

Fouille de motifs séquentiels pour l'élicitation de stratégies à partir de traces d'interactions entre agents en compétition

Guillaume Bosc*, Mehdi Kaytoue*, Chedy Raïssi**, Jean-François Boulicaut*

*Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France

**INRIA Nancy - Grand Est, Villers-lès-Nancy, F-54600, France
prenom.nom@insa-lyon.fr, chedy.raïssi@inria.fr

Résumé. Pour atteindre un but, tout agent en compétition élabore inévitablement des stratégies. Lorsque l'on dispose d'une certaine quantité de traces d'interactions entre agents, il est naturel d'utiliser la fouille de motifs séquentiels pour découvrir de manière automatique ces stratégies. Dans cet article, nous proposons une méthodologie qui permet l'élicitation de stratégies et leur capacité à discriminer une réussite ou un échec. La méthodologie s'articule en trois étapes : (i) les traces brutes sont transformées en une base de séquences selon des choix qui permettent, (ii) l'extraction de stratégies fréquentes, (iii) lesquelles sont munies d'une mesure originale d'émergence. C'est donc une méthodologie de découverte de connaissances que nous proposons. Nous montrons l'intérêt des motifs extraits et la faisabilité de l'approche à travers des expérimentations quantitatives et qualitatives sur des données réelles issues du domaine émergent du sport électronique.

1 Introduction

Tout processus observé par le biais d'un capteur ou d'une autre technique d'acquisition fournit des traces. On peut reprendre par exemple la définition de Lund et Mille (2009) : *l'observation d'un processus exige la collecte d'observés temporellement situés si l'on veut prendre en compte le "quand", le "comment" et le "pourquoi" pour un "qui" particulier*. Chaque trace est décrite par une liste ordonnée d'actions, lesquelles sont effectuées par un agent, situées dans le temps et possiblement dans l'espace. Il arrive également que la finalité d'une trace soit matérialisée par un but atteignable au moyen des actions qui modifient l'état inhérent au processus. On peut par exemple noter les processus d'apprentissage ou de rééducation qui consistent en une série d'exercices réussis ou non par l'agent.

Il arrive souvent qu'une trace implique plusieurs agents où leurs actions sont mutuellement dépendantes. On parle alors de traces d'interactions (Lund et Mille (2009)). Lorsque l'on étudie le cas d'interactions entre deux agents en compétition pour un même but, un exemple classique qui a servi de support à de nombreux travaux en intelligence artificielle est celui des jeux à deux joueurs (Fürnkranz (2001)). Au jeu des échecs par exemple, deux agents sont en compétition et les actions qu'ils effectuent sont mutuellement dépendantes, modifient l'état du jeu, et ont pour finalité la victoire. On peut également considérer l'exemple de traces d'interac-

tions générées par deux équipes de joueurs de football et acquises par des procédés d'analyse vidéo ou d'annotation par des observateurs.

Inévitablement, chaque agent met alors en place des stratégies pour atteindre le but avant son adversaire, et il est intéressant de découvrir automatiquement des stratégies fréquemment utilisées. En effet, si l'on possède suffisamment de traces d'interactions, on peut chercher des généralisations ou abstractions de traces qui discriminent le plus le succès, la défaite ou l'équilibre. Par exemple, le système Deep Blue d'IBM utilisa lors de son match contre Kasparov un *livre d'ouvertures* construit à partir de 700 000 parties d'échecs jouées par des grands maîtres. Pour tout état du jeu atteignable par au plus 30 coups, le système avait à sa disposition une évaluation de la qualité du prochain coup y menant (ratio du nombre de parties gagnées par rapport aux nombres de parties impliquant la série de coups menant à l'état considéré). La fréquence de chaque coup est aussi utilisée dans l'évaluation pour donner une tolérance aux exceptions. Le cas des échecs, et des jeux étudiés en intelligence artificielle en général, est particulier dans la mesure où il se joue tour par tour, dans un environnement certain (à tout moment les joueurs connaissent l'état complet du jeu) et fiable (une action ne peut pas échouer). Cela rend relativement directe la construction d'un livre d'ouvertures.

