Contrôle optimal et apprentissage automatique, applications aux interactions homme-machine

Matthieu Geist

Unité de Formation et de Recherche : IEEA
Ecole Doctorale : SPI
Laboratoire : CRIStAL
Spécialité : Mathématiques Appliquées

Mémoire

présenté pour l'obtention de

l'Habilitation à Diriger des Recherches de l'Université de Lille 1

par **Matthieu Geist**

Contrôle optimal et apprentissage automatique, applications aux interactions homme-machine

Soutenu le 1er février 2016

Membres du jury

Rapporteurs :	Olivier Cappé	Directeur de Recherche, CNRS (France)
	Michèle Sebag	Directrice de Recherche, CNRS (France)
	Björn Schuller	Professeur, Université de Passau (Allemagne) et Imperial College London (Royaume-Uni)
Examinateurs :	Olivier Pietquin	Professeur, Université de Lille 1 (France)
	Philippe Preux	Professeur, Université de Lille 3 (France)

CentraleSupélec
Groupe de Recherche MaLIS (Machine Learning & Interactive Systems)
&
UMI 2958 (GeorgiaTech - CNRS)

À Claire et Paul.

Résumé

Nos travaux de recherche portent sur le *machine learning*, soit l'apprentissage à partir de données (ou apprentissage automatique), et plus particulièrement sur des problèmes de contrôle optimal de systèmes dynamiques, vus sous le prisme de l'apprentissage par renforcement. Dans ce paradigme, la dynamique du système est formalisée de façon probabiliste (par des noyaux de transition markoviens, observables via des données d'interactions entre un agent et le système) et la qualité du contrôle est quantifiée localement par une information de récompense (qui quantifie l'objectif du contrôle, et non la façon d'atteindre cet objectif). Dans ce cadre, l'objectif est d'apprendre la séquence d'actions maximisant une fonction cumulative des récompenses.

Une notion importante en apprentissage par renforcement est celle de fonction de valeur, qui quantifie globalement la qualité d'un contrôleur (en associant à chaque configuration du système le cumul espéré des récompenses par application du contrôleur) et est une quantité intermédiaire nécessaire à de nombreux schémas d'apprentissage du contrôle optimal. Une partie de nos travaux portent sur l'estimation de cette fonction de valeur, que l'estimation soit en-ligne ou hors-ligne, *on-policy* ou *off-policy*, avec un soucis constant d'efficacité en termes d'échantillons (pour apprendre avec aussi peu de données que possible, ces données ayant généralement un coût) et d'adaptabilité (dans le sens d'estimateurs nécessitant aussi peu d'a priori que possible, notamment via des approches non-paramétriques). Plus généralement, nos travaux portent sur les schémas d'apprentissage du contrôle optimal. Beaucoup sont des approximations des schémas de programmation dynamique, conçus pour le cas où le modèle est connu. Nous pensons qu'il est nécessaire de prendre en compte le fait que l'on travaille avec des données plus en amont, et donc en quelque sorte de dépasser le cadre de la programmation dynamique.

Nos travaux portent également sur l'apprentissage par imitation. La problématique est toujours celle de l'apprentissage du contrôle optimal d'un système dynamique à partir de données, mais la nature de ces données change. Plutôt qu'une information de récompense qui quantifie l'objectif du contrôle, on observe des traces d'un comportement expert (supposé optimal), qu'il s'agit donc d'imiter. Ce cadre est très proche de l'apprentissage supervisé (généralisation du contrôleur observé). Toutefois, nous pensons très important d'également prendre en compte l'information donnée par la dynamique du système (ce que ne fait pas une approche supervisée classique) : ce n'est pas tant l'action choisie que son effet qui est important. Ce problème d'imitation est donc traité par le prisme de l'apprentissage par renforcement inverse (apprendre la récompense à partir de traces d'un comportement optimal) et de nouvelles méthodes d'apprentissage supervisé tenant compte de l'information de dynamique. A la croisée de ces deux domaines (apprentissage par renforcement et apprentissage par imitation) se pose le problème de l'apprentissage du contrôle optimal quand les deux sources d'information (récompense et comportement expert) sont disponibles. Gé-

néralement, comme pour l'apprentissage par renforcement, les algorithmes proposés sont motivés par une exploitation maximale des données disponibles (généralement supposées imposées, pas d'accès à un modèle génératif ou au système réel) avec aussi peu d'a priori que possible (par le biais de méthodes non-paramétriques).

D'un point de vue applicatif, nos travaux portent sur l'utilisation de l'apprentissage par renforcement et de l'apprentissage par imitation pour des problèmes d'interactions entre humain et machine (systèmes de dialogue parlé, tutorat intelligent et rire dans les interactions homme-machine). Ce sont généralement des problèmes de contrôle optimal de systèmes dynamiques, mais pour lesquels c'est l'humain avec qui interagit la machine qui définit la dynamique du système. Dans ce cadre, les donnés d'interactions sont généralement coûteuses, en petit nombre, imposées. De plus, il est généralement difficile d'avoir un modèle génératif (simuler l'humain) et on peut difficilement interagir avec le système réel à volonté (comme l'humain peut vite se lasser, il est important d'interagir via un contrôleur correct).

Pour résumer, nos travaux portent généralement sur la conception d'algorithmes d'apprentissage à partir de données (et plus particulièrement de l'apprentissage du contrôle optimal d'un système dynamique, par le biais d'une récompense et/ou d'exemples de comportements experts), en exploitant aussi bien que possible les données disponibles, idéalement en offrant également des garanties théoriques et en leur trouvant des champs d'application utiles (ou inversement, motiver les développements plus théoriques par les problèmes applicatifs).

Table des matières

Chapitre 1

Introduction

Ce document présente les recherches que nous avons menées depuis nos travaux doctoraux [63], soutenus fin 2009, qui ont servi de socle aux travaux ultérieurs et que nous commencerons par résumer rapidement dans ce chapitre, après avoir présenté brièvement nos motivations globales. Nous donnerons ensuite un résumé des travaux effectués après la thèse, en les structurant et en les plaçant en perspective par rapport aux encadrements doctoraux et aux projets collaboratifs auxquels nous avons participé.

1.1 Motivations

Nos travaux de recherche portent principalement sur la problématique du contrôle optimal, vu sous le prisme de l'apprentissage automatique. Il s'agit généralement de concevoir des algorithmes permettant à un agent d'apprendre à contrôler de façon optimale un système dynamique, à partir d'exemples d'interactions avec le système, que ces exemples soient imposés a priori ou que l'agent ait une certaine latitude pour les choisir (notamment dans le cadre d'un apprentissage en ligne, mais aussi en envisageant l'accès à un simulateur).

Une façon de caractériser l'optimalité du contrôle est de fournir à l'agent des récompenses numériques après chaque interaction, qui sont des informations locales de la qualité des décisions prises, l'objectif étant de maximiser une fonction cumulative de ces récompenses (appelée fonction de valeur). Cela est principalement couvert par la programmation dynamique lorsque le système est connu, par l'apprentissage par renforcement plus généralement, qui est notre domaine de recherche initial.

D'un point de vue applicatif, nous nous sommes particulièrement intéressé à des problèmes d'interactions homme-machine. Nous les décrirons par la suite, mais il s'agit de façon abstraite pour l'agent d'un problème de contrôle d'un système où c'est l'humain avec qui interagit la machine qui définit la dynamique. Cela rend la collecte et l'annotation de données d'interactions difficiles, simuler un tel système n'est pas évident, et il faut être prudent si l'on envisage un apprentissage en ligne (il faut apprendre rapidement, sans ennuyer l'humain, ce qui pose notamment des problèmes d'efficacité en termes d'échantillons et de dilemme entre exploration et exploitation).

Ces problèmes applicatifs ont influencé nos contributions plus fondamentales ou algorithmiques. Beaucoup d'algorithmes d'apprentissage par renforcement se reposent sur la programmation dynamique, et impliquent donc souvent d'estimer une fonction de valeur. C'est un problème difficile (plus difficile que la régression en apprentissage supervisé, qui en est un cas particulier). Nous nous sommes attaché à proposer de nouveaux estimateurs

de la valeur, avec notamment comme objectifs la possibilité de passer à l'échelle (traiter de grands problèmes), la possibilité d'introduire aussi peu d'a priori que possible (grâce notamment à des approches non-paramétriques) ou encore l'efficacité en termes d'échantillons (apprendre aussi bien que possible avec aussi peu de données que possible), cela en nous interrogeant sur les garanties théoriques associées, quand cela est possible, et en nous efforçant de prendre tout le recul nécessaire par rapport à l'état de l'art. Plus généralement (et plus récemment), nous nous sommes également intéressé aux schémas d'apprentissage du contrôle optimal, en nous efforçant de dépasser le cadre de la programmation dynamique approchée (qui a ses limites).

L'agent n'est pas forcément informé de la qualité du contrôle via des récompenses. Il peut observer plus directement des démonstrations faites par un expert, et apprendre à l'imiter, ce qui est le domaine de l'apprentissage par imitation. Cette imitation peut se faire soit directement (en apprenant un contrôleur), soit moins directement, en estimant une récompense qu'optimiserait l'expert. On parle alors d'apprentissage par renforcement inverse. Nous nous sommes intéressé à ce domaine, avec comme motivation de supprimer un écueil de l'état de l'art d'alors, à savoir devoir résoudre le problème direct (de renforcement) de multiples fois pour estimer une récompense (la motivation fondamentale étant d'exploiter au mieux les données). En effet, cela nous semblait une condition nécessaire pour envisager d'utiliser l'apprentissage par renforcement inverse dans des problèmes d'interactions homme-machine, en raison des contraintes sur les données que pose généralement ce domaine applicatif. Nous nous sommes également intéressé à l'apprentissage direct d'un contrôleur. Une solution simple est de voir cela comme un problème d'apprentissage supervisé. Toutefois, nous pensons que dans le cadre du contrôle, ce n'est pas tant la décision prise que l'effet de cette décision qui importe. Nous avons donc cherché à introduire l'information de dynamique dans des algorithmes d'apprentissage supervisé classiques. Enfin, l'agent peut également avoir accès aux deux types d'information, récompenses et démonstrations. Ce cadre est peu étudié, mais il peut avoir un grand intérêt pratique. Nous avons donc étudié comment tenir compte des démonstrations dans les algorithmes de renforcement.

Pour résumer, nous nous intéressons à des problèmes de contrôle optimal de systèmes dynamiques, l'agent ayant accès à des récompenses et/ou des démonstrations. Ce contrôle doit être appris à partir de données, nous l'étudions donc sous le prisme de l'apprentissage automatique. De part la nature des problèmes applicatifs qui nous ont intéressé jusque-là, à savoir les interactions homme-machine, nous avons effectué nos recherches en nous imposant généralement un cadre contraint sur les données disponibles (typiquement, nous ne nous autorisons pas l'accès à un simulateur), tout en gardant pour objectif de proposer des méthodes qui passent à l'échelle, nécessitent aussi peu d'a priori (ou d'expertise du domaine applicatif) que possible, tout en proposant des garanties théoriques lorsque c'est possible.

1.2 Résumé des travaux doctoraux

Notre thèse, financée par ArcelorMittal dans le cadre d'un dispositif CIFRE [1] avec pour partenaires académiques Supélec et INRIA, portait de façon générale sur la problématique de l'optimisation de chaînes de production dans l'industrie sidérurgique. Nous avons choisi de l'aborder par l'apprentissage par renforcement, qui peut être vu comme la réponse du domaine de l'apprentissage machine au problème du contrôle optimal d'un système dynamique.

1. Convention Industrielle de Formation par la Recherche.

Dans ce paradigme, un agent apprend à contrôler son environnement, en interagissant avec ce dernier ou à partir d'exemples d'interactions préétablis. Il reçoit régulièrement une information locale de la qualité du contrôle effectué sous la forme d'une récompense numérique, son objectif étant de maximiser une fonction cumulante de ces récompenses sur le long terme, appelée fonction de valeur.

L'estimation de cette fonction de valeur est un sous-problème très commun en apprentissage par renforcement, c'est sur ce point plus particulier qu'ont porté nos travaux doctoraux. Notre contribution principale est le cadre des différences temporelles de Kalman (KTD, pour *Kalman Temporal Differences*) [70]. Le postulat de départ est assez simple. Il consiste à exprimer le problème d'estimation d'une fonction de valeur paramétrée comme un problème de filtrage, c'est-à-dire d'inférence des variables cachées que sont les paramètres (pour lesquels une dynamique peut être imposée) en fonction de l'historique des observations que sont les récompenses obtenues. Fondamentalement, il s'agit de la minimisation d'un résidu de Bellman, des cas particuliers de KTD donnent des algorithmes connus ou évidents [2]. Toutefois, le cadre de filtrage apporte un certain nombre d'avantages.

Tout d'abord, des extensions du filtre de Kalman permettent de prendre en compte des modèles d'observation non-linéaires (ici, ce modèle d'observation est le lien entre la récompense et les paramètres de la fonction de valeur, soit une équation de Bellman). Nous avons tout particulièrement étudié la transformation non-parfumée, ou linéarisation statistique [74], dont le principe est le suivant : pour la transformation non-linéaire d'une variable aléatoire, si l'approche standard consiste à approcher la fonction (typiquement via un développement de Taylor), il peut être plus intéressant d'approcher la variable aléatoire même (avec une approximation particulaire ici). Dans le cadre de KTD, cela permet de considérer naturellement des paramétrisations non-linéaires de la fonction de valeur (par exemple via un réseau de neurones) ainsi que l'opérateur d'optimalité de Bellman (autant que nous le sachions, KTD est l'unique approche quadratique pour la minimisation du résidu de Bellman pour l'opérateur d'optimalité).

Par ailleurs, le cadre du filtrage fournit naturellement une information d'incertitude sur les paramètres estimés, ce qui s'avère utile pour développer des heuristiques traitant du dilemme entre exploration et exploitation [76]. Le fait que les paramètres puissent suivre une dynamique (typiquement une marche aléatoire) permet d'obtenir des estimateurs adaptatifs de la valeur (dans le cadre d'un environnement éventuellement non-stationnaire, mais essentiellement dans le cadre du contrôle) [88]. Imposer un bruit coloré particulier permet d'introduire un estimateur prenant en compte le concept de traces d'éligibilités [68], ce qui mène, dans le cas limite, à un estimateur non-biaisé (ce problème de biais étant commun à toutes les approches résiduelles). Nous avons également considéré l'estimateur générique qu'est KTD dans le cadre du contrôle (en ligne), que ce soit dans un schéma de type itération de la politique, itération de la valeur ou encore acteur-critique [72]. Enfin, nous avons appliqué ce cadre à un problème (simulé) de gestion des flux de gaz dans un complexe sidérurgique [63].

Pour résumer, nos travaux doctoraux ont principalement porté sur l'estimation de fonctions de valeur, via une classe d'estimateurs inspirée du filtrage en traitement du signal, estimateurs qui présentent certains avantages par rapport à l'état de l'art.

2. Par analogie, on peut penser dans le cadre supervisé aux moindres carrés récursifs qui sont un cas particulier du filtre de Kalman.

1.3 Synthèse des activités de recherche

Peu après la soutenance de notre thèse, nous avons eu la chance d'obtenir, début 2010, un poste d'enseignant-chercheur permanent au sein de Supélec. Les activités de recherche résumées dans cette section s'inscrivent donc dans ce cadre. Pour cette synthèse (ainsi que pour le développement qui en est fait dans le reste du manuscrit), nous choisissons une présentation thématique plutôt que chronologique, en faisant le lien aux encadrements doctoraux et projets collaboratifs associés.

1.3.1 Apprentissage par renforcement

Cette thématique de recherche, que nous développerons dans le **chapitre 2**, est la suite et l'élargissement des travaux doctoraux. Elle est essentiellement le fruit de collaborations avec Olivier Pietquin (co-directeur de notre propre thèse puis collaborateur proche) puis avec Bruno Scherrer (CR INRIA). Elle est peu liée à des projets collaboratifs ou des encadrements doctoraux (sans que l'intersection soit nulle pour autant). Cette thématique peut elle-même se subdiviser en deux sujets généraux : estimation de la valeur et schémas d'apprentissage (du contrôle optimal).

Nous avons expliqué section 1.2 avoir travaillé principalement sur l'estimation de fonction de valeur lors de nos travaux doctoraux, problème que nous avons continué à étudier. Nous avons notamment étendu l'idée de la linéarisation statistique (appliquée à la minimisation de résidu de Bellman pour KTD) au principe de point fixe projeté (sous-jacent à l'algorithme LSTD, *Least-Squares Temporal Differences*) [73]. C'est, autant que nous le sachions, l'unique extension de LSTD qui permette de prendre en compte des paramétrisations non-linéaires de la valeur ainsi que l'opérateur d'optimalité de Bellman. Fort de cette expérience dans l'estimation de la valeur, nous avons proposé une synthèse originale et unificatrice de tous les estimateurs paramétriques (linéaires ou non) de la valeur [3] [77, 78]. Grâce à ce point de vue unificateur, en nous restreignant toutefois aux paramétrisations linéaires, nous avons pu étendre génériquement toutes ces approches au concept de traces d'éligibilité dans le cadre *off-policy* [93], ce qui a permis de retrouver les algorithmes existants, mais aussi d'en dériver de nouveaux (très mécaniquement et généralement). Toutefois, l'inconvénient des approches paramétriques est qu'il faut choisir la représentation de la fonction de valeur a priori, ce qui est nécessairement très problème-dépendant. Nous avons donc également travaillé sur l'estimation non-paramétrique de la fonction de valeur, en nous inspirant notamment du domaine de la régularisation ℓ_1 en apprentissage supervisé. Nous avons ainsi proposé des approches s'inspirant de Lasso (*Least Absolute Shrinkage and Selection Operator*) [91] ou généralisant le sélecteur de Dantzig [94].

Si l'estimation de la valeur est un problème important, ce n'est qu'une partie du problème plus général de la détermination de la politique de contrôle optimale, sujet sur lequel nous avons également travaillé. Les approches classiques sont des variations approchées et possiblement asynchrones d'algorithmes de programmation dynamique (itération de la politique et itération de la valeur, voire programmation linéaire). En programmation dynamique standard, l'algorithme d'itération de la politique modifiée offre un continuum entre itération de la valeur et de la politique (qui en sont des cas particuliers). Nous avons participé à l'élaboration de sa version approchée, AMPI [186, 191] (*Approximate Modified Policy Iteration*). Ce cadre, qui offre des garanties théoriques (généralisant celles des versions approchées de

3. Il est à noter que l'estimation de la valeur est, en général, bien plus complexe que le problème de régression en apprentissage supervisé.

l'itération de la valeur et de la politique), fournit un canevas général des schémas algorithmiques de programmation dynamique approchée, généralisant une bonne partie de l'état de l'art (itérations approchées de la politique ou de la valeur standards, mais aussi les approches utilisant un classifieur pour évaluer la politique gloutonne par rapport à une valeur, entre autre). Nous avons également étudié une alternative à la programmation dynamique, à savoir la recherche directe dans un espace de politiques, qui consiste à maximiser la valeur moyenne (pour une distribution sur les états donnée), la variable d'optimisation étant la politique même [189]. Nous nous sommes particulièrement intéressé aux garanties offertes par ce type d'approche ainsi que ses liens à la programmation dynamique usuelle (en formalisant le fait qu'il s'agit d'une forme conservative d'itération de la politique). Enfin, nous avons également étudié des alternatives moins orthodoxe à la programmation dynamique, en exprimant la minimisation du résidu de Bellman pour l'opérateur d'optimalité comme un problème d'optimisation d'une différence de fonctions convexes [171].

1.3.2 Apprentissage par imitation

En apprentissage par renforcement, le problème est, étant donnée une fonction de récompense connue ou observée, de déterminer la politique optimale correspondante. En apprentissage par renforcement inverse, le but est d'estimer la fonction de récompense optimisée (mais inconnue) par un expert observé. Cela s'inscrit dans le cadre plus général de l'apprentissage par imitation, où un agent apprenant doit imiter le comportement d'un expert (ce qui est fait en optimisant la récompense estimée dans le cadre du renforcement inverse).

Cette thématique de recherche, que nous développerons dans le **chapitre 3**, est liée au projet Ilhaire[4] et a fait l'objet des thèses d'Edouard Klein (2010-2013) et de Bilal Piot (2011-2014), que nous avons co-dirigées :

- Edouard Klein. *Contributions à l'apprentissage par renforcement inverse.* Thèse de Doctorat en Informatique, Université de Lorraine, 2013 ;
- Bilal Piot. *Apprentissage hors-ligne avec Démonstrations Expertes.* Thèse de Doctorat en Informatique, Université de Lorraine, 2014.

L'apprentissage par renforcement inverse est, pour un certain nombre de raisons, un problème mal posé (tout du moins en 2010, quand nous avons commencé à l'étudier). Pour lever les ambiguïtés possibles, le principe général des approches de l'état de l'art était d'estimer une récompense telle que les trajectoires de la politique optimale pour cette récompense soient proches des trajectoires observées de l'expert. Un inconvénient majeur des algorithmes résultants est leur caractère itératif : ils nécessitent de résoudre le problème de renforcement direct comme étape intermédiaire (pour un certain nombre de récompenses arbitraires). En nous basant sur la similarité fonctionnelle entre les notions de politique gloutonne (respectivement de fonction de valeur sur les couples état-action) en renforcement et de règle de décision (respectivement de fonction de score) en classification, nous avons pu proposer des algorithmes originaux [117, 119] qui ne nécessitent pas de résoudre le problème direct et se réduisent pour l'essentiel à de l'apprentissage supervisé d'un point de vue algorithmique (ce qui permet de bénéficier de la riche littérature sur le sujet), tout en

4. Ilhaire (*Incorporating laughter into human-avatar interactions : research and evaluation*) était un projet européen ICT FET Open (2012-2014) regroupant l'université de Mons, le CNRS, l'Universität Augsburg, l'Universita degli Studi di Genova, l'University College London, la Queen's University Belfast, l'Universität Zürich, Supélec, La Cantoche Production ainsi que l'université de Lille 1. Il avait pour objet l'étude du rôle du rire dans les interactions homme-machine et le développement de nouveaux paradigmes pour une interaction plus naturelle, y compris avec des avatars anthropomorphiques.

offrant des garanties théoriques et de bons résultats empiriques. Ce cadre a plus tard été abstrait par la théorie des politiques d'ensembles [165], qui montre qu'il y a en fait bijection entre les solutions du renforcement inverse et celles de la classification à base de fonctions de score.

Lorsque le renforcement inverse est utilisé dans le cadre de l'apprentissage par imitation (c'est-à-dire dans l'objectif d'imiter l'agent expert), une alternative simple consiste à utiliser de l'apprentissage supervisé (de la classification dans le cas d'un espace d'action discret). La classification ne prendra pas en compte la dynamique du système, mais elle est plus étudiée et mieux comprise [5]. Dès lors, on peut se demander quelle approche privilégier (tant du point de vue théorique qu'empirique), question que nous avons étudiée [168]. La réponse (simple mais incomplète) est que cela dépend de l'importance de la dynamique et des données disponibles (et plus généralement de la difficulté à résoudre le problème direct). Dans ce cadre, l'idéal serait d'avoir un algorithme qui présente la souplesse de la classification tout en tenant compte de la dynamique du système. C'est ce que nous avons proposé via l'introduction d'un terme de régularisation rendant compte de la dynamique du système [169] (combinable avec tout risque basé sur une perte de classification à base de score, le problème d'optimisation résultant n'étant pas forcément simple).

Une problématique connexe est celle de l'apprentissage par renforcement avec démonstrations expertes. Dans certaines situations, il est possible que l'on ait accès à la fois à la fonction de récompense et à des démonstrations expertes (de la politique optimale pour cette récompense). Dans ce cas, il faut tirer parti des deux sources d'information. Nous avons proposé une des premières solutions [168] à ce problème encore peu étudié.

1.3.3 Applications aux interactions homme-machine

Nous avons travaillé à l'application (d'une partie) des méthodes mentionnées en sections 1.3.1 et 1.3.2 à des problèmes d'interaction homme-machine, plus particulièrement à des systèmes de dialogue parlé, de tutorat intelligent et à l'introduction du rire dans une interaction humain-avatar. Cette thématique de recherche, que nous développerons dans le **chapitre 4**, est liée aux projets Classic [6], Allegro [7] et Ilhaire (mentionné en section 1.3.2) et à fait l'objet des thèses de Senthilkumar Chandramohan (2009-2012) et de Lucie Daubigney (2010-2013), que nous avons co-encadrées, ainsi que de celle de Bilal Piot, déjà mentionnée section 1.3.2 :

- Senthilkumar Chandramohan. *Revisiting User Simulation in Dialogue Systems: Do we still need them? Will imitation play the role of simulation?* Thèse de Doctorat en Informatique, Université d'Avignon, 2012 ;
- Lucie Daubigney. *Gestion de l'incertitude pour l'optmisation de systèmes interactifs.* Thèse de Doctorat en Informatique, Université de Lorraine, 2013 ;

5. Même dans le cas des approches sus-mentionnées, qui estiment directement une récompense en se basant essentiellement sur de l'apprentissage supervisé, il reste à optimiser la récompense apprise pour imiter l'agent expert, ce qui est un problème difficile en général.

6. Classic était un projet européen ICT STREP (2009-2011) regroupant l'Heriott Watt University, l'Edinburgh University, Cambridge, l'université de Genève, France Télécom et Supélec. Il avait pour objet l'optimisation de bout en bout d'un système de dialogue homme-machine à partir de données et de méthodes statistiques, ainsi que la gestion de la propagation des incertitudes introduites par chacun des modules.

7. Allegro était un projet européen FEDER - INTERREG (2010-2012) regroupant l'Université de la Sarre, INRIA Nancy Grand-Est, Supélec et le DFKI. Il avait pour objet le développement de technologies avancées d'enseignement par le web (*e-learning*) ayant pour but l'accompagnement de l'accroissement de la multilingualité dans la Grande Région.

– Bilal Piot. *Apprentissage hors-ligne avec Démonstrations Expertes*. Thèse de Doctorat en Informatique, Université de Lorraine, 2014.

Les systèmes de dialogue parlé sont un medium d'interaction entre un humain et une machine par le biais de la parole, avec pour but (dans notre cas) l'accomplissement d'une tâche spécifique (demande d'informations touristiques ou réservation d'un billet par exemple). Le problème est, pour la machine, de conduire le dialogue en fonction des échanges (bruités) passés de façon à accomplir la tâche voulue. Les systèmes de tutorat intelligent, tels que nous les avons étudiés, consistent à personnaliser l'enseignement, de façon automatique et individualisée, la décision pour la machine consistant en le choix de la succession de leçons et évaluations en fonction des réactions de l'étudiant. Enfin, l'introduction du rire dans une interaction humain-avatar consiste à décider, pour l'avatar, s'il doit rire ou non (et quel type de rire) en fonction de mesures multi-modales de l'activité du ou des participants humains (dans le cadre d'une interaction ludique pour les problèmes étudiés). Le point commun des ces trois grands problèmes est qu'ils peuvent se formaliser comme des problèmes de contrôle optimal d'un système dynamique, dans lequel c'est l'humain dans la boucle qui définit la dynamique du système.

Jusque là, dans les systèmes de dialogue parlé, l'utilisateur était généralement simulé (par exemple grâce à un réseau bayésien dynamique) afin de permettre l'apprentissage d'une politique de contrôle au niveau du gestionnaire de dialogue. Nous avons proposé d'utiliser des méthodes d'apprentissage par renforcement *batch* et *off-policy* pour apprendre directement une politique d'interaction optimale à partir des données d'interactions récoltées préalablement (par exemple par un *wizard of Oz*) [163]. Nous avons également proposé d'imiter l'utilisateur en utilisant l'apprentissage par renforcement inverse [21] (possiblement après apprentissage non-supervisé de différents comportements type [23]), ce qui permet d'étudier la co-adaptation entre utilisateur et gestionnaire de dialogue [24]. Nous avons également étudié l'apprentissage en ligne du contrôle optimal (idéalement en interagissant avec de vrais utilisateurs humains, bien que nos résultats soient simulés), ce qui pose notamment des problèmes d'apprentissage efficace (en terme de données), de gestion du dilemme entre exploration et exploitation et de non-stationnarité [38]. Le problème de tutorat intelligent peut également être vu comme un problème d'apprentissage par renforcement [160]. Toutefois, il pose en plus un problème d'observabilité partielle, dans la mesure où l'état, à savoir la connaissance (le savoir) de l'étudiant essentiellement, n'est que partiellement observé. Nous avons proposé d'aborder cela en combinant des méthodes d'apprentissage par renforcement avec des réseaux neuronaux récurrents [43]. Pour ces deux grands domaines, la définition d'une fonction de récompense est assez naturelle, ce qui n'est plus le cas lorsqu'il s'agit de faire rire une machine. C'est ce constat qui a motivé nos travaux amont esquissés en section 1.3.2 et que nous avons appliqués à l'introduction du rire dans des interactions homme-machine [152, 174].

1.3.4 Perspectives

Notre projet de recherche pour la suite s'inscrit naturellement dans le cadre de la problématique du contrôle optimal d'un système dynamique, vu sous le prisme de l'apprentissage automatique, que l'agent ait accès à des récompenses et/ou à des démonstrations. De façon générale, nous souhaitons faciliter l'applicabilité de ces paradigmes à des problèmes réels. Nous pensons que pour cela il faut dépasser les cadres usuels de la programmation dynamique approchée en apprentissage par renforcement et d'appariement de trajectoires en apprentissage par renforcement inverse. Cela peut passer par un développement des in-

terconnexions entre apprentissage par renforcement (direct ou inverse) et les autres types d'apprentissage (notamment supervisé). Pour atteindre cette objectif, un axe de recherche qui nous semble également prioritaire est l'étude des méthodes de sélection ou d'évaluation de modèle en renforcement. En effet, avant d'appliquer un contrôleur appris à un système réel, il est plus que souhaitable d'évaluer a priori sa qualité, chose encore très difficile aujourd'hui. Nous développons ce projet scientifique au **chapitre 5**.

Chapitre 2

Apprentissage par renforcement

Cette thématique de recherche est la suite et l'élargissement de nos travaux doctoraux. Elle est principalement le fruit de collaborations avec Olivier Pietquin puis Bruno Scherrer. Ces travaux impliquent bien sûr d'autres co-auteurs, comme Bilal Piot ou Alessandro Lazaric et Mohammad Ghavamzadeh par exemple, voir également les publications associées.

Nous commencerons par présenter section 2.1 quelques généralités concernant l'apprentissage par renforcement, afin notamment de poser les notations. Pour plus de détails sur le domaine, le lecteur pourra se référer à quelques ouvrages de référence [176, 13, 14, 197, 203, 193]. Nous présenterons ensuite section 2.2 nos contributions à l'estimation de la fonction de valeur, tant paramétrique que non-paramétrique. La section 2.3 est dévolue aux contributions concernant plus généralement les schémas d'apprentissage du contrôle optimal (programmation dynamique approchée et alternatives).

