INFERENCE CAUSALE ET BIAIS

DANS L'APPRENTISSAGE AUTOMATIQUE

Mémoire de fin d'études d'Odelin TAMAYO, août 2020

Diplôme d'ingénieur systèmes d'information,
spécialité informatique décisionnelle

EPISEN – École publique d'ingénieurs de la santé et du numérique

Eric PIPARD – Tuteur pédagogique, EPISEN

Viviane LE NAOUR – Maître d'apprentissage, Natixis SA

INFERENCE CAUSALE ET BIAIS

DANS L'APPRENTISSAGE AUTOMATIQUE

Mémoire de fin d'études d'Odelin TAMAYO

Diplôme d'ingénieur systèmes d'information,
spécialité informatique décisionnelle

EPISEN – École publique d'ingénieurs de la santé et du numérique

Eric PIPARD – Tuteur pédagogique, EPISEN

Viviane LE NAOUR – Maître d'apprentissage, Natixis SA

Remerciements

Je tiens à remercier toutes les personnes qui m'ont accompagné et soutenu durant mes études.

Merci à l'ensemble de l'équipe pédagogique de l'École publique d'ingénieurs de la santé et du numérique de m'avoir formé à différents domaines de l'informatique, la logique et aux responsabilités d'un ingénieur, ainsi que de m'avoir ouvert à d'autres thèmes tels que la philosophie des sciences, les sciences humaines, les sciences de gestion.

Merci à Olivier Le Briand, qui m'a porté dans un environnement où j'ai pu acquérir une grande part de ma culture informatique, dont une bonne vision du cycle de vie d'applications.

Merci à Viviane Le Naour de m'avoir offert l'opportunité de travailler proche du besoin sur des questions métier variées, ainsi que de découvrir des clés de lecture des enjeux réglementaires et organisationnels d'une banque.

Merci à Slim Cheikh Rouhou, pour ses conseils, son expérience et son accompagnement rigoureux de mon travail au quotidien.

Merci à ma mère pour ses conseils que je devrais davantage écouter.

Merci à mon frère et à mon père, toujours présents pour m'aider.

Merci à l'équipe de l'escalade, particulièrement Tony, sur qui on peut toujours compter.

Abstract

Français

L'inférence de causalité à partir de données est une question majeure en statistiques. La causalité est une part importante de la cognition humaine. De leur côté, les algorithmes d'apprentissage automatique sur lequel reposent les intelligences artificielles se basent essentiellement sur des corrélations, des liens statistiques entre des variables. Corrélation n'est pas causalité, et cette distinction est parfois présentée comme l'un des freins majeurs au développement des intelligences artificielles. Un type d'apprentissage automatique se distingue des autres sur la question. Il s'agit de l'apprentissage par renforcement, qui consiste à laisser un algorithme (un agent) apprendre en interagissant avec son environnement. En l'appliquant à des situations biaisées, on peut donc observer dans quelle mesure un algorithme est sensible à ces biais, donc dans quelle mesure cet algorithme est capable de différencier une corrélation d'une causalité. La première expérience a quantifié l'intérêt de trop contraindre un agent sur le court terme et le risque que cela occasionne dans certaines conditions sur le long terme. La seconde montre la difficulté d'un agent à appréhender une relation de causalité avec une variable qu'il ne peut observer lorsque cette dernière évolue.

English

Causality inference from data is a major issue in statistics. Causality is an important part of human cognition. On the other hand, the automatic learning algorithms used for artificial intelligences are based are essentially based on correlations, statistical links between variables. Correlation is not causality, and this distinction is sometimes presented as one of the major obstacles to the development of artificial intelligence. One type of machine learning stands out from the others on the issue. This is reinforcement learning, which consists in letting an algorithm (an agent) learn by interacting with its environment. By applying it to biased situations, we can therefore observe how sensitive an algorithm is sensitive to these biases, and therefore to what extent it is able to differentiate a correlation from a causality. The first of the experiments quantified the benefit of over-constraining an agent in the short term, and the risk that this entails under certain conditions in the long term. The second shows the difficulty of an agent in understanding a causal relationship with a variable that it cannot observe when this variable evolves.

Table des matières

Introduction

A- Réflexion

L'idée de ce sujet vient d'une réflexion sur la différence entre corrélation et causalité. Puisqu'une grande part de la cognition humaine est basée sur la notion de causalité, tandis que les algorithmes sont basés sur des corrélations, l'inférence causale est une question majeure en statistiques. Ce problème est parfois posé comme un des freins majeurs au développement des intelligences artificielles.

Les productions humaines qui s'appuient sur l'apprentissage automatique peuvent également subir ce biais, ne serait-ce que parce que le biais vient des données en entrée, et ce, malgré une validation croisée.

Quelle réponse peut-on apporter à ce biais ?

Dans le monde réel, une méthode souvent appliquée est la randomisation : elle consiste à couper aléatoirement un échantillon en deux, puis à agir sur seulement l'une des moitiés, et enfin à comparer les résultats entre les deux groupes.

Est-ce applicable à l'apprentissage automatique ?

Il faut pour cela que l'algorithme puisse agir sur son environnement pour apprendre non seulement à partir de données initiales, mais aussi à partir du résultat de ses actions. Il s'agit justement de la définition de l'apprentissage par renforcement, un des types d'apprentissage automatique.

L'apprentissage par renforcement, en agissant sur son environnement, est-il capable de différencier une corrélation d'une causalité ?

On peut lister différentes situations biaisées et observer dans quelle mesure un algorithme d'apprentissage par renforcement (un agent) tombe dans ces pièges. Autrement dit, il s'agit d'étudier différentes situations biaisées qui rendent difficile pour cet agent d'atteindre son objectif.

Les différents cas de corrélation

Cas théoriques de corrélation et causalité

Dans le cas où A est corrélé à B, il peut y avoir plusieurs explications :

- A cause B
- B cause A
- A cause B et B cause A
- C connu cause A et B
- C inconnu cause A et B
- Coïncidence.

Si on ne peut observer C, peut-on différencier le cas 5 du cas 6 ? Peut-on estimer que nos données semblent répondre à une variable supplémentaire inconnue à partir d'une certaine quantité de données permettant d'écarter l'idée d'une coïncidence ? Si oui, peut-on mesurer cette variable ? Ou bien la différencier avec le vrai bruit statistique ?

Illustration de ces cas

Prenons le cas où A, la quantité vendue, est corrélé à une variable B

- La quantité vendue cause le résultat net
- Le prix de vente cause la quantité vendue
- Le nombre de ventes cause la fréquentation à venir, qui cause le nombre de ventes à venir
- Les jours d'affluence causent la quantité vendue d'un produit et celle d'un autre produit
- La complémentarité entre la farine et la levure cause une corrélation entre les quantités vendues de l'un et de l'autre
- Une variable n'ayant aucun lien de causalité avec quoi que ce soit (exemple : l'horoscope des sagittaires) peut néanmoins se retrouver corrélé avec d'autres variables si on ne dispose pas d'un échantillon suffisamment grand.

B- Plusieurs approches

a) Effet cigogne intrinsèque aux algorithmes

La définition formelle en statistique du terme biais, "différence entre la valeur de l'espérance d'un estimateur et la valeur qu'il est censé estimer" correspondrait à un problème de régression.

Pour un algorithme d'apprentissage par renforcement, l'objectif n'est pas tant d'estimer au plus proche une variable que d'en maximiser une. Cela dit, comme les algorithmes d'apprentissage automatique utilisent des modèles statistiques, ils sont aussi sensibles à ces biais et soumis au dilemme biais-variance. Cela peut déjà poser plusieurs questions :

- Peut-on utiliser l'apprentissage par renforcement pour différencier des corrélations de causalités ? Pour prouver l'existence de causalités ?
- Dans quelle mesure ce biais influence-t-il le résultat de prédictions basées sur l'apprentissage automatique ?
- En cas de correction d'un modèle biaisé, avec quelle inertie les algorithmes apprennent de leurs erreurs ?

b) Biais induits par l'humain dans son utilisation de la technologie

Comme nous l'avons vu, la corrélation est une considération statistique, tandis que la causalité appartient au réel. Un algorithme d'apprentissage automatique n'a donc pas nécessairement besoin de la notion de causalité pour répondre à sa fonction.

C'est après des décisions humaines telles que le paramétrage de l'algorithme, la décision de considérer telle ou telle variable comme un objectif à atteindre, que la solution finira par utiliser des corrélations comme s'ils s'agissait de causalités.

c) Comparaison entre types d'apprentissage automatique

Une approche pour isoler dans quelle mesure l'apprentissage par renforcement distingue une corrélation d'une causalité serait de soumettre ses données d'apprentissage à un algorithme d'apprentissage automatique différent.

Pour cela, on pourrait imaginer enregistrer toutes les expériences réalisées par un agent pour apprendre, et utiliser ensuite ces dernières pour un apprentissage supervisé.

