + \frac{k-1}{k+2}\left(x^k - x^{(k-1)}\right)$
    $x^{(k+1)} = \text{prox}_{\mu_k}\left(v - \mu_k \nabla g(v)\right)$
  end while

2.3.3 Recherche linéaire accélérée proximale

Enfin, il est possible d'adapter la recherche linéaire pour la méthode du gradient proximal. Celle ci est donnée par l'algorithme 6.

Algorithme 6 Recherche linéaire proximale accélérée

Entrées : $0 < \alpha < 1$, μ^{k-1}, $0 < c_1 < 1$, v voir algo 5
Sortie : μ
  $\mu = \mu^{k-1}$
  $x^+ = \text{prox}_\mu (v - \mu \nabla g(v))$
  while $g(x^+) > g(v) - \nabla g(v)^T (x^+ - v) + \mu c_1 ||x^+ - v||_2^2$ **do**
    $\mu = \alpha \mu$.
    $x^+ = \text{prox}_\mu^h (v - \mu \nabla g(v))$
  end while

2.4 Descente de gradient stochastique

2.4.1 Algorithme du gradient stochastique

Nous avons vu dans le chapitre 1 que le choix d'un estimateur reposait sur la minimisation du risque empirique. Or le gradient du risque empirique est constitué d'une somme de fonctions indépendantes sur chaque individus. Plutôt que de calculer le gradient sur l'ensemble des individus, la descente de gradient stochastique se contente de l'évaluer sur une seule observation (ou un petit sous-ensemble d'observations), tirée de manière aléatoire à chaque itération. L'avantage est que l'évaluation d'un gradient individuel a un coût de l'ordre de $o(p)$ dans le cas d'une régression linéaire, contre $o(np)$ pour le gradient complet. Cependant cette méthode est bruitée, puisque chaque individu n'est pas représentatif de l'ensemble de la population. Contrairement à la descente de gradient, la descente de gradient stochastique n'est pas décroissante à chaque itération, voir figure 2.4. Sur le plan théorique, la descente de gradient stochastique converge en espérance vers la solution (cf. Bottou (1998), Kiwiel (2001)). Sur le plan pratique, elle offre des performances supérieures à la descente de gradient complète dans de nombreux cas, notamment lorsque le nombre d'observations est très grand.

L'algorithme de descente de gradient stochastique est décrit par l'algorithme 7.

Algorithme 7 Descente de gradient stochastique

Entrées : x^0, (μ^k) une suite de pas de descente
Sortie : x^k
  while on n'a pas convergé **do**
    choisir aléatoirement i parmi $1..n$
    $x^{k+1} = x^k - \mu^k \nabla f_i(x^k)$
  end while

où $\nabla f_i(x^k)$ est le gradient calculé à partir de l'observation i et $\nabla f(x^k) = \sum_{i=1}^n \nabla f_i(x^k)$.

En pratique, pour les grands jeu de données, la descente de gradient stochastique converge plus rapidement vers un voisinage de la solution et nécessite de parcourir moins de fois les observations que la descente de gradient complète Bousquet & Bottou (2008),

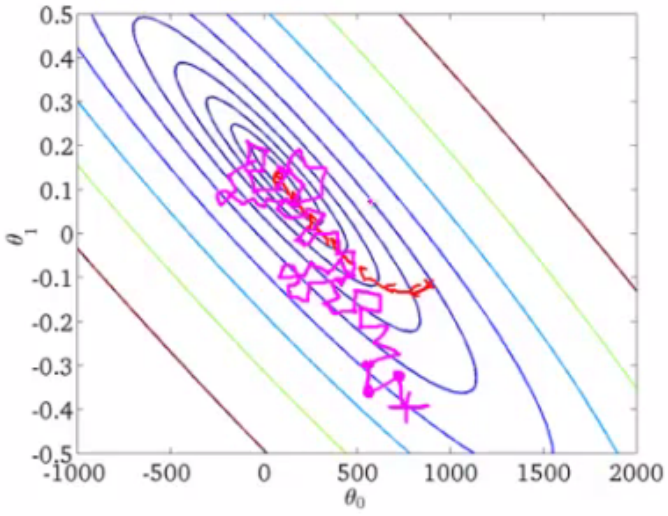

FIGURE 2.4 – Illustration de la descente de gradient complète en rouge et stochastique en rose. Source [3]

Bottou & Le Cun (2005). Il est d'ailleurs possible d'arrêter l'optimisation à ce moment là, puisque certains voient un lien entre la prolongation de l'optimisation et le sur-apprentissage. Cependant, si l'on souhaite poursuivre l'optimisation jusqu'à obtenir un gradient plus proche de zéro, alors l'approche la plus rapide est de finir avec quelques étapes de gradient complet.

Notons enfin, qu'avec une descente de gradient stochastique il n'est pas raisonnable d'utiliser une recherche linéaire, puisque l'évaluation de la fonction coût requiert n fois plus d'opération que l'évaluation d'un gradient individuel $\nabla f_i(x^k)$. D'autre part, $f_i(x)$ est une estimation bien trop bruitée de la fonction à minimiser pour être utilisée dans la recherche linéaire.

En conclusion, si la recherche linéaire constitue l'approche privilégiée pour choisir les pas de descente de la descente de gradient complet, cette méthode n'est pas performante sur la descente de gradient stochastique. D'autres méthodes, dites adaptatives, on été proposés ces dernières année pour accélérer la convergence de la descente de gradient stochastique.

2.4.2 AdaGrad

La descente de gradient stochastique classique utilise le même pas de descente pour tous les paramètres du modèle. Or certaines variables sont sans doute plus sujette que d'autres à de fortes variations à travers les observations. L'idée d'AdaGrad (Duchi et al. (2011)) est d'individualiser l'actualisation des pas de descente en fonction des gradients précédents, plus précisément en fonction du moment d'ordre 2 des précédents gradients.

3. http://www.holehouse.org/mlclass/17_Large_Scale_Machine_Learning.html

Algorithme 8 AdaGrad

Entrées : x^0, μ
Sortie : x^k
  initialiser les v_j à zéro.
  while on n'a pas convergé **do**
    choisir aléatoirement i parmi $1..n$
    for $j \in 1..p$ **do**
      $v_j = v_j + (\nabla f_i(x_j^k))^2$
      $x_j^{k+1} = x_j^k - \frac{\mu}{\sqrt{v_j}} \nabla f_i(x_j^k)$
    end for
  end while

2.4.3 AdaDelta

Dans AdaDelta (Zeiler (2012)), la somme sur les moments d'ordre 2 est remplacée par une moyenne mobile. Ceci évite l'accumulation de nombreuses valeurs positives pouvant conduire à des actualisations très petites et donc à l'arrêt prématuré de la descente de gradient stochastique.

Algorithme 9 AdaDelta

Entrées : x^0, γ
Sortie : x^k
  initialiser les v_j et w_j à zéro.
  $x^1 = x^0$
  while on n'a pas convergé **do**
    choisir aléatoirement i parmi $1..n$
    for $j \in 1..p$ **do**
      $v_j = \gamma v_j + (1-\gamma)(\nabla f_i(x_j^k))^2$
      $w_j = \gamma w_j + (1-\gamma)(x_j^k - x_j^{k-1})$
      $x_j^{k+1} = x_j^k - \frac{\sqrt{w_j}}{\sqrt{v_j}} \nabla f_i(x_j^k)$
    end for
  end while

2.4.4 ADAM

Enfin plusieurs auteurs ont suggéré d'ajouter de l'information sur le moment d'ordre 1 (voir par exemple Schaul et al. (2012)). Nous présentons ici l'algorithme ADAM (Kingma & Ba (2014)), qui est l'un des plus utilisé au moment où nous écrivons cette thèse.

Algorithme 10 ADAM

Entrées : x^0, γ_1, γ_2
Sortie : x^k
  initialiser les m_j et v_j à zéro.
  $x^1 = x^0$
  while on n'a pas convergé **do**
    choisir aléatoirement i parmi $1..n$
    for $j \in 1..p$ **do**
      $m_j = \gamma_1 m_j + (1 - \gamma_1)\nabla f_i(x_j^k)$
      $v_j = \gamma_2 v_j + (1 - \gamma_2)(\nabla f_i(x_j^k))^2$
      $\hat{m}_j = \frac{m_j}{1-\gamma_1}$
      $\hat{v}_j = \frac{v_j}{1-\gamma_2}$
      $x_j^{k+1} = x_j^k - \frac{\sqrt{\hat{m}_j}}{\sqrt{\hat{v}_j}}\nabla f_i(x_j^k)$
    end for
  end while

2.5 Synthèse

Nous avons vu à travers ce chapitre quelques unes des méthodes d'optimisation adaptées à la fois aux grands jeux de données et permettant de prendre en compte les différentes pénalités. Parmi les modèles présentés, nous retiendrons la descente de gradient proximale accélérée avec des pas de descente déterminés par une recherche linéaire, qui est particulièrement adaptée pour les données de taille moyenne et facilement accessibles en une fois. Lorsque les données sont de grande taille et/ou accessibles par block (typiquement lors d'un apprentissage *en ligne* ou via un accès séquentiel) nous privilégierons une descente de gradient stochastique aux pas adaptatifs.

Dans le chapitre suivant, nous introduisons des modèles d'apprentissage statistique capable de tirer pleinement partie des méthodes d'optimisation présentés tout au long de ce chapitre. Ces méthodes d'optimisation sont implémentées dans le package *CoxTV*, présenté au chapitre 7 et sont utilisés dans les chapitres 8 et 9 afin de mieux comprendre le lien entre les éléments constitutifs d'un jeu vidéo et le temps de jeu des joueurs.

Chapitre 3

Analyse de survie et processus de comptage

3.1 Definitions

L'analyse de survie est une partie de l'apprentissage statistique supervisé dans laquelle les labels, ou les observations à prédire, sont des temps d'évènements et permettant de prendre en compte des données censurées. Un événement peut prendre plusieurs formes. Traditionnellement on s'intéresse au décès d'un individu, c'est pourquoi les termes relatifs à la santé sont restés dans la littérature. Cependant on peut tout aussi bien s'intéresser à la rémission d'une maladie, la panne d'un composant ou, comme c'est le cas dans cette thèse, un joueur qui arrête définitivement de jouer à un jeu particulier. Nous présentons ici le cadre de l'analyse de survie et celui des processus de comptage.

Une des particularités des variables de durée est que l'évènement considéré n'a pas forcément été observé au moment où l'étude prend fin. On parle alors de censure. Définissons à présent les notations utilisées en analyse de survie.

Soit $T^\star$ et C deux variables aléatoires positives et indépendantes, représentant respectivement le temps de réalisation de l'évènement d'intérêt et le temps de censure (moment à partir du quel on n'a plus d'observation). Si un évènement intervient après le temps de censure, il n'est pas observé. En pratique, on observe donc

$$T = \min(T^\star, C) \text{ et } \delta = \mathbb{1}(T^\star \leq C). \tag{3.1}$$

L'objectif de l'analyse de survie est d'inférer sur la distribution de $T^\star$ à partie d'une collection d'observations de n couples (t_i, δ_i), issues de la loi jointe de T et δ.

Une approche alternative, proposée par Aalen (Aalen (1978)) consiste à modéliser les temps d'évènement et de censure via des processus stochastiques. Dans ce formalisme, les processus N et Y sont définis comme :

$$N(t) = \mathbb{1}(T \leq t, \delta = 1) \text{ et } Y(t) = \mathbb{1}(T \geq t), \text{ pour } t \geq 0. \tag{3.2}$$

Dans le cas des processus de comptage ou des modèles de survie, la période d'observation est limitée dans le temps ; nous noterons τ cette limite. Les observations, pour l'individu i, considérées jusqu'au temps τ de fin d'étude sont $(N_i(s), Y_i(s), 0 \leq s \leq \tau)$. La décomposition de Doob-Meyer (Meyer et al. (1962)) indique que le processus de comptage N peut se décomposer de façon unique comme la somme d'une martingale M et d'un compensateur Λ, tel que

$$N(t) = \Lambda(t) + M(t), \text{ pour tout } t \geq 0. \tag{3.3}$$

Si $T^\star$ admet une densité $f^\star$ et une fonction de répartition $F^\star$, alors le compensateur de N s'écrit :

$$\Lambda^\star(t) = \int_0^t \alpha^\star(s) Y(s) ds, \text{ pour tout } t \geq 0 \tag{3.4}$$

la fonction $\alpha^\star$ apparaissant dans le compensateur est aussi le taux de risque instantané de $T^\star$, que l'on peut définir comme :

$$\alpha^\star(t) = \frac{f^\star(t)}{1 - F^\star(t)}. \tag{3.5}$$

3.2 Vraisemblance d'un processus de comptage

La construction de la vraisemblance d'un processus de comptage général est détaillée dans Martinussen & Scheike (2007)(p.62). Nous la présentons à nouveau ci dessous.

Soit un processus de comptage N^* adapté à une filtration $(\mathcal{F}_t^*)$ et d'intensité λ^*. Notons le i-eme saut de N^* par t_i^*, à valeur infinie si $N^\star$ ne réalise pas i sauts. Par convention, posons $t_0^* = 0$. Le compensateur, défini pour tout t par $\Lambda^*(t) = \int_0^t \lambda^*(s) ds$ fait de $M^* = N^* - \Lambda^*$ une martingale. L'intensité peut prendre une forme paramétrique $\lambda^*(t) = \lambda^*(t, \beta)$ que nous n'expliciterons pas pour l'instant.

La vraisemblance d'un processus de comptage observé jusqu'au temps τ est donnée par :

$$\begin{aligned}
\mathcal{L}(\lambda) &= \left\{ \prod_{t_i^* \leq \tau} \lambda(t_i^*) \right\} \exp\left(- \int_0^\tau \lambda(s) ds \right) \\
&= \left\{ \prod_{t_i^* \leq \tau} \exp\left(- \int_{t_{i-1}^*}^{t_i^*} \lambda(s) ds \right) \lambda(t_i^*) \right\} \exp\left(- \int_{t_N^*}^\tau \lambda^*(s) ds \right).
\end{aligned} \tag{3.6}$$

où t_N^* correspond au dernier saut.

La dernière équation nous donne un aperçu de l'intuition cachée derrière cette vraisemblance. Dans le produit, chaque terme contribue à la probabilité qu'aucun évènement ne se réalise entre $[t_{i-1}^*, t_i^*[$ et qu'un évènement ait lieu au temps t_i^*, le tout conditionnellement au passé du processus. Le dernier terme contribue à la probabilité qu'il n'y ait aucun évènement jusqu'à la fin de la période d'observation, conditionnellement au passé.

Dans le cas des processus N et Y définis au paragraphe précédent, comme N saute au plus une fois, la vraisemblance associée à l'observation de N sur $[0, \tau]$ est donnée par

$$\begin{aligned} &= \begin{cases} \lambda(T) \exp\left(-\int_0^\tau \lambda(s)ds\right) & \text{si } T \leq \tau \\ \exp\left(-\int_0^\tau \lambda(s)ds\right) & \text{sinon} \end{cases} \\ &= \exp\left(\delta \log\left(\lambda(T)\right) - \int_0^\tau \lambda(s)ds\right) \end{aligned} \tag{3.7}$$

3.3 Vraisemblance du modèle de Cox

En pratique, on observe en même temps que les processus N et Y, des covariables qui peuvent dépendre du temps. Nous donnerons des exemples de telles covariables dans les chapitres suivants. Nous notons X le processus de covariables, qui correspond à l'observation au cours du temps de p covariables, de sorte que X contient p fonctions. Le modèle le plus célèbre qui lie la valeur des covariables à l'intensité est le modèle de Cox (Cox (1972)). Dans ce modèle, l'intensité prend la forme suivante

$$\lambda^\star(t) = Y(t) \exp\left(X^T(t)\beta^\star(t)\right) \tag{3.8}$$

que l'on peut aussi écrire

$$\lambda^\star(t) = Y(t)\lambda_0^\star(t) \exp\left(X^T(t)\beta^\star(t)\right). \tag{3.9}$$

Dans le premier cas, la baseline $\lambda_0^\star$ est incluse dans l'exponentielle, i.e le premier coefficient de β est tel que $\lambda_0 = \exp(\beta_0)$ et l'on a ajouté un vecteur unité dans la matrice X. Cette fonction $\lambda_0^\star$ correspond a un intercept, qui dépend du temps.

Rappelons qu'ici les variables X dépendent du temps, ce qui permet de prendre en compte les trajectoires temporelles. D'autre part, cette modélisation inclue aussi le cas des variables constantes au cours du temps. Notons que les processus X_i de chaque individu i sont de dimension p et sont observés au même moment que N_i et Y_i. En conséquence, on ne dispose pas d'information sur les individus au delà de leurs période d'observation, e.g après leur censure.

Dans le cadre du modèle de Cox, pour n individus indépendants, la vraisemblance s'écrit :

$$\begin{aligned} \mathcal{L}(\beta) &= \prod_{i=1}^n \exp\left(\delta_i \log\left(\lambda(T_i)\right) - \int_0^\tau \lambda(s)ds\right) \\ &= \prod_{i=1}^n \exp\left(\delta_i \log\left(\lambda(T_i)\right) - \int_0^\tau Y_i(s) \exp\left(X_i^T(s)\beta(s)\right) ds\right). \end{aligned} \tag{3.10}$$

En pratique, on minimise l'opposé de la log vraisemblance, notée $\ell_n(\beta)$, que l'on peut aussi écrire, avec les notations des processus de comptage introduite dans le paragraphe

3.1, comme :

$$\ell_n(\beta) = -\frac{1}{n}\sum_{i=1}^{n}\Big\{\int_0^\tau X_i^T(t)\beta(t)dN_i(t) - \int_0^\tau Y_i(t)\exp\big(X_i^T(t)\beta(t)\big)dt\Big\}. \tag{3.11}$$

Cette vraisemblance correspond à la vraisemblance *complète* du modèle de Cox. Nous montrons dans le paragraphe suivant comment en déduire la vraisemblance *partielle* de Cox.

3.4 Vraisemblance partielle de Cox

La vraisemblance partielle de Cox offre l'avantage de ne pas spécifier, ni estimer la baseline. On parle alors de modèle semi-paramétrique. Nous présentons ici le cas où les variables et les coefficients dépendent du temps et montrons comment la vraisemblance partielle découle de la vraisemblance *complète*.

Pour bien visualiser la disparition de la baseline dans la vraisemblance partielle sortons la de la matrice de design. L'intensité s'écrit alors :

$$\lambda(t) = Y(t)\lambda_0(t)\exp(X^T(t)\beta(t)).$$

L'opposé de la log-vraisemblance de l'équation (3.11) devient

$$\begin{aligned}
\ell_n(\beta) &= \frac{-1}{n}\sum_{i=1}^{n}\left\{\int_0^\tau \log\big(\lambda_0(t)\exp(X_i^T(t)\beta(t))\big)\,dN_i(t) - \int_0^\tau Y_i(t)\lambda_0(t)\exp(X_i^T(t)\beta(t))dt\right\}\\
&= \frac{-1}{n}\sum_{i=1}^{n}\left\{\int_0^\tau \log\big(\lambda_0(t)\exp(X_i^T(t)\beta(t))\big)\,dN_i(t)\right\}\\
&\quad + \frac{1}{n}\int_0^\tau \lambda_0(t)\sum_{i=1}^{n} Y_i(t)\exp(X_i^T(t)\beta(t))dt.
\end{aligned} \tag{3.12}$$

En notant $S_n(t) = \sum_{i=1}^n Y_i(t)\exp(X_i^T(t)\beta(t))$, on a

$$\begin{aligned}
\ell_n(\beta) &= \frac{-1}{n}\sum_{i=1}^{n}\left\{\int_0^\tau \log\big(\lambda_0(t)\exp(X_i^T(t)\beta(t))\big)\,dN_i(t)\right\} + \frac{1}{n}\int_0^\tau \lambda_0(t)S_n(t)dt\\
&= \frac{-1}{n}\sum_{i=1}^{n}\left\{\int_0^\tau \log\left(\lambda_0(t)\frac{\exp(X_i^T(t)\beta(t))}{S_n(t)}S_n(t)\right)dN_i(t)\right\} + \frac{1}{n}\int_0^\tau \lambda_0(t)S_n(t)dt\\
&= \frac{-1}{n}\sum_{i=1}^{n}\left\{\int_0^\tau \log\big(\lambda_0(t)S_n(t)\big)\,dN_i(t)\right\} - \frac{1}{n}\sum_{i=1}^{n}\left\{\int_0^\tau \log\left(\frac{\exp(X_i^T(t)\beta(t))}{S_n(t)}\right)dN_i(t)\right\}\\
&\quad + \frac{1}{n}\int_0^\tau \lambda_0(t)S_n(t)dt.
\end{aligned} \tag{3.13}$$

Dans le cas où les covariables ne dépendent pas du temps et où les processus N_i sautent au plus une fois (comme dans le paragraphe 3.1), le second terme correspond à la log-vraisemblance partielle de Cox.

$$\begin{aligned}\ell_n^P(\beta) &= \frac{-1}{n}\sum_{i=1}^{n}\left\{\int_0^\tau \log\left(\frac{\exp(X_i^T\beta)}{S_n(t)}\right) dN_i(t)\right\} \\ &= \frac{-1}{n}\sum_{i\in D}\log\left(\frac{\exp(X_i^T\beta)}{\sum_{j\in R_i}\exp(X_i^T\beta)}\right). \end{aligned} \tag{3.14}$$

où D est l'ensemble des individu ayant réalisé un évènement avant τ, et R_i l'ensemble des individus à risque au moment où l'individu i réalise l'évènement t_i^*.

Intuitivement : de l'intégrale sur $dN_i(t)$, on ne garde que le moment où l'individu i réalise un évènement, i.e à $t = t_i^*$. Au dénominateur, il ne reste plus que les individus toujours à risque à $t = t_i^*$, i.e $i \in R_i$

La vraisemblance partielle de Cox est la méthode privilégiée par défaut pour faire l'estimation des paramètres du modèle de Cox. Bien que cette vraisemblance partielle offre de nombreux avantages, nous allons voir dans les paragraphes suivants qu'elle n'est pas adaptée aux algorithmes d'optimisation de type descente de gradient stochastique.

3.5 Estimation des paramètres de la vraisemblance partielle de Cox

Nous montrons ici comment estimer les paramètres du modèles de Cox à partir de la vraisemblance partielle. Pour alléger les notation, nous nous plaçons dans le cas où les variables ne dépendent pas du temps. Cependant ces résultats restent vrai dans le cas de variables qui dépendent du temps.

Nous avons vu dans le chapitre 2 que les méthodes d'optimisation que nous utilisons reposent sur le gradient de la fonction à minimiser. Le gradient de l'opposé de la log-vraisemblance partielle de Cox s'écrit :

$$\Delta\ell_n^P(\beta) = \frac{1}{n}\sum_{i\in D}\left[-X_i + \sum_{j\in R_i}\frac{X_j\exp(X_j^T\beta)}{\sum\limits_{k\in R_i}\exp(X_k^T\beta)}\right]. \tag{3.15}$$

A première vu, le produit et la somme laissent à penser que le gradient requiert $o(n^2p)$ opérations. Cependant la formule récursive suivante permet de calculer le gradient en $o(np)$ opération.

Supposons que les temps d'évènement et de censure sont ordonnées de façon décroissante. Soit $(R_i)_{i\in D}$ une séquence croissante, i.e. $R_j \subset R_{j+1}$, alors on a :

$$\sum_{j \in R_{i+1}} \exp(X_j^T \beta) = \sum_{j \in R_i} \exp(X_j^T \beta) + \sum_{j \in R_{i+1} \setminus R_j} \exp(X_j^T \beta). \tag{3.16}$$

Soit x_1 la valeur des covariables au dernier temps d'évènement, x_2 leur valeur au temps précédent, etc. On a alors :

$$\begin{aligned} n\Delta\ell_n^P(\beta) = &- X_1 + \frac{X_1 \exp(X_1^T \beta)}{\exp(X_1^T \beta)} \\ &- X_2 + \frac{X_1 \exp(X_1^T \beta) + X_2 \exp(X_2^T \beta)}{\exp(X_1^T \beta) + \exp(X_2^T \beta)} \\ &\dots \end{aligned} \tag{3.17}$$

D'où émerge une écriture récursive :

$$\begin{aligned} n\Delta\ell_n^P(\beta) = &- X_1 + \frac{S_1^1}{S_2^1} \\ &- X_2 + \frac{X_2 \exp\left(X_2^T \beta\right) + S_1^1}{\exp\left(X_2^T \beta\right) + S_2^1} \\ &\dots \\ &- X_i + \frac{X_i \exp\left(X_i^T \beta\right) + S_1^{i-1}}{\exp\left(X_i^T \beta\right) + S_2^{i-1}} \\ &\dots \end{aligned} \tag{3.18}$$

où l'on a pour $i \in i_m, ..., i_1$

$$S_1^i = X_i \exp(X_i^T \beta) + S_1^{i-1} \tag{3.19}$$

$$S_2^i = \exp(X_i^T \beta) + S_2^{i-1} \tag{3.20}$$

$$S^k = \frac{S_1^k}{S_2^k}. \tag{3.21}$$

Le gradient s'écrit donc comme une somme sur les temps d'évènements

$$\Delta\ell_n^P(\beta) = \frac{1}{n} \sum_{i \in D} \left[-X_i + \frac{S_1^i}{S_2^i} \right]. \tag{3.22}$$

et sont évaluation ne nécessite que $o(np)$ opérations.

La descente de gradient complète offre donc de bonnes performances sur la vraisemblance partielle de Cox. Nous verrons dans le chapitre 5 que la descente de gradient stochastique ne peut être utilisée de façon efficace sur la vraisemblance partielle de Cox. En conséquence l'apprentissage ne peut se faire *online* et on ne peut profiter des performance des algorithmes de descente de gradient stochastique sur les grands jeux de données.

3.6 Synthèse

L'analyse de survie et les processus de comptage permettent de modéliser la durée d'occurrence d'un ou plusieurs évènements et de déterminer la relation entre cette durée et des variables explicatives. L'application de ces modèles sur les données issues de l'industrie du jeux vidéo pourrait nous permettre de mieux comprendre le lien entre les éléments de design d'un jeu vidéo et le comportement des joueurs. Cependant, leur application à de très grands jeux de données reste un challenge, comme nous le verrons dans le paragraphe 5.1.

L'application d'un modèle statistique à un domaine d'étude ne saurait être complète sans un passage en revue des connaissances du domaine étudié. Nous présentons donc, dans le chapitre 4 un extrait des théories du game design, en particulier celles relatives à la motivation du joueur.

Chapitre 4

Théories du Game design

Nous entrons à présent dans le second champ de recherche de la thèse : la conception de jeux vidéo (ou Game Design). Le game design, à l'image de nombreux processus créatif, n'est pas une science exacte. Il n'existe pas plus de recette magique pour faire un bon jeu, qu'il n'y en a pour faire un bon film ou un bon livre. La conception de jeux vidéo est un processus itératif. Lorsqu'une nouvelle idée est proposée, elle est implémentée dans le jeu, puis testée. Aux premières heures du jeu vidéo, les différentes versions étaient essentiellement testées en interne, par les équipes chargées de la conception du jeu vidéo. A partir des années 2000 de plus en plus de studio se sont mis à également faire tester les jeux directement par le consommateur. Un panel de joueur est sélectionné dans la population de façon à être le plus représentatif possible du public ciblé par le jeu. Ces joueurs sont invités à jouer à une version donnée du jeu, pendant que des équipes observent leur comportement. Ces études qualitatives permettent d'avoir un retour d'expérience des utilisateurs et les jeux sont modifiés en fonction de ces retours. La plupart des acteurs de l'industrie du jeu vidéo disposent de ces laboratoires de test et une grande partie des modifications apportées aux jeux sont issues de ces expérimentations. Cependant, ces tests sont coûteux et le nombre de candidats reçus pour un test donné excède rarement quelques dizaines. D'autre part, plus la durée du test est élevée, plus ces derniers sont difficiles à mettre en place. En conséquence, il est difficile de mesurer des effets à long terme avec cette méthode.

Avec la démocratisation d'Internet, le taux de consoles et de PC connectés n'a cessé de croitre. De même, les jeux mobiles nécessitent souvent une connexion internet, ne serait-ce que pour les télécharger. Si bien qu'il est ainsi devenu particulièrement aisé de récolter des informations sur l'activité des joueurs.