2.1 Généralités

Le cadre formel considéré de l'apprentissage par renforcement est celui des processus décisionnels de Markov (PDM). Un PDM $\mathcal{M}$ est un tuple $\{\mathcal{S}, \mathcal{A}, P, \mathcal{R}, \gamma\}$ où $\mathcal{S}$ est l'espace d'états, supposé ici fini [1], $\mathcal{A}$ l'espace d'actions également supposé fini [2], $P \in \Delta_{\mathcal{S}}^{\mathcal{S}\times\mathcal{A}}$ un noyau de transition markovien [3], $\mathcal{R} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ une fonction de récompense et $\gamma \in [0, 1[$ un facteur d'actualisation. Une politique stochastique $\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}$ associe à chaque état s une loi sur les actions, $\pi(.|s)$. Nous considérerons fréquemment le cas particulier des politiques déterministes, où $\pi \in \mathcal{A}^{\mathcal{S}}$ associe à chaque état s l'action $\pi(s)$.

2.1.1 Fonction(s) de valeur

La qualité d'une politique de contrôle π est quantifiée par la fonction de valeur $v_\pi \in \mathbb{R}^{\mathcal{S}}$, qui associe à chaque état le cumul espéré et pondéré des récompenses obtenues en suivant

1. Cette hypothèse est essentiellement faite pour la clarté d'exposition.

2. Cette hypothèse est plus critique, dans la mesure où l'on est souvent amené à déterminer les actions qui maximisent certaines quantités.

3. Nous notons de façon générale $\Delta_{\mathcal{X}}$ l'ensemble des mesures de probabilité sur $\mathcal{X}$ muni de sa tribu discrète et $\mathcal{Y}^{\mathcal{X}}$ l'ensemble des applications de $\mathcal{X}$ dans $\mathcal{Y}$. Dans notre cas $P(s'|s, a)$ désigne la probabilité de transiter dans l'état s' sachant que l'action a a été appliquée dans l'état s.

la politique π à partir de cet état :

$$v_\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(S_t, A_t) | S_0 = s, A_t \sim \pi(.|S_t), S_{t+1} \sim P(.|S_t, A_t)\right].$$

Notons respectivement

$$\mathcal{R}_\pi = \left(\sum_{a\in\mathcal{A}} \pi(a|s)\mathcal{R}(s,a)\right)_{s\in\mathcal{S}} \text{ et } P_\pi = \left(\sum_{a\in\mathcal{A}} \pi(a|s)P(s'|s,a)\right)_{s,s'\in\mathcal{S}}$$

la récompense et le noyau de transition induits par la politique π, la fonction de valeur v_π est l'unique point fixe de l'opérateur d'évaluation de Bellman :

$$v_\pi = T_\pi v_\pi \text{ où } \forall v \in \mathbb{R}^{\mathcal{S}},\ T_\pi v = \mathcal{R}_\pi + \gamma P_\pi v.$$

Il existe une politique π_*, optimale dans le sens où, pour tout état s et toute politique π, on a $v_{\pi_*}(s) \geq v_\pi(s)$. Sa fonction de valeur $v_* = v_{\pi_*}$ est l'unique point fixe de l'opérateur d'optimalité de Bellman T :

$$v_* = Tv_* \text{ où } \forall v \in \mathbb{R}^{\mathcal{S}},\ Tv = \max_{\pi\in\mathcal{A}^{\mathcal{S}}} T_\pi v,$$

l'opérateur max s'entendant ici par composantes. Pour une fonction $v \in \mathbb{R}^{\mathcal{S}}$, une politique π sera dite gloutonne par rapport à v si $T_\pi v = Tv$. Nous noterons $\mathcal{G}(v)$ l'ensemble des politiques gloutonnes par rapport à v. Nous avons bien sûr que $\pi_* \in \mathcal{G}(v_*)$, et nous pouvons caractériser l'optimalité d'une politique par la propriété de point fixe suivante (évidente, mais moins commune) :

$$\pi \in \mathcal{G}(v_\pi).$$

Écrivons plus explicitement une politique gloutonne :

$$\pi \in \mathcal{G}(v) \Leftrightarrow \forall s \in \mathcal{S},\ \pi(s) \in \operatorname*{argmax}_{a\in\mathcal{A}} \left(\mathcal{R}(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a)v(s')\right).$$

Connaissant v, déterminer une politique gloutonne nécessite donc de connaître le modèle, c'est-à-dire le noyau de transition ainsi que la récompense (qui ne seront généralement plus disponibles dans le cas approché). Cela motive l'introduction de la fonction de valeur sur les couples état-action (où fonction de qualité, ou encore Q-fonction), qui associe à chaque couple le cumul espéré et pondéré de récompenses obtenues en appliquant l'action a dans l'état s, puis en suivant la politique π par la suite :

$$Q_\pi(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(S_t, A_t) | S_0 = s, A_0 = a, S_{t+1} \sim P(.|S_t, A_t), A_{t+1} \sim \pi(.|S_{t+1})\right].$$

Cette fonction est également unique point fixe d'un opérateur de Bellman [4] :

$$Q_\pi = T_\pi Q_\pi \text{ avec } [T_\pi Q](s,a) = \mathcal{R}(s,a) + \gamma\mathbb{E}[Q(S',A')|S' \sim P(.|s,a), A' \sim \pi(.|S')].$$

4. Avec un léger abus de notation, nous le notons également T_π, la fonction sur lequel il agit levant l'ambiguïté.

Il existe également une unique Q-fonction optimale associée à toute politique optimale, notée Q_* et solution du problème de point fixe suivant :

$$Q_* = TQ_* \text{ avec } [TQ](s,a) = \mathcal{R}(s,a) + \gamma \mathbb{E}[\max_{a' \in \mathcal{A}} Q(S', a') | S' \sim P(.|s,a)]. \tag{2.1}$$

Nous avons toujours la notion de politique gloutonne, mais qui ne fait plus intervenir le modèle (ce qui est commode lorsque ce dernier n'est pas connu) :

$$\pi \in \mathcal{G}(Q) \Leftrightarrow \forall s \in \mathcal{S},\ \pi(s) \in \operatorname*{argmax}_{a \in \mathcal{A}} Q(s,a). \tag{2.2}$$

2.1.2 Programmation dynamique

Par programmation dynamique nous entendons l'ensemble des méthodes qui permettent de déterminer une politique optimale d'un PDM donné (connaissant le modèle), à savoir programmation linéaire, itération de la politique et itération de la valeur. Nous considérons tous ces algorithmes avec la fonction de valeur, mais ils s'écrivent de la même manière avec la Q-fonction.

La valeur optimale peut-être caractérisée comme point fixe de l'opérateur d'optimalité de Bellman, qui est non-linéaire. Ce point fixe est la solution du programme linéaire suivant ($\mathbf{1}$ est un vecteur dont toutes les composantes sont égales à 1) :

$$\min_{v \in \mathbb{R}^{\mathcal{S}}} \mathbf{1}^\top v \text{ tel que } Tv \geq v.$$

Cette approche n'est pas majoritaire dans le cas approché, nous la discuterons peu [5].

L'opérateur d'optimalité de Bellman est une γ-contraction (en norme ℓ_∞), le théorème du point fixe de Banach donne donc une méthode de calcul de v_*, appelée algorithme d'itération sur les valeurs :

$$v_{k+1} = Tv_k.$$

Enfin, l'algorithme d'itération sur les politiques consiste à répéter des phases d'évaluation de la politique (trouver le point fixe de T_π, qui est un opérateur affine) et d'amélioration de la politique (politique gloutonne par rapport à la valeur calculée) :

$$\begin{cases} v_k = T_{\pi_k} v_k \\ \pi_{k+1} \in \mathcal{G}(v_k) \end{cases}.$$

Dans les cas qui nous intéresseront en pratique, l'espace d'états sera trop grand pour permettre une représentation exacte de la fonction de valeur ou de la politique (problème de représentation), et le modèle est inconnu, la dynamique du système et la fonction de récompense ne sont connues que via des données d'interactions, obtenues en ligne en contrôlant le système ou imposées a priori (problème d'échantillonnage).

Nous reviendrons sur la programmation dynamique approchée (qui traite ce cadre) en section 2.3, nous nous intéressons d'abord à un sous-problème important, l'estimation de la valeur.

5. Voir tout de même le chapitre 5, sur les perspectives en apprentissage par renforcement avec démonstrations expertes.

2.2 Estimation de la valeur

Nous considérons ici généralement le problème de l'estimation de la fonction de valeur (ou de la Q-fonction), pour une politique donnée ou directement pour la politique optimale. Comme cela a été annoncé dans la section précédente, deux problèmes se posent, celui de la représentation et celui de l'échantillonnage.

Le problème d'échantillonnage vient du fait que le modèle (noyau stochastique et fonction de récompense) n'est pas connu. Toutefois, nous supposons disposer d'une information partielle de ce modèle sous la forme d'une collection de transitions

$$\left\{(s_j, a_j, r_j, s'_j)_{1\leq j\leq i}\right\},$$

où $r_j = \mathcal{R}(s_j, a_j)$ et $s'_j \sim P(.|s_j, a_j)$. Ces transitions peuvent être issues d'une même trajectoire (générée par une politique donnée), mais ce n'est pas obligatoire (on peut envisager que les couples (s_j, a_j) soient générés selon une certaine loi, l'état suivant étant échantillonné selon la dynamique, cela dépendra du contexte). Le modèle étant inconnu, les opérateurs de Bellman ne peuvent pas être appliqués, mais on peut en considérer des versions échantillonnées (les estimateurs résultants étant non-biaisés) :

$$\begin{aligned}
\hat{T}_{\pi,j} : v \in \mathbb{R}^{\mathcal{S}} &\to \hat{T}_{\pi,j} v = r_j + \gamma v(s'_j) \in \mathbb{R} \text{ (seulement si } a_j \sim \pi(.|s_j)),\\
\hat{T}_{\pi,j} : Q \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}} &\to \hat{T}_{\pi,j} Q = r_j + \gamma \mathbb{E}[Q(s'_j, A)|A \sim \pi(.|s'_j)] \in \mathbb{R},\\
\hat{T}_{*,j} : Q \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}} &\to \hat{T}_{*,j} Q = r_j + \gamma \max_{a\in\mathcal{A}} Q(s'_j, a) \in \mathbb{R}.
\end{aligned}$$

Nous noterons parfois génériquement $\hat{T}_j$ l'opérateur échantillonné (sans spécifier s'il s'agit de l'opérateur d'évaluation ou d'optimalité). Nous ne définissons pas de version échantillonnée de l'opérateur d'optimalité de Bellman pour une valeur $v \in \mathbb{R}^{\mathcal{S}}$, dans la mesure où le modèle est nécessaire dans ce cas. Il s'agit donc de produire de bonnes approximations des points fixes de ces opérateurs, sous la contrainte de n'avoir accès qu'à des versions échantillonnées de ces derniers.

Le problème de représentation vient du fait que l'espace d'état est généralement trop grand pour permettre une représentation exacte de la fonction de valeur (sur les couples état-action). Nous considérerons dans un premier temps le cas où la valeur est paramétrée, c'est-à-dire que nous restreindrons notre recherche de la valeur aux espaces d'hypothèses

$$\mathcal{H} = \{v_\theta,\ \theta \in \mathbb{R}^d\} \text{ ou } \mathcal{H} = \{Q_\theta,\ \theta \in \mathbb{R}^d\}.$$

Un exemple classique est la paramétrisation linéaire. Soit $\phi : s \in \mathcal{S} \to \phi(s) \in \mathbb{R}^d$ un vecteur de fonctions de base (par exemple un réseau de fonctions à base radiale), la valeur sera $v_\theta(s) = \theta^\top \phi(s)$ (le raisonnement étant similaire pour Q_θ). Notons que ce cadre généralise celui où la valeur peut être exactement représentée, en choisissant $\phi(s) = e_s$, où e_s est le vecteur dont toutes les composantes sont nulles sauf celle (égale à 1) correspondant à l'état s. Même en ignorant le problème d'échantillonnage, rien ne garantit que le point fixe de l'opérateur d'intérêt appartienne effectivement à l'espace d'hypothèses considéré.

Enfin, notons que si le cadre de l'estimation de valeur peut paraître reminiscent de celui de la régression en apprentissage supervisé, ils sont en fait fort différents (même si le premier se réduit au second lorsque le facteur d'actualisation γ est nul, ce qui conduit à la régression de la récompense). En effet, la fonction d'intérêt (la valeur) n'est jamais observée, seule la récompense qui la définit l'est. De plus, le cadre probabiliste (lois des échantillons d'apprentissage) est (généralement) plus fin à gérer dans le cas de l'estimation de valeur.

2.2.1 Différences temporelles de Kalman

Nous présentons dans un premier temps brièvement la contribution principale de nos travaux doctoraux [63], les différences temporelles de Kalman [70]. L'idée de base est d'exprimer le problème d'estimation de la valeur comme un problème de filtrage :

$$\begin{cases} \theta_i & = \theta_{i-1} + \eta_i \\ 0 & = Q_{\theta_i}(s_i, a_i) - \hat{T}_i Q_{\theta_i} + n_i \end{cases}.$$

Dans ce modèle, les paramètres sont considérés comme étant des variables cachées qu'il s'agit d'inférer en fonction de l'historique des observations que sont les récompenses, le lien entre variable cachée et observations étant l'équation de Bellman (échantillonnée). La première équation (généralement appelée équation d'évolution) spécifie que les paramètres suivent une marche aléatoire (conduite par le bruit d'évolution η_i, au choix de l'utilisateur), ce qui permet d'être robuste au non-stationnarités (que l'environnement soit non-stationnaire ou que l'estimation se fasse conjointement au contrôle du système) [88]. La seconde équation, appelée équation d'observation, spécifie le lien entre variables cachées et observées. Ici, nous notons génériquement $\hat{T}_i$ pour $\hat{T}_{\pi,i}$ et $\hat{T}_{*,i}$, qui correspond respectivement aux modèles d'observation $r_i = Q_{\theta_i}(s_i, a_i) - \gamma Q_{\theta_i}(s'_i, \pi(s'_i)) + n_i$ dans le cas de l'évaluation d'une politique π (déterministe ici) et $r_i = Q_{\theta_i}(s_i, a_i) - \gamma \max_{a \in \mathcal{A}} Q_{\theta_i}(s'_i, a) + n_i$ dans le cas de l'approximation de Q_*. Le bruit d'observation n_i, au choix de l'utilisateur, rend compte notamment du bruit d'échantillonnage.

Le fait d'exprimer le problème d'estimation de la valeur comme un problème de filtrage permet de se reposer sur la vaste littérature traitant du sujet, plus particulièrement sur le filtrage de Kalman ici [6]. Nous avons tout particulièrement considéré le filtrage de Kalman non-parfumé [105], qui permet de prendre en compte des modèles d'observation non-linaires sans nécessiter de calcul de dérivée [7]. Cela permet de prendre en compte des paramétrisations non-linéaires, comme des réseaux de neurones à couche(s) cachée(s), mais aussi de considérer l'opérateur d'optimalité de Bellman (qui implique un opérateur max, et donc des non-différentiabilités).

Ce cadre de filtrage présente d'autres avantages. Il fournit naturellement une information de variance sur les paramètres estimés, ce qui s'avère utile pour développer des heuristiques traitant du dilemme entre exploration et exploitation [76]. Le bruit n_i est blanc en général, ce qui pose des problèmes de biais lorsque la dynamique est stochastique (KTD étant une approche résiduelle, voir la section suivante). Introduire un bruit coloré (particulier) à la place permet de supprimer ce biais, cette coloration étant réminiscente du concept plus

6. Notons que filtrage bayésien et filtrage de Kalman sont souvent confondus dans la littérature. Si la solution du filtrage bayésien est le filtre de Kalman dans le cas linéaire gaussien, les objectifs ne sont pas les mêmes. Le filtrage bayésien a vocation à estimer la loi de la variable cachée conditionnée aux observations passées. Les ambitions du filtrage de Kalman sont moindres : il s'agit de minimiser l'erreur quadratique moyenne (entre variable cachée estimée et réelle, conditionnellement aux observations) sous contrainte d'une mise à jour linéaire. Pour cela, il faut être capable de propager des moments d'ordre un et deux, mais aucune hypothèse gaussienne n'est nécessaire, même dans le cas non-linéaire.

7. De façon abstraite, pour filtrer, on doit calculer moments d'ordre 1 et 2 de la variable aléatoire $Y = f(X)$, connaissant ceux de X. Une première approche est de supposer X gaussien et de linéariser f en faisant un développement de Taylor, c'est le principe de Kalman étendu [194]. Connaissant la loi de X, une autre approche serait d'estimer ces moments par une méthode de Monte Carlo. Le principe de la transformation non-parfumée est similaire, sauf qu'au lieu d'échantillonner les points aléatoirement, on les choisit de façon déterministe en fonction des moyenne et variance de X. Voir [213] pour une vue d'ensemble du sujet.

classique en renforcement des traces d'éligibilités [68]. Nous avons considéré l'estimateur générique qu'est KTD dans le cadre du contrôle (en ligne), que ce soit dans un schéma de type itération de la politique, itération de la valeur ou encore acteur-critique [72] (la politique a une représentation propre, elle n'est plus déduite de l'acteur par gloutonnerie).

2.2.2 Une approche unifiée de l'estimation paramétrique de la valeur

Nous proposons ici un point de vue unifié des estimateurs paramétriques de la fonction de valeur [77, 78]. Si ce tour d'horizon ne peut pas être exhaustif, il se veut aussi large que possible (dans les limites du cadre considéré). Nous distinguerons trois types d'approches : le bootstrapping, les approches résiduelles et le point fixe projeté.

Le bootstrapping consiste informellement à poser le problème de l'estimation de la valeur comme un problème d'apprentissage supervisé (par exemple, $\min_\theta \sum (q_j - Q_\theta(s_j, a_j))^2$), à en déduire un algorithme itératif (par exemple par une descente de gradient stochastique ou des moindres carrés récursifs, génériquement $\theta_i = \theta_{i-1} + K_i(q_i - Q_{\theta_{i-1}}(s_i, a_i))$ où K_i est un gain) puis à remplacer les valeurs non observées par des estimations "bootstrappées", plus particulièrement par le résultat de l'application d'un opérateur de Bellman échantillonné à l'estimation courante de la valeur ($q_i \leftarrow \hat{T}_i Q_{\theta_{i-1}}$). On obtient ainsi les algorithmes classiques que sont TD [196] ou Q-learning [215], mais aussi FPKF [32] (*Fixed-Point Kalman Filter*). Plus formellement, on peut associer à ce principe le problème d'optimisation suivant [32] :

$$\theta_i \in \operatorname*{argmin}_{\theta \in \mathbb{R}^d} \frac{1}{i} \sum_{j=1}^{i} (\hat{T}_j Q_{\theta_{j-1}} - Q_\theta(s_j, a_j))^2.$$

Les approches résiduelles cherchent à minimiser directement le résidu de Bellman, $\|Q - TQ\|$. Le problème d'optimisation généralement associé est de la forme

$$\theta_i \in \operatorname*{argmin}_{\theta \in \mathbb{R}^d} \frac{1}{i} \sum_{j=1}^{i} (\hat{T}_j Q_\theta - Q_\theta(s_j, a_j))^2.$$

Le problème est que minimiser cette fonction objectif en considérant l'opérateur échantillonné fournit un estimateur biaisé [8] [5]. Toutefois, ce bias disparaît lorsque la dynamique est déterministe. KTD, présenté section 2.2.1, est en fait une approche résiduelle. C'est également le cas de GPTD [50] (*Gaussian Process Temporal Differences*) dans sa version paramétrique [49] ou de la descente de gradient proposée dans [9].

Enfin, les approches de type point fixe projeté cherchent à trouver un point fixe de l'opérateur de Bellman composé avec l'opérateur d'approximation (généralement de projection) sous-jacent à la régression. Elles peuvent différer selon qu'elles cherchent ce point fixe directement ($Q_\theta = \mathfrak{A} T Q_\theta$, avec $\mathfrak{A}$ opérateur d'approximation) ou de façon itérée ($Q_{\theta_i} = \mathfrak{A} T Q_{\theta_{i-1}}$). Les problèmes d'optimisation associés peuvent s'écrire respectivement

$$\theta_i \in \operatorname*{argmin}_{\theta \in \mathbb{R}^d} \frac{1}{i} \sum_{j=1}^{i} (\hat{T}_j Q_{\theta_i} - Q_\theta(s_j, a_j))^2$$

$$\text{et } \theta_i \in \operatorname*{argmin}_{\theta \in \mathbb{R}^d} \frac{1}{i} \sum_{j=1}^{i} (\hat{T}_j Q_{\theta_{i-1}} - Q_\theta(s_j, a_j))^2.$$

8. Essentiellement, la moyenne d'un carré n'est pas le carré de cette moyenne.

	bootstrapping	app. résiduelles	point fixe projeté direct / itéré	
descente de gradient stochastique	TD [196] TD-Q [180] Q-learning [215]	grad. résiduel [9]	(nl)GTD2 ([138])[198] (nl)TDC ([138])[198] Greedy-GQ [139]	
moindres carrés (récursifs)	FPKF [32]	GPTD [50] KTD [70]	LSTD [18] slLSTD [73]	LSPE [148] Q-OSP [219]

TABLE 2.1 – Taxinomie de quelques estimateurs paramétriques de la valeur.

Les représentants classiques de ces approches sont respectivement LSTD [18] (*Least-Squares Temporal Differences*) et LSPE [148] (*Least-Squares Policy Evaluation*), qui fournissent des solutions exactes à ces problèmes d'optimisation sous hypothèse de paramétrisation linéaire et d'opérateur d'évaluation (généralisable grâce au principe de linéarisation statistique sous-jacent à KTD [73]). Récemment, des algorithmes basés sur une descente de gradient stochastique minimisant ces problèmes d'optimisation ont été introduits [138, 198, 139], pour palier certains défauts de TD.

Pour résumer, l'ensemble des méthodes mentionnées peuvent se résumer par le problème d'optimisation

$$\theta_i \in \operatorname*{argmin}_{\theta\in\mathbb{R}^d} \frac{1}{i}\sum_{j=1}^{i}(\hat{T}_j Q_\xi - Q_\theta(s_j,a_j))^2, \tag{2.3}$$

où l'instantiation de ξ spécifie la méthode, bootstrapping ($\xi = \theta_{j-1}$), résiduelle ($\xi = \theta$) ou point fixe projeté direct ($\xi = \theta_i$) ou itéré ($\xi = \theta_{i-1}$). Une fois l'instantiation choisie, la minimisation peut être faite en considérant une descente de gradient (stochastique) ou une méthode de moindres carrés (récursifs), combinée éventuellement à une linéarisation (fonctionnelle, c'est-à-dire un développement de Taylor, ou statistique). Le tableau 2.1 résume ainsi un certain nombre d'approches, l'article [78] fournissant un état de l'art plus complet.

2.2.3 Extension aux traces d'éligibilité dans le cadre *off-policy*

Nous nous plaçons ici dans le cadre plus restreint de l'estimation de valeur (l'extension à l'estimation de la Q-fonction étant facile) linéairement paramétrée. Dans ce cadre, nous étendons l'idée développée section 2.2.2 aux traces d'éligibilité dans le cadre *off-policy* [188, 93].

Les méthodes d'estimation de la valeur sont des méthodes dites de différences temporelles. Une alternative possible (lorsqu'un simulateur est disponible) est, pour une collection d'états, d'estimer les valeurs par simulation (on parle de *monte carlo rollouts*, on tire un certain nombre de trajectoires dont on moyenne ensuite les récompenses cumulées et décomptées) puis de traiter l'approximation de la valeur comme un problème de régression. Les traces d'éligibilité [197] (initialement combinées à l'algorithme TD, pour donner TD(λ)) constituent un élégant pont entre ces deux approches. Elles permettent dans une certaine mesure de contrôler le compromis entre biais et variance [109, 212, 205]. Nous avons étendu le modèle décrit Eq. (2.3) aux traces d'éligibilité, pour dériver mécaniquement tous les algorithmes associés possibles (et les comparer) [93]. Nous nous sommes de plus placé dans le cadre *off-policy*, c'est-à-dire que la politique utilisée pour générer les trajectoires fournissant

les échantillons d'apprentissage diffère de la politique à évaluer, ce qui nécessite un correctif (basé sur la notion d'échantillonnage préférentiel [179]).

L'idée des traces d'éligibilité peut se formaliser comme la recherche du point fixe de la variation suivante de l'opérateur de Bellman [14],

$$\forall v \in \mathbb{R}^{\mathcal{S}}, \quad T_\pi^\lambda v = (1-\lambda)\sum_{k=0}^{\infty} \lambda^k T_\pi^{k+1} v,$$

qui est une moyenne géométrique de paramètre de profondeur $\lambda \in [0,1]$ des puissances de l'opérateur d'évaluation de Bellman T_π. Ces deux opérateurs partagent bien sûr le même point fixe. Cet opérateur peut s'exprimer de façon équivalente comme

$$[T_\pi^\lambda v](s) = v(s) + \sum_{k=1}^{\infty} (\gamma\lambda)^{k-i} \delta_{i,k}(s)$$
$$\text{avec } \delta_{i,k}(s) = \mathbb{E}[\mathcal{R}(S_k, A_k) + \gamma v(S_{k+1}) - v(S_k) | S_i = s, \pi].$$

Le terme $\delta_{i,k}(s)$ est un terme de différence temporelle dont nous souhaitons un estimateur basé sur les données. L'apprentissage se fait à partir d'une trajectoire

$$(s_0, a_0, r_0, s_1, \ldots s_n, a_n, r_n, s_{n+1}).$$

Toutefois, nous nous plaçons dans le cadre *off-policy*, nous supposons donc que la trajectoire a été échantillonnée selon une politique π_0 différente de la politique π à évaluer. Pour obtenir un estimateur non biaisé de l'erreur de différence temporelle, nous utilisons l'échantillonnage préférentiel. Notons respectivement

$$\rho(s,a) = \frac{\pi(a|s)}{\pi_0(a|s)} \text{ et } \rho_j^k = \prod_{l=j}^{k} \rho(s_l, a_l)$$

le poids (préférentiel) d'un couple état-action et le poids d'un morceau de trajectoire (avec la convention que si $k < j$, alors $\rho_j^k = 1$). Avec ces notations, le terme

$$\hat{\delta}_{i,k} = \rho_i^k \hat{T}_k v - \rho_i^{k-1} v(s_k)$$

est un estimateur non-biaisé de $\delta_{i,k}(s_i)$, à partir duquel il est possible de construire un estimateur $\hat{T}_{j,i}^\lambda v$ de $T^\lambda v_\pi(s_j)$ (la construction de l'estimateur dépendant de la perspective prise, gradient stochastique ou moindres carrés).

	gradient	moindres carrés
bootstrapping	off-policy TD(λ) [15]	off-policy FPKF(λ)
résiduel	off-policy gBRM(λ)	off-policy BRM(λ)
point fixe projeté	GQ(λ) a.k.a. off-policy TDC(λ) [140] off-policy GTD2(λ)	off-policy LSTD(λ) [218] off-policy LSPE(λ)

TABLE 2.2 – Taxinomie des algorithmes d'apprentissage *off-policy* avec traces d'éligibilité. Les algorithmes sans référence ont tous été proposés dans [93].

L'idée est ensuite de résoudre le problème d'optimisation

$$\theta_i \in \operatorname*{argmin}_{\theta\in\mathbb{R}^d} \frac{1}{i}\sum_{j=1}^{i}(\hat{T}^{\lambda}_{j,i} v_\xi - v_\theta(s_j))^2,$$

où ξ est instantié à θ_{j-1}, θ, θ_{i-1} ou θ_i, comme dans la section 2.2.2. Le problème est alors résolu avec une descente de gradient stochastique où une approche de moindres carrés récursifs (ce qui implique donc des opérateurs échantillonnés légèrement différent, comme dit précédemment). Le tableau 2.2 propose un résumé des algorithmes résultants, certains existant déjà et d'autres ayant été introduits pour la première fois par cette approche. Dans [93], nous faisons également une synthèse des garanties théoriques offertes par ces différentes approches et nous proposons une comparaison empirique exhaustive des différents estimateurs (sur des problèmes générés aléatoirement, appelés Garnets [6]). LSTD reste souvent l'estimateur le plus efficace, et TD est souvent la meilleure alternative lorsqu'on souhaite réduire le coût computationnel.

2.2.4 Vers les approches non-paramétriques

Les approches que nous avons vues jusqu'ici présentent comme défaut d'être paramétriques, c'est-à-dire de devoir choisir la représentation de la fonction de valeur a priori, ce qui est très dépendant du problème. C'est pourquoi nous nous sommes intéressé aux approches non-paramétriques. Nous nous sommes plus particulièrement inspiré de la régularisation ℓ_1 en apprentissage supervisé : la fonction de valeur est linéairement paramétrée, mais il y a beaucoup plus de fonctions de base que de transitions à partir desquelles apprendre (cadre classique de sur-apprentissage), et il faut donc découvrir une représentation parcimonieuse de la valeur. Nous nous intéressons ici aux variations de l'algorithme LSTD [9] (donc sur le principe du point fixe projeté). Cette section présente brièvement nos contributions à cette thématique [91, 94], respectivement inspirées des approches supervisées que sont Lasso [211, 48] et le sélecteur de Dantzig [19].

Commençons par rappeler quelques généralités sur LSTD, qui a pour objectif l'évaluation d'une politique π donnée (supposée ici déterministe, sans perte de généralité) pour une valeur linéairement paramétrée. Le modèle est inconnu, mais nous supposons disposer d'un ensemble de n transitions $\{(s_i, r_i, s'_i)_{1\leq i\leq n}\}$ où les états s_i sont tirés selon une distribution μ, les récompenses associées sont $r_i = \mathcal{R}(s_i, \pi(s_i))$ et les états s'_i sont tirés selon $P(.|s_i, \pi(s_i))$. Nous considérons une paramétrisation linéaire de la valeur, $v_\theta(s) = \theta^\top \phi(s)$. Notons $\Phi \in \mathbb{R}^{|S|\times d}$ la matrice de fonctions de base dont les lignes contiennent $\phi(s)^\top$, pour $s \in \mathcal{S}$. L'espace d'hypothèses peut alors s'écrire $\mathcal{H} = \{v_\theta = \Phi\theta,\ \theta \in \mathbb{R}^d\}$. Notons Π_μ la projection orthogonale sur $\mathcal{H}$ respectivement à la loi μ. Soit D_μ la matrice diagonale (supposée inversible) de diagonale μ et $M_\mu = \Phi^\top D_\mu \Phi$ la matrice de Gram associée. L'opérateur de projection s'écrit alors $\Pi_\mu = \Phi M_\mu^{-1}\Phi^\top D_\mu$. LSTD calcule asymptotiquement le point fixe de l'opérateur composé $\Pi_\mu T_\pi$: $v_{\theta_*} = \Pi_\mu T_\pi v_{\theta_*}$. Posons $A = \Phi^\top D_\mu(I - \gamma P_\pi)\Phi \in \mathbb{R}^{d\times d}$ (supposée inversible) et $b = \Phi^\top D_\mu \mathcal{R}_\pi$. On vérifie facilement que

$$v_{\theta_*} = \Pi_\mu T v_{\theta_*} \Leftrightarrow A\theta_* = b,$$

qui est donc (l'unique, sous les hypothèses d'inversibilité) solution asymptotique de LSTD.

9. Notons tout de même que, à notre connaissance, la première approche à mêler estimation de la valeur et régularisation ℓ_1 est [136], basée sur une approche résiduelle.

