Rapport sur le mémoire de thèse présenté par Hector KOHLER:

« Interpretability, Decision Trees, and Sequential Decision Making »

Hector Kohler présente dans son mémoire de doctorat ses travaux autour de l'interprétabilité dans le cadre de la prise de décision séquentielle, en s'intéressant plus particulièrement aux politiques représentées sous la forme d'arbres de décision. Après de premiers chapitres généraux, il présente ses contributions en trois parties : 1. la première étudie l'apprentissage par renforcement des politiques directement sous la forme d'arbres de décision, montrant que cette approche est difficile, et ce, du fait du caractère partiellement observable du problème considéré ; 2. la deuxième montre, à l'inverse, que la prise de décision séquentielle (par programmation dynamique) peut être adaptée pour l'inférence d'arbres de décision, ici à des fins de classification ; et 3. la troisième revient sur la question de l'interprétabilité des politiques pour proposer une méthode d'évaluation de l'interprétabilité « in silico » (sans utilisateurs). Je reviens maintenant plus précisément sur chaque chapitre.

Chapitre 1 - Introduction Monsieur Kohler commence ici avec une brève introduction à la prise de décision séquentielle comme problématique de la vie courante, et abordable informatiquement par exemple par apprentissage par renforcement. Le besoin est souligné, dans des cas tels que des scénarios médicaux ou militaires, de solutions qui soient interprétables par l'utilisateur final humain. Monsieur Kohler introduit ainsi la problématique générale de son doctorat : Comment chercher efficacement des solutions dans des formes plus accessibles à l'humain, telles que des arbres de décision, plutôt que de plus usuels réseaux de neurones.

Chapitre 2 - Related work Dans ce second chapitre, Monsieur Kohler rapporte des travaux antérieurs sur l'apprentissage automatique interprétable en général. Il distingue l'*interprétabilité globale*, qui vise à apprendre des modèles compréhensibles (des programmes dont on puisse vérifier le fonctionnement), et l'*interprétabilité locale* (ou explicabilité), pour laquelle, à l'exécution, un processus va chercher à identifier des explications *post hoc* aux résultats fournis par un modèle boîte noire.

Les approches globales sont ensuite séparées en deux catégories. Les approches indirectes, les plus courantes, apprennent d'abord un modèle non-interprétable tel qu'un réseau de neurones avant d'apprendre (par imitation) un modèle interprétable. Les approches directes apprennent directement un modèle interprétable, tâche difficile parce que ces modèles sont généralement non-différentiables, mais qui peut obtenir de meilleures solutions. Les paragraphes suivants décrivent des mises en œuvre d'approches indirectes dans le cadre de l'apprentissage par renforcement.

Élargissant un peu la discussion, ce chapitre se termine en évoquant deux points particuliers : une stratégie complémentaire consistant à ajouter des régularisations visant à favoriser l'interprétabilité du modèle appris, et le sujet de la détection d'anomalies telles que des objectifs mal spécifiés, détection que des solutions interprétables favorisent.

Ce court chapitre esquisse de manière assez claire le paysage de l'interprétabilité en apprentissage automatique (et plus particulièrement par renforcement), préparant le terrain pour les chapitres plus techniques à venir.

Chapitre 3 - Technical Preliminaries Dans ce troisième chapitre, Monsieur Kohler pose les bases théoriques qui vont servir tout au long du mémoire. Il décrit d'abord le cadre des arbres de décision, puis un algorithme d'apprentissage type (CART). Viennent ensuite les processus de décision markoviens, illustrés par un monde-grille jouet. Des approches de résolution sont présentées dans des contextes de plus en plus difficiles : 1. d'abord celui de la planification probabiliste, en supposant le modèle (dynamique et

fonction de récompense) connus, et les états et actions énumérables de manière exhaustive, 2. puis celui de l'apprentissage par renforcement, en supposant le modèle non connu, mais des interactions possibles avec le monde réel ou un simulateur, et 3. enfin celui de l'apprentissage par renforcement profond, où l'on ne peut raisonner sur chaque état (voire chaque action) parce qu'ils sont trop nombreux, voire indénombrables, d'où le besoin d'approximateurs tels que des réseaux de neurones profonds. Ces sections décrivent des algorithmes reposant sur des critiques et/ou des acteurs. Enfin sont abordées les bases de l'apprentissage par imitation, ce qui aboutit à une section mettant en œuvre ces préliminaires pour apprendre un arbre de décision, illustrant ainsi certains détails pratiques.

Ce chapitre fournit de manière très pédagogique un bagage tout à fait pertinent pour ce manuscrit, et permet d'introduire clairement le plan général du manuscrit.