Cependant, les résultats d'un apprentissage supervisé dépendent énormément de différents paramètres liés à l'environnement. En paramétrant l'environnement et l'algorithme d'une certaine manière, l'humain utilise de la connaissance du contexte que l'agent n'aurait pas eu seul. Cela peut optimiser les résultats du programme, voire ajouter des biais de cette façon.

d) Implémenter une logique de randomisation

Pour éviter qu'un algorithme d'apprentissage par renforcement soit soumis à des biais, peut-on appliquer une solution analogue à celle que les humains font sur le réel ?

Peut-on développer un tel algorithme ou forcer un algorithme existant à suivre des principes ?

Quel impact cela aura-t-il sur ses résultats ?

L'ajout d'une randomisation devrait ralentir l'agent dans son analyse et ralentir sa progression dans les situations non biaisées.

L'agent devrait également être ralenti dans une situation biaisée si les algorithmes actuels sont suffisants pour appréhender la différence entre corrélation et causalité pour les mêmes raisons.

Cette expérimentation pourrait même être affinée en la combinant à une situation de causalité impliquant des paramètres que l'agent ne peut pas tous observer.

I- Revue de littérature

A- L'inférence causale

L'inférence causale est un problème majeur en statistiques. La cognition humaine est basée sur des liens de causalité, tandis que les statistiques se basent sur des corrélations, des liens statistiques entre des variables. Des approches existent et parviennent à reconnaître statiquement les corrélations les plus susceptibles d'être des causalités à partir de données "à plat". Certaines de ces approches constituent des avancées récentes.

D'après cette étude de 2018, il y a beaucoup d'approches dans beaucoup de champs de recherche, mais il n'existe pas pour l'heure d'outil statistique commun largement répandu entre les disciplines.

Dans le cas de l'épidémiologie, on dispose souvent de grandes quantités de données, mais il est parfois difficile de prouver l'existence de causalité. C'est pourquoi on préfère décrire les corrélations entre causes probables comme des facteurs de risque, plus que comme des causes de certaines maladies.

a) Modélisation graphique

Les relations de causalité entre des phénomènes peuvent être modélisées par un graphe orienté acyclique entre les variables qui mesurent ces phénomènes.

Les variables observables ou non sont représentées par des sommets. Une arête allant d'un sommet X à un sommet Y signifie que X cause Y. Seules les relations entre les variables observables sont représentées par des flèches pleines. Les autres relations peuvent rendre le graphe cyclique.

Cet outil est très largement utilisé pour modéliser des relations de causalité, par exemple entre des facteurs de risque.

b) Exemple d'algorithme d'inférence causale : Greedy Equivalence Search

Le graphe de causalité est une manière de modéliser les relations de causalité entre plusieurs variables. Pour établir ce modèle, plusieurs algorithmes existent. Greedy Equivalence Search en fait partie. Il sert à calculer le plus probable graphe de causalité bayésienne entre un ensemble de variables. Pour cela, il part d'un graphe vide et y ajoute au fur et à mesure les arêtes qui augmentent son score bayésien, puis retire les moins utiles.

c) L'apprentissage par renforcement pour créer le graphe de causalité

En 2020, des chercheurs ont obtenu des résultats prometteurs en utilisant l'apprentissage par renforcement pour produire des graphes de causalité.

Leur agent est basé sur l'algorithme asynchronous advantage actor-critic. Il doit générer des graphes de causalité à partir de données réelles et fictives, puis est récompensé en fonction de la précision et de la véracité des graphes produits.

Finalement, leur agent a réussi à produire des graphes avec le score bayésien optimal.

B- Une question triviale dans le cas de l'apprentissage automatique ?

La question de la confusion entre corrélation et causalité dans le cas de l'apprentissage automatique peut être considérée comme triviale du point de vue des algorithmes. En effet, ceux-ci ne sont pas basés sur les causalités, qui sont des phénomènes réels, mais seulement des corrélations, qui sont des liens statistiques. Selon cette considération, c'est au moment où ces algorithmes sont utilisés au sein d'une application que l'intelligence humaine (des développeurs, des analystes) va biaiser la situation.

Nous allons cependant analyser ce postulat au regard des spécificités des différentes types d'apprentissage automatique :

a) Réduction de biais dans l'apprentissage automatique

Une grande partie des biais peuvent être réduits par une validation croisée, consistant à :

- Couper l'échantillon en deux de façon aléatoire
- Apprendre à partir d'une partie des données
- Vérifier la qualité de l'apprentissage en regardant dans quelle mesure le modèle se trompe sur les données restantes.

Une autre partie des biais dépend directement de la qualité des données en entrée. Si les données sont elles-mêmes non représentatives de la situation, une validation croisée ne résoudra pas le problème.

Le cas de la causalité

Dans le cas où confusion entre corrélation et causalité il y a, ça n'est pas au niveau des algorithmes d'apprentissage automatique, qui n'étudient que les corrélations. Comme nous l'avons vu plus tôt, c'est lors de l'utilisation de ces algorithmes qu'il peut y avoir confusion.

Puisque les modèles consistent à mesurer les corrélations entre toutes les variables afin d'en estimer une à partir des autres, il suffit que certaines variables soient corrélées avec celle à estimer pour biaiser les résultats. Pour combattre ce biais, il faut qu'un être humain analyse le contexte pour déterminer s'il y a causalité entre les variables.

b) Dans le cas de l'apprentissage par renforcement

Si le cas de la régression semblait trivial, c'est peut-être parce que l'algorithme n'a pas l'occasion d'interagir avec son environnement pour tester ce qui est une corrélation et ce qui est une causalité.

Cependant, peut-on utiliser des algorithmes d'apprentissage par renforcement pour trouver ou prouver des relations de causalité entre des variables ?

Pour les variables de ses actions, on peut se dire que oui : il suffit de le laisser agir pour qu'il fasse ou non varier les autres observations et la récompense.

Cependant, si la récompense augmente et que les observations évoluent de façon linéaire au fil du temps, cela ne prouve pas forcément que les actions font s'améliorer la récompense. Il est possible que la seule relation de causalité soit non pas entre les actions d'un côté, la récompense et les observations de l'autre, mais entre le temps et ces dernières.

On peut d'ailleurs différencier trois types de variables qui construisent la réalité avec laquelle interagit un agent :

- Les données intrinsèques à l'environnement (une partie des observations)
- Les entrées (les actions de l'agent sur l'environnement)
- Les sorties issues des actions sur l'environnement (la récompense et une partie des observations).

Au sein des librairies que nous utiliserons, ces variables sont découpées de la sorte :

- Les observations (qu'elles dépendent des actions de l'agent ou non)
- Les actions (les décisions prises par l'agent)
- La récompense (ce que l'agent doit maximiser).

Ainsi, contrairement aux autres types d'apprentissage automatique, il est déterminé dès le départ sur quoi l'agent peut agir. L'agent ne peut pas vérifier l'existence de causalité entre deux variables qui ne dépendent pas de lui.

On pourrait se dire que l'algorithme explorera l'environnement à sa disposition pour maximiser sa récompense indépendamment de toute notion de causalité. Cependant, il reste possible d'expérimenter si et dans quelle mesure l'algorithme tombe dans des biais.

En 2019, une étude a montré que dans des situations où des causalités existent entre des variables, des algorithmes d'apprentissage par renforcement comprenant une modélisation de la causalité obtiennent de meilleurs résultats (dans 2 expériences sur 3) que les algorithmes qui ne sont pas basés sur un modèle particulier (model-free reinforcement learning).

c) Exemples réels de logiciels biaisés

De nombreux exemples d'applications biaisées de l'apprentissage automatique sont connus du grand public. Le cas de l'application de sélection de CV d'Amazon est un cas assez représentatif de ces exemples.

Dans ce système, qui défavorisait les profils féminins, l'erreur venait du fait qu'on ait utilisé des données biaisées (recrutements ayant eu lieu durant 10 années).

Derrière l'expression "l'application défavorise les profils féminins", on pourrait comprendre "l'algorithme suppose qu'il y a une causalité entre le genre d'un individu et ses compétences". Or, il s'agit davantage de "l'algorithme observe une corrélation entre le genre des individus et leur proportion au sein des recrutements à laquelle il lui a été demandé de correspondre".

Le problème n'est donc pas intrinsèque à l'algorithme, mais dû aux données en entrée et à la décision de partir de ces données pour obtenir ce résultat. Il s'agit d'un biais présent dans les données d'apprentissage, ainsi que d'une erreur humaine de confusion entre corrélation et causalité, les concepteurs ayant supposé que les choix de recrutement passés étaient des décisions sinon optimales, au moins de bons exemples vers lesquels tendre, alors que ces choix étaient marqués d'erreurs humaines.

C'est principalement le même problème de données non représentatives ou de métrique à maximiser discutables qui causent les problèmes dont on entend parler :

- La reconnaissance faciale qui résiste mal face à de nombreux cas particuliers car peu représentés dans les données d'apprentissage.
- Les recommandations de youtube, qui favorisent les contenus conspirationnistes car les individus qui les visionnent passent beaucoup de temps sur la plateforme. Or, c'est justement ce temps passé sur la plateforme que doivent maximiser les algorithmes de youtube.
- Tay, le chatbot qui devait apprendre en échangeant avec des internautes sur les réseaux sociaux, que des internautes se sont amusés à rendre raciste et misogyne en lui faisant interagir dans des conversations qui l'étaient.