$\text{pen}_1 \setminus \text{pen}_2$	$\emptyset$	$\|.\|_2$	$\|.\|_1$
$\emptyset$	LSTD	✓	ℓ_1-PBR [91]
$\|.\|_2$	✓	$\ell_{2,2}$-LSTD [53]	$\ell_{2,1}$-LSTD [102]
$\|.\|_1$	LASSO-TD [125]	?	?

TABLE 2.3 – Résumé des approches régularisant LSTD (les coches correspondent à des cas particuliers et les points d'interrogation à des combinaisons qui n'ont pas encore été étudiées).

Dans les faits, le modèle étant inconnu, nous devons considérer des approximations stochastiques des quantités d'intérêt, basées sur les transitions observées. Notons $\tilde{\Phi}$ (respectivement $\tilde{\Phi}'$) $\in \mathbb{R}^{n\times d}$ la matrice empirique de fonctions de base dont les lignes sont $\phi(s_i)^\top$ (resp. $\phi(s_i')^\top$) pour $1 \leq i \leq n$ et $\tilde{\mathcal{R}} \in \mathbb{R}^n$ le vecteur de composantes r_i. Définissons les matrices aléatoires $\tilde{A} = \frac{1}{n}\tilde{\Phi}^\top \Delta\tilde{\Phi}$ et $\tilde{b} = \frac{1}{n}\tilde{\Phi}^\top \tilde{\mathcal{R}}$ avec $\Delta\tilde{\Phi} = \tilde{\Phi} - \gamma\tilde{\Phi}'$, qui sont clairement des estimateurs non-biaisés de A et b. LSTD résout le système linéaire $\tilde{A}\theta_n = \tilde{b}$, ce qui peut s'exprimer de façon équivalente comme le problème d'optimisation imbriqué suivant [10] :

$$\begin{cases} \omega_\theta & = \text{argmin}_\omega \|\tilde{\mathcal{R}} + \gamma\tilde{\Phi}'\theta - \tilde{\Phi}\omega\|_2^2 \\ \theta_n & = \text{argmin}_\theta \|\tilde{\Phi}\theta - \tilde{\Phi}\omega_\theta\|_2^2 \end{cases},$$

où la première équation correspond à la projection de l'opérateur et la seconde exprime le point fixe.

Lorsque des problèmes de sur-apprentissage se posent, il est naturel d'introduire un terme de régularisation. La régularisation ℓ_1 peut être particulièrement intéressante, dans la mesure où elle promeut la parcimonie des solutions calculées (en forçant certains coefficients à zéro). Le premier algorithme à combiner cette idée à LSTD est Lasso-TD [125], qui résout le problème suivant :

$$\theta_{l,\alpha} = \underset{\theta}{\text{argmin}} \|\tilde{\mathcal{R}}+\gamma\tilde{\Phi}'\theta_{l,\alpha}-\tilde{\Phi}\theta\|_2^2+\alpha\|\theta\|_1 \Leftrightarrow \begin{cases} \omega_\theta & = \text{argmin}_\omega \|\tilde{\mathcal{R}} + \gamma\tilde{\Phi}'\theta - \tilde{\Phi}\omega\|_2^2 + \alpha\|\omega\|_1 \\ \theta_{l,\alpha} & = \text{argmin}_\theta \|\tilde{\Phi}\theta - \tilde{\Phi}\omega_\theta\|_2^2 \end{cases}.$$

Lasso-TD peut donc de façon équivalente être vu comme un problème de point fixe régularisé (façon dont il a été introduit) où comme un problème d'optimisation imbriqué dans lequel c'est l'opérateur de projection qui est régularisé. L'inconvénient est que Lasso-TD ne découle pas d'un problème d'optimisation convexe. En conséquence, il requiert des méthodes de résolution ad-hoc et il nécessite certaines hypothèses qui ne sont pas vérifiées dans le cadre *off-policy* (cadre d'un grand intérêt pratique). Notons qu'avec $\gamma = 0$, on retrouve Lasso (problème de régression de la récompense avec régularisation ℓ_1, il n'y a plus de point fixe).

En voyant Lasso-TD sous sa forme imbriquée, on peut songer à faire varier les pénalisations :

$$\begin{cases} \omega_\theta & = \text{argmin}_\omega \|\tilde{\mathcal{R}} + \gamma\tilde{\Phi}'\theta - \tilde{\Phi}\omega\|_2^2 + \alpha_1\text{pen}_1(\omega) \\ \theta_{\alpha_1,\alpha_2} & = \text{argmin}_\theta \|\tilde{\Phi}\theta - \tilde{\Phi}\omega_\theta\|_2^2 + \alpha_2\text{pen}_2(\theta) \end{cases}$$

C'est ce que nous avons proposé en injectant le terme de régularisation sur l'équation de point fixe plutôt que celle de projection [91]. Cela permet d'avoir un problème d'optimisation convexe assez générique, qui ne fait pas d'hypothèse particulière sur le caractère *off-policy*

10. Ce problème imbriqué est lui-même équivalent à la formulation donnée section 2.2.2.

des données, mais sans offrir de garantie sur la qualité de la solution. Notons que le même algorithme a été introduit parallèlement et indépendamment par [102]. Le tableau 2.3 propose une vue d'ensemble des différentes variations de LSTD régularisées, sous ce point de vue. Notons que l'on pourrait pousser l'idée, par exemple en mélangeant régularisation ℓ_1 et ℓ_2 (c'est-à-dire, avec $\text{pen}(w) = \alpha\|w\|_1 + \beta\|w\|_2$), dans l'idée de ce que fait l'*elastic net* [221].

Nous avons également proposé une approche alternative, appelée D-LSTD (Dantzig-LSTD) [94]. L'idée est que si l'on souhaite résoudre le système linéaire $\tilde{A}\theta = \tilde{b}$, autant le régulariser directement (notons que cette idée est partagée par [175], avec une approche différente). Nous avons donc proposé le problème d'optimisation suivant :

$$\theta_{d,\alpha} = \operatorname*{argmin}_{\theta\in\mathbb{R}^d} \|\theta\|_1 \text{ sujet à } \|\tilde{A}\theta - \tilde{b}\|_\infty \leq \alpha.$$

C'est un problème d'optimisation convexe. Plus encore, c'est un programme linéaire, qu'on peut exprimer de façon équivalente par

$$\min_{u,\theta\in\mathbb{R}^d} \mathbf{1}^\top u \text{ sujet à } \begin{cases} -u \leq \theta \leq u \\ -\alpha\mathbf{1} \leq \tilde{A}\theta - \tilde{b} \leq \alpha\mathbf{1} \end{cases}.$$

Cette algorithme est fortement lié au sélecteur de Dantzig [19], que l'on retrouve avec $\gamma = 0$. Il fait moins d'hypothèses que Lasso-TD (le cadre *off-policy* ne pose aucun problème) et il peut être résolu avec n'importe quel solveur de programme linéaire (notamment la méthode de point intérieur de [19], qui utilise l'identité de Woodbury lorsque $n \ll d$). De plus, cet estimateur offre certaines garanties intéressantes. Notamment, en posant $B_{\infty,\phi} = \max_{s\in\mathcal{S}} \|\phi(s)\|_\infty$, on a, avec probabilité d'au moins $1-\delta$,

$$\inf_\alpha \|A\theta_{d,\alpha} - b\|_\infty \leq 2\left(\|\theta_*\|_1(1+\gamma)B_{\infty,\phi} + \|\mathcal{R}\|_\infty\right) B_{\infty,\phi}\sqrt{\frac{4}{n}\ln\frac{8d}{\delta}}.$$

C'est bien une mesure de l'erreur, car $\|A\theta - b\| = \|A(\theta - \theta_*)\|$. Cette borne est intéressante, dans la mesure où la dépendance (explicite) en le nombre de fonctions de base d n'est que logarithmique. Toutefois, c'est une borne d'oracle, car elle est vraie pour le meilleur facteur de régularisation α et elle fait intervenir la norme $\|\theta_*\|_1$ (l'algorithme sera donc d'autant plus efficace que la solution est effectivement parcimonieuse). D'avantage de discussions de ce résultat ainsi que des corollaires sur l'erreur de la valeur et des heuristiques de validation croisée sont discutées dans l'article original [94].

2.3 Schémas d'apprentissage du contrôle optimal

Nous avons discuté jusqu'à présent l'estimation de la valeur. C'est un aspect important, mais ce n'est toutefois qu'une partie du problème. En effet, en apprentissage par renforcement, c'est la politique optimale que l'on souhaite estimer. Nous avons vu section 2.1.2 que lorsque le modèle est connu, la programmation dynamique permet de calculer cette politique optimale. Nous nous intéresserons principalement aux méthodes dérivées de l'itération de la valeur et de la politique, respectivement données par

$$v_k = Tv_{k-1} \text{ et } \begin{cases} v_k & = v_{\pi_k} \\ \pi_{k+1} & = \mathcal{G}(v_k) \end{cases}.$$

La programmation dynamique approchée consiste principalement à remplacer l'application de ces opérateurs (d'optimalité pour l'itération de la valeur, point fixe de l'opérateur d'évaluation et opérateur de gloutonnerie pour l'itération de la politique) par des approximations, éventuellement dans un cadre asynchrone. L'itération de la valeur approchée (ou AVI pour *approximate value iteration*) peut se formaliser généralement par

$$v_k = Tv_{k-1} + \epsilon_k,$$

où ϵ_k est un terme d'erreur (causé par les problèmes de représentation et d'échantillonnage). Un représentant classique est *fitted-Q* [52] (voir aussi [97]), mais l'algorithme du Q-learning avec schéma d'exploration ϵ-glouton entre également dans ce cadre, entre autres. L'itération de la politique approchée (ou API pour *approximate policy iteration*) peut se formaliser généralement par :

$$\begin{cases} v_k & = v_{\pi_k} + \epsilon_k \\ \pi_{k+1} & = \mathcal{G}_{\epsilon'_k}(v_k) \end{cases},$$

où ϵ_k est l'erreur d'estimation de la valeur et ϵ'_k l'erreur sur l'opérateur de gloutonnerie :

$$\pi_{k+1} = \mathcal{G}_{\epsilon'_k}(v_k) \Leftrightarrow \forall \pi',\ T_{\pi'} v_k \leq T_{\pi_{k+1}} v_k + \epsilon'_k.$$

LSPI (*least-squares policy iteration*) [126] en est un représentant classique (dans ce cas, $\epsilon'_k = 0$, ce facteur n'étant non-nul que lorsque la politique a une représentation propre). L'algorithme SARSA avec schéma d'exploration ϵ-glouton peut en être vu comme une version asynchrone et optimiste. Dans ce cadre entre également ce que nous nommerons génériquement DPI (*direct policy iteration*) [129][127], où l'opérateur glouton est remplacé par un problème de classification (potentiellement multi-classe à coût sensitif), la valeur étant par exemple estimée par Monte Carlo.

Nous avons étudié des généralisations de ces approches ainsi que des alternatives.

2.3.1 Approximation de l'itération sur les politiques modifiée

L'itération sur les politiques modifiée [177] est une approche de programmation dynamique qui généralise l'itération de la valeur et l'itération de la politique. Le schéma algorithmique associé est le suivant :

$$\begin{cases} \pi_{k+1} & = \mathcal{G} v_k \\ v_{k+1} & = (T_{\pi_{k+1}})^m v_k \end{cases}.$$

On peut voir ce schéma comme une itération sur les politiques optimiste, où la phase d'évaluation est remplacée par m itérations de l'opérateur d'évaluation, le paramètre entier $m \geq 1$ étant libre. Avec $m = 1$, on retrouve l'itération de la valeur et avec $m = \infty$, on retrouve l'itération de la politique. Cette approche a un coût computationnel moindre que l'itération de la politique, tout en bénéficiant d'une convergence plus rapide que l'itération de la valeur [177].

Nous avons contribué à AMPI [186, 191] (*approximate modified policy iteration*), la généralisation au cas approché de l'itération sur les politiques modifiée. Le schéma algorithmique associé peut synthétiquement s'écrire comme :

$$\begin{cases} \pi_{k+1} & = \mathcal{G}_{\epsilon'_k} v_k \\ v_{k+1} & = (T_{\pi_{k+1}})^m v_k + \epsilon_k \end{cases}.$$

Cela généralise les schémas du type AVI [52, 4, 146] et du type API, avec ([127, 56, 129]) ou sans ([126]) étape de classification pour représenter la politique. L'analyse de la propagation d'erreurs dans AMPI généralise celles d'API [144] et d'AVI [145] (et la technique de preuve est originale, les arguments classiques pour API et AVI étant différents et incompatibles).

Le choix du facteur m n'est pas anodin. Considérons le cas où la phase d'évaluation est une régression et la phase d'amélioration une classification [11], l'instantiation correspondante étant appelée CBMPI [186, 191] (*classification-based modified policy iteration*). Dans ce cadre, le facteur m permet de faire un compromis entre la qualité d'estimation de la politique (classification) et celle de la fonction de valeur (régression). Appliqué au jeu Tetris (traditionnellement difficile pour le renforcement), cet algorithme fournit les meilleurs résultats publiés à ce jour [191] (meilleurs que ceux obtenus avec des approches de type optimisation en boîte noire, compétiteur principal jusqu'alors).

2.3.2 Recherche directe dans un espace de politiques

Une alternative à la programmation dynamique approchée est l'ensemble des méthodes de recherche dans un espace de politiques (ou LPS pour *local policy search*). Le principe consiste à se donner un espace de politiques Π (typiquement un espace de politiques stochastiques paramétrées) puis à chercher dans cet espace un maximum (local) de la fonction objectif [12]

$$J_\nu(\pi) = \mathbb{E}[v_\pi(S)|S \sim \nu] = \nu v_\pi. \tag{2.4}$$

Notons $d_{\mu,\pi}$ la mesure d'occupation γ-pondérée induite par la politique π lorsque l'état initial est échantillonné selon μ :

$$d_{\mu,\pi} = (1-\gamma)\mu(I - \gamma P_\pi)^{-1}.$$

Un résultat important est celui qui donne le gradient de cette fonction objectif [200] :

$$\nabla J_\nu(\pi) = \frac{1}{1-\gamma}\mathbb{E}[Q_\pi(S,A)\nabla \ln \pi(A|S)|S \sim d_{\nu,\pi}, A \sim \pi(.|S)].$$

Sous certaines conditions, la valeur Q_π peut être remplacée par une approximation dans ce gradient [200].

Il existe beaucoup d'approches de LPS (de maximisation de $J_\nu(\pi)$), qu'elles soient basées sur une montée de gradient [11] ou de gradient naturel [107], éventuellement avec un critique [200, 16, 156, 72] (c'est-à-dire que la valeur définissant le gradient est elle-même approchée), sur une approche de type EM (*expectation-maximization*) [122], voire sur une approche de type optimisation en boîte noire [101, 57]. Nous avons proposé quelques contributions algorithmiques à LPS [72, 57]. Nous avons également contribué à l'étude des garanties que ce type d'approche peut offrir, ainsi qu'au lien qui existe entre recherche directe de politique et programmation dynamique approchée [189]. C'est cela que nous discutons ici.

Dans la suite de cette section, nous supposons que l'espace de politiques Π est stable par mélange stochastique, c'est-à-dire que $\forall \pi, \pi' \in \Pi$, $\forall \alpha \in [0,1]$, $\alpha\pi + (1-\alpha)\pi' \in \Pi$. Introduisons une relaxation de l'opérateur de gloutonnerie,

$$\mathcal{G}_\Pi(\pi, \mu, \epsilon) = \{\pi' \in \Pi \text{ tel que } \forall \pi'' \in \Pi,\ \mu T_{\pi'} v_\pi + \epsilon \geq \mu T_{\pi''} v_\pi\}, \tag{2.5}$$

11. Nous nous plaçons donc en quelque sorte dans un cadre acteur-critique, dans la mesure où la valeur et la politique sont représentées, même si ce terme est plus classiquement utilisé dans le cadre de recherche directe dans un espace de politiques.

12. Par défaut, les vecteurs sont des vecteurs colonne, sauf ceux correspondant à des distributions qui sont des vecteurs ligne, ce qui explique la notation de multiplication à gauche νv_π pour noter l'espérance.

qui est l'ensemble des politiques (en se restreignant à Π) en moyenne (par rapport à μ) approximativement gloutonne (à ϵ près) par rapport à la valeur v_π. Il est possible de montrer qu'une politique qui est approximativement un optimum local de J_ν (dans le sens défini ci-après) est point fixe de l'opérateur de gloutonnerie approché :

$$\forall \pi' \in \Pi,\ \lim_{\alpha \to 0} \frac{J_\nu((1-\alpha)\pi + \alpha\pi') - J_\nu(\pi)}{\alpha} \leq \epsilon \Leftrightarrow \pi \in \mathcal{G}_\Pi(\pi, d_{\nu,\pi}, (1-\gamma)\epsilon).$$

Rappelons que la gloutonnerie (dans le sens $\pi \in \mathcal{G}(\pi)$) caractérise l'optimalité, cela s'étend au cas approché. Notons $\mathcal{E}_\nu(\Pi)$ la capacité ν-gloutonne de l'espace Π :

$$\mathcal{E}_\nu(\Pi) = \max_{\pi \in \Pi} \min_{\pi' \in \Pi} d_{\nu,\pi}(Tv_\pi - T_{\pi'} v_\pi)$$

Cette quantité caractérise la capacité de Π à représenter la politique gloutonne associée à n'importe quelle valeur induite par une politique de cet espace. Pour deux distributions μ et ν, notons $\|\frac{\mu}{\nu}\|_\infty = \max_{s \in \mathcal{S}} \frac{\mu(s)}{\nu(s)}$. On peut quantifier la qualité d'une politique qui est point fixe de l'opérateur de gloutonnerie approchée :

$$\pi \in \mathcal{G}_\Pi(\pi, d_{\nu,\pi}, \epsilon) \Rightarrow \forall \pi',\ \forall \mu \in \Delta_\mathcal{S},\ \mu v_{\pi'} \leq \mu v_\pi + \frac{1}{(1-\gamma)^2} \left\| \frac{d_{\mu,\pi'}}{\nu} \right\|_\infty (\mathcal{E}_\nu(\Pi) + \epsilon).$$

Ainsi, tout maximum local de $J_\nu(\pi)$ offre la garantie suivante :

$$\begin{aligned} &\forall \pi' \in \Pi,\ \lim_{\alpha \to 0} \frac{J_\nu((1-\alpha)\pi + \alpha\pi') - J_\nu(\pi)}{\alpha} \leq \epsilon \\ \Rightarrow &\mathbb{E}[v_*(S) - v_\pi(S) | S \sim \mu] \leq \frac{1}{1-\gamma} \left\| \frac{d_{\mu,\pi^*}}{\nu} \right\|_\infty \left(\frac{\mathcal{E}_\nu(\Pi)}{1-\gamma} + \epsilon \right). \end{aligned}$$

Ce résultat est particulièrement intéressant dans la mesure où tout maximum local offre une garantie globale. En effet, cette borne est tout à fait similaire à celles que l'on peut obtenir en programmation dynamique approchée (et même meilleure si l'on considère le coefficient de concentrabilité $\|\frac{d_{\mu,\pi^*}}{\nu}\|_\infty$, voir [189] pour une discussion à ce sujet).

Toutefois, cette approche présente également un inconvénient majeur : autant que nous le sachions, l'hypothèse de stabilité de Π par mélange stochastique n'est jamais vérifiée. En effet, considérant un ensemble d'actions discret, un choix classique de représentation est une paramétrisation de Gibbs : $\pi_\theta(a|s) \propto \exp(\theta^\top \phi(s,a))$. Pour des actions continues, il est usuel de paramétrer une politique déterministe u_θ, puis de la considerer comme moyenne d'une gaussienne : $\pi_\theta(a|s) \propto \exp(-\frac{1}{2\sigma^2}\|a - u_\theta(s)\|^2)$. Dans ces deux cas, Π n'est pas stable par mixture stochastique (en général). La portée du résultat énoncé en est sévèrement limité.

C'est pourquoi nous avons proposé un algorithme, dans le cadre LPS, qui satisfasse les hypothèses nécessaires. L'idée est de se donner un espace de politiques $\mathcal{P}$ et de chercher un maximum local de J_ν dans $\Pi = \text{co}(\mathcal{P})$, l'enveloppe convexe de $\mathcal{P}$, en faisant une montée de gradient fonctionnel (approche réminiscente du boosting [142]). A l'itération k, étant donnée la politique courante π_{k-1}, cela consiste à trouver le meilleur représentant du gradient fonctionnel ∇J_ν dans $\mathcal{P}$, puis à mettre à jour la politique :

1. calculer $h_k \in \text{argmax}_{h \in \mathcal{P}} \langle \nabla J_\nu(\pi_{k-1}), h \rangle$;
2. mettre à jour la politique : $\pi_k = (1-\alpha_k)\pi_{k-1} + \alpha_k h_k$, où $\alpha_k \in [0,1]$ est un taux d'apprentissage.

On peut montrer que

$$\operatorname*{argmax}_{h\in\mathcal{P}}\langle\nabla J_\nu(\pi),h\rangle=\operatorname*{argmin}_{h\in\mathcal{P}}d_{\nu,\pi}(Tv_\pi-T_hv_\pi).$$

En se restreignant à un espace $\mathcal{P}$ de politiques déterministes, cela peut se réécrire

$$\operatorname*{argmax}_{h\in\mathcal{P}}\langle\nabla J_\nu(\pi),h\rangle=\operatorname*{argmin}_{h\in\mathcal{P}}\mathbb{E}[\max_{a\in\mathcal{A}}Q_\pi(S,a)-Q_\pi(S,h(S))|S\sim d_{\nu,\pi}],$$

qui est un problème de classification multi-classe à coût sensitif (les entrées sont les états, les labels les actions, le coût pour choisir l'action a en l'état s étant $\max_{a'}Q_\pi(s,a')-Q_\pi(s,a)$) dont la solution idéale est bien sûr la politique gloutonne.

Il s'avère que l'algorithme résultant est en fait une variation de CPI [108] (*conservative policy iteration*). C'est une version amortie (dans le sens du mélange stochastique, considérer $\alpha_k=1$ supprimerait l'amortissement) de l'algorithme d'itération sur les politiques avec approximations, conçu de façon à garantir une amélioration de la politique à chaque itération (en tenant compte des approximations). Dans la mesure où nous l'avons dérivé comme solution possible de l'approche LPS, cela fait un lien entre recherche directe dans un espace de politiques et programmation dynamique approchée. Notons également le lien à DPI [129], où le taux d'apprentissage α_k vaut 1 et où le problème de classification à résoudre est

$$\operatorname*{argmin}_{h\in\mathcal{P}}\mathbb{E}[\max_{a\in\mathcal{A}}Q_\pi(S,a)-Q_\pi(S,h(S))|S\sim\nu].$$

La différence réside donc dans la distribution pondérant les états (qui ne tient plus compte de la politique courante, ce qui se traduit par un coefficient de concentrabilité dégradé, voir également [185] à ce sujet).

2.3.3 Valeur optimale et différences convexes

Une autre alternative à la programmation dynamique approchée est de minimiser directement $\|Q-TQ\|^2$, le résidu de Bellman pour l'opérateur d'optimalité. Cela pose le problème de biais habituel si les transitions sont stochastiques. Il existe des solutions, comme le double échantillonnage [9], l'utilisation d'un estimateur de Nadaraya-Watson [209] ou le plongement dans un espace de Hilbert à noyau reproduisant [133]. Pour simplifier, nous supposerons ici la dynamique déterministe. Nous motivons la minimisation de ce résidu et présentons une approche originale permettant de le faire [171].

Nous supposons ici une Q-fonction linéairement paramétrée, $Q_\theta(s,a)=\theta^\top\phi(s,a)$ avec $\phi(s,a)\in\mathbb{R}^d$, et nous notons $\mathcal{H}$ l'espace d'hypothèses associé. Pour une distribution μ sur les couples état-action, nous notons $\|Q\|_\mu^2=\mathbb{E}[Q(S,A)^2|(S,A)\sim\mu]$ la norme ℓ_2 pondéré par μ. L'apprentissage se fait grâce à un jeu de transitions $\{(s_i,a_i,r_i,s_i')_{1\leq i\leq n}\}$ où les couples (s_i,a_i) sont tirés selon μ (et s_i' est l'unique état résultant possible). Nous notons $\|Q\|_n^2=\frac{1}{n}\sum_{i=1}^n Q^2(s_i,a_i)$ la norme empirique correspondant. La minimisation du résidu de Bellman optimal (OBRM pour *optimal bellman residual minimization*) consiste donc à résoudre

$$\theta_n\in\operatorname*{argmin}_{\theta\in\mathbb{R}^d}\|TQ_\theta-Q_\theta\|_n^2=\operatorname*{argmin}_{\theta\in\mathbb{R}^d}\frac{1}{n}\sum_{i=1}^n(r_i+\gamma\max_{a\in\mathcal{A}}Q_\theta(s_i',a)-Q_\theta(s_i,a_i))^2.$$

Avant de discuter la minimisation effective, nous motivons le fait de poser ce problème d'optimisation.

algorithme	terme d'horizon	terme de concentrabilité	terme d'erreur
API/AVI	$\frac{2\gamma}{(1-\gamma)^2}$	$C_2(\nu,\mu)$	$\epsilon_{\text{API/AVI}}$
OBRM	$\frac{2}{(1-\gamma)}$	$C_1(\nu,\mu,\pi,\pi_*)$	ϵ_{OBRM}

TABLE 2.4 – Comparaison des bornes de API/AVI et OBRM.

Il est possible de montrer que pour tout $Q \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, en notant $\pi = \mathcal{G}(Q)$, on a

$$\|Q_* - Q_\pi\|_\nu \leq \frac{2}{1-\gamma} C_1(\nu,\mu,\pi,\pi_*)\epsilon_{\text{OBRM}} \text{ avec } \epsilon_{\text{OBRM}} = \|TQ - Q\|_\mu,$$

où $C_1(\nu,\mu,\pi,\pi_*)$ est un coefficient de concentrabilité (voir [171] pour l'expression exacte). D'autre part, on peut écrire génériquement la garantie offerte par l'itération de la valeur approchée ou de la politique approchée comme [144, 145, 55]

$$\limsup_{k\to\infty} \|Q_* - Q_{\pi_{k,\text{API/AVI}}}\|_p \leq \frac{2\gamma}{(1-\gamma)^2} C_2(\nu,\mu)\epsilon_{\text{API/AVI}},$$

où ϵ est un supremum sur les erreurs faites à chaque itération (qui dépend de l'algorithme considéré et de son instantiation particulière) et $C_2(\nu,\mu)$ est un coefficient de concentrabilité (voir à nouveau [171] pour l'expression exacte). Le tableau 2.4 résume la comparaison des bornes. Si les termes d'erreurs sont difficilement comparables (ils dépendent de l'instantiation particulière de chaque approche), le terme d'horizon est meilleur pour OBRM (et ce n'est pas un artefact de preuve dans le cas de la programmation dynamique approchée, seul considérer des politiques non-stationnaires permet à ce jour de supprimer le carré, voir [192] à ce sujet). De plus, le terme de concentrabilité relatif à OBRM est généralement meilleur (voir [171]).

Ce résultat est en faveur de l'approche proposée, mais il reste à savoir si elle est consistante au sens de Vapnik [214], c'est-à-dire si contrôler le résidu empirique $\|TQ-Q\|_n^2$ permet de contrôler le résidu d'intérêt $\epsilon_{\text{OBRM}} = \|TQ - Q\|_\mu$. La réponse est positive, elle repose essentiellement sur le calcul de la dimension de Vapnik-Chervonenkis du résidu [171]. En effet, on peut montrer qu'avec probabilité d'au moins $1-\delta$, on a

$$\forall Q_\theta \in \mathcal{H}, \quad \|TQ_\theta - Q_\theta\|_\mu^2 \leq \|TQ_\theta - Q_\theta\|_n^2 + O\left(\frac{\|\mathcal{R}\|_\infty}{1-\gamma}\sqrt{\frac{d|\mathcal{A}|}{n}\ln\frac{n}{\delta}}\right).$$

De plus, avec $\theta_n \in \text{argmin}_\theta \|TQ_\theta - Q_\theta\|_n$, on peut montrer qu'avec probabilité d'au moins $1-\delta$, on a

$$\|TQ_{\theta_n} - Q_{\theta_n}\|_\mu^2 \leq \inf_{Q_\theta\in\mathcal{H}} \|TQ_\theta - Q_\theta\|_\mu^2 + O\left(\frac{\|\mathcal{R}\|_\infty}{1-\gamma}\sqrt{\frac{d|\mathcal{A}|}{n}\ln\frac{n}{\delta}}\right).$$

Le facteur d'horizon est au carré sur la variance, mais pas sur le biais inductif. Ce résultat suggère donc qu'il peut être avantageux de minimiser directement le résidu optimal, car on contrôle alors l'écart à la politique optimale de la politique gloutonne respectivement à l'estimateur Q_{θ_n}.

Malheureusement, ce n'est pas un problème d'optimisation convexe, il n'y a pas de garantie aisée de trouver le minimum global. Une solution classique serait alors de chercher

un minimum local en utilisant une descente de sous-gradient (sous-gradient pour cause de présence de l'opérateur max). Toutefois, une alternative existe. Dans le domaine de l'optimisation, il arrive assez fréquemment qu'une fonction non-convexe puisse se décomposer en la différence de deux fonctions convexes. On parle alors de programmation DC (pour programmation par différences convexes) et des algorithmes de résolution efficaces existent [206, 207]. Il s'avère qu'il est possible d'exprimer le résidu empirique $\|TQ_\theta - Q_\theta\|_n^2$ comme la différence de deux fonctions convexes (voir [171] pour la décomposition exacte, le cas d'une norme ℓ_1 pondérée y étant également traité), ce qui permet de se reposer sur un pan important de la littérature en optimisation.

Chapitre 3

Apprentissage par imitation

Cette thématique de recherche est liée au projet ILHAIRE [1] et a fait l'objet des thèses co-dirigées par nos soins d'Edouard Klein (2010-2013) et de Bilal Piot (2011-2014) :

- Edouard Klein. *Contributions à l'apprentissage par renforcement inverse.* Thèse de Doctorat en Informatique, Université de Lorraine, 2013 ;
- Bilal Piot. *Apprentissage hors-ligne avec Démonstrations Expertes.* Thèse de Doctorat en Informatique, Université de Lorraine, 2014.

Nous commencerons par présenter section 3.1 quelques généralités sur l'apprentissage par renforcement inverse et plus généralement sur l'apprentissage par imitation. Le domaine étant relativement jeune, il n'y a pas encore d'ouvrage de référence sur le sujet, pour autant que nous le sachions. Le lecteur pourra cependant se référer aux thèses susmentionnées [113, 165] pour une introduction au sujet. La section 3.2 présentera les contributions en apprentissage par renforcement inverse, qui se basent sur un parallèle intéressant entre renforcement et classification. Les algorithmes résultants sont efficaces, mais nécessitent toujours d'optimiser une récompense pour imiter l'expert. Nous nous intéressons donc plus généralement à l'apprentissage par imitation (éventuellement également récompensé) dans la section 3.3.

