Apprentissage par renforcement de PDM factorisés avec effets corrélés

Boris Lesner, Bruno Zanuttini

Apprentissage par renforcement de PDM factorisés avec effets corrélés

Boris LESNER, Bruno ZANUTTINI

GREYC, CNRS UMR 6072 Université de Caen Basse-Normandie, ENSICAEN
Campus Côte de Nacre, Bd. Maréchal Juin 14032 Caen CEDEX
boris.lesner@info.unicaen.fr, bruno.zanuttini@info.unicaen.fr

Résumé : Nous nous intéressons au problème de l'apprentissage par renforcement dans les Processus de Décision Markoviens Factorisés, dans le cas où les effets des actions sur les variables sont corrélés et partiellement observables, tel que dans les représentations en Opérateurs STRIPS Probabilistes. Nous présentons un algorithme qui apprend à la fois la structure du problème et les effets des actions. Pour ce faire, nous partons d'algorithmes existants traitant les effets indépendants, pour ensuite les généraliser grâce à l'introduction d'une mesure de similarité entre distributions d'effets ambigus.

Mots-clés : apprentissage par renforcement, PDM factorisé, exploitation de structure

1 Introduction

Supposez un instant que vous n'êtes pas riche, et habituellement malchanceux. Pour devenir riche, vous décidez donc de tenter votre chance au Loto. N'étant pas familier avec le calcul des probabilités, vous décidez d'estimer vos chances de *devenir riche* en regardant vos amis jouer. Il s'avère que vos amis *chanceux* deviennent riches 10% du temps. Cependant, tous vos amis qui sont *malchanceux* ont toujours été riches. Dans ce cas, à chaque fois qu'ils ont joué, la seule chose que vous avez observé est qu'ils *sont restés* riches (c'est-à-dire que rien n'a changé). Que pouvez-vous en conclure sur *vos chances* de gagner ?

Puisqu'il est impossible de distinguer dans quels cas vos amis riches ont gagné ou non, une hypothèse cohérente est que *n'importe qui* a 10% de chances de gagner. Mais il est aussi plausible de penser *qu'aucun* de vos amis riches n'a gagné, et d'en conclure que seules les personnes chanceuses ont 10% de chances de gagner, et les autres (les malchanceux), aucune. Puisque vous êtes malchanceux, ces deux explications sont différentes de votre point de vue.

L'important est qu'en tant que preneur de décision, vous avez des actions à disposition, pour lesquelles les effets et conditions sont *inconnus* et *ambiguës* dans certaines des situations à partir desquelles vous tentez de les apprendre. Dans cet article, nous abordons ce problème dans le cadre de l'apprentissage par renforcement dans les Processus de Décision Markoviens Factorisés (propositionnels).

Dans le contexte des Processus de Décision Markoviens (PDM), un agent agit dans un environnement modélisé par des états, des actions et des récompenses. Les actions provoquent des transitions stochastiques entre états, ce qui représente l'incertitude sur les effecteurs de l'agent et sur la dynamique de l'environnement. Les transitions peuvent être représentées en extension par des matrices, mais les représentations factorisées permettent de réduire la taille de représentation des problèmes, et de bénéficier de sa structure sous-jacente lors de la résolution (Boutilier *et al.*, 1999).

Dans un PDM factorisé, l'espace d'états est représenté par l'ensemble des valuations possibles d'un ensemble de variables, et la structure est exploitée pour des représentations plus concises et une résolution plus efficace (Dearden & Boutilier, 1997; Boutilier *et al.*, 1999; Hoey *et al.*, 1999; Boutilier *et al.*, 2000; Guestrin *et al.*, 2003). De telles représentations se classent en deux catégories. Les réseaux bayesiens dynamiques (DBN) (Dean & Kanazawa, 1989), qui sont particulièrement adaptés lorsque les actions ont des effets sur les variables sont indépendants en termes de probabilités, ont été largement étudiés dans la littérature, que ce soit du point de vue de la résolution ou de l'apprentissage. D'un autre côté, les Opérateurs STRIPS Probabilistes (PSO) (Kushmerick, 1995), auxquels nous nous intéressons ici, sont préconisés dans le cas où les effets des actions sur les variables sont fortement corrélés.

Bien que de telles descriptions d'actions puissent être connues et des politiques (approximatives) calculées, un agent peut ne pas avoir de modèle de ses actions *a priori*. Dans ce cas, il doit *apprendre* à se com-

porter de manière optimale. Cette tâche est le plus souvent formalisée comme un problème d'apprentissage par renforcement (AR) (Sutton & Barto, 1998). Ici, nous nous intéressons aux approches « *model-based* » où l'agent construit un modèle de la tâche à résoudre et agit en planifiant à partir de celui-ci.

L'apprentissage par renforcement est très présent dans la littérature lorsqu'il s'agit de PDM représentés par des réseaux bayesiens dynamiques. DBN-E^3 (Kearns & Koller, 1999) et Factored $R_{\max}$ (Guestrin *et al.*, 2002) proposent des approches où la *structure* (le graphe d'influence du réseau) est connu à l'avance. Plus récemment, SPITI (Degris *et al.*, 2006), l'approche VISA (Jonsson & Barto, 2007) et SLF-$R_{\max}$ (Strehl *et al.*, 2007) ont été proposés, et apprennent la structure en même temps que les paramètres du problème.

Néanmoins, toutes ces approches partagent l'hypothèse que les actions ont des effets indépendants sur chaque variable (les DBN n'ont pas d'arcs synchrones), et moins de travaux ont été entrepris en apprentissage par renforcement quand ces hypothèses sont levées. Walsh *et al.* (2009) proposent une approche générique pour l'apprentissage de structure, qu'ils illustrent sur des PSO, mais ils supposent que les effets sont connus *a priori*. À notre connaissance, seuls Pasula *et al.* (2007) abordent le problème d'apprentissage de PSO, dans un cadre relationnel (plus général) et ils ne considèrent que l'apprentissage passif et non par renforcement.

Dans cet article, nous proposons une nouvelle approche d'apprentissage de PSO propositionnels dans un cadre par renforcement. Notre approche s'inspire tout particulièrement de SLF-$R_{\max}$. Nous maintenons l'hypothèse que les conditions d'actions ne font intervenir qu'un petit nombre borné de variables. En ce qui concerne les effets, nous donnons une caractérisation de tous les effets, potentiellement ambigus, qui sont cohérents avec les observations, et en déduisons des heuristiques pour identifier la combinaison correcte d'effets et la distribution de probabilité associée. Dans le cas particulier où les effets ne sont pas ambigus (ou sont connus à l'avance), cette approche ne nécessite qu'un nombre polynomial d'observations. Finalement, nous montrons par des expériences que, malgré le recours à des heuristiques dans le cas général, l'approche fonctionne efficacement en pratique.

2 Préliminaires

Processus de Décision Markoviens

Un Processus de Décision Markovien (PDM) $M = (S, A, T, R, \gamma)$ fait intervenir des ensembles finis d'états S et d'actions A. Pour $s, s' \in S$, la probabilité de transition $T(s'|s, a)$ donne la probabilité de passer à l'état s' en exécutant l'action a dans l'état s. Pour tous s, a, la quantité $R(s, a)$ détermine la récompense obtenue en exécutant l'action a dans l'état s. Le facteur d'actualisation $0 < \gamma < 1$ réduit l'importance des récompenses futures.

Résoudre un PDM revient à calculer une *politique* $\pi : S \mapsto A$ qui maximise l'espérance de récompense à un certain horizon (potentiellement infini). Une politique optimale à l'horizon $h + 1$ peut être dérivée de la fonction de valeur optimale à l'horizon h : $V^*_{h+1} = \max_{a \in A} Q^a_{h+1}(s)$ avec :

$$Q^a_{h+1}(s) = R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a) V^*_h(s')$$

et $V^*_0(s) = 0$. Alors la politique optimale π^*_h est donnée par $\pi^*_h(s) = \operatorname{argmax}_a Q^a_h(s)$. On remarque que $\gamma < 1$ garantit que $V^*_h(s)$ est bornée par $\max_{s,a} R(s,a)/(1-\gamma)$. Tout au long de cet article, nous supposons sans perte de généralité $R(s,a) \in [0, 1]$.

Opérateurs STRIPS Probabilistes

Nous considérons des PDM représentés de manière compacte, où un ensemble $\mathcal{X} = \{x_1, \ldots, x_n\}$ de variables propositionnelles est donné, et où les états sont des affectations complètes de valeurs de vérité à $\mathcal{X}$, c'est-à-dire $S = 2^{\mathcal{X}}$. Nous représentons les affectations comme des ensembles de littéraux, par exemple $x_1 x_2 \overline{x}_3$ affecte VRAI à x_1 et x_2 et FAUX à x_3. Ainsi, $s \cap s'$ dénote l'ensemble de littéraux communs à s et s' ; et $s' \setminus s$, l'ensemble des littéraux de s' dont la variable a une valeur différente dans s. Par exemple, pour $s = x_1 x_2 \overline{x}_3$ et $s' = \overline{x}_1 x_2 x_3$, on a $s \cap s' = x_2$ et $s' \setminus s = \overline{x}_1 x_3$. Enfin, pour un sous-ensemble $\mathbf{X}$ de $\mathcal{X}$, $s[\mathbf{X}]$ dénote la restriction de s aux variables de $\mathbf{X}$, par exemple : $(x_1 x_2 \overline{x}_3)[\{x_1, x_3\}] = x_1 \overline{x}_3$.