C- Exemples de biais possibles dans l'apprentissage par renforcement

a) Biais induits au sein des algorithmes

Quand un agent est développé pour répondre à un besoin spécifique, on peut être tenté de le paramétrer via des connaissances préexistantes afin d'améliorer ses résultats. Cela peut causer des erreurs supplémentaires, en plus de rendre l'algorithme moins généralisable.

Exemple de ce problème dans notre expérimentation

Prenons l'exemple où notre agent doit acheter des marchandises puis les vendre, avec comme récompense la marge sur coûts variables.

Pour lui éviter d'essayer des cas triviaux et a priori contreproductifs, on serait tenté de le paramétrer de telle sorte qu'il ne fixe jamais de prix de vente inférieur au prix d'achat. Cela peut sembler une bonne idée si on oublie que :

- Un produit d'appel peut être vendu à perte et pourtant améliorer le revenu global
- Si l'agent doit gérer ses stocks, il peut arriver qu'il doive vendre à perte pour déstocker (ex: péremption)
- Vendre ou acheter à perte peut parfois être une obligation légale
- Exemple : EDF qui achète à un prix plancher l'électricité issue d'énergies renouvelables sur le marché à terme de l'électricité.

b) Coïncidence

La coïncidence constitue un cas trivial mais néanmoins possible, où différentes variables sont liées statistiquement dans les observations par hasard, sans que les phénomènes observés le soient. On serait tenté de se dire qu'il s'agit là d'une question habituelle de quantité de données insuffisante.

Cependant, le fait est que l'étude des petits jeux de données est un champ de recherche à part entière, aussi bien en statistiques que dans l'apprentissage automatique. Il apparait en effet que tous les algorithmes n'apprennent pas aussi bien selon les volumétries.

Dans quelle mesure les informaticiens sont-ils formés aux conditions de volumétrie des différents algorithmes et de leurs différentes implémentations ? Ces conditions sont-elles suffisamment documentées ?

c) Données d'apprentissage non représentatives

Peut-on tromper l'agent s'il ne peut déterminer l'importance de ses actions dans la récompense finale ?

Si 80% de sa récompense est basée sur 20% de ses actions sans qu'il le sache, l'algorithme mettra plus de temps à estimer l'importance respective de chaque variable.

On peut maximiser ce biais :

- En fournissant une récompense et/ou des observations agrégées à une granularité trop épaisse
- Nous arrivons alors à une coïncidence et à un biais de surapprentissage.

Ce biais n'est pas propre à l'apprentissage par renforcement. Il ne correspond d'ailleurs pas à la définition statistique de biais, puisque le problème disparaît dès lors que l'on dispose de données suffisantes.

Données non représentatives : l'avantage de l'apprentissage par renforcement sur l'apprentissage supervisé

Voilà une publication intéressante dont le protocole peut se résumer ainsi :

- On prend deux agents, A et B, et un jeu compétitif.
- A apprend à jouer à partir de données de véritables joueurs.
- Puis B apprend à jouer contre A.

Il en résulte que la meilleure manière pour B de gagner consiste à ne pas jouer. En effet, A n'a appris à jouer que contre des personnes qui savent jouer. B faisant des choses inattendues, A perd tout seul. L'apprentissage par renforcement gagne sur le long terme face à un programme exclusivement formé sur des données qui ne recouvrent pas assez de cas.

Conclusion de l'étude : l'apprentissage par renforcement a moins de difficultés face à des données partielles en entrée.

d) Biais de confirmation

Les êtres humains qui utilisent des algorithmes pour répondre à leurs besoins peuvent induire leurs propres biais dans leur manière d'utiliser les algorithmes.

Cependant, il devrait être possible de tester si les algorithmes sont également intrinsèquement soumis à ce biais. Pour cela, on peut imaginer créer une causalité au début qui décroit avec le temps, puis observer dans quelle mesure l'algorithme continuera à supposer que la situation continuera dans le sens initial.

Biais du rasoir d'Ockham

Le biais du rasoir d'Ockham porte un nom qui peut surprendre. Il désigne le fait qu'un principe de raisonnement parfois considéré comme fondamental de la philosophie des sciences peut aussi être la source de biais. Il n'est pas directement lié au biais de confirmation, mais avec notre protocole expérimental, il devrait être facile de tester les deux en même temps.

Conclusion

En conclusion, tandis que de nombreuses approches existent sur la question de l'inférence causale, mais aucune ne fait consensus dans le domaine de l'apprentissage automatique. Le cas particulier de l'apprentissage par renforcement semble intéressant et expérimentable en plus d'être relativement peu étudié. C'est pourquoi nous l'expérimenterons.

II- Description des expérimentations

Les expérimentations ont consisté à faire interagir un agent suivant plusieurs algorithmes d'apprentissage par renforcement pour apprendre face à un environnement biaisé.

Pour créer des situations biaisées, on préfèrera utiliser une librairie permettant de créer un environnement.

J'ai choisi Tensorflow, car la documentation semble claire et bien fournie, et que la librairie implémente de nombreux algorithmes de différentes catégories. Enfin, elle semble mieux maintenue que sa principale concurrente.

Indicateur	*Keras-RL*	*Tensorflow agents*
Année de début du projet	2016	2018
Date de la dernière release	01/05/2018	29/05/2020
Durée entre les deux dernières releases	1 mois	2 mois
Nombre d'étoiles sur Github	4800	1500
Nombre de contributeurs	39	60
Nombre de commits	1352	308

Le diagramme de classes simplifié figure en annexe.

Le code, quant à lui, est accessible sur le repository suivant : https://github.com/OdelinT/Memoire

A- L'agent

a) Différents algorithmes d'apprentissage par renforcement

La liste des algorithmes d'apprentissage par renforcement présents par défaut dans la librairie Tensorflow :

- Behavioral_cloning
- DDPG : Deep Deterministic Policy Gradient

- DQN : Deep Q Network
- Categorical_DQN : un type d'agent DQN
- PPO : Proximal Policy Optimization
- Reinforce
- SAC : Soft Actor Critic
- TD3 : Twin Delayed Deep Deterministic policy gradient

Lors de la mise en place du projet de test, il s'est avéré que l'intégralité des implémentations de ces algorithmes dans la librairie Tensorflow étaient contraignantes, nécessitant des conditions spécifiques non documentées. Par exemple, l'agent DQN ne s'applique qu'à des environnements dont les actions sont des dimensions spécifiques.

La majorité de ces problèmes se sont révélés dans des messages d'erreur lors de l'application des agents à l'environnement. Pour éviter de perdre du temps à chercher à différencier ce qui pouvait être une erreur de ma part de ce qui est une contrainte non documentée, je me suis concentré dans les expérimentations sur l'agent SAC, qui est le premier à avoir donné des résultats satisfaisants. Finalement, ce dernier a lui aussi été la source de quelques difficultés, comme vous le verrez plus tard.

b) Soft Actor Critic agent

SAC est un algorithme « sans modèle » qui basé à la fois sur l'algorithme DDPG et l'optimisation stochastique. Il implémente des optimisations issues de l'agent TD3 et d'autres issues des algorithmes de Q-learning.

c) Les polices d'exécution d'un agent

Un même agent peut s'exécuter de plusieurs manières, chacune servant à répondre à un besoin différent.

On peut en distinguer quatre principales :

- La politique de base d'un agent
- Random-policy, qui consiste à effectuer des actions simplement au hasard, indépendamment d'un apprentissage précédent. Il peut être utilisé pour tester si un environnement fonctionne correctement dans de nombreuses situations sans utiliser de temps de calcul pour faire fonctionner un algorithme plus complexe, ou pour constituer un premier jeu d'observations qui peut ensuite être utilisé par d'autres algorithmes
- La politique d'exploration, pour demander à un algorithme d'explorer lui-même l'environnement, en testant des possibilités qui lui semblent les plus utiles pour approfondir son modèle
- La politique cupide (greedy), qui consiste à chercher à maximiser sa récompense à partir des observations déjà réalisées.

B- L'environnement

On peut imaginer des prix mis à jour en temps réel par l'agent, l'objectif de l'agent étant de trouver le prix maximisant le résultat net. L'environnement répondrait, pour chaque offre, à une demande (un nombre d'achats).

Exemples de cas réels qui correspondraient : prix dans un centre commercial connecté, sur un site d'e-commerce, sur un marché à terme en temps réel (financier, de l'électricité, du blé), etc.

Dans Tensorflow, on peut créer plusieurs types d'environnement. Tous prennent en compte des paramètres similaires. Dans notre exemple, l'environnement de base à partir duquel seront construits les autres aura les paramètres suivants :

- Le temps est linéaire et discret
- Action : pour chaque lieu et/ou produit, un prix de vente
- Observation : pour chaque lieu et/ou produit, une demande
- Récompense : la somme, pour chaque lieu et/ou produit, du prix de vente auquel on soustrait le prix d'achat.

C- Les biais à implémenter

La classe de l'environnement basique sera dupliquée en plusieurs versions, chacune ayant pour but de tester un biais ou une situation spécifique.