3.1 Généralités

En apprentissage par imitation (*apprenticeship learning*, parfois également appelé apprentissage par démonstrations), le comportement d'un agent (dit expert) est observé et un autre agent (dit apprenti) doit apprendre à l'imiter. Cela fait naturellement penser à de l'apprentissage supervisé, mais il y a une dimension dynamique supplémentaire, ce type d'apprentissage étant considéré dans un contexte de contrôle. Formellement, nous nous plaçons ici encore dans le cadre des processus décisionnels de Markov. L'expert suit dans ce cas une politique π_E, optimale pour une récompense inconnue $\mathcal{R}_E$, c'est-à-dire $\pi_E = \pi_{*,\mathcal{R}_E}$ (dans la suite, nous indicerons naturellement les objets d'intérêt, comme les politiques et les fonctions de valeur, par la récompense). En toute généralité, les données d'apprentissage

1. Ce projet a motivé certains développements présentés dans ce chapitre, comme la conception d'algorithmes d'apprentissage par renforcement inverse qui ne nécessitent pas de résoudre le problème direct comme étape intermédiaire et qui puissent estimer une récompense à partir de données d'interactions, c'est-à-dire sans avoir recours à un simulateur. Nous reviendrons sur cet aspect au chapitre 4.

sont des transitions effectuées par l'expert :

$$\mathcal{D}_E = \{(s_i, a_i = \pi_E(s_i), s'_i)_{1 \leq i \leq n}\}.$$

Souvent, ces transitions sont issues de trajectoires (c'est-à-dire que $s'_i = s_{i+1}$ au sein d'une même trajectoire). L'apprenti doit donc apprendre à imiter l'expert. Cela peut se faire directement en cherchant à approcher l'objet π_E, ou moins directement en cherchant à estimer une récompense pour laquelle la politique de l'expert soit optimale (la politique de l'agent π_A optimisant alors cette récompense estimée).

3.1.1 Approche supervisée

Une approche naturelle consiste à directement estimer la politique π_E. Les actions étant supposées discrètes (et en nombre fini), c'est en fait un problème de classification multi-classe, les entrées étant les états et les labels étant les actions. Cela correspond (idéalement) à la minimisation du risque suivant (en supposant ici expert et agent déterministes) :

$$R(\pi_A) = \mathbb{E}[\mathbb{I}_{\{\pi_A(S) \neq \pi_E(S)\}}],$$

où $\mathbb{I}$ est la fonction indicatrice. Nous noterons parfois ce risque ϵ_c et l'appellerons erreur de classification. Le risque empirique correspondant est

$$R_n(\pi_A) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{\{\pi_A(s_i) \neq \pi_E(s_i)\}}.$$

Notons que nous n'utilisons qu'une partie de la base d'entraînement $\mathcal{D}_E$ (c'est-à-dire les couples $(s_i, \pi_E(s_i))_{1 \leq i \leq n}$, tout ce qui a trait à la dynamique du système est ignoré (les états suivants $(s'_i)_{1 \leq i \leq n}$).

Il n'est pas usuel de minimiser directement le risque basé sur la perte binaire (notamment pour des raisons de convexité, et même de continuité). Plutôt que de chercher à apprendre directement une politique (appelée règle de décision en classification), il est plus commun de chercher à apprendre une fonction de score $q \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$ dont on déduit la règle de décision :

$$\pi_A(s) \in \operatorname*{argmax}_{a \in \mathcal{A}} q(s, a). \tag{3.1}$$

Pour apprendre cette fonction de score, il faut introduire un substitut (ou proxy) convexe au risque binaire, par exemple [130, 8] (en supposant ici que le score satisfait $\sum_a q(s, a) = 0$) :

$$R_{n,\varphi}(q) = \frac{1}{n} \sum_{i=1}^{n} \sum_{a \in \mathcal{A}} \mathbb{I}_{\{a \neq \pi_E(s_i)\}} \varphi(q(s_i, a)), \tag{3.2}$$

où $\varphi(x)$ est une fonction convexe de sous-différentielle positive en zéro [2], par exemple $\varphi(x) = \exp(x)$ ou $\varphi(x) = \max(1 + x, 0)$. Notons que le choix d'un bon substitut convexe pour la classification multi-classe (éventuellement à coût sensitif) reste une question largement ouverte [8, 12]. Dans la suite, lorsque nous aurons besoin d'instancier un algorithme de classification, nous considérerons l'approche à large marge structurée de [208] (choix commode, mais ce que nous présenterons s'adapte à d'autres substituts convexes, tant qu'ils sont

2. D'autres hypothèses (techniques) sur φ peuvent être nécessaires, voir [8] et les références qui s'y trouvent pour plus de détails.

basés sur des fonctions de score). Soit $\mathcal{L} \in \mathbb{R}_+^{\mathcal{S}\times\mathcal{A}}$ une fonction de marge choisie par l'utilisateur, satisfaisant $\mathcal{L}(s, \pi_E(s)) \leq \mathcal{L}(s, a)$ (un choix canonique peut être $\mathcal{L}(s_i, \pi_E(s_i)) = 0$ et $\mathcal{L}(s_i, a \neq \pi_E(s_i)) = 1$). Le risque considéré est alors :

$$R_{n,\mathcal{L}}(q) = \frac{1}{n}\sum_{i=1}^{n}\left(\max_{a\in\mathcal{A}} q(s_i, a) + \mathcal{L}(s_i, a) - q(s_i, \pi_E(s_i))\right). \tag{3.3}$$

Dans tous les cas, traiter le problème de l'apprentissage par imitation sous le prisme de l'apprentissage supervisé permet de se reposer sur un large pan de la littérature et de bénéficier d'algorithmes dont les performances ont été éprouvées [3]. Toutefois, l'aspect dynamique est totalement ignoré (et le cadre posé des processus décisionnels de Markov est inutile).

3.1.2 Renforcement inverse

Si l'on considère le comportement de l'expert π_E comme une politique optimale pour une récompense inconnue $\mathcal{R}_E$, c'est-à-dire $\pi_E = \pi_{*,\mathcal{R}_E}$, l'imitation peut se faire en estimant une récompense $\hat{\mathcal{R}}$ pour laquelle l'expert est (approximativement) optimal, puis en optimisant cette récompense, la politique de l'agent étant alors $\pi_A = \pi_{*,\hat{\mathcal{R}}}$ (ou une estimation de cette politique optimale, en général). Estimer une récompense à partir d'un comportement supposé optimal relève de l'apprentissage par renforcement inverse [181, 151]. Il peut y avoir un intérêt d'interprétabilité : estimer la récompense pourrait permettre de comprendre quel est l'objectif de l'expert. Plus prosaïquement, dans le cadre de l'imitation, le renforcement inverse permet de prendre en compte la dynamique du système. On peut en espérer, entre autre, une meilleure capacité de généralisation et une moins grande sensibilité à des perturbations éventuelles de la dynamique. Fondamentalement, en utilisant l'apprentissage supervisé on cherche à apprendre le contrôleur, tandis qu'avec l'apprentissage par renforcement inverse on cherche à apprendre l'objectif du contrôle.

Toutefois, le problème tel que posé ci-dessus est ambigu. En effet, toute politique est optimale pour la récompense nulle, notamment la politique de l'expert. Cette solution ne présente bien sûr aucun intérêt. On pourrait contraindre le cadre, en cherchant à estimer une récompense telle que la politique de l'expert soit l'unique politique optimale. Au delà de la difficulté de formaliser cela, il y a toujours une infinité de solutions à ce problème (les politiques optimales étant invariantes par nombre de transformations de la récompense [150]). Le renforcement inverse est donc un problème difficile.

Pour palier cela, les algorithmes de l'état de l'art sont souvent basés sur un principe d'appariement de trajectoires. L'idée est d'estimer une récompense $\hat{\mathcal{R}}$ telle que les trajectoires induites par la politique optimale pour cette récompense, $\pi_{*,\hat{\mathcal{R}}}$, soient proches des trajectoires induites par la politique de l'expert. Les différents algorithmes se distinguent notamment par la notion de distance entre trajectoires (par exemple basée sur l'attribut vectoriel moyen [1, 201], que nous introduisons après, ou sur la divergence de Kullback-Leibler entre la distribution des trajectoires de l'expert et celle des trajectoires de l'agent [220]). L'article [149] résume certaines de ces approches (voir également [113, 165]), nous illustrons le principe de quelques-unes ici.

Supposons que la fonction de récompense est linéairement paramétrée : $\mathcal{R}_\theta(s, a) = \theta^\top\phi(s, a)$, avec $\phi(s, a) = (\phi_1(s, a), \ldots, \phi_d(s, a))^\top$. Cela implique pour la Q-fonction d'une

3. Leurs performances ont été éprouvées en terme de risque de classification, mais on peut se demander si c'est la bonne mesure qualitative à choisir dans le cadre de l'imitation. Nous y reviendrons.

politique π que (de même pour la fonction de valeur) :

$$\begin{aligned} Q_{\pi,\mathcal{R}_\theta}(s,a) &= \mathbb{E}[\sum_{t=0}^{\infty}\gamma^t\mathcal{R}_\theta(S_t,A_t)|S_0=s,A_0=a,S_{t+1}\sim P(.|S_t,A_t),A_{t+1}\sim\pi(.|S_{t+1})] \\ &= \theta^\top\mu_\pi(s,a) \\ \text{avec } \mu_\pi(s,a) &= \mathbb{E}[\sum_{t=0}^{\infty}\gamma^t\phi(S_t,A_t)|S_0=s,A_0=a,S_{t+1}\sim P(.|S_t,A_t),A_{t+1}\sim\pi(.|S_{t+1})]. \end{aligned}$$

Le terme $\mu_\pi(s,a)$ est appelé attribut vectoriel moyen (*feature expectation*), il dépend de la politique et de la dynamique mais pas de la récompense. Notons que chaque composante de cet attribut vectoriel moyen est en fait une Q-fonction pour la récompense qu'est la $i^{\text{ème}}$ fonction de base :

$$(\mu_\pi)_i = Q_{\pi,\phi_i}.$$

Estimer μ_π revient donc à estimer d Q-fonctions (le problème d'estimer cette quantité se posant en pratique), ce que nous avons proposé de faire avec une variation de l'algorithme LSTD [115] (l'algorithme résultant, LSTD-μ, est essentiellement LSTD où les calculs liés à la dynamique sont factorisés).

L'attribut vectoriel moyen peut être utilisé pour quantifier la proximité de trajectoires. En effet, restreignons-nous (sans perte de généralité, par invariance des politiques optimales aux translations de récompenses) au cas où $\|\theta\| = 1$. Nous avons (par Cauchy-Schwartz) :

$$|Q_{\pi_E,\mathcal{R}_\theta}(s,a) - Q_{\pi,\mathcal{R}_\theta}(s,a)| \leq \|\theta\|\|\mu_{\pi_E}(s,a) - \mu_\pi(s,a)\| = \|\mu_{\pi_E}(s,a) - \mu_\pi(s,a)\|.$$

Cette inégalité est vraie pour toute récompense $\mathcal{R}_\theta$. Ainsi, si deux politiques ont un même attribut moyen, elles auront la même valeur, quelle que soit la récompense (cela étant même vrai pour la récompense inconnue $\mathcal{R}_E$, si elle appartient à l'espace d'hypothèses). Cela suggère de chercher une récompense $\mathcal{R}_\theta$ telle que la politique optimale associée, $\pi_{*,\mathcal{R}_\theta}$, ait un attribut moyen vectoriel $\mu_{\pi_{*,\mathcal{R}_\theta}}$ aussi proche que possible de celui de l'expert. Par exemple, l'algorithme proposé dans [1] revient à résoudre

$$\min_{\theta\in\mathbb{R}^d}\left\|\mathbb{E}\left[\mu_{\pi_E}(S,A) - \mu_{\pi_{*,\mathcal{R}_\theta}}(S,A)|(S,A)\sim\nu\right]\right\|^2, \tag{3.4}$$

où ν est une loi sur les couples état-action, typiquement ici $\nu(s,a) = \nu_E(s)\pi_E(a|s)$ avec ν_E la loi sur les états initiaux des trajectoires de l'expert. L'algorithme proposé dans [201] consiste à résoudre

$$\begin{aligned} &\min_{\theta\in\mathbb{R}^d:\|\theta\|=1}\theta^\top\mathbb{E}\left[\mu_{\pi_{*,\mathcal{R}_\theta}}(S,A) - \mu_{\pi_E}(S,A)|(S,A)\sim\nu\right] \\ =&\min_{\theta\in\mathbb{R}^d:\|\theta\|=1}\mathbb{E}\left[Q_{*,\mathcal{R}_\theta}(S,A) - Q_{\pi_E,\mathcal{R}_\theta}(S,A)|(S,A)\sim\nu\right]. \end{aligned} \tag{3.5}$$

D'autres approches existent (pas forcément basées sur l'attribut vectoriel moyen), nous n'avons pas vocation à faire un état de l'art complet. Toutefois, la majorité de ces méthodes ont en commun la nécessité de résoudre un certain nombre de fois le problème (de renforcement) direct. Génériquement, il s'agit de minimiser une fonction de dissimilarité comme dans les équations (3.4) et (3.5). Cela se fait de façon itérative (penser par exemple à une descente de gradient, même si ce n'est pas nécessairement la technique d'optimisation utilisée). A chaque itération, il faut alors, pour le vecteur de paramètres courant θ_k,

estimer la politique optimale associée $\pi_{*,\mathcal{R}_{\theta_k}}$. Ceci n'est pas souhaitable, dans la mesure où le problème direct est complexe et source d'erreurs. Une exception notable est l'algorithme proposé par [17] (fondamentalement, le principe est le même, mais la résolution du problème direct est évitée grâce à une utilisation astucieuse de l'échantillonnage préférentiel). D'autre part, beaucoup de ces approches supposent connaître explicitement la dynamique, ou au moins avoir accès à un simulateur. Cela pose problème lorsqu'on ne dispose que de logs d'interactions et que le système est difficile à simuler [4]. Les développements qui suivent avaient notamment pour objectif de lever ces contraintes, ce qui suppose un changement de paradigme (par rapport au cadre de l'appariement de trajectoires).

3.2 Entre renforcement inverse et classification

Nous avons défini section 2.2 les notions de fonction de qualité et de politique gloutonne en apprentissage par renforcement (voir notamment Eq. (2.2)) et section 3.1.1 les notions de fonction de score et de règle de décision en classification (voir notamment Eq. (3.1)). On peut facilement faire un lien entre ces notions, même si elles sont issues de cadres différents [66]. La Q-fonction quantifie la valeur de chaque action pour un état donné, tandis que la fonction de score permet de classer les différents labels pour un état donné. La règle de décision en classification est en fait une politique gloutonne par rapport à une fonction de score. Ces différents objets ont donc des utilisations semblables (même si leurs définitions diffèrent), ce qui a inspiré les développements présentés ci-après.

3.2.1 Classification structurée

Nous présentons ici l'algorithme SCIRL [117] (*structured classification for inverse reinforcement learning*). Le principe est le suivant. La politique de l'expert (la règle de décision de l'oracle pour la classification) est gloutonne par rapport à la Q-valeur $Q_{\pi_E,\mathcal{R}_E}$ (vu comme une fonction de score pour la classification). Avec une politique linéairement paramétrée, nous avons que $Q_{\pi_E,\mathcal{R}_\theta}(s,a) = \theta^\top \mu_{\pi_E}(s,a)$. Nous avons vu que la gloutonnerie caractérisait l'optimalité (voir section 2.2), il s'agit donc de trouver un vecteur de paramètres θ_A tel que $\pi_E \in \mathcal{G}(Q_{\pi_E,\mathcal{R}_{\theta_A}})$ (ce qui signifie que la politique π_E est optimale pour la récompense $\mathcal{R}_A = \mathcal{R}_{\theta_A}$). En interprétant $Q_{\pi_E,\mathcal{R}_\theta}$ comme une fonction de score linéairement paramétrée, c'est en fait un problème de classification multi-classe. Le vecteur de paramètres appris sera celui de la récompense. SCIRL peut donc être résumé comme suit :

1. paramétrer la fonction de score par l'attribut vectoriel moyen de la politique de l'expert : $q_\theta(s,a) = \theta^\top \mu_{\pi_E}(s,a)$;
2. estimer le vecteur de paramètres θ_A à partir de la base d'apprentissage $\{(s_i, \pi_E(s_i))_{1\leq i\leq n}\}$, de façon à minimiser l'erreur de classification $\epsilon_c = \mathbb{E}[\mathbb{I}_{\{\pi_c(S)\neq\pi_E(S)\}}]$, où $\pi_c \in \mathcal{G}(q_{\theta_A})$ (c'est-à-dire que π_c est la règle de décision du classifieur) ;
3. le résultat est la récompense $\mathcal{R}_A = \mathcal{R}_{\theta_A}$.

Bien sûr, il n'est généralement par raisonnable de supposer connaître μ_{π_E} (ce qui reviendrait peu ou prou à connaître la politique de l'expert). Toutefois, cet attribut moyen peut être estimé à partir des données, via Monte Carlo [117] ou LSTD-μ [115] par exemple. SCIRL permet donc de réduire le problème de l'apprentissage par renforcement inverse à un

4. Cela arrive notamment lorsque c'est l'humain qui définit la dynamique du système, voir le chapitre 4 à ce sujet.

problème de classification (structuré par la dynamique du système, via la paramétrisation de la fonction de score). Il faut estimer μ_{π_E}, mais c'est un problème d'évaluation de politique (*on-policy* qui plus est), pour les récompenses que sont les fonctions de base $(\phi_i)_{1\leq i\leq d}$. C'est un problème beaucoup plus simple que celui de l'estimation d'une politique optimale pour une récompense arbitraire (sous-jacent aux approches d'appariement de trajectoires).

Si l'idée sous-jacente à SCIRL est raisonnablement intuitive, on peut se demander dans quelle mesure l'algorithme produit une bonne récompense. On peut montrer que [117]

$$0 \leq \mathbb{E}[v_{*,\mathcal{R}_A}(S) - v_{\pi_E,\mathcal{R}_A}(S)] \leq \frac{C_f}{1-\gamma}\left(\epsilon_Q + \epsilon_c \frac{2\gamma\|\mathcal{R}_A\|_\infty}{1-\gamma}\right), \tag{3.6}$$

où C_f est un coefficient de concentrabilité, ϵ_Q quantifie l'erreur faite lors de l'estimation de μ_{π_E} et ϵ_c est l'erreur de classification définie plus tôt. Ainsi, si l'on estime bien l'attribut vectoriel moyen (problème d'estimation de valeur) et que la règle de décision π_c est efficace (problème de classification), alors la politique π_E sera quasi-optimale pour la récompense estimée $\mathcal{R}_A$. On peut remarquer que cette borne est trivialement vraie pour $\mathcal{R}_A = 0$. Toutefois, ce cas n'est pas plausible, dans le sens où le classifieur a pour objectif de minimiser ϵ_c, alors que $\mathcal{R}_A = 0$ correspond à l'erreur de classification maximale (règle de décision uniformément aléatoire).

3.2.2 Cascade d'apprentissages supervisés

Nous présentons ici l'algorithme CSI [119] (*cascaded supervised inverse reinforcement learning*), qui repose également sur le parallèle entre classification et renforcement. Le principe est le suivant. Notons tout d'abord que, connaissant une Q-fonction optimale, la récompense associée peut être déduite de l'équation d'optimalité de Bellman (2.1) :

$$\mathcal{R}(s,a) = Q_{*,\mathcal{R}}(s,a) - \gamma\mathbb{E}[\max_{a'\in A} Q_{*,\mathcal{R}}(S',a')|S' \sim P(.|s,a)].$$

Apprenons un classifieur multi-classe sur la base d'entraînement $\{(s_i, \pi_E(s_i))_{1\leq i\leq n}\}$ et notons q_c la fonction de score calculée et $\pi_c \in \mathcal{G}(q_c)$ la règle de décision associée. Si l'on interprète cette fonction de score comme une Q-fonction, on peut en déduire une récompense $\mathcal{R}_A$ par l'équation précédente :

$$\begin{aligned}\mathcal{R}_A(s,a) &= q_c(s,a) - \gamma\mathbb{E}[\max_{a'\in A} q_c(S',a')|S' \sim P(.|s,a)] \\ &= q_c(s,a) - \gamma\mathbb{E}[q_c(S',\pi_c(S'))|S' \sim P(.|s,a)].\end{aligned}$$

La dynamique n'étant généralement pas connue, on suppose avoir accès à un ensemble de transitions $\{(s_j, a_j, s'_j)_{1\leq j\leq m}\}$ représentatives de la dynamique (les actions a_j ne sont donc pas systématiquement prises en accord avec la politique de l'expert). Si seules les transitions de l'expert sont disponibles, des heuristiques sont envisageables [119]. Le terme $r_j = q_c(s_j,a_j) - \gamma q_c(s'_j, \pi_c(s'_j))$ est alors un échantillon non-biaisé de $\mathcal{R}_A(s_j,a_j)$, et estimer la récompense revient à résoudre un problème de régression associant aux entrées (s_j,a_j) les sorties r_j. CSI peut donc se résumer comme suit :

1. estimer une fonction de score q_c et la règle de décision associée π_c par un classifieur multi-classe entraîné sur la base d'exemples $\{(s_i, \pi_E(s_i))_{1\leq i\leq n}\}$;
2. estimer la récompense $\mathcal{R}_A$ par un régresseur entraîné sur la base d'exemples

$$\{((s_j,a_j), r_j = q_c(s_j,a_j) - \gamma q_c(s'_j, \pi_c(s'_j)))_{1\leq j\leq m}\}.$$

CSI permet donc de réduire le problème de l'apprentissage par renforcement à la cascade de deux étapes supervisées, une classification puis une régression (et de se reposer sur la large littérature sur le sujet, notamment concernant les approches non-paramétriques).

L'idée sous-jacente à CSI est intuitive, mais on peut se demander dans quelle mesure la récompense estimée est de bonne qualité. On peut montrer que [119]

$$0 \leq \mathbb{E}[v_{*,\mathcal{R}_A}(S) - v_{\pi_E,\mathcal{R}_A}(S)] \leq \frac{C_g}{1-\gamma}\left(\epsilon_{\mathcal{R}} + \epsilon_c \frac{2\|\mathcal{R}_A\|_\infty}{1-\gamma}\right), \tag{3.7}$$

où C_g est un coefficient de concentrabilité (généralement plus fin que le coefficient C_f de la borne de SCIRL), ϵ_c est toujours l'erreur de classification et $\epsilon_{\mathcal{R}}$ quantifie l'erreur faite lors de l'étape de régression (terme plus facile à contrôler que le terme ϵ_Q de SCIRL). On a donc pour CSI une borne semblable à celle de SCIRL, son interprétation est similaire. Notons que dans les deux cas, la preuve est basée sur une propagation d'erreurs [5].

3.2.3 Abstraction : politiques d'ensembles

Le lien entre classification et renforcement, qui est l'idée sous-jacente à ces algorithmes, peut s'abstraire et se formaliser par le cadre des politiques d'ensembles [6].

Notons $\text{Supp}(\pi(.|s)) = \{a \in \mathcal{A} : \pi(a|s) > 0\}$ le support d'une politique π pour un état s. L'optimalité d'une politique peut se caractériser de la façon suivante :

$$v_{\pi,\mathcal{R}} = v_{*,\mathcal{R}} \Leftrightarrow \forall s \in \mathcal{S},\ \text{Supp}(\pi(.|s)) \subset \underset{a\in\mathcal{A}}{\text{argmax}}\, Q_{*,\mathcal{R}}(s,a). \tag{3.8}$$

Pour caractériser l'optimalité d'une politique, il est donc nécessaire et suffisant de considérer pour chaque état l'ensemble des actions de probabilité non nulle, pas les probabilités mêmes [7]. Cela suggère d'introduire la notion de politique d'ensembles $\bar{\pi} \in (2^{\mathcal{A}}\backslash\emptyset)^{\mathcal{S}}$, qui à chaque état associe un ensemble non-vide d'actions. On a naturellement des notions d'inclusion,

$$\bar{\pi}_1 \subset \bar{\pi}_2 \Leftrightarrow \forall s,\ \bar{\pi}_1(s) \subset \bar{\pi}_2(s),$$

et d'égalité,

$$\bar{\pi}_1 = \bar{\pi}_2 \Leftrightarrow \bar{\pi}_1 \subset \bar{\pi}_2 \text{ et } \bar{\pi}_2 \subset \bar{\pi}_1.$$

On peut également naturellement associer à une politique stochastique $\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}$ une politique d'ensembles $\bar{\pi}$ définie par $\bar{\pi} : s \in \mathcal{S} \to \text{Supp}(\pi(.|s)) \in 2^{\mathcal{A}}\backslash\emptyset$. Une politique déterministe peut donc s'identifier avec la politique d'ensembles associée. Enfin, on peut définir la politique d'ensembles optimale $\bar{\pi}_{*,\mathcal{R}}$ par $\bar{\pi}_{*,\mathcal{R}}(s) = \text{argmax}_{a\in\mathcal{A}}\, Q_{*,\mathcal{R}}(s,a)$. La caractérisation d'optimalité de l'équation (3.8) peut se réécrire dans ce cadre : $v_{\pi,\mathcal{R}} = v_{*,\mathcal{R}} \Leftrightarrow \bar{\pi} \subset \bar{\pi}_{*,\mathcal{R}}$.

Retournons au problème de l'apprentissage par renforcement inverse. Supposons que la politique de l'expert π_E est optimale pour une récompense inconnue $\mathcal{R}_E$, ce qui se traduit par

$$v_{\pi_E,\mathcal{R}_E} = v_{*,\mathcal{R}_E} \Leftrightarrow \bar{\pi}_E \subset \bar{\pi}_{*,\mathcal{R}_E}. \tag{3.9}$$

5. Cette propagation peut se faire plus ou moins finement, elle dépendra également de la distribution des données d'apprentissage et de ce que l'on veut faire apparaître comme loi sur les états dans le terme borné. Tout cela impacte les coefficients de concentrabilité. La forme la plus fine (à ce jour) de ces bornes se trouve dans [165].

6. Un article relatif à ce paradigme est en cours de soumission, mais plus de détails peuvent se trouver dans la thèse [165].

7. Un résultat classique est qu'il existe au moins une politique optimale déterministe. Toutefois, dans le contexte de l'apprentissage par renforcement inverse, il est souhaitable d'élargir ce cadre, notamment car l'expert n'est pas forcément déterministe.

Si l'on formalise le renforcement inverse comme étant le problème de trouver une récompense $\mathcal{R}_A$ telle que la politique de l'expert soit optimale, cela se traduit directement en termes de politiques d'ensembles :

$$v_{\pi_E,\mathcal{R}_A} = v_{*,\mathcal{R}_A} \Leftrightarrow \bar{\pi}_E \subset \bar{\pi}_{*,\mathcal{R}_A}. \tag{3.10}$$

On peut très bien vérifier les conditions (3.9) et (3.10) tout en ayant par ailleurs [8]

$$\bar{\pi}_{*,\mathcal{R}_E} \subset \bar{\pi}_{*,\mathcal{R}_A}.$$

Cela signifie que des actions optimales pour $\mathcal{R}_A$ ne le sont pas pour $\mathcal{R}_E$, ce qui n'est pas souhaitable. Une façon exacte de résoudre le problème de l'apprentissage par renforcement inverse (ARI) serait alors de chercher une fonction de récompense $\mathcal{R}_A$ telle que $\bar{\pi}_{*,\mathcal{R}_A} = \bar{\pi}_E$ (ARI exact). On peut également relâcher la contrainte d'égalité et se contenter de trouver une récompense $\mathcal{R}_A$ telle que $\bar{\pi}_{*,\mathcal{R}_A} \subset \bar{\pi}_E$ (ARI relâché), ce qui par transitivité donne

$$\bar{\pi}_{*,\mathcal{R}_A} \subset \bar{\pi}_E \subset \bar{\pi}_{*,\mathcal{R}_E}.$$

Cela signifie que toute action choisie par la politique optimale respectivement à $\mathcal{R}_A$ sera également optimale pour la politique optimisant $\mathcal{R}_E$, récompense inconnue de l'expert.

Soit $\bar{\pi}$ une politique d'ensemble, notons $C_{\bar{\pi}}$ l'ensemble des récompenses pour lesquelles $\bar{\pi}$ est optimale :

$$C_{\bar{\pi}} = \left\{\mathcal{R} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}},\ \bar{\pi} = \bar{\pi}_{*,\mathcal{R}}\right\}.$$

L'ARI exact revient à chercher un élément de $C_{\bar{\pi}_E}$, tandis que l'ARI relâché revient à chercher un élément de $\bigcup_{\bar{\pi}\subset\bar{\pi}_E} C_{\bar{\pi}}$ (qui contient notamment $C_{\bar{\pi}_E}$, l'ARI relâché est donc plus simple). Notons J_* l'opérateur de Bellman inversé (introduit section 3.2.2 pour CSI) :

$$\forall Q \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}},\quad [J_*Q](s,a) = Q(s,a) - \gamma\mathbb{E}[\max_{a'\in\mathcal{A}} Q(S',a')|S' \sim P(.|s,a)]. \tag{3.11}$$

Cet opérateur est une bijection de l'ensemble $\mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ dans lui-même et nous avons

$$\forall Q \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}},\quad Q = Q_{*,\mathcal{R}} \text{ avec } \mathcal{R} = J_*Q.$$

Enfin, introduisons $H_{\bar{\pi}}$, l'ensemble des fonctions de score dont le support de l'argmax pour chaque état s est l'image de $\bar{\pi}(s)$:

$$H_{\bar{\pi}} = \left\{Q \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}} : \forall s \in \mathcal{S},\ \operatorname*{argmax}_{a\in\mathcal{A}} Q(s,a) = \bar{\pi}(s)\right\}.$$

Cet ensemble est en fait l'image par J_* de l'ensemble des récompenses pour lesquelles $\bar{\pi}$ est optimale :

$$J_*(H_{\bar{\pi}}) = C_{\bar{\pi}}.$$

Par ailleurs, trouver un élément de $H_{\bar{\pi}}$ à politique d'ensembles donnée (ou observée) est en fait essentiellement un problème de classification multi-classe à fonction de score. Ainsi, nous venons de montrer que la classification et l'apprentissage par renforcement inverse sont en bijection par J_*.

Cela suggère deux approches canoniques pour le problème du renforcement inverse (relâché ici). La première peut se résumer par :

8. C'est le pendant pour les politiques d'ensembles de la récompense nulle pour laquelle toute politique est optimale, notamment celle de l'expert.

1. trouver $Q \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ tel que $Q \in \bigcup_{\bar{\pi}\subset\bar{\pi}_E} H_{\bar{\pi}}$;
2. définir $\mathcal{R} = J_* Q$.

On retrouve ici une forme abstraite de l'algorithme CSI, où la première étape correspond à un problème de classification et la seconde à une régression. La deuxième approche peut se résumer par :

1. trouver $\mathcal{R} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ telle que $J_*^{-1}\mathcal{R} \in \bigcup_{\bar{\pi}\subset\bar{\pi}_E} H_{\bar{\pi}}$.

On retrouve ici une forme abstraite de l'algorithme SCIRL, où la fonction de score cherchée (problème de classification) est l'image réciproque par J_* d'une récompense (ce qui se traduit pour une récompense linéairement paramétrée par l'expression de la fonction de score comme combinaison linéaire des composantes de l'attribut vectoriel moyen).