Cette connexion permanente à rendu possible l'enregistrement des sessions de jeux de millions de joueurs. Les acteurs du jeu vidéo disposent potentiellement de l'ensemble des interactions entre le joueur et l'environnement virtuel. L'analyse de ces interactions peut fournir de nombreuses informations aux designers sur le comportement des joueurs à plus long terme, ainsi qu'une évaluation plus complète. De nombreux jeux proposent par exemple des mondes ouverts et émergents, et un faible nombre de joueur ne pourra couvrir

qu'une faible proportions des situations pouvant émerger dans ce type de jeu. Cependant il n'existe pas aujourd'hui de modèle capable de prendre en compte la complexité de ces données dépendantes du temps et de traiter un si grand volume d'information, tout en étant facilement interprétable par les designers.

Nous introduisons ici quelques unes des théories du game design en lien avec la motivation des joueurs et renvoyons le lecteur vers Salen & Zimmerman (2003) et Schell (2014) pour une présentation plus large du game design.

4.1 Mécaniques, dynamique et esthétiques

Salen & Zimmerman (2003) ont mis en évidence le fait que la conception d'un jeu vidéo constitue un processus de second ordre, dans lequel les designers ne créent pas directement l'expérience vécue par les joueurs. En fait, les concepteurs conçoivent des mécaniques de jeu, qui vont engendrer à leur tour des expériences utilisateur qui correspondent ou ne correspondent pas à ce que les designers avaient l'intention de délivrer aux joueurs.

Suivant ce raisonnement, Hunicke et al. (2004) décomposent un jeu vidéo en trois éléments : les mécaniques, les dynamiques et l'esthétique.

Les mécaniques décrivent les composantes élémentaires du jeu, telles que les règles du jeu : la physique du monde virtuel par exemple. C'est le niveau de détail le plus précis, tel qu'il est est codé par les programmeurs. Les données de tracking des joueurs sont quasi exclusivement des données relatives aux mécaniques du jeu : la position du joueur, l'utilisation d'un objet ou d'un véhicule.

Les dynamiques émergent des mécaniques. Dans un jeu de stratégie, les tactiques et stratégies que créent les joueurs émergent des mécaniques et justement, les développeurs sont souvent amenés à modifier le code de ces jeux après leur sortie pour qu'ils conservent l'équilibre désiré, après avoir observé des stratégies inattendues mises en places par les joueurs.

L'esthétique du jeu décrit les émotions ressenties par le joueur à travers l'expérience du jeu, telles que le fun, le sentiment de liberté, la tristesse ou la peur. La difficulté pour les designers consiste à générer des émotions en agissant uniquement sur les mécaniques du jeu. Un des leviers les plus exploités par les designers consiste à construire la difficulté du jeu de manière à ce qu'elle permette au joueur d'entretenir un sentiment de progression, de réussite et de maîtrise.

Puisque les mécaniques constituent les principaux leviers d'actions du designer, nous nous intéressons, dans le chapitre 8, à l'effet des mécaniques sur le temps de jeu des joueurs.

Dans la section suivante, nous explorons les différentes sources de motivation de joueurs. Comme nous le verrons, la difficulté est un paramètre fondamental de ces théories, et c'est pourquoi l'étude présentée dans le chapitre 9 nous explorons l'effet de la difficulté sur la motivation des joueurs.

4.2 Les théories de la motivation

4.2.1 Le Fun

Les fun est un mot couramment employé pour décrire l'expérience optimale que devrait ressentir un joueur en jouant à un jeu vidéo. Selon le game designer Raph Koster, "le fun est juste un autre mot pour l'apprentissage". Dans son livre A Theory of Fun for Game Design Koster (2005), il met en évidence que l'attrait principal que nous avons pour les jeux, et en particulier les jeux vidéo, vient de l'apprentissage que l'on en tire. Il donne, à titre d'exemple, le cas du tic-tac-toe (plus communément appelé morpion). Ce jeu de stratégie pure est apprécié des enfants tant qu'ils n'ont pas conscience des stratégies dominantes (combinaisons de coups forçant la victoire ou le null). Dès que le joueur prend conscience de ces stratégies, le jeu perd de son intérêt puisqu'il est dès lors possible d'anticiper la victoire (ou la défaite). Le jeu n'offre alors plus rien à nous apprendre et n'est donc plus fun.

Sur le plan neurologique, l'organisme libère de la dopamine lorsque nous apprenons (Koster (2005)). Cet héritage de l'évolution récompense physiquement l'acquisition de nouvelles connaissances, offrant un avantage sélectif aux individus capables d'anticiper et de réagir aux réactions de son environnement. Il n'est donc pas étonnant que chaque individu cherche à assouvir sa soif de connaissance, fusse à travers l'expérimentation virtuelle qu'offre un jeu vidéo.

La sensation d'apprendre quelque chose peut donc constituer un élément essentiel de motivation qui peut amener le joueur à poursuivre l'expérience qu'offre le jeu afin d'en maîtriser tous les tenants et aboutissants.

4.2.2 La théorie du flow

Csikszentmihalyi (1991) décrit l'état optimal d'expérience comme un équilibre entre les compétences du joueur et la difficulté du challenge. Un challenge à la difficulté trop élevée tend à frustrer le joueur et à créer une certaine anxiété face à la tâche. D'un autre coté, un challenge trop facile ne procure que de l'ennui.

L'état de flow représente un idéal pour le game designer comme pour le joueur. Ce dernier, absorbé par le jeu, est entièrement concentré sur les challenger à relever et le sentiment de compétence que cela lui procure l'incite à poursuivre le jeu tant que cette sensation durera.

La difficulté pour le designer est donc d'équilibrer les niveaux de difficulté, de sorte que le joueur vive une expérience optimale, dite de flow, qui ne soit ni trop complexe, ni trop facile, au regard des compétences du joueur. Nous proposons dans le chapitre 9 une méthode pour mesurer la difficulté d'un challenge.

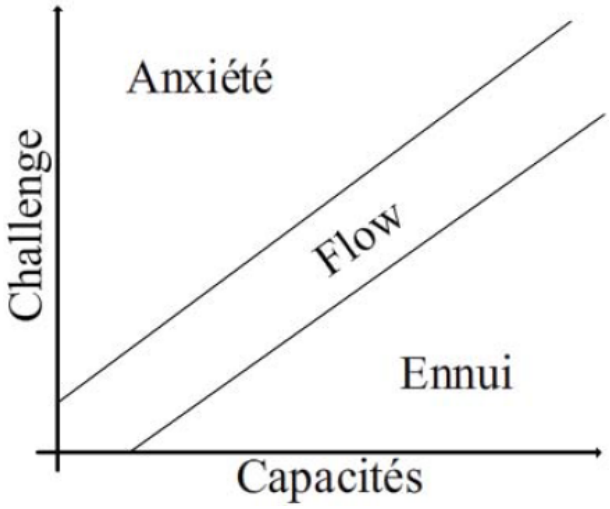

FIGURE 4.1 – théorie du flow

4.2.3 L'auto détermination

Parmi les théories qui peuvent nous aider à mieux comprendre les facteurs de motivations du joueur, nous présentons ici la théorie de l'auto détermination. Ryan & Deci (2000) distinguent deux types de motivation. La motivation extrinsèque se produit lorsque l'on réalise une action dans un but extérieur à la tâche en cours. C'est par exemple la cas d'un individu faisant du sport dans l'unique but de perdre du poids. Il n'est pas motivé par l'action de faire du sport, en elle même, mais uniquement par la perte de poids. A contrario, la motivation intrinsèque caractérise le fait de réaliser une action pour l'intérêt et le plaisir tiré de l'action elle même.

Ryan et Rigbi ont adapté la théorie de l'auto détermination aux jeux vidéo et proposent d'étudier trois besoins psychologiques primaires : l'autonomie, la compétence et le sentiment d'appartenance Rigby & Ryan (2007). Le sentiment d'autonomie est satisfait lorsque le joueur se sent libre de faire ce qu'il souhaite dans le jeu et ne se sent pas contraint de réaliser des actions qu'il ne souhaiterait pas réaliser naturellement. Le besoin de compétence est comblé si le joueur est satisfait des challenges qui lui sont proposés. On retrouve ici un lien avec la théorie du flow et la notion de difficulté. Les challenges proposés doivent êtres en adéquation avec les compétences du joueurs. Enfin le sentiment d'appartenance est satisfait si le joueur se sent en relation avec les autres joueurs. Ces relations peuvent êtres collaboratives, comme l'appartenance à un groupe de joueur où la coopération face à un objectif commun, ou bien conflictuelles avec l'opposition directe des joueurs ou à travers les résultats d'un tableau de scores. Elles peuvent aussi prendre place en dehors du jeu, par exemple sur les forums ou au travers du regard des personnes entourant le joueur. Dans tous les cas, le joueur à besoin de sentir que ses actions font sens au travers du regard des autres et il ressent le besoin d'être connecté au reste du groupe.

De ces trois besoins psychologiques émergent également divers profils de joueurs pour lesquels les motivations différent. Nous présentons dans le paragraphe suivant plusieurs typologies de joueurs, couramment utilisées dans l'industrie du jeu vidéo, qui permettent de mieux cerner les motivation des joueur et donc les manières d'accroitre leurs temps de jeu.

4.3 Les modèles du joueur

4.3.1 Bartle

Le premier à avoir proposé une typologie des joueurs est Bartle (1996). Développeur de l'un des premier Multi-User Dungeon - ancêtre des jeux en ligne massivement multijoueurs - il put observer le comportement de nombreux joueurs. A partir de ces observations, il proposa la première typologie des joueurs. Dans cette classification, représentée sur la figure 4.2, Bartle propose quatre catégories, selon si les joueurs ont tendance à agir ou a interagir avec le monde ou avec les autres joueurs. Le premier axes divise les joueurs et le monde, alors que le second traduit les actions ou les interactions. Ainsi un joueur motivé par les interactions avec les autres joueurs est qualifié de *socializer*, alors qu'un joueur motivé essentiellement par la complétion de tous les objectifs du jeu est définit comme un *achiever*. De même, un *exploreur* est un joueur qui aime interagir avec le monde, en parcourant chaque recoin, alors qu'un *killer* aime agir sur les autres joueurs au travers d'une confrontation.

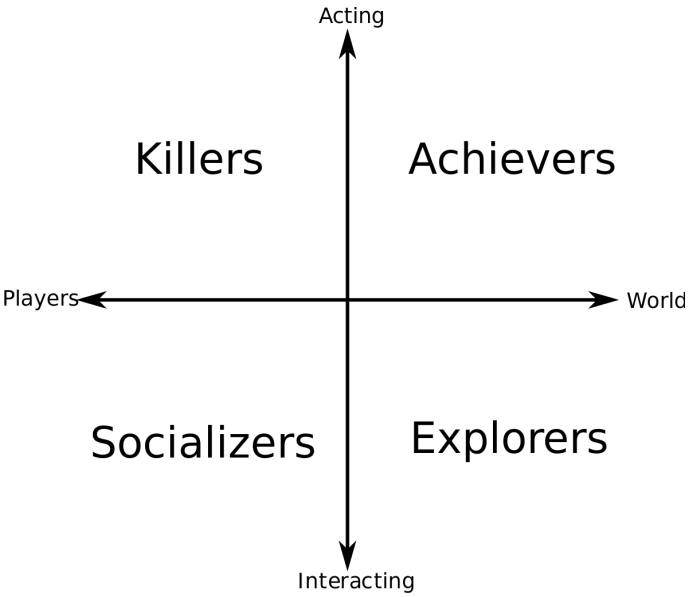

FIGURE 4.2 – Classification des joueurs de Bartle

Il est peu fréquent qu'un joueur soit motivé uniquement par l'un de ces quatre facteurs.

Bien souvent les motivations sont multiples et varient au cours du temps.

Bien que le modèle de Bartle soit issu de l'observation d'un type particulier de jeu, il reste applicable à nombre d'entre eux. Nous présentons dans le paragraphe suivant une classification des motivations des joueurs issue de l'étude d'un des type de jeu les plus joués dans le monde : les MMORPG.

4.3.2 Yee

Parmis les autres modèles proposés depuis, citons Yee (2006), qui fut le premier à proposer une classification des joueurs sur *World of Warcraft*. A travers des questionnaires en ligne, Yee put extraire la classification suivante, dont la traduction est en partie reprise de Mader (2015).

— Accomplissement : Avancement, mécaniques, compétition
— Social : Socialiser, relations, facteur humain
— Immersion : Découverte, jeu de rôle, personnalisation, évasion

Yee (2006) propose dix facteurs liés à l'engagement du joueur dans le jeu.

4.4 Synthèse

Nous avons vu au travers de ce chapitre que la conception d'un jeu vidéo est un processus complexe. Les designers visent à transmettre des émotions aux joueurs, mais pour ce faire ils peuvent uniquement agir sur les mécaniques du jeu. Les différentes théories de la motivation apportent des directives sur la conception d'une expérience optimale. D'autre part, les modèles du joueur permettent d'entrevoir la multitude de facteur liés à la motivation du joueur. Cependant, bien que ces modèles donnent des directions générales aux game designers, ils fournissent peu d'information sur les modification à apporter à un jeu pour l'améliorer. Dans le chapitre 8 nous montrons comment l'analyse de l'activité des joueurs permet de comprendre l'influence du design sur le temps de jeu des joueurs. Puis, dans le chapitre 9, nous proposons une première approche pour quantifier l'influence de la difficulté sur la motivation des joueurs.

Deuxième partie

Contributions

Chapitre 5

Problématique

Dans ce chapitre, nous présentons les problématiques principales de la thèse. Sur le plan algorithmique, les méthodes existantes ne permettent pas de passer à l'échelle. Nous montrons dans le paragraphe 5.1 que la descente de gradient stochastique n'est pas très efficace sur la vraisemblance partielle de Cox. Au paragraphe 5.2, nous montrons les limites en termes de mémoire utilisée des algorithmes existants. Concernant les applications à l'analyse des jeux vidéo, nous notons l'importance de prendre en compte les données longitudinales au paragraphe 5.3 et, au paragraphe 5.4, l'absence de modèles permettant de quantifier l'influence du design sur la motivation du joueur.

5.1 Descente de gradient stochastique sur la vraisemblance partielle de Cox

Dans l'état de l'art sur les méthodes d'optimisation, nous avons vu que la descente de gradient stochastique était la méthode la plus performante pour les très grands jeux de données. Nous montrons ici, que la vraisemblance partielle de Cox n'est pas adaptée pour ce type d'algorithme, puisqu'elle n'est pas séparable selon les individus.

Nous avons vu dans le paragraphe 3.5, que le gradient de l'opposé de la log-vraisemblance partielle de Cox s'écrit :

$$\frac{\partial \ell_n}{\partial \beta}(\beta) = \frac{1}{n}\sum_{i=1}^{n} \delta_i \left[-X_i + \sum_{j \in R_i} \frac{X_j \exp(X_j^T \theta)}{\sum\limits_{k \in R_i} \exp(X_k^T \theta)} \right], \tag{5.1}$$

La descente de gradient stochastique n'évalue le gradient que sur un seul individu (ou un petit nombre). Un gradient individuel, dans notre cas, a pour valeur :

$$\frac{\partial \ell}{\partial \beta}(\beta, X_i) = \delta_i \left[-X_i + \sum_{j \in R_i} \frac{X_j \exp(X_j^T \theta)}{\sum_{k \in R_i} \exp(X_k^T \theta)} \right] \tag{5.2}$$

On remarque, dans l'équation (5.2), que l'évaluation d'un gradient individuel requiert de l'information sur tous les individus à risque au moment où l'évènement se réalise pour l'individu i. Ceci implique, d'une part, que la vraisemblance partielle du modèle de Cox ne peut pas être calculée en ligne. D'autre part, chaque étape d'une descente de gradient stochastique serait d'une complexité de $O(np)$ - contre $O(p)$ pour une régression linéaire - conduisant la méthode a être moins efficace qu'une descente de gradient complète.

Dans le chapitre 6, nous proposons une méthode dont les gradients individuels ne dépendent pas des observations sur les autres individus. Le modèle obtenu peut ainsi profiter pleinement des avantages de la descente de gradient stochastique et donc passer plus facilement à l'échelle des grands jeux de données.

5.2 L'espace mémoire nécessaire aux algorithmes concurrents

Plusieurs implémentations permettent de prendre en compte des données longitudinales pour l'analyse des temps d'évènement. A notre connaissance, celle du package R *timereg* (Scheike et al. (2016)) est la plus performante. Cependant la méthode proposée et décrite dans Martinussen & Scheike (2007) repose sur des itérées de lissages par noyaux, rendant un passage à l'échelle difficile. D'autre part, l'implémentation qui en est faite sous R fait intervenir des matrices temporaires de grande taille, si bien que son application est rapidement limitée par la taille des données.

Dans le chapitre 7 nous proposons une implémentation qui ne souffre pas du problème de duplication des données, ni de l'inversion de très grandes matrices. Non seulement notre implémentation nécessite peu d'espace supplémentaire à celui occupé par les données d'origine, mais elle permet aussi de traiter des volumes de données d'une taille supérieure à la mémoire vive disponible sur la machine, en réalisant une lecture directe des données sur disque, sans les charger toutes en mémoire.

5.3 Analyse des données longitudinales dans les jeux vidéo

Les variables explicatives issues de l'analyse du comportement des joueurs peuvent être fixes ou dépendantes du temps. Une variable fixe ne change pas au cours du temps. C'est le cas de la date de naissance, du jour où le joueur a installé le jeu pour la première fois, ou du niveau de difficulté choisi en début de partie. Les variables dépendantes du temps ont des valeurs qui changent au cours du temps, par exemple le niveau du personnage, la

valeur des objets que le joueur possède dans son inventaire ou la position du joueur sur la carte.

On pourrait être tenté d'effectuer une analyse à partir, non pas des données longitudinales, mais d'une photographie des caractéristiques des joueurs au moment où ils arrêtent de joueur. Cela éviterait de conserver tout l'historique des joueurs en mémoire et entrainerait une réduction du coût de stockage de ces données. Par ailleurs, cela nous permettrait d'utiliser une large variété de modèles bien étudiés dans la littérature, qui ne prennent pas en compte les données longitudinales. Cependant, si l'on appliquait ces modèles sur cette photographie de l'état final de joueurs, le résultat serait biaisé par l'évolution des caractéristiques au cours du temps, comme illustré au travers de l'exemple suivant.

Pour faire simple, considérons deux joueurs numérotés 1 et 2 dont les temps de jeu total respectifs sont de 1h et 3h. Résumons leurs caractéristiques au travers des états a et b. Sur les données longitudinales, on constate que le joueur 1 est resté dans l'état a avant de quitter le jeu. Le joueur 2 est resté une heure dans l'état a, puis est passé dans l'état b avant de quitter le jeu au bout de trois heures.

1. a (1h)
2. a b (3h)

Au regard de ces données il n'est pas possible de conclure sur l'influence de l'état a ou b sur le temps de jeu des joueurs, puisque les deux joueurs étaient dans l'état a quand l'un a arrêté de joueur, alors que l'autre a continué la partie. Regardons ce qu'il en est sur la photographie de leur état final.

1. a (1h)
2. b (3h)

Puisque la photographie ne tient pas compte du passé du joueur, on conclurait ici que les joueurs dans l'état a ont plus de risque de quitter le jeu rapidement, i.e qu'il ont un temps de jeu moindre, que les joueurs dans l'état b. Quelque soit la manière dont on enregistre les données, par exemple en calculant la moyenne des états au cours du temps, on peut toujours construire des contre exemples qui montrent l'importance de prendre en compte l'information temporelle.

En conclusion, nous devons prendre en compte l'historique des joueurs et comparer leurs caractéristiques à des temps de jeu similaires, pour s'assurer d'écarter les effets liés à l'évolution des variables au cours du temps.

5.4 Approche formelle des modèles motivationnels

Nous avons vu dans le chapitre 4 que plusieurs modèles ont été proposés afin de mieux comprendre la motivation des joueurs. Les modèles du flow ou de l'auto détermination

permettent de mieux cerner les attentes des joueurs. Les classifications de Bartle ou Yee nous aident à entrevoir la variété des motivations des joueurs, respectivement de Multi-User Dungeon et des MMORPG. Cependant, aucun de ces modèles ne permet de quantifier l'influence du design sur la motivation du joueur. Dans le chapitre 9, nous proposons une première approche pour quantifier l'influence de la difficulté sur la motivation des joueurs.

5.5 Conclusion

Les quatre prochains chapitres présentent les contributions de la thèse et constituent une proposition de réponse aux problématiques que nous venons d'évoquer.

Chapitre 6

Modèles de Cox et d'Aalen dépendant du temps et en grande dimension

Nous considérons ici le problème d'estimation de l'intensité d'un processus de comptage, dans le cadre de covariables en grandes dimensions, pour les modèles de Cox et d'Aalen dépendant du temps. Nous introduisons une pénalisation par variation totale pondérée, dont les poids, axés sur les données, permettent de propager la pénalité sur les intervalles d'observation. Nous présentons un algorithme proximal pour résoudre le problème convexe étudié. Enfin, nous illustrons l'efficacité pratique de la méthode proposée sur des simulations et des données réelles.

Ce travail est issu d'une collaboration avec Mokhtar Z. Alaya, Agathe Guilloux et Sarah Lemler.

6.1 Introduction

Les données longitudinales sont monnaie courante dans de nombreux domaines, tels que la médecine, l'assurance, l'économie ou l'analyse des jeux vidéo. A titre d'exemple, le dossier médical partagé collecte, pour un grand nombre de patients et (dans certains pays) depuis plusieurs décennies, de nombreux marqueurs cliniques au cours du temps, ainsi que les temps d'évènements associés, voir e.g. Hayrinen et al. (2008), Baker et al. (2004), Coloma et al. (2011). Pour ces données longitudinales, les modèles de régression dynamiques, tels que Cox (Cai & Sun (2003), Tian et al. (2005)) ou Aalen (Martinussen & Scheike (1999)) avec des coefficients dépendant du temps, sont suffisamment flexibles pour détecter l'influence des covariables sur le risque. Pour des exemples d'applications, voir Kalantar-Zadeh et al. (2006), Bellera et al. (2010).

Lorsqu'un grand nombre de covariables est disponible pour chaque individu, ces modèles

dynamiques font face à trois principales difficultés qui doivent être prises en compte par les algorithmes statistiques : le nombre de covariables, la complexité du modèle (puisqu'une fonction doit être estimée pour chaque covariable) et la taille du jeu de données lorsque le nombre d'individus et/ou de covariables et le nombre d'enregistrements par individus est important.

Dans la littérature, les deux premières difficultés ont été, en partie, résolues via la sélection de modèle. Les contributions les plus récentes pour le modèle de Cox avec des coefficients dépendants du temps incluent Yan & Huang (2012), Honda & Hardle (2014). Ces auteurs proposent d'effecteur la sélection de modèle à partir de la sparsité induite par des pénalités group (Lasso et SCAD) sur des bases de splines. Nous renvoyons aussi le lecteur vers Cheng et al. (2014) pour des méthodes similaires dans le cas de variables longitudinales et lorsque le processus de réponse est totalement observé, i.e. non censuré.

A notre connaissance, la troisième difficulté, induite par la taille du jeu de données a seulement été traité dans Perperoglou et al. (2006), He et al. (2016). Les algorithmes existants, basés soit sur des méthodes à noyaux Martinussen & Scheike (2009*a*), Tian et al. (2005) ou des splines Yan & Huang (2012), Honda & Hardle (2014) ne passent pas à l'échelle. Dans le cas des estimateurs à noyaux, la sélection de modèles est entièrement basée sur des tests, et de fait, n'est pas applicable aux données de grandes dimensions (p grand), voir en particulier le package `timereg` Scheike et al. (2016). Les estimateurs à base de splines reposent souvent sur la création d'une matrice temporaire dense de très grande taille et ne passent donc pas à l'échelle. De plus, nous renvoyons le lecteur à Tibshirani et al. (2014) pour une discussion sur les performances des splines et des modèles pénalisés par variation totale dans le cas d'un modèle signal + bruit.

Nous introduisons, ici, un nouvel estimateur construit à partir d'estimateurs constants par morceaux comme introduits dans Murphy & Sen (1991). Nous ajoutons une pénalité group de type variation totale à laquelle nous ajoutons la pénalité Lasso sur le premier coefficient. Les poids de cette pénalité sont spécifiques à chaque covariable et dépendent des données. Cette pénalité induit la sélection de variable, ainsi que la sparsité temporelle de l'estimateur. Tout comme les estimateurs basés sur les splines Yan & Huang (2012), Honda & Hardle (2014), notre estimateur s'auto-adapte aux trois types de covariables : il exclut les covariables qui n'ont aucune influence sur le modèle, assigne un effet constant aux covariables dont l'effet est constant et donne une estimation constante par morceaux pour les variables dont l'effet dépend du temps. De fait, il traite donc les deux premières difficultés mentionnées plus haut.

Notre estimateur traite la troisième difficulté de deux manières. Premièrement il ne nécessite aucune duplication des données, contrairement aux implémentations de la plupart des approches basées sur les splines. Deuxièmement, le problème d'optimisation peut être traité efficacement par descente de gradient stochastique et de fait, passe facilement à l'échelle Bottou (2010, 2012), Rosasco et al. (2014), Atchade et al. (2014).

6.1.1 Cadre et modèles

Nous considérons le cadre usuel des processus de comptage où un processus $\tilde{N}$ compte le nombre de réalisations de l'événement d'intérêt dans un intervalle de temps fixé $[0, \tau]$ avec $0 < \tau < \infty$ et avec pour convention $\tilde{N}(0) = 0$ (voir Andersen et al. (1993), Martinussen & Scheike (2007)). Soit $\lambda_\star$ l'intensité du processus $\tilde{N}$ dépendant à la fois du temps et d'un processus prévisible, de dimension p, de covariables notées X (incluant éventuellement un intercept).

Le processus $\tilde{N}$ est alors indépendamment filtré (voir Andersen et al. (1993)) par un processus de censure Y. Nous notons N le processus résultant observé, dont l'intensité est donnée pour tout $t \in [0, \tau]$ par

$$Y(t)\lambda_\star(t, X(t)). \tag{6.1}$$

Dans ce cadre, nous considérons deux modèles dynamiques pour la fonction $\lambda_\star$:

— un modèle d'Aalen dépendant du temps

$$\lambda_\star^{\mathrm{A}}(t, X(t)) = X(t)\beta^\star(t), \tag{6.2}$$

— un modèle de Cox dépendant du temps

$$\lambda_\star^{\mathrm{M}}(t, X(t)) = \exp\big(X(t)\beta^\star(t)\big), \tag{6.3}$$

où, dans les deux cas, $\beta^\star$ est une fonction inconnue de $[0, \tau]$ dans $\mathbb{R}^p$ à estimer. Nous considérons le problème d'estimation des paramètres $\beta^\star$ dans les modèles dynamiques (6.2) et (6.3) à partir d'un échantillon de n individus indépendants :

$$\mathcal{D}_n = \big\{(X_i(t), Y_i(t), N_i(t)) : t \in [0, \tau], i = 1, \ldots, n\big\}. \tag{6.4}$$

L'estimation des modèles (6.2) et (6.3) a fait l'objet d'une attention importante ces quarante dernières années. Parmi les références pour le modèle additif d'Aalen, citons Aalen (1980, 1989), McKeague (1988), Huffer & McKeague (1991). Pour le modèle de Cox dépendant du temps Zucker & Karr (1990), Murphy & Sen (1991), Grambsch & Therneau (1994), Martinussen et al. (2002), Cai & Sun (2003), Winnett & Sasieni (2003) et plus recemment Yan & Huang (2012), Honda & Hardle (2014). Dans Martinussen & Scheike (2007), le lecteur pourra trouver une présentation complète des modèles, ainsi que des procédures d'estimation des modèles et de nombreux resultats pratiques. Le package R `timereg`, voir l'appendice C dans Martinussen & Scheike (2007), implémente ces estimateurs.

Ces modèles sont des extensions connues des modèles classiques d'Aalen (Aalen (1980)) et de Cox (Cox (1972)) avec des paramètres de régression constants. Les modèles dynamiques sont évidemment plus flexibles que leurs équivalents constants, mais ils souffrent de leur complexité : p fonctions inconnues doivent être estimées à partir des données. Nous proposons dans ce chapitre une procédure pénalisée qui réalise un compromis entre ces deux situations extrêmes, et qui, de plus, applique une sélection de variables.