Partie I - A Difficult Problem : Reinforcement Learning of Decision Tree Policies

Chapitre 4 - Introduction Dans ce quatrième chapitre, Monsieur Kohler distingue d'abord, parmi les approches directes pour l'apprentissage par renforcement d'arbres de décision, celles (les plus courantes) qui reposent sur des arbres paramétriques (à structure fixe mais à seuils réglables), et celles qui reposent sur des arbres non-paramétriques. Cette première partie va s'appuyer sur les travaux de Topin et al., les seuls entrant dans cette dernière catégorie. Ils ont introduit le formalisme des *iterative bounding MDP* (IBMDP) décrit ici, et dans lequel le MDP de base est complété 1. par des observations représentant des intervalles de valeurs possibles pour chaque variable d'état, et 2. par des actions de collecte d'information pour chaque variable d'état. Les politiques sont alors contraintes à ne choisir les actions qu'en fonction des observations, ce qui les rend équivalentes à des arbres de décision.

Ce chapitre complète ainsi l'état de l'art et les préliminaires techniques pour préparer le lecteur plus spécifiquement à cette première partie. Des interrogations sont soulevées quand aux liens avec les MDP partiellement observables sur lesquels Monsieur Kohler reviendra ultérieurement. L'exemple illustratif fourni est bienvenu pour aider à la compréhension de ce nouveau formalisme un peu particulier.

Chapitre 5 - Direct reinforcement learning of decision tree policies Dans ce cinquième chapitre, Monsieur Kohler explique d'abord comment DQN et PPO sont adaptés pour des IBMDP (algorithmes *modified DQN* (mDQN) et *modified PPO* (mPPO)), l'objectif du chapitre étant de reproduire des expérimentations de Topin et al. comparant leur approche directe avec une approche indirecte.

Les algorithmes comparés sont : mDQN, mPPO, DQN, PPO, mais aussi DQN+VIPER, DQN+Dagger, et PPO+Dagger, ces trois derniers apprenant des arbres de décision par imitation. Pour compenser l'absence de détails dans l'article de référence de Topin et al., plusieurs variantes de mDQN et mPPO sont considérées, par exemple avec des fonctions d'activation soit tanh, soit relu.

Les résultats expérimentaux sur le problème CartPole montrent d'abord que les approches reposant sur PPO obtiennent globalement de nettement meilleurs résultats que celles reposant sur DQN. On voit aussi que les algorithmes modifiés s'avèrent généralement moins efficaces que les algorithmes originaux, que ces derniers soient employés sur le MDP de base ou l'IBMDP. En outre, les algorithmes modifiés, s'ils peuvent trouver d'assez bonnes politiques, ne le font pas de manière assez régulière, à l'inverse des algorithmes originaux. L'observation des arbres de décision obtenus par les différentes approches montre que les algorithmes modifiés génèrent souvent des arbres dégénérés répétant tout le temps la même action alors que les meilleures solutions sont d'assez petits arbres.

Les résultats expérimentaux de Topin et al. sont ici confirmés et approfondis. C'est aussi l'occasion d'observer de manière concrète les difficultés de cette approche particulière.

Chapitre 6 - Limits of direct reinforcement learning of decision tree policies Dans ce sixième chapitre, Monsieur Kohler introduit les *MDP partiellement observables* (POMDP) pour montrer qu'on peut transformer tout IBMDP en un *IBMDP partiellement observable* (POIBMDP), classe de problèmes qui s'avère être une sous-classe des POMDP. Il justifie ainsi que résoudre un IBMDP revient à trouver une politique partiellement observable déterministe dans un POMDP particulier dont la fonction d'observation révèle certaines variables d'état. Pour mieux étudier le sujet, des politiques (arbres de décision) spécifiques

sont construites et évaluées sur un petit monde-grille pour différentes valeur du coût unitaire de collecte d'information ζ, ce qui permet de connaître une politique optimale pour chaque valeur de $\zeta \in]-1;+1[$.

Avoir mieux identifié le problème considéré permet de trouver dans la littérature des résultats de complexité (disant que le problème est NP-difficile) et des approches de référence, dites d'apprentissage par renforcement assymétrique. Ces approches sont expérimentées sur le monde-grille précédemment cité (en faisant varier leurs paramètres), montrant qu'aucune ne trouve assez fréquemment les solutions optimales pour les valeurs intéressantes de ζ ($\in]0;1[$), même si les approches assymétriques s'avèrent nettement meilleures que les algorithmes de base.

En comparaison avec les études précédentes, celle-ci permet une compréhension plus fine des comportements des algorithmes en ayant pour référence des solutions optimales connues, et en étudiant les politiques obtenues, ce qui permet de constater que les optima locaux atteints correspondent souvent à des arbres très simples.