3.3 Imitation et contrôle

Ainsi, les algorithmes présentés section 3.2 permettent de réduire l'apprentissage par renforcement inverse à une combinaison d'estimation de valeur et de classification pour SCIRL, à une cascade de classification et de régression pour CSI. Cela permet notamment de ne pas avoir à résoudre le problème direct (estimer la politique pour une récompense arbitraire) de façon intermédiaire, contrairement à la majorité des algorithmes de l'état de l'art. Toutefois, ce qui est estimé est une récompense, qu'il faut encore optimiser pour imiter l'expert. Dans la mesure où optimiser une récompense est un problème difficile, on peut légitimement se demander s'il n'est pas plus simple et efficace d'adopter une approche supervisée, telle que décrite section 3.1.1.

3.3.1 Nécessité d'estimer une récompense ?

Cette discussion a initialement été présentée dans l'article [168]. Supposons que l'imitation soit faite par une approche supervisée, c'est-à-dire que la politique de l'agent π_A soit estimée de façon à minimiser l'erreur de classification $\epsilon_c = \mathbb{E}[\mathbb{I}_{\{\pi_A(S)\neq\pi_E(S)\}}]$ (ce qui est fait en pratique en minimisant un proxy empirique et convexe, mais nous supposons ici que la méthode de classification mise en oeuvre permet de contrôler ϵ_c). Comme nous l'avons dit précédemment, l'imitation se fait dans un contexte de contrôle, plus particulièrement d'un processus décisionnel de Markov ici. C'est plus l'écart à la valeur optimale que l'erreur de classification qui permet de quantifier la qualité du contrôle. Il est possible de lier ces deux termes [9].

En effet, on peut montrer que pour toute fonction de récompense $\mathcal{R} \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, on a

$$\mathbb{E}[|v_{\pi_E,\mathcal{R}}(S) - v_{\pi_A,\mathcal{R}}(S)|] \leq \frac{C_c}{(1-\gamma)^2} 2\|\mathcal{R}\|_\infty \epsilon_c,$$

où C_c est un coefficient de concentrabilité [10] et ϵ_c l'erreur de classification définie ci-avant. On peut mettre en parallèle ce résultat avec les bornes obtenus par SCIRL (3.6) et CSI (3.7). Tout d'abord, ce résultat est vrai pour toute récompense, et notamment pour la récompense $\mathcal{R}_E$, alors qu'il ne l'est que pour la récompense estimée dans le cas de SCIRL et CSI. Ensuite,

9. Nous le faisons ici dans le cadre à horizon infini, mais cela a été fait auparavant dans le cadre à horizon fini [202].

10. Ce coefficient peut être aussi petit que 1 (si la loi des données est la distribution stationnaire de la politique de l'expert, et que la même mesure est utilisée pour mesurer l'écart entre les valeurs).

algo.	SCIRL	CSI	classif.
terme contrôlé	$\mathbb{E}[v_{\pi_A,\mathcal{R}_A}(S) - v_{\pi_E,\mathcal{R}_A}(S)]$		$\mathbb{E}[\lvert v_{\pi_A,\mathcal{R}}(S) - v_{\pi_E,\mathcal{R}}(S)\rvert]$, $\forall\mathcal{R}$
horizon	$\frac{1}{(1-\gamma)^2}$	$\frac{1}{(1-\gamma)^2}$	$\frac{1}{(1-\gamma)^2}$
concentrabilité	C_g	C_f	C_c
erreur classif.	$2\gamma\lVert\mathcal{R}_A\rVert_\infty\epsilon_c$	$2\lVert\mathcal{R}_A\rVert_\infty\epsilon_c$	$2\lVert\mathcal{R}\rVert_\infty\epsilon_c$
erreur autre	$(1-\gamma)\epsilon_Q$	$(1-\gamma)\epsilon_\mathcal{R}$	0

TABLE 3.1 – Comparaison des bornes de SCIRL, CSI et de la classification.

on peut comparer les autres termes, à l'aide du tableau 3.1. Les termes d'horizon sont les mêmes et les termes liés à l'erreur de classification sont comparables. Nous n'entrons pas dans le détail des coefficients de concentrabilité, mais C_c est raisonnablement le plus petit. Enfin, il n'y pas de terme d'erreur lié à l'estimation de valeur ou à une régression pour l'approche supervisée.

Ainsi, ces différentes bornes laissent penser que l'approche supervisée est la plus efficace, dans le sens où elle offre une meilleure garantie. Toutefois, rien n'assure que ces bornes soient fines, et elle ne font aucune hypothèse particulière sur la structure de la fonction de récompense, par exemple. Cela appelle donc une étude empirique, aussi exhaustive que possible. C'est ce qui a été fait dans [168]. Les problèmes considérés sont des Garnets [6], qui sont des processus décisionnels de Markov générés aléatoirement. Dans ces expériences, la structure de la récompense varie également : soit c'est une récompense sur les couples état-action, non-parcimonieuse (récompense très informative), soit c'est une récompense parcimonieuse, uniquement sur les états (récompense non-informative). Notons qu'aucun algorithme n'a accès à la récompense, seules des traces de l'expert sont observées. Il s'avère que si dans le cas d'une récompense informative, la classification est tout à fait compétitive (et même fournit les meilleurs résultats, en moyenne), dans le cas d'une récompense parcimonieuse c'est l'apprentissage par renforcement inverse qui fournit les meilleurs résultats (et plus particulièrement les algorithmes introduits section 3.2). Ainsi, la structure de la récompense joue : informellement, pour une récompense parcimonieuse, l'horizon d'optimisation est plus important, un algorithme de renforcement inverse fournit de meilleurs résultats. Nous avons également considéré le cas où la dynamique du système est perturbée (cas également non couvert par l'analyse). Dans ce cas, le renforcement inverse fait mieux que le classifieur idéal (qui correspond au transfert de la politique optimisant la récompense inconnue sur la dynamique originale). Le lecteur peut consulter [168, 165] pour plus de détails.

3.3.2 Régulariser par la récompense

Il y a donc un avantage à choisir une approche de type renforcement inverse, d'autant plus que la dynamique du système est importante (ce qui est a priori le cas dans un contexte de contrôle, cadre de l'imitation). Toutefois, une approche par classification (qui ignore l'aspect dynamique) reste attractive, car plus simple, mieux étudiée et ne nécessitant pas d'optimiser de récompense. L'idéal serait de combiner le meilleur des deux mondes : souplesse de la classification et prise en compte de la dynamique du renforcement inverse. C'est l'objet de l'algorithme RCAL [169] (*reward-regularized classification for apprenticeship learning*) résumé ici.

Considérons généralement le problème de classification multi-classe basée sur une fonc-

tion de score q, tel que discuté section 3.1.1. Le problème d'optimisation associé est alors la minimisation d'un risque $R_n(q)$, par exemple ceux donnés équations (3.2) ou (3.3). Un classique de l'apprentissage supervisé est l'ajout d'un terme de régularisation qui va pénaliser les solutions complexes (typiquement pour éviter les problèmes de sur-apprentissage). La fonction à minimiser devient

$$\mathcal{J}_{n,\alpha}(q) = R_n(q) + \alpha\Omega(q),$$

où $\Omega(q)$ est un terme de pénalisation de la complexité de q et α un facteur de compromis entre minimisation du risque et complexité de la solution. Par exemple, pour une représentation paramétrique de q, $\Omega(q)$ peut être la norme ℓ_2 ou la norme ℓ_1 du vecteur de paramètres. Dans le cas d'une approche par noyaux, $\Omega(q)$ peut être la norme (dans l'espace de Hilbert à noyau reproduisant considéré) de la fonction de score.

Rappelons que dans le cadre de l'imitation (formalisé avec des processus décisionnels de Markov), à chaque fonction de score q est associée une unique récompense $\mathcal{R}$ par l'opérateur J_* (3.11), $\mathcal{R} = J_*q$, comme nous l'avons vu section 3.2.3. L'idée sous-jacente à RCAL pour prendre en compte la dynamique du système est simple, elle consiste à pénaliser la complexité de la récompense associée à la fonction de score, plutôt que celle du score lui-même. Le problème d'optimisation associé est génériquement :

$$\mathcal{J}_{n,\alpha}(q) = R_n(q) + \alpha\Omega(J_*q). \tag{3.12}$$

Dans l'article original [169], cette idée est instanciée en considérant le risque de l'équation (3.3) et en régularisant par la norme ℓ_1 (empirique) de la récompense. La fonction $\mathcal{J}_{n,\alpha}(q)$ est ensuite minimisée avec une descente de gradient fonctionnel restreint (c'est-à-dire du boosting [142, 99]). Nous n'avons pas de garantie théorique pour cet algorithme, toutefois il présente effectivement de très bonnes performances empiriques [169, 165], en donnant généralement de meilleurs résultats que la classification et que le renforcement inverse, que le cadre experimental soit avantageux à l'une ou l'autre méthode (*cf.* section 3.3.1). Il partage cependant les inconvénients des méthodes qui n'estiment pas de récompense, comme un manque de stabilité face aux perturbations de la dynamique.

3.3.3 Imitation et interactions récompensées

Jusqu'à présent, nous avons supposé ne disposer que de démonstrations d'un expert (et éventuellement de données représentatives de la dynamique du système). Dans cette section, nous nous plaçons dans le cas où un signal de récompense est également disponible. Autrement dit, nous considérons avoir à disposition les deux jeux de données

$$\mathcal{D}_E = \{(s_i, a_i = \pi_E(s_i))_{1\leq i\leq n}\} \text{ et } \mathcal{D}_{AR} = \{(s_j, a_j, r_j, s'_j)_{1\leq j\leq m}\}.$$

On parle alors d'apprentissage par renforcement avec démonstrations expertes. C'est un cadre d'étude intéressant pour l'apprentissage par renforcement inverse [11], mais aussi pour le renforcement classique (où l'on peut supposer à la fois avoir une récompense et des exemples de bons contrôleurs, au moins localement). Dans la mesure où deux sources d'information, comportement expert et signal de récompense, sont disponibles, il est souhaitable de les

11. Par exemple, la récompense peut être apprise hors-ligne avec l'un des algorithmes présentés section 3.2, puis peut être optimisée en tenant compte des exemples de l'expert, le jeu de données (non expertes) pouvant de plus être augmenté d'interactions avec le système. Voir [165] pour une discussion plus poussée.

exploiter de concert. C'est ce qui est proposé dans l'article [170] que nous résumons ici. Notons que ce cadre est, autant que nous le sachions, très peu étudié [12], les principales alternatives étant [112, 30].

Nous avons proposé section 2.3.3 d'estimer une politique optimale en minimisant directement le résidu de Bellman pour l'opérateur d'optimalité (nous supposons temporairement la dynamique déterministe, comme nous considérons une approche résiduelle, mais une dynamique stochastique peut être prise en compte grâce aux méthodes évoquées section 2.3.3) :

$$\min_{Q\in\mathcal{H}} \|TQ - Q\|_n^2 = \min_{Q\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}(r_i + \gamma \max_{a\in\mathcal{A}} Q(s_i', a) - Q(s_i, a_i))^2. \tag{3.13}$$

Par ailleurs, les données $\mathcal{D}_E$ nous indiquent, pour une collection d'états, quelle action doit maximiser la Q-fonction optimale :

$$\forall 1 \leq j \leq m,\ \forall a \neq a_j \quad Q(s_j, a_j) \geq Q(s_j, a). \tag{3.14}$$

Il est alors assez naturel d'ajouter les contraintes (3.14) au problème d'optimisation (3.13), en considérant de plus une marge $\mathcal{L}$ définie par l'utilisateur (telle que définie section 3.1.1, un choix canonique étant $\mathcal{L}(s_j, a) = \mathbb{I}_{\{a\neq a_j\}}$) et des variables d'ajustement qui permettent une violation partielle des contraintes. Le problème d'optimisation considéré est alors :

$$\begin{aligned} &\min_{Q\in\mathcal{H}} \|TQ - Q\|_n^2 + \frac{\alpha}{m}\sum_{j=1}^{m}\xi_j \\ &\text{sujet à } \max_{a\in\mathcal{A}} (Q(s_j, a) + \mathcal{L}(s_j, a)) - Q(s_j, a_j) \leq \xi_i. \end{aligned} \tag{3.15}$$

Avec $\alpha = 0$, on retrouve la minimisation du résidu, et avec $\alpha = \infty$ on retrouve le problème de classification correspondant au risque (3.3). Les cas intermédiaires correspondent à un compromis entre ces deux objectifs (optimiser la récompense et imiter l'expert), non contradictoires si la politique de l'expert optimise bien la récompense observée. Le problème (3.15) n'est ni convexe, ni différentiable partout. Il est proposé dans [170] de le résoudre grâce à une descente de gradient fonctionnel restreint (c'est-à-dire du boosting [142, 99]), ce qui donne un algorithme non-paramétrique.

La principale alternative lors de l'introduction de cette méthode était APID [112] (*approximate policy iteration with demonstration*). Le problème correspondant est le même que celui donné équation (3.15), en remplaçant l'opérateur d'optimalité T par l'opérateur d'évaluation T_π et en choisissant pour $\mathcal{L}$ la marge canonique, le tout étant placé dans le cadre d'itération de la politique [13]. Toutefois, chercher un point fixe de T_π n'est pas forcément consistant avec les contraintes imposées par l'expert. L'approche a l'avantage de poser un problème convexe, mais l'inconvénient d'être paramétrique. Empiriquement, sur les problèmes considérés, l'avantage va à la minimisation de résidu optimal contraint plutôt qu'à APID [170]. Plus récemment, il a été proposé dans [30] d'utiliser les démonstrations expertes

12. Pour être plus précis, c'est le cadre où l'apprentissage du contrôle optimal se fait de façon hors-ligne, conjointement à partir d'informations de récompense et de démonstrations expertes, qui est peu étudié. En élargissant ce cadre, d'autres méthodes considèrent disposer de ces deux types d'information. Par exemple, certaines méthodes utilisent les démonstrations expertes pour obtenir une politique initiale, améliorée par renforcement [103, 122]. D'autres supposent que l'apprenti peut (ponctuellement) interroger l'expert ou que l'expert peut corriger l'apprenti (voir par exemple [31]).

13. Notons qu'un résidu est considéré pour l'analyse, mais un point fixe projeté pour l'implémentation pratique, voir [112].

dans un schéma de type DPI, ce qui se fait très naturellement. En effet, rappelons (voir section 2.3) que DPI est un schéma d'itération de la politique approchée où l'étape d'amélioration de la politique est faite par un classifieur et l'étape d'évaluation par Monte Carlo. Dans ce cadre, ajouter des démonstrations expertes est très naturel (il suffit d'augmenter la base d'apprentissage du classifieur).

Chapitre 4

Applications aux interactions homme-machine

Nous présentons dans ce chapitre les contributions applicatives de l'apprentissage par renforcement (direct et inverse) auxquelles nous avons participé, centrées sur les interactions homme-machine (IHM). Elles ont en commun de pouvoir se formaliser comme un problème de contrôle optimal d'un système dynamique dans lequel c'est l'humain avec lequel la machine interagit qui définit la dynamique du système. Cette thématique de recherche est liée aux projets européens CLASSIC (pour le dialogue), ALLEGRO (pour le tutorat intelligent) et ILHAIRE (pour le rire) et a été l'objet principal des thèses co-encadrées par nos soins de Senthilkumar Chandramohan (2009-20012) et Lucie Daubigney (2010-2013), ainsi que de celle de Bilal Piot (dont les contributions théoriques, motivées par la problématique applicative, ont été présentées au chapitre 3) :

- Senthilkumar Chandramohan. *Revisiting User Simulation in Dialogue Systems: Do we still need them? Will imitation play the role of simulation?* Thèse de Doctorat en Informatique, Université d'Avignon, 2012 ;
- Lucie Daubigney. *Gestion de l'incertitude pour l'optmisation de systèmes interactifs.* Thèse de Doctorat en Informatique, Université de Lorraine, 2013 ;
- Bilal Piot. *Apprentissage hors-ligne avec Démonstrations Expertes.* Thèse de Doctorat en Informatique, Université de Lorraine, 2014.

Chaque section de ce chapitre sera dévolue à une application particulière, les systèmes de dialogue parlé pour la section 4.1, le tutorat intelligent pour la section 4.2 et l'introduction du rire dans les IHM pour la section 4.3.

4.1 Systèmes de dialogue parlé

Les systèmes de dialogue parlé [158, 161, 132] sont un medium d'interaction entre un humain et une machine par le biais de la parole, avec pour but (dans notre cas) l'accomplissement d'une tâche spécifique (demande d'informations touristiques ou réservation d'un billet par exemple). Le problème est, pour la machine, de conduire le dialogue en fonction des échanges (bruités) passés de façon à accomplir la tâche voulue.

4.1.1 Problématique

Pour réaliser un système de dialogue parlé (SDS pour *spoken dialogue system*), il n'est pas suffisant de combiner des modules de reconnaissance et de synthèse de la parole. D'une part, il faut leur adjoindre des modules de compréhension et de génération du langage naturel. Surtout, et particulièrement dans le cadre qui nous intéresse, il faut leur adjoindre un gestionnaire de l'interaction, qui est responsable de la gestion de l'échange d'information entre la machine et l'utilisateur, dans le but d'accomplir la tâche. Notamment, ce module est responsable des stratégies de confirmation (implicite ou explicite), nécessaires du fait du caractère imparfait des systèmes de reconnaissance et de compréhension de la parole.

Nous décrivons brièvement la modélisation d'un problème de gestion de dialogues sous la forme d'un PDM. Pour cela, nous considérons le cas particulier d'un système d'information touristique permettant de trouver un restaurant. L'objectif pour la machine est de proposer un restaurant à l'utilisateur en fonction des indications de ce dernier (localisation, prix, type de nourriture, etc.). En d'autres termes, il faut remplir un certain nombre de champs, en fonction de ce que demande l'utilisateur. L'état sera composé de ces champs et de la confiance associée en fonction de l'historique du dialogue [1]. Les actions permettent de conduire le dialogue (demande de la valeur d'un champ, demande de confirmation implicite ou explicite d'un champ, combinaison éventuelle de ces actions, proposer un restaurant à l'utilisateur). Généralement, la seule récompense informative est donnée lorsque le restaurant est proposé à l'utilisateur, et dépend de l'adéquation de cette proposition à la demande initiale. Enfin, la dynamique du système est le résultat du comportement de l'utilisateur.

Dans les travaux que nous présentons par la suite, nous avons travaillé avec deux gestionnaires de dialogue, Dipper [131] et HIS [217, 216] (la politique de contrôle, à apprendre, étant une composante du module de gestion du dialogue, la seule à laquelle nous nous intéressons).

4.1.2 Se passer de la simulation d'utilisateurs ?

On peut donc formaliser la gestion de dialogue comme un processus décisionnel de Markov, que l'on peut envisager de résoudre en utilisant une méthode d'apprentissage par renforcement. Toutefois, les SDS (et plus généralement les problèmes d'interaction homme-machine) posent le problème de l'acquisition et de l'annotation des données, dans la mesure où c'est l'humain qui définit la dynamique. Pour pallier ce problème de rareté et de coût des données (et pallier plus généralement le fait que la dynamique ne puisse être simulée à la demande, aspect très commode dans le cadre de l'apprentissage et de l'évaluation d'un contrôle), le domaine s'est orienté vers la simulation d'utilisateurs (génération artificielle de données par le biais de simulation de dialogues) [134, 159, 183]. Toutefois, simuler des utilisateurs induit de nouvelles sources d'erreurs de modélisation (en plus de celles des autres composantes du SDS) dont les effets restent assez peu connus [184] (voir tout de même [164] pour des éléments de réponse). Un axe de recherche a donc consisté à s'affranchir [2] de cette

1. L'état est construit par le SDS, et dépend du système considéré. Sa construction est souvent basée sur un paradigme appelé état d'information (IS pour *information state*) [128]. En effet, le problème est fondamentalement partiellement observable (ne serait-ce qu'à cause de la reconnaissance imparfaite). L'état construit par le gestionnaire de dialogue peut informellement être vu comme un état de croyance (ou *belief state*), classique dans le cadre des PDM partiellement observables [7, 106]. Toutefois, nous n'en tirons pas parti et traitons le problème comme un PDM. Notons que la construction de cet état est loin d'être triviale.

2. Pour des raisons pratiques, les expérimentations menées sont cependant faites en utilisant un simulateur d'utilisateur, même dans un cadre *off-policy*, ne serait-ce que pour évaluer plus facilement les politiques

simulation d'utilisateurs, et à en proposer des alternatives.

Pour s'affranchir de la simulation d'utilisateurs, nous avons proposé d'utiliser des méthodes de programmation dynamique approchée [3], hors-ligne, *off-policy* et efficaces en terme d'échantillons [163]. Ainsi, il est possible d'envisager l'apprentissage d'un bon contrôle à partir de (peu de) données collectées à l'avance, par exemple en utilisant un contrôleur sous-optimal mais acceptable ou un dispositif de type *wizard-of-oz* [110, 178]. Nous avons plus particulièrement proposé d'utiliser des algorithmes d'itération de la politique approchée [27] et d'itération de la valeur approchée [26], où la fonction de valeur sur les couples état-action apprise appartient à un espace de Hilbert à noyau reproduisant (ou RKHS pour *reproducing kernel Hilbert space*), et tel que le schéma d'apprentissage permette d'obtenir une représentation parcimonieuse de cette fonction (grâce à une méthode de dictionnaire inspirée de [51]). Les approches proposées sont donc non-paramétriques, elles permettent d'éviter de choisir a priori les fonctions de base représentant la valeur (le noyau doit tout de même être choisi), problème qui peut être difficile dans le cadre des systèmes de dialogue parlé.

Par ailleurs, nous avons également proposé de simuler les utilisateurs en utilisant l'apprentissage par renforcement inverse [21]. En effet, du point de vue de l'utilisateur, nous sommes également face à un problème de contrôle, que l'on peut formaliser comme un processus décisionnel de Markov, mais dont la récompense est inconnue. Cette récompense peut être apprise à partir de données d'interaction, puis être optimisée afin de fournir un simulateur d'utilisateur. D'autre part, il est commun de considérer un seul utilisateur générique dans le cadre de la simulation de dialogues, bien qu'il puisse y avoir plusieurs comportements distincts (par exemple, un utilisateur néophyte et un utilisateur averti ne se comporteront pas de la même manière). Nous avons donc proposé une méthode de quantification vectorielle des comportements d'utilisateurs [22], basée sur la notion d'attribut vectoriel moyen présentée section 3.1.2. Enfin, lorsqu'un système de dialogue parlé est déployé, les utilisateurs ont tendance à s'adapter au système, et donc à modifier leur comportement (et donc la dynamique du gestionnaire de dialogue), ce qui suggère que le gestionnaire de dialogue devrait également s'adapter, ce qui se traduit par une modification de la dynamique du point de vue de l'utilisateur, etc. Modéliser à la fois le gestionnaire de dialogue et l'utilisateur comme des processus décisionnels de Markov (la politique de l'un définissant la dynamique de l'autre) permet de s'intéresser à ce phénomène de co-adaptation [24].

4.1.3 Tenir compte des particularités du dialogue

Si la programmation dynamique approchée permet d'apprendre un contrôleur à partir d'un corpus de dialogues, on peut souhaiter continuer à améliorer le gestionnaire de dialogue, en ligne, une fois que la politique est mise en oeuvre. Cela pose donc le problème de l'apprentissage en ligne d'une politique de contrôle. Dans le cadre du dialogue, il est souhaitable de tenir compte de plusieurs aspects (généraux à l'apprentissage par renforcement, mais s'instanciant de façon particulière dans ce cadre). Tout d'abord, l'apprentissage doit être efficace en terme d'échantillons (apprendre une aussi bonne politique que possible avec le moins d'échantillons possible). Il se pose également classiquement le problème du dilemme entre exploration et exploitation. Ici, choisir une action exploratrice peut ennuyer l'utilisateur, il s'agit donc d'explorer prudemment [37] (et, comme nous l'avons mentionné

apprises.

3. Si l'approche est très naturelle, elle n'avait été que peu envisagée dans le domaine du dialogue, l'unique alternative à ce moment (à notre connaissance) étant [135].

section 2.2.1, disposer d'une information d'incertitude sur l'estimation courante de la valeur peut être utile). On peut également souhaiter avoir un apprentissage *off-policy* (ici, estimer la Q-fonction optimale, quelle que soit la politique de comportement). L'état sous-jacent à un gestionnaire de dialogue peut être de grande dimension, et l'on souhaiterait pouvoir représenter la fonction de valeur avec un réseau de neurones multi-couche, ce qui permet une représentation plus compacte [42]. Dans la mesure où l'utilisateur peut s'adapter au système, il est également souhaitable que l'algorithme d'apprentissage soit robuste aux non-stationnarités. Le cadre de travail de KTD, que nous avons présenté section 2.2.1, permet de prendre en compte tous ces aspects à la fois. Il a donc été appliqué au problème du dialogue [38] (à la fois sur DIPPER et HIS), avec succès (voir [38] pour l'étude expérimentale).

D'autre part, beaucoup d'applications des SDS sont naturellement parallèles (plusieurs utilisateurs appellent en même temps un centre d'appel, par exemple). Pour tirer profit de cet aspect, nous avons proposé d'optimiser la politique du gestionnaire de dialogue [45] en appliquant l'algorithme PSO [111] (*particle swarm optimization*), qui est un algorithme d'optimisation en boîte noire, à la recherche directe dans un espace de politique [57]. De façon générale, l'algorithme maintient une population de particules, chacune correspondant à une politique. Les particules sont évaluées, puis leurs positions évoluent en fonction du résultat de ces évaluations. Dans ce cadre, évaluer une particule revient à jouer un dialogue avec la politique associée (voir [45] pour les résultats expérimentaux). Un autre avantage est que ce type d'approche ne nécessite pas d'hypothèse markovienne, et peut donc s'appliquer au cas partiellement observable. Ce n'est généralement pas critique pour le dialogue, où un état markovien peut être construit, mais ce n'est pas le cas pour le tutorat intelligent.

4.2 Tutorat intelligent

Nous nous intéressons ici au tutorat automatique [4]. Ce sont des programmes informatiques qui ont vocation à enseigner une matière à un étudiant, comme les mathématiques [123], la physique [98], l'informatique [35] ou les langues étrangères [3], par exemple. Le tuteur a pour tâche de proposer une succession d'activités pédagogiques (leçon théorique, exercice, indice, évaluation, etc.) à l'étudiant afin d'accroître ses connaissances et compétences. C'est bien un problème de décisions séquentielles, que l'on peut envisager traiter par l'apprentissage par renforcement. Toutefois, l'état du système (notamment la connaissance de l'élève, mais aussi éventuellement d'autres aspects, comme son intérêt ou son ennui, ses capacités, ses préférences, etc.) n'est pas directement observable, et il est beaucoup plus difficile de construire un état de croyance (ou quelque chose qui y ressemble) que dans le cas du dialogue.

4.2.1 Problématique

La problématique que nous posons ici est l'apprentissage d'une bonne politique de contrôle à partir de logs d'interactions entre un étudiant et un tuteur (par exemple de type *wizard-of-oz*, ou conçu de façon non-automatique). C'est donc celle d'un apprentissage hors-ligne et *off-policy*. Pour cela, nous considérons une tâche simple, où le tuteur dispose de trois activités pour faire progresser l'élève : une leçon qui permet d'augmenter ses connaissances, une question qui permet d'estimer le savoir et un examen final qui clôt l'interaction. Comme pour le dialogue, nous nous plaçons toutefois dans un cadre simulé.

4. Pour un discussion plus complète sur l'intérêt et les enjeux des systèmes de tutorat intelligent, voir [36].

L'étudiant est simulé en utilisant le modèle probabiliste initialement proposé par [34], avec les méta-paramètres proposés par [29], calés sur des données réelles. L'état du système, non observable, est la probabilité que l'étudiant maîtrise le savoir, qui augmente lorsque une leçon est donnée. Lors d'une évaluation, l'étudiant aura un score d'autant meilleur que son savoir est important, mais il peut aussi répondre juste par hasard ou se tromper alors que le savoir est effectivement maîtrisé. Pour le tuteur, la récompense est toujours nulle, sauf lorsque l'examen final est posé, cas où la récompense est le score obtenu à l'examen par l'étudiant.

4.2.2 Observabilité partielle

Le problème est donc partiellement observable. L'état n'est pas accessible, le tuteur ne peut expérimenter que trois observations : une neutre (dans le cas d'une leçon), une correcte et une incorrecte (dans le cas d'une question). Comme preuve de concept [160] (d'un apprentissage hors-ligne et *off-policy* appliqué au problème de tutorat intelligent), nous avons d'abord construit un état pour le tuteur, supposé suffisant pour prendre de bonnes décisions. Cet état a deux composantes, la première correspondant à l'observation courante et la seconde à un compteur des leçons données. En appliquant ensuite LSPI à ce problème, nous obtenons des résultats corrects (meilleurs qu'une stratégie aléatoire et qu'une stratégie conçue de façon non-automatique). Toutefois, cette solution n'est pas satisfaisante, car très *ad hoc* à l'exemple considéré.

Nous avons donc proposé de combiner LSPI à un ESN (*echo state network*) [43]. Un ESN [104] est un réseau de neurones récurrent dont les connexions de la couche d'entrée vers la couche cachée et à l'intérieur de la couche cachée sont générées aléatoirement (en respectant toutefois certaines règles). Il est possible de montrer qu'avec forte probabilité (fonction notamment de la taille de la couche cachée), l'état interne de l'ESN (l'activation des neurones cachés) est markovien [204]. Le réseau est donc nourri avec une succession de couples observation-action, la Q-fonction associée est une combinaison linéaire des activations des neurones cachés (et ce sont donc ces poids qui sont appris dans le cadre de LSPI, soit les connexions de la couche cachée vers la couche de sortie). Sur le problème considéré, cette approche fournit de meilleurs résultats que celle décrite précédemment (prenant un compte un compteur de leçons).

Lorsqu'un ESN est utilisé, il y a un compromis à faire sur la taille de la couche cachée (également appelé réservoir dans ce cadre). On souhaite l'avoir aussi grande que possible, de façon à ce que l'état interne soit le plus markovien possible. Mais il ne faut pas qu'elle soit trop grande, pour éviter les problèmes de sur-apprentissage. Nous avons donc proposé de combiner ESN (avec un grand réservoir) et projection aléatoire [46] (une projection aléatoire [2] est une projection dont la matrice associée est générée aléatoirement, de façon à préserver le produit scalaire). Cette combinaison a été étudiée dans le cas supervisé, mais elle pourrait s'appliquer de même au cas du renforcement.

4.3 Rire

Nous nous intéressons ici à l'introduction du rire dans les interactions homme-machine. Un parallèle peut être fait avec le dialogue. A chaque instant, un module de reconnaissance va fournir à un organe de décision un ensemble d'indicateurs (de rire ou de sourire par exemple), basés sur une analyse multimodale du signal (typiquement audiovisuel, mais pro-

venant aussi de capteurs abdominaux ou d'une caméra Kinect, par exemple). En fonction de ces indicateurs, cet organe de décision, appelé gestionnaire de rire, doit choisir entre rire ou non, et quel type de rire (par exemple en distinguant plusieurs intensités). Une fois l'action choisie, elle est transmise à un module de synthèse de rire (qui combine synthèse vocale et synthèse visuelle). Il s'agit donc d'un problème de prise de décisions séquentielles pour lequel c'est l'humain qui définit la dynamique du système à contrôler. Toutefois, contrairement au dialogue, l'objectif de l'interaction (donc la récompense) est loin d'être évident.