6.1.2 Estimateurs pénalisés constant par morceaux

Suivant Murphy & Sen (1991), nous considérons des estimateurs constants par morceaux (i.e sur base d'histogramme) pour les fonctions inconnues p-dimensionnelles $\beta^\star$. Pour cela, nous considérons une L-partition de l'intervalle de temps $[0, \tau]$, où $L \in \mathbb{N}^*$ sera défini plus tard, notée :

$$\varphi_l = \mathbb{1}(I_l) \text{ et } I_l = \Big(\frac{l-1}{L}\tau, \frac{l}{L}\tau\Big]. \tag{6.5}$$

Pour tout $j = 1, \ldots, p$, les candidats à l'estimation de la j-ème coordonnée $\beta_j^\star$ de $\beta^\star$ appartiennent à l'ensemble des fonctions univariées constantes par morceaux

$$\mathcal{H}_L = \Big\{\alpha(\cdot) = \sum_{l=1}^{L} \alpha_l \varphi_l(\cdot) : (\alpha_l)_{1 \leq l \leq L} \in \mathbb{R}_+^L\Big\}. \tag{6.6}$$

Pour des jeux de données de petite taille n et/ou avec des covariables en grande dimension et/ou une partition fine, les estimateurs résultants vont souffrir d'une sur-paramétrisation, au sens où $\sqrt{n}$ sera possiblement plus petit que $p \times L$. D'un autre coté, les modèles de Cox et d'Aalen, dont les fonctions $\beta_j^\star$ sont constantes sur $[0, \tau]$, sont souvent trop limités (voir les discussions aux pages 205 et suivantes de Martinussen & Scheike (2007)).

Dans le but d'obtenir un compromis entre ces deux situations extrêmes, nous introduisons une pénalité $\ell_1 + \ell_1$-variation-totale, dont les poids sont spécifiques à chaque covariable (définies dans (6.15)). La partie variation totale de la pénalité conduit à des estimateurs simples et interpretables, qui ne varient pas trop souvent au cours du temps. La pénalité ℓ_1 permet à notre procédure de s'adapter au cas des covariables en grande dimension (p grand) en effectuant une sélection de variable.

Nos algorithmes ont des similarités avec les algorithmes du fused-Lasso. Ce dernier a été introduit et étudié pour des signaux bruités constants par morceaux, voir Tibshirani et al. (2014), Rinaldo (2009), Harchaoui & Levy-Leduc (2010), ou Dalalyan et al. (2014). Une pénalité par variation totale a été envisagée par Alaya et al. (2015) pour estimer l'intensité d'un processus de comptage, alors que Bouaziz & Guilloux (2015), Alaya et al. (2017) utilisent cette catégorie d'estimateurs dans un autre contexte.

L'estimateur du Lasso dans le contexte de l'analyse de survie en grande dimension a été introduit par Martinussen & Scheike (2009*b*), Gaiffas & Guilloux (2012) pour le modèle d'Aalen et par Tibshirani et al. (2005), Huang et al. (2013), Lemler (2013) entre autres pour le modèle de Cox.

Les principales contributions de ce chapitre sont les suivantes.

— Théorique : Nous proposons de nouveaux estimateurs pour le problème évoqué plus haut, voir leurs définitions dans (6.17) et (6.16).

— Pratique : Nous proposons des algorithmes, voir section 6.3, pour calculer nos estimateurs des modèles dynamiques (6.2) et (6.3). Dans la section 7.5, nous illustrons que nos estimateurs offrent de meilleurs résultats que les algorithmes existants, tant en termes de précision de l'estimation qu'en termes de complexité algorithmique.

La suite de ce chapitre est organisée comme suit. La section 6.1.2 est dédiée à la définition de nos estimateurs et à l'exposé de leurs propriétés théoriques. Dans la section 6.3, nous décrivons les algorithmes proposés. Nous rappelons comment simuler des données d'évènements en fonction de variables et de coefficients dépendant du temps par thinning. Enfin, le résultat de l'estimation sur ces simulations et des données réelles est présenté dans la section 7.5.

6.2 Procédure d'estimation

Nous présentons, dans cette section, de nouvelles procédures d'estimation impliquant une pénalité $\ell_1 + \ell_1$-variation-totale spécifique aux modèles multiplicatif et additif. Premièrement, nous décrivons en détail les modèles (6.3) et (6.2).

6.2.1 Modèles et hypothèses

Hypothese 1 *Nous supposons que*

$$\mathbb{P}[Y(\tau) > 0] > 0. \tag{6.7}$$

Il s'agit d'une hypothèse classique en analyse de survie (voir par exemple Andersen et al. (1993)).

6.2.2 Estimation

L'estimation des modèles à coefficients constants traditionnels est basée sur la minimisation d'un critère des moindres carrés (partiel) dans le cas du modèles d'Aalen (Aalen (1980)) et sur la maximisation de la log-vraisemblance (partielle) dans le cas du modèle de Cox (Cox (1972)). Nous renvoyons le lecteur vers Martinussen & Scheike (2009*a*), Gaiffas & Guilloux (2012) et Cox (1975) pour les détails.

Nous introduisons maintenant quelques notations. Pour chaque individu i et son processus X_i de p covariables, on note $X_{i,j}$ le processus associé à la $j$ième covariable. Respectivement, pour toute fonction p-dimensionnelle β, candidate à l'estimation de $\beta^\star$, on note β_j sa j-ième coordonnée. Nous définissons l'ensemble des fonctions candidates à l'estimation par

$$\Lambda^{\mathrm{A}} = \{x, t \in [0,\tau] \mapsto \lambda^{\mathrm{M}}_\beta(t, x(t)) = x(t)\beta(t) \mid \forall j \ \beta_j \in \mathcal{H}_L\} \tag{6.8}$$

pour le modèle d'Aalen ((6.2)) et

$$\Lambda^{\mathrm{M}} = \{x, t \in [0,\tau] \mapsto \lambda^{\mathrm{M}}_\beta(t, x(t)) = \exp\big(x(t)\beta(t)\big) \mid \forall j \ \beta_j \in \mathcal{H}_L\} \tag{6.9}$$

pour le modèle de Cox (6.3).

Puisque dans les fonctions candidates de Λ^{A} ou Λ^{M}, chaque coefficient dépendant du temps β_j est une fonction constante par morceaux, nous nous réfèrerons de manière équivalente à β comme une fonction p-dimensionnelle ou comme le vecteur de dimension $p \times L$

$$\beta = (\beta_{1,\cdot}^\top, \ldots, \beta_{p,\cdot}^\top)^\top = (\beta_{1,1}, \ldots, \beta_{1,L}, \ldots, \beta_{p,1}, \ldots, \beta_{p,L})^\top,$$

où chaque $\beta_{j,\cdot}$ est dans $\mathbb{R}^L$ et $\beta_{j,l}$ est la valeur prise par la j-ième coordonnée sur le l-ième intervalle de temps de notre L-partition $\{I_1, \ldots, I_L\}$:

$$\forall j = 1 \ldots, p,\ \forall l = 1, \ldots, L \text{ et } \forall t \in I_l, \beta_j(t) = \beta_{j,l}.$$

6.2.2.1 Estimation des modèles d'Aalen et de Cox dépendant du temps

Les estimateurs du modèle additif (6.2) et multiplicatif (6.3) dépendant du temps sont aussi définis via respectivement les moindres carrés et la log-vraisemblance (voir pages 108 et 206 et suivantes de Martinussen & Scheike (2007)).

Le modèle d'Aalen dépendant du temps Pour le modèle d'Aalen dépendant du temps, nous considérons le critère des moindres carrés sur les données et la fonction candidate à l'estimation $\lambda_\beta^{\mathrm{A}}$, définie par

$$\ell_n^{\mathrm{A}}(\beta) = \frac{1}{n} \sum_{i=1}^n \Big\{ \int_0^\tau \big(\lambda_\beta^{\mathrm{A}}(t, X_i(t))\big)^2 Y_i(t) dt - 2 \int_0^\tau \lambda_\beta^{\mathrm{A}}(t, X_i(t)) dN_i(t) \Big\}, \tag{6.10}$$

voir Gaiffas & Guilloux (2012) pour les détails sur ce critère. Lorsque le candidat $\lambda_\beta^{\mathrm{A}}$ est de la classe Λ^{A}, l'équation (6.10) se simplifie en

$$\begin{aligned} \ell_n^{\mathrm{A}}(\beta) &= \frac{1}{n} \sum_{i=1}^n \Big\{ \int_0^\tau \big(\sum_{j=1}^p X_i^j(t) \beta_j(t)\big)^2 Y_i(t) dt - 2 \int_0^\tau \sum_{j=1}^p X_i^j(t) \beta_j(t) dN_i(t) \Big\} \\ &= \frac{1}{n} \sum_{i=1}^n \sum_{l=1}^L \Big\{ \int_{I_l} \big(\sum_{j=1}^p X_i^j(t) \beta_{j,l}\big)^2 Y_i(t) dt - 2 \sum_{j=1}^p \big(\int_{I_l} X_i^j(t) dN_i(t) \big) \beta_{j,l} \Big\}. \end{aligned} \tag{6.11}$$

Le modèle de Cox dépendant du temps Nous considérons, dans ce paragraphe, l'estimation du modèle de Cox dépendant du temps. L'opposé de la log-vraisemblance pour nos données et un candidat $\lambda_\beta^{\mathrm{M}}$ est donné par

$$\ell_n^{\mathrm{M}}(\beta) = -\frac{1}{n} \sum_{i=1}^n \Big\{ \int_0^\tau \log\big(\lambda_\beta^{\mathrm{M}}(t, X_i(t))\big) dN_i(t) - \int_0^\tau Y_i(t) \lambda_\beta^{\mathrm{M}}(t, X_i(t)) dt \Big\}, \tag{6.12}$$

voir Andersen et al. (1993) pour les détails. Lorsque $\lambda_\beta^{\mathrm{M}}$ est de la classe Λ^{M}, cette dernière expression se réduit à

$$\begin{aligned}\ell_n^{\mathrm{M}}(\beta) &= -\frac{1}{n}\sum_{i=1}^{n}\Big\{\int_0^\tau \sum_{j=1}^{p} X_i^j(t)\beta_j(t)dN_i(t) - \int_0^\tau Y_i(t)\exp\big(\sum_{j=1}^{p} X_i^j(t)\beta_j(t)\big)dt\Big\} \\ &= -\frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{L}\Big\{\sum_{j=1}^{p}\big(\int_{I_l} X_i^j(t)dN_i(t)\big)\beta_{j,l} - \int_{I_l} Y_i(t)\exp\big(\sum_{j=1}^{p} X_i^j(t)\beta_{j,l}\big)dt\Big\}. \quad (6.13)\end{aligned}$$

Procédure d'estimation Nous introduisons un vecteur de poids bien choisis et adaptés aux données $\hat{\gamma} = (\hat{\gamma}_{1,\cdot}^\top, \ldots, \hat{\gamma}_{p,\cdot}^\top)^\top$. Les poids $\hat{\gamma}_{j,l} > 0$ sont complètement dépendants des données et leur forme est donnée par

$$\hat{\gamma}_{j,l} = \mathcal{O}\Big(\sqrt{\frac{\log pL}{n}\hat{V}_{j,l}}\Big), \quad \text{avec} \quad \hat{V}_{j,l} = \frac{1}{n}\sum_{i=1}^{n}\sum_{u=l}^{L}\int_{I_u}(X_i^j(t))^2 dN_i(t). \quad (6.14)$$

La pénalité $\ell_1 + \ell_1$-variation-totale associée au vecteur $\hat{\gamma}$ est définie par

$$\|\beta\|_{\mathrm{gTV},\hat{\gamma}} = \sum_{j=1}^{p}\Big(\hat{\gamma}_{j,1}|\beta_{j,1}| + \sum_{l=2}^{L}\hat{\gamma}_{j,l}|\beta_{j,l} - \beta_{j,l-1}|\Big) \quad (6.15)$$

pour chaque $\beta \in \mathbb{R}^{p\times L}$. Nos estimateurs sont respectivements définis par $\hat{\lambda}^{\mathrm{A}} = \lambda^{\mathrm{A}}_{\hat{\beta}^{\mathrm{A}}}$ et $\hat{\lambda}^{\mathrm{M}} = \lambda^{\mathrm{M}}_{\hat{\beta}^{\mathrm{M}}}$, où

$$\hat{\beta}^{\mathrm{A}} = \operatorname*{argmin}_{\beta\in\mathbb{R}^{p\times L}}\Big\{\ell_n^{\mathrm{A}}(\beta) + \|\beta\|_{\mathrm{gTV},\hat{\gamma}}\Big\}, \quad (6.16)$$

pour le modèle d'Aalen et

$$\hat{\beta}^{\mathrm{M}} = \operatorname*{argmin}_{\beta\in\mathbb{R}^{p\times L}}\Big\{\ell_n^{\mathrm{M}}(\beta) + \|\beta\|_{\mathrm{gTV},\hat{\gamma}}\Big\} \quad (6.17)$$

pour le modèle de Cox

6.3 Algorithmes

Dans cette section, nous décrivons deux algorithmes d'optimisation pour estimer les coefficients des modèles (6.16) et (6.17). Le premier est un algorithme de type descente de gradient proximale (PGD) (voir Daubechies et al. (2004), Beck & Teboulle (2009), Bach et al. (2011), Parikh & Boyd (2014)), alors que le second appartient à la classe des algorithme de descente de gradient stochastique proximale (SPGD) (voir Bottou (2010, 2012), Rosasco et al. (n.d.), Atchade et al. (2014)).

Nous avons vu dans le chapitre 2, que les algorithmes proximaux reposaient sur la connaissance de l'opérateur proximal associé à la pénalité utilisée. Pour les algorithmes 12

et 13 ci-dessous, nous aurons besoin de l'opérateur proximal de la pénalité introduite en (6.15), c'est à dire

$$\text{prox}_{\hat{\gamma}}^{\text{gTV}}(\beta) = \underset{x\in\mathbb{R}^{p\times L}}{\text{argmin}} \left\{ \frac{1}{2}\|\beta - x\|_2^2 + \sum_{j=1}^{p} \left(\hat{\gamma}_{j,1}|x_{j,1}| + \sum_{l=2}^{L} \hat{\gamma}_{j,l}|x_{j,l} - x_{j,l-1}| \right) \right\} \tag{6.18}$$

Puisque $\|\cdot\|_{\text{gTV},\hat{\gamma}}$ est séparable par blocs, nous avons $\left(\text{prox}_{\hat{\gamma}}^{\text{gTV}}(\beta)\right)_{j,\cdot} = \text{prox}_{\hat{\gamma}}^{\text{gTV}}(\beta_{j,\cdot})$ pour tout $j = 1, \ldots, p$. En conséquence, nous pouvons nous focaliser sur l'opérateur proximal associé au $j$ème bloc. L'algorithme 11 exprime $\text{prox}_{\hat{\gamma}}^{\text{gTV}}(\beta)$ en fonction de l'opérateur proxiaml de la variation totale pondérée, décrite par Alaya et al. (2015), i.e. $\text{prox}_{\hat{\gamma}}^{\text{TV}}$.

Algorithme 11 Opérateur proximal de $||.||_{\text{gTV},\hat{\gamma}}$ (voir (6.18))

Entrées : $\beta \in \mathbb{R}^{p\times d}$ et les poids $\hat{\gamma}_{j,l}$ pour $j = 1, \ldots, p$ et $l = 1, \ldots, L$.
Sortie : $\theta = \text{prox}_{\hat{\gamma}}^{\text{gTV}}(\beta)$
  for $j = 1$ **to** p **do**
    $\eta_{j,\cdot} \leftarrow \text{prox}_{\hat{\omega}_{j,\cdot}}^{\text{TV}}(\beta_{j,\cdot})$ où $(\hat{\omega}_{j,\cdot} = \hat{\gamma}_{j,\cdot}\backslash\{\hat{\gamma}_{j,1}\})$
    $\vartheta_{j,\cdot} \leftarrow \eta_{j,\cdot} - \eta_{j,1}\mathbf{1}_L$
    $\theta_{j,\cdot} \leftarrow \vartheta_{j,\cdot} + \eta_{j,1} \max\left(1 - \frac{\hat{\gamma}_{j,1}}{L|\eta_{j,1}|}, 0\right)\mathbf{1}_L$
  end for

L'algorithme de descente de gradient proximale pour le modèles d'Aalen dépendant du temps est le suivant

Algorithme 12 Descente de gradient proximale pour le modèle d'Aalen dépendent du temps (6.16)

Entrées : $\beta^0 \in \mathbb{R}^{p\times d}$ et les poids $\hat{\gamma}_{j,l}$ pour $j = 1, \ldots, p$ et $l = 1, \ldots, L$.
Sortie : β^k
  $\beta^1 = \beta^0$
  Calcul de la constante de Lipschitz L de $\nabla\ell_n^{\text{A}}$
  while on n'a pas convergé **do**
    $v = \beta^k + \frac{k-1}{k+2}\left(\beta^k - \beta^{(k-1)}\right)$
    $\beta^{(k+1)} = \text{prox}_{\hat{\gamma}/L}^{\text{gTV}}\left(v - \frac{1}{L}\nabla\ell_n^A(v)\right)$
  end while

Dans l'algorithme 13, nous implementons une descente de gradient stochastique proximale. Le gradient de la log-vraisemblance du modèle de Cox dépendant du temps (eq. (6.17)) n'étant pas Lipschitzien, nous utilisons ici des pas de descente adaptatif, repris de Schaul et al. (2012). Notons qu'ici le produit $\odot$ et la division sont des opérations élément par éléments.

Algorithme 13 Descente de gradient stochastique proximale pour le modèle de Cox dépendent du temps (6.17)

Entrées : $\beta^0 \in \mathbb{R}^{p\times d}$, les poids $\hat{\gamma}_{j,l}$ pour $j = 1,\dots,p$ et $l = 1,\dots,L.$ et $r^{(1)} \in [0,1]$
Sortie : β^k

$\beta^1 = \beta^0$
while on n'a pas convergé **do**
Choisir aléatoirement $i_k \in \{1,\dots,n\}$ et calculer $\nabla_{i_k} = \nabla \ell^{\mathrm{M}}_{i_k}(\beta^{(k)})$
Actualisation des moyennes mobiles
$a^{(k)} \leftarrow (1-(r^{(k)})^{-1})a^{(k)} + (r^{(k)})^{-1}\nabla^{\mathrm{M}}_{i_k}$
$b^{(k)}_j \leftarrow (1-(r^{(k)})^{-1})b^{(k)}_j + (r^{(k)})^{-1}\|\nabla_{j,\cdot}\|^2$
$c^{(k)} \leftarrow (1-(r^{(k)})^{-1})c^{(k)} + (r^{(k)})^{-1}\operatorname{diag}(H_{i_k})$ where $H_{i_k} = \left(\frac{\partial^2\left(\ell^{\mathrm{M}}_{i_k}(\beta^{(k)})\right)}{\partial^2\beta}\right)$
Estimation du pas de descente
$\epsilon^{(k)}_j \leftarrow \frac{1}{c^+_j}\frac{\sum_{l=1}^{L}\left(a^{(k)}_{j,l}\right)^2}{b^{(k)}_j}$; where $c^+_j = \max_{1\le l\le L} c_{j,l}$
$\eta_j \leftarrow \epsilon^{(k)}_j$
$\epsilon^{(k)} \leftarrow \left(\epsilon^{(k)}_1\mathbf{1}_L,\dots,\epsilon^{(k)}_p\mathbf{1}_L\right)^\top$
Actualisation de la taille de la mémoire
$r^{(k)} \leftarrow \left(1-\frac{\sum_{l=1}^{L}\left(a^{(k)}_{j,l}\right)^2}{b^{(k)}_j}\right)\odot r^{(k)} + 1$
$\theta^{(k)} \leftarrow \beta^{(k)} - \epsilon^{(k)}\odot\nabla_{i_k}$
$\beta^{(k+1)} \leftarrow \left(\operatorname{prox}^{\mathrm{gTV}}_{\eta_1\gamma}\left(\theta^{(k)}_{1,\cdot}\right),\dots,\operatorname{prox}^{\mathrm{gTV}}_{\eta_p\gamma}\left(\theta^{(k)}_{p,\cdot}\right)\right)^\top$
end while

Pour actualiser les pas de descentes ϵ_k de l'algorithme 13, nous pouvons utiliser les pas décroissants de Bottou (2012), i.e. $\epsilon_k = \epsilon_0\frac{1}{(1+\epsilon_0\xi k)^\alpha}$ où $\alpha \in [\frac{1}{2},1], \xi > 0$, et ϵ_0 appris sur un petit echantillon des données. Différentes approches proposent une actualisation adaptative de ϵ_k, telle que les algorithmes AdaGrad in Duchi et al. (2011), AdaDelta in Zeiler (2012) or vSGD in Schaul et al. (2012). Dans notre cas, nous avons implémenté une variation de vSGD avec un calcul explicite de la diagonale de la hessienne de la log vraisemblance (les autres méthodes utilisent une approximation par différence finie de celle ci).

6.4 Conclusion

Nous avons introduit dans ce chapitre une méthode pour l'analyse des données longitudinales de grande taille. Ce travail est issu d'une collaboration avec Mokhtar Z. Alaya, Agathe Guilloux et Sarah Lemler et sera prochainement soumis à publication. L'article qui en découle inclut l'analyse théorique des propriétés de convergence des estimateurs proposés, résultante de leurs travaux. Pour finir indiquons qu'il découle de ces démonstrations un ordre de grandeur théorique pour le choix de L. Celui ci est de l'ordre de $n^{1/3}$ et c'est cette valeur que nous utiliserons dans les applications numériques présentées au chapitre

suivant.

Chapitre 7

La librairie CoxTV

Dans ce chapitre, nous présentons le package R coxtv, développé pendant la thèse, qui permet d'estimer l'influence de variables longitudinales sur des temps d'évènements suivant la méthode introduite dans le chapitre 6. Ce package permet de prendre en compte des effets dépendants du temps lorsque cela s'avère nécessaire. De plus, le package coxtv permet d'effectuer une sélection de variable par approche pénalisée, ce qui permet de traiter des cas en grande dimension. L'implémentation repose sur une librairie entièrement écrite en C++ et interfacée à R via Rcpp (Eddelbuettel & François (2011)). En plus de disposer de nombreuses méthodes d'optimisation, le package dispose d'un mode *online* qui permet de traiter les données directement sur disque, sans avoir à les charger entièrement en mémoire. Cette fonctionnalité permet de traiter des volumes de données dépassant la mémoire vive de la machine utilisée.

Dans un premier temps, nous rappelons la vraisemblance du modèle proposé au chapitre précédent et détaillons les calculs utiles à l'estimation de ses paramètres. Puis nous présentons l'utilisation du package au travers que quelques exemples, pour illustrer sa syntaxe et ses fonctionnalités. Enfin, nous illustrons ses performances à travers des simulations, en le comparant à deux implémentations existantes de modèles concurrents. Pour finir, nous illustrons les résultats obtenus à l'aide de notre librairie sur un jeu de données réelles.

7.1 Modèle de Cox dépendant du temps

Nous avons vu dans le paragraphe 3.3 que l'opposé de la log-vraisemblance du modèle de Cox dépendant du temps s'écrit

$$\ell_n(\beta) = -\frac{1}{n}\sum_{i=1}^{n}\Big\{\int_0^\tau \sum_{j=0}^{p} X_{i,j}(t)\beta_j(t)dN_i(t) - \int_0^\tau Y_i(t)\exp\big(\sum_{j=0}^{p} X_{i,j}(t)\beta_j(t)\big)dt\Big\}. \tag{7.1}$$

Dans le chapitre 2 sur l'optimisation, nous avons vu que l'estimation des paramètres d'un modèle faisait intervenir le gradient et éventuellement la hessienne. Nous présentons ici leurs formules respectives

$$\frac{\partial \ell_n}{\partial \beta_k}(\beta) = -\frac{1}{n}\sum_{i=1}^{n}\Big\{\int_0^\tau X_{i,k}(t)dN_i(t) - \int_0^\tau Y_i(t)X_{i,k}(t)\exp\big(\sum_{j=0}^{p} X_{i,j}(t)\beta_j(t)\big)dt\Big\} \tag{7.2}$$

$$\frac{\partial^2 \ell_n}{\partial \beta_l \partial \beta_k}(\beta) = \frac{1}{n}\sum_{i=1}^{n}\int_0^\tau Y_i(t)X_{i,k}(t)X_{i,l}(t)\exp\big(\sum_{j=0}^{p} X_{i,j}(t)\beta_j(t)\big)dt \tag{7.3}$$

Remarquons que toutes ces équations sont séparables selon les individus. Le calcul de la vraisemblance, du gradient et de la hessienne sur un seul individu ne fait pas intervenir les observations des autres individus, contrairement à la vraisemblance partielle de Cox présentée dans le section 3.5.

Dans le paragraphe suivant, nous spécifions la forme des coefficients $\beta(t)$ en introduisant des coefficients constant par morceaux au cours du temps.

7.1.1 Coefficients constants par morceaux

Nous donnons maintenant le détail des calculs lorsque les coefficients β_j dépendent du temps. Soit $(I_l)_{l\in\{0,L\}}$ une partition de $[0,\tau]$. Nous utiliserons des coefficients β constant par morceaux sur ces intervalles, comme dans le chapitre précédent :

$$\beta_j(t) = \sum_{l=1}^{L}\beta_{j,l}\mathbb{1}_{(I_l)}(t). \tag{7.4}$$

Avec ces coefficients, l'opposé de la log-vraisemblance s'écrit :

$$\begin{aligned}\ell_n(\beta) = -\frac{1}{n}\sum_{i=1}^{n}\Big\{&\int_0^\tau \sum_{j=0}^{p} X_{i,j}(t)\sum_{l=1}^{L}\beta_{j,l}\mathbb{1}_{(I_l)}(t)dN_i(t) \qquad (7.5)\\ &- \int_0^\tau Y_i(t)\exp\big(\sum_{j=0}^{p} X_{i,j}(t)\sum_{l=1}^{L}\beta_{j,l}\mathbb{1}_{(I_l)}(t)\big)dt\Big\}\\ = -\frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{L}\Big\{&\sum_{j=0}^{p}\beta_{j,l}\int_{I_l} X_{i,j}(t)dN_i(t) - \int_{I_l} Y_i(t)\exp\big(\sum_{j=0}^{p} X_{i,j}(t)\beta_{j,l}\big)dt\Big\} \qquad (7.6)\end{aligned}$$

Nous donnons aussi le gradient et la hessienne correspondants :

$$\frac{\partial \ell_n}{\partial \beta_{k,s}}(\beta) = -\frac{1}{n}\sum_{i=1}^{n}\Big\{\int_{I_s} X_{i,k}(t)dN_i(t) - \int_{I_s} Y_i(t)X_{i,k}(t)\exp\big(\sum_{j=0}^{p} X_{i,j}(t)\beta_{j,s}\big)dt\Big\}, \tag{7.7}$$

ce gradient est dont un vecteur de taille $p \times L$, quand l'hessienne

$$\frac{\partial^2 \ell_n}{\partial \beta_{k,s}\partial \beta_{k',s}}(\beta) = \frac{1}{n}\int_{I_s} Y_i(t)X_{i,k}(t)X_{i,k'}(t)\exp\big(\sum_{j=0}^{p} X_{i,j}(t)\beta_{j,s}\big)dt\Big\} \tag{7.8}$$

est de taille $(p \times L, p \times L)$ et sparse car elle contient $p^2L(L-1)$ coefficients nuls.

$$\frac{\partial^2 \ell_n}{\partial \beta_{k,s}\partial \beta_{k',s'}}(\beta) = 0 \quad \textbf{si} \quad s \neq s' \tag{7.9}$$

7.1.2 Données longitudinales

En pratique nous n'observons pas les valeurs des covariables en temps continu. Les temps d'observation sont propres à chaque individu et varient d'une covariable à l'autre. Le stockage et l'analyse de ses données constituent un défit intéressant en soit, puisque contrairement à la plupart des modèles statistiques où les données se présentent sous la forme d'une matrice $n \times p$, les données temporelles ont une structure naturelle plus proche d'un tenseur 3D.

Dans la pratique les observations ne sont pas faites en temps continu et la fréquence d'observation varie selon les variables et les individus, comme le montre la figure 7.1.