Les actions sont décrites par des *Opérateurs STRIPS Probabilistes* (PSO) (Kushmerick, 1995; Dearden & Boutilier, 1997). Une action a est définie par un ensemble de conditions $c_1, c_2, \ldots$, qui sont des formules sur $\mathcal{X}$ (ou, de manière équivalente, des ensembles d'états) telles que chaque état $s \in S$ satisfait au plus

Condition	Effets	Prob.
$\mathit{Office} \wedge \mathit{Rain} \wedge \overline{\mathit{Umb}}$	$\overline{\mathit{Office}}\,\mathit{Wet}$	0,8
	$\overline{\mathit{Office}}$	0,1
	$\emptyset$	0,1
$\mathit{Office} \wedge (\overline{\mathit{Rain}} \vee \mathit{Umb})$	$\overline{\mathit{Office}}$	0,9
	$\emptyset$	0,1
$\overline{\mathit{Office}} \wedge \mathit{Rain} \wedge \overline{\mathit{Umb}}$	$\mathit{Office} \wedge \mathit{Wet}$	0,8
	Office	0,1
	$\emptyset$	0,1
$\overline{\mathit{Office}} \wedge (\overline{\mathit{Rain}} \vee \mathit{Umb})$	Office	0,9
	$\emptyset$	0,1

FIG. 1 – Représentation PSO de l'action MOVE du problème *Coffee*

une condition c_i. À chaque condition c est associé un ensemble *d'effets* $\mathit{Eff}(c)$, muni d'une distribution de probabilités $T(\cdot|c,a)$. Chaque effet $e \in \mathit{Eff}(c)$ est un ensemble cohérent (potentiellement vide) de littéraux de $\mathcal{X}$.

Appliquer un effet e à un état s produit un nouvel état $s' = e(s) = s \setminus \{\ell \mid \overline{\ell} \in e\} \cup e$. Les effets pouvant être appliqués à un état s sont exactement ceux de $\mathit{Eff}(c)$, où c est l'unique condition satisfaite par s. La fonction de transition du PDM non factorisé sous-jacent peut être reconstruite par :

$$T(s'|s,a) = \sum \{T(e|c,a) \mid e \in \mathit{Eff}(c), e(s) = s'\}$$

On remarque que l'effet vide, noté $\emptyset$ ne modifie pas l'état auquel il est appliqué.

Exemple 1
Soient $s = x_1\overline{x}_2x_3$ et $e_1 = \overline{x}_2\overline{x}_3$, $e_2 = \overline{x}_3$. Alors $e_1(s) = x_1\overline{x}_2\overline{x}_3$. On remarque que $e_2(s) = x_1\overline{x}_2\overline{x}_3$: différents effets peuvent produire les mêmes transitions à partir d'un même état. Comme nous le verrons par la suite, ceci a des conséquences majeures sur la tâche d'apprentissage.

À titre d'exemple également, la figure 1 montre une représentation PSO de l'action MOVE *du problème* Coffee *(Dearden & Boutilier, 1997) : se déplacer peut mener le robot à l'extérieur ou à l'intérieur du bureau (Office), mais l'action peut échouer ; de plus, s'il pleut (Rain) et que le robot n'a pas de parapluie ($\overline{\mathit{Umb}}$) , il peut devenir mouillé (Wet).*

Autres représentations

La représentation d'actions par PSO est en quelque sorte orthogonale à celle par des réseaux bayesiens dynamiques (DBN). Les DBN sont aussi expressifs que les PSO à la condition que l'on autorise les *arcs synchrones*, c'est-à-dire des dépendances entre les variables post-action. En ce qui concerne la planification, la plupart des approches gèrent ces arcs synchrones, par exemple en utilisant les *ancêtres* (Guestrin *et al.*, 2003). En revanche, toutes les approches d'apprentissage par renforcement qui s'intéressent aux DBN présupposent l'absence de tels arcs (Degris *et al.*, 2006; Jonsson & Barto, 2007; Strehl *et al.*, 2007), et il n'est pas certain que ces approches puissent être étendues pour les gérer.

Le cas typique où des effets corrélés (c'est-à-dire des arcs synchrones) apparaissent est lorsqu'on dispose d'actions qui peuvent changer simultanément la valeur de vérité de certaines variables, mais peuvent échouer pour certaines. Par exemple, jouer au Loto peut rendre riche et heureux, simplement riche ou ne rien changer. Dans ce cas, n'importe quelle distribution indépendante sur les variables post-action donnerait une probabilité non nulle de devenir heureux et non riche (c'est-à-dire d'être heureux d'avoir perdu) ; dans ce cas, les arcs synchrones sont obligatoires. Un autre cas typique est lorsqu'un effet a des ramifications : l'action a usuellement $\emptyset$ et $x_1x_2 \ldots x_k$ comme effets, où $x_2 \ldots x_k$ sont les ramifications de x_1.

3 Cadre d'apprentissage

La tâche d'apprentissage par renforcement que nous abordons fait intervenir un ensemble de variables $\mathcal{X}$, d'actions A et une fonction de récompense R, supposés connus à l'avance par l'apprenant. Ce qui est caché est le modèle de transition : pour chaque action a, il s'agit d'apprendre les conditions $c_1, c_2, \ldots$, leurs ensembles d'effets *Eff*(c_1), *Eff*$(c_2), \ldots$ et leur probabilités respectives $p_i = T(e_i|c_j, a)$, pour chaque effet $e_i \in$ *Eff*(c_j).

De façon classique, nous considérons des *traces* dans lesquels à chaque instant t, l'apprenant observe son état courant s_t, exécute une action a et arrive dans un état s_{t+1} tiré selon la fonction de transition (cachée), et reçoit la récompense $R(s_t, a)$.

Pour évaluer notre approche, nous nous reposons sur le récent formalisme « Knows What it Knows » (KWIK) (Walsh *et al.*, 2009). L'apprenant est initialisé avec les paramètres $\epsilon, \delta \in]0,1]$ (en plus de $\mathcal{X}$, A, R), et à chaque instant t et pour chaque action a il peut soit : (i) avoir une estimation ϵ-correcte des probabilités de transition depuis s_t *via* l'action a, ou (ii) admettre qu'il n'a pas de telle estimation. C'est un *apprenant KWIK* s'il existe un polynôme p tel que pour n'importe quelle trace, quelle que soit sa durée, l'apprenant admet son ignorance (cas (ii)) à pas plus de $p(1/\epsilon, 1/\delta, |\mathcal{X}|, |A|)$ instants, et qu'avec confiance $1 - \delta$ les estimations du cas (i) sont en effet ϵ-correctes. Il est alors connu (Walsh *et al.*, 2009) qu'un apprenant KWIK peut être transformé en un apprenant par renforcement efficace, au sens classique, en gérant le dilemme exploration/exploitation comme pour $R_{\max}$ (Brafman & Tennenholtz, 2003). Intuitivement, ceci est dû au fait qu'un agent explorant selon une stratégie $R_{\max}$ a principalement besoin de savoir s'il connaît ou non le comportement de a dans un état s.

Ici, l'ϵ-correction est mesurée comme suit. Pour une action a et un état s, soit E l'ensemble (caché) des effets possibles dans s *via* a, avec leur probabilité respective, et soit $\hat{E}$ un autre ensemble d'effets muni d'une distribution de probabilités. Alors nous mesurons la précision de $\hat{E}$ vis-à-vis de E via une généralisation de la distance L_1. Intuitivement, nous considérons que les effets mentionnés dans E mais pas dans $\hat{E}$ ont une probabilité 0, et vice-versa.

Formellement, pour $E = \{(e_1,p_1), \ldots, (e_n, p_n)\}$ et $\hat{E} = \{(\hat{e}_1,\hat{p}_1), \ldots, (\hat{e}_{\hat{n}}, \hat{p}_{\hat{n}})\}$ avec tous les e_i (resp. $\hat{e}_i$) disjoints deux à deux, $||E - \hat{E}||_1$ est défini par :

$$||E - \hat{E}||_1 = \sum_{e_i = \hat{e}_j} |p_i - \hat{p}_j| + \sum_{(e_i,\cdot) \notin \hat{E}} p_i + \sum_{(\hat{e}_j,\cdot) \notin E} \hat{p}_j$$

Par exemple, pour $E = \{(e_1, 0{,}1),(e_2, 0{,}4),(e_3, 0{,}5)\}$ et $\hat{E} = \{(e_2, 0{,}6),(e_3, 0{,}2),(e_4, 0{,}2)\}$, on a $||E - \hat{E}||_1 = 0{,}8$.

Il est clair que ceci définit en effet une distance qui correspond exactement à la distance L_1 dans le cas où $n = \hat{n}$ et $e_i = \hat{e}_i$ pour tout i. De par le résultat suivant, qui est une adaptation d'un résultat classique pour les PDM non factorisés (Strehl *et al.*, 2009), une estimation précise du modèle de transitions en ce sens assure une estimation précise des Q-valeurs d'action. Dans ce qui suit, nous notons $E(a,s)$ (resp. $\hat{E}(a,s)$) l'ensemble d'effets disponibles dans l'état s pour l'action a selon M (resp. $\hat{M}$).