4.3.1 Problématique

Nous avons considéré particulièrement deux scénarii. Pour le premier [152], un humain et un avatar informatique regardent une même vidéo amusante. L'état observé [5] par l'avatar contient deux indicateurs résultant de l'analyse du signal (un indicateur de rire et un indicateur d'intensité du rire), ainsi qu'une composante de contexte (qui indique ici si le passage de la vidéo est drôle ou non, ceci étant le résultat d'un annotation préalable). L'avatar doit donc décider de rire ou non selon le comportement de l'humain et le contenu (annoté) de la vidéo. Le deuxième scénario [141, 174, 173] correspond à un jeu du "ni oui, ni non", qui consiste à répondre à une série de questions sans prononcer les mots "oui" ou "non". L'avatar pose des questions [6] à un premier humain qui doit répondre sans utiliser les mots interdits, pendant qu'un second humain observe les deux autres protagonistes. L'état observé contient quatre indicateurs par humain (de rire, de sourire, de parole et d'intensité du rire) ainsi qu'un indicateur de contexte (état du jeu).

Dans les deux cas, il est difficile de quantifier l'objectif du contrôle, a fortiori la récompense. Toutefois, il est aisé d'obtenir un comportement "expert", en remplaçant l'avatar informatique par un humain et en annotant son comportement. Nous nous trouvons donc dans le cadre d'un apprentissage par imitation, couvert au chapitre 3.

4.3.2 Imitation

Nous sommes donc face à un problème de prises de décisions séquentielles, où la récompense est difficile à formaliser mais où une politique experte (l'humain) peut être observée. Il est donc naturel de penser à l'apprentissage par renforcement inverse, et plus généralement à l'apprentissage par imitation. Toutefois, nous sommes dans un cadre où l'apprentissage doit se faire à partir de logs d'interactions fournis a priori (et plus particulièrement d'interaction entre l'expert et le système) et où il est difficile de simuler le système (notamment dans la mesure où c'est l'humain avec lequel interagit l'expert qui définit la dynamique). Cette problématique a, en partie, motivé le développement des algorithmes de renforcement inverse présentés section 3.2, qui ne nécessitent pas de résoudre le problème direct pour estimer une récompense. Il reste nécessaire d'optimiser la récompense finale pour imiter l'expert, mais cela peut être fait en s'aidant également des traces de l'expert (comme exemples d'une politique optimale), ce qui a en partie motivé les travaux sur l'apprentissage par renforcement avec démonstrations expertes présentés section 3.3.3. Etant donné la nature des données et la difficulté de simuler, il peut sembler plus simple d'imiter l'expert en utilisant une approche d'apprentissage supervisé. Toutefois, nous avons vu section 3.3.1 que dans un contexte d'imitation d'un contrôle, prendre en compte la dynamique est important. Cela

5. La dénomination "état" peut être abusive, dans la mesure où nous ne sommes pas formellement assurés que la dynamique est markovienne, mais nous faisons comme si c'était le cas.

6. Le fait de poser des questions n'est pas géré par le gestionnaire de rire (qui ne s'occupe que du rire).

a motivé le fait de combiner simplicité d'une approche supervisée avec prise en compte de la dynamique comme en renforcement inverse, ce qui a été instancié par RCAL, présenté section 3.3.2. Pour plus de détails concernant les scénarii étudiés et les expérimentations faites, voir notamment [152, 141, 174, 173, 165].

Chapitre 5

Projet scientifique

Nos recherches s'inscrivent donc dans la thématique du contrôle optimal d'un système dynamique (d'une façon générale, l'agent ayant accès à des récompenses et/ou des démonstrations), sous le prisme de l'apprentissage automatique. Notre projet scientifique s'inscrit naturellement dans ce même cadre.

De façon très générale, notre objectif est de pouvoir appliquer (facilement) l'apprentissage par renforcement (direct ou inverse) à des problèmes réels, de façon sûre. Cela pose un certain nombre de problèmes, comme le passage à l'échelle, l'adaptabilité (au sens de fournir aussi peu d'expertise du domaine applicatif que possible), le dilemme entre exploration ou exploitation ou encore la sélection de modèle.

On peut faire ici un parallèle avec l'apprentissage supervisé. Lorsque l'on parle de "grand problème" en renforcement ou en supervisé, il y a généralement un fossé entre les échelles considérées (à l'avantage du supervisé). Cela n'est pas surprenant, l'apprentissage par renforcement est un problème plus difficile que l'apprentissage supervisé. Notamment, la régression est un cas particulier de l'estimation de valeur et la classification (dans sa forme la plus générale) est un cas particulier de la recherche locale de politique. De plus, d'autres problèmes s'ajoutent en renforcement, comme le dilemme entre exploration et exploitation et de fréquents problèmes de désaccords entre distributions (dont les effet sont encore mal compris, ou difficilement prévisibles).

Pour progresser, nous pensons qu'il est nécessaire de dépasser le cadre de la programmation dynamique approchée, socle d'une grande partie des approches actuelles. En effet, nous pensons que plutôt que d'introduire des erreurs dans des méthodes conçues pour le cas exact, il faudrait tenir compte du fait que l'on travaille dans un cadre approché plus en amont. Pour cela, l'apprentissage supervisé peut servir d'inspiration. Par exemple, la recherche locale de politique généralisant la classification, on pourrait penser à introduire des substituts convexes à la fonction objectif usuellement maximisée. Inversement, le domaine de l'apprentissage supervisé pourrait bénéficier du renforcement (par exemple, en s'inspirant de la recherche locale de politiques pour faire de nouvelles contributions au domaine de la classification).

Quelle que soit la méthode utilisée pour apprendre un contrôleur, si cela est fait à partir de transitions collectées a priori, il est important de pouvoir évaluer la qualité de cette politique avant de la mettre en application. C'est un problème de sélection ou d'évaluation de modèle, standard en apprentissage supervisé (une solution simple étant la validation croisée), mais où énormément reste à faire en apprentissage par renforcement. Nous pensons que c'est un axe prioritaire pour l'adoption de ce paradigme par l'industrie.

L'apprentissage par renforcement inverse pose le même genre de problèmes. Nous avons dépassé le cadre usuel d'appariement de trajectoires, mais beaucoup reste à faire, le domaine est encore jeune. Là encore, nous pensons qu'il y a beaucoup à gagner en cherchant à généraliser ou adapter des concepts d'apprentissage supervisé. Nous pensons que l'idée de régulariser un classifieur par la dynamique est un premier pas dans cette direction.

Les sections suivantes décrivent quelques perspectives plus factuelles, s'articulant autour des thématiques discutées dans ce manuscrit.

5.1 Apprentissage par renforcement

Concernant l'apprentissage par renforcement, notre projet scientifique s'articule autour de trois axes, l'estimation de la valeur (particulièrement non-paramétrique, et plus généralement la sélection de modèles), les schémas d'apprentissage (notamment les alternatives à la programmation dynamique approchée) et, d'une façon plus prospective, les interactions entre apprentissage par renforcement et apprentissage supervisé.

5.1.1 Estimation de la valeur

Nous avons discuté section 2.2 l'importance du problème de l'estimation d'une fonction de valeur, ainsi que les limites de l'approche paramétrique [1], ce qui a motivé nos premiers travaux sur des approches non-paramétriques de l'estimation de valeur. Nos contributions à ce domaine sont largement inspirées de la régularisation ℓ_1 en apprentissage supervisé, domaine qui propose bien d'autres approches non-paramétriques. Nous souhaitons donc poursuivre notre étude de l'estimation non-paramétrique de la valeur, que nous pensons être un axe de recherche important pour obtenir des machines les plus autonomes possibles.

Il existe de nombreuses pistes de recherche à ce sujet, dont certaines ont partiellement été abordées dans la littérature. Nous n'en ferons pas une liste exhaustive ici, mais nous en mentionnons quelques-unes. Il a été proposé de construire des modèles de PDM (modèles de transition et de récompense) non-paramétriques basés sur des estimateurs de Nadaraya-Watson [153, 137, 10] (*kernel smoothing*). Ce sont systématiquement des modèles locaux constants qui sont considérés, une extension à des modèles locaux linéaires ou polynômiaux pourrait être intéressante. Dans ces approches, le noyau est donné a priori, il serait intéressant d'étudier dans quelle mesure il pourrait également être appris. Une autre approche qui nous semble prometteuse est le plongement de distributions (conditionnelles) dans des RKHS [195] et son application au renforcement [100]. Là aussi, l'apprentissage du noyau est une perspective intéressante. Une autre approche que nous souhaitons étudier est l'application du boosting [182] vu comme une descente de gradient fonctionnelle [143, 99] à l'estimation de valeur. Le problème est qu'il peut être difficile d'exprimer un estimateur de la valeur comme solution d'un problème d'optimisation (a fortiori convexe). Ce qui est le plus naturel est de considérer une approche résiduelle, mais se pose alors le problème de biais. Une solution pourrait alors être d'utiliser (ici aussi) un estimateur de Nadaraya-Watson pour prendre en compte la stochasticité de la dynamique (voir [209]).

1. Il s'agit de trouver de "bonnes" fonctions de base, ce qui est très problème-dépendant. Dans le cadre de la conception d'une machine aussi autonome que possible, on souhaite apprendre la représentation de la fonction de valeur, en plus des valeurs des paramètres dans cette représentation. Toutefois, pour certains champs applicatifs, s'il existe de bonnes fonctions de base, résultat d'une ingénierie poussée, il est souhaitable de les utiliser.

Il y donc de nombreuses pistes à explorer, nous ne les discutons pas toutes. Toutefois, de façon complémentaire à l'estimation de valeur se pose le problème de l'évaluation de modèles. Supposons par exemple disposer d'un jeu de données et de deux jeux de fonctions de base. En appliquant un algorithme standard comme LSTD, nous pouvons obtenir deux estimations de la fonction de valeur. Une question très naturelle est de savoir quelle est la meilleure, ce à partir des données et sans recourir à un modèle génératif (qui pourrait fournir des estimations non-biaisées de la valeur par Monte Carlo). C'est un problème très commun en apprentissage supervisé, qui peut être traité par des méthodes classiques comme la validation croisée. La réponse est loin d'être aussi évidente en renforcement. A notre connaissance, peu d'articles s'intéressent au sujet [54, 210] (et nous proposons des heuristiques pour le choix du paramètre de régularisation de Dantzig-LSTD [94]). Ce problème nous semble fondamental, tant pour l'apprentissage de la structure de la valeur (une question préalable étant la comparaison de deux structures) que plus généralement (par exemple, pour la valeur quantifiant une politique, afin de comparer plusieurs contrôleurs a priori, sans les appliquer au système réel).

5.1.2 Schémas d'apprentissage

Nous avons vu section 2.3 que beaucoup d'approches d'apprentissage du contrôle optimal sont dérivées de la programmation dynamique. Des approximations de différentes natures sont introduites dans des algorithmes conçus pour le cas exact, et l'étude de l'effet de ces approximations est faite a posteriori. Informellement, si un schéma d'itération de la politique (ou de la valeur) approchée fait une erreur ϵ à chaque itération, la politique obtenue sera optimale à $\frac{2\gamma}{(1-\gamma)^2}\epsilon$ près. La dépendance à l'horizon moyen (le terme $\frac{1}{1-\gamma}$) est fine (voir [192], où il est proposé d'utiliser des politiques non-stationnaires pour améliorer cette dépendance). L'idée serait donc de tenir compte a priori de ces approximations (ou plus généralement d'envisager des alternatives à la programmation dynamique approchée), pour obtenir des garanties plus fortes (et donc potentiellement de meilleures performances empiriques). Voici deux approches possibles (encore une fois, elles n'ont rien d'exhaustif).

Nous avons discuté section 2.3.3 la possibilité d'exprimer la minimisation du résidu de Bellman pour l'opérateur d'optimalité comme un problème de différences convexes. Cela permet d'avoir une dépendance linéaire en l'horizon, plutôt que quadratique (voir le tableau 2.4). Les travaux sur ce sujet (du renforcement sous le prisme de la programmation par différences convexes) sont encore très préliminaires, il reste beaucoup de choses à faire (considérer des décomposition alternatives, des algorithmes de résolution alternatifs, étudier plus finement les garanties offertes, étudier d'autres façons de prendre en compte le biais, etc.). C'est donc un thème que nous souhaitons continuer à étudier.

Nous avons étudié section 2.3.2 les garanties offertes par l'alternative à la programmation dynamique approchée qu'est la recherche directe dans un espace de politiques (sous l'hypothèse très contraignante d'un espace de politiques convexe). En utilisant les mêmes notations et la même technique de preuve, on peut montrer que [189] :

$$\pi \in \mathcal{G}_\Pi(\pi, \nu, \epsilon) \Rightarrow \forall \pi',\ \forall \mu \in \Delta_\mathcal{S},\ \mu v_{\pi'} \leq \mu v_\pi + \frac{1}{1-\gamma} \left\| \frac{d_{\mu,\pi'}}{\nu} \right\|_\infty (\mathcal{E}_\nu(\Pi) + \epsilon).$$

On gagne donc un facteur d'horizon par rapport à une recherche directe dans un espace de politique (qui revient à chercher $\pi \in \mathcal{G}_\Pi(\pi, d_{\nu,\pi}, \epsilon)$). La question est de savoir s'il existe des schémas algorithmiques efficaces pour résoudre $\pi \in \mathcal{G}_\Pi(\pi, \nu, \epsilon)$, c'est-à-dire pour trouver

un point fixe de l'opérateur de gloutonnerie approchée définit Eq. (2.5). Nous souhaitons étudier ce problème ouvert.

Enfin, une autre thématique que nous souhaitons poursuivre, plus prospective, est celle de l'apprentissage *off-policy*. En effet, d'un point de vue applicatif, il est fréquent que les données disponibles soient imposées, car coûteuses ou difficiles à produire, et qu'elles soient générées selon un contrôleur sous-optimal, qui de plus n'explore pas tout l'espace d'état. A partir de ces données, il s'agit d'estimer le contrôleur optimal, ce malgré le désaccord entre la distribution sous-jacente aux données et celle dont on aurait besoin (liée à la politique optimale) pour faire une estimation correcte. La plupart des approches existantes se concentrent sur la proposition d'algorithmes qui ne divergent pas, sans avoir de garanties très fines sur ce vers quoi ils convergent. Très peu d'approches tentent de corriger ce désaccord entre distributions [124, 199]. Cela nous semble être une piste de recherche importante.

5.1.3 Interactions entre apprentissages par renforcement et supervisé

Il existe de nombreuses interactions entre apprentissage par renforcement et apprentissage supervisé. D'une part, le renforcement peut se réduire à une succession de problèmes supervisés [2] : par exemple, l'itération de la valeur approchée se réduit à une succession de problèmes de régression [97, 52] et l'itération de la politique peut se réduire à une succession de problèmes de classification [129]. Ce sont plus particulièrement des problèmes de classification multi-classe à coût sensitif (et bruité), réputés difficiles en apprentissage supervisé, ce qui a motivé nos récents travaux sur le sujet [65]. D'autre part, des aspects du renforcement peuvent être vus comme des généralisation de problèmes supervisés. Par exemple, il est connu que l'estimation de valeur se réduit à un problème de régression (de la récompense) lorsque le facteur d'actualisation γ est nul. De même, la fonction objectif (2.4) maximisée lors de la recherche directe dans un espace de politique se réduit à un problème de classification multi-classe à coût sensitif (ce constat ayant inspiré nos travaux récents en classification [65]). Nous pensons qu'étudier ces connexions [3] permettra de contribuer à la fois aux domaines du renforcement et de l'apprentissage supervisé.

5.2 Apprentissage par imitation

Concernant l'apprentissage par imitation, notre projet scientifique s'articule autour de trois axes : l'apprentissage par renforcement inverse, tenir compte de la dynamique dans les approches supervisées, ainsi que l'apprentissage par renforcement avec démonstrations expertes.

5.2.1 Apprentissage par renforcement inverse

Nous avons présenté sections 3.2.1 et 3.2.2 des algorithmes d'apprentissage par renforcement inverse qui permettent d'estimer une récompense optimisée par l'expert, de façon

2. C'est également le cas pour le renforcement inverse, comme nous l'avons expliqué section 3.2.

3. Par exemple, chercher un élément de $\mathcal{G}_\Pi(\pi, \nu, \epsilon)$ est proche d'un problème de classification multi-classe à coût sensitif, ce qui peut suggérer des approches pour résoudre le problème posé section précédente. Une autre piste intéressante est la connexion entre recherche de politique et classification multi-classe. En classification, il est courant de considérer un proxy convexe au risque d'intérêt, l'idée pourrait être étendue à l'objectif (2.4) (l'extension étant toutefois loin d'être triviale, surtout si l'on cherche la convexité).

hors-ligne (sans recours à un modèle génératif) et sans résoudre le problème direct (optimisation de récompenses intermédiaires arbitraires). Ces approches ont été inspirées par le lien entre classification et renforcement (fonction de score et Q-fonction, règle de décision et politique gloutonne). Elles ont été abstraites par le cadre des politiques d'ensembles présenté section 3.2.3, qui montre notamment que les solutions de l'apprentissage par renforcement inverse et de la classification à base de fonctions de score sont en bijection via l'opérateur de Bellman inverse. Ce cadre général ouvre de nombreuses perspectives, tant algorithmiques que théoriques, dont nous souhaitons poursuivre l'étude. Nous avons ainsi par exemple récemment contribué au premier algorithme d'apprentissage par renforcement inverse dans des domaines relationnels [147].

5.2.2 Régularisation par la dynamique

Dans le cadre de l'apprentissage par imitation, le renforcement inverse a pour avantage de prendre en compte la dynamique (dans un contexte de contrôle, c'est plus l'effet de l'action que l'action elle-même qui est important), tandis que les approches supervisées ont pour avantage leur simplicité [4]. Nous avons présenté section 3.3.2 l'algorithme RCAL, qui est un classifieur dont l'apprentissage tient compte de la dynamique. Toutefois, c'est une instantiation particulière (par le choix du proxy convexe pour la classification et de la pénalisation de la complexité de la récompense associée) d'une idée bien plus générale résumée par l'Eq. (3.12), que nous rappelons :

$$\mathcal{J}_{n,\alpha}(q) = R_n(q) + \alpha\Omega(J_*q).$$

Il s'agit donc de régulariser n'importe quel risque (relatif à un classifieur basé sur une fonction de score) par un terme qui pénalise la complexité de la récompense associée (image de la fonction de score par l'opérateur de Bellman inverse). Nous souhaitons étudier les garanties théoriques que peut offrir un tel schéma d'apprentissage (de façon générale ou plus instanciée), ainsi qu'en déduire des algorithmes alternatifs à RCAL. Nous souhaitons également étudier plus généralement si cette idée peut s'appliquer à d'autres types d'apprentissage (en régularisant un risque non pas par la pénalisation de la quantité d'intérêt, mais par la pénalisation d'une quantité qui lui est liée, choisie judicieusement).

5.2.3 Renforcement avec démonstrations expertes

Nous avons présenté la problématique de l'apprentissage par renforcement avec démonstrations expertes en section 3.3.3, où l'on dispose à la fois d'un signal de récompense et d'exemples de choix d'actions optimales. Ce domaine est très jeune, à notre connaissance peu d'articles ont été publiés sur ce sujet [5] [112, 170, 30], alors que de nombreux domaines d'application peuvent être envisagés (où à la fois une récompense et des démonstrations, au moins localement, de bonnes politiques sont disponibles, les données étant imposées a priori et l'apprentissage devant se faire hors-ligne et *off-policy*). Il reste donc beaucoup de choses à faire, tant du point de vue théorique (notamment étudier à quel point avoir les deux

4. Ceci ne veut pas forcément dire que ces approches sont simples, mais elles ne nécessitent pas d'optimiser une récompense pour imiter l'expert, et on peut se reposer sur une très large littérature sur le sujet.

5. Ceci peut être nuancé. Comme nous l'avons dit section 3.3.3, si l'on élargit le cadre, d'autres approches utilisent à la fois une information de récompense et l'avis d'un expert, par exemple non conjointement [103, 122] ou de façon plus interactive [31].

types d'information améliore l'estimation) que pratique. L'information donnée par politique optimale a été combinée avec l'itération de la politique [112] ainsi qu'avec la minimisation d'un résidu de Bellman pour l'opérateur d'optimalité [170], comme présenté section 3.3.3. Concernant cette contribution, on pourrait envisager de formuler le problème d'optimisation associé comme un problème de différences convexes. Plus généralement, on pourrait étendre d'autres approches de programmation dynamique approchée pour prendre en compte l'information experte. Rappelons que cette information est exprimée en tant que contraintes sur la Q-fonction (c'est-à-dire, $Q(s, \pi_E(s)) > Q(s, a \neq \pi_E(s))$). Ces contraintes pourraient très naturellement s'intégrer dans un approche de programmation linéaire approchée pour estimer la valeur optimale (voir section 2.1.2 pour l'estimation de valeur par programmation linéaire et par exemple [47, 157, 209] pour le cas approché). Elles pourraient également s'intégrer à une approche de type itération de la valeur (en contraignant les problèmes de régression intermédiaires). On pourrait également envisager d'intégrer cette information fournie par un expert sous une forme différente (que des contraintes sur les Q-valeurs). Pour résumer, cette thématique très jeune ouvre de nombreuses perspectives, beaucoup reste à faire, et nous souhaitons continuer à y contribuer. Nous souhaitons également étudier comment le cadre considéré, assez restrictif, s'intègre plus généralement dans les approches qui considèrent qu'à la fois un signal de récompense et un signal expert (qui peut être utilisé dans une phase d'initialisation ou requis ponctuellement de façon interactive, par exemple) sont disponibles.

5.3 Applications

Le champ d'application de nos travaux a été principalement les interactions homme-machine, comme discuté au chapitre 4. C'est un domaine d'application particulièrement intéressant, par les problèmes qu'il pose, du fait de la présence de l'humain dans la boucle (humain qui définit la dynamique du système à contrôler, comme nous l'avons déjà dit). C'est également un domaine où les données sont difficiles à récolter et coûteuses (ce qui explique que beaucoup de nos contributions sont faites sur des systèmes simulés). Nous avons travaillé sur d'autres champs applicatifs, dans le cadre de collaborations académiques, de contrats industriels ou de projets étudiants, non discutés dans ce manuscrit (pour des raisons de confidentialité pour certains) : sidérurgie [63], biomédical [155, 154], médiamétrie, instrumentation médicale ou encore robotique.

Dans le futur, nous souhaitons continuer à porter une attention particulière aux problèmes applicatifs. D'une part, des problèmes pratiques peuvent motiver des développements théoriques, et donc nourrir notre recherche plus fondamentale. Inversement, nous pensons également intéressant de chercher de nouveaux débouchés pratiques à des développements non motivés par l'applicatif.

Annexe A

Notice des titres et travaux

A.1 Formation

2006 - 2009 ▷ Doctorat en Mathématiques de l'Université Paul Verlaine de Metz, "Optimisation des chaînes de production dans l'industrie sidérurgique : une approche statistique de l'apprentissage par renforcement".
Rapporteurs : Oliver Sigaud (Professeur à l'UPMC) et Rémi Munos (Directeur de Recherche INRIA).
Direction : Gabriel Fricout, encadrant (ingénieur de recherche chez ArcelorMittal), Olivier Pietquin, co-directeur (enseignant-chercheur à Supélec) et Jean-Claude Vivalda, directeur (Directeur de Recherche INRIA).
Thèse CIFRE ArcelorMittal - Supélec - INRIA soutenue le 9 novembre 2009 à Metz.

2006 ▷ Master Recherche en Mathématiques appliquées de Supélec,
en cohabilitation avec l'université Paul Verlaine de Metz, major.

2003 - 2006 ▷ Diplôme d'ingénieur de Supélec (Ecole Supérieure d'Électricité),
option SIF (Signaux, Images et Formes), major.

Langues ▷ Français, langue maternelle.
Anglais, pratique professionnelle.
Allemand, notions.

A.2 Expérience professionnelle

Depuis 2015 ▷ CentraleSupélec (fusion Centrale Paris et Supélec), campus de Metz
Enseignant-chercheur contractuel au sein de l'équipe IMS-MaLIS
Membre associé de l'UMI 2958 (GeorgiaTech-CNRS)

2010-2014 ▷ Supélec, campus de Metz (école d'ingénieurs privée)
Enseignant-chercheur au sein de l'équipe IMS (Information, Multimodalité et Signal), dans le groupe MaLIS (*Machine Learning and Interactive Systems*)
Depuis fin 2013, également membre associé de l'UMI 2958 (GeorgiaTech-CNRS)

2006-2009 ▷ ArcelorMittal
Ingénieur CIFRE (thèse en co-tutelle avec Supélec et l'INRIA)

2006-2009 ▷ Supélec
Vacataire (Probabilités, Signaux et Systèmes, Traitement Statistique du Signal)

A.3 Enseignement

Proba. ▷ Depuis 2012, en charge du cours magistral (18h/an) "Probabilités"
Depuis 2006, travaux dirigés (12h/an) (vacation puis enseignant-chercheur)

App. Stat. ▷ Depuis 2010, en charge du cours magistral (24h/an) "Apprentissage Statistique"
Support en ligne :
`http://www.metz.supelec.fr/~geist_mat/pdfs/poly_as_v2.pdf`

Gest. Inc. ▷ Depuis 2013, en charge d'une partie du cours magistral (6h/an)
(cours commun avec GoergiaTech Lorraine, dispensé en anglais)
Depuis 2010, travaux dirigés (9,5h/an) du cours "Gestion de l'incertain"
TL de *machine learning*

Math. Ing. ▷ Depuis 2013, en charge d'une partie du cours magistral (9h/an) du cours "Mathématiques pour l'ingénieur"

Sign. Syst. ▷ Depuis 2014, travaux dirigés (6h/an) et pratiques (36h/an) du cours "Signaux et Systèmes"

RASS ▷ Depuis 2014, travaux dirigés (12h/an) et pratiques (36h/an) du cours "Représentation et Analyse Statistique des Signaux"

Stats. ▷ De 2010 à 2012, en charge des travaux dirigés (6h/an) du cours "Statistiques"

Trait. Signal ▷ De 2010 à 2012, en charge des travaux dirigés (6h/an) et pratiques (36h/an) du cours "Traitement du Signal"
Examinateur à l'examen (oral)

Stat. Sign. ▷ De 2006 à 2009, travaux dirigés (12h/an) du cours "Traitement Statistique du Signal"

Sign. Syst. ▷ De 2006 à 2008, travaux dirigés (6h/an) et pratiques (45h/an) du cours "Signaux et Systèmes"

CEI ▷ Convention d'Étude Industrielle/Interne, pour les étudiants de troisième année
Volume de 120h pour l'étudiant

Proj. Concept. ▷ Projets de conception, pour les étudiants de deuxième année
Volume de 36h pour l'étudiant

Proj. Info. ▷ Projets de développement logiciel, pour les étudiants de deuxième année (axés informatique et programmation)
Volume de 36h pour l'étudiant

Proj. Synthèse ▷ Projets de synthèse, pour les étudiants de première année
Volume de 36h pour l'étudiant

Le tableau suivant résume de façon plus quantitative les enseignements effectués jusqu'à présent. Je distingue cours magistraux (CM), travaux dirigés (TD), travaux pratiques (TP) et projets étudiants (Proj.). Le total est exprimé en équivalent-TD (1 CM = 1,5 TD = 1,5 TP). La partie basse du tableau correspond à la période de vacation, la haute à la période enseignant-chercheur.

année	CM	TD	TP	Proj.	Total
2014 - 2015	60h	31,5h	72h	30h	223,5h
2013 - 2014	57h	21,5h		30h	137h
2012 - 2013	42h	20h		30h	113h
2011 - 2012	24h	26h	36h	30h	128h
2010 - 2011	24h	26h	36h	30h	128h
2008 - 2009		18h			18h
2007 - 2008		24h	45h		69h
2006 - 2007		24h	45h		69h

A.4 Encadrement de thèses

J'ai co-encadré jusqu'à présent quatre thèses de doctorat (toutes en informatique). Mon taux d'encadrement effectif est d'environ 50 % pour chacune de ces thèses.

2011-2014 ▷ **Bilal Piot**
"Apprentissage par imitation et transfert de tâches pour des interactions homme-machine naturelles"
UMI 2958 (CNRS - GeorgiaTech)
Thèse co-dirigée par moi-même et dirigée par Olivier Pietquin (maintenant Professeur à l'université de Lille)

2010-2013 ▷ **Edouard Klein**
"Contributions à l'apprentissage par renforcement inverse"
Équipe ABC du LORIA
Thèse co-encadrée par Olivier Pietquin (alors enseignant-chercheur à Supélec), co-dirigée par moi-même et dirigée par Yann Guermeur (Directeur de Recherche au CNRS)

2010-2013 ▷ **Lucie Daubigney**
"Gestion de l'incertitude pour l'optimisation de systèmes interactifs"
Équipe MAIA du LORIA
Thèse co-encadrée par moi-même, co-dirigée par Olivier Pietquin (alors enseignant-chercheur à Supélec) et dirigée par Alain Dutech (Chargé de Recherche à INRIA)

2009 - 2012 ▷ **Senthilkumar Chandramohan**
"Revisiting User Simulation in Dialogue Systems : Do we still need them ? Will imitation play the role of simulation ?"
Laboratoire d'Informatique d'Avignon
Thèse co-encadrée par moi-même, co-dirigée par Olivier Pietquin (alors enseignant-chercheur à Supélec) et dirigée par Fabrice Lefevre (Professeur à l'université d'Avignon)

A.5 Projects collaboratifs

ILHAIRE ▷ "Incorporating laughter into human-avatar interactions : Research and evaluation"
Projet européen - ICT FET Open (2012-2014)
Université de Mons, CNRS, Universität Augsburg, Universita degli Studi di Genova, University College London, Queen's University Belfast, Universität Zürich, Supélec, La Cantoche Production
Étudier le rôle du rire dans les interactions entre humains et systèmes informatiques, et développer de nouveaux paradigmes pour une interaction homme-machine plus naturelle, y compris avec des avatars anthropomorphes, amenés à jouer un rôle important dans l'avenir des médias numériques.

METHODEO ▷ "Test methodology and metrics definition to benchmark algorithms for videoprotection"
Projet ANR - programme CSOSG (2011-2013)
Thalès, IRIT, CEA, Telecom Sud Paris, Supélec, Keeneo
Développement de méthodologies et de métriques pour évaluer des algorithmes et des bases de données dédiés à la vidéosurveillance.

ALLEGRO ▷ "Adaptive Language LEarning technology for the Greater Region"
Projet européen FEDER - INTERREG (2010-2012)
Université de la Sarre, INRIA Nancy Grand Est, Supélec, DFKI
Développement et mise à disposition de technologies avancées d'enseignement par le web (e-Learning) ayant pour finalité d'accompagner l'accroissement de la multilingualité dans la Grande Région.