Nous pourrions approcher la fonctionnelle de chaque $X_{i,j}$ par des splines. Cependant, le calcul de l'intégrale rendrait le processus inapplicable aux grand jeux de données. Nous proposons donc d'approximer $X_{i,j}$ par des fonctions constantes par morceaux. Il est à noter que si les variables $X_{i,j}$ sont elles mêmes des processus de comptage, alors elles sont naturellement représentées par des fonctions constantes par morceaux, sans perte d'information. C'est le cas de la plupart des données issues de l'industrie du jeux vidéo, que nous étudierons par la suite.

7.1.3 Vraisemblance discrète

Dans le cas où les observations X_i et les coefficients β sont constants par morceaux, la vraisemblance peut s'écrire sans intégrale. Indiçons par k les intervalles de temps sur lesquels $X_{i,j}$ est constant et notons $X_{i,j,k}$ les valeurs de $X_{i,j}$ sur ces intervalles. Notons

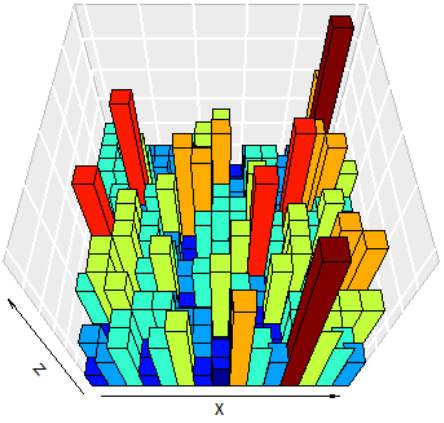

FIGURE 7.1 – Représentation des données temporelles en 3D. Individus et covariables selon les axes x et y et fréquence d'observation selon l'axe z

$\beta_{j,l}$ la valeur du coefficient β_j aux différents intervalles de temps, dont la discrétisation est propre aux coefficients et différente de celle des données. Alors le calcul de l'intégrale reviens à effectuer une sommation sur les sous intervalles, que nous noterons I_s, formés par les $X_{i,j,k}$ et $\beta_{j,l}$, comme illustré sur la figure 7.2. Nous noterons X_{i,j,I_s} les valeurs de $X_{i,j}$ sur ces intervalles.

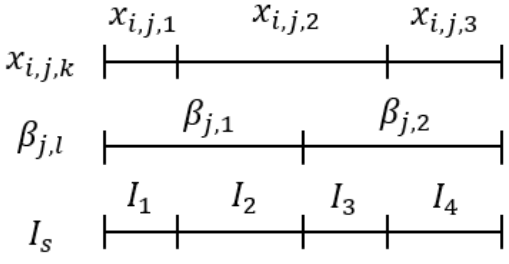

FIGURE 7.2 – Intervalles communs à $X_{i,j}$ et β_j.

En utilisant ces notations, l'opposé de la log-vraisemblance s'écrit :

$$\ell_n(\beta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{L}\Big\{\sum_{j=1}^{p}\sum_{s} X_{i,j,I_s}\beta_{j,l}N_i(I_s) - \sum_{s} Y_i(I_s)\exp\big(\sum_{j=1}^{p} X_{i,j,I_s}\beta_{j,l}\big)|I_s|\Big\} \tag{7.10}$$

où $N_i(I_s) = 1$ si un évènement a lieu dans I_s, zero sinon et $Y_i(t) = 1$ tant que l'individu est à risque.

Le gradient que nous utilisons pour la descente de gradient proximale est donc le suivant

$$\begin{aligned}\frac{\partial \ell_n(\beta)}{\partial \beta} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{l=1}^{L}\Big\{&\sum_{j=1}^{p}\sum_{s} X_{i,j,I_s}N_i(I_s) \\ &- \sum_{s} Y_i(I_s)\left(\sum_{j=1}^{p} X_{i,j,I_s}\right)\exp\left(\sum_{j=1}^{p} X_{i,j,I_s}\beta_{j,l}\right)|I_s|\Big\}.\end{aligned} \tag{7.11}$$

Pour la descente de gradient stochastique proximale, nous utilisons les gradients individuel, i.e où la somme sur i ne concerne plus qu'un seul individu.

Notons qu'il est possible de pré-calculer les sous intervalles I_s lors de la première lecture des données, tous comme les temps de réalisation d'un évènement, pour éviter de refaire le calcul à chaque itérations.

7.2 Implémentation

Une part importante du temps de la thèse a été consacré à l'implémentation du modèle proposé sous la forme d'une librairie C++. Le choix de ce langage c'est fait pour des raisons de performances, puisque nous devons traiter de gros volumes de données, dépassant la capacité de la mémoire vive. D'autre part, ce langage permet une intégration simplifiée vers un des langages les plus utilisés par la communauté statistique : R. Cette intégration au sein d'un package R facilite la distribution de notre solution, puisque la compilation des fichier C++ est entièrement gérée par R et ne demande aucunes connaissance à l'utilisateur de la librairie.

Nous donnons ici quelques unes des caractéristiques de la librairie C++ développée. La librairie est composée des modules suivants :

- **Data**, qui implémente les méthodes d'accès aux données, par exemple le cas classique Y X ou les données longitudinales et censurées des modèles de survie. Quelque soit le type de données, il est possible d'y accéder dans son ensemble en les chargeant entièrement en mémoire, soit par la lecture d'un fichier, soit par un passage par référence si les données ont déjà été chargées depuis R, si la taille de la mémoire vive le permet. La limitation de la mémoire vive est un problème inhérent à de nombreux package R, puisque contrairement à des langages de programmation

comme SAS, R nécessite dans la plupart des cas de charger les données en mémoire. Nous proposons, pour chaque type de données, un accès direct sur disque. Cette fonctionnalité permet d'analyser tout jeux de données, dès lors qu'il est possible de le stocker sur un espace physique.

- **Model**, qui implémente par exemple le cas de la régression linéaire ou le modèle de Cox dépendant du temps proposé plus haut.
- **Penalty** contient le calcul des normes ℓ_1, TV et celle proposée au chapitre 6, ainsi que les opérateurs proximaux correspondants. Le calcul du support associé à une pénalité est aussi faite par ce modèle et la projection sur un support correspond à un opérateur proximal.
- **Solver** inclut différentes méthodes d'optimisation, notamment les algorithmes de descente de gradient accéléré avec les différentes méthodes de recherches linéaires proximales, présentés dans le chapitre 2. Ainsi que plusieurs variations des algorithmes de descente de gradient stochastique aux pas de descente adaptatifs.

La construction de la librairie, en bloc relativement indépendants, permet d'utiliser n'importe quel solveur sur n'importe quel modèle, tout en ajoutant une ou des pénalités spécifiques. Cette modularité permet facilement d'étendre notre librairie à d'autres modèles, pénalités ou méthodes d'optimisation.

D'autre part, elle possède les propriétés suivantes :

- Header-only : Elle contient uniquement des fichier .h (par opposition aux fichiers .cpp), pour faciliter son intégration au sein d'autres librairies et son intégration à R via Rcpp.
- Template : pour faciliter l'interaction entre les différents modules (voir plus bas).
- Basée sur la librairie Eigen (Guennebaud et al. (2010)), pour, entre autre, une gestion efficace des données orientés ligne ou colonne.

7.3 Exemples d'utilisation du package

Nous illustrons à présent l'utilisation du package *coxtv* sur un jeu de données. Une des représentation classique des données longitudinales constantes par morceaux est d'utiliser des variables start et stop pour définir l'intervalle de temps sur lequel la variable x prend une valeur donnée. La variable event indique ici la présence où non d'un évènement à la fin de l'intervalle de temps. La syntaxe pour l'utilisation du package est la suivante :

```
library(coxtv)
model_1 <- coxtv(Surv(start, stop, event) ~ x, data=sample)
```

La syntaxe coïncide avec celle du package *survival*, référence pour l'analyse de survie en R.

D'autre part, il est possible d'utiliser la flexibilité offerte par les formules, comme le montre l'exemple suivant :

```
model_2 <- coxtv(Surv(start, stop, event) ~ x1 + x3*x4,
```

```
                    data=sample)
```

Les paramètres supplémentaires sont accessibles via une liste, de la facon suivante

```
model_3 <- coxtv(Surv(start, stop, event) ~ x, data=sample),
                 param=list(useBinarsity = T,
                            crossValidation=T,
                            Kfolds=5,
                            BinarsityGrid=c(0, 10^(-5:0)),
                            maxIter=500,
                            gradNorm=1e-6,
                            lineSearch=T,
                            optimWeightOnSupport=T)
                 )
```

De nombreux paramètres sont disponibles et leur documentation est détaillée dans l'aide associée au package.

D'autre part, il est possible d'effectuer l'estimation à partir d'un fichier de taille supérieure à la mémoire vive disponible, sans que celui ci ne soit entièrement chargé en mémoire, comme le montre le code suivant

```
file <- path_to_data/data.csv
model_4 <- coxtv(param = list(FileName=file, sep=","))
```

Dans ce cas, les données sont chargées individu par individu et l'algorithme tire partie des avantages de la descente de gradient stochastique pour limiter le nombre d'accès au données.

7.4 Estimation avec support

Nous rappelons la procédure d'estimation par restriction à un support via l'utilisation d'une pénalité, évoquée en 1.3.4 et détaillons les procédures d'identification et de restriction au support.

L'estimation passe donc par les trois étapes suivantes,

1. Calculer les coefficients pénalisés

$$\hat{\beta} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \hat{R}(\beta) + \operatorname{pen}(\beta) \right\} \tag{7.12}$$

2. En déduire le support des coefficients

$$S = \operatorname{support}_{\operatorname{pen}}(\beta) \tag{7.13}$$

3. Faire l'estimation (non pénalisée) sur le support

$$\tilde{\beta} \in \underset{\beta \in S}{\operatorname{argmin}} \hat{R}(\beta) \tag{7.14}$$

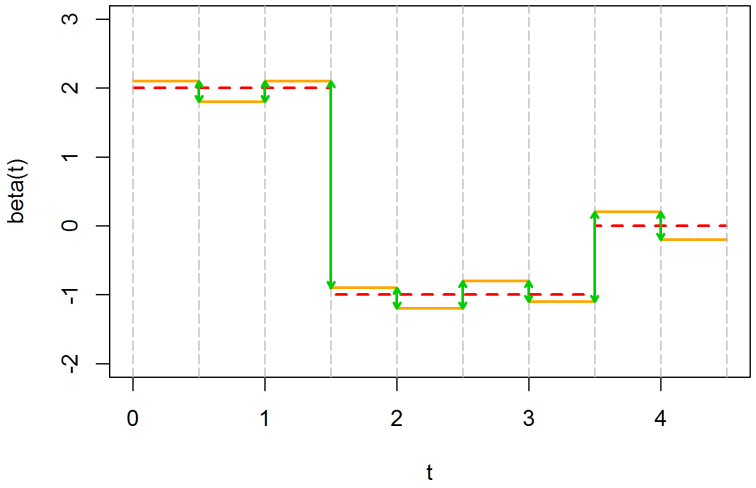

FIGURE 7.3 – Illustration de la pénalité par variation totale. Coefficients non pénalisés (orange) et pénalisés (rouge). Valeur de la pénalité (vert).

Le support est relatif à une pénalité. Pour la pénalité Lasso, le support correspond aux coefficients non nuls. La troisième étape consiste donc à chercher les coefficients qui minimisent le risque empirique non pénalisée, en retirant les variables dont les coefficients ont été mis à zéro par le Lasso de l'étape 1. Dans le cas de la pénalité TV, le support correspond aux coefficient regroupés par la pénalité. La figure 7.3 présente, en pointillés rouges, les coefficients obtenus après l'estimation avec pénalité TV. Lors de l'estimation sur le support, coefficients 1, 2 et 3 seront forcés à prendre la même valeur. Il en va de même pour les coefficients de 4 à 7 et pour les coefficients 8 et 9. En résumé, alors qu'il y avait 9 coefficients lors de l'estimation pénalisée, la restriction au support relative à la pénalité TV a regroupé ces coefficients en trois blocs.

7.5 Expérimentation numérique

7.5.1 Données simulées

Nous avons réalisé des simulations pour le modèle des Cox dépendant du temps avec des données de différentes tailles ($n = 100$, $n = 1000$ and $n = 10000$). Cela nous permet de comparer notre estimateur avec l'estimateur basé sur des splines et un group-Lasso proposé par Honda & Hardle (2014) et l'estimateur à base de noyau de la librarie `timereg` de Scheike et al. (2016). L'estimateur de Honda & Hardle (2014) (dont le code nous a gentiment été fourni par T. Honda) est implémenté seulement dans le cas où les individus ne réalisent qu'un évènement (pas d'évènements répétés). Nous avons donc restreint nos simulations à ce cas, bien que notre algorithme permette aussi de prendre en compte de evènements répétés.

Nous considérons un modèle de Cox où la fonction de risque instantané du temps d'intérêt T est $\lambda^\star(t, X) = \exp(\beta_0^\star(t) + X\beta^\star(t))$. Les covariables X_i, $i = 1, \dots, n$, sont des réalisation i.i.d. d'une loi aléatoire Gaussienne centrée de dimension p et de variance proportionnelle à la matrice identité. Soient $t \mapsto \lambda_\star(t, X_i)$ et les covariables X_i, $i = 1, \dots, n$, les temps T_i sont simulés comme la premier évènement d'un processus de Poisson nonhomogene d'intensité $\lambda_\star(t, X_i)$ via thinning (voir Lewis & Shedler (1979)).

Notre estimateur a été calculé en utilisant les poids de l'équation (6.14) et le paramètre réglable a été choisi par validation croisée. De plus, nous avons calculé un second estimateur, sans pénalité mais restreint au support déterminé par celle-ci, comme décrit précédemment.

Pour évaluer les performances de ces quatres estimateurs, nous avons réalisé $M = 500$ expérimentations Monte-Carlo à partir du modèle décrit plus haut. La qualité de l'estimation de chaque modèle est mesurée via une erreur des moindre carrés, comme définie ci-dessous

$$\mathrm{MSE}(\hat{\beta}^{\mathrm{M}}_{j,m}) = \frac{1}{M} \sum_{m=1}^{M} \int_0^\tau \left(\hat{\beta}^{\mathrm{M}}_{j,m}(t) - \beta^\star_j(t)\right)^2, \tag{7.15}$$

où $\hat{\beta}^{\mathrm{M}}_{j,m}$ est l'estimation de $\beta^\star_j$ sur l'échantillon m, pour $j = 1, \dots, p$. Les intégrales sont approximées en utilisant une grille de taille 50 régulièrement espacée sur $[0, \tau]$. Nous considérons alors une MSE cumulée définie comme

$$\mathrm{MSE}(\hat{\beta}^{\mathrm{M}}_{m}) = \sum_{j=1}^{p} \mathrm{MSE}(\hat{\beta}^{\mathrm{M}}_{j})_m. \tag{7.16}$$

De plus, nous évaluons la capacité de ces estimateurs à détecter les variables non actives, i.e celles qui n'ont aucune influence sur le taux de risque instantané. Pour cela, nous définissons

$$\begin{aligned}
\mathrm{TP}(\hat{\beta}^{\mathrm{M}}_{m}) &= \#\left\{j = 1, \dots, p : \hat{\beta}^{\mathrm{M}}_{j,m} \neq 0 \text{ and } \beta^\star_j \neq 0\right\} \\
\mathrm{FP}(\hat{\beta}^{\mathrm{M}}_{m}) &= \#\left\{j = 1, \dots, p : \hat{\beta}^{\mathrm{M}}_{j,m} \neq 0 \text{ and } \beta^\star_j = 0\right\} \\
\mathrm{TN}(\hat{\beta}^{\mathrm{M}}_{m}) &= \#\left\{j = 1, \dots, p : \hat{\beta}^{\mathrm{M}}_{j,m} = 0 \text{ and } \beta^\star_j = 0\right\} \\
\mathrm{FN}(\hat{\beta}^{\mathrm{M}}_{m}) &= \#\left\{j = 1, \dots, p : \hat{\beta}^{\mathrm{M}}_{j,m} = 0 \text{ and } \beta^\star_j \neq 0\right\}.
\end{aligned}$$

	CoxTV	Support	H&H	Timereg
$p = 5$	4.32	2.35	1.56	2.33
$p = 10$	4.70	2.53	1.45e+5	10.88
$p = 50$	4.93	4.39	NA	NA

TABLE 7.1 – Résultats en MSE pour $n = 100$. CoxTV est la méthode pénalisé que nous avons introduit, Support correspond à sa version non pénalisée avec restriction au support déterminé par CoxTV, H&H est la méthode de Honda et Hardle et Timereg correspond aux résultats de la librairie timereg.

Les résultats montrent que notre estimateur pénalisé performe bien en sélection de variable, en forcant les coefficients des variables non actives à zéro, voir Table 7.4 pour une

	CoxTV	Support	H&H	Timereg
$p = 10$	1.95	0.54	NA	0.62
$p = 50$	3.55	0.78	NA	4.56
$p = 100$	4.39	1.20	NA	23.81

TABLE 7.2 – Résultats en MSE pour $n = 1000$. CoxTV est la méthode pénalisé que nous avons introduit, Support correspond à sa version non pénalisée avec restriction au support déterminé par CoxTV, H&H est la méthode de Honda et Hardle et Timereg correspond aux résultats de la librairie timereg.

	CoxTV			H&H		
	TP	FP	TN	TP	FP	TN
$p = 5$	4.27	0.00	0.00	3.30	0.00	0.00
$p = 10$	2.38	0.98	4.02	4.20	3.00	2.00
$p = 50$	1.00	3.00	42.00	NA	NA	NA

TABLE 7.3 – Résultats en sélection de variables pour $n = 100$. CoxTV est la méthode pénalisé que nous avons introduit et H&H est la méthode de Honda et Hardle.

	CoxTV		
	TP	FP	TN
$p = 10$	4.47	1.92	3.08
$p = 50$	3.76	4.15	40.85
$p = 100$	3.61	6.78	88.22

TABLE 7.4 – Résultats en sélection de variables pour $n = 1000$ de la méthode CoxTV que nous avons introduit.

comparaison avec l'estimateur de Honda & Hardle (2014) dans le cas où $n = 100$. Pour des n plus larges, les performances de sélection sont toujours très satisfaisantes, voir Table 7.4. En conséquence, dans les suituations où p atteint ou dépasse $\sqrt{n}$, la version "Support" de notre estimateur offre de meilleures performances que les estimateurs de Honda & Hardle (2014) et de Scheike et al. (2016) en terme d'erreur des moindre carrés, voir les Tables 7.1 et 7.2.

De plus, notre méthode est plus rapide que celles de Honda & Hardle (2014) et Scheike et al. (2016) en terme de temps de calcul, dès que n est plus grand que 100, voir table 7.5.

	CoxTV + On supp.	Timereg
$n = 1000$ and $p = 10$	1.63	49.36
$n = 1000$ and $p = 50$	4.61	160.96

TABLE 7.5 – Temps d'estimation en minutes pour les méthodes CoxTV, sa version support, Honda et Hardle et timereg.

7.5.2 Données réelles

Nous illustrons notre modèle sur les données PBC (cholangite biliaire primitive), décrites dans Fleming & Harrington (1991), issues d'un essaie clinique de la Mayo Clinic, portant sur l'analyse de la cholangite biliaire primitive (PBC) du foie, conduite entre 1974 et 1984. Un total de 418 patients sont inclus dans le jeux de données et ont été suivis jusqu'à leur décès ou censurés. Nous nous concentrons sur les 8 premières années de l'étude et considérons les covariables suivantes : age, edema, log(bilirubin), log(albumin) et log(protime). Toutes les covariables sont centrées autour de leur moyenne. La figure 7.4 montre les coefficients cumulés estimés, i.e $(\hat{B}_j^{\mathrm{M}}(t) = \int_0^t \hat{\beta}_j(s)ds)$ sur les données PBC par les packages R coxtv and `timereg`. On observe sur cette figure que notre estimateur admet moins de variation que `timereg`, ce qui le rend plus facilement interprétable. Notons aussi que la variable age n'a pas été sélectionnée par notre modèle. En dehors de celle-ci, les résultats sont très similaires.

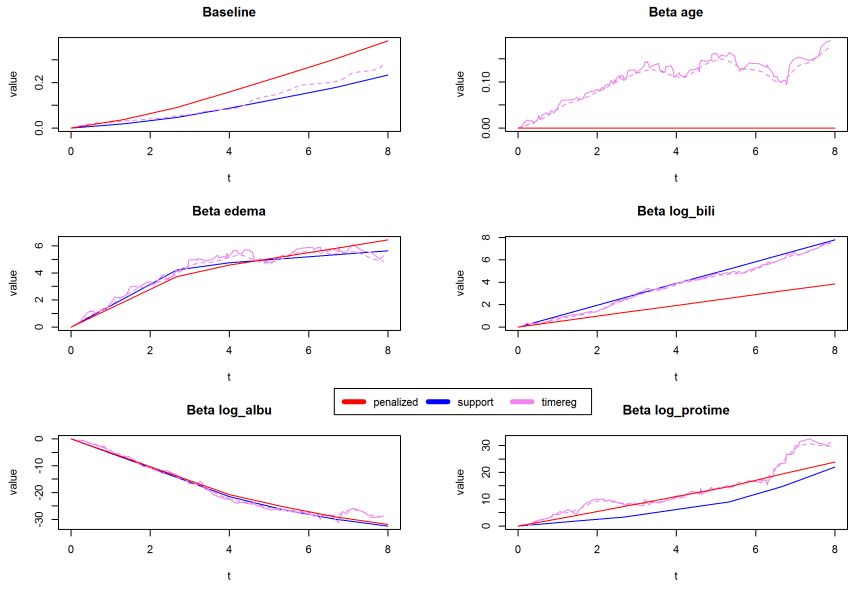

FIGURE 7.4 – Coefficients de régression cumulés estimés sur les données PBC par CoxTV(rouge), la version support (bleu) et `timereg` (rose).

7.6 Conclusion

Au cours des deux derniers chapitres, nous avons introduit une méthode pour l'analyse des temps d'évènements à partir de données longitudinales. Sa construction et son implémentation permettent l'analyse de grands jeux de données, comme ceux présent dans l'industrie du jeu vidéo. Nous avons détaillé les algorithmes utilisés et illustré leurs performances à travers des simulations. Sur des jeux de données de petite taille et lorsque la sélection de variable n'est pas nécessaire, les résultats sont proches de ceux obtenus par les

méthodes de Honda & Hardle (2014) et Scheike et al. (2016). Toutefois, notre méthode est la seule à pouvoir passer à l'échelle des grands jeux de données et à pouvoir effectuer une sélection de variables.

Nous disposons à présent d'un modèle et de son implémentation pour répondre à la problématique principale de la thèse : l'estimation de l'influence des éléments de design sur l'activité des joueurs. Dans les chapitres 8 et 9 suivants, nous utilisons ce modèle pour répondre à cette problématique.

Chapitre 8

Influence du design sur le temps de jeu des joueurs

Dans ce chapitre nous proposons une méthode pour aider à comprendre l'influence du design des jeux vidéo sur la rétention des joueurs. En nous basant sur les données de Far Cry 4, nous montrons comment utiliser la mesure du temps de jeu des joueurs pour détecter les moments où ils ont le plus de chances d'arrêter de jouer. Premièrement, nous montrons qu'il est facile de réaliser une comparaison de l'ensemble des jeux disponibles sur Steam, en utilisant des données publiques. Ensuite, nous introduisons comment l'analyse de survie peut aider à modéliser l'influence des variables de design sur la rétention des joueurs. Dans la pratique, l'environnement du jeu et les caractéristiques du joueur changent au cours du temps. Les systèmes d'information, utilisés dans l'industrie du jeu vidéo, permettent déjà de stocker l'ensemble de ces changements au cours de la vie du joueur. Cependant, les modèles actuels, qui prennent en compte ces données longitudinales, ne permettent pas de traiter des jeux de données de si grande taille. C'est pourquoi nous proposons un modèle permettant à la fois de prendre en compte l'information temporelle et capable de passer à l'échelle sur de grand jeux de données. D'autre part, nous considérons que l'effet d'une variable n'a pas, à priori, de raison d'être le même au cours du temps. Partant de ce constat, nous utilisons des coefficients pouvant eux-mêmes dépendre du temps. Enfin, nous utilisons ce modèle pour quantifier le lien entre l'utilisation des différentes armes de Far Cry 4 et la rétention des joueurs.

8.1 Introduction

L'industrie du jeu vidéo collecte de plus en plus d'information sur la manière dont leurs jeux sont joués. Depuis quelques années, la plupart des consoles et PC sont connectés à internet permettant aux studios de jeux vidéo de traquer potentiellement toutes les interactions du joueur avec l'environnement du jeu vidéo. Cependant, l'analyse et l'interprétation de ces grand jeux de données reste un challenge clé pour les entreprises du jeu vidéo.

L'objectif des développeurs de jeux vidéo n'est pas seulement d'avoir beaucoup de joueurs, c'est aussi que ces joueurs passent du temps dans ces jeux. Ils n'attendent pas que les joueurs se contentent d'essayer le jeu quelques minutes, mais qu'ils l'apprécient réellement et explorent le contenu que les développeurs ont mis du temps à créer. Le fait qu'une majorité des joueurs s'arrêtent dans les premières heures de jeu peut être considéré comme un échec du design, d'autant plus quand les joueurs ont acheté le jeu.

En conséquence, les studios de jeux, analysent souvent la rétention des joueurs, c'est à dire la proportion de joueurs restant dans le jeu après n heures de jeu. Ce chapitre se concentre sur l'utilisation des méthodes d'analyses statistique du comportement des joueurs permettant d'augmenter la rétention des joueurs. Nous souhaitons aider les designers à identifier les éléments de design qui font qu'un joueur continue l'aventure, et ceux qui au contraire le poussent à partir. Par nature, la conception d'un jeu vidéo est un processus itératif et les jeux doivent être testés pour déterminer si les joueurs ressentent du *fun* en y jouant Salen & Zimmerman (2003).

Il est possible d'apprendre beaucoup de choses en ne monitorant que le temps de jeu des joueurs. Cela permet de détecter quand un joueur est sur le point de quitter le jeu. Pour ne citer qu'un exemple, un problème dans le tutoriel du jeu se traduit par une chute de la rétention dès les premières minutes du jeu. Cependant, bien souvent, l'influence d'un élément de design n'est pas si facile à détecter. L'influence d'une arme mal équilibrée n'apparaitra pas clairement à la lecture du taux de départ, puisque chaque joueur utilisera (ou pas) cette arme à des temps qui lui sont propres. L'effet sera donc dilué sur différentes échelles de temps. Bien que ces effets ne soient pas visibles par une simple lecture de courbe, il est possible de les détecter en utilisant un modèle statistique approprié. Dans le but d'informer les designers sur les modifications à apporter au design, nous devons explorer comment les variables de jeu sont liés à la rétention.

Un des aspects les plus intéressant des systèmes d'enregistrement des jeux vidéo est qu'ils nous permettent d'accéder à un grand nombre de données précises sur le comportement des joueurs. Cela contraste avec d'autres domaines, tel que celui de la santé (aujourd'hui), où le nombre d'individus est plus proche de la centaine que des millions. D'autre part, dans les études médicales longitudinales, les patients peuvent êtres suivis toutes les semaines, alors que dans le jeu vidéo il est possible d'enregistrer chaque action à tout moment. Cette richesse de données peut nous aider à mieux comprendre le lien entre le design et la rétention des joueurs, mais pour cela, nous devons disposer d'un modèle capable de traiter de telles données.

8.2 Contribution et résultat principal

Ce chapitre propose une méthode pour extraire des informations actionnables à partir des données de suivi des joueurs. De nombreux modèles "boite noire", tels que les forets aléatoires ou les réseaux de neurones profond permettent de prédire le départ des joueurs.

Cependant, puisque notre objectif est de donner des indications aux designers sur les leviers d'amélioration, nous nous restreindrons à des modèles plus facilement interprétables.

Premièrement, nous décrivons comment la simple mesure du temps de jeu des joueurs permet d'identifier les périodes où les joueurs quittent le jeu. Nous montrons aussi que cette information peut être extraite de jeux de données public, et ainsi être utilisée par quiconque souhaite établir un point de référence afin de comparer les performances d'un jeu donné à ses concurrents. Certaines plateformes de jeux en ligne comme Steam, rassemblent des données sur leurs joueurs et ces données peuvent facilement être extraites. Cela signifie que les chercheurs peuvent aujourd'hui analyser des données caractérisant les habitudes des joueurs. Cependant, ces données sont souvent limitées au temps passé par les joueurs sur chaque jeu, et ne donnent pas d'aperçu détaillé de leur comportement au sein du jeu.