La graine des paramètres aléatoires sera la même dans tous les environnements, et on créera un test unitaire pour vérifier que les paramètres générés dans les différents environnements seront bien identiques. Cela permettra d'écarter la possibilité que certains environnements soient, au moment de l'exécution des tests, plus favorables que les autres.

On pourrait utiliser un système d'héritage, mais les classes sont assez courtes et les paramètres à faire évoluer ne sont présents que dans deux méthodes à redéfinir. Il est donc bien plus simple et lisible de dupliquer les classes sans lien d'héritage entre elles.

a) Trop paramétrer l'environnement

Comme mentionné plus tôt, on peut être tenté de contraindre notre agent dans ses actions et lui éviter d'essayer des actions qui nous paraissent contreproductives.

https://arxiv.org/pdf/1907.02908.pdf

Dans notre cas, et afin d'obtenir des résultats plus rapidement, on peut interdire à notre environnement de vendre à un prix inférieur à son coût unitaire.

```
self._action_spec = array_spec.BoundedArraySpec(
 shape=(1,), dtype=np.float32, minimum=1, maximum=100, name='action')
self._action_spec = array_spec.BoundedArraySpec(
 shape=(1,), dtype=np.float32, minimum=1, maximum=100, name='action')
```

Ici, l'action correspond au prix auquel on vend un produit, exprimé en un coefficient multiplicateur du coût unitaire de ce produit.

Avec certains algorithmes, on obtient les premiers résultats positifs dès les premières itérations si le prix minimal est le coût unitaire, au bout de plusieurs milliers d'itérations si le prix minimal est de 0.

Cependant, ce genre d'approche peut empêcher l'agent d'être optimal dans certains cas :

- Péremption d'un produit
- Un produit d'appel peut être vendu à perte afin de permettre de vendre plus au final (essence à la station-service d'un hypermarché, par exemple)
- Spéculation (il faut parfois accepter de vendre avec un déficit pour pouvoir réinvestir sur un produit qui offre de meilleures perspectives)

Pour le cas d'un produit d'appel, nous ne verrons pas ici de mesure du manque à gagner possible, car celui-ci ne peut dépendre que de cas réels très spécifiques.

Le cas de la péremption comprend beaucoup de paramètres, et en établir une simulation réaliste risque d'être trop complexe. On fera donc ici une approximation de la possible différence de résulat entre un agent pouvant vendre à perte et un autre qui ne le peut pas.

b) Variable importante invisible

Il s'agira de mesurer la difficulté supplémentaire d'apprendre dans un environnement lorsqu'une variable importante permettant de mesurer l'efficience relative de ses actions est manquante. En effet, les liens de causalité avec des variables non mesurées sont souvent intégrées dans des modèles statistiques, et disposent même d'une notation spécifique dans les graphes causaux.

Un paramètre inconnu est créé et influence les résultats. Ensuite, on modifie ce paramètre et on observe l'inertie de l'agent en comparant ses résultats à ceux qu'il aurait obtenu sur un environnement qui aurait été dès le départ à l'étape finale.

Exemples de variables invisibles :

- La taille des magasins. Si on imagine que l'expérience était sur les magasins carrefour city et qu'elle inclut par la suite également les carrefour market, d'une taille en moyenne différente. Toutes les quantités varient.
- La flexibilité de la demande selon le prix. On peut imaginer qu'au fil du temps ce paramètre influe plus ou moins les décisions d'achat.

- Dans l'environnement de base, on a pour seule observation la quantité vendue. Le fait que les actions soient le prix de vente de chaque produit exprimé en un coefficient multiplicateur du coût et la récompense la marge sur coûts variables, rendent extrêmement difficile de déterminer l'importance relative de chaque produit dans le résultat final.

On testera donc en ajoutant tous ces paramètres dans les observations fournies par notre environnement à notre agent.

Par la suite, on fera évoluer ces paramètres et on mesurera l'inertie des agents.

Difficulté d'implémentation

Un bug dans la librairie Tensorflow perturbe cette expérience.

Lors de la création de l'environnement, on doit spécifier les dimensions des observations (retournées par l'environnement) ainsi que des actions (à effectuer sur l'environnement) de la manière suivante :

```
self._action_spec = array_spec.BoundedArraySpec(
   shape=(10,), dtype=np.float32, minimum=1, maximum=100, name='action')
self._observation_spec = array_spec.ArraySpec(
   shape = (6, 10),dtype=np.float32,name = 'observation')
```

La spécification signale que les actions doivent être de dimension 10 (un prix pour chaque produit) et que les observations retournées seront de dimension 6*10.

En effet, pour cette expérience, les observations ne seront plus seulement la quantité vendue de chaque produit, mais également 5 autres tableaux de dimension 10 représentant respectivement :

- Le coût des produits
- Leur taux de marge usuel
- Leur taux d'achat habituel
- Leur prix

- La flexibilité de la demande liée au prix

L'environnement est valide selon la méthode prévue à cet effet par la librairie Tensorflow :

```
utils.validate_py_environment(self.BetterObservations_env, episodes=5)
```

Cela signifie qu'il est conforme à ses spécifications, et qu'il fonctionne si on le manipule.

Mais les actions suggérées par l'algorithme SAC sont conformes aux spécifications des observations et non des actions. Ce problème n'apparaît que maintenant étant donné que jusqu'à présent ces spécifications étaient les mêmes.

Plusieurs solutions semblent possibles :

- Changer de librairie
  - Trop long
- Changer d'algorithme
  - Très long, SAC était le seul à donner des résultats satisfaisants rapidement
- Mettre à jour la librairie concernée
  - Passer de Tensorflow 2.2.0 à 2.3.0 (la seule mise à jour possible) apporte trop de changements pour que le reste du code fonctionne, sans garantie que le problème soit résolu.
  - Le problème risque d'être le même en utilisant la version nightly (build quotidien).
- Redimensionner les actions reçues.

Nous avons finalement retenu cette dernière solution, bien plus rapide à mettre en œuvre sans risquer de perturber les autres expérimentations.

Les 6 lignes des actions suggérées par l'agent seront additionnées pour n'en former qu'une seule. Pour ces raisons, l'expérience pourra fournir des résultats, mais il sera très difficile de pouvoir extrapoler ces derniers dans la mesure où les actions de l'agent ne seront pas appliquées telles qu'elles sont censées l'être.

```
if action.shape != self.productsCosts.shape:
    action = np.sum(action, axis=0)
```

c) Inertie face au changement de poids de variables

Est-ce que le modèle se complexifie pour prendre en compte les anciennes données en plus des récentes ?

Quels paramètres pour modifier le taux d'apprentissage et le poids des variables au fil du temps permettent de limiter ce problème ?

Les différents paramètres évolueront dans des sens différents (favorables ou défavorables au résultat), afin d'une part de ne pas trop complexifier les comparaisons, et d'autre part pour qu'exploiter les environnements évolués nécessite des "tactiques" différentes de l'environnement initial.

La nuance avec l'expérience précédente est la suivante : contrairement à l'expérience précédente, qui consistait à comparer des résultats issus d'observations complètes de résultats issus d'observations partielles, il s'agit ici de comparer les résultats avec ou sans changement des paramètres qui ne font pas partie des observations.

Il faudrait comparer les résultats de l'agent sur 3 environnements différents :

- Un environnement constant avec les valeurs initiales
- Un environnement qui évolue depuis les valeurs initiales aux valeurs finales
- Un environnement constant avec les valeurs finales.

Si le deuxième environnement donne de moins bons résultats que les deux autres, c'est que l'inertie de l'agent est élevée.

d) Créer une corrélation sans causalité et observer l'inertie de l'agent

Pour créer une corrélation sans causalité, on peut forcer arbitrairement les actions de l'agent dans un premier temps à aller dans une certaine direction.

L'environnement sera alors paramétré pour retourner de meilleurs résultats au début des tests. Cela créera une corrélation sans causalité entre la direction des variables et le résultat.

En mesurant l'inertie de l'agent pour rétablir ses variables vers quelque chose qui cause effectivement le résultat, on devrait déterminer dans quelle mesure celui-ci confond corrélation et causalité.

e) Inciter au biais du rasoir d'Ockham

Parfois, privilégier le modèle que l'on pense autosuffisant le plus simple peut conduire à sur-simplifier le réel. Ce biais existe en sciences sociales, et il est parfois également observé dans l'apprentissage automatique.

On devrait pouvoir créer une situation incitant un agent à mettre en évidence un biais du rasoir d'Ockham pour reproduire ce dernier et mesurer son impact. Pour cela, on peut imaginer créer la situation suivante :

- Ajouter deux variables corrélées, l'une expliquant beaucoup les observations, l'autre moins, pour que l'agent se concentre sur la première
- Inverser l'importance de ces variables au fil du temps.

Il faudrait alors comparer les résultats de la même manière que pour l'expérience C.

III- Analyse des résultats

A- Trop paramétrer l'environnement

a) Avantage du sur-paramétrage en termes de vitesse d'apprentissage

En annexe, vous trouverez le tableau des résultats obtenus après au fur et à mesure d'un apprentissage sur 10 000 étapes.