CLASSIC ▷ "Computational Learning in Adaptive Systems for Spoken Conversation"
Projet européen ICT STREP (2009-2011)
Heriott Watt University,Université d'Edinburgh, Université de Cambridge, Université de Genève, France Télécom, Supélec
Optimisation de bout en bout d'un système de dialogue homme-machine à partir de données et de méthodes statistiques, gestion et propagation des incertitudes introduites par chacun des modules.

A.6 Animation scientifique

Relecture revues ▷ Journal of Machine Learning Research (JMLR)
Journal of Artificial Intelligence Research (JAIR)
IEEE Trans. on Neural Networks and Learning Systems (IEEE TNNLS)
IEEE Trans. on Systems, Man and Cybernetics (IEEE TSMC)
Computer & Industrial Engineering (Elsevier)
IEEE Journal of Selected Topics in Signal Processing (IEEE JSTSP)
ACM Trans. on Speech and Language Processing (ACM TSLP)
Elsevier Computers & Industrial Engineering (CAIE)
Revue d'Intelligence Artificielle (RIA)

Relecture conf. ▷ Neural Information Processing Systems (NIPS)

International Joint Conference on Artificial Intelligence (IJCAI)
European Conference on Machine Learning (ECML)
Conference on Uncertainty in Artificial Intelligence (UAI)
European Workshop on Reinforcement Learning (EWRL)
IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)
Conférence sur l'APprentissage automatique (CAP)
Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA)

Organisation ▷ Membre du comité d'organisation de l'école d'été eNTERFACE 2012 `http://enterface12.metz.supelec.fr/`
Membre du comité d'organisation de la conférence SIGdial 2013 `http://www.sigdial.org/workshops/conference14/`

A.7 Production logicielle

Langages ▷ Matlab, Python, C/C++, Java
(OS Linux ou Windows)

rl ▷ Collaboration sur la librairie `rl`, utilisant la programmation générique pour l'apprentissage par renforcement
(voir section "Software" du site `http://malis.metz.supelec.fr/`)

gaml ▷ Participation à la définition des concepts mathématiques de la librairie `gaml`, utilisant la programmation générique pour l'apprentissage machine
(voir section "Software" du site `http://malis.metz.supelec.fr/`)

A.8 Liste des publications

Mémoires

1. Matthieu Geist. *Optimisation des chaînes de production dans l'industrie sidérurgique : une approche statistique de l'apprentissage par renforcement.* Doctorat en Mathématiques, Université Paul Verlaine de Metz, Novembre 2009
2. Matthieu Geist. Modélisation de chaînes de production et de leurs interactions. Master's thesis, Supélec (M2R Mathématiques), Septembre 2006
3. Matthieu Geist. Analyse des données pour l'analyse, le suivi et le contrôle des dispersions. Master's thesis, Supélec, Septembre 2006

Articles de journaux internationaux

1. Matthieu Geist. Soft-max boosting. *Machine Learning*, 2015
2. Bruno Scherrer, Mohammad Ghavamzadeh, Victor Gabillon, Boris Lesner, et Matthieu Geist. Approximate Modified Policy Iteration and its Application to the Game of Tetris. *Journal of Machine Learning Research*, 2015

3. Matthieu Geist et Bruno Scherrer. Off-policy Learning with Eligibility Traces: A Survey. *Journal of Machine Learning Research (JMLR)*, 15:289–333, 2014
4. Matthieu Geist et Olivier Pietquin. An Algorithmic Survey of Parametric Value Function Approximation. *IEEE Transactions on Neural Networks and Learning Systems*, 24(6):845–867, 2013
5. Hervé Frezza-Buet et Matthieu Geist. A C++ Template-Based Reinforcement Learning Library: Fitting the Code to the Mathematics. *Journal of Machine Learning Research*, 14:399–402, 2013
6. Lucie Daubigney, Matthieu Geist, Senthilkumar Chandramohan, et Olivier Pietquin. A Comprehensive Reinforcement Learning Framework for Dialogue Management Optimisation. *IEEE Journal of Selected Topics in Signal Processing*, 6(8):891–902, 2012
7. Olivier Pietquin, Matthieu Geist, Senthilkumar Chandramohan, et Hervé Frezza-Buet. Sample-Efficient Batch Reinforcement Learning for Dialogue Management Optimization. *ACM Transactions on Speech and Language Processing*, 7(3), 2011
8. Matthieu Geist et Olivier Pietquin. Kalman Temporal Differences. *Journal of Artificial Intelligence Research (JAIR)*, 39:483–532, 2010
9. Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. From Supervised to Reinforcement Learning: a Kernel-based Bayesian Filtering Framework. *International Journal On Advances in Software*, 2(1):101–116, 2009

Articles de conférences internationales

1. Matthieu Geist. A multiplicative UCB strategy for Gamma rewards. In *European Workshop on Reinforcement Learning (EWRL)*, 2015
2. Bilal Piot, Matthieu Geist, et Olivier Pietquin. Imitation Learning Applied to Embodied Conversational Agents. In *Machine Learning and Interactive Systems (MLIS)*, 2015
3. Thibaut Munzer, Bilal Piot, Matthieu Geist, Olivier Pietquin, et Manuel Lopes. Inverse Reinforcement Learning in Relational Domains. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2015
4. Bilal Piot, Matthieu Geist, et Olivier Pietquin. Difference of Convex Functions Programming for Reinforcement Learning. In *Advances in Neural Information Processing Systems (NIPS 2014)*, 2014
5. Bilal Piot, Olivier Pietquin, et Matthieu Geist. Predicting when to laugh with structured classification. In *Annual Conference of the International Speech Communication Association (InterSpeech)*, 2014
6. Bruno Scherrer et Matthieu Geist. Local Policy Search in a Convex Space and Conservative Policy Iteration as Boosted Policy Search. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, 2014
7. Bilal Piot, Matthieu Geist, et Olivier Pietquin. Boosted Bellman Residual Minimization Handling Expert Demonstrations. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*. Springer, 2014

8. Bilal Piot, Matthieu Geist, et Olivier Pietquin. Boosted and Reward-regularized Classification for Apprenticeship Learning. In *13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014)*, Paris, France, 2014
9. Lucie Daubigney, Matthieu Geist, et Olivier Pietquin. Model-free POMDP optimisation of tutoring systems with echo-state networks. In *Proceedings of the 14th SIGDial Meeting on Discourse and Dialogue (SIGDial 2013)*, pages 102–106, 2013
10. Matthieu Geist, Edouard Klein, Bilal Piot, Yann Guermeur, et Olivier Pietquin. Around Inverse Reinforcement Learning and Score-based Classification. In *1st Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM 2013)*, 2013
11. Bilal Piot, Matthieu Geist, et Olivier Pietquin. Learning from demonstrations: Is it worth estimating a reward function? In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, et Filip Zelezny, editors, *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2013)*, volume 8188 of *Lecture Notes in Computer Science*, pages 17–32. Springer, 2013
12. Edouard Klein, Bilal Piot, Matthieu Geist, et Olivier Pietquin. A cascaded supervised learning approach to inverse reinforcement learning. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, et Filip Zelezny, editors, *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2013)*, volume 8188 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2013
13. Lucie Daubigney, Matthieu Geist, et Olivier Pietquin. Particle Swarm Optimisation of Spoken Dialogue System Strategies. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association (Interspeech 2013)*, 2013
14. Radoslaw Niewiadomski, Jennifer Hofmann, Jérome Urbain, Tracey Platt, Johannes Wagner, Bilal Piot, Huseyin Cakmak, Sathish Pammi, Tobias Baur, Stéphane Dupont, Matthieu Geist, Florian Lingenfelser, Gary McKeown, Olivier Pietquin, et Willibald Ruch. Laugh-aware virtual agent and its impact on user amusement . In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, 2013
15. Lucie Daubigney, Matthieu Geist, et Olivier Pietquin. Random Projetctions: a Remedy for Overfitting Issues in Time Series Prediction with Echo State Networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, 2013
16. Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefèvre, et Olivier Pietquin. Co-adaptation in Spoken Dialogue Systems. In *International Workshop on Spoken Dialog Systems (IWSDS 2012)*, 2012
17. Edouard Klein, Matthieu Geist, Bilal Piot, et Olivier Pietquin. Inverse Reinforcement Learning through Structured Classification. In *Advances in Neural Information Processing Systems (NIPS 2012)*, 2012
18. Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefèvre, et Olivier Pietquin. Behavior Specific User Simulation in Spoken Dialogue Systems. In *ITG Conference on Speech Communication*, 2012
19. Bruno Scherrer, Victor Gabillon, Mohammad Ghavamzadeh, et Matthieu Geist. Approximate Modified Policy Iteration. In *International Conference on Machine Learning (ICML)*, 2012

20. Matthieu Geist, Bruno Scherrer, Alessandro Lazaric, et Mohammad Ghavamzadeh. A Dantzig Selector Approach to Temporal Difference Learning. In *International Conference on Machine Learning (ICML)*, 2012

21. Julien Oster, Matthieu Geist, Olivier Pietquin, et Gary Clifford. Filtering of pathological ventricular rhythms during MRI scanning. In *International Workshop on Biosignal Interpretation*, 2012

22. Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefèvre, et Olivier Pietquin. Clustering Behaviors Of Spoken Dialogue Systems Users. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)*, 2012

23. Lucie Daubigney, Matthieu Geist, et Olivier Pietquin. Off-policy Learning in Large-scale POMDP-based Dialogue Systems. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)*, pages 4989 – 4992, 2012

24. Jérémy Fix et Matthieu Geist. Monte-Carlo Swarm Policy Search. In *Symposium on Swarm Intelligence and Differential Evolution*, Lecture Notes in Artificial Intelligence (LNAI). Springer Verlag - Heidelberg Berlin, 2012

25. Matthieu Geist et Olivier Pietquin. Kalman filtering & colored noises: the (autoregressive) moving-average case. In *IEEE Workshop on Machine Learning Algorithms, Systems and Applications (MLASA 2011)*, 2011

26. Edouard Klein, Matthieu Geist, et Olivier Pietquin. Reducing the dimentionality of the reward space in the Inverse Reinforcement Learning problem. In *IEEE Workshop on Machine Learning Algorithms, Systems and Applications (MLASA 2011)*, 2011

27. Hadrien Glaude, Fadi Akrimi, Matthieu Geist, et Olivier Pietquin. A Non-Parametric Approach to Approximate Dynamic Programming. In *IEEE International Conference on Machine Learning and Applications (ICMLA 2011)*, pages 317–322, 2011

28. Olivier Pietquin, Lucie Daubigney, et Matthieu Geist. Optimization of a Tutoring System from a Fixed Set of Data. In *ISCA workshop on Speech and Language Technology in Education (SLaTE 2011)*, 2011

29. Lucie Daubigney, Milica Gasic, Senthilkumar Chandramohan, Matthieu Geist, Olivier Pietquin, et Steve Young. Uncertainty management for on-line optimisation of a POMDP-based large-scale spoken dialogue system. In *Annual Conference of the International Speech Communication Association (Interspeech 2011)*, pages 1301–1304, 2011

30. Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefèvre, et Olivier Pietquin. User Simulation in Dialogue Systems using Inverse Reinforcement Learning. In *Annual Conference of the International Speech Communication Association (Interspeech 2011)*, pages 1025–1028, 2011

31. Rémi Chou, Yvo Boers, Martin Podt, et Matthieu Geist. Performance Evaluation for Particle Filters. In *International Conference on Information Fusion (FUSION 2011)*. IEEE, 2011

32. Olivier Pietquin, Matthieu Geist, et Senthilkumar Chandramohan. Sample Efficient On-line Learning of Optimal Dialogue Policies with Kalman Temporal Differences. In *International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 1878–1883, 2011

33. Jérémy Fix, Matthieu Geist, Olivier Pietquin, et Hervé Frezza-Buet. Dynamic Neural Field Optimization using the Unscented Kalman Filter. In *IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB 2011)*, 2011

34. Matthieu Geist et Olivier Pietquin. Parametric Value Function Approximation: a Unified View. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL 2011)*, pages 9–16, 2011

35. Matthieu Geist et Olivier Pietquin. Managing Uncertainty within the KTD Framework. In *Workshop on Active Learning and Experimental Design (AL&E collocated with AISTAT 2010)*, Journal of Machine Learning Research (Conference and Workshop Proceedings), pages 157–168, 2011

36. Matthieu Geist et Bruno Scherrer. ℓ_1-penalized projected Bellman residual. In *European Workshop on Reinforcement Learning (EWRL 2011)*, Lecture Notes in Computer Science (LNCS). Springer Verlag - Heidelberg Berlin, 2011

37. Edouard Klein, Matthieu Geist, et Olivier Pietquin. Batch, Off-policy and Model-free Apprenticeship Learning. In *European Workshop on Reinforcement Learning (EWRL 2011)*, Lecture Notes in Computer Science (LNCS). Springer Verlag - Heidelberg Berlin, 2011

38. Bruno Scherrer et Matthieu Geist. Recursive Least-Squares Learning with Eligibility Traces. In *European Workshop on Machine Learning (EWRL 2011)*, Lecture Notes in Computer Science (LNCS). Springer Verlag - Heidelberg Berlin, 2011

39. Matthieu Geist et Olivier Pietquin. Eligibility Traces through Colored Noises. In *IEEE International Conference on Ultra Modern Control systems (ICUMT 2010)*, pages 458 – 465, 2010. (best paper award)

40. Matthieu Geist et Olivier Pietquin. Statistically Linearized Least-Squares Temporal Differences. In *IEEE International Conference on Ultra Modern Control systems (ICUMT 2010)*, pages 450 – 457, 2010

41. Senthilkumar Chandramohan, Matthieu Geist, et Olivier Pietquin. Optimizing Spoken Dialogue Management with Fitted Value Iteration. In *International Conference on Speech Communication and Technologies (Interspeech 2010)*, pages 86–89. ISCA, 2010

42. Senthilkumar Chandramohan, Matthieu Geist, et Olivier Pietquin. Sparse Approximate Dynamic Programming for Dialog Management. In *SIGDial Conference on Discourse and Dialogue*, pages 107–115. ACL, 2010

43. Matthieu Geist et Olivier Pietquin. Statistically Linearized Recursive Least Squares. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2010)*, pages 272 – 276, 2010

44. Matthieu Geist et Olivier Pietquin. Revisiting natural actor-critics with value function approximation. In V. Torra, Y. Narukawa, et M. Daumas, editors, *Modeling Decisions for Artificial Intelligence (MDAI 2010)*, volume 6408 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 207–218. Springer Verlag - Heidelberg Berlin, 2010

45. Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Tracking in Reinforcement Learning. In *International Conference on Neural Information Processing (ICONIP 2009)*, volume 5863, Part I, pages 502–511. Springer LNCS, 2009. ENNS best student paper award

46. Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Kernelizing Vector Quantization Algorithms. In *European Symposium on Artificial Neural Networks (ESANN 09)*, pages 541–546, 2009
47. Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Kalman Temporal Differences: the deterministic case . In *IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL 2009)*, pages 185–192, 2009
48. Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Bayesian Reward Filtering. In S. Girgin et al., editor, *Recent Advances in Reinforcement Learning*, volume 5323 of *Lecture Notes in Computer Science (LNCS)*, pages 96–109. Springer Verlag, 2008. Revised and selected papers of EWRL 2008
49. Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Online Bayesian Kernel Regression from Nonlinear Mapping of Observations. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2008)*, pages 309–314, 2008
50. Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. A Sparse Nonlinear Bayesian Online Kernel Regression. In *IEEE International Conference on Advanced Engineering Computing and Applications in Sciences (AdvComp 2008)*, volume 1, pages 199–204, 2008. (best paper award)

Brevet

1. Julien Oster, Gary Clifford, Olivier Pietquin, et Matthieu Geist. Periodic Artifact Reduction from Biomedical Signals, 2013. patent WO2013052944

Articles de journaux nationaux

1. Edouard Klein, Bilal Piot, Matthieu Geist, et Olivier Pietquin. Classification structurée pour l'apprentissage par renforcement inverse. *Revue d'Intelligence Artificielle*, 2013
2. Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Différences temporelles de Kalman : cas déterministe. *Revue d'Intelligence Artificielle*, 24(4):423–442, 2010

Articles de conférences nationales

1. Bilal Piot, Matthieu Geist, et Olivier Pietquin. Méthode de minimisation du résidu de Bellman boostée qui tient compte des démonstrations expertes. In *Journées Francophones de Plannification, Décision et Apprentissage (JFPDA)*, 2014
2. Bruno Scherrer et Matthieu Geist. Quand l'optimalité locale implique une garantie globale : recherche locale de politique dans un espace convexe et algorithme d'itéation sur les politiques conservatif vu comme une montée de gradient. In *Journées Francophones de Plannification, Décision et Apprentissage (JFPDA)*, 2014
3. Lucie Daubigney, Matthieu Geist, et Olivier Pietquin. Optimisation par essaims particulaires de stratégies de dialogue. In *Journées Francophones de Plannification, Décision et Apprentissage (JFPDA)*, 2013
4. Edouard Klein, Bilal Piot, Matthieu Geist, et Olivier Pietquin. Apprentissage par renforcement inverse en cascadant classification et régression. In *Journées Francophones de Plannification, Décision et Apprentissage (JFPDA)*, 2013

5. Bilal Piot, Matthieu Geist, et Olivier Pietquin. Classification régularisée par la récompense pour l'Apprentissage par Imitation. In *Journées Francophones de Plannification, Décision et Apprentissage (JFPDA)*, 2013
6. Bilal Piot, Matthieu Geist, et Olivier Pietquin. Apprentissage par démonstrations : vaut-il la peine d'estimer une fonction de récompense? In *Journées Francophones de Plannification, Décision et Apprentissage (JFPDA)*, 2013
7. Edouard Klein, Bilal Piot, Matthieu Geist, et Olivier Pietquin. Classification structurée pour l'apprentissage par renforcement inverse. In *Conférence Francophone sur l'Apprentissage Automatique (Cap 2012)*, 2012
8. Jérémy Fix et Matthieu Geist. Optimisation de contrôleurs par essaim de particules. In *Conférence Francophone sur l'Apprentissage Automatique (Cap 2012)*, 2012
9. Lucie Daubigney, Matthieu Geist, et Olivier Pietquin. Apprentissage off-policy appliqué à un système de dialogue basé sur les PDMPO. In *Congrès francophone sur la Reconnaissance de Formes et l'Intelligence Artificielle (RFIA 2012)*, 2012
10. Matthieu Geist, Bruno Scherrer, Alessandro Lazaric, et Mohammad Ghavamzadeh. Un sélecteur de Dantzig pour l'apprentissage par différences temporelles. In *Journées Francophones sur la Planification, la Décision et l'Apprentissage pour la conduite des systèmes (JFPDA)*, 2012
11. Bruno Scherrer, Victor Gabillon, Mohammad Ghavamzadeh, et Matthieu Geist. Approximations de l'algorithme Itérations sur les Politiques Modifié. In *Journées Francophones sur la Planification, la Décision et l'Apprentissage pour la conduite des systèmes (JFPDA)*, 2012
12. Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefèvre, et Olivier Pietquin. Regroupement non-supervisé d'utilisateurs par leur comportement pour les systèmes de dialogue parlé. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2012)*, 2012
13. Lucie Daubigney, Matthieu Geist, et Olivier Pietquin. Apprentissage par renforcement pour la personnalisation d'un logiciel d'enseignement des langues. In *Conférence sur les Environnements Informatiques pour l'Apprentissage Humain (EIAH 2011)*, 2011
14. Matthieu Geist et Bruno Scherrer. Moindres carrés récursifs pour l'évaluation off-policy d'une politique avec traces d'éligibilité. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2011)*, 2011
15. Edouard Klein, Matthieu Geist, et Olivier Pietquin. Apprentissage par imitation étendu au cas batch, off-policy et sans modèle. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2011)*, 2011
16. Lucie Daubigney, Matthieu Geist, et Olivier Pietquin. Gestion de l'incertitude pour l'optimisation en ligne d'un gestionnaire de dialogues parlés à grande échelle basé sur les POMDP. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2011)*, 2011
17. Senthilkumar Chandramohan, Matthieu Geist, et Olivier Pietquin. Apprentissage par Renforcement Inverse pour la Simulation d'Utilisateurs dans les Systèmes de Dialogue. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2011)*, 2011

18. Matthieu Geist et Olivier Pietquin. Gestion de l'incertitude dans le cadre de l'approximation de la fonction de valeur pour l'apprentissage par renforcement. In *Conférence francophone sur l'apprentissage automatique (CAP 2010)*, pages 101–112, 2010
19. Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Astuce du Noyau & Quantification Vectorielle. In *Colloque sur la Reconnaissance des Formes et l'Intelligence Artificielle (RFIA'10)*, 2010
20. Matthieu Geist et Olivier Pietquin. Linéarisation statistique pour les différences temporelles par moindres carrés. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2010)*, 2010
21. Matthieu Geist et Olivier Pietquin. Architectures acteur-critique avec approximation de la valeur. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2010)*, 2010
22. Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Différences Temporelles de Kalman : le cas stochastique. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2009)*, 2009
23. Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Différences Temporelles de Kalman. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2009)*, 2009
24. Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Filtrage bayésien de la récompense. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2008)*, pages 113–122, 2008

Résumé des publications

Résumé	
Articles de journaux internationaux (1^{er} auteur)	9 (5)
Articles de conférences internationales	50 (17)
Brevet	1 (0)
Articles de journaux nationaux	2 (1)
Articles de conférences nationales	24 (9)

Bibliographie

[1] Pieter Abbeel et Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2004.

[2] Dimitris Achlioptas. Database-friendly random projections : Johnson-Lindenstrauss with binary coins. *Journal of computer and System Sciences*, 66(4) :671–687, 2003.

[3] Luiz A Amaral et Detmar Meurers. On using intelligent computer-assisted language learning in real-life foreign language teaching and learning. *ReCALL*, 23(01) :4–24, 2011.

[4] Andras Antos, Csaba Szepesvári, et Rémi Munos. Fitted Q-iteration in continuous action-space MDPs. In *Advances in neural information processing systems (NIPS)*, pages 9–16, 2008.

[5] András Antos, Csaba Szepesvári, et Rémi Munos. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1) :89–129, 2008.

[6] T. Archibald, K. McKinnon, et L. Thomas. On the generation of Markov decision processes. *Journal of the Operational Research Society*, 46 :354–361, 1995.

[7] Karl J Astrom. Optimal control of Markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1) :174, 1965.

[8] Bernardo Ávila Pires, Csaba Szepesvari, et Mohammad Ghavamzadeh. Cost-sensitive multiclass classification risk bounds. In *International Conference on Machine Learning (ICML)*, pages 1391–1399, 2013.

[9] Leemon C. Baird. Residual Algorithms : Reinforcement Learning with Function Approximation. In *International Conference on Machine Learning (ICML)*, pages 30–37, 1995.

[10] André MS Barreto, Joelle Pineau, et Doina Precup. Policy iteration based on stochastic factorization. *Journal of Artificial Intelligence Research*, pages 763–803, 2014.

[11] Jonathan Baxter et Peter L. Bartlett. Infinite-Horizon Gradient-Based Policy Search. *Journal of Artificial Intelligence Research (JAIR)*, 15 :319–350, 2001.

[12] Oscar Beijbom, Mohammad Saberian, David Kriegman, et Nuno Vasconcelos. Guess-Averse Loss Functions For Cost-Sensitive Multiclass Boosting. In *International Conference on Machine Learning (ICML)*, pages 586–594, 2014.

[13] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[14] Dimitri P. Bertsekas et John N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[15] Dimitri P. Bertsekas et Huizhen Yu. Projected equation methods for approximate solution of large linear systems. *Journal of Computational and Applied Mathematics*, 227 :27–50, 2009.

[16] Shalabh Bhatnagar, Richard S. Sutton, Mohammad Ghavamzadeh, et Mark Lee. Natural Actor-Critic Algorithms. *Automatica*, 2009.

[17] Abdeslam Boularias, Jens Kober, et Jan R Peters. Relative entropy inverse reinforcement learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 182–189, 2011.

[18] Steven J. Bradtke et Andrew G. Barto. Linear Least-Squares algorithms for temporal difference learning. *Machine Learning*, 22(1-3) :33–57, 1996.

[19] Emmanuel Candes et Terence Tao. The Dantzig selector : statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6) :2313–2351, 2007.

[20] Senthilkumar Chandramohan. *Revisiting User Simulation in Dialogue Systems : Do we still need them ? Will imitation play the role of simulation ?* Thèse de Doctorat en Informatique, Université d'Avignon, 2012.

[21] Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefèvre, et Olivier Pietquin. User Simulation in Dialogue Systems using Inverse Reinforcement Learning. In *Annual Conference of the International Speech Communication Association (Interspeech 2011)*, pages 1025–1028, 2011.

[22] Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefèvre, et Olivier Pietquin. Behavior Specific User Simulation in Spoken Dialogue Systems. In *ITG Conference on Speech Communication*, 2012.

[23] Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefèvre, et Olivier Pietquin. Clustering Behaviors Of Spoken Dialogue Systems Users. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)*, 2012.

[24] Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefèvre, et Olivier Pietquin. Co-adaptation in Spoken Dialogue Systems. In *International Workshop on Spoken Dialog Systems (IWSDS 2012)*, 2012.

[25] Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefèvre, et Olivier Pietquin. Regroupement non-supervisé d'utilisateurs par leur comportement pour les systèmes de dialogue parlé. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2012)*, 2012.

[26] Senthilkumar Chandramohan, Matthieu Geist, et Olivier Pietquin. Optimizing Spoken Dialogue Management with Fitted Value Iteration. In *International Conference on Speech Communication and Technologies (Interspeech 2010)*, pages 86–89. ISCA, 2010.

[27] Senthilkumar Chandramohan, Matthieu Geist, et Olivier Pietquin. Sparse Approximate Dynamic Programming for Dialog Management. In *SIGDial Conference on Discourse and Dialogue*, pages 107–115. ACL, 2010.

[28] Senthilkumar Chandramohan, Matthieu Geist, et Olivier Pietquin. Apprentissage par Renforcement Inverse pour la Simulation d'Utilisateurs dans les Systèmes de Dialogue. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2011)*, 2011.

[29] Kai-min Chang, Joseph Beck, Jack Mostow, et Albert Corbett. A bayes net toolkit for student modeling in intelligent tutoring systems. In *Intelligent Tutoring Systems*, pages 104–113. Springer, 2006.

[30] Jessica Chemali et Alessandro Lazaric. Direct Policy Iteration with Demonstrations. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[31] Sonia Chernova et Manuela Veloso. Interactive policy learning through confidence-based autonomy. *Journal of Artificial Intelligence Research*, 34(1) :1, 2009.

[32] David Choi et Benjamin Van Roy. A Generalized Kalman Filter for Fixed Point Approximation and Efficient Temporal-Difference Learning. *Discrete Event Dynamic Systems*, 16 :207–239, 2006.

[33] Rémi Chou, Yvo Boers, Martin Podt, et Matthieu Geist. Performance Evaluation for Particle Filters. In *International Conference on Information Fusion (FUSION 2011)*. IEEE, 2011.

[34] Albert T Corbett et John R Anderson. Knowledge tracing : Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4) :253–278, 1994.

[35] Albert T Corbett, John R Anderson, et Alison T O'Brien. Student modeling in the ACT Programming Tutor. *Cognitively diagnostic assessment*, pages 19–41, 1995.

[36] Lucie Daubigney. *Gestion de l'incertitude pour l'optmisation de systèmes interactifs*. Thèse de Doctorat en Informatique, Université de Lorraine, 2013.

[37] Lucie Daubigney, Milica Gasic, Senthilkumar Chandramohan, Matthieu Geist, Olivier Pietquin, et Steve Young. Uncertainty management for on-line optimisation of a POMDP-based large-scale spoken dialogue system. In *Annual Conference of the International Speech Communication Association (Interspeech 2011)*, pages 1301–1304, 2011.

[38] Lucie Daubigney, Matthieu Geist, Senthilkumar Chandramohan, et Olivier Pietquin. A Comprehensive Reinforcement Learning Framework for Dialogue Management Optimisation. *IEEE Journal of Selected Topics in Signal Processing*, 6(8) :891–902, 2012.

[39] Lucie Daubigney, Matthieu Geist, et Olivier Pietquin. Apprentissage par renforcement pour la personnalisation d'un logiciel d'enseignement des langues. In *Conférence sur les Environnements Informatiques pour l'Apprentissage Humain (EIAH 2011)*, 2011.

[40] Lucie Daubigney, Matthieu Geist, et Olivier Pietquin. Gestion de l'incertitude pour l'optimisation en ligne d'un gestionnaire de dialogues parlés à grande échelle basé sur les POMDP. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2011)*, 2011.

[41] Lucie Daubigney, Matthieu Geist, et Olivier Pietquin. Apprentissage off-policy appliqué à un système de dialogue basé sur les PDMPO. In *Congrès francophone sur la Reconnaissance de Formes et l'Intelligence Artificielle (RFIA 2012)*, 2012.

[42] Lucie Daubigney, Matthieu Geist, et Olivier Pietquin. Off-policy Learning in Large-scale POMDP-based Dialogue Systems. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)*, pages 4989 – 4992, 2012.

[43] Lucie Daubigney, Matthieu Geist, et Olivier Pietquin. Model-free POMDP optimisation of tutoring systems with echo-state networks. In *Proceedings of the 14th SIGDial Meeting on Discourse and Dialogue (SIGDial 2013)*, pages 102–106, 2013.

[44] Lucie Daubigney, Matthieu Geist, et Olivier Pietquin. Optimisation par essaims particulaires de stratégies de dialogue. In *Journées Francophones de Plannification, Décision et Apprentissage (JFPDA)*, 2013.

[45] Lucie Daubigney, Matthieu Geist, et Olivier Pietquin. Particle Swarm Optimisation of Spoken Dialogue System Strategies. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association (Interspeech 2013)*, 2013.

[46] Lucie Daubigney, Matthieu Geist, et Olivier Pietquin. Random Projetctions : a Remedy for Overfitting Issues in Time Series Prediction with Echo State Networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, 2013.

[47] Daniela Pucci de Farias et Benjamin Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6) :850–865, 2003.

[48] Bradley Efron, Trevor Hastie, Iain Johnstone, et Robert Tibshirani. Least Angle Regression. *Annals of Statistics*, 32(2) :407–499, 2004.

[49] Yaakov Engel. *Algorithms and Representations for Reinforcement Learning.* PhD thesis, Hebrew University, 2005.

[50] Yaakov Engel, Shie Mannor, et Ron Meir. Bayes Meets Bellman : The Gaussian Process Approach to Temporal Difference Learning. In *International Conference on Machine Learning (ICML)*, pages 154–161, 2003.

[51] Yaakov Engel, Shie Mannor, et Ron Meir. The kernel recursive least-squares algorithm. *IEEE Transactions on Signal Processing*, 52(8) :2275–2285, 2004.

[52] Damien Ernst, Pierre Geurts, et Louis Wehenkel. Tree-Based Batch Mode Reinforcement Learning. *Journal of Machine Learning Research*, 6 :503–556, 2005.

[53] Amir M. Farahmand, Mohammad Ghavamzadeh, Shie Mannor, et Csaba Szepesvári. Regularized policy iteration. In *Advances in Neural Information Processing Systems (NIPS)*, pages 441–448, 2009.