Par la suite, nous cherchons à modéliser le lien entre le comportement du joueur et son départ du jeu. Ainsi, nous présentons un modèle bien connu d'analyse de survie et montrons comment il peut s'appliquer avec des données constantes dans le temps. Ce type de modèle fonctionne bien avec des variables qui ne varient pas au cours du temps. Cependant, utiliser ce modèle avec des données qui dépendantes du temps, qui sont fréquentes dans les jeux vidéo, peut mener à des conclusions erronées.

Finalement, nous abordons la complexité des données variant en fonction du temps. Nous montrons comment l'information peut être visualisée et expliquons pourquoi il est important de prendre en compte l'information temporelle. Nous présentons un nouvel algorithme capable à la fois d'analyser de grand volumes de données et de prendre en compte les variables dépendant du temps. Nous montrons comment ce modèle peut s'appliquer à l'analyse de l'usage des armes dans Far Cry 4, et quels types de recommandations de design peuvent en découler.

8.3 État de l'art

Un jeu vidéo est un système multimédia interactif complexe. De fait, il peut être décrit par une multitude de variables quantifiables en fonction de son gameplay ainsi que, par exemple, de ses aspects techniques. Le gameplay caractérise la relation entre le joueur et le système de jeu et par conséquent, les variables de gameplay peuvent décrire en partie le système de jeu (par exemple : la cadence de tir d'une arme à feu), de même que le comportement du joueur (par exemple : son arme préférée). Des variables techniques peuvent décrire la position d'un élément d'interface ou bien la valeur moyenne des latences réseaux. Nombre de ces variables peuvent être ajustées par l'équipe de développement, et sont fréquemment qualifiées de fonctionnalités du jeu lorsqu'elles sont liées au gameplay. Nous utiliserons ici le terme plus abstrait de variables de jeu.

L'analyse statistique des jeux vidéo est un vaste domaine englobant des méthodes de visualisation Wallner & Kriglstein (2013), Bowman et al. (2012), de classification Bartle (1996), Yee (2006) et de prédiction Mahlmann et al. (2010), Xie et al. (2015). Nous nous

concentrerons ici sur les méthodes permettant de comprendre le lien entre les variables de jeu et le risque de départ des joueurs. Ce sujet a déjà été évoqué par plusieurs auteurs.

Weber (Weber et al. (2011)) a utilisé un modèle de régression afin de déterminer quelle variable de jeu a le plus d'influence sur la rétention du joueur. Au cours de cette recherche, Weber et al. étudie le cas d'un jeu de football américain dans lequel le joueur doit effectuer plusieurs matchs. Au lieu de prédire directement la durée du jeu, les auteurs se servent du nombre de matchs joués comme variable cible. Notre approche étend ce travail aux cas où l'on n'étudie pas des parties répétées, mais une mesure de durée plus générale, comme le temps de jeu.

Harrison et al.(Harrison & Roberts (2014)) ont utilisé des modèles N-gram pour attribuer des quêtes aux joueurs de façon dynamique. Ils peuvent donc sélectionner les séquences de quêtes que les joueurs sont plus à même d'accepter en vue d'accroître la rétention des joueurs. Cependant, ces modèles présentent une complexité exponentielle en fonction du nombre d'actions disponibles pour le joueur, et peuvent devenir inapplicables à des jeux de donnée comprenant de nombreuses variables de jeu.

La plus grande étude de la distribution du temps de jeu a été réalisée par Bauckhage (Bauckhage et al. (2012)) sur approximativement 250 000 joueurs sur Steam. Ils en ont conclu que le temps de jeu semble suivre une distribution de Weibull. Cependant, en utilisant les données de suivi des joueurs recueillies par Ubisoft®, nous observons que les jeux de l'éditeur semblent suivre une distribution log-Normale. Cela peut s'expliquer par des différences dans le système de recueil de données. De plus, la distribution du temps de jeu dans les Free To Play est assez différente de celle des jeux payants du fait de la grande proportion de *droppers* (des joueurs qui ne jouent qu'un seul jour). Sachant cela, l'utilisation d'un modèle de régression paramétrique (basé sur une distribution de densité) peut ne pas s'avérer pertinente pour chaque jeu. Nous résolvons ce problème par l'utilisation d'une approche semi-paramétrique.

L'analyse de survie a déjà été utilisée dans l'analyse de jeux vidéo. Chen et al.(Chen et al. (2009)) l'utilisent pour quantifier l'effet de la qualité du réseau sur la rétention des joueurs dans les jeux en ligne. Ils ont trouvé que la latence du réseau, et la perte de réseau avaient toutes deux un impact sur la rétention des joueurs. Nous proposons de compléter ce travail en incluant dans le modèle des informations variables en temps.

8.4 Méthode

Dans cette section, nous proposons une méthode pour analyser la rétention d'un jeu vidéo en commençant par une approche simple et générale, avant d'introduire une modélisation plus complexe des liens entre les variables de jeu et la rétention du joueur.

Premièrement, nous proposons d'analyser le temps de jeu et d'en extraire une courbe de survie horaire permettant d'obtenir un premier aperçu de la rétention du jeu, et de

le comparer à d'autres jeux dont les données ont été rendues publiques. Ensuite, nous présenterons une manière de quantifier le lien entre une variable de jeu et la rétention du joueur, à savoir le modèle de Cox. A partir de là nous mettons en évidence que de nombreuses variables de jeu changent au cours du temps. Cette évolution temporelle doit être prise en compte dans le modèle, sans quoi elle risque de conduire à des résultats erronés. Nous proposons un modèle permettant d'inclure des variables temporelles dans le modèle de Cox et capable de traiter de grands jeux de données.

8.4.1 Analyse du temps de jeu

Le temps de jeu est l'une des mesures les plus importantes dans l'analyse de jeu vidéo. Premièrement, il nous permet de savoir quand les joueurs s'arrêtent de jouer. A partir de là, nous pouvons représenter et calculer des mesures telles que la rétention, le taux de départ et la courbe de survie horaire.

8.4.1.1 Rétention

La rétention mesure combien de joueurs continuent à jouer après un temps donné. Nous nous concentrons ici sur la rétention en fonction du temps de jeu, mais elle peut être calculée en utilisant des mesures de temps différentes (nombre de jours écoulés depuis le lancement du jeu, nombre de jours joués,...). La rétention est représentée comme la courbe de survie des joueur. Soit T la variable aléatoire du temps auquel les joueurs quittent le jeu, nous avons alors : $S(t) = \mathbb{P}(T > t)$ Pour détecter les moments qui entrainent le départ de nombreux joueurs, on peut aussi regarder le taux de départ (*quit rate*).

8.4.1.2 Taux de départ

La fonction de risque instantanée, aussi appelée taux de départ dans l'industrie du jeu vidéo nous donne le taux instantané auquel les joueurs quittent le jeu :

$$h(t) = \lim_{\Delta t \to 0} \frac{\mathbb{P}(t \leq T < t + \Delta t | T \geq t)}{\Delta t} \tag{8.1}$$

La rétention et le taux de départ n'ont besoin que de la mesure de temps de jeu pour être calculés.

Une valeur localement haute du taux de départ peut indiquer un problème dans le jeu.

8.4.1.3 Survie horaire

En utilisant des intervalles d'une heure, le taux de départ peut être interprété comme le nombre de joueurs restants d'une heure à l'autre. Nous appellerons cela la survie horaire dans la section suivante. Cette mesure est moins fine que le taux de départ, mais se révèle

utile lorsque nous ne disposons pas de mesure du temps de jeu précise, comme c'est le cas dans nos données de comparaison.

8.4.2 Comparaison aux autres jeux

La rétention peut être analysée à la fois de manière relative et absolue. Premièrement, de manière absolue, toute inflexion dans la courbe peut révéler un problème de design et doit être examinée. Mais d'un point de vue relatif, comparée à la courbe de rétention d'autres jeux similaires, elle peut aider à comprendre où se situent les forces et les faiblesses du jeu. Il est difficile de déterminer si un taux de survie horaire à 85% pour la première heure est une bonne chose, cependant si à un moment donné le jeu s'avère moins performant que ses concurrents, alors il s'agit peut être d'un problème spécifique qui, avec un peu de chance, peut être corrigé.

Cependant, pour effectuer une telle analyse comparative, nous avons besoin de disposer de beaucoup d'informations sur le temps de jeu, et beaucoup d'entreprises de jeux vidéo ne partagent pas ce type d'informations. Heureusement, la plateforme de jeux en ligne de Valve Steam fournit une interface de programmation applicative (API) qui permet aux développeurs de rechercher une partie des informations sur les utilisateurs de Steam, telles que le temps de jeu passé sur les titres détenus par un joueur spécifique. Une telle API peut être utilisée pour recueillir régulièrement des informations sur un panel aléatoire de joueurs, et calculer à partir de là des statistiques sur le temps de jeu, comme l'on fait par exemple Bauckhage et al. (2012).

De plus, cette approche est utilisée par le site SteamSpy (Galyonkin (2016)) pour donner une approximation de la distribution du temps de jeu pour chacun des jeux disponibles sur Steam. Dès lors, si nous souhaitons obtenir des informations sur le temps de jeu passé sur les jeux Steam, au lieu d'interroger directement l'API de Steam, nous pouvons nous référer aux données de SteamSpy. Actuellement, SteamSpy ne fournit pas d'API mais les données peuvent être aisément récupérées manuellement sur le site web.

Bien entendu, les données de SteamSpy sont moins précises que les données internes des systèmes de traçage intrinsèques aux jeux. C'est pourquoi nous préférons nous restreindre à des intervalles de temps de une heure et utiliser une courbe de survie horaire pour effectuer les comparaisons. Chaque valeur représente un pourcentage de joueurs ayant joué au moins n heures et qui continueront à jouer durant au moins $n+1$ heures. Par exemple, nous avons calculé la courbe de survie horaire pour quelques jeux parmi les plus joués sur Steam, en nous basant sur les données de SteamSpy (fig. 8.1).

Ces courbes peuvent être interprétées comme suit : sur les joueurs de Counter Strike : Global Offensive qui ont joué au moins pendant 4 heures, 98% d'entre eux ont joué au moins une cinquième heure. Ce qui signifie que 2% des joueurs qui ont joué au moins 4 heures ont quitté le jeu entre la quatrième et cinquième heure de jeu.

Nous pouvons voir que DOTA 2 et Team Fortress 2 ont un taux de survie par heure

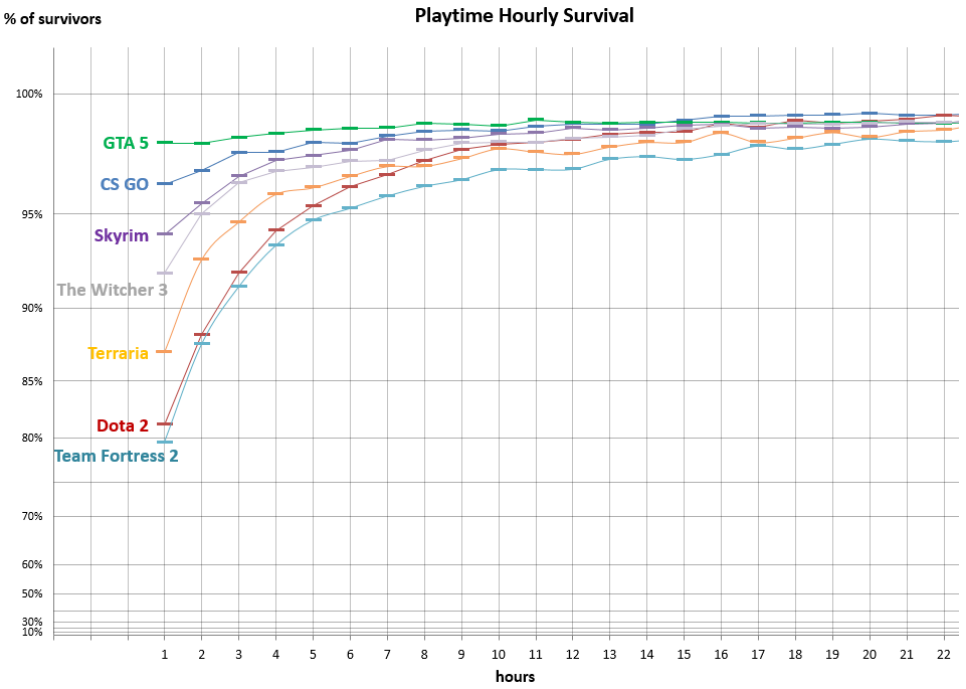

FIGURE 8.1 – Survie horaire de quelques jeux disponibles sur Steam.

aux alentours de 80% durant la première heure. ce qui signifie que 20% des joueurs quittent le jeu lors de la première heure. Nous pouvons supposer que, comme ces deux jeux entrent dans la catégorie des free to play, beaucoup plus de joueurs les ont essayés, alors qu'ils ne les auraient pas achetés s'ils étaient payants. Ces joueurs ont donc plus de chances d'abandonner le jeu, ce qui se traduit par un taux de survie par heure bas lors des premières heures de jeu.

A l'inverse, Grand Theft Auto V possède une courbe de survie par heure exceptionnellement plate. Le jeu parvient à retenir 98% des joueurs d'heures en heures.

L'observation des courbes des survie et du taux de départ peut aider à identifier des fenêtres temporelles où des nombreux joueurs quittent le jeu. Dans les aventures linéaires, cette information peut directement être reliée à un niveau précis du jeu. Cependant, dans les jeux non-linéaires, tels que les mondes ouverts où plus de liberté est laissé aux joueurs, il est souvent impossible de relier une mesure temporelle à une localisation spatiale ou à une mesure de progression, telle que la progression du joueur dans ses missions.

L'analyse du temps de jeu est ainsi intrinsèquement limitée, et bien qu'elle soit une première étape utile, il faut inclure d'autres variables pour construire un modèle plus complet.

8.4.3 Régression de Cox

Il existe de nombreuses manières de modéliser l'effet de covariables sur une mesure donnée. Dans notre cas, nous nous intéressons à la rétention qui est une valeur de durée. On peut modéliser la rétention des joueurs en effectuant une analyser de survie. Un des modèles les plus utilisés en analyse de survie est le modèle à risque proportionnel de Cox (Cox (1972)). Ce modèle considère que chaque joueur à un risque donné de quitter le jeu à un temps t. Le risque dépend des caractéristiques individuelles de chaque joueur, représentées par un ensemble de p variables de jeu $(X_1, \ldots, X_p)$ dans le modèle. Les caractéristiques d'un joueur peuvent être considérées comme une variable quantifiable, qui peut être mesurée pour n'importe quel joueur, et qui nous donne des informations sur une variable de jeu que nous souhaitons étudier. Par exemple, si nous voulons savoir si une Intelligence Artificielle (IA) n'est pas trop forte en combat rapproché, on peut utiliser la variable de jeu "AICloseCombatDamage", et calculer la caractéristique du joueur "NbDeathInCloseCombat" en comptant le nombre de morts pour chaque joueur en combat rapproché avec l'IA. Le coefficient β_j associé à "NbDeathInCloseCombat" peut alors nous aider à ajuster l'IA pour améliorer la rétention. Nous notons $X_{i,j}$ la valeur de la covariable j pour un joueur i donné. Le coefficient β_j détermine l'effet de la covariable j sur le taux de risque. Le risque que le joueur cesse de jouer à un temps t est modélisé de la manière suivante :

$$\lambda(t|X_i) = \lambda_0(t) \exp\big(\sum_{j=1}^{p} X_{i,j}\beta_j\big) \tag{8.2}$$

Ce modèle à risque proportionnel a été nommé ainsi car la fonction du taux de départ λ pour chaque joueur est proportionnelle à un fonction taux de départ de référence $\lambda_0(t)$.

Le modèle de Cox permet de modéliser l'influence de variables sur la rétention des joueurs. Cependant, nous devons prendre en compte le fait que bien que certaines variables soient constantes, la plupart des variables de jeu que nous voulons étudier varient fortement au cours du jeu. Par exemple, dans un jeu de tir, le joueur n'utilise pas la même arme tout au long du jeu. Dans un jeu de rôle, les aptitudes du personnage évoluent au cours du temps et la manière dont le joueur va les utiliser varie également. Notre modèle doit prendre en compte ces variations au cours du temps, en intégrant des variables dépendant du temps dans le modèle.

8.4.4 Données dépendantes du temps

Les variables peuvent changer très souvent au cours d'une session de jeu, comme le nombre d'ennemis tués par le joueur, la somme d'argent gagnée ou dépensée, ou encore sa position dans le jeu. Chaque caractéristique du joueur peut alors être considérée comme une série temporelle (fig. 8.2).

Une manière classique de traiter les données dépendant du temps consiste à ajouter

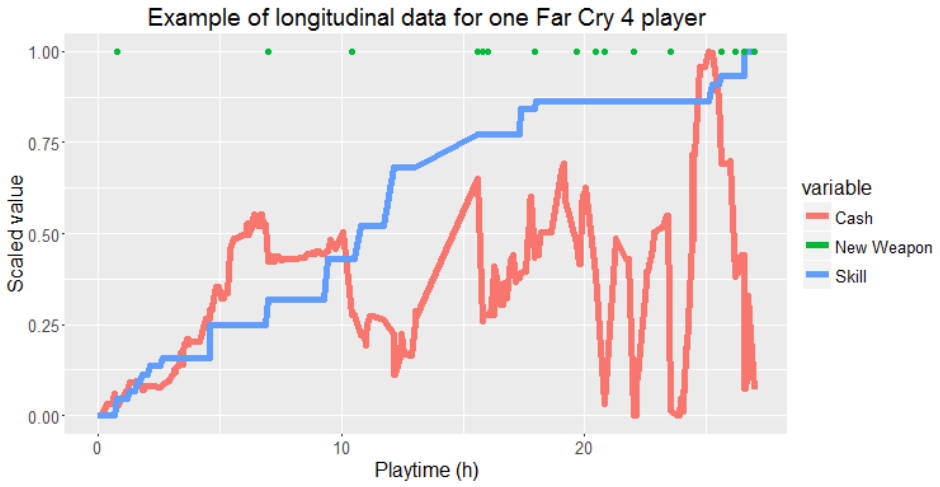

FIGURE 8.2 – Exemple de données longitudinales pour un joueur de Far Cry 4. La courbe bleu represente le nombre de compétences débloquées par le joueur. La courbe rouge correspond à l'argent (virtuel) dont dispose le joueur. Chaque point vert illustre l'acquisition d'une nouvelle arme par le joueur.

de nouvelles variables pour tenter de résumer l'information temporelle, en utilisant par exemple la moyenne $mean(x(t))$ et la variance $var(x(t))$ ou bien en décomposant la série sur des fonctions de base, telles que les ondelettes.

Cependant, résumer des variables dépendant du temps mène à une perte d'informations, et la décomposition en fonctions de base rend délicate toute interprétation. De plus, appliquer un modèle conçu uniquement pour des données constantes sur des données agrégées dépendantes du temps entraine un problème d'échelle temporelle. Les mesures prises sur un joueur qui a joué 5 minutes vont être comparées à celles prises sur un joueur qui a joué 80 heures. Il est possible d'essayer de créer un ratio en divisant les variables du jeu par le temps de jeu, mais l'évolution des variables au cours du temps n'est souvent pas linéaire. En fait, l'évolution des variables est liée au design et celles-ci ne peuvent pas être facilement comparées à des moments différents.

Pour éviter tous ces problèmes, nous présentons un modèle qui traite les variables du jeu dépendantes du temps $X(t)$. Pour pratiquement la même complexité, nous avons également permis l'évolution de l'effet des covariables au cours du temps $\beta(t)$, ce qui signifie qu'un même élément peut avoir différents effets au cours du temps.

8.4.5 Modélisation des données dépendantes du temps

Des extensions du modèle à risque proportionnel de Cox ont été proposées pour traiter les variables et les coefficients dépendantes du temps (Martinussen et al. (2002)).

$$\lambda(t|X_i) = \lambda_0(t) \exp\big(\sum_{j=1}^{p} X_{i,j}(t)\beta_j(t)\big) \tag{8.3}$$

Toutefois, les implémentations actuelles d'un tel modèle sont basées sur des inversions de matrice et des lissages itératifs par noyaux qui rendent le modèle inapplicable sur de très grands jeux de données. Nous proposons un modèle constant par morceaux qui rend le problème de minimisation séparable entre les individus. Ce qui nous permet d'utiliser un algorithme de descente de gradient stochastique tel qu'Adagrad (Duchi et al. (2011)) ou Adadelta (Zeiler (2012)) pour résoudre le problème d'optimisation. La partie technique et les garanties théoriques sont développées dans Alaya et al. (2016). L'algorithme de gradient stochastique nous permet de charger seulement quelques points de données en mémoire et ainsi d'analyser des jeux de données plus grands que la capacité de la RAM (mémoire vive).

Notre modèle a été implémenté dans une librairie C++ et une interface est disponible pour le langage de programmation R (R Core Team (2016)) grâce à la librairie RCpp (Eddelbuettel & François (2011)) afin de faciliter son utilisation.

Comme pour le modèle à risque proportionnel de Cox, chaque coefficient traduit le lien entre la variable associée et la rétention. Les coefficients dépendants du temps ne peuvent être résumés à un tableau de valeurs scalaires. A la place, nous devons tracer les courbes $\beta(t)$. Une valeur positive des coefficients signifie que la variable de jeu relative à ce coefficient a un lien positif avec le taux d'abandon, ce qui veut dire que le joueur a plus de chances de quitter le jeu dans un court intervalle de temps. Chaque courbe peut être interprétée comme suit : pour un temps donné t, si nous fixons toutes les valeurs des caractéristiques excepté celle étudiée $X_j(t)$, alors une augmentation d'un point de la valeur de cette caractéristique revient à multiplier les chances que les joueurs abandonnent le jeu dans une courte période de temps par $\exp(\beta_j(t))$.

8.5 Jeux de données

Far Cry 4 étant un jeu de tir, l'utilisation des armes fait partie du noyau dur de son gameplay. En conséquence, nous avons concentré cette première analyse sur l'utilisation des armes. Cependant, cette méthode peut être utilisée avec n'importe quelle covariable, dépendante du temps ou non.

8.5.1 Far Cry 4

Far Cry 4 est un jeu de tir à la première personne dans lequel le joueur explore un monde ouvert nommé Kyrat. Le jeu est sorti le 18 novembre 2014. Au cours de la partie, le joueur découvre de nouvelles armes. Il peut en porter jusqu'à quatre à la fois et les intervertir afin de modifier sa stratégie. Nous estimons que la sélection des armes est étroitement liée au style de jeu et peut dès lors donner un grand nombre d'informations sur la manière dont le

gameplay est expérimentée par le joueur. Un joueur qui voudrait jouer de manière furtive peut s'approcher d'un camp en utilisant une arme de longue portée, telle qu'un fusil sniper équipé d'un silencieux, puis un arc, et finalement parvenir à son objectif avec une arme de courte portée également dotée d'un silencieux. Un autre joueur pourra choisir de voler au dessus du camp avec un hélicoptère armé d'un lance-roquettes, tirer 3 ou 4 projectiles sur la zone la plus gardée avant de s'y poser et de dégainer un fusil à pompe pour terminer la mission. Ces deux approches diamétralement opposées démontrent qu'un même jeu (en l'occurrence Far Cry 4), peut proposer des expériences de gameplay très différentes selon les joueurs. De ce fait, si les joueurs qui utilisent le lance-roquette ont un taux rétention très différent de ceux qui utilisent le silencieux, cela peut nous donner un très bon aperçu des parties du gameplay que les développeurs doivent modifier.

8.5.2 Description des données

Le système d'enregistrement de Far Cry 4 mesure le nombre d'ennemis tués avec les armes suivantes :

— Assault (Fusils d'assaut) incluant AK-47, STG-90, F1, MS16, P416 et A52
— Auto Crossbow (Arbalète automatique)
— Bait (viande). Le joueur peut lancer de la viande pour appâter des animaux qui peuvent attaquer les ennemis. L'ennemi tué compte comme une victime du joueur.
— Bow (arc) comprenant l'arc recourbé et l'arc de chasse.
— Grenade
— Knife thrown (Lancer de couteau)
— Light Machine Guns (Fusils mitrailleurs légers) incluant PKM, U100, MKG, MG42 and DshK
— Machete (Machette). Seulement disponible dans l'arène de Shanath.
— Molotov cocktail (Cocktail molotov)
— Pistol (Pistolets) incluant Mark IV, M-712, 1911, 6P9, A.J.M. 9, D50 and .44 Magnum
— Rocket launcher (Lance roquettes) incluant M79, RPG-7, GL-94, GL-A87 and LK-1018
— Shotgun (Fusils à pompe) incluant D2, M133, 1887 and SPAS-12
— Submachine Guns (Fusils mitrailleurs) incluant A2000, MP5, A99, Slorpio, Vector .45 ACP, BZ19
— Sniper rifle (Fusils sniper) incluant SVD, M-700, SA-50, Z93

Les autres variables de jeu associées à l'action de tuer un ennemi sont :

— Alert (Alerte) : Le nombre d'ennemis tués avec une IA en état d'alerte.
— Animals (Animaux) : Le nombre d'animaux tués par le joueur.
— Cautious (Méfiance) : Le nombre d'ennemis tués avec une IA en état de méfiance.
— Combat : Le nombre d'ennemis tués avec une IA en état de combat.
— Distance Kills (Mort à distance) : Le nombre d'ennemis tués à une distance de plus

de 40 mètres.
— Close Kills (Mort proche) : Le nombre d'ennemis tués à une distance de moins de 40 mètres.
— Headshot (Tir dans la tête) : Le nombre d'ennemis tués par un tir dans la tête.
— Idle (Inactif) : Le nombre d'ennemis tués avec une IA en état inactif (non détecté).
— Silencer (Silencieux) : Le nombre d'ennemis tués avec une arme équipée d'un silencieux.
— TagCamera : Le nombre d'ennemis tués après qu'ils aient été marqués par le joueur.

8.5.3 Visualisation des informations dépendantes du temps

Pour les variables changeant au cours du temps, il est peu approprié de résumer l'information par la moyenne ou l'écart type. Dans ce cas, il est plus pertinent de visualiser comment cette variable change au cours du temps. Nous pouvons représenter cette variation au cours du temps, pour l'ensemble des joueurs, sous la forme d'une densité en deux dimensions (fig.8.3).

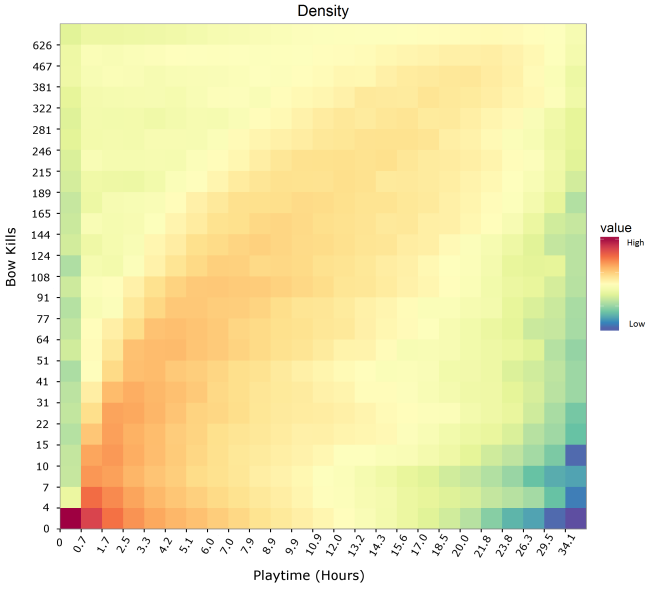

FIGURE 8.3 – Densité au cours du temps du nombre d'ennemis tués avec l'arc.

Les différents utilisateurs pouvant être suivis plus ou moins fréquemment dans la base de données, nous devons prendre certaines précautions. Chaque carré du graphique représente la densité des joueurs ayant, au temps de jeu considéré, une valeur donnée de la variable. Il est important, d'une part de ne considérer à l'instant t, que les joueurs encore actifs à ce moment. Et d'autre part de pondérer le calcul des densité par l'inverse du nombre d'observations obtenues pour un joueur donné, sans quoi un joueur fréquemment observé risque de biaiser le calcul.

L'inconvénient de cette visualisation est qu'elle ne traduit pas les trajectoires individuelles de chaque joueurs, mais uniquement la densité de la variable a un instant donné. Malgré tout, elle permet d'avoir un bon aperçu de la trajectoire suivie par les joueurs.