Proposition 2
Soient $V^*, \hat{V}^*$ les fonctions de valeur optimales pour deux PDM factorisés M et $\hat{M}$ avec les mêmes actions et récompenses. Si, pour tout s,a, $||E(a,s) - \hat{E}(a,s)||_1 \leq \alpha(1-\gamma)^2$, alors $||V^* - \hat{V}^*||_\infty \leq \alpha$ est vérifiée.

Puisque nous nous basons sur l'algorithme SLF-$R_{\max}$ (Strehl *et al.*, 2007), nous faisons la même hypothèse que pour chaque action a, il existe un ensemble $\mathbf{X} \subseteq \mathcal{X}$ contenant au plus k variables et tel que les conditions de a ne mentionnent que des variables de $\mathbf{X}$. Ci-après, cet ensemble est appelé *l'ensemble condition* de a. Nous supposons k *fixé* et *connu* de l'apprenant. Dans, ce cas, la tâche d'apprentissage revient à déterminer, pour chaque action a, (i) l'ensemble $\mathbf{X}$, ci-après nommé *l'ensemble condition* de a, (ii) les effets *Eff*$(\mathbf{x})$ pour chacune des 2^k affectations[1] $\mathbf{x}$ à $\mathbf{X}$ et (iii) leurs probabilités associées.

La définition suivante sera utile par la suite. Elle formalise ce qui sera observé si l'on fusionne les observations pour tous les états s (uniformément distribués) qui satisfont une condition $\mathbf{x}$.

[1]Dans le cas où les variables d'état sont multivaluées, on considèrerait les d^k affectations, où d est la taille du domaine des variables.

Définition 3
Soient a une action, $\mathbf{X} \subseteq \mathcal{X}$ n'importe quel ensemble de variables et $\mathbf{x}$ une affectation de $\mathbf{X}$. La distribution d'effets induite par a sur $\mathbf{x}$ *est l'ensemble de tous les effets e de a qui peuvent survenir dans n'importe quel état satisfaisant $\mathbf{x}$, avec la probabilité induite p_e de e étant :*

$$p_e = \frac{1}{2^{|\mathbf{X}|}} \sum \{T(e|s,a) \mid s \in S, s[\mathbf{X}] = \mathbf{x}\}$$

De manière évidente, si $\mathbf{X}$ est l'ensemble condition pour a, alors la distribution induite pour toute affectation $\mathbf{x}$ de $\mathbf{X}$ est $T(\cdot|s,a)$.

4 Apprentissage des conditions

Par souci de simplicité, nous présentons dans un premier temps notre approche en supposant que *les effets qui se sont produits peuvent être uniquement identifiés* à partir de l'observation d'une transition entre deux états. Dans ce cas, le problème se limite à identifier l'ensemble condition de k variables pour chaque action, puis d'estimer les probabilités de chaque effet en utilisant leur fréquence observée.

Nous utilisons la même approche que l'algorithme SLF-$R_{\max}$ (Strehl *et al.*, 2007), c'est-à-dire que nous maintenons des statistiques pour chaque sous-ensemble possible de k variables (et $2k$ variables, ce qui sera expliqué par la suite) et les utilisons pour découvrir l'ensemble condition cible, ou une estimation de celui-ci garantie d'être suffisamment précise.

Plus précisément, pour chaque action a, chaque ensemble $\mathbf{X}$ de k et $2k$ variables et leurs affectations $\mathbf{x}$, *l'algorithme d'Apprentissage de Conditions* pour l'action a, AC_a, stocke tous les effets e pour lesquels il a observé une transition de la forme $(s,a,e(s))$ quand $s[\mathbf{X}] = \mathbf{x}$, ainsi que leurs fréquences observées. On note :

- $C_a^{\mathbf{X}}(\mathbf{x},e)$ pour le nombre d'observations $(s,a,e(s))$ avec $s[\mathbf{X}] = \mathbf{x}$; on appelle les effets *vus par* $\mathbf{x}$, les effets e tels que $C_a^{\mathbf{X}}(\mathbf{x},e) > 0$,
- $C_a^{\mathbf{X}}(\mathbf{x})$ pour le nombre d'observations $(s,a,\cdot)$ avec $s[\mathbf{X}] = \mathbf{x}$, c'est-à-dire $C_a^{\mathbf{X}}(\mathbf{x}) = \sum_e C_a^{\mathbf{X}}(e,\mathbf{x})$.

Par conséquent, à tout moment durant l'apprentissage, la quantité $\hat{T}_a^{\mathbf{X}}(e|\mathbf{x},a) = C_a^{\mathbf{X}}(\mathbf{x},e)/C_a^{\mathbf{X}}(\mathbf{x})$ est une estimation de la probabilité de e selon a induite par $\mathbf{x}$. Quand le contexte le permet, nous écrivons simplement $C_a^{\mathbf{X}}(s,e)$ pour $C_a^{\mathbf{X}}(s[\mathbf{X}],e)$ et, de manière similaire, $C_a^{\mathbf{X}}(s)$ et $\hat{T}_a^{\mathbf{X}}(e|s,a)$. Pour simplifier, nous disons que les transitions sont observées/comptées par $\mathbf{x} = s[\mathbf{X}]$. On remarque notamment que tout effet vu par $\mathbf{x}'$ est aussi vu par $\mathbf{x} \subseteq \mathbf{x}'$.

Pour identifier l'ensemble condition de a, Strehl *et al.* (2007) ont montré que pour un ensemble $\mathbf{X}$ de k variables et un état s, si l'estimation empirique d'une distribution d'effets sur $s[\mathbf{X}]$ est suffisamment proche de celle sur tout $\mathbf{X}' \supset \mathbf{X}$, $|\mathbf{X}'| = 2k$, alors $\hat{T}^{\mathbf{X}}(\cdot|s,a)$ est proche de la distribution cible $T(\cdot|s,a)$. Bien que le résultat porte sur des DBN sans arcs synchrones, leur preuve peut s'étendre directement à notre approche[2].

Pour résumer, pour un état s et un ensemble de k variables $\mathbf{X}$ nous considérons la condition suivante :

$$\forall \mathbf{X}' \supset \mathbf{X}, |\mathbf{X}'| = 2k, \quad \left|\left|\hat{T}_a^{\mathbf{X}}(\cdot|s[\mathbf{X}],a) - \hat{T}_a^{\mathbf{X}'}(\cdot|s[\mathbf{X}'],a)\right|\right|_1 \leq \epsilon/2 \tag{1}$$

Alors, pour $\epsilon,\delta \in]0,1]$, dès lors que les compteurs $C_a^{\mathbf{X}}(s)$ atteignent m (les observations suivantes sont simplement ignorées), où m, ϵ et δ seront précisés dans le théorème 5 :

- l'ensemble condition correct $\mathbf{X}$ satisfait la condition (1) avec probabilité au moins $1-\delta$,
- si $\mathbf{Y}$ (de taille k) satisfait la condition (1), alors $||\hat{T}^{\mathbf{Y}}(\cdot|s,a) - T(\cdot|s,a)||_1 \leq \epsilon$.

Par conséquent, n'importe quel ensemble de k variables qui satisfait la condition (1) peut être utilisé comme une estimation ϵ-correcte de $T(\cdot|s,a)$. Intuitivement, le premier point s'appuie sur le fait que des observations aléatoires pour les états satisfaisant $s[\mathbf{X}']$ sont tirés selon une distribution qui, par hypothèse, ne dépend que de $\mathbf{X}$. Le deuxième point est valide car $\mathbf{Y}$ doit être proche en particulier de $\mathbf{Y} \cup \mathbf{X}$, qui est lui-même proche de $\mathbf{X}$ d'après le premier point.

On est en droit de se demander s'il est réellement nécessaire de considérer k variables additionnelles plutôt qu'une seule. L'exemple suivant illustre cette nécessité.

[2] Strehl *et al.* (2007) utilisent la condition que $\hat{T}^{\mathbf{X}'}(\cdot|s,a)$ soit ϵ-proche de $\hat{T}^{\mathbf{X}''}(\cdot|s,a)$ pour tout $\mathbf{X}', \mathbf{X}'' \supset \mathbf{X}$. Cependant, sous la condition que tous les $\mathbf{X}', \mathbf{X}''$ aient reçu le même nombre d'exemples, $\hat{T}^{\mathbf{X}}(\cdot|s,a)$ est la moyenne pondérée, effet par effet, de tous les $\hat{T}^{\mathbf{X}'}(\cdot|s,a)$, et donc les deux conditions sont équivalentes.