Chapitre 7 - Direct RL of decision tree policies for classification tasks Dans ce septième chapitre, Monsieur Kohler observe d'abord qu'une tâche de classification peut être vue comme un MDP dans lequel 1. les états correspondent aux données échantillonnées ; 2. les actions correspondent aux classes ; 3. les transitions entre états sont uniformément aléatoires ; et 4. les récompenses dépendent de la bonne action de classification de l'état courant. De là, il explique comment construire un POIBMDP pour la classification, et que l'apprentissage par renforcement « classique » va ici fonctionner parce que la transition d'un échantillon à l'autre est indépendante de l'échantillon précédent et de l'action de classification choisie. Des résultats expérimentaux sur un problème de classification simple (partageant une structure similaire avec l'exemple principal du chapitre précédent) confirment clairement cette analyse, les trois algorithmes de référence obtenant de bien meilleurs résultats qu'auparavant.

En abordant le cas particulier des MDP pour la classification, ce chapitre souligne que la difficulté des chapitres précédents est liée aux transitions entre états de base et donc à la probabilité de présence dans l'un ou l'autre.

Partie II - An Easier Problem : Decision Tree Induction as Solving MDPs

Chapitre 8 - Introduction Dans ce huitième chapitre, Monsieur Kohler introduit ses travaux visant à proposer un nouvel algorithme pour la classification à l'aide d'arbres de décision. Il explique d'abord que des algorithmes gloutons sous-optimaux sont souvent employés (tels que CART), les algorithmes optimaux étant très coûteux, et que sa contribution trouve sa place entre ces deux extrêmes. Une revue de l'état de l'art analyse plus précisément CART, lequel cherche à chaque itération à « couper » au mieux les données, puis plusieurs algorithmes optimaux, généralement limités à des caractéristiques binaires ou à des arbres peu profonds. Ce chapitre introductif se termine avec une définition formelle précise du problème de classification par arbres de décisions qui va être considéré, en évoquant une variété d'autres problèmes étudiés dans la littérature.

Chapitre 9 - Decision tree induction as solving an MDP Dans ce neuvième chapitre, Monsieur Kohler formule tout d'abord le problème de l'induction d'un arbre de décision comme la construction et la résolution d'un MDP. Au sein de ce MDP, l'état courant représente les hypothèses restantes après un certain nombre de tests. Les actions disponibles sont des tests choisis de manière heuristique, ici en extrayant les tests au sein d'un arbre construit par l'algorithme glouton CART. Une fois la construction du MDP effectuée, une programmation dynamique simple est effectuée, suivie d'une extraction d'arbre de décision. Il est démontré que cette approche fait nécessairement aussi bien qu'un algorithme glouton, et même mieux avec forte probabilité sur certains problèmes. Le chapitre se conclut en discutant l'implémentation pratique de DPDT, en évoquant le coût (non négligeable) de construction du MDP et la complexité spatiale qui conduit à préférer l'utilisation d'une recherche en profondeur d'abord. Ce dernier choix nous rapproche peut-être d'un algorithme de type diviser-pour-régner plus que de programmation dynamique.

Chapitre 10 - Dynamic programming decision trees in practice Dans ce dixième chapitre, Monsieur Kohler présente dans un premier temps des expérimentations comparant DPDT avec l'algorithme glouton

CART, l'algorithme optimal Quant-BNB, et l'algorithme « intermédiaire » Top-B, et ce sur un ensemble de problèmes de classification différents. DPDT et Top-B obtiennent des résultats assez comparables en ce qu'ils fournissent des classifications généralement proches de l'optimum, mais avec des coûts computationnels très inférieurs. Une deuxième expérience vise à évaluer les capacités de généralisation de DPDT dans une situation où ses paramètres (et ceux des autres algorithmes considérés : ici CART et STreeD) sont progressivement optimisés. Cette fois sont distingués d'une part des problèmes de classification catégorielle et d'autre part des problèmes de classification numérique. Dans les deux cas DPDT donne de très bons résultats à relativement faible coût computationnel. Enfin, une troisième expérience illustre la possibilité de pratiquer du boosting (différentes approches de références étant considérées) reposant sur DPDT, avec de meilleurs modèles obtenus qu'en s'appuyant sur CART.

Ce chapitre conclut la présentation du travail de Monsieur Kohler sur l'utilisation de la programmation dynamique pour l'inférence d'arbres de décision. L'approche proposée est simple et a de bonnes propriétés théoriques comme expérimentales. Elle peut probablement servir de base à des travaux futurs, des extensions. Une question soulevée est par exemple celle du choix des règles de coupe, même si l'exploitation de CART s'avère très efficace.