De plus, ces graphiques sont un bon moyen pour détecter les valeurs aberrantes. Par exemple, les carrés jaunes en haut à gauche de la figure 8.3 montrent que quelques joueurs ont tué des centaines d'ennemis à l'arc en moins d'une heure. Ce sont des valeurs aberrantes que nous devons retirer avant de poursuivre notre analyse.

8.6 Analyse

8.6.1 Rétention et taux de départ

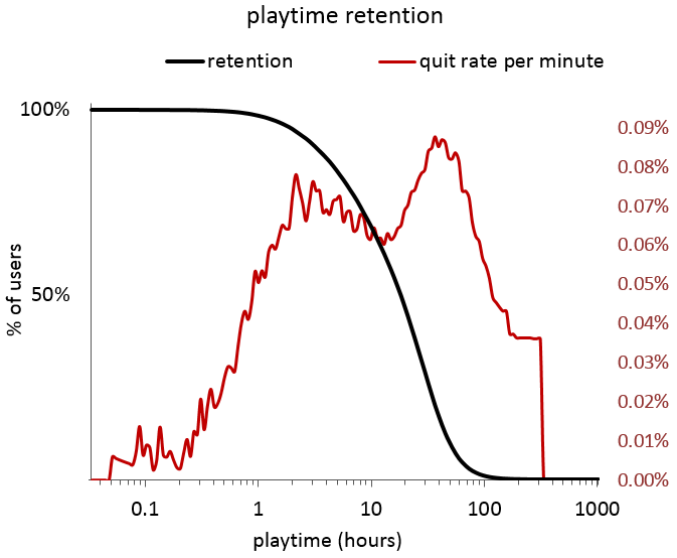

FIGURE 8.4 – Rétention (en noir) et taux de départ (en rouge) des joueurs de Far Cry 4.

Nous observons dans la figure 8.4 que le taux de départdes joueurs de Far Cry 4 possède deux pics principaux : un entre une et dix heures de jeu, et un entre 30 et 60 heures de jeu. Le second pic correspond aux joueurs ayant terminé le jeu. Cependant, finir un jeu est une notion subjective. Les joueurs ne finiront pas tous le jeu au même moment, pas seulement parce que certains d'entre eux atteignent un objectif plus vite que d'autres, mais également parce qu'ils ont des styles de jeu différents, et de fait, des objectifs différents Bartle (1996). Certains joueurs, plus intéressés par l'aspect jeu de tir, peuvent passer beaucoup de temps à réaliser des quêtes annexes du moment qu'elles impliquent de tirer sur des PNJ (personnages non joueurs), tandis que d'autres, plus motivés par l'histoire principale, arrêteront de jouer dès que cette dernière sera terminée.

Le premier pic entre 1 et 10 heures correspond aux joueurs qui ont quitté le jeu rapidement. A partir uniquement de ce graphique, il nous est très difficile de savoir pourquoi

ces joueurs ont quitté le jeu entre 1 et 10 heures. Mais comme nous savons que le taux d'abandon y est assez haut, il peut être intéressant d'examiner ces intervalles de temps de manière plus poussée, et de réaliser une analyse de survie plus détaillée.

8.6.2 Comparaison

Nous avons réalisé une comparaison en utilisant les données de SteamSpy, et avons représenté les courbes de survie horaire sur le graphique fig.8.5.

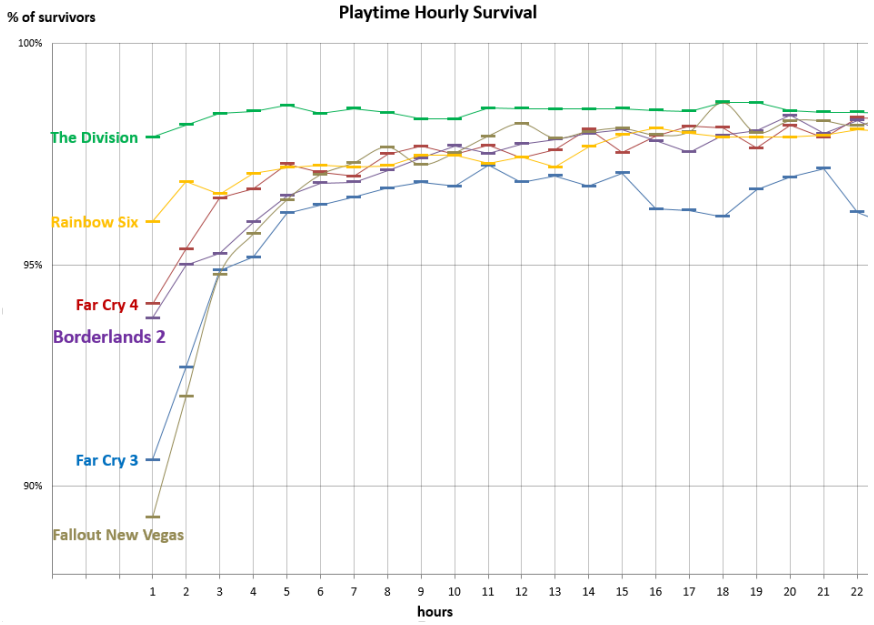

FIGURE 8.5 – Survie horaire de Far Cry 4 comparée à d'autres jeux.

Far Cry 4 possède un meilleur taux de survie par heure que Far Cry 3. C'est vrai sur le long terme autant que dans les premières heures de jeu. Le plus gros progrès a été réalisé dans les deux premières heures du jeu.

Borderlands 2 a une courbe de survie par heure assez similaire à celle de Far Cry 4, mais davantage de joueurs quittent le jeu entre deux et cinq heures.

Tom Clancy's Rainbow Six® : Siege est également un jeu de tir à la première personne, mais le gameplay est davantage orienté multijoueurs que la série Far Cry. Même si Tom Clancy's Rainbow Six : Siege a quasiment la même courbe de survie à long terme que Far Cry 4, nous pouvons observer que Rainbow Six a une meilleure rétention que Far Cry 4 dans les deux premières heures de jeu.

Tom Clancy's The Division est un jeu de tir à la troisième personne dans un monde ouvert, qui possède également des éléments de jeu de rôle plus poussé que ceux de Far Cry 4, à travers la progression des compétences et de l'équipement du joueurs. Il a globalement une bien meilleure rétention que Far Cry 4.

Nous n'observons pas d'inflexion particulière dans les courbes de Far Cry 4. Toutes les courbes suivent à peu près les mêmes trajectoires, et le jeu se situe dans la moyenne des titres similaires publiés par Ubisoft. En conséquence, il est difficile d'en tirer des recommandations de design spécifiques. Dans la section suivante, nous allons procéder à une analyse plus détaillée en utilisant l'analyse de survie.

8.6.3 Le modèle de Cox dépendant du temps

Les coefficients estimés sur le jeux de donnée relatif à l'utilisation des armes de Far Cry 4 sont montrés dans la figure 8.6. Rappelons, qu'un coefficient positif est corrélé positivement avec le départ du joueur. Dans la pratique, nous préfèrerons parler de l'effet de la variable sur la rétention. Dans ce cas, il faut s'intéresser à la valeur opposée au coefficient.

Trois armes ont un impact très positif sur la rétention des joueurs : la machette, le lance-roquette et le fusil à pompe. Le joueur ne peut utiliser la machette qu'en arène, ce qui nous informe sur un aspect spécifique du gameplay. Jouer en mode arène permet au joueur de débloquer des armes exclusives. Les joueurs qui passent du temps à débloquer de nouvelles armes ont de bonnes chances de continuer à jouer encore un moment.

L'utilisation répétée du lance-roquette et le fusil à pompe sont caractéristique d'un gameplay plus brutal. Il semble que ce soit un style de jeu apprécié car les joueurs qui tuent beaucoup d'ennemis avec ces armes ont tendance à jouer plus longtemps que les autres. D'autre part, les armes liées à un gameplay plus furtif comme le silencieux ou le fusil sniper ne semblent pas avoir d'impact positif sur la courbe de survie du joueur.

Peu de variables sont toujours significatives sur le long terme (après 20 heures de jeu), excepté le nombre d'ennemis tués à la machette ou à la grenade. Cet effet est dû en partie au fait que, tandis que le temps de jeu augmente, le nombre de joueurs restants diminue et, par conséquent, l'intervalle de confiance s'accroît.

L'arbalète est la seule arme corrélée négativement à la rétention. Il est à noter qu'elle n'a aucun effet avant 10 heures de jeu car c'est en moyenne le temps mis par les joueurs pour l'obtenir. Cela peut être dû à de multiples facteurs. Il pourrait être intéressant d'effectuer un playtest afin de comprendre pourquoi les joueurs ayant tué de nombreux ennemis à l'arbalète ont plus de chances de quitter le jeu que les autres.

8.7 Conclusion

Nous avons proposé une méthode pour analyser l'influence du design et du comportement des joueurs sur la rétention. Pour commencer, nous avons défini la rétention des joueurs et avons montré comment calculer le taux d'abandon et la courbe de survie par heure. Le taux d'abandon s'avère utile pour la détection de problèmes de design ponctuels, un pic dans la courbe indiquant une perte soudaine de joueurs. La courbe de survie horaire est plus adaptée aux jeux de données pour lesquels les mesures du temps de jeu sont moins

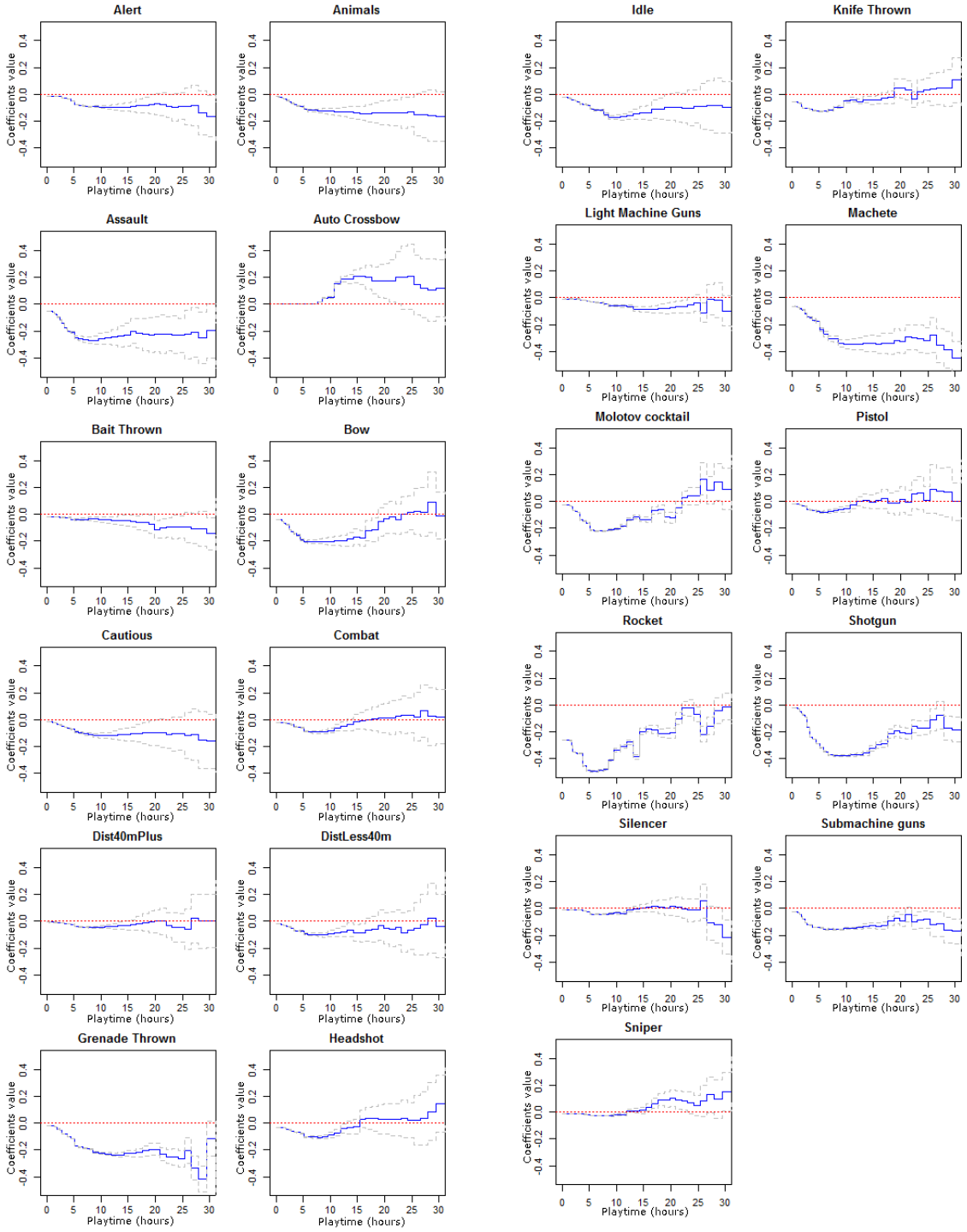

FIGURE 8.6 – Coefficients dépendant du temps (en bleu) estimés à partir des données sur l'utilisation des armes de Far Cry 4. Les lignes pointillés représentent les intercalles de confiance à 95%. La ligne constante à zero sert de référence pour l'effet nul.

précises, comme c'est le cas pour les données disponibles sur SteamSpy. Nous avons ensuite proposé un modèle basé sur l'analyse de survie pour évaluer le lien entre les variables du jeu et la rétention des joueurs, qui étend le modèle de Cox à des variables de jeu dépendantes du temps et de grands jeux de données.

Nous avons ensuite utilisé la méthode proposée pour analyser Far Cry 4. Tout d'abord, nous avons analysé la courbe du taux d'abandon, et avons observé un premier pic du taux d'abandon au début du jeu, entre 1 et 10 heures de jeu, ce qui demande un examen plus approfondi avec l'équipe de développement. Ensuite, la courbe de survie par heure n'a pas révélé de problème spécifique. Cette courbe est facile à calculer, mais sa mesure est moins fine.

Nous nous sommes pas la suite penchés sur les variables liées aux armes. Nous les avons sélectionnées car elles constituent une part essentielle du gameplay de Far Cry 4. De ces données, nous avons pu observer que beaucoup d'armes étaient corrélées positivement à la rétention. Globalement, il semblerait que les armes liées à un style de jeu furtif aient un lien moins positif avec la rétention.

Nous avons montré que notre approche fonctionne bien avec des données de bas niveau, telles que l'utilisation des armes. Mais pour établir des recommandations de design plus poussées, nous avons besoin de données de plus haut niveau.

La difficulté est l'un des facteurs principaux de motivation Malone (1981). Nous montrerons, dans le chapitre suivant (9) comment une mesure objective de difficulté, telle que celle proposée par Levieux (2011), peut être liée à la rétention des joueurs.

Chapitre 9

Influence de la difficulté au cours du temps sur la motivation

Dans ce chapitre, nous étudions le lien entre la difficulté et la motivation des joueurs dans deux jeux vidéo développés par Ubisoft® : *Rayman®Legends* and *Tom Clancy's The Division®*. Nous décrivons une méthode pour estimer la difficulté des joueurs au cours du temps et son lien, possiblement variable au cours du temps, avec la rétention des joueurs. Les résultats obtenus confirment les théories du flow et d'auto-efficacité. Nous notons toutefois, que les résultats différent entre les deux jeux au cours des premières heures de jeu. Nous expliquons cette différence au travers de la théorie de l'attribution : dans *Rayman Legends*, l'échec est principalement le reflet des compétences du joueur, alors que dans *Tom Clancy's The Division*, les caractéristiques de l'avatar jouent un rôle fondamental dans les chances de succès d'une mission. Et ces caractéristiques peuvent rapidement être améliorés par le joueur.

9.1 Introduction

Dès lors qu'il s'agit de favoriser et de maintenir la motivation d'un joueur de jeu vidéo, la difficulté est l'un des paramètres les plus important à ajuster. Différents auteurs ont tenté d'expliquer l'attrait pour les jeux vidéo. Nombre d'entre eux considèrent le challenge comme une caractéristique fondamentale. Cependant, le challenge et la difficulté sont des notions complexes et subjectives ; et leur lien avec la motivation est loin d'être évident. Dans ce chapitre, notre but est d'investiguer ces relations et si possible d'aider à la conception d'un design plus attractif.

Afin d'étudier l'aspect motivationnel des jeux vidéo, de nombreux auteurs observent un panel de joueurs dans des expériences contrôlés, utilisant parfois des jeux spécifiquement conçus pour l'étude. Ce type d'étude permet de faire des tests A/B et d'observer chaque comportements des joueurs. Cependant, de telles expérimentations ont un coût important et ne peuvent être maintenues sur de longues périodes de temps.

Pourtant, lorsqu'il s'agit d'étudier le challenge et la difficulté, le temps semble être un élément important à prendre en compte. Les connaissances et les compétences d'un joueur changent au cours du temps. De ce fait, la difficulté absolue doit évoluer à un rythme proche de la progression du joueur. Même d'un point de vue relatif, les joueurs peuvent ne pas apprécier le même niveau de difficulté au début du jeu, lorsqu'ils découvrent totalement les mécaniques de jeu, ou à la fin du jeu, lorsqu'ils combinent les mécaniques élémentaires pour découvrir des combinaisons plus avancés. D'un point de vue expérimental, le problème est que l'impact de la difficulté sur la motivation peut être dur à observer sur de courtes sessions de jeu, alors que la progression des connaissances et des compétences du joueur peut demander de nombreuses heures.

Hors des laboratoires, l'enregistrement de l'activité des joueurs est de plus en plus courante. Les machines de jeu actuelles sont pratiquement toutes connectées à internet, ce qui permet aux entreprises du jeu vidéo d'analyser le comportement de millions de joueurs. Ces données permettent aux développeurs d'avoir un retour plus détaillé sur leurs créations et de proposer des mises à jour en conséquence du comportement des joueurs. Ces enregistrements nous permettent d'étudier le comportement des joueurs en dehors des laboratoires, sur des jeux polis et de longues périodes de temps.

Dans le premier paragraphe, nous décrivons comment nous avons calculé la difficulté pour les deux jeux étudiés et rappelons quelques propriétés du modèle de survie utilisé. Ensuite, nous analysons le lien entre la difficulté et la rétention des joueurs mesurée dans les deux jeux. Enfin, nous expliquons comment ces résultats semblent confirmer les théories motivationnelles dans le contexte des jeux vidéo.

9.2 La difficulté dans les jeux vidéo

De nombreux auteurs considèrent que le challenge est l'un des attrait principal du jeux vidéo. Malone a étudié la motivation de jeunes enfants vis a vis des jeux vidéo. Il propose trois facteurs qui rendent les jeux vidéo si captivants : le challenge, la curiosité et la fantaisie Malone (1982). Dans son modèle, le challenge est directement lié à la difficulté et correspond à l'incertitude entourant l'atteinte d'un objectif par le joueur. Lazzaro propose un modèle à quatre facteurs, où le *Hard Fun* traduit le sentiment éprouvé lors de la réussite d'une tache ardue Lazzaro (2004). Sweetser et al voient dans le challenge une des partie les plus importante de leur *Game Flow framework* Sweetser & Wyeth (2005). Les travaux de Sweetser et al découlent de la théorie du Flow de Mihaly Csikszentmihalyi's Csikszentmihalyi (1991).

Csikszentmihalyi a tenté d'extraire les propriétés des activités reconnues comme forte source de motivation intrinsèque. Ses recherches ont établi que ces activités procurent *des impressions de challenge ou des possibilités d'action qui étendent (ni trop, ni trop peu) leurs aptitudes existantes* Csikszentmihalyi (1991). Ryan et al ont également étudié la motivation intrinsèque et ont appliqué la Théorie d'Auto-Détermination aux jeux vidéos en montrant

comment la satisfaction est liée au sentiment de *compétence*, qui repose sur un niveau optimal de challenge et, de ce fait, sur la difficulté du jeu Ryan et al. (2006). Jesper Juul a founi un aperçu de la manière dont l'échec, et donc la difficulté, est l'un des aspects clés de la satisfaction apportée par un jeu vidéo, et de la progression dans l'apprentissage Juul (2013).

Cependant, tandis qu'il semble y avoir consensus autour du fait que la difficulté soit un aspect central du jeu vidéo, il est toujours difficile de décrire précisément la relation entre la difficulté et la motivation dans les jeux vidéo. Premièrement parce que la difficulté, tout comme la motivation, sont des notions complexes qui ne peuvent pas être directement mesurées. Nous pouvons utiliser des questionnaires pour évaluer les aspects subjectifs de la difficulté et de la motivation, mais chaque mesure interrompant le jeu, cette évaluation ne peut de fait pas être réalisée souvent. C'est d'autant plus problématique si nous souhaitons évaluer l'impact de la difficulté sur la motivation au cours du temps, car nous pouvons difficilement nous fier à un questionnaire post-expérimentation pour évaluer précisement l'évolution de la difficulté au cours du jeu. Il est possible d'intégrer cette évaluation dans le gameplay, comme Constant et al l'ont fait pour la difficulté subjective (Constant et al. (2017)). Mais ce type d'évaluation est limité aux jeux basés sur le tour par tour et nécessite une modification du gameplay.

Une autre approche consiste à estimer la difficulté et la motivation directement à partir du comportement du joueur dans le jeu. Premièrement, la performance du joueur est liée à la difficulté du jeu. Si le jeu est difficile, le joueur perdra plus souvent, et ce taux d'échec peut être observé. Toutefois, comme Levieux et al le précisent, cette mesure de difficulté est également liée à la motivation du joueur Levieux (2011). En effet, la performance du joueur peut diminuer à la fois parce que le jeu est trop exigeant et parce que la motivation du joueur est retombée, et qu'il investit moins d'attention dans le jeu. Nous devons donc émettre les hypothèses suivantes afin d'étudier la difficulté d'un jeu à partie des performances des joueurs : lorsqu'il joue, un joueur fait de son mieux pour gagner. D'autre part, si la motivation du joueur descend en dessous d'un certain seuil, alors il arrête de jouer. Avec ces hypothèses, nous pouvons considérer que la performance du joueur nous permet d'estimer la difficulté objective du jeu, et que le lien entre la difficulté et la motivation est le même que celui entre la difficulté et la rétention.

Plusieurs recherches ont montré l'importance de l'aspect temporel du jeu Kim et al. (2008) et ont proposé une manière de le modéliser Wallner (2015). Dans ce chapitre, nous nous sommes particulièrement intéressés à l'impact de la difficulté sur la motivation au cours du temps. Premièrement, il est à noter que la difficulté varie au cours du temps dans un jeu vidéo. Le joueur s'améliore en jouant, mais le jeu augmente en permanence le niveau de performance requis. De plus, la difficulté n'a pas le même impact sur la motivation au début et à la fin du jeu. Klimmt et al ont réalisé une expérience sur un jeu de tir où le joueur préfèrait le plus bas niveau de difficulté. Ils ont suggéré qu'au début du jeu, la représentation de la difficulté par le joueur n'est pas calibrée et qu'il a tendance à apprécier

le fait d'avoir de nombreux retours positifs de la part du jeu et à ne pas réaliser que la difficulté est faible. Avoir une faible difficulté au début du jeu est également cohérent avec la théorie d'auto-efficacité : Bandura explique que l'échec diminue le sentiment de maitrise, plus particulièrement s'il a lieu dès le départ. Si l'échec survient après qu'une impression de grande efficacité se soit développée, alors l'échec devrait avoir un impact négatif moindre sur la motivation, voire avoir un effet positif si les joueurs savent que l'échec peut être surmonté avec un effort soutenu de sa part Bandura (1977).

L'influence de la difficulté au cours du temps peut être difficile à saisir au cours d'une expérience contrôlée, principalement en raison de la complexité et du coût liés au suivi des joueurs sur une longue période de temps. Heureusement, les mesures à distance dans les jeux vidéos nous donnent accès à l'historique complet des comportements du joueur. Dans ce chapitre, nous décrivons une méthode pour analyser ces données longitudinales, et appliquons cette méthode à deux jeux vidéos où les joueurs ont été suivis à distance lors de leurs sessions de jeu.

9.2.1 Estimation de la difficulté

Nous estimons la difficulté d'un challenge en nous basant sur les échecs et les victoires des joueurs sur ce challenge, tels que Levieux et al l'ont définie (Levieux (2011)). Soit X les variables de jeu qui représentent à la fois les caractéristiques du joueur et des challenges, nous estimons la difficulté comme la probabilité d'échec sachant ces caractéristiques.

$$\mathbb{P}(\text{fail}|X)$$

A cet endroit, n'importe quel modèle statistique peut être utilisé pour modéliser la difficulté. Cependant, pour modéliser la difficulté d'un challenge, nous avons besoin de variables décrivant à la fois le challenge et les compétences du joueur. Les données télémétriques décrivent souvent le taux de complétion d'un challenge, mais sont rarement assez précises pour décrire l'habilité du joueur vis a vis des mécaniques du jeu, en d'autre termes, ses compétences. Dès lors, nous avons besoin de prendre en compte la variation de compétences parmi les joueurs, c'est à dire le fait qu'un challenge ne représentera pas la même difficulté pour tous les joueurs. Nous faisons l'hypothèse que la variabilité des compétences est distribuée aléatoirement parmi les joueurs, en suivant une distribution normale. Par conséquent, nous pouvons modéliser une difficulté de challenge et prendre en compte cet effet aléatoire non observé en utilisant une régression logistique à effet mixte. Les caractéristiques du challenge sont modélisées comme des effet fixes et un effet aléatoire est ajouté pour prendre en compte les compétences de chaque joueur. Une fois que l'on dispose d'un modèle de la difficulté rencontrée par un joueur, associée à un challenge donné, aux caractéristiques du joueur et de ses compétences, nous pouvons estimer la difficulté.

Notre modèle de difficulté nous permet d'estimer la difficulté du jeu au cours du temps. En effet, pour chaque valeur de temps de jeu, nous connaissons la valeur des variables de

jeu associées au challenge en cours. Dès lors, nous pouvons calculer la difficulté estimée tout au long du temps de jeu [1] pour chaque joueur comme illustré dans le graphique 9.1. Une difficulté de 0,3 signifie que la probabilité que le joueur échoue lors du challenge sachant ses caractéristiques est de 30%.

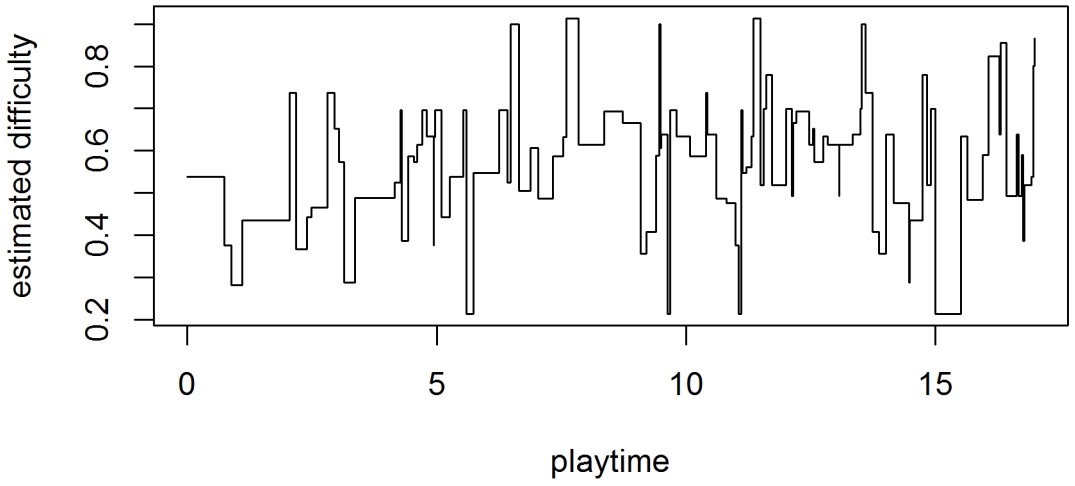

FIGURE 9.1 – Estimation de la difficulté pour un joueur de *Rayman Legends*

9.3 Analyse de survie

9.3.1 Rétention

La motivation est une construction psychologique complexe qui ne peut être directement observée. Dans ce chapitre, nous faisons une estimation de la motivation des joueurs au travers de la rétention. En effet, nous supposons que moins le joueur est motivé, plus il a de chances de cesser de jouer.