Exemple 4
Soient les variables $x_1, \ldots, x_4$, et une action a avec pour effets e_1,e_2. Sous la condition $x_1 = x_2$, e_1 et e_2 ont pour probabilité $0{,}5$, tandis que sous la condition $x_1 \neq x_2$, e_1 a pour probabilité $0{,}1$ et e_2 a $0{,}9$. Alors, même en estimant exactement les probabilités de transition, on obtient pour $\mathbf{X} = \{x_3,x_4\}$: $\hat{T}_a^{\mathbf{X}}(e_1 \mid x_3x_4,a) = \hat{T}_a^{\mathbf{X}\cup\{x_1\}}(e_1 \mid x_1x_3x_4,a) = \hat{T}_a^{\mathbf{X}\cup\{x_2\}}(e_1 \mid x_2x_3x_4,a) = 0{,}3$, avec la probabilité complémentaire pour e_2 (ceci vient du fait que $0{,}3 = (0{,}1 + 0{,}5)/2$). Dans ce cas, $\mathbf{X}$ satisfait effectivement la condition (1) (pour $s = x_1x_2x_3x_4$) mais n'est pas ϵ-correcte.

Strehl *et al.* (2007) proposent de considérer une action a comme *connue* dans un état s dès qu'il y a un ensemble $\mathbf{X}$ tel que $C_a^{\mathbf{X}'}(s) = m$ pour tous les ensembles $\mathbf{X}' \supset \mathbf{X}$, $|\mathbf{X}'| = 2k$, où m dépend de ϵ et δ et qui satisfait la condition (1). Alors *l'algorithme d'apprentissage de conditions* pour a renvoie $\hat{T}^{\mathbf{X}}(\cdot|s,a)$, sinon il renvoie $\emptyset$ (c'est-à-dire qu'il admet son ignorance). En s'inspirant de leur preuve et en l'adaptant à la norme L_1, nous obtenons le résultat suivant (rappelons que les effets sont supposés observables).

Théorème 5
Si $m \geq N\frac{2}{\epsilon^2}\ln\frac{N(2^E-2)}{\delta}$, alors, avec confiance au moins $1-\delta$, l'algorithme d'apprentissage de conditions CL_a renvoie $\emptyset$ au plus m fois, et sinon produit des estimations ϵ-correctes, avec $E = \max_c |\mathit{Eff}(c)|$ et $N = \binom{n}{2k}2^{2k}$.

Démonstration (idée). Nous devons d'abord nous assurer que pour toutes les affectations $\mathbf{x}'$ aux ensembles $\mathbf{X}'$ de taille $2k$, $\hat{T}_a^{\mathbf{X}'}(\cdot|\mathbf{x}',a)$ est ϵ-proche de la distribution induite par $\mathbf{x}'$ avec confiance $1-\delta_{\mathbf{x}'}$. Weissman *et al.* (2003) montrent que $m_{\mathbf{x}'} = 2/\epsilon^2 \ln\frac{2^E-2}{\delta_{\mathbf{x}'}}$ exemples sont suffisants. Comme nous voulons cette précision pour les N tels $\mathbf{X}'$, $\mathbf{x}'$, il faut $\delta_{\mathbf{x}'} = \delta/N$ et $m = Nm_{\mathbf{x}'}$. □

Par conséquent, pour une valeur de k fixée, le nombre d'erreurs/d'admission d'ignorance est polynomial, de manière similaire à SLF-$R_{\max}$. L'algorithme d'apprentissage de conditions requiert en outre un espace en $O(\binom{n}{2k}2^{2k}E)$. Néanmoins, il est courant d'utiliser des valeurs bien plus petites pour m (voir section 7), le théorème 5 ayant seulement pour but de présenter des bornes théoriques.

5 Apprentissage des effets

Nous arrivons maintenant à la partie centrale de notre contribution, à savoir l'apprentissage des effets non directement observables. Pour ce faire, nous proposons une extension de la distance L_1 utilisée par la condition (1).

Rappelons qu'une transition observée (s,a,s') peut être expliquée par différents effets, que nous appelons *ambigus* (pour s). La caractérisation suivante de ces effets découle directement de la définition de $e(s)$. Intuitivement, $s' \setminus s$ est l'ensemble de tous les littéraux qui *doivent* changer dans s, tandis que les littéraux communs à s et s' *peuvent*, avoir été mis à leur valeur dans s' par l'effet, ou simplement avoir persisté.

Lemme 6
Soient s,s' deux états. Les effets e tels que $e(s) = s'$ sont exactement ensembles de littéraux sur $\mathcal{X}$ tels que $s' \setminus s \subseteq e \subseteq s'$.

On note $[s' \setminus s, s'] = \{e \mid s' \setminus s \subseteq e \subseteq s'\}$ pour l'intervalle de tous les effets qui peuvent avoir causé la transition observée de s à s'. Ce sont des intervalles d'ensembles, en particulier $[a,b] \neq \emptyset \Leftrightarrow a \subseteq b$, et $[a,b] \cap [c,d] = [a \cup c, b \cap d]$ si $a \cup c$ est cohérent, et $\emptyset$ sinon.

À partir de ceci, nous proposons d'étendre les algorithmes d'apprentissage de conditions de la section précédente en maintenant des statistiques, non pas sur les effets, mais sur des intervalles d'effets. On note alors :

- $I_a^{\mathbf{X}}(\mathbf{x})$ pour l'ensemble de tous les intervalles $[s' \setminus s, s']$ tels que (s,a,s') a été observé quand $s[\mathbf{X}] = \mathbf{x}$,
- $C_a^{\mathbf{X}}(\mathbf{x}, I)$, pour tout $I \in I_a^{\mathbf{X}}(\mathbf{x})$, pour le nombre d'observations (s,a,s') avec $s[\mathbf{X}] = \mathbf{x}$ et $I = [s' \setminus s, s']$,
- $C_a^{\mathbf{X}}(\mathbf{x})$, pour le nombre total d'observations (s,a,s') avec $s[\mathbf{X}] = \mathbf{x}$: $C_a^{\mathbf{X}}(\mathbf{x}) = \sum_{I \in I_a^{\mathbf{X}}(\mathbf{x})} C_a^{\mathbf{X}}(\mathbf{x}, I)$.

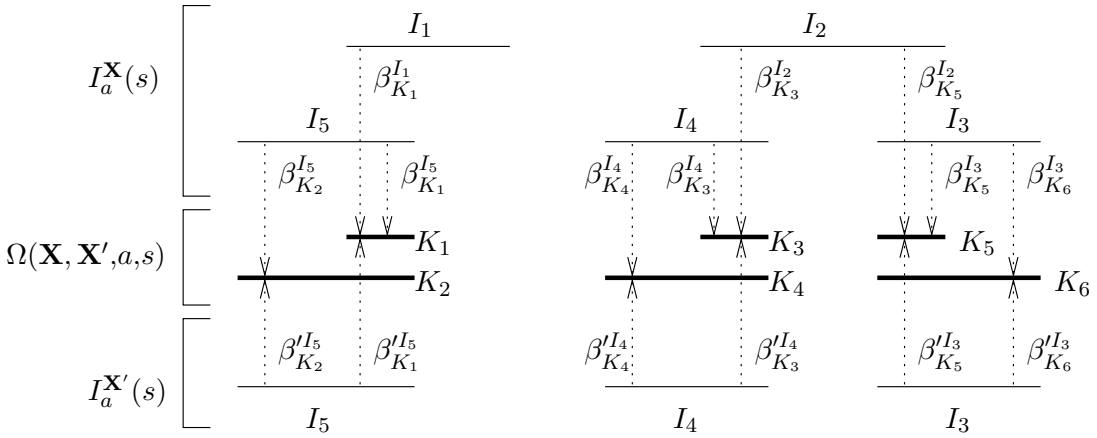
Exemple 7
Soient deux variables x_1, x_2, une action a avec pour conditions $x_1, \overline{x}_1$ et $k = 1$. Supposons que les effets pour x_1 sont $\overline{x}_1$, $\overline{x}_2$ et $\emptyset$.

Quand l'algorithme observe une transition de $x_1\overline{x}_2$ vers $\overline{x}_1\overline{x}_2$, il ajoute l'intervalle $I_1 = [\overline{x}_1,\overline{x}_1\overline{x}_2]$ à $I_a^{\{x_1\}}(x_1)$ et à $I_a^{\{x_2\}}(\overline{x}_2)$, et met leurs compteurs $C_a^{\{x_1\}}(x_1,I_1)$ et $C_a^{\{x_2\}}(\overline{x}_2,I_1)$ à 1. Maintenant, supposons qu'il observe une transition de $x_1\overline{x}_2$ vers lui-même. L'intervalle correspondant est $I_2 = [\emptyset,x_1\overline{x}_2]$, qui est ajouté à $I_a^{\{x_1\}}(x_1)$ et $I_a^{\{x_2\}}(\overline{x}_2)$ avec un compteur de 1 pour chacun. Si la même transition est encore observée, $I_a^{\{x_1\}}(x_1)$ et $I_a^{\{x_2\}}(\overline{x}_2)$ ne changent pas, mais les compteurs pour I_2 sont incrémentés.

Supposons maintenant qu'une transition de x_1x_2 à $x_1\overline{x}_2$ est observée, avec pour intervalle associé $I_3 = [\overline{x}_2,x_1\overline{x}_2]$. C'est un nouvel intervalle candidat pour la condition candidate x_1, donc $I_a^{\{x_1\}}(x_1)$ devient $\{I_1,I_2,I_3\}$. De façon similaire, $I_a^{\{x_2\}}(x_2)$ est initialisé à $\{I_3\}$, et leurs compteurs $C_a^{\{x_1\}}(x_1,I_3)$, $C_a^{\{x_2\}}(x_2,I_3)$ sont mis à 1.