[54] Amir-massoud Farahmand et Csaba Szepesvári. Model selection in reinforcement learning. *Machine learning*, 85(3) :299–332, 2011.

[55] Amir-massoud Farahmand, Csaba Szepesvári, et Rémi Munos. Error propagation for approximate policy and value iteration. In *Advances in Neural Information Processing Systems (NIPS)*, pages 568–576, 2010.

[56] Alan Fern, Sung Wook Yoon, et Robert Givan. Approximate Policy Iteration with a Policy Language Bias : Solving Relational Markov Decision Processes. *Journal of Artificial Intelligence Research (JAIR)*, 25 :75–118, 2006.

[57] Jérémy Fix et Matthieu Geist. Monte-Carlo Swarm Policy Search. In *Symposium on Swarm Intelligence and Differential Evolution*, Lecture Notes in Artificial Intelligence (LNAI). Springer Verlag - Heidelberg Berlin, 2012.

[58] Jérémy Fix et Matthieu Geist. Optimisation de contrôleurs par essaim de particules. In *Conférence Francophone sur l'Apprentissage Automatique (Cap 2012)*, 2012.

[59] Jérémy Fix, Matthieu Geist, Olivier Pietquin, et Hervé Frezza-Buet. Dynamic Neural Field Optimization using the Unscented Kalman Filter. In *IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB 2011)*, 2011.

[60] Hervé Frezza-Buet et Matthieu Geist. A C++ Template-Based Reinforcement Learning Library : Fitting the Code to the Mathematics. *Journal of Machine Learning Research*, 14 :399–402, 2013.

[61] Matthieu Geist. Analyse des données pour l'analyse, le suivi et le contrôle des dispersions. Master's thesis, Supélec, Septembre 2006.

[62] Matthieu Geist. Modélisation de chaînes de production et de leurs interactions. Master's thesis, Supélec (M2R Mathématiques), Septembre 2006.

[63] Matthieu Geist. *Optimisation des chaînes de production dans l'industrie sidérurgique : une approche statistique de l'apprentissage par renforcement.* Doctorat en Mathématiques, Université Paul Verlaine de Metz, Novembre 2009.

[64] Matthieu Geist. A multiplicative UCB strategy for Gamma rewards. In *European Workshop on Reinforcement Learning (EWRL)*, 2015.

[65] Matthieu Geist. Soft-max boosting. *Machine Learning*, 2015.

[66] Matthieu Geist, Edouard Klein, Bilal Piot, Yann Guermeur, et Olivier Pietquin. Around Inverse Reinforcement Learning and Score-based Classification. In *1st Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM 2013)*, 2013.

[67] Matthieu Geist et Olivier Pietquin. Architectures acteur-critique avec approximation de la valeur. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2010)*, 2010.

[68] Matthieu Geist et Olivier Pietquin. Eligibility Traces through Colored Noises. In *IEEE International Conference on Ultra Modern Control systems (ICUMT 2010)*, pages 458 – 465, 2010. (best paper award).

[69] Matthieu Geist et Olivier Pietquin. Gestion de l'incertitude dans le cadre de l'approximation de la fonction de valeur pour l'apprentissage par renforcement. In *Conférence francophone sur l'apprentissage automatique (CAP 2010)*, pages 101–112, 2010.

[70] Matthieu Geist et Olivier Pietquin. Kalman Temporal Differences. *Journal of Artificial Intelligence Research (JAIR)*, 39 :483–532, 2010.

[71] Matthieu Geist et Olivier Pietquin. Linéarisation statistique pour les différences temporelles par moindres carrés. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2010)*, 2010.

[72] Matthieu Geist et Olivier Pietquin. Revisiting natural actor-critics with value function approximation. In V. Torra, Y. Narukawa, et M. Daumas, editors, *Modeling Decisions for Artificial Intelligence (MDAI 2010)*, volume 6408 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 207–218. Springer Verlag - Heidelberg Berlin, 2010.

[73] Matthieu Geist et Olivier Pietquin. Statistically Linearized Least-Squares Temporal Differences. In *IEEE International Conference on Ultra Modern Control systems (ICUMT 2010)*, pages 450 – 457, 2010.

[74] Matthieu Geist et Olivier Pietquin. Statistically Linearized Recursive Least Squares. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2010)*, pages 272 – 276, 2010.

[75] Matthieu Geist et Olivier Pietquin. Kalman filtering & colored noises : the (autoregressive) moving-average case. In *IEEE Workshop on Machine Learning Algorithms, Systems and Applications (MLASA 2011)*, 2011.

[76] Matthieu Geist et Olivier Pietquin. Managing Uncertainty within the KTD Framework. In *Workshop on Active Learning and Experimental Design (AL&E collocated with AISTAT 2010)*, Journal of Machine Learning Research (Conference and Workshop Proceedings), pages 157–168, 2011.

[77] Matthieu Geist et Olivier Pietquin. Parametric Value Function Approximation : a Unified View. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL 2011)*, pages 9–16, 2011.

[78] Matthieu Geist et Olivier Pietquin. An Algorithmic Survey of Parametric Value Function Approximation. *IEEE Transactions on Neural Networks and Learning Systems*, 24(6) :845–867, 2013.

[79] Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. A Sparse Nonlinear Bayesian Online Kernel Regression. In *IEEE International Conference on Advanced Engineering Computing and Applications in Sciences (AdvComp 2008)*, volume 1, pages 199–204, 2008. (best paper award).

[80] Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Bayesian Reward Filtering. In S. Girgin et al., editor, *Recent Advances in Reinforcement Learning*, volume 5323 of *Lecture Notes in Computer Science (LNCS)*, pages 96–109. Springer Verlag, 2008. Revised and selected papers of EWRL 2008.

[81] Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Filtrage bayésien de la récompense. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2008)*, pages 113–122, 2008.

[82] Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Online Bayesian Kernel Regression from Nonlinear Mapping of Observations. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2008)*, pages 309–314, 2008.

[83] Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Différences Temporelles de Kalman. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2009)*, 2009.

[84] Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Différences Temporelles de Kalman : le cas stochastique. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2009)*, 2009.

[85] Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. From Supervised to Reinforcement Learning : a Kernel-based Bayesian Filtering Framework. *International Journal On Advances in Software*, 2(1) :101–116, 2009.

[86] Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Kalman Temporal Differences : the deterministic case . In *IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL 2009)*, pages 185–192, 2009.

[87] Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Kernelizing Vector Quantization Algorithms. In *European Symposium on Artificial Neural Networks (ESANN 09)*, pages 541–546, 2009.

[88] Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Tracking in Reinforcement Learning. In *International Conference on Neural Information Processing (ICONIP 2009)*, volume 5863, Part I, pages 502–511. Springer LNCS, 2009. ENNS best student paper award.

[89] Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Astuce du Noyau & Quantification Vectorielle. In *Colloque sur la Reconnaissance des Formes et l'Intelligence Artificielle (RFIA'10)*, 2010.

[90] Matthieu Geist, Olivier Pietquin, et Gabriel Fricout. Différences temporelles de Kalman : cas déterministe. *Revue d'Intelligence Artificielle*, 24(4) :423–442, 2010.

[91] Matthieu Geist et Bruno Scherrer. ℓ_1-penalized projected Bellman residual. In *European Workshop on Reinforcement Learning (EWRL 2011)*, Lecture Notes in Computer Science (LNCS). Springer Verlag - Heidelberg Berlin, 2011.

[92] Matthieu Geist et Bruno Scherrer. Moindres carrés récursifs pour l'évaluation off-policy d'une politique avec traces d'éligibilité. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2011)*, 2011.

[93] Matthieu Geist et Bruno Scherrer. Off-policy Learning with Eligibility Traces : A Survey. *Journal of Machine Learning Research (JMLR)*, 15 :289–333, 2014.

[94] Matthieu Geist, Bruno Scherrer, Alessandro Lazaric, et Mohammad Ghavamzadeh. A Dantzig Selector Approach to Temporal Difference Learning. In *International Conference on Machine Learning (ICML)*, 2012.

[95] Matthieu Geist, Bruno Scherrer, Alessandro Lazaric, et Mohammad Ghavamzadeh. Un sélecteur de Dantzig pour l'apprentissage par différences temporelles. In *Journées Francophones sur la Planification, la Décision et l'Apprentissage pour la conduite des systèmes (JFPDA)*, 2012.

[96] Hadrien Glaude, Fadi Akrimi, Matthieu Geist, et Olivier Pietquin. A Non-Parametric Approach to Approximate Dynamic Programming. In *IEEE International Conference on Machine Learning and Applications (ICMLA 2011)*, pages 317–322, 2011.

[97] Geoffrey Gordon. Stable Function Approximation in Dynamic Programming. In *International Conference on Machine Learning (IMCL)*, 1995.

[98] Arthur C Graesser, Patrick Chipman, Brian C Haynes, et Andrew Olney. AutoTutor : An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, 48(4) :612–618, 2005.

[99] Alexander Grubb et Drew Bagnell. Generalized Boosting Algorithms for Convex Optimization. In *International Conference on Machine Learning (ICML)*, pages 1209–1216, 2011.

[100] Steffen Grünewälder, Guy Lever, Luca Baldassarre, Massimilano Pontil, et Arthur Gretton. Modelling transition dynamics in MDPs with RKHS embeddings. In *International Conference on Machine Learning (ICML)*, pages 535–542. Omnipress, 2012.

[101] Verena Heidrich-Meisner et Christian Igel. Evolution strategies for direct policy search. In *Parallel Problem Solving from Nature–PPSN X*, pages 428–437. Springer, 2008.

[102] Matthew W. Hoffman, Alessandro Lazaric, Mohammad Ghavamzadeh, et Rémi Munos. Regularized Least Squares Temporal Difference learning with nested ℓ_2 and ℓ_1 penalization. In *European Workshop on Reinforcement Learning (EWRL)*, 2011.

[103] Auke J Ijspeert, Jun Nakanishi, et Stefan Schaal. Learning attractor landscapes for learning motor primitives. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1523–1530, 2002.

[104] Herbert Jaeger. The "echo state" approach to analyzing and training recurrent neural networks. Technical Report GMD Report 148, Fraunhofer Institute for Autonomous Intelligent Systems, 2001.

[105] Simon J. Julier et Jeffrey K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3) :401–422, 2004.

[106] Leslie Pack Kaelbling, Michael L. Littman, et Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2) :99–134, 1998.

[107] Sham Kakade. A Natural Policy Gradient. In *Neural Information Processing Systems (NIPS)*, pages 1531–1538, 2001.

[108] Sham Kakade et John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning (ICML)*, volume 2, pages 267–274, 2002.

[109] Michael Kearns et Satinder Singh. Bias-Variance Error Bounds for Temporal Difference Updates. In *Conference on Learning Theory (COLT)*, 2000.

[110] John F Kelley. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1) :26–41, 1984.

[111] James Kennedy et Russell Eberhart. Particle swarm optimization. In *IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995.

[112] Beomjoon Kim, Amir massoud Farahmand, Joelle Pineau, et Doina Precup. Learning from limited demonstrations. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2859–2867, 2013.

[113] Edouard Klein. *Contributions à l'apprentissage par renforcement inverse.* Thèse de Doctorat en Informatique, Université de Lorraine, 2013.

[114] Edouard Klein, Matthieu Geist, et Olivier Pietquin. Apprentissage par imitation étendu au cas batch, off-policy et sans modèle. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2011)*, 2011.

[115] Edouard Klein, Matthieu Geist, et Olivier Pietquin. Batch, Off-policy and Model-free Apprenticeship Learning. In *European Workshop on Reinforcement Learning (EWRL 2011)*, Lecture Notes in Computer Science (LNCS). Springer Verlag - Heidelberg Berlin, 2011.

[116] Edouard Klein, Matthieu Geist, et Olivier Pietquin. Reducing the dimentionality of the reward space in the Inverse Reinforcement Learning problem. In *IEEE Workshop on Machine Learning Algorithms, Systems and Applications (MLASA 2011)*, 2011.

[117] Edouard Klein, Matthieu Geist, Bilal Piot, et Olivier Pietquin. Inverse Reinforcement Learning through Structured Classification. In *Advances in Neural Information Processing Systems (NIPS 2012)*, 2012.

[118] Edouard Klein, Bilal Piot, Matthieu Geist, et Olivier Pietquin. Classification structurée pour l'apprentissage par renforcement inverse. In *Conférence Francophone sur l'Apprentissage Automatique (Cap 2012)*, 2012.

[119] Edouard Klein, Bilal Piot, Matthieu Geist, et Olivier Pietquin. A cascaded supervised learning approach to inverse reinforcement learning. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, et Filip Zelezny, editors, *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2013)*, volume 8188 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2013.

[120] Edouard Klein, Bilal Piot, Matthieu Geist, et Olivier Pietquin. Apprentissage par renforcement inverse en cascadant classification et régression. In *Journées Francophones de Plannification, Décision et Apprentissage (JFPDA)*, 2013.

[121] Edouard Klein, Bilal Piot, Matthieu Geist, et Olivier Pietquin. Classification structurée pour l'apprentissage par renforcement inverse. *Revue d'Intelligence Artificielle*, 2013.

[122] Jens Kober et Jan Peters. Policy search for motor primitives in robotics. *Machine Learning*, 84(1) :171–203, 2011.

[123] Kenneth R Koedinger, John R Anderson, William H Hadley, et Mary A Mark. Intelligent Tutoring Goes To School in the Big City. *International Journal of Artificial Intelligence in Education*, 8 :30–43, 1997.

[124] J. Zico Kolter. The Fixed Ponts of Off-Policy TD. In *Neural Information Processing Systems (NIPS)*, 2011.

[125] J. Zico Kolter et Andrew Y. Ng. Regularization and Feature Selection in Least-Squares Temporal Difference Learning. In *International Conference on Machine Learning (ICML)*, 2009.

[126] Michail G. Lagoudakis et Ronald Parr. Least-squares policy iteration. *The Journal of Machine Learning Research*, 4 :1107–1149, 2003.

[127] Michail G. Lagoudakis et Ronald Parr. Reinforcement Learning as Classification : Leveraging Modern Classifiers. In *International Conference on Machine Learning (ICML)*, pages 424–431, 2003.

[128] Staffan Larsson et David R Traum. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural language engineering*, 6(3& 4) :323–340, 2000.

[129] Alessandro Lazaric, Mohammad Ghavamzadeh, et Rémi Munos. Analysis of a classification-based policy iteration algorithm. In *International Conference on Machine Learning (ICML)*, pages 607–614, 2010.

[130] Yoonkyung Lee, Yi Lin, et Grace Wahba. Multicategory support vector machines : Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465) :67–81, 2004.

[131] Oliver Lemon, Kallirroi Georgila, James Henderson, et Matthew Stuttle. An ISU dialogue system exhibiting reinforcement learning of dialogue policies : generic slot-filling in the TALK in-car system. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics : Posters & Demonstrations*, pages 119–122. Association for Computational Linguistics, 2006.

[132] Oliver Lemon et Olivier Pietquin. *Data-Driven Methods for Adaptive Spoken Dialogue Systems : Computational Learning for Conversational Interfaces* . Springer, 2012. 177 pages.

[133] Guy Lever, Luca Baldassarre, Arthur Gretton, Massimiliano Pontil, et Steffen Grünewälder. Modelling transition dynamics in MDPs with RKHS embeddings. In *International Conference on Machine Learning (ICML)*, pages 535–542, 2012.

[134] Esther Levin, Roberto Pieraccini, et Wieland Eckert. A stochastic model of human-machine interaction for learning dialog strategies. *Speech and Audio Processing, IEEE Transactions on*, 8(1) :11–23, 2000.

[135] Lihong Li, Jason D Williams, et Suhrid Balakrishnan. Reinforcement learning for dialog management using least-squares policy iteration and fast feature selection. In *InterSpeech*, pages 2475–2478, 2009.

[136] Manuel Loth, Manuel Davy, et Philippe Preux. Sparse temporal difference learning using lasso. In *Approximate Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 352–359. IEEE, 2007.

[137] Jun Ma et Warren B. Powell. Convergence Analysis of Kernel-based On-policy Approximate Policy Iteration Algorithms for Markov Decision Processes with Continuous, Multidimensional States and Actions. Technical report, Princeton University, 2010.

[138] Hamid Maei, Csaba Szepesvari, Shalabh Bhatnagar, Doina Precup, David Silver, et Richard S. Sutton. Convergent Temporal-Difference Learning with Arbitrary Smooth Function Approximation. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, et A. Culotta, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 1204–1212, 2009.

[139] Hamid R. Maei et Richard S. Sutton. GQ(λ) : A general gradient algorithm for temporal-difference prediction learning with eligibility traces. In *Conference on Artificial General Intelligence (AGI)*, 2010.

[140] Hamid R. Maei, Csaba Szepesvari, Shalabj Bhatnagar, et Richard S. Sutton. Toward Off-Policy Learning Control with Function Approximation. In *International Conference on Machine Learning (ICML)*, 2010.

[141] Maurizio Mancini, Laurent Ach, Emeline Bantegnie, Tobias Baur, Nadia Berthouze, Debajyoti Datta, Yu Ding, Stéphane Dupont, Harry J Griffin, Florian Lingenfelser, Radoslaw Niewiadomski, Catherine Pelachaud, Olivier Pietquin, Bilal Piot, Jérôme Urbain, Gualtiero Volpe, et Johannes Wagner. Laugh When You're Winning. In *Innovative and Creative Developments in Multimodal Interaction Systems*, pages 50–79. Springer, 2014.

[142] Llew Mason, Jonathan Baxter, Peter Bartlett, et Marcus Frean. Boosting algorithms as gradient descent in function space. Neural Information Processing Systems (NIPS), 1999.

[143] Llew Mason, Jonathan Baxter, Peter Bartlett, et Marcus Frean. Boosting algorithms as gradient descent in function space. Neural Information Processing Systems (NIPS, 1999.

[144] Rémi Munos. Error bounds for approximate policy iteration. In *International Conference on Machine Learning (ICML)*, pages 560–567, 2003.

[145] Rémi Munos. Performance bounds in ℓ_p-norm for approximate value iteration. *SIAM journal on control and optimization*, 46(2) :541–561, 2007.

[146] Rémi Munos et Csaba Szepesvári. Finite-time bounds for fitted value iteration. *The Journal of Machine Learning Research (JMLR)*, 9 :815–857, 2008.

[147] Thibaut Munzer, Bilal Piot, Matthieu Geist, Olivier Pietquin, et Manuel Lopes. Inverse Reinforcement Learning in Relational Domains. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2015.

[148] A. Nedić et Dimitri P. Bertsekas. Least Squares Policy Evaluation Algorithms with Linear Function Approximation. *Discrete Event Dynamic Systems : Theory and Applications*, 13 :79–110, 2003.

[149] Gergely Neu et Csaba Szepesvári. Training parsers by inverse reinforcement learning. *Machine learning*, 77(2-3) :303–337, 2009.

[150] Andrew Y Ng, Daishi Harada, et Stuart Russell. Policy invariance under reward transformations : Theory and application to reward shaping. In *International Conference on Machine Learning (ICML)*, volume 99, pages 278–287, 1999.

[151] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 663–670, 2000.

[152] Radoslaw Niewiadomski, Jennifer Hofmann, Jérome Urbain, Tracey Platt, Johannes Wagner, Bilal Piot, Huseyin Cakmak, Sathish Pammi, Tobias Baur, Stéphane Dupont, Matthieu Geist, Florian Lingenfelser, Gary McKeown, Olivier Pietquin, et Willibald Ruch. Laugh-aware virtual agent and its impact on user amusement . In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, 2013.

[153] Dirk Ormoneit et Śaunak Sen. Kernel-based reinforcement learning. *Machine learning*, 49(2-3) :161–178, 2002.

[154] Julien Oster, Gary Clifford, Olivier Pietquin, et Matthieu Geist. Periodic Artifact Reduction from Biomedical Signals, 2013. patent WO2013052944.

[155] Julien Oster, Matthieu Geist, Olivier Pietquin, et Gary Clifford. Filtering of pathological ventricular rhythms during MRI scanning. In *International Workshop on Biosignal Interpretation*, 2012.

[156] Jan Peters et Stefan Schaal. Natural Actor-Critic. *Neurocomputing*, 71 :1180–1190, 2008.

[157] Marek Petrik, Gavin Taylor, Ronald Parr, et Shlomo Zilberstein. Feature Selection Using Regularization in Approximate Linear Programs for Markov Decision Processes. In *International Conference on Machine Learning (ICML)*, pages 871–878, 2010.

[158] Olivier Pietquin. *A framework for unsupervised learning of dialogue strategies*. PhD thesis, Faculté Polytechnique de Mons, 2004.

[159] Olivier Pietquin. Consistent goal-directed user model for realisitc man-machine task-oriented spoken dialogue simulation. In *IEEE International Conference on Multimedia and Expo*, pages 425–428, 2006.

[160] Olivier Pietquin, Lucie Daubigney, et Matthieu Geist. Optimization of a Tutoring System from a Fixed Set of Data. In *ISCA workshop on Speech and Language Technology in Education (SLaTE 2011)*, 2011.

[161] Olivier Pietquin et Thierry Dutoit. A probabilistic framework for dialog simulation and optimal strategy learning. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(2) :589–599, 2006.

[162] Olivier Pietquin, Matthieu Geist, et Senthilkumar Chandramohan. Sample Efficient On-line Learning of Optimal Dialogue Policies with Kalman Temporal Differences. In *International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 1878–1883, 2011.

[163] Olivier Pietquin, Matthieu Geist, Senthilkumar Chandramohan, et Hervé Frezza-Buet. Sample-Efficient Batch Reinforcement Learning for Dialogue Management Optimization. *ACM Transactions on Speech and Language Processing*, 7(3), 2011.

[164] Olivier Pietquin et Helen Hastie. A survey on metrics for the evaluation of user simulations. *The knowledge engineering review*, 28(01) :59–73, 2013.

[165] Bilal Piot. *Apprentissage hors-ligne avec Démonstrations Expertes*. Thèse de Doctorat en Informatique, Université de Lorraine, 2014.

[166] Bilal Piot, Matthieu Geist, et Olivier Pietquin. Apprentissage par démonstrations : vaut-il la peine d'estimer une fonction de récompense ? In *Journées Francophones de Plannification, Décision et Apprentissage (JFPDA)*, 2013.

[167] Bilal Piot, Matthieu Geist, et Olivier Pietquin. Classification régularisée par la récompense pour l'Apprentissage par Imitation. In *Journées Francophones de Plannification, Décision et Apprentissage (JFPDA)*, 2013.

[168] Bilal Piot, Matthieu Geist, et Olivier Pietquin. Learning from demonstrations : Is it worth estimating a reward function ? In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, et Filip Zelezny, editors, *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2013)*, volume 8188 of *Lecture Notes in Computer Science*, pages 17–32. Springer, 2013.

[169] Bilal Piot, Matthieu Geist, et Olivier Pietquin. Boosted and Reward-regularized Classification for Apprenticeship Learning. In *13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014)*, Paris, France, 2014.

[170] Bilal Piot, Matthieu Geist, et Olivier Pietquin. Boosted Bellman Residual Minimization Handling Expert Demonstrations. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*. Springer, 2014.

[171] Bilal Piot, Matthieu Geist, et Olivier Pietquin. Difference of Convex Functions Programming for Reinforcement Learning. In *Advances in Neural Information Processing Systems (NIPS 2014)*, 2014.

[172] Bilal Piot, Matthieu Geist, et Olivier Pietquin. Méthode de minimisation du résidu de Bellman boostée qui tient compte des démonstrations expertes. In *Journées Francophones de Plannification, Décision et Apprentissage (JFPDA)*, 2014.

[173] Bilal Piot, Matthieu Geist, et Olivier Pietquin. Imitation Learning Applied to Embodied Conversational Agents. In *Machine Learning and Interactive Systems (MLIS)*, 2015.

[174] Bilal Piot, Olivier Pietquin, et Matthieu Geist. Predicting when to laugh with structured classification. In *Annual Conference of the International Speech Communication Association (InterSpeech)*, 2014.

[175] Bernardo A Pires et Csaba Szepesvári. Statistical linear estimation with penalized estimators : an application to reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 1535–1542, 2012.

[176] Martin L. Puterman. *Markov Decision Processes : Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 1994.

[177] Martin L Puterman et Moon Chirl Shin. Modified policy iteration algorithms for discounted markov decision problems. *Management Science*, 24(11) :1127–1137, 1978.

[178] Verena Rieser. *Bootstrapping reinforcement learning-based dialogue strategies from wizard-of-oz data*. PhD thesis, Saarland University, 2008.

[179] Brian D. Ripley. *Stochastic Simulation*. Wiley & Sons, 1987.

[180] Gavin A. Rummery et Mahesan Niranjan. Online Q-Learning using Connectionist Systems. Technical Report CUED/F-INFENG/TR 166, Cambridge University, 1994.

[181] Stuart Russell. Learning agents for uncertain environments. In *Conference on Computational Learning Theory (COLT)*, pages 101–103. ACM, 1998.

[182] Robert E Schapire et Yoav Freund. *Boosting : Foundations and algorithms*. MIT press, 2012.

[183] Jost Schatzmann, Karl Weilhammer, Matt Stuttle, et Steve Young. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The knowledge engineering review*, 21(02) :97–126, 2006.

[184] J Schatztnann, Matthew N Stuttle, Karl Weilhammer, et Steve Young. Effects of the user model on simulation-based learning of dialogue strategies. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 220–225, 2005.

[185] Bruno Scherrer. Approximate Policy Iteration Schemes : A Comparison. In *International Conference on Machine Learning (ICML)*, pages 1314–1322, 2014.

[186] Bruno Scherrer, Victor Gabillon, Mohammad Ghavamzadeh, et Matthieu Geist. Approximate Modified Policy Iteration. In *International Conference on Machine Learning (ICML)*, 2012.

[187] Bruno Scherrer, Victor Gabillon, Mohammad Ghavamzadeh, et Matthieu Geist. Approximations de l'algorithme Itérations sur les Politiques Modifié. In *Journées Francophones sur la Planification, la Décision et l'Apprentissage pour la conduite des systèmes (JFPDA)*, 2012.

[188] Bruno Scherrer et Matthieu Geist. Recursive Least-Squares Learning with Eligibility Traces. In *European Workshop on Machine Learning (EWRL 2011)*, Lecture Notes in Computer Science (LNCS). Springer Verlag - Heidelberg Berlin, 2011.

[189] Bruno Scherrer et Matthieu Geist. Local Policy Search in a Convex Space and Conservative Policy Iteration as Boosted Policy Search. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, 2014.

[190] Bruno Scherrer et Matthieu Geist. Quand l'optimalité locale implique une garantie globale : recherche locale de politique dans un espace convexe et algorithme d'itéation sur les politiques conservatif vu comme une montée de gradient. In *Journées Francophones de Plannification, Décision et Apprentissage (JFPDA)*, 2014.

[191] Bruno Scherrer, Mohammad Ghavamzadeh, Victor Gabillon, Boris Lesner, et Matthieu Geist. Approximate Modified Policy Iteration and its Application to the Game of Tetris. *Journal of Machine Learning Research*, 2015.

[192] Bruno Scherrer et Boris Lesner. On the use of non-stationary policies for stationary infinite-horizon markov decision processes. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1826–1834, 2012.

[193] Olivier Sigaud et Olivier Buffet. *Markov decision processes in artificial intelligence.* John Wiley & Sons, 2013.

[194] Dan Simon. *Optimal State Estimation : Kalman, H Infinity, and Nonlinear Approaches.* Wiley & Sons, 2006.

[195] Le Song, Jonathan Huang, Alex Smola, et Kenji Fukumizu. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *International Conference on Machine Learning (ICML)*, pages 961–968, 2009.

[196] Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1) :9–44, 1988.

[197] Richard S. Sutton et Andrew G. Barto. *Reinforcement Learning : An Introduction.* The MIT Press, 1998.

[198] Richard S. Sutton, Hamid R. Maei, Doina Precup, Shalabj Bhatnagar, David Silver, Csaba Szepesvári, et Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *International Conference on Machine Learning (ICML)*, pages 993–1000, New York, NY, USA, 2009. ACM.

[199] Richard S Sutton, A Rupam Mahmood, et Martha White. An Emphatic Approach to the Problem of Off-policy Temporal-Difference Learning. *arXiv preprint arXiv :1503.04269*, 2015.

[200] Richard S. Sutton, David A. McAllester, Satinder P. Singh, et Yishay Mansour. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Neural Information Processing Systems (NIPS)*, pages 1057–1063, 1999.

[201] Umar Syed et Robert E Schapire. A game-theoretic approach to apprenticeship learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1449–1456, 2007.

[202] Umar Syed et Robert E. Schapire. A reduction from apprenticeship learning to classification. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2253–2261, 2010.

[203] Csaba Szepesvári. Algorithms for reinforcement learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 4(1) :1–103, 2010.

[204] Istvan Szita, Viktor Gyenes, et Andras Lorincz. Reinforcement Learning with Echo State Networks. In Springer, editor, *International Conference on Artificial Neural Networks (ICANN)*, 2006.

[205] Manel Tagorti et Bruno Scherrer. On the Rate of Convergence and Error Bounds for LSTD(λ). In *International Conference on Machine Learning (ICML)*, 2015.

[206] Pham Dinh Tao et Le Thi Hoai An. Convex analysis approach to dc programming : Theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1) :289–355, 1997.

[207] Pham Dinh Tao et Le Thi Hoai An. The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. *Annals of Operations Research*, 133(1-4) :23–46, 2005.

[208] Ben Taskar, Vassil Chatalbashev, Daphne Koller, et Carlos Guestrin. Learning structured prediction models : A large margin approach. In *International Conference on Machine learning (CIML)*, pages 896–903. ACM, 2005.

[209] Gavin Taylor et Ron Parr. Value function approximation in noisy environments using locally smoothed regularized approximate linear programs. In *Uncertainty in Artificial Intelligence (UAI)*, 2012.

[210] P Thomas, Georgios Theocharous, et Mohammad Ghavamzadeh. High confidence off-policy evaluation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[211] Robert Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society*, 58(1) :267–288, 1996.

[212] J.N. Tsitsiklis et B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5) :674–690, 1997.

[213] Rudolph van der Merwe. *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models.* PhD thesis, OGI School of Science & Engineering, Oregon Health & Science University, 2004.

[214] Vladimir NN Vapnik. *Statistical learning theory.* Wiley, 1998.

[215] Christopher J. Watkins et Peter Dayan. Q-learning. *Machine Learning*, 8 :279–292, 1992.

[216] Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, et Kai Yu. The hidden information state model : A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, 24(2) :150–174, 2010.

[217] Steve Young, Jost Schatzmann, Karl Weilhammer, et Hui Ye. The hidden information state approach to dialog management. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4. IEEE, 2007.

[218] Huizhen Yu. Convergence of least-squares temporal difference methods under general conditions. In *International Conference on Machine Learning (ICML)*, 2010.

[219] Huizhen Yu et Dimitri P. Bertsekas. Q-Learning Algorithms for Optimal Stopping Based on Least Squares. In *European Control Conference*, Kos, Greece, 2007.

[220] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, et Anind K Dey. Maximum Entropy Inverse Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*, pages 1433–1438, 2008.

[221] Hui Zou et Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 67(2) :301–320, 2005.