Bien que les résultats puissent changer aléatoirement lors de l'exécution de l'algorithme SAC, on observe en général plus rapidement de bien meilleurs résultats sur un environnement où l'agent ne vendra pas à perte.

Cette autre annexe comprend les résultats d'un test mesurant l'efficacité des agents lors de 100 tests à la fin de 1 000 étapes d'apprentissage.

Pour rappel, les environnements ont des paramètres identiques. En effet, la même graine est utilisée pour la génération des nombres aléatoires, et nous disposons d'un test unitaire qui vérifie que ce soit bien le cas. Ainsi, **les données peuvent être interverties en colonne**, faire la moyenne en ligne n'a donc pas de sens.

On peut observer que la moyenne des 10 résultats est plus élevée de 23% lorsque l'environnement ne peut pas vendre à perte (8 469) que lorsqu'il le peut (6 895).

b) Inconvénients du sur-paramétrage en termes de résultat

Si, dans la partie précédente, nous avons vu que dans notre environnement fictif, il est néfaste de pouvoir vendre à perte, cela peut être utile voire nécessaire dans le monde réel.

Si on suppose qu'un produit périmé à un prix supérieur ou égal à son coût ne se vendra pas à un prix supérieur à son coût

Il existe des magasins spécialisés sur les produits périmés. Dans cet exemple, ceux-ci sont vendus avec un taux de réduction de 30% par rapport à un prix en magasin :

Le taux de marge de la distribution alimentaire, très soumise aux questions de péremption, est selon l'INSEE compris entre 13% et 27% (on retiendra 20%).

Les pertes représentent 3,3% du poids des denrées alimentaires transitant par la distribution (on supposera le même ordre de grandeur en valeur).

Empêcher un agent de vendre à perte des denrées alimentaires peut occasionner un manque à gagner de l'ordre de 3,3% des denrées vendues à 70% de 120% de leur coût.

Soit un manque à gagner de 3,3% * 70% * 120% ~= 2,8% de ses coûts, soit (2,8 / 120%) / 2,9 ~= 80% de la marge opérationnelle courante d'une entreprise de grande distribution telle que Carrefour.

En conclusion, dans une situation où les produits vendus peuvent périmer et où ils sont à faible valeur ajoutée, il est théoriquement possible qu'un agent chargé de gérer les prix avec interdiction de vendre à perte ait de moins bons résultats qu'un autre. Cette différence pourrait être d'un ordre de grandeur comparable à celui de la marge opérationnelle de celui qui a le droit de vendre à perte.

B- Variable importante invisible

Les difficultés d'implémentations ne se sont finalement pas arrêtées là.

En effet, l'exécution des expériences s'interrompt purement et simplement sans levée d'exception ni message d'erreur, et stoppe même le module de tests unitaires censé gérer ces comportements.

Ce qui se produit :

```
20:26:01 INFO Starting agent training over 1000 steps
20:26:04 INFO step = 0:
20:26:32 INFO Average return: [0.]
20:27:01 INFO Agent training finished
20:27:01 INFO Test agent result
27:10] C:/workspace/memoire>
```

Comment est censé se terminer un test réussi :

```
2020-08-15 20:34:41 INFO Average return: [0.]
.
----------------------------------------------------------------------
Ran 1 test in 78.471s

OK
```

Comment est censé se terminer un test échoué :

```
E
======================================================================
ERROR: testRaiseExcept (test.test.test)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\workspace\memoire\test\test.py", line 853, in testRaiseExcept
    raise("Exemple d'exception levée")
TypeError: exceptions must derive from BaseException

----------------------------------------------------------------------
Ran 1 test in 0.024s

FAILED (errors=1)
```

Face à ces difficultés et au temps que cela prendrait de tenter de les résoudre, cette expérience ne sera pas poursuivie.

C- Inertie face au changement de poids de variables

a) Résultats

Les résultat sur 10 tests d'apprentissage sur 1 000 étapes sont consultables dans cette annexe.

b) Première analyse

Les résultats de la colonne du milieu sont compris entre ceux des deux autres colonnes. Cela pourrait laisser penser que l'agent obtient des résultats qui dépendent davantage de son environnement que de l'apprentissage qu'il a eu de ce dernier, et des freins qu'il a pu subir.

Cependant, les résultats étant trop faibles, il est difficile de dire que la colonne du milieu est significativement supérieure ou égale à celle de droite.

En effet, une autre explication est possible : les paramètres devaient ne simplement pas être viables dans la deuxième situation.

c) Résolution et nouveaux tests

Les résultats du après modification de l'évolution des paramètres pour qu'ils restent plus viables dans cette annexe.

On obtient un résultat moyen de 6 624 dans le premier cas, 0 dans le deuxième, 4 826 dans le troisième.

d) Deuxième analyse

Les résultats pourraient être interprétés tels quels : faire évoluer les paramètres, même progressivement, donne de moins bons résultats que de garder des paramètres constants.

Cependant, il est très surprenant de n'obtenir **aucun** résultat sur l'environnement qui évolue au cours du temps après 1 000 épisodes.

Aussi pouvons-nous formuler l'hypothèse suivante : si les variables de l'environnement n'étaient pas réinitialisées entre chaque épisode et que les évolutions tendent à rendre l'environnement moins viable (ce qui est probable au regard des résultats obtenus en moyenne 37% plus faibles), il n'est pas surprenant qu'au bout de 1000 épisodes * 30 étapes = 30 000 modifications, il devienne extrêmement difficile d'obtenir un gain.

e) Vérification de cette hypothèse

En effet, la méthode de réinitialisation d'environnement ne régénérait pas les variables modifiées à chaque étape dans le cas de l'environnement que nous testions ici.

Nous obtenons à peu près les mêmes résultats. (Annexe)

Une autre hypothèse pouvant expliquer la difficulté d'obtenir des résultats est la suivante :

Dans le 3e environnement, les paramètres évoluent à l'initialisation, c'est-à-dire au moment où la graine est fixée. Son instance d'entrainement et son instance de test doivent donc avoir des paramètres identiques. L'environnement qui évolue au fil du temps voit ses paramètres évoluer petit à petit. Peut-être que lui se fait tester sur une instance avec des paramètres différents.

f) Vérification de la seconde hypothèse

On fait en sorte que l'environnement qui évolue génère des instances aussi similaires au cours du temps que l'environnement qui évolue à l'initialisation.

Pour cela, on va créer une graine différente pour chaque étape :

```
self.seeds = []
for i in range(self.duration):
    seeds.append(i)
```

Et ces graines seront appliquées respectivement à chaque étape :

```
seed = self.seeds[self._state % self.duration]
random.seed(seed)
np.random.seed(seed)
```

Résultats en annexe.

Ces résultats montrent que bien que l'environnement soit viable, le fait d'apprendre sur un environnement dont les paramètres évoluent au fil du temps est bien plus complexe. En effet, un agent basé sur l'algorithme SAC obtient de meilleurs résultats (en politique d'exploration) avant d'avoir appris qu'après (en politique cupide comme en politique d'exploration). Cela signifie explicitement que l'agent est capable d'obtenir des résultats lorsqu'il apprend, mais pas lorsqu'il cherche à maximiser son résultat à partir de ce qu'il a appris.

E) Inertie face à une corrélation temporaire

L'expérience n'a pu avoir lieu pour les mêmes raisons que l'expérience de l'ajout d'une variable importante invisible qui évolue : l'interruption de l'exécution des tests sans levée d'exception ni affichage de message d'erreur.

F) Biais du rasoir d'Ockham

Toutes les expérimentations qui ont nécessité des observations ou des actions à plus d'une dimension avec l'algorithme SAC s'interrompent sans explication. Pour cette raison, nous n'allons malheureusement pas investir de temps sur cette expérience.

Conclusion

Après avoir listé différentes approches aux questions d'inférence causale et de réduction de biais, nous avons réfléchi à dans quelle mesure ces approches étaient applicables dans le cas de l'apprentissage automatique. Ensuite, nous avons réfléchi à plusieurs cas de test pour observer dans quelle mesure l'apprentissage par renforcement était soumis à des biais. Deux expériences ont été menées à bien.

Tout d'abord, nous avons mesuré l'utilité en termes de rapidité d'apprentissage du fait de trop contraindre un programme d'apprentissage automatique par renforcement en éliminant des cas que l'on peut considérer comme triviaux. Cependant, ces paramètres peuvent également énormément diminuer l'efficacité du programme sur le long terme dans certaines conditions. Dès lors, c'est à l'humain de connaître le contexte exact où sera appliqué le programme qu'il a développé, et de définir le juste milieu entre apprendre plus rapidement et être efficace dans plus de situations.

Dans notre contexte et pour notre récompense à maximiser, cela a permis d'obtenir des résultats 23% plus élevés en moyenne au bout de la même durée d'apprentissage. Cependant, on a également pu démontrer que paramétrer un programme de la sorte pouvait causer un manque à gagner de l'ordre de grandeur proche de la marge opérationnelle.