Une fois que tous les effets ont été observés pour les états $x_1\overline{x}_2$ et x_1x_2 (s'ils le sont un jour), on a $I_a^{\{x_1\}}(x_1) = \{I_1,I_2,I_3,I_4,I_5\}$, $I_a^{\{x_2\}}(x_2) = \{I_1,I_2\}$, et $I_a^{\{x_2\}}(\overline{x}_2) = \{I_3,I_4,I_5\}$ avec $I_4 = [\emptyset,x_1x_2]$ et $I_5 = [\overline{x}_1,\overline{x}_1x_2]$.

Intuitivement, pour chaque ensemble $\mathbf{X}$ de taille k et une affectation $\mathbf{x}$ de ses variables, nous maintenons l'ensemble de tous les effets qui sont candidats pour expliquer les observations (de façon similaire à la notion d'*espace des versions* en classification supervisée). Comme dans la section précédente, nous faisons de même pour les ensembles $\mathbf{X}'$ de $2k$ variables et leurs affectations (ce n'est pas montré dans l'exemple 7). Maintenant, ils nous faut généraliser la distance L_1 à des ensembles d'intervalles car, dès qu'un intervalle observé est en intersection avec plus d'un autre, cette distance ne peut plus être utilisée.



$$\delta_{\vec{\beta}}(\mathbf{X},\mathbf{X}',a,s) =$$
$$\left|\frac{\beta_{K_1}^{I_1}+\beta_{K_1}^{I_5}}{20} - \frac{\beta_{K_1}^{I^5}}{10}\right| + \left|\frac{\beta_{K_2}^{I_5}}{20} - \frac{\beta_{K_2}^{I^5}}{10}\right| + \left|\frac{\beta_{K_3}^{I_2}+\beta_{K_3}^{I_4}}{20} - \frac{\beta_{K_3}^{I^4}}{10}\right| + \left|\frac{\beta_{K_4}^{I_4}}{20} - \frac{\beta_{K_4}^{I^4}}{10}\right| + \left|\frac{\beta_{K_5}^{I_2}+\beta_{K_5}^{I_3}}{20} - \frac{\beta_{K_5}^{I^3}}{10}\right| + \left|\frac{\beta_{K_6}^{I_5}}{20} - \frac{\beta_{K_6}^{I^5}}{10}\right|$$

FIG. 2 – Représentation graphique des intervalles communs et de la distance L_1 étendue pour les observations de l'exemple 8

Exemple 8 (suite)
Supposons que les probabilités des effets sont 0,1, 0,7, *et* 0,2, *respectivement, qu'à un certain moment les compteurs de $I_1,\ldots,I_5$ pour $\mathbf{x} = x_1$ sont $2,8,1,7,2$, respectivement, et que ceux de I_3,I_4,I_5 sont $1,7,2$ pour $\mathbf{x}' = x_1x_2$.*

Puisque seulement $I_1 \cap I_5$,$I_2 \cap I_3$ et $I_2 \cap I_4$ sont non vides, il est raisonnable d'en déduire que a a trois effets : $e_{15} \in I_1 \cap I_5$, $e_{23} \in I_2 \cap I_3$, $e_{24} \in I_2 \cap I_4$ (par exemple, e_{24} peut être $\emptyset$, ou x_1). Mais selon que l'on suppose que toutes les transitions de $x_1\overline{x}_2$ vers lui-même (I_2) observées par x_1 soient (i) toutes des manifestations de e_{23}, ou (ii) des manifestations e_{23} (1 d'entre elles) et de e_{24} (7 d'entre elles), on obtient une distance L_1 de :

(i) $\left|\left(\frac{2}{20}+\frac{2}{20}\right) - \frac{2}{10}\right| + \left|\left(\frac{8}{20}+\frac{1}{20}\right) - \frac{1}{10}\right| + \left|\frac{7}{20} - \frac{7}{10}\right| = 0{,}7$

(ii) $\left|\left(\frac{2}{20}+\frac{2}{20}\right) - \frac{2}{10}\right| + \left|\left(\frac{1}{20}+\frac{1}{20}\right) - \frac{1}{10}\right| + \left|\left(\frac{7}{20}+\frac{7}{20}\right) - \frac{7}{10}\right| = 0$

De manière générale, il n'y a pas de raison *a priori* de supposer que certaines transitions observées (s,a,s') sont dues à un effet précis de $[s'\backslash s, s']$, ou à plusieurs d'entre eux. C'est pourquoi nous caractérisons toutes ces hypothèses valides et les distances correspondantes. Ceci revient à généraliser à des ensembles *d'intervalles* la distance L_1 utilisée dans la condition (1).

Pour ce faire, nous considérons un ensemble particulier d'effets qui a pu produire les intervalles observés à la fois par $\mathbf{X}$ et $\mathbf{X}'$.

Nous définissons tout d'abord l'ensemble des intervalles *communs* à $I_a^{\mathbf{X}}(s)$ et $I_a^{\mathbf{X}'}(s)$ comme :

$$\begin{aligned}\Omega(\mathbf{X},\mathbf{X}',s,a) = & \{I \cap J \mid I \in I_a^{\mathbf{X}}(s), J \in I_a^{\mathbf{X}'}(s), I \cap J \neq \emptyset\} \\ \cup & \{I \mid I \in I_a^{\mathbf{X}}(s), \forall J \in I_a^{\mathbf{X}'}(s), I \cap J = \emptyset\} \\ \cup & \{J \mid J \in I_a^{\mathbf{X}'}(s), \forall I \in I_a^{\mathbf{X}}(s), I \cap J = \emptyset\}\end{aligned}$$

Informellement, $\Omega(\mathbf{X},\mathbf{X}',s,a)$ est l'ensemble de toutes les intersections d'intervalles (un de $I_a^{\mathbf{X}}(s)$ et un de $I_a^{\mathbf{X}'}(s)$), en plus des intervalles de chacun qui n'ont aucune intersection dans l'autre. Comme, par construction, tous les effets d'un intervalle de $I_a^{\mathbf{X}}(s)$ ou $I_a^{\mathbf{X}'}(s)$ produisent la même transition à partir d'un état s, pour n'importe quelle combinaison d'effets qui peut expliquer les transitions observées à la fois par $\mathbf{x}$ et $\mathbf{x}'$, il en existe un équivalent dans un des intervalles de $\Omega(\mathbf{X},\mathbf{X}',s,a)$ (Proposition 9). Nous mesurons alors $||\hat{T}^{\mathbf{X}}(\cdot|s,a) - \hat{T}^{\mathbf{X}'}(\cdot|s,a)||_1$ sur les intervalles de $\Omega(\mathbf{X},\mathbf{X}',s,a)$. La figure 2 présente l'ensemble d'intervalles $\Omega(\mathbf{X},\mathbf{X}',s,a)$ pour les observations de l'exemple 8 : les intervalles sont représentés par des segments horizontaux et un chevauchement vertical entre eux représente une intersection.

Maintenant, quand un intervalle $I \in I_a^{\mathbf{X}}(s)$ a une intersection non vide avec $J,J' \in I_a^{\mathbf{X}'}(s)$, il reste à déterminer combien des transitions observées pour I comptent pour J et J'. Ceci revient à déterminer une valeur β_K^I pour chaque $I \in I_a^{\mathbf{X}}(s)$ et $K \in \Omega(\mathbf{X},\mathbf{X}',s,a)$, $K \subseteq I$, et similairement pour chaque $J \in I_a^{\mathbf{X}'}(s)$, sous les contraintes suivantes :

$$\begin{aligned}&\forall I, && \textstyle\sum\{\beta_K^I \mid K \in \Omega(\mathbf{X},\mathbf{X}',s,a), K \subseteq I\} = C_a^{\mathbf{X}}(s,I) \\ &\forall J, && \textstyle\sum\{\beta_K^J \mid K \in \Omega(\mathbf{X},\mathbf{X}',s,a), K \subseteq J\} = C_a^{\mathbf{X}'}(s,J) \\ &\forall I,J,K, && \beta_K^I \geq 0,\ \beta_K^J \geq 0\end{aligned} \tag{2}$$

Pour un vecteur fixé $\vec{\beta}$ de valeurs pour les β_K^I et β_K^J, la distance L_1 résultante $||\hat{T}^{\mathbf{X}}(\cdot|s,a) - \hat{T}^{\mathbf{X}'}(\cdot|s,a)||_1$ est alors :

$$\delta_{\vec{\beta}}(\mathbf{X},\mathbf{X}',s,a) = \sum_{K \in \Omega(\mathbf{X},\mathbf{X}',s,a)} \left| \sum_{K \subseteq I} \frac{\beta_K^I}{C_a^{\mathbf{X}}(s)} - \sum_{K \subseteq J} \frac{\beta_K^J}{C_a^{\mathbf{X}'}(s)} \right| \tag{3}$$

Remarquons que le poids des intervalles de $I_a^{\mathbf{X}}(s)$ sans intersection dans $I_a^{\mathbf{X}'}(s)$ (et *vice-versa*) est pris en compte par des sommes vides. La figure 2 montre le calcul de distance pour les intervalles de l'exemple 8.