Partie III - Beyond Decision Trees : Evaluation of Interpretable Policies

Chapitre 11 - Introduction Dans ce onzième chapitre, Monsieur Kohler soulève le problème de l'évaluation des politiques interprétables. Il souligne la facilité de comparer « in silico » des modèles de la même classe, typiquement en comparant leurs nombres de paramètres, la difficulté principale se situant plutôt dans la comparaison inter-classe. La notion de simulabilité a été introduite pour cela, et parfois approchée par l'étude des complexités spatiales et temporelles des programmes, mais l'utilisation de langages et supports de calculs différents (CPU vs GPU) complique la comparaison. Les chapitres suivants vont proposer une démarche d'évaluation sur la base 1. du temps d'inférence ces politiques, et 2. de la taille des politiques en proposant une représentation simple et adaptée à l'humain permettant de les comparer.

Chapitre 12 - Validating our methodology Dans ce douzième chapitre, Monsieur Kohler présente plus précisément l'étude empirique annoncée, dans laquelle divers experts servent à apprendre des politiques de diverses classes (politiques linéaires, arbres de décision obliques ou non, réseaux de neurones relu) avec divers algorithmes d'apprentissage par imitation (Behavior Cloning, DAgger et VIPER) sur divers problèmes, l'ensemble des combinaisons valides (certaines combinaisons d'algorithmes sont impossibles) fournissant 40 000 politiques à étudier. Les premiers résultats d'expérience montrent que DAgger surclasse BC sur des tâches de contrôle classiques comme sur des tâches de MuJoCo, et que VIPER surclasse DAgger sur des jeux Atari. On observe aussi que les imitations ne sont proches des politiques originales que sur les tâches de contrôle classiques. Une autre analyse permet de voir que les réseaux de neurones se « débrouillent » mieux sur ces mêmes tâches de contrôle, et les arbres de décision sur les jeux Atari. Dans un dernier temps, pour étudier la simulabilité des meilleures politiques obtenues, celles-ci sont transformées en code Python uniformisé (respectant le style PEP 8) avant de comparer les nombres d'opérations exécutées et les tailles de ces politiques-programmes. On observe par exemple que les arbres de décision s'avèrent meilleurs en nombres d'opérations exécutées et les réseaux de neurones en termes de tailles, ce qui souligne la complémentarité des deux critères.

Ce travail de validation illustre bien la pertinence de l'approche proposée pour mesurer et comparer la simulabilité de diverses politiques, et est d'autant mieux valorisé que les données produites ont été mises à disposition de la communauté.

Chapitre 13 - Interpretability-performance trade-offs Dans ce treizième chapitre, Monsieur Kohler aborde le sujet des compromis entre interprétabilitié et performance. Il illustre d'abord sur une sélection de problèmes représentatifs que la dimensionalité de l'espace d'états joue un rôle important, des politiques relativement interprétables pouvant donner de bons résultats en termes de récompenses cumulées sur de « petits » problèmes, ce qui s'avère impossible sur de grands problèmes. Une analyse permet de confirmer que le nombre d'états est un prédicteur prépondérant de l'interprétabilité des politiques solutions d'un problème donné, autant pour le nombre de pas que pour la taille. Les résultats obtenus montrent aussi que les

performances peuvent augmenter avec l'interprétabilité, ce qui correspond à un phénomène d'amélioration de la généralisation par régularisation de la solution. Enfin, s'appuyant sur des travaux liant, dans un réseau de neurones relu, nombre de paramètres et coût de vérification formelle de propriétés, des expérimentations sont conduites sur les mêmes politiques, en générant des requêtes de vérification aléatoires. Elles amènent à observer que, effectivement, plus un modèle est interprétable au sens de la mesure proposée dans ce manuscrit, plus les temps de vérification de non-satisfaction de requêtes sont faibles.

Ce chapitre éclaire donc de diverses manières la mesure d'interprétabilité proposée par Monsieur Kohler, confirmant sa pertinence comme la pertinence de rechercher des politiques interprétables.

Conclusion – Le mémoire de Monsieur Kohler aborde la question de l'interprétabilité en prise de décision séquentielle sous plusieurs angles, faisant ressortir trois contributions principales d'ordres distincts. Dans chaque cas le sujet abordé est présenté clairement et accompagné d'un état de l'art méthodique qui conduit naturellement à la proposition d'une contribution. Si elles reposent principalement sur des validations expérimentales, la compréhension et l'analyse théorique ne sont pas négligées. En outre l'ensemble du mémoire est organisé et rédigé de manière claire et pédagogique.

Pour toutes ces raisons, je donne un avis favorable et sans réserve à la soutenance d'Hector Kohler en vue de l'obtention du grade de docteur de l'université de Lille dans la discipline informatique.

À Nancy, le 19 novembre 2025,

Olivier Buffet
Chargé de recherche INRIA, HDR