Le second cas de test consistait à mesurer l'inertie d'un agent face à l'évolution de conditions de son environnement.

On observe faire évoluer les paramètres, même progressivement, donne de moins bons résultats que de garder des paramètres constants, qu'ils soient identiques à la situation d'arrivée ou de départ.

Trois autres cas de tests avaient été conçus, mais n'ont pas pu être réalisés en raison de difficultés techniques et du manque de temps pour les résoudre.

La première de ces expériences aurait consisté à mesurer la difficulté supplémentaire d'apprendre dans un environnement lorsqu'une variable importante permettant de mesurer l'efficience relative de ses actions est manquante. En effet, les liens de causalité avec des variables non mesurées sont souvent intégrées dans des modèles statistiques, et disposent même d'une notation spécifique dans les graphes causaux.

Le second cas se serait intéressé au biais de confirmation que développera ou non un agent mis face à une causalité qui s'atténuera avec le temps. Cette expérience aurait été mesurée comme la seconde à avoir été réalisée : en comparant les résultats obtenus face aux situations statiques initiales et finales.

La troisième et dernière expérience porte sur le biais du rasoir d'Ockham. Parfois, privilégier le modèle que l'on pense autosuffisant le plus simple peut conduire à sur-simplifier le réel. Ce biais existe en sciences sociales, et il est parfois également observé dans l'apprentissage automatique. On devrait pouvoir créer une situation incitant un agent à mettre en évidence un biais du rasoir d'Ockham pour reproduire ce dernier, et mesurer son impact. Pour cela, on peut imaginer créer la situation suivante :

- Ajouter deux variables corrélées, l'une expliquant beaucoup les observations, l'autre moins, pour que l'agent se concentre sur la première
- Inverser l'importance de ces variables au fil du temps
- Enfin, d'observer et comparer les résultats tout comme dans l'expérience précédente

Les résultats des expériences ont montré des cas dans lesquels l'existence d'une relation de causalité vraie ou supposée a pu induire un biais. Ce biais peut être induit par l'humain dans la première expérience, ou être intrinsèque à l'algorithme dans la seconde. Ils ont en tout cas été néfastes pour les résultats d'un apprentissage automatique par renforcement.

Extended abstract

Causal inference from data is a major issue in statistics. Causation is an important part of human cognition. For their part, the machine learning algorithms on which artificial intelligences are based are essentially based on correlations, statistical links between variables. Correlation is not causation, and this distinction is sometimes presented as one of the major obstacles to the development of artificial intelligences.

What response can we bring to this bias?

In the real world, a method often applied is randomization: randomly cutting a sample in half, acting on only one of the halves, and comparing the results between the two groups.

Is this applicable to machine learning?

One type of machine learning stands out from others on the issue. The algorithm would have to be able to act on its environment to learn not only from initial data, but also from the result of its actions. This is precisely the definition of reinforcement learning, one of the types of machine learning, which involves letting an algorithm (an agent) learn by interacting with its environment.

Is reinforcement learning, by acting on its environment, able to differentiate a correlation from a causality?

We can list different biased situations and observe to what extent a reinforcement learning algorithm (an agent) falls into these traps, that is, to what extent these biases make it difficult for this agent to achieve its objective.

This can be viewed as a matter of causal inference. This is a major problem in statistics. Approaches exist that manage to statically recognize the correlations most likely to be causalities from "flat" data, including recent advances.

There are many approaches in many fields of research, but not yet a common statistical tool widely used across disciplines. For example, in the case of epidemiology, there are often large amounts of data available, but sometimes it is difficult to prove the existence of causation. Therefore, we prefer to describe the variables correlated with diseases as risk factors, rather than as causes of diseases.

The question of confusion between correlation and causation in the case of machine learning can be considered trivial from the point of view of algorithms. Indeed, these are not based on causalities, which are real phenomena, only correlations, which are statistical links. According to this consideration, it is when these algorithms are used within an application that human intelligence (developers, analysts) will skew the situation.

Reinforcement learning gives the algorithm the opportunity to interact with its environment to test what is correlation and what is causation.

Can we use reinforcement learning algorithms to find or prove causal relationships between variables?

For the variables of his actions, we can say yes: it suffices to let him act so that he will vary the other observations and the reward or not.

Another part of the bias depends directly on the quality of the input data. If the data itself is not representative of the situation, cross-validation will not solve the problem.

The experiment carried out here consists in applying a reinforcement learning algorithm to biased situations to observe to what extent the bias hinders its learning.

They are implemented via the Tensorflow library. The agent used is SAC, Soft Agent Critic, a model-free agent. The environment, on the other hand, asks the agent to tell them at what price to sell different products. The observations returned to the agent depend on the test cases, but they usually consist of a table containing the number of sales for each product. Finally, the variable to be maximized is the margin on variable costs.

The first test case consists in measuring the acceleration gained when over-parameterizing the environment. Indeed, we may be tempted to coerce our agent into his actions and prevent him from trying actions that we believe to be counterproductive. In our case, in order to obtain results more quickly, we can prevent our environment from selling at a price lower than its unit cost.

In our context and for our reward to be maximized, this resulted in 23% higher results on average after the same learning period. However, it has also been shown that setting up a program in this way could cause a shortfall of the order of magnitude close to the operating margin.

The second test case consisted of measuring the inertia of an agent in the face of changing conditions in its environment.

We observe that the parameters change, even gradually, gives poorer results than keeping the parameters constant, whether they are identical to the arrival or departure situation.

Three other test cases had been designed but could not be performed due to technical difficulties and lack of time to resolve them.

The first of these experiments would have consisted in measuring the additional difficulty of learning in an environment when an important variable allowing to measure the relative efficiency of its actions is missing. Indeed, the causal links with unmeasured variables are often integrated into statistical models, and even have a specific notation in the causal graphs.

The second case would have been interested in the confirmation bias that an agent will develop or not develop when faced with a causality that will attenuate over time. This experiment would have been measured as the second to have been carried out: by comparing the results obtained in the face of the initial and final static situations.

The third and final experiment involves the Ockham razor bias. Sometimes, favoring the model that we think is the simplest self-sufficient can lead to over-simplifying the real. This bias exists in the social sciences, and it is sometimes also observed in machine learning (2). You should be able to create a situation where an agent can identify an Ockham razor bias to replicate it and measure its impact. For this, we can imagine creating the following situation:

- Add two correlated variables, one explaining the observations a lot, the other less so, so that the agent can focus on the first
- Reverse the importance of these variables over time
- Finally, observe and compare the results as in the previous experiment.

The results of the experiments showed cases in which the existence of a true or assumed causal relationship could have induced a bias. This bias can be human induced in the first experiment or be intrinsic to the algorithm in the second. In any case, they were detrimental to the results of reinforcement machine learning.

All the code is accessible on this repository: https://github.com/OdelinT/Memoire

Bibliographie

1

« Antoine - 2016 - Pertes et gaspillages alimentaires l'état des li.pdf ». Page 8. Consulté le 17 août 2020. https://www.ademe.fr/sites/default/files/assets/documents/pertes-gaspillages-alimentaires-etat-lieux-201605-synt.pdf.

Antoine, VERNIER. « Pertes et gaspillages alimentaires : l'état des lieux et leur gestion par étapes de la chaîne alimentaire », 2016, 16.

2

IBM-France. « Apprentissage automatique et biais : Impacts et solutions », 26 septembre 2019. https://www.ibm.com/blogs/ibm-france/2019/09/26/apprentissage-automatique-et-biais/.

3

Bonawitz, Elizabeth Baraff, Isabel Y. Chang, Catherine Clark, et Tania Lombrozo. « Ockham's razor as inductive bias in preschooler's causal explanations ». In *2008 7th IEEE International Conference on Development and Learning*, 7-12, 2008. https://doi.org/10.1109/DEVLRN.2008.4640797.

4

BUSINESS, BFM. « Carrefour a renoué avec les bénéfices en 2019 après deux années dans le rouge ». BFM BUSINESS. BFM BUSINESS. Consulté le 17 août 2020. https://bfmbusiness.bfmtv.com/entreprise/carrefour-a-renoue-avec-les-benefices-en-2019-apres-deux-annees-dans-le-rouge-1865256.html.

5

« Carvalho - Scoring functions for learning Bayesian networks.pdf ». Consulté le 17 août 2020. http://www.lx.it.pt/~asmc/pub/talks/09-TA/ta_pres.pdf.

Carvalho, Alexandra M. « Scoring Functions for Learning Bayesian Networks ». *Bayesian Networks*, s. d., 48.

6

Dasgupta, Ishita, Jane Wang, Silvia Chiappa, Jovana Mitrovic, Pedro Ortega, David Raposo, Edward Hughes, Peter Battaglia, Matthew Botvinick, et Zeb Kurth-Nelson. « Causal Reasoning

from Meta-reinforcement Learning ». *arXiv:1901.08162 [cs, stat]*, 23 janvier 2019. http://arxiv.org/abs/1901.08162.

7

« Des marges commerciales variées selon les produits, mais proches entre grandes surfaces - Insee Focus - 45 ». Consulté le 17 août 2020. https://www.insee.fr/fr/statistiques/1304045.