Par conséquent, au lieu d'avoir une mesure de distance définie de manière unique entre $\hat{T}^{\mathbf{X}}(\cdot|s,a)$ et $\hat{T}^{\mathbf{X}'}(\cdot|s,a)$, nous obtenons un ensemble de mesures *possibles*, en fonction de la manière dont les ambigüités sont résolues (c'est-à-dire la valeur de $\vec{\beta}$). Néanmoins, le fait est que nous pouvons calculer des bornes qui encadrent la distance *réelle* entre $\hat{T}^{\mathbf{X}}(\cdot|s,a)$ et $\hat{T}^{\mathbf{X}'}(\cdot|s,a)$ quand $\mathbf{X}$ est un ensemble condition candidat pour a.

Proposition 9
Supposons que $\mathbf{X}$ *soit l'ensemble condition correct pour* a*, et soit* $\mathbf{X}' \supset \mathbf{X}$*,* $|\mathbf{X}'| = 2k$*. Si* $s[\mathbf{X}]$ *et* $s[\mathbf{X}']$ *ont vu au moins* $m = \frac{2}{\epsilon^2} \ln \frac{2^E-2}{\delta}$ *transitions pour un état* s*, alors il existe des valeurs de* $\vec{\beta}$ *pour les intervalles de* $I_a^{\mathbf{X}}(s,a)$ *telles que* $||\hat{T}^{\mathbf{X}}(\cdot|s,a) - \hat{T}^{\mathbf{X}'}(\cdot|s,a)||_1$ *est au plus* 2ϵ *avec confiance* $1-\delta$*.*

Démonstration (idée). Partant du fait que la distribution cible $T(\cdot|s,a)$ a généré les transitions observées par $\mathbf{X}$ et $\mathbf{X}'$, avec une grande confiance les effets ont été observés par les deux. Par conséquent ils apparaissent dans les intervalles de $\Omega(\mathbf{X},\mathbf{X}',s,a)$. Puisque à la fois $\hat{T}^{\mathbf{X}}(\cdot|s,a)$ et $\hat{T}^{\mathbf{X}'}(\cdot|s,a)$ sont ϵ-proches de $T(\cdot|s,a)$ de par la valeur de m, le vecteur $\vec{\beta}$ qui reflète la distribution cible $T(\cdot|s,a)$ donne une faible distance $\delta_{\vec{\beta}}(\mathbf{X},\mathbf{X}',s,a)$. □

Autrement dit, le vecteur $\vec{\beta}$ qui minimise (resp. maximise) $\delta_{\vec{\beta}}(\mathbf{X},\mathbf{X}',s,a)$ donne une borne inférieure (resp. supérieure) sur la distance réelle entre l'ensemble condition cible $\mathbf{X}$ et ses sur-ensembles $\mathbf{X}'$.

En ce qui concerne la condition (1), il en découle que si $\max_{\vec{\beta}} \delta_{\vec{\beta}}(\mathbf{X},\mathbf{X}',s,a)$ est inférieure à ϵ pour tous $\mathbf{X}' \supset \mathbf{X}$ (et tous ont reçu m observations), alors $\hat{T}^{\mathbf{X}}(\cdot|s,a)$ est une estimation 2ϵ-correcte de $T(\cdot|s,a)$. A l'inverse, si $\min_{\vec{\beta}} \delta_{\vec{\beta}}(\mathbf{X},\mathbf{X}',s,a)$ est plus grand que ϵ, alors $\hat{T}^{\mathbf{X}}(\cdot|s,a)$ *n'est pas* précis. Enfin, avec grande confiance, l'ensemble condition correct $\mathbf{X}$ vérifie $\min_{\vec{\beta}} \delta_{\vec{\beta}}(\mathbf{X},\mathbf{X}',s,a) \leq \epsilon$, pour tous $\mathbf{X}' \supset \mathbf{X}$.

Ceci suggère deux généralisations de l'approche de la section précédente : remplacer la distance L_1 par $\min_{\vec{\beta}} \delta_{\vec{\beta}}(\mathbf{X},\mathbf{X}',s,a)$ garantit que l'ensemble condition correct sera toujours parmi les candidats, et permet d'en éliminer certains de manière correcte, tandis qu'utiliser $\max_{\vec{\beta}} \delta_{\vec{\beta}}(\mathbf{X},\mathbf{X}',s,a)$ permet de sélectionner des candidats *garantis* d'être corrects. Ce dernier choix peut permettre d'utiliser par la suite une approche minimisant le *regret maximal*.

Du point de vue des temps de calcul, on peut voir que la distance minimale peut être obtenue par la résolution d'un *programme linéaire* sur les variables β, sous les contraintes de l'équation 2, ce qui en pratique est relativement efficace. D'un autre côté, déterminer la distance maximale est plus difficile car maximiser de valeurs absolues nécessite d'utiliser un *programme linéaire mixte* (« *Mixed Integer Program* » (Diwekar, 2008)), ce qui est nettement plus coûteux en temps de calcul.

6 L'algorithme PSO-$R_{\max}$

Nous avons maintenant défini tous les éléments nécessaires à la construction de notre algorithme d'apprentissage par renforcement « *model-based* » : PSO-$R_{\max}$. Cet algorithme construit, par expérience dans l'environnement, un modèle de la tâche à résoudre en utilisant, pour chaque action a, un algorithme d'apprentissage de conditions AC_a comme défini précédemment. À chaque instant, il utilise un solveur de PDM (en l'occurrence *Value Iteration*) pour construire une politique afin de choisir la prochaine action à exécuter.

À chaque fois que le planificateur requiert des probabilités de transitions $T(\cdot|s,a)$, il demande une estimation $\hat{T}(\cdot|s,a)$ à AC_a. Alors, AC_a recherche un ensemble condition $\mathbf{X}$ de k variables tel que pour tout $\mathbf{X}' \supset \mathbf{X}$, $|\mathbf{X}'| = 2k$, $C_a^{\mathbf{X}'}(s) = m$, et tel que $\mathbf{X}$ satisfasse la condition (1). Si un tel ensemble est trouvé, il retourne $\hat{T}^{\mathbf{X}}(\cdot|s,a)$, sinon il retourne $\emptyset$. Le planificateur utilise alors :

$$Q(s,a) = \begin{cases} 1/(1-\gamma) & \textbf{si } \hat{T}(\cdot|s,a) = \emptyset, \\ R(s,a) + \gamma \sum_e \hat{T}(e|s,a) \max_{a'} Q(e(s),a') & \textbf{sinon} \end{cases} \quad (4)$$

pour mettre à jour ses Q-valeurs d'action.

Puisque $1/(1-\gamma)$ est la plus grande récompense possible dans n'importe quelle fonction de valeur, la première équation attire l'agent vers les états dont les transitions sont encore inconnues. Pour une présentation détaillée de cette technique d'exploration, nous renvoyons le lecteur à Brafman & Tennenholtz (2003).

On remarque que la connaissance *a priori* de la fonction de récompense est nécessaire. En effet, lors de l'apprentissage tous les états ne sont pas nécessairement visités, puisque l'on tire profit de la généralisation offerte par la représentation factorisée du problème. Si la fonction de récompense n'a pas de structure permettant son apprentissage de manière efficace (polynomiale), on perd le bénéfice de la structure en PSO des actions.

La mesure de distance introduite précédemment laisse la possibilité de choisir, soit la distance minimale, soit la distance maximale pour vérifier la condition (1). Des expériences préliminaires ont montré que le choix de la distance maximale n'était pas très concluant. A l'opposé, la distance minimale $\min_{\vec{\beta}} \delta_{\vec{\beta}}$ a révélé un très bon comportement lors des expériences (voir section 7). Plus précisément, les algorithmes d'apprentissage de conditions ont utilisé l'instanciation suivante de la condition (1) :

$$\forall \mathbf{X}' \supset \mathbf{X}, |\mathbf{X}'| = 2k, \quad \min_{\vec{\beta}} \delta_{\vec{\beta}}(\mathbf{X},\mathbf{X}',s,a) \leq \epsilon/2$$

Dans le cas où plusieurs candidats possibles sont retenus, celui qui minimise la distance minimale vers son sur-ensemble le plus éloigné est choisi :

$$\mathbf{X} = \operatorname*{argmin}_{\mathbf{X}} \max_{\mathbf{X}'} \min_{\vec{\beta}} \delta_{\vec{\beta}}(\mathbf{X},\mathbf{X}',s,a)$$

Enfin, la distribution d'effets $\hat{T}^{\mathbf{X}}(\cdot|s,a)$ qui est renvoyée est celle sur les effets de $\Omega(\mathbf{X},\mathbf{X}',s,a)$ (en prenant arbitrairement un effet dans chaque intervalle). La distribution de probabilité sur ces effets est donnée par les valeurs $\{\beta_K^I \mid I \in I_a^{\mathbf{X}}(s), K \in \Omega(\mathbf{X},\mathbf{X}',s,a)\}$ retenues pour le calcul de la distance.

Le pseudo-code de PSO-$R_{\max}$ est donné dans l'algorithme 1.