Beaucoup d'auteurs ont étudié l'impact des variables de design sur la rétention Weber et al. (2011), Debeauvais et al. (2014), Kawale et al. (2009), Borbora et al. (2011), Hadiji et al. (2014), Allart et al. (2016). Aucune de ces recherches ne s'est néanmoins concentrée sur l'impact de la difficulté sur la rétention. Notre objectif ici n'est pas de prédire la rétention mais plutôt d'être capable de mieux décrire le lien entre la difficulté et la rétention, et par conséquent la motivation du joueur.

1. Les unités de temps de jeu sont exprimées en heures pour chaque graphique

Pour ce faire, nous nous appuyons sur l'analyse de survie. Pour un aperçu de l'analyse de survie appliquée aux mesures de temps de jeu, voir Viljanen et al. (2017). Soit T une variable aléatoire associée au temps de jeu total des joueurs. La rétention est définie comme le nombre de joueurs restant sur le jeu après t heures de jeu, et est notée $S(t)$.

$$S(t) = \mathbf{P}(T > t) \tag{9.1}$$

Dans notre recherche, nous calculons le lien entre la difficulté et la rétention en utilisant un modèle de régression qui repose sur une vision légèrement différente de la rétention : le taux de départ. Le taux de départ, noté $\lambda(t)$, est la probabilité qu'un joueur cesse de jouer et ne revienne jamais, dans un court intervalle de temps, sachant qu'il jouait juste avant. Entre autres auteurs, ce modèle a été utilisé par Chen et al pour modéliser l'impact de la qualité du réseau sur les départs de joueurs (Chen et al. (2009)).

$$\lambda(t) = \lim_{h \to 0} \frac{\mathbf{P}\left(t < T \leq t + h | \, T > t\right)}{h} \tag{9.2}$$

Il est connu que l'on peut calculer le taux d'abandon à partir de la rétention et vice-versa. Une manière naturelle de modéliser l'influence des variables du jeu sur le taux d'abandon consiste à utiliser le modèle à risque proportionnel de Cox Cox (1972).

L'équation (9.3) montre une version étendue du risque proportionnel de Cox qui inclut des coefficients et des variables dépendants du temps.

$$\lambda(t) = \lambda_0(t) \exp\left(\sum_{j=1}^{p} X_{i,j}(t)\beta_j(t)\right) \tag{9.3}$$

Diverses approches ont été proposées pour traiter les variables dépendantes du temps Winnett & Sasieni (2003), les coefficients dépendants du temps Yan & Huang (2012) ou les deux Martinussen & Scheike (2007). Cependant, aucun d'entre eux n'inclut à la fois les coefficients et les variables dépendants du temps, et ils ne peuvent être étendus aux très grandes quantités de données disponibles dans l'industrie vidéoludique.

9.3.2 Estimation des paramètres du modèle de Cox

De manière analogue à de nombreuses méthodes statistiques, nous cherchons les valeurs de paramètres qui maximisent la probabilité de voir les évènements observés, c'est à dire, la vraisemblance du modèle. La log-vraisemblance du modèle à risque proportionnel de Cox est donnée dans l'équation (9.4) [2].

La log-vraisemblance peut être décomposée en deux parties. La première intégrale correspond à la probabilité donnée au modèle de voir un évènement se produire, lorsque cet

2. $N_i(t)$ est le processus de comptage associé à un évènement (départ d'un joueur), $Y_i(t)$ est égal à 1 avant que le joueur ne cesse de jouer, et à 0 après qu'il ait cessé de jouer

évènement s'est effectivement produit, tandis que la seconde intégrale correspond à la probabilité de ne pas avoir vu cet évènement auparavant.

$$\begin{aligned}\ell_n(\beta) =& \frac{1}{n}\sum_{i=1}^{n}\Big\{\int_0^\tau \sum_{j=0}^{p} X_{i,j}(t)\beta_j(t)dN_i(t) \\ &- \int_0^\tau Y_i(t)\exp\big(\sum_{j=0}^{p} X_{i,j}(t)\beta_j(t)\big)dt\Big\}\end{aligned} \tag{9.4}$$

9.3.3 Les coefficients et variables dépendants du temps

Dans les jeux vidéos, les variables changent au cours du temps : les caractéristiques de l'avatar change au fur et à mesure que le joueur équipe de nouvelles armes, la difficulté augmente peut à peu et connaît parfois un pic ou un creux, etc. Dans l'équation (9.4), nous notons $X(t)$ ces variables temporelles. La prise en compte de la temporalité des variables permet d'obtenir des résultats plus fiable, puisque nous ne résumons pas la complexité de l'information temporelle sous la forme de quelques scalaires, mais au contraire, nous prenons en compte la diversité des varaitions des variables au cours du temps.

L'influence des variables change également au cours du temps. Par exemple, la théorie d'auto-efficacité expose le fait qu'un haut niveau de difficulté peut s'avérer plus néfaste s'il se trouve au début du jeu. Nous pouvons modéliser ce lien dépendant du temps entre la difficulté et la motivation avec des coefficients dépendants du temps.

Afin de permettre aux coefficients de varier au cours du temps sans trop augmenter la complexité de calcul, nous présentons un estimateur constant par morceaux. L'intervalle de temps est discrétisé en plusieurs ensembles, chaque ensemble ayant son coefficient spécifique.

Soit $(I_l)_{l\in\{0,L\}}$ une partition de $[0,\tau]$, alors

$$\beta_j(t) = \sum_{l=1}^{L} \beta_{j,l}\mathbb{1}_{(I_l)}(t).$$

Pour éviter le fléau de la dimension, nous introduisons une pénalité par variation totale groupée (9.5), forçant ainsi le modèle à utiliser le moins de variations dans les paramètres que possible. La figure 9.2 illustre l'effet de la pénalisation par variation totale. Le modèle sans pénalité (lignes oranges) a 9 paramètres. La valeur de pénalité associée à ces coefficients est représentée en vert. Cela correspond au coût additionnel lié à la combinaison de coefficients successifs. Pour finir, en utilisant un modèle avec une variation totale, nous obtenons un modèle (en rouge) avec seulement 3 paramètres.

Cette pénalité peut être combinée avec une pénalité Lasso Tibshirani (1996) afin de réduire le nombre de variables de jeu, à l'exception de celles qui sont moins prédictives de

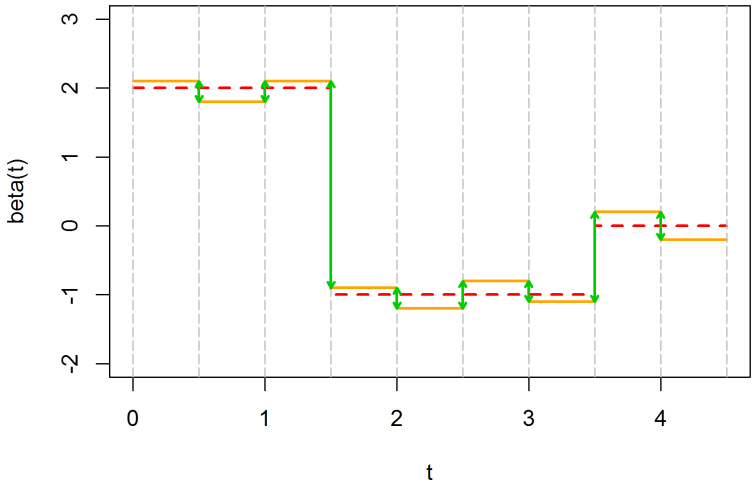

FIGURE 9.2 – Illustration de la pénalité par variation totale. Coefficients non pénalisés (orange) et pénalisés (rouge). Valeur de la pénalité (vert).

la rétention des joueurs.

$$\|\beta\|_{\mathrm{TV}} = \sum_{j=1}^{p} \left(|\beta_{j,1}| + \sum_{l=2}^{L} |\beta_{j,l} - \beta_{j,l-1}| \right) \tag{9.5}$$

Pour commencer, nous calculons les paramètres qui maximisent la log-vraisemblance tout en minimisant le terme de pénalité. Le choix du paramètre supplémentaire λ est défini par validation croisée.

$$\hat{\beta}^{\mathrm{TV}} \in \underset{\beta \in \mathbb{R}^L}{\operatorname{argmin}} \left\{ -\ell_n(\beta) + \lambda \|\beta\|_{\mathrm{TV}} \right\}, \lambda > 0 \tag{9.6}$$

Pour finir, une fois que nous disposons de l'estimation des coefficients pénalisés, nous calculons le support de ces coefficients. Cela consiste à rassembler les intervalles proches qui ont des valeurs similaires. Nous calculons ensuite les coefficients finaux en utilisant cette nouvelle discrétisation, sans pénalité.

9.4 Données et jeux

Dans cette étude, nous utilisons le modèle précédent sur les données internes d'Ubisoft pour estimer le lien entre la difficulté et la motivation sur deux jeux pour lesquels nous avons accès aux données longitudinales des joueurs.

Le premier jeu est *Rayman Legends*, un jeu de plate-formes développé par Ubisoft et sorti en 2013. Dans ce jeu, les joueurs doivent sauter entre des plate-formes pour atteindre la fin d'un niveau, en collectant un maximum d'items différents et en évitant de se faire tuer par les divers pièges.

Dans *Rayman Legends*, on considère que le joueur échoue à chaque fois qu'il quitte un niveau sans le terminer. Lorsque les joueurs meurent dans le niveau, ils réapparaissent au précédent checkpoint, mais la mort ou la réapparition ne sont pas traquées individuellement. De ce fait, nous calculons la difficulté comme la probabilité qu'un joueur quitte le niveau en cours.

Tom Clancy's The Division est un jeu de tir à la troisième personne en monde ouvert qui intègre aussi des mécaniques de jeu de rôle, également développé par Ubisoft et sorti en mars 2016. Dans ce jeu, le joueur doit accomplir plusieurs missions pour accroître les capacités de son personnage, pour débloquer de nouvelles régions du monde ouvert et pour en apprendre davantage sur l'histoire du jeu.

Dans *Tom Clancy's The Division*, nous calculons la difficulté en considérant qu'un joueur échoue à chaque fois qu'il n'arrive pas à terminer une mission, soit parce que son avatar est mort, soit parce qu'il n'a pas rempli un objectif de mission spécifique et a du recommencer. L'estimation de la difficulté est donc fonction du type de mission, de la difficulté sélectionnée par le joueur pour cette mission, et des diverses caractéristiques de son avatar [3].

Dans les deux jeux, l'ordre des niveaux n'est pas totalement imposé aux joueurs. Nombre d'entre eux sont débloqués à des moments spécifiques et le joueur peut choisir librement entre divers niveaux, voire recommencer ceux déjà terminés. Par exemple, dans *Tom Clancy's The Division*, le joueur peut choisir entre des missions annexes ou celles qui feront avancer l'histoire principale, et possède une indication de la difficulté de chaque mission en fonction du niveau de son personnage. Il peut également choisir de faire une même mission à différents niveaux de difficulté, sachant que plus la difficulté est élevée, plus les récompenses sont importantes.

9.5 Résultats

9.5.1 Difficulté estimée

Pour les deux jeux, nous estimons la difficulté en utilisant une régression par noyaux pondérée (fig. 9.3). L'aire sous la courbe ROC (Receiver Operating Characteristic) pour des régressions logistiques est de 0.80 pour Rayman Legends et de 0.81 pour The Division. L'estimation de la difficulté pour *Rayman Legends* part de 15% et augmente continuellement jusqu'à atteindre quasiment 30% après 30 heures de jeu. La difficulté augmente de

3. santé, dommages par seconde de l'arme primaire, de l'arme secondaire et des armes annexes, endurance, niveaux de compétences, électronique et armes à feu

façon régulière à mesure que les joueurs prennent des niveaux.

D'autre part, la difficulté de *Tom Clancy's The Division's* augmente rapidement au cours des premières heures de jeu, puis atteint un plateau aux alentours de 30%. Nous pouvons remarquer que la variation de la difficulté est plus grande pour les joueurs de *Rayman Legends* que pour ceux de *Tom Clancy's The Division*, comme indiqué par les lignes de quantiles.

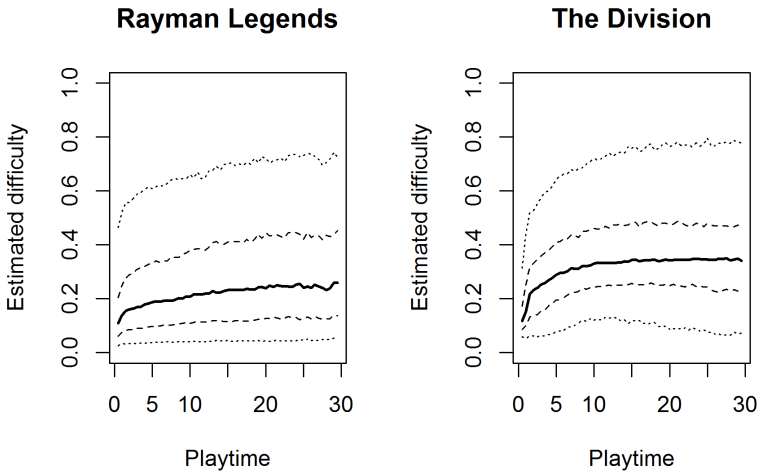

FIGURE 9.3 – Difficulté estimée de *Rayman Legends* et *Tom Clancy's The Division* : Médiane, quantiles 0.25 et 0.75 (tirets) et 0.05 et 0.95 (pointilles)

Les différences entre les courbes de difficulté peuvent s'expliquer de deux manières. Premièrement, dans *Tom Clancy's The Division*, le joueur peut très facilement réduire la difficulté du jeu, tandis qu'il ne peut pas le faire dans *Rayman Legends*. Dans *Tom Clancy's The Division*, la difficulté d'une mission dépend des compétences du joueur, mais aussi et surtout des caractéristiques de son avatar. Le fait est que le joueur a toujours la possibilité d'accomplir des missions annexes pour acheter de meilleures armes, améliorer son personnage, et ainsi diminuer la difficulté des missions suivantes. Dans *Rayman Legends*, le joueur a le choix entre de nombreux niveaux, mais ne peut pas changer leur difficulté. Si chaque niveau disponible est difficile, le joueur n'aura pas d'autres choix que de les terminer pour pouvoir progresser.

De plus, dans *Tom Clancy's The Division*, l'échec est bien plus punitif que dans *Rayman Legends*. Lorsque le joueur entame une mission, il doit se rendre sur le lieu de la mission depuis son abri, et les missions sont souvent plus longues qu'un niveau de *Rayman Legends*. Le rythme moins soutenu de *Tom Clancy's The Division's* peut amener les joueurs à chercher à perdre le moins possible et à être mieux préparés pour chaque mission dans le but d'éviter la frustration engendrée par le temps perdu. A l'inverse, dans *Rayman Legends*, le joueur peut recommencer un niveau très rapidement.

9.5.2 Lien avec la rétention

9.5.2.1 Rayman Legends

Premièrement, nous avons étudié l'effet de la difficulté sur le taux de départ, afin d'estimer quel niveau de difficulté entretient le plus haut niveau de motivation chez les joueurs. Les résultats obtenus par le modèle à risque proportionnel de Cox dépendant du temps sont montrés sur le graphique 9.4 pour *Rayman Legends*[4].

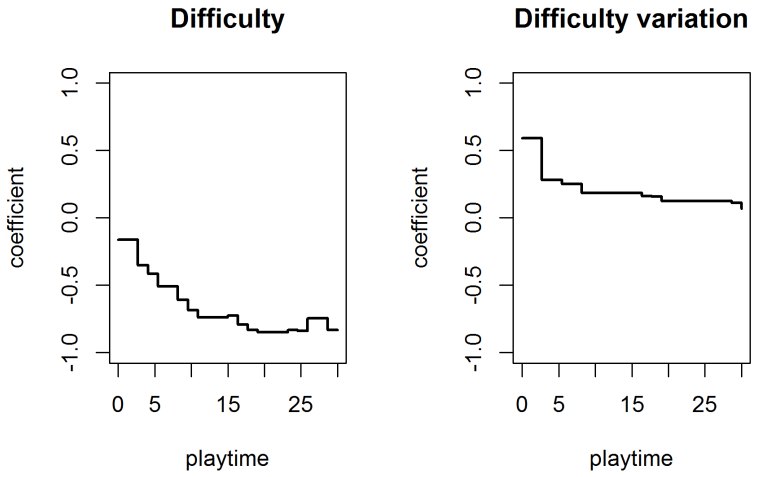

FIGURE 9.4 – Coefficients dépendant du temps pour *Rayman Legend*

Dans *Rayman Legends*, nous pouvons observer que la difficulté a un lien négatif avec le taux de départ. Cela revient à dire que lorsqu'ils sont confrontés à des niveaux plus difficiles, les joueurs ont moins tendance à abandonner. Ce lien entre difficulté et rétention devient plus fort au cours du temps, se rapprochant de -0.1 au début et de -0.8 à la fin.

Plus précisément, la courbe de difficulté peut être interprétée comme suit. Admettons que nous prenons deux joueurs A et B et qu'ils entament leur 12ème heure de jeu. Selon leurs compétences et le niveau qu'ils choisissent, la difficulté peut être de 20% pour le joueur A et de 40% pour le joueur B. L'écart de difficulté entre eux est de 20%. Sur la courbe de gauche du graphique 9.4, nous pouvons voir que la valeur de coefficient associée à la difficulté au bout de 12 heures de jeu se situe aux alentours de -0.7. Dès lors, les chances que le joueur A quitte le jeu rapidement sont $\exp(-0.2 * (-0.7)) = 1.15$ fois plus grandes que pour le joueur B. A l'inverse, le joueur B a 15% de chances en plus de rester sur le jeu que le joueur A.

Si la causalité est avérée, ces résultats signifient qu'en augmentant la difficulté, plus particulièrement dans les dernières heures du jeu, on peut augmenter la rétention des

4. A noter que nous n'avons pas représenté les intervalles de confiance sur les graphiques 9.4 et 9.5 car ils sont faibles et impactent négativement la lisibilité

joueurs. De plus, nous pouvons donner une estimation du gain de rétention induit par ce changement. Cependant, comme il ne s'agit pas d'une expérience contrôlée où les joueurs sont obligés de joueur à un niveau de difficulté donné, nous ne pouvons dès lors pas conclure à une relation de causalité. D'autre part, de nombreuses informations non observées peuvent affecter à la fois la difficulté et la rétention. Cette expérience permet de d'extraire des corrélations intéressantes qui doivent être approfondies lors d'expériences contrôlées afin d'en déduire un lien de causalité.

Ensuite, nous étudions l'effet de la difficulté sur le taux d'abandon. En effet, les joueurs ont tendance à préférer un niveau plus haut de difficulté, mais d'un autre côté, ils pourraient ne pas apprécier que la difficulté passe brutalement de facile à difficile, ou inversement. La courbe de droite sur le graphique 9.4 décrit l'effet de la variation de difficulté sur le taux d'abandon. Ici, la variation de difficulté doit être interprétée comme la dérivée de la difficulté au cours du temps.

Les résultats montrent que dans *Rayman Legends*, une variation importante de la difficulté dans les premières heures de jeu aura un impact négatif sur la rétention des joueurs. Une fois encore, si la causalité est avérée, et au vu des résultats précédents, on pourrait recommander à l'équipe de design d'augmenter légèrement la difficulté dans les premières heures de jeu, puis de l'accroître de manière plus prononcée au fur et à mesure que le temps de jeu augmente.

9.5.2.2 Tom Clancy's The Division

Le graphique 9.5 montre la même analyse sur *Tom Clancy's The Division*. Ici aussi nous remarquons qu'une difficulté plus élevée est corrélée avec une plus grande rétention. Cependant, cet effet reste marqué même durant les premières heures de jeu. Au début du jeu, un joueur avec une difficulté de 30% a 21% ($\exp(-0.2 * (-0.95))$) de chances en plus de continuer à jouer qu'un joueur ayant une difficulté à 10%. Cet effet atteint 27% après 8 heures de jeu.

Dans *Tom Clancy's The Division*, la variation de difficulté n'a pas d'influence sur la rétention des joueurs. Même une augmentation brutale de la difficulté ne semble pas avoir d'impact sur le taux de départ des joueurs.

Ces deux résultats sont cohérents avec les aspects motivationnels du jeu vidéo que nous avons décrits en début de chapitre. Premièrement, la théorie du flow établit que les joueurs ont tendance à préférer un niveau optimal de challenge. Cependant, nous ne connaissons pas ce seuil idéal de difficulté. Nous pouvons postuler que la difficulté est équilibrée lorsque le joueur a une chance sur deux de gagner. Si nous regardons le graphique 9.3, nous observons que la difficulté se situe largement en dessous de ce seuil pour les deux jeux. Si un joueur a entre 70% et 80% de chances d'atteindre un objectif, le niveau peut être considéré comme moins *difficile* que ce qu'il devrait être. Ainsi, tel que Malone l'expose, un challenge est motivant parce qu'il est source d'incertitude, mais il peut ne pas l'être si un joueur gagne

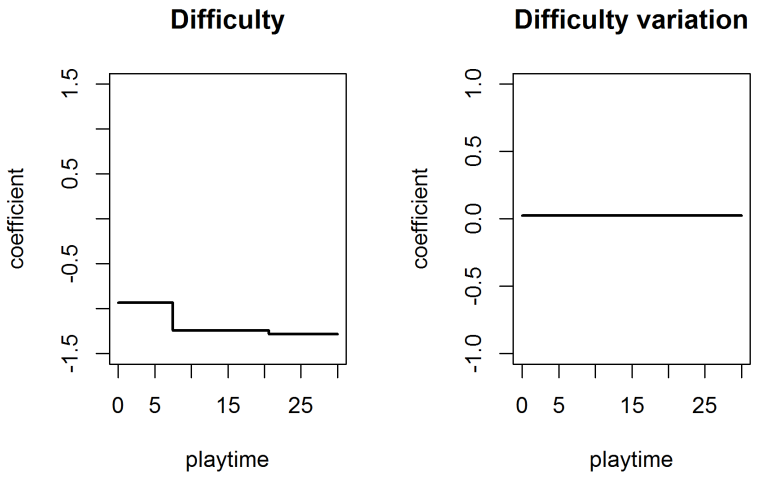

FIGURE 9.5 – Coefficients dépendant du temps pour *Tom Clancy's The Division*

aussi souvent.

Dès lors, les résultats de *Rayman Legends* sont cohérents avec la théorie d'auto-efficacité ainsi qu'avec l'expérience de Klimmt. Au début du jeu, les joueurs ont tendance à préférer un niveau de difficulté moins élevé, et n'apprécient pas une variation trop forte de ce niveau. En effet, nous pouvons postuler que les joueurs ont besoin de pouvoir s'entraîner sans risque au début du jeu pour gagner suffisamment confiance dans le fait qu'ils parviendront à relever tous les challenges que le jeu propose. De ce fait, l'échec, qu'il soit du au joueur ou à un évènement imprévisible peut être dommageable. Mais après quelques heures de jeu, les joueurs réalisent que le jeu est juste, croient en leur capacité à le finir, et sont prêts à apprécier des niveaux de difficulté plus élevés et plus variables.

Ce qui peut surprendre, au premier abord, est que les résultats ne sont pas les mêmes sur *Tom Clancy's The Division*. Dans ce jeu, les joueurs semblent toujours préférer des niveaux de difficulté plus élevés, mais cet impact de la difficulté sur la rétention semble varier beaucoup moins au cours du temps. Par ailleurs, il semblerait que la variation de la difficulté au cours du temps n'ait aucun impact sur la rétention des joueurs. Toutefois, nous pouvons expliquer ce résultat au travers de la Théorie de l'Attribution de Weiner Weiner (2005). Dans *Rayman Legends*, les compétences du joueur constituent la principale cause d'échec. Cette cause peut être maîtrisée, mais elle a un locus de contrôle intrinsèque, et ne varie pas rapidement au cours du temps. Dans *Tom Clancy's The Division*, lorsque le joueur échoue, cela est également dû à ses compétences, mais aussi au fait que son personnage n'était pas assez fort. Heureusement, cette dernière cause est externe, et peut être rapidement maîtrisée en effectuant quelques missions annexes ou en achetant de meilleures armes. Cela peut expliquer pourquoi l'échec semble avoir un impact bien plus négatif sur la motivation dans *Rayman Legends* que dans *Tom Clancy's The Division*.

Une dernière chose à noter dans *Tom Clancy's The Division* est que, même si les joueurs peuvent changer la difficulté du jeu en changeant le niveau de la mission ou en améliorant son personnage, les résultats montrent que les joueurs préfèreraient que le jeu soit plus difficile. Ceci peut s'expliquer par le fait que la progression du personnage ne peut pas uniquement être interprétée en termes de difficulté : les joueurs préfèrent avoir un personnage plus fort en soi, mais le regrettent par la suite lorsque le jeu devient trop facile. De même, comme nous l'avons précisé dans la section 9.5.1, l'échec est bien plus punitif dans *Tom Clancy's The Division*, et dès lors, les joueurs font beaucoup plus attention en jouant afin d'éviter la frustration de l'échec, aux dépens d'un niveau de difficulté équilibré.

9.6 Conclusion

Dans ce chapitre, nous étudions le lien entre la difficulté et la motivation dans les jeux vidéo en utilisant des données télémétriques provenant de deux jeux développés par Ubisoft : *Rayman Legends* et *Tom Clancy's The Division*. Nous définissons la difficulté comme une probabilité d'échec, que nous estimons en utilisant une régression logistique à effets mixtes. Nous estimons la motivation du joueur en considérant que lorsqu'elle est trop basse, il cesse de jouer. Nous utilisons ensuite une analyse de survie avec des coefficients et des variables dépendants du temps pour estimer le lien entre la difficulté et la motivation du joueur.

Nos résultats tendent à confirmer les théories sur la motivation des joueurs. Premièrement, la difficulté est en soi une variable explicative de la rétention des joueurs. Ensuite, sur les jeux étudiés, les joueurs ont tendance à préférer une difficulté plus élevée. La théorie du flow n'établit pas de seuil optimal de difficulté, mais dans les deux jeux, la probabilité d'échec est presque toujours en dessous de ce que nous pouvons appeler une difficulté équilibrée, à 50% de chances de réussite. Plus précisément, dans *Rayman Legends*, les joueurs ne souhaitent pas de changement brutal de difficulté, de même qu'une difficulté plus élevée au cours des premières heures de jeu. Ceci est cohérent avec la théorie d'auto-efficacité : l'échec, s'il survient au début du jeu, peut nuire à leur conviction qu'ils sont capables de réussir. Ce point est essentiel dans le fait que nous n'observons pas les mêmes résultats au début de *Tom Clancy's The Division*. En effet, dans *Tom Clancy's The Division*, les causes d'échec peuvent être attribuées au fait que le personnage ne soit pas assez fort, ce qui peut être rapidement modifié. Dans *Rayman Legends*, l'échec est principalement dû à un manque de compétences de la part du joueur, et est de ce fait bien plus dommageable à sa motivation au cours des premières heures de jeu.

Conclusion

Bilan des travaux

L'un des objectifs de la thèse est d'analyser les grandes bases de données dont dispose l'industrie du jeu vidéo pour en extraire des recommandations de design. Le jeu vidéo est, en l'occurrence, un domaine d'application de problèmes plus généraux portant sur l'analyse et le traitement des masses de données. Notre problématique c'est centrée sur quatre questions :

1. Existe-t-il un modèle d'analyse des temps d'évènements à partir de données longitudinales, dont les résultats sont facilement interprétables et capable à la fois de traiter de grands jeux de données et un grand nombre de covariables ?

Afin de répondre à cette question, nous nous sommes tournés vers les modèles d'analyse de survie permettant de prendre en compte des données longitudinales.

2. Dispose-t-on d'une implémentation de ce modèle, pouvant traiter des jeux de données de grande taille en temps raisonnable ?

Dans le chapitre 6, nous avons introduit un nouveau modèle permettant de prendre en compte à la fois des variables dépendantes du temps et des coefficients variables au cours du temps. Ce modèle est construit pour passer facilement à l'échelle et permettre l'analyse de grands jeux de données. D'autre part, il permet d'effectuer une sélection de variables au travers de la pénalité Lasso. Le chapitre 7 propose une implémentation de ce modèle dans le package R coxtv.

Une fois ces deux problèmes résolus, nous avons traité les deux questions suivantes :

3. Peut-on extraire de l'analyse de l'activité des joueurs des recommandations de design spécifiques au jeu étudié ?

Nous avons montré dans le chapitre 8 comment modéliser l'influence des variables de design sur la rétention des joueurs. Ces travaux font l'objet d'une publication dans IEEE Computational Intelligence and Games (Allart et al. (2016)).