8

« Dilemme biais-variance ». In *Wikipédia*, 27 août 2019. https://fr.wikipedia.org/w/index.php?title=Dilemme_biais-variance&oldid=162160067.

9

« Fast Greedy Equivalence Search (FGES) Algorithm for Continuous Variables - Center for Causal Discovery ». Consulté le 17 août 2020. https://www.ccd.pitt.edu/wiki/index.php/Fast_Greedy_Equivalence_Search_(FGES)_Algorithm_for_Continuous_Variables.

10

Fjelland, Ragnar. « Why General Artificial Intelligence Will Not Be Realized ». *Humanities and Social Sciences Communications* 7, n° 1 (17 juin 2020): 1-9. https://doi.org/10.1057/s41599-020-0494-4.

11

Friston, Karl J., Jean Daunizeau, et Stefan J. Kiebel. « Reinforcement Learning or Active Inference? » *PLoS ONE* 4, n° 7 (29 juillet 2009). https://doi.org/10.1371/journal.pone.0006421.

12

Fujimoto, Scott, Herke van Hoof, et David Meger. « Addressing Function Approximation Error in Actor-Critic Methods ». *arXiv:1802.09477 [cs, stat]*, 22 octobre 2018. http://arxiv.org/abs/1802.09477.

13

Haarnoja, Tuomas, Aurick Zhou, Pieter Abbeel, et Sergey Levine. « Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor ». *arXiv:1801.01290 [cs, stat]*, 8 août 2018. http://arxiv.org/abs/1801.01290.

14

Hahn, P. Richard, Jared S. Murray, et Carlos M. Carvalho. « Bayesian Regression Tree Models for Causal Inference: Regularization, Confounding, and Heterogeneous Effects ». *Bayesian Analysis*, 2020. https://doi.org/10.1214/19-BA1195.

15

Hessel, Matteo, Hado van Hasselt, Joseph Modayil, et David Silver. « On Inductive Biases in Deep Reinforcement Learning ». *arXiv:1907.02908 [cs, stat]*, 5 juillet 2019. http://arxiv.org/abs/1907.02908.

16

L, +Bastien. « Amazon abandonne son IA de recrutement qui discrimine les femmes ». *LeBigData.fr* (blog), 11 octobre 2018. https://www.lebigdata.fr/amazon-abandonne-ia-misogyne.

17

« L'actualité médicale vue par le professeur Claude Béraud ». Consulté le 17 août 2020. http://www.prclaudeberaud.fr/?129-erreur-ecologique-erreur-atomiste-lepidemiologie-contextuelle.

18

Lewis, Paul, et Erin McCormick. « How an Ex-YouTube Insider Investigated Its Secret Algorithm ». *The Guardian*, 2 février 2018, sect. Technology. https://www.theguardian.com/technology/2018/feb/02/youtube-algorithm-election-clinton-trump-guillaume-chaslot.

19

Lillicrap, Timothy P., Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, et Daan Wierstra. « Continuous control with deep reinforcement learning ». *arXiv:1509.02971 [cs, stat]*, 5 juillet 2019. http://arxiv.org/abs/1509.02971.

20

Liu, Furui, et Laiwan Chan. « Causal Inference on Discrete Data via Estimating Distance Correlations ». *Neural Computation* 28, nº 5 (18 février 2016): 801-14. https://doi.org/10.1162/NECO_a_00820.
https://www.mitpressjournals.org/doi/full/10.1162/NECO_a_00820

21

Madumal, Prashan, Tim Miller, Liz Sonenberg, et Frank Vetere. « Explainable Reinforcement Learning Through a Causal Lens ». *arXiv:1905.10958 [cs, stat]*, 20 novembre 2019. http://arxiv.org/abs/1905.10958.

22

Time. « Microsoft Takes Chatbot Offline After It Starts Tweeting Racist Messages ». Consulté le 17 août 2020. https://time.com/4270684/microsoft-tay-chatbot-racism/.

23

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, et al. « Human-Level Control through Deep Reinforcement Learning ». *Nature* 518, nº 7540 (février 2015): 529-33. https://doi.org/10.1038/nature14236.

24

Mooney, Raymond J. « Comparative Experiments on Disambiguating Word Senses: An Illustration of the Role of Bias in Machine Learning ». *arXiv:cmp-lg/9612001*, 9 décembre 1996. http://arxiv.org/abs/cmp-lg/9612001.

25

« Ockham's razor cuts to the root: Simplicity in causal explanation. » Consulté le 17 août 2020. https://psycnet.apa.org/fulltext/2017-54956-007.html.

26

Pearl, Judea. « Causal Diagrams for Empirical Research ». *Biometrika* 82, nº 4 (1 décembre 1995): 669-88. https://doi.org/10.1093/biomet/82.4.669.

27

Pingault, Jean-Baptiste, Paul F. O'Reilly, Tabea Schoeler, George B. Ploubidis, Frühling Rijsdijk, et Frank Dudbridge. « Using Genetic Data to Strengthen Causal Inference in Observational Research ». *Nature Reviews Genetics* 19, n° 9 (septembre 2018): 566-80. https://doi.org/10.1038/s41576-018-0020-3.

28

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, et Oleg Klimov. « Proximal Policy Optimization Algorithms ». *arXiv:1707.06347 [cs]*, 28 août 2017. http://arxiv.org/abs/1707.06347.

29

« Soft Actor-Critic — Spinning Up documentation ». Consulté le 17 août 2020. https://spinningup.openai.com/en/latest/algorithms/sac.html.

30

Torabi, Faraz, Garrett Warnell, et Peter Stone. « Behavioral Cloning from Observation ». *arXiv:1805.01954 [cs]*, 11 mai 2018. http://arxiv.org/abs/1805.01954.

31

« Une étude démontre les biais de la reconnaissance faciale, plus efficace sur les hommes blancs ». *Le Monde.fr*, 12 février 2018. https://www.lemonde.fr/pixels/article/2018/02/12/une-etude-demontre-les-biais-de-la-reconnaissance-faciale-plus-efficace-sur-les-hommes-blancs_5255663_4408996.html.

32

Cuvelliez, Charles, Quisquate, Jean-Jacques. « Les biais biométriques et ethniques des logiciels de reconnaissance faciale » *Le Monde.fr*, 17 février 2020. https://www.lemonde.fr/blog/binaire/2020/02/17/les-biais-biometriques-et-ethniques-des-logiciels-de-reconnaissance-faciale/.

33

Franceinfo. « VIDEO. Le supermarché anti-gaspi qui vend des produits périmés », 3 septembre 2019. https://www.francetvinfo.fr/sante/alimentation/video-le-supermarche-anti-gaspi-qui-vend-des-produits-perimes_3595515.html.

34

Waldmann, Michael. *The Oxford Handbook of Causal Reasoning*. Oxford University Press, 2017.

W&B. « Weights & Biases ». Consulté le 17 août 2020. https://app.wandb.ai/stacey/aprl/reports/Adversarial-Policies-in-Multi-Agent-Settings%E2%80%93VmlldzoxMDEyNzE.

35

Williams, Ronald J. « Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning ». *Machine Learning* 8, nº 3 (1 mai 1992): 229-56. https://doi.org/10.1007/BF00992696.

36

Williams, Thomas C., Cathrine C. Bach, Niels B. Matthiesen, Tine B. Henriksen, et Luigi Gagliardi. « Directed Acyclic Graphs: A Tool for Causal Studies in Paediatrics ». *Pediatric Research* 84, nº 4 (octobre 2018): 487-93. https://doi.org/10.1038/s41390-018-0071-3.

37

Zhou, Zhengyuan, Michael Bloem, et Nicholas Bambos. « Infinite Time Horizon Maximum Causal Entropy Inverse Reinforcement Learning ». *IEEE Transactions on Automatic Control* 63, nº 9 (septembre 2018): 2787-2802. https://doi.org/10.1109/TAC.2017.2775960.

38

Zhu, Shengyu, Ignavier Ng, et Zhitang Chen. « Causal Discovery with Reinforcement Learning ». *arXiv:1906.04477 [cs, stat]*, 8 juin 2020. http://arxiv.org/abs/1906.04477.

39

« Part 2: Kinds of RL Algorithms — Spinning Up documentation ». Consulté le 17 août 2020. https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html.

40

« Biais algorithmique ». In *Wikipédia*, 14 août 2020. https://fr.wikipedia.org/w/index.php?title=Biais_algorithmique&oldid=173807970.

Glossaire

Définitions issues principalement du Journal officiel n° 0285 du 09/12/2018 :

https://www.legifrance.gouv.fr/jo_pdf.do?id=JORFTEXT000037783813

Apprentissage automatique : "Processus par lequel un algorithme évalue et améliore ses performances sans l'intervention d'un programmeur, en répétant son exécution sur des jeux de données jusqu'à obtenir, de manière régulière, des résultats pertinents".

Le journal officiel le traduit par "machine learning" en anglais. Cela n'est pas certain quand on compare les fréquences de ces expressions au cours du temps avec Ngrams viewer.