Algorithme 1 : PSO-$R_{\max}$

Données : k : nombre de variables présentes dans les conditions d'action.
Données : m : seuil d'exemples pour stopper l'exploration.
Données : R : fonction de récompense du problème.
pour chaque *action* a **faire**
  Instancier un appreneur de conditions AC_a avec les paramètres k,m,ϵ
  $Q(s,a) \leftarrow 1/(1-\gamma)$ pour tout état s
Observer l'état courant s_1
pour tous les *les instants* $t = 1, 2, \ldots$ **faire**
  Exécuter l'action $a' \in \text{argmax}_a Q(s_t,a)$
  Observer le nouvel état s_{t+1}
  Mettre à jour $AC_{a'}$ avec (s_t,s_{t+1})
  répéter
    pour chaque *action* a *et état* s **faire**
      $\hat{T}(\cdot|s,a) \leftarrow AC_a(s)$
      Mettre à jour $Q(s,a)$ en utilisant l'équation 4 avec $\hat{T}(\cdot|s,a)$
  jusqu'à *convergence*

7 Expériences

Afin de tester le comportement en pratique de l'heuristique de distance minimale choisie, nous avons évalué PSO-$R_{\max}$ sur deux problèmes à 512 états ($n = 9$ variables) et mesuré leur récompense accumulée au cours du temps, en prenant la moyenne sur 10 simulations. L'algorithme non factorisé $R_{\max}$ a lui aussi été lancé sur ces problèmes afin de comparer les deux approches. Les deux algorithmes ont utilisé un nombre d'exemples $m = 10$.

Le domaine Builder

Ce domaine, présenté par Dearden & Boutilier (1997), fait intervenir une ligne de production sur laquelle un agent doit nettoyer, mettre en forme, peindre et percer deux objets A et B. Chaque action a un effet « positif » (qui augmente la récompense), un effet « négatif » et l'effet vide. De plus, il n'y a pas d'action « attendre » déterministe, l'agent doit donc agir indéfiniment et prendre à tout moment le risque de voir sa récompense décroître.

La domaine est décrit par 9 variables. Les 8 premières décrivent le statut des objets A et B : $Clean_X$, $Painted_X$,$Shaped_X$,$Drilled_X$ avec leur signification évidente. Une variable $Joined$ indique si les deux objets sont joints ou non. L'agent perçoit une récompense de $+0{,}1$ (resp. $-0{,}1$) pour chaque objet propre (resp. sale), $+0{,}2$ pour chaque objet peint ($-0{,}2$ sinon) et $+0{,}4$ si les objets sont joints ($-0{,}4$ sinon). Les actions disponibles pour chacun des objets sont *Wash*, *Paint*, *Shape*, *Drill*, et deux actions sont disponibles pour joindre les objets : *Glue* qui a pour effet de salir les objets et *Bolt*, qui n'altère pas les objets mais requiert qu'ils soient préalablement percés (avec l'action *Drill*). Dans notre variante du problème, k vaut 2 pour l'action *Glue*, 5 pour l'action *Bolt* et 1 pour toutes les autres actions. Les actions ont jusqu'à 4 variables modifiées et 3 effets différents. La description complète des actions du domaine est présentée a l'annexe A.

La figure 3 montre une comparaison entre PSO-$R_{\max}$ et $R_{\max}$ sur ce domaine. Ici, PSO-$R_{\max}$ exploite efficacement la représentation compacte en PSO du problème et est très rapide à converger vers une politique quasi-optimale (et ce, en explorant en moyenne 57 états seulement). $R_{\max}$, de son côté, a besoin de visiter les 512 états du problème avant d'agir correctement. Comme les actions ont des effets corrélés, SLF-$R_{\max}$ ne peut pas être utilisé sur ce problème.

De plus, après avoir ajouté 3 variables sans signification précise avec des actions déterministes pour changer leur valeur, ce qui étend donc artificiellement le problème à 4096 états, le temps de convergence de PSO-$R_{\max}$ n'a subi qu'un faible impact, comme on peut le voir sur la figure 3.

Le domaine Stocktrading

Par souci de comparaison avec une approche par DBN comme SLF-$R_{\max}$, nous avons aussi exécuté notre algorithme sur le domaine « *Stocktrading* » (Strehl *et al.*, 2007), où les effets des actions sur les variables

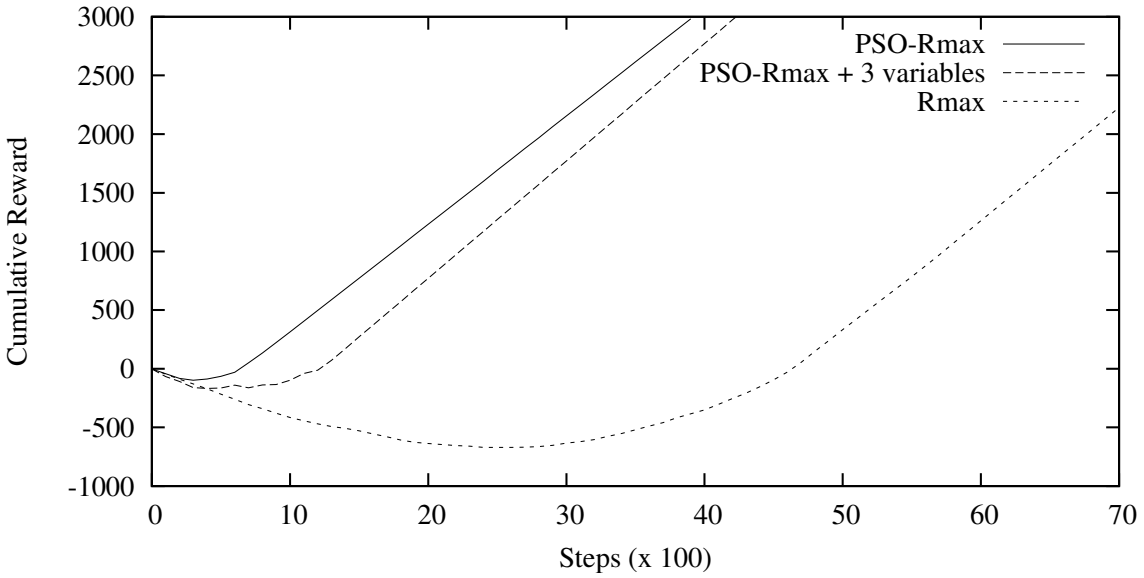


FIG. 3 – Récompense cumulée sur le domaine Builder

sont indépendants les uns des autres. Ce domaine met en scène un agent boursier pouvant acheter et vendre des titres répartis en trois secteurs de deux titres chacun. Les actions déterministes *Buy* et *Sell* permettent respectivement d'acheter ou de vendre les titres d'un secteur. Pour chaque secteur, une variable indique si l'agent possède ou non les titres du secteur, et pour chaque titre une variable indique si son cours augmente ou diminue. La probabilité p qu'un titre augmente au temps $t+1$ est donnée par

$$p = 0{,}1 + 0{,}8 \times \frac{\text{nombre de titres du secteur augmentant au temps } t}{\text{nombre de titres dans le secteur}}$$

Réciproquement, la probabilité qu'un titre décroisse est $1-p$. L'annexe B présente le Réseau Bayésien Dynamique associé à ce domaine.

Avec une représentation en PSO, un domaine *Stocktrading* à 3 secteurs de 2 titres chacun a une valeur de $k=7$ pour les actions *Buy* et *Sell* et $k=6$ pour l'action *Wait*. Comme k est proche du nombre total de variables ($n=9$), les PSO ne sont clairement pas une représentation compacte pour ce domaine. À l'inverse, SLF-$R_{\max}$ apprend un modèle pour chaque action et chaque variable d'effet indépendamment, mais tire parti du fait que chaque variable d'effet n'a que $k=2$ parents dans le DBN. Les deux approches ont une complexité en nombre d'exemples exponentielle en *leur* paramètre k ; SLF-$R_{\max}$ tire donc ici un meilleur parti de la structure du problème.

La figure 4 montre une comparaison entre $R_{\max}$ et PSO-$R_{\max}$. On peut voir que notre algorithme se comporte de manière similaire à $R_{\max}$, alors que d'après Strehl *et al.* (2007), SLF-$R_{\max}$ est très efficace en comparaison (il ne nécessite que 1600 étapes pour converger). Néanmoins, du fait que $2k \geq n$ pour PSO-$R_{\max}$, aucune structure ne peut être exploitée, ce qui semble confirmer que l'heuristique de distance minimale ne dégrade pas pour autant les résultats.

8 Conclusion et perspectives

Nous avons proposé l'algorithme PSO-$R_{\max}$, qui apprend efficacement à agir dans des domaines qui peuvent être représentés de façon compacte avec des Opérateurs STRIPS Probabilistes Propositionnels (ou des DBNs avec arcs synchrones). Malgré les inévitables *effets ambigus* qui requièrent des choix heuristiques, PSO-$R_{\max}$ se comporte bien en pratique. De plus, nous avons démontré qu'il ne nécessite qu'un nombre d'exemples polynomial pour se comporter ϵ-efficacement lorsque les effets ne sont pas ambigus (ou sont connus à l'avance).