4. Est-il possible d'extraire des recommandations de design plus larges, sur des variables partagées entre plusieurs jeux ?

Dans le chapitre 9 nous avons établi un premier lien entre la difficulté et la motivation des joueurs. Ces résultats sont publiés dans la conférence internationale Foundations of Digital Games (Allart et al. (2017)).

De façon générale, nous avons essayé, au travers de ce document, de montrer l'importance du facteur temps sur les évènements et le besoin de l'intégrer dans nos analyses. Les caractéristiques d'un individu, qu'il s'agisse d'un joueur de jeu vidéo, d'un patient ou d'un consommateur, changent au cours du temps et il apparait important de prendre en compte cette variabilité. D'autre part l'effet d'une caractéristique donnée peut lui même changer au cours du temps et les grands volumes de données dont on dispose nous permettent de modéliser ces changements. Ce modèle nous a permis de répondre a quelques unes des questions de design soulevées par l'industrie du jeu vidéo.

Perspectives

D'un point de vue général cette thèse est une contribution à l'analyse statistique des grandes masses de données. Elle se situe donc dans le domaine très actif de l'utilisation de ces données via des algorithmes dits d'intelligence artificielle. Nous avons d'emblée situé notre approche dans le contexte de l'apprentissage statistique. Notre analyse est focalisée sur une problématique liée à un temps d'arrêt, le moment ou un joueur quitte le jeu. Historiquement les modèles utilisés sont issus de la théorie de la fiabilité des systèmes. L'interprétation des courbes de survies est alors relativement simple en fonction de phénomènes physiques d'usure ou d'apparition de pannes catalectiques. Dans notre cadre l'analyse est plus complexe puisqu'il s'agit d'un comportement lié à la psychologie et la sociologie des joueurs. Pour aboutir à des conclusions nous nous sommes appuyés sur la théorie du Game Design, mais également sur la connaissance des pratiques des joueurs au sein d'Ubisoft. Nous devons poursuivre sur cette voie afin de mieux comprendre le comportement des joueurs.

Du point de vue des techniques statistiques, une amélioration importante de nos travaux résiderait dans l'ajout d'une mesure de l'importance des variables de notre modèle. Cependant, à notre connaissance aucune méthode ne permet de mesurer l'influence des variables dans un tel cas. De récents travaux abordent cette problématique dans le cas du modèle de Cox où ni les coefficients, ni les variables ne dépendent du temps (Viallon & Latouche (2011)). Mais ces résultats ont besoin d'être étendus aux cas où variables et coefficients dépendent du temps.

Nous avons utilisé dans ce travail la variation totale (TV), qui vise à regrouper les coefficients adjacents sous un seul et unique coefficient, si leur différence n'est pas justifiée par les données. Par construction, cette pénalité suppose qu'il existe un ordre sur les coefficients. Cet ordre peut être naturel, c'est le cas pour les coefficients dépendants du temps que nous avons considérés dans cette thèse. Cependant, plutôt que de regrouper des coefficients adjacents on peut vouloir regrouper n'importe quels coefficients entre eux, ce qui revient à effectuer une factorisation, du type $Y \sim \beta_1(X_1 + X_3) + \beta_2(X_2 + X_4 + X_5)$ et permet ici de passer d'un modèle à cinq paramètres à un modèle à deux paramètres.

Différents travaux ont déjà proposé d'étendre la variation totale au cas d'un graphe Tibshirani (2011). Nous considérons ici le cas où ce graphe est complet, c'est à dire que l'ensemble des coefficients est relié à tous les autres. C'est pourquoi nous utiliserons pour cette pénalité le nom de variation totale complète et la noterons cTV (pour complete Total-Variation).

$$\|\beta\|_{\mathrm{cTV}} = \sum_{k \neq j} |\beta_k - \beta_j| \tag{9.7}$$

Nous pensons qu'il est possible de calculer l'opérateur proximal de cTV à partir de celui de TV, et nous continuons nos recherches pour en obtenir une démonstration théorique.

Bibliographie

Aalen, O. (1978), 'Nonparametric inference for a family of counting processes', *Ann. Statist.* **6**(4), 701–726.

Aalen, O. O. (1980), A model for nonparametric regression analysis of counting processes, *in* 'Mathematical statistics and probability theory (Proc. Sixth Internat. Conf., Wisl a, 1978)', Vol. 2 of *Lecture Notes in Statist.*, Springer, New York-Berlin, pp. 1–25.

Aalen, O. O. (1989), 'A linear regression model for the analysis of life times', *Statistics in medicine* **8**(8), 907–925.

Akaike, H. (1974), 'A new look at the statistical model identification', *Automatic Control, IEEE Transactions on* **19**(6), 716–723.

Alaya, M. Z., Allart, T., Guilloux, A. & Lemler, S. (2016), 'Time-varying high-dimensional aalen and cox models with change-points', *preprint* .

Alaya, M. Z., Gaiffas, S. & Guilloux, A. (2015), 'Learning the intensity of time events with change-points', *Information Theory, IEEE Transactions on* **61**(9), 5148–5171.

Alaya, M. Z., Gaiffas, S. & Guilloux, A. (2017), 'Binarsity : a penalization for one-hot encoded features', *preprint* .

Allart, T., Levieux, G., Pierfitte, M., Guilloux, A. & Natkin, S. (2016), Design influence on player retention : A method based on time varying survival analysis, *in* 'Proc. IEEE Computational Intelligence and Games Conf., IEEE'.

Allart, T., Levieux, G., Pierfitte, M., Guilloux, A. & Natkin, S. (2017), Difficulty influence on motivation over time in video games using survival analysis, *in* 'Foundation of Digital Games'.

Andersen, P. K., Borgan, O., Gill, R. D. & Keiding, N. (1993), *Statistical models based on counting processes*, Springer Series in Statistics, Springer-Verlag, New York.

Atchade, Y. F., Fort, G. & Moulines, E. (2014), 'On stochastic proximal gradient algorithms', *arXiv preprint arXiv :1402.2365* .

Bach, F., Jenatton, R., Mairal, J. & Obozinski, G. (2011), 'Optimization with Sparsity-Inducing Penalties', *Foundations and Trends in Machine Learning* **4**(1), –.

Baker, G. R., Norton, P. G., Flintoft, V., Blais, R., Brown, A., Cox, J., Etchells, E., Ghali, W. A., Hebert, P., Majumdar, S. R. et al. (2004), 'The canadian adverse events study : the incidence of adverse events among hospital patients in canada', *Canadian medical association journal* **170**(11), 1678–1686.

Bandura, A. (1977), 'Self-efficacy : toward a unifying theory of behavioral change.', *Psychological review* **84**(2), 191.

Bartle, R. (1996), 'Hearts, clubs, diamonds, spades : Players who suit muds', *Journal of MUD research* **1**(1), 19.

Bauckhage, C., Kersting, K., Sifa, R., Thurau, C., Drachen, A. & Canossa, A. (2012), How players lose interest in playing a game : An empirical study based on distributions of total playing times, *in* 'Computational Intelligence and Games (CIG), 2012 IEEE conference', IEEE, pp. 139–146.

Beck, A. & Teboulle, M. (2009), 'A fast iterative shrinkage-thresholding algorithm for linear inverse problems', *SIAM Journal on Imaging Sciences* **2**(1), 183–202.

Bellera, C. A., MacGrogan, G., Debled, M., de Lara, C. T., Brouste, V. & Mathoulin-Pelissier, S. (2010), 'Variables with time-varying effects and the cox model : some statistical concepts illustrated with a prognostic factor study in breast cancer', *BMC medical research methodology* **10**(1), 20.

Borbora, Z., Srivastava, J., Hsu, K.-W. & Williams, D. (2011), Churn prediction in mmorpgs using player motivation theories and an ensemble approach, *in* 'Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on', pp. 157–164.

Bottou, L. (1998), Online algorithms and stochastic approximations, *in* D. Saad, ed., 'Online Learning and Neural Networks', Cambridge University Press, Cambridge, UK. revised, oct 2012.

Bottou, L. (2010), Large-scale machine learning with stochastic gradient descent, *in* 'Proceedings of COMPSTAT'2010', Springer, pp. 177–186.

Bottou, L. (2012), Stochastic gradient descent tricks., *in* G. Montavon, G. B. Orr & K.-R. Muller, eds, 'Neural Networks : Tricks of the Trade (2nd ed.)', Vol. 7700 of *Lecture Notes in Computer Science*, Springer, pp. 421–436.

Bottou, L. & Le Cun, Y. (2005), 'On-line learning for very large data sets', *Applied stochastic models in business and industry* **21**(2), 137–151.

Bouaziz, O. & Guilloux, A. (2015), 'A penalized algorithm for event-specific rate models for recurrent events', *Biostatistics* **16**(2), 281–294.

Bousquet, O. & Bottou, L. (2008), The tradeoffs of large scale learning, *in* 'Advances in neural information processing systems', pp. 161–168.

Bowman, B., Elmqvist, N. & Jankun-Kelly, T. (2012), 'Toward visualization for games : Theory, design space, and patterns', *IEEE transactions on visualization and computer graphics* **18**(11), 1956–1968.

Boyd, S. & Vandenberghe, L. (2004), *Convex optimization*, Cambridge University Press, Cambridge.

Cai, Z. & Sun, Y. (2003), 'Local linear estimation for time-dependent coefficients in cox's regression models', *Scandinavian Journal of Statistics* **30**(1), 93–111.

Cauchy, A. (1847), 'Methode generale pour la resolution des systemes d'equations simultanees', *Comp. Rend. Sci. Paris* **25**(1847), 536–538.

Chen, K.-T., Huang, P. & Lei, C.-L. (2009), 'Effect of network quality on player departure behavior in online games', *Parallel and Distributed Systems, IEEE Transactions on* **20**(5), 593–606.

Cheng, M.-Y., Honda, T., Li, J., Peng, H. et al. (2014), 'Nonparametric independence screening and structure identification for ultra-high dimensional longitudinal data', *The Annals of Statistics* **42**(5), 1819–1849.

Coloma, P. M., Schuemie, M. J., Trifiro, G., Gini, R., Herings, R., Hippisley-Cox, J., Mazzaglia, G., Giaquinto, C., Corrao, G., Pedersen, L. et al. (2011), 'Combining electronic healthcare databases in europe to allow for large-scale drug safety monitoring : the eu-adr project', *Pharmacoepidemiology and drug safety* **20**(1), 1–11.

Constant, T., Levieux, G., Buendia, A. & Natkin, S. (2017), 'From objective to subjective difficulty evaluation in video games'.

Cox, D. R. (1972), 'Regression models and life-tables', *Journal of the Royal Statistical Society. Series B (Methodological)* **34**(2), 187–220.

Cox, D. R. (1975), 'Partial likelihood', *Biometrika* **62**(2), 269–276.

Csikszentmihalyi, M. (1991), *Flow : The psychology of optimal experience*, Vol. 41, Harper-Perennial New York.

Dalalyan, A. S., M.H. & Lederer, J. (2014), On the prediction performance of the lasso, to appear in Bernoulli 1402.1700, arXiv.

Daubechies, I., Defrise, M. & De Mol, C. (2004), 'An iterative thresholding algorithm for linear inverse problems with a sparsity constraint', *Communications on Pure and Applied Mathematics* **57**(11), 1413–1457.

Debeauvais, T., Lopes, C. V., Yee, N. & Ducheneaut, N. (2014), Retention and progression : Seven months in world of warcraft., *in* 'FDG'.

Donoho, D. L. (1995), 'De-noising by soft-thresholding', *IEEE transactions on information theory* **41**(3), 613–627.

Duchi, J., Hazan, E. & Singer, Y. (2011), 'Adaptive subgradient methods for online learning and stochastic optimization', *J. Mach. Learn. Res.* **12**, 2121–2159.

Eddelbuettel, D. & François, R. (2011), 'Rcpp : Seamless R and C++ integration', *Journal of Statistical Software* **40**(8), 1–18.

Fleming, T. R. & Harrington, D. P. (1991), *Counting processes and survival analysis*, Wiley Series in Probability and Mathematical Statistics : Applied Probability and Statistics, John Wiley & Sons, Inc., New York.

Gaiffas, S. & Guilloux, A. (2012), 'High-dimensional additive hazards models and the Lasso', *Electron. J. Stat.* **6**, 522–546.

Galyonkin, S. (2016), 'Steamspy'.
URL: *http ://steamspy.com/ last retreived May 2016*

Grambsch, P. M. & Therneau, T. M. (1994), 'Proportional hazards tests and diagnostics based on weighted residuals', *Biometrika* **81**(3), 515–526.

Guennebaud, G., Jacob, B. et al. (2010), 'Eigen v3', http ://eigen.tuxfamily.org.

Hadiji, F., Sifa, R., Drachen, A., Thurau, C., Kersting, K. & Bauckhage, C. (2014), Predicting player churn in the wild, *in* 'Computational intelligence and games (CIG), 2014 IEEE conference on', IEEE, pp. 1–8.

Harchaoui, Z. & Levy-Leduc, C. (2010), 'Multiple change-point estimation with a total variation penalty', *J. Amer. Statist. Assoc.* **105**(492), 1480–1493.

Harrison, B. E. & Roberts, D. (2014), Analytics-driven dynamic game adaption for player retention in a 2-dimensional adventure game, *in* 'Tenth Artificial Intelligence and Interactive Digital Entertainment Conference'.

Hastie, T., Tibshirani, R., Friedman, J. & Franklin, J. (2001), 'The elements of statistical learning : data mining, inference and prediction', *The Mathematical Intelligencer* .

Hastie, T., Tibshirani, R. & Wainwright, M. (2015), *Statistical learning with sparsity : the lasso and generalizations*, CRC press.

Hayrinen, K., Saranto, K. & Nykanen, P. (2008), 'Definition, structure, content, use and impacts of electronic health records : a review of the research literature', *International journal of medical informatics* **77**(5), 291–304.

He, K., Yang, Y., Li, Y., Zhu, J. & Li, Y. (2016), 'Modeling time-varying effects with large-scale survival data : An efficient quasi-newton approach', *Journal of Computational and Graphical Statistics* (just-accepted).

Hoerl, A. E. & Kennard, R. W. (1970), 'Ridge regression : Biased estimation for nonorthogonal problems', *Technometrics* **12**(1), 55–67.

Honda, T. & Hardle, W. K. (2014), 'Variable selection in cox regression models with varying coefficients', *Journal of Statistical Planning and Inference* **148**, 67–81.

Huang, J., Sun, T., Ying, Z., Yu, Y. & C.-H., Z. (2013), 'Oracle inequalities for the lasso in the cox model', *Ann. Statist.* **41**(3), 1142–1165.

Huffer, F. W. & McKeague, I. W. (1991), 'Weighted least squares estimation for aalen's additive risk model', *Journal of the American Statistical Association* **86**(413), 114–129.

Hunicke, R., LeBlanc, M. & Zubek, R. (2004), Mda : A formal approach to game design and game research, *in* 'Proceedings of the AAAI Workshop on Challenges in Game AI', Vol. 4, p. 1722.

Juul, J. (2013), *The art of failure : An essay on the pain of playing video games*, Mit Press.

Kalantar-Zadeh, K., Kuwae, N., Regidor, D., Kovesdy, C., Kilpatrick, R., Shinaberger, C., McAllister, C., Budoff, M., Salusky, I. & Kopple, J. (2006), 'Survival predictability of time-varying indicators of bone disease in maintenance hemodialysis patients', *Kidney international* **70**(4), 771–780.

Kawale, J., Pal, A. & Srivastava, J. (2009), Churn prediction in mmorpgs : A social influence based approach, *in* 'Computational Science and Engineering, 2009. CSE'09. International Conference on', Vol. 4, IEEE, pp. 423–428.

Kim, J. H., Gunn, D. V., Schuh, E., Phillips, B., Pagulayan, R. J. & Wixon, D. (2008), Tracking real-time user experience (true) : a comprehensive instrumentation solution for complex systems, *in* 'Proceedings of the SIGCHI conference on Human Factors in Computing Systems', ACM, pp. 443–452.

Kingma, D. & Ba, J. (2014), 'Adam : A method for stochastic optimization', *arXiv preprint arXiv :1412.6980* .

Kiwiel, K. C. (2001), 'Convergence and efficiency of subgradient methods for quasiconvex minimization', *Mathematical programming* **90**(1), 1–25.

Koster, R. (2005), *A Theory of Fun for Game Design*, Paraglyph Press, Scottsdale, Arizona.

Lazzaro, N. (2004), Why we play games : Four keys to more emotion without story, *in* 'Game Developers Conference'.

Lemler, S. (2013), 'Oracle inequalities for the lasso in the high-dimensional multiplicative aalen intensity model', *Les Annales de l'Institut Henri Poincare, arXiv preprint* .

Levieux, G. (2011), Mesure de la difficulte des jeux video, PhD thesis, Conservatoire national des arts et metiers-CNAM.

Lewis, P. A. W. & Shedler, G. S. (1979), 'Simulation of nonhomogeneous poisson processes by thinning', *Naval Research Logistics Quarterly* **26**(3), 403–413.

Mader, S. (2015), Le" game design" de jeux therapeutiques : Modeles et methodes pour la conception du gameplay, PhD thesis, Conservatoire national des arts et metiers-CNAM.

Mahlmann, T., Drachen, A., Togelius, J., Canossa, A. & Yannakakis, G. N. (2010), Predicting player behavior in tomb raider : Underworld, *in* 'Computational Intelligence and Games (CIG), 2010 IEEE Symposium on', IEEE, pp. 178–185.

Mallows, C. L. (1973), 'Some comments on c p', *Technometrics* **15**(4), 661–675.

Malone, T. W. (1981), 'Toward a theory of intrinsically motivating instruction*', *Cognitive science* **5**(4), 333–369.

Malone, T. W. (1982), Heuristics for designing enjoyable user interfaces : Lessons from computer games, *in* 'Proceedings of the 1982 conference on Human factors in computing systems', ACM, New York, NY, USA, pp. 63–68.

Martinussen, T. & Scheike, T. H. (1999), 'A semiparametric additive regression model for longitudinal data', *Biometrika* pp. 691–702.

Martinussen, T. & Scheike, T. H. (2007), *Dynamic regression models for survival data*, Springer Science & Business Media.

Martinussen, T. & Scheike, T. H. (2009*a*), 'The additive hazards model with high-dimensional regressors', *Lifetime Data Anal.* **15**(3), 330–342.

Martinussen, T. & Scheike, T. H. (2009*b*), 'Covariate selection for the semiparametric additive risk model', *Scand. J. Statist.* **36**(4), 602–619.

Martinussen, T., Scheike, T. H., Skovgaard, I. M. & Matinerssen, T. (2002), 'Efficient estimation of fixed and time-varying covariate effects in multiplicative intensity models', *Scandinavian Journal of Statistics* **29**(1), pp. 57–74.

McKeague, I. W. (1988), Asymptotic theory for weighted least squares estimators in, *in* 'Statistical Inference from Stochastic Processes : Proceedings of the AMS-IMS-SIAM Joint Summer Research Conference Held August 9-15, 1987, with Support from the

National Science Foundation and the Army Research Office', Vol. 80, American Mathematical Soc., p. 139.

Meyer, P. et al. (1962), 'A decomposition theorem for supermartingales', *Illinois Journal of Mathematics* **6**(2), 193–205.

Murphy, S. A. & Sen, P. K. (1991), 'Time-dependent coefficients in a cox-type regression model', *Stochastic Processes and their Applications* **39**(1), 153–180.

Nesterov, Y. (2007), Gradient methods for minimizing composite objective function, CORE Discussion Papers 2007076, Universite catholique de Louvain, Center for Operations Research and Econometrics (CORE).

Parikh, N. & Boyd, S. P. (2014), 'Proximal algorithms.', *Foundations and Trends in optimization* **1**(3), 127–239.

Perperoglou, A., le Cessie, S. & van Houwelingen, H. C. (2006), 'A fast routine for fitting cox models with time varying effects of the covariates', *Computer methods and programs in biomedicine* **81**(2), 154–161.

R Core Team (2016), *R : A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.

Rigby, S. & Ryan, R. (2007), 'Rethinking carrots : A new method for measuring what players find most rewarding and motivating about your game', *Retrieved December* **3**, 2007.

Rinaldo, A. (2009), 'Properties and refinements of the fused lasso', *Ann. Statist.* **37**(5B), 2922–2952.

Rosasco, L., Villa, S. & Vu, B. C. (2014), 'Convergence of stochastic proximal gradient algorithm', *arXiv preprint arXiv :1403.5074* .

Rosasco, L., Villa, S. & Vu, B. C. (n.d.), 'Learning with stochastic proximal gradient'.

Roux, N. L., Schmidt, M. & Bach, F. R. (2012), A stochastic gradient method with an exponential convergence _rate for finite training sets, *in* 'Advances in Neural Information Processing Systems', pp. 2663–2671.

Ryan, R. M. & Deci, E. L. (2000), 'Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being.', *American psychologist* **55**(1), 68.

Ryan, R. M., Rigby, C. S. & Przybylski, A. (2006), 'The motivational pull of video games : A self-determination theory approach', *Motivation and emotion* **30**(4), 344–360.

Salen, K. & Zimmerman, E. (2003), *Rules of Play : Game Design Fundamentals*, The MIT Press.

Schaul, T., Zhang, S. & LeCun, Y. (2012), 'No more pesky learning rates', *arXiv preprint arXiv :1206.1106* .

Scheike, T., Martinussen, T. & Silver, J. (2016), 'Timereg : timereg package for flexible regression models for survival data', *R package version* pp. 1–9.

Schell, J. (2014), *The Art of Game Design : A book of lenses*, CRC Press.

Schwarz, G. et al. (1978), 'Estimating the dimension of a model', *The annals of statistics* **6**(2), 461–464.

Su, W., Boyd, S. & Candes, E. (2014), A differential equation for modeling nesterov's accelerated gradient method : Theory and insights, *in* 'Advances in Neural Information Processing Systems', pp. 2510–2518.

Sweetser, P. & Wyeth, P. (2005), 'Gameflow : a model for evaluating player enjoyment in games', *Computers in Entertainment (CIE)* **3**(3), 3–3.

Tian, L., Zucker, D. & Wei, L. (2005), 'On the cox model with time-varying regression coefficients', *Journal of the American statistical Association* **100**(469), 172–183.

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

Tibshirani, R. J. (2011), 'The solution path of the generalized lasso'.

Tibshirani, R. J. et al. (2014), 'Adaptive piecewise polynomial estimation via trend filtering', *The Annals of Statistics* **42**(1), 285–323.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. & Knight, K. (2005), 'Sparsity and smoothness via the fused lasso', *J. R. Stat. Soc. Ser. B Stat. Methodol.* **67**(1), 91–108.

Tikhonov, A. (1963), Solution of incorrectly formulated problems and the regularization method, *in* 'Soviet Math. Dokl.', Vol. 5, pp. 1035–1038.

Vapnik, V. (1995), *The Nature of Statistical Learning Theory*, Springer.

Viallon, V. & Latouche, A. (2011), 'Discrimination measures for survival outcomes : connection between the auc and the predictiveness curve', *Biometrical Journal* **53**(2), 217–236.

Viljanen, M., Airola, A., Heikkonen, J. & Pahikkala, T. (2017), 'Playtime measurement with survival analysis', *arXiv preprint arXiv :1701.02359* .

Wallner, G. (2015), Sequential analysis of player behavior, *in* 'Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play', ACM, pp. 349–358.

Wallner, G. & Kriglstein, S. (2013), 'Visualization-based analysis of gameplay data–a review of literature', *Entertainment Computing* **4**(3), 143–155.

Weber, B. G., John, M., Mateas, M. & Jhala, A. (2011), Modeling player retention in madden nfl 11., *in* 'IAAI'.

Weiner, B. (2005), 'Motivation from an attribution perspective and the social psychology of perceived competence', *Handbook of competence and motivation* pp. 73–84.

Winnett, A. & Sasieni, P. (2003), 'Iterated residuals and time-varying covariate effects in cox regression', *Journal of the Royal Statistical Society : Series B (Statistical Methodology)* **65**(2), 473–488.

Xie, H., Devlin, S., Kudenko, D. & Cowling, P. (2015), Predicting player disengagement and first purchase with event-frequency based data representation, *in* 'Computational Intelligence and Games (CIG), 2015 IEEE Conference on', IEEE, pp. 230–237.

Yan, J. & Huang, J. (2012), 'Model selection for cox models with time-varying coefficients', *Biometrics* **68**(2), 419–428.

Yee, N. (2006), 'Motivations for play in online games', *CyberPsychology & behavior* **9**(6), 772–775.

Yuan, M. & Lin, Y. (2006), 'Model selection and estimation in regression with grouped variables', *Journal of the Royal Statistical Society, Series B* **68**, 49–67.

Zeiler, M. D. (2012), 'Adadelta an adaptive learning rate method', *CoRR* .

Zou, H. & Hastie, T. (2005), 'Regularization and variable selection via the elastic net', *Journal of the Royal Statistical Society : Series B (Statistical Methodology)* **67**(2), 301–320.

Zucker, D. M. & Karr, A. F. (1990), 'Nonparametric survival analysis with time-dependent covariate effects : a penalized partial likelihood approach', *The Annals of Statistics* pp. 329–353.

Glossaire

— $CNAM$: Conservatoire National des Arts et Métiers
— $Free - to - play$: Jeu gratuit à l'acquisition
— $MMORPG$: jeu de rôle en ligne massivement multijoueur

le cnam

Thibault ALLART

Apprentissage statistique sur données longitudinales de grande taille et applications au design des jeux vidéo

le cnam

Résumé :
Cette thèse s'intéresse à l'analyse des données longitudinales, potentiellement grandes selon les trois axes suivants : nombre d'individus, fréquence d'observation et nombre de covariables. A partir de ces données, éventuellement censurées, nous considérons comme facteur d'étude le temps d'apparition d'un ou plusieurs évènements. Nous cherchons dans des classes de modèles à coefficients dépendant du temps à estimer l'intensité d'apparition des événements. Or les estimateurs actuels, ne permettent pas de traiter efficacement un grand nombre d'observations et/ou un grand nombre de covariables. Nous proposons un nouvel estimateur défini via la vraisemblance complète de Cox et une pénalisation permettant à la fois la sélection de variables et de forcer, quand c'est possible, les coefficients à être constants. Nous introduisons des algorithmes d'optimisation proximaux, permettant d'estimer les coefficients du modèle de manière efficace. L'implémentation de ces méthodes en C++ et dans le package R coxtv permet d'analyser des jeux de données de taille supérieure à la mémoire vive ; via un streaming du flux de données et des méthodes d'apprentissage en ligne, telles que la descente de gradient stochastique proximale aux pas adaptatifs. Nous illustrons les performances du modèle sur des simulations en nous comparant aux méthodes existantes. Enfin, nous nous intéressons à la problématique du design des jeux vidéo. Nous montrons que l'application directe de ce modèle, sur les grands jeux de données dont dispose l'industrie du jeu vidéo, permet de mettre en évidence des leviers d'amélioration du design des jeux étudiés. Nous nous intéressons d'abord à l'analyse des composantes bas niveau, telles que les choix d'équipement fait par les joueurs au fils du temps et montrons que le modèle permet de quantifier l'effet de chacun de ces éléments de jeu, offrant ainsi aux designers des leviers d'amélioration direct du design. Enfin, nous montrons que le modèle permet de dégager des enseignements plus généraux sur le design tels que l'influence de la difficulté sur la motivation des joueurs.

Mots clés :
Apprentissage statistique, analyse de survie, jeux vidéo, optimisation

Abstract :
This thesis focuses on longitudinal time to event data possibly large along the following tree axes : number of individuals, observation frequency and number of covariates. We introduce a penalised estimator based on Cox complete likelihood with data driven weights. We introduce proximal optimization algorithms to efficiently fit models coefficients. We have implemented thoses methods in C++ and in the R package coxtv to allow everyone to analyse data sets bigger than RAM ; using data streaming and online learning algorithms such that proximal stochastic gradient descent with adaptive learning rates. We illustrate performances on simulations and benchmark with existing models. Finally, we investigate the issue of video game design. We show that using our model on large datasets available in video game industry allows us to bring to light ways of improving the design of studied games. First we have a look at low level covariates, such as equipment choices through time and show that this model allows us to quantify the effect of each game elements, giving to designers ways to improve the game design. Finally, we show that the model can be used to extract more general design recommendations such as dificulty influence on player motivations.

Keywords :
Statistical learning, survival analysis, optimization, video games