Soit l'expression "apprentissage automatique" était porteuse d'un tout autre sens dans les années 1960, soit il s'agit de deux notions différentes avec le même nom.

Apprentissage par renforcement : "Apprentissage automatique dans lequel un programme extérieur évalue positivement ou négativement les résultats successifs de l'algorithme, l'accumulation des résultats permettant à l'algorithme d'améliorer ses performances jusqu'à ce qu'il atteigne un objectif préalablement fixé."

Agent : algorithme d'apprentissage par renforcement.

Autres notions clé

Corrélation : lien mesurable statistiquement entre deux variables.

Causalité : lien de cause à effet entre deux phénomènes.

Effet cigogne : confusion entre corrélation et causalité

Rasoir d'Ockham : aussi appelé principe de parcimonie, est un principe de raisonnement consistant à préférer le modèle auto-suffisant le plus simple pour expliquer un phénomène. On le trouve souvent sous cet énoncé : "les hypothèses suffisantes les plus simples doivent être préférées".

Biais du rasoir d'Ockham : privilégier les modèles les plus simples peut éliminer une explication qui est tout aussi vraie.

Biais (définition générale) : une démarche ou un procédé qui engendre des erreurs.

Biais (en statistiques) : différence entre la valeur de l'espérance d'un estimateur et la valeur qu'il est censé estimer.

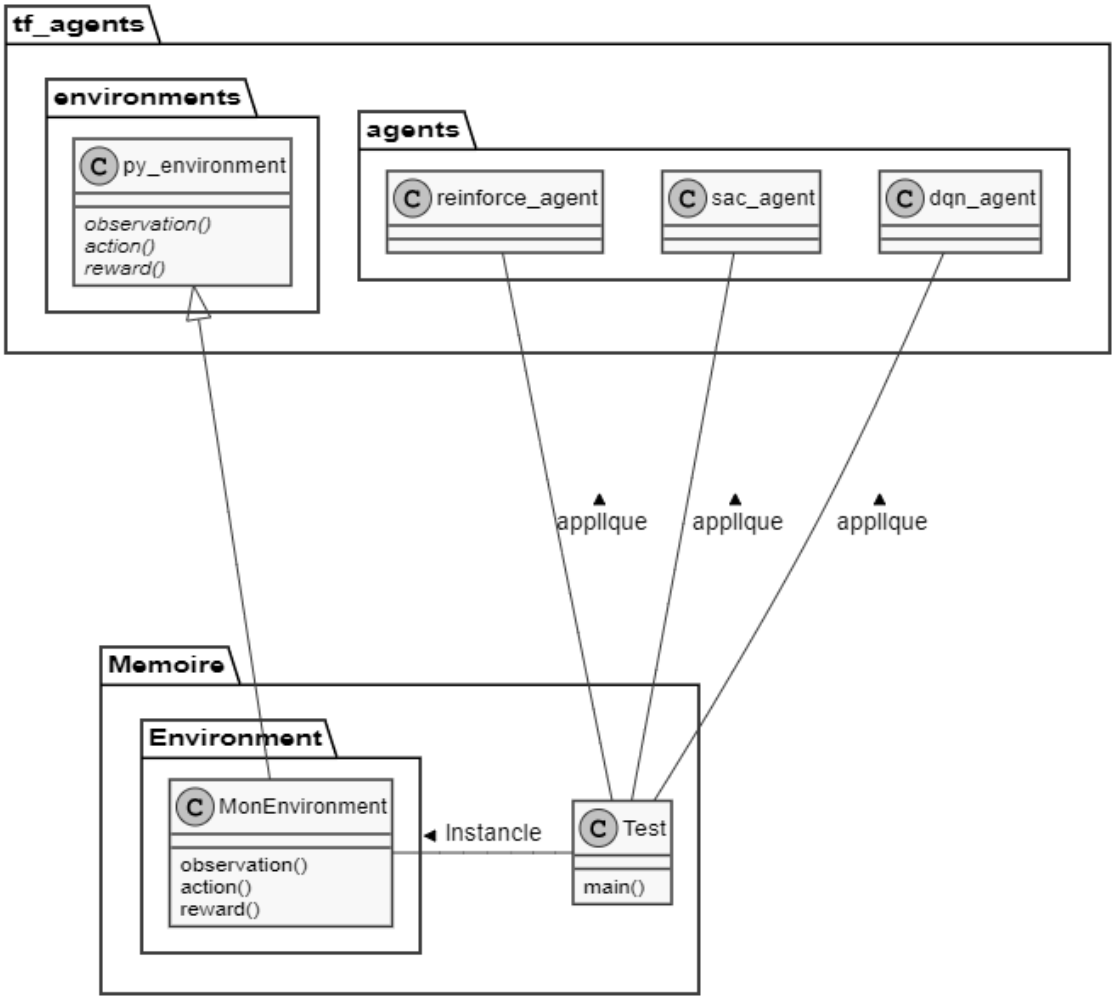
Dilemme biais-variance : le biais diminue à mesure que la complexité du modèle s'approche de la complexité (ni plus simple ni plus complexe) que l'environnement.

Inférence causale : L'inférence causale désigne le processus par lequel on peut établir une relation de causalité entre un élément et ses effets.

https://fr.wikipedia.org/wiki/Inf%C3%A9rence_causale

Annexes

Diagramme de classes simplifié



Expérience A conditions 1

Etape	*Résultats en cas de vente à perte autorisée*	*Résultats en cas de vente à perte interdite*
0	0	0
2000	2 995	4 994
4000	5 621	11 849

6000	4 766	15 283
8000	4 112	18 320
10000	8 165	20 326

Expérience A conditions 2

Résultats en cas de vente à perte autorisée	*Résultats en cas de vente à perte interdite*
7 400	1 380
2 519	8 485
8 901	16 389
2 175	2 676
5 767	3 754
8 797	13 232
8 815	4 335
9 207	14 685
3 811	10 903
11 554	8 848

Expérience C résultats a

Environnement statique	*Environnement qui évolue au fil du temps*	*Environnement qui suit une évolution comparable à l'initialisation puis qui devient statique*
2 775	536	0
2 501	0	0
7 881	0	0
2 679	0	0

4 323	0	0
4 597	0	0
2 988	0	0
3 144	0	0
4 376	0	0
8 633	0	0

Expérience C résultats c

Environnement statique	*Environnement qui évolue au fil du temps*	*Environnement qui suit une évolution comparable à l'initialisation puis qui devient statique*
5 157	0	1 747
11 084	0	2 776
8 247	0	2 426
4 975	0	1 705
8 943	0	9 835
4 987	0	5 759
11 351	0	2 186
3 705	0	4 634
3 325	0	2 715
4 463	0	14 475

Expérience C résultats e

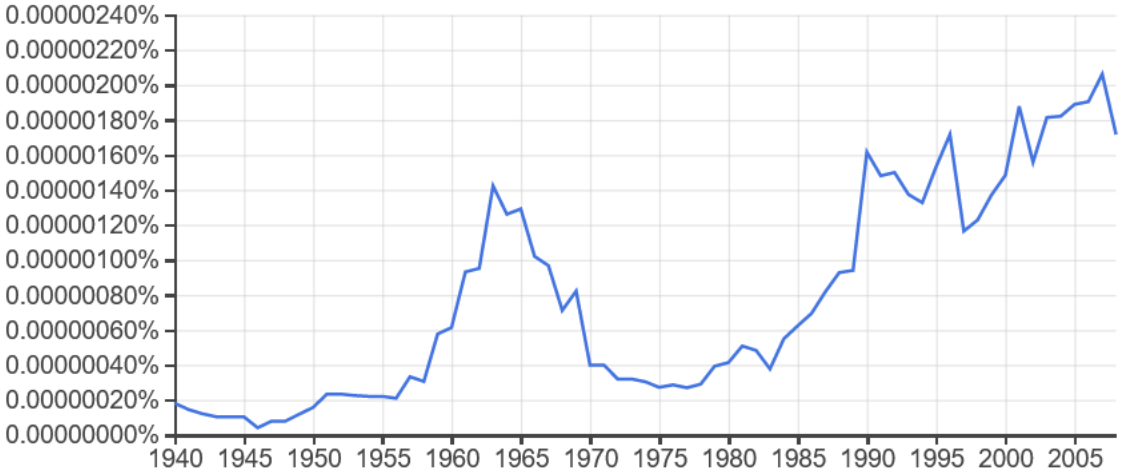
Environnement statique	*Environnement qui évolue au fil du temps*	*Environnement qui suit une évolution comparable à l'initialisation puis qui devient statique*
13 808	0	112
7 296	0	12 079
5 353	0	3 194
2 007	0	2 109
11 138	0	1 821
9 203	0	973
10 095	0	2 602
4 896	0	124
11 008	0	3 918
5 688	0	6 754

Expérience C résultats f

Tests **avant** apprentissage	Tests **après** apprentissage
3 638	0
6 661	0
5 676	0
2 761	0
3 110	0
1 836	0
3 493	0
2 958	0
1 291	0

3 304 0

« Apprentissage automatique » dans le corpus francophone :



« Machine learning » dans le corpus anglophone :