Nos expériences ont été conçues pour valider, en termes de *nombre d'étapes d'apprentissage*, nos choix heuristiques. Les travaux en cours portent sur des expériences à plus grande échelle et des comparaisons avec d'autres approches, telles que celle de Pasula *et al.* (2007) qui abordent le problème d'apprentissage

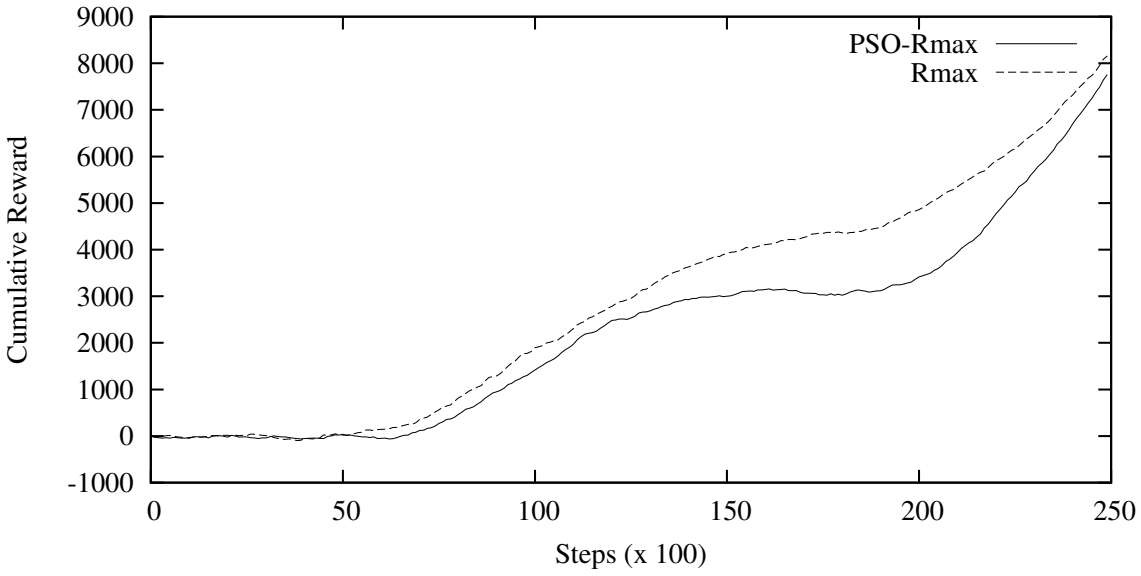


FIG. 4 – Récompense cumulée sur le domaine Stocktrading

de structure (et non par renforcement) dans un cadre relationnel. Le passage à des domaines de plus grande taille nécessite cependant l'utilisation d'un planificateur approximatif qui exploite la représentation en PSO. Il faut alors extraire un modèle unique et compact pour chaque action. Actuellement, pour un état s donné, une hypothèse sur le modèle (c'est-à-dire un ensemble condition de k variables) est choisie. Cependant, pour une même action, l'hypothèse retenue peut varier selon l'état. Des résultats (non présentés dans cet article) ont montré que choisir l'ensemble condition qui minimise la pire distance pour tout état donne les mêmes résultats qu'en section 7, tout en fournissant un modèle compact pour chaque action.

Une autre voie à explorer consiste à tirer parti des avantages respectifs des DBN et des PSO en étendant notre approche aux *actions avec aspects* (Dearden & Boutilier, 1997), qui permettent de représenter des *ensembles* de variables indépendants à l'intérieur d'une action. Cette représentation est pertinente quand les transitions proviennent à la fois des actions de l'agent et d'événements exogènes correspondant à la dynamique de l'environnement. De plus, on peut envisager d'utiliser notre notion de distance pour étendre des algorithmes d'induction d'arbres de décision tels que SPITI (Degris *et al.*, 2006).

Références

BOUTILIER C., DEAN T. & HANKS S. (1999). Decision theoretic planning : Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, **11**(1), 1–94.

BOUTILIER C., DEARDEN R. & GOLDSZMIDT M. (2000). Stochastic Dynamic Programming with Factored Representations. *Journal of Artificial Intelligence Research*.

BRAFMAN R. & TENNENHOLTZ M. (2003). R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, **3**, 213–231.

DEAN T. & KANAZAWA K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence*, **5**(3), 142–150.

DEARDEN R. & BOUTILIER C. (1997). Abstraction and approximate decision-theoretic planning. *Artificial Intelligence*, **89**, 219–283.

DEGRIS T., SIGAUD O. & WUILLEMIN P.-H. (2006). Learning the structure of factored markov decision processes in reinforcement learning problems. In *Proceedings ICML 2006*, p. 257–264.

DIWEKAR U. (2008). *Introduction to Applied Optimization*. Springer, 2 edition.

GUESTRIN C., KOLLER D., PARR R. & VENKATARAMAN S. (2003). Efficient solution algorithms for factored mdps. *Journal of Artificial Intelligence Research*, **19**, 399–468.

GUESTRIN C., PATRASCU R. & SHUURMANS D. (2002). Algorithm-directed exploration for model-based reinforcement learning in factored MDPs. In *Proceedings ICML 2002*, p. 235–242.

HOEY J., ST-AUBIN R., HU A. & BOUTILIER C. (1999). SPUDD : Stochastic planning using decision diagrams. In *Proceedings UAI 1999*, p. 279–288.

JONSSON A. & BARTO A. (2007). Active learning of dynamic bayesian networks in markov decision processes. In *Proceedings SARA 2007*, p. 273–284.

KEARNS M. & KOLLER D. (1999). Efficient reinforcement learning in factored MDPs. In *Proceedings IJCAI 1999*, volume 16, p. 740–747.

KUSHMERICK N. (1995). An algorithm for probabilistic planning. *Artificial Intelligence*, **76**, 239–286.

PASULA H., ZETTLEMOYER L. & KAELBLING L. (2007). Learning symbolic models of stochastic domains. *Journal of Artificial Intelligence Research*, **29**, 309–352.

STREHL A., DIUK C. & LITTMAN M. (2007). Efficient structure learning in factored-state MDPs. In *Proceedings AAAI 2007*, p. 645–650.

STREHL A., LI L. & LITTMAN M. (2009). Reinforcement Learning in Finite MDPs : PAC Analysis. *Journal of Machine Learning Research*, **10**, 2413–2444.

SUTTON R. & BARTO A. (1998). *Reinforcement Learning : An introduction*. MIT Press.

WALSH T., SZITA I., DIUK C. & LITTMAN M. (2009). Exploring compact reinforcement-learning representations with linear regression. In *Proceedings UAI 2009*.

WEISSMAN T., ORDENTLICH E., SEROUSSI G. & VERDU S. (2003). Inequalities for the L1 deviation of the emprical distribution. *Hewlett Packard Labs, Technical Report*.

A Description PSO du domaine BUILDER

Condition	Effets	Prob.
PAINT$_A$		
$Clean_A$	$Painted_A$	0,75
	$\overline{Clean}_A$	0,15
	$\emptyset$	0,05
PAINT$_B$		
$Clean_B$	$Painted_B$	0,75
	$\overline{Clean}_B$	0,15
	$\emptyset$	0,05
SHAPE$_A$		
$\overline{Joined}$	$\overline{Painted}_A Shaped_A$	0,8
	$\overline{Painted}_A \overline{Shaped}_A \overline{Clean}_A \overline{Drilled}_A$	0,1
	$\overline{Painted}_A$	0,1
$Joined$	$\overline{Joined}$	1
SHAPE$_B$		
$\overline{Joined}$	$\overline{Painted}_B Shaped_B$	0,8
	$\overline{Painted}_B \overline{Shaped}_B \overline{Clean}_B \overline{Drilled}_B$	0,1
	$\overline{Painted}_B$	0,1
$Joined$	$\overline{Joined}$	1
DRILL$_A$		
$\overline{Joined}$	$\overline{Drilled}_A$	0,9
	$\emptyset$	0,1
$Joined$	$\overline{Joined}$	0,9
	$\emptyset$	0,1
DRILL$_B$		
$\overline{Joined}$	$\overline{Drilled}_B$	0,9
	$\emptyset$	0,1
$Joined$	$\overline{Joined}$	0,9
	$\emptyset$	0,1
WASH$_A$		
$\overline{Clean}_A$	$Clean_A$	0,9
	$\emptyset$	0,1
WASH$_B$		
$\overline{Clean}_B$	$Clean_B$	0,9
	$\emptyset$	0,1
BOLT		
$\overline{Joined} \wedge Shaped_A \wedge Shaped_B \wedge Drilled_A \wedge Drilled_B$	$Joined$	0,8
	$\emptyset$	0,2
GLUE		
$Shaped_A \wedge Shaped_B$	$\overline{Clean}_A \overline{Clean}_B Joined$	0,35
	$Joined$	0,35
	$\overline{Clean}_A \overline{Clean}_B$	0,15
	$\emptyset$	0,15
$\overline{Shaped}_A \vee \overline{Shaped}_B$	$\overline{Clean}_A \overline{Clean}_B$	0,5
	$\emptyset$	0,5

B Description DBN du domaine STOCKTRADING