Nous considérons dans cet article le cas d'interactions entre agents en concurrence pour un même et unique but, dans un environnement incertain (non complètement connu), où les actions peuvent se produire simultanément en temps réel et sont bruitées (certaines actions ne modifient pas l'état du jeu). Nous proposons une méthodologie basée sur l'extraction de motifs séquentiels fréquents (Pei et al. (2001)) pour caractériser des stratégies fréquentes, et la notion de motifs émergents (Dong et Li (1999)) pour qualifier leur équilibre (gagnantes, perdantes, ou équilibrées). Un motif fréquent dénote alors une suite d'actions fréquemment utilisée par les deux agents lors de leur interaction. L'émergence d'un motif est fonction de sa fréquence d'apparition en cas de victoire et de défaite. Cependant, nous pouvons difficilement utiliser la notion d'émergence comme utilisée classiquement en fouille de données, c'est-à-dire où chaque séquence est munie d'une classe (victoire ou défaite). Parce que la classe est partagée entre deux agents dont les actions sont dépendantes, ce sont les événements de la séquence d'interaction qui portent l'information de la classe. Nous montrons alors comment définir cette mesure et la calculer avec un parcours efficace de l'arbre de motifs fréquents généré par un algorithme existant d'extraction de motifs séquentiels fréquents (*PrefixSpan*, Pei et al. (2001)).

L'article est organisé comme suit. Tout d'abord nous rappelons les notions d'extraction de fouille de motifs séquentiels nécessaires au développement de notre méthode (Section 2). Nous présentons ensuite notre méthodologie dans la Section 3 qui s'articule en trois étapes : (i) transformation des traces brutes en base de séquences, (ii) fouille et (iii) calcul de la mesure d'émergence. La Section 4 présente les principes algorithmes utilisés pour le calcul de l'émergence et la suppression d'une redondance bien définie dans la collection de motifs extraits. Enfin la Section 5 présente à travers des expérimentations quantitatives et qualitatives la faisabilité et l'utilité de notre approche sur des données du monde réel issues d'un jeu de stratégie en temps réel compétitif dans le domaine d'application émergent du sport électronique.

2 Extraction de motifs séquentiels fréquents

Nous reprenons ici les définitions classiques pour introduire le problème d'extraction de motifs séquentiels fréquents. Nous nous limitons aux notions utilisées dans le reste de l'article.

Définition (Item, itemset, séquence). Soit \mathcal{I} un ensemble fini d'éléments (*items*). Un sous-ensemble non vide et ordonné de \mathcal{I} est appelé *itemset*. Une séquence est une liste ordonnée d'éléments $\langle s_1, \dots, s_l \rangle$, où $l > 0$ et $s_i \subseteq \mathcal{I}$ est un itemset pour tout $1 \leq i \leq l$.

Une interprétation possible d'une séquence $s = \langle s_1, \dots, s_l \rangle$ est la suivante : chaque itemset représente un ensemble d'évènements se produisant au même moment (où l'ordre dans lequel ils apparaissent n'est pas significatif : on peut donc munir une relation d'ordre annexe à cet ensemble, par exemple lexicographique), l'ordre entre deux itemsets représente l'ordre dans lequel les deux groupes d'évènements se produisent au cours du temps.

Définition (Sous-séquence). Une séquence $s = \langle s_1, s_2, \dots, s_n \rangle$ est une *sous-séquence* d'une séquence $s' = \langle s'_1, s'_2, \dots, s'_m \rangle$ (ou s' est une *sur-séquence* de s), noté $s \sqsubseteq s'$, s'il existe des entiers $1 \leq j_1 < j_2 < \dots < j_n \leq m$ tel que $s_1 \subseteq s'_{j_1}, s_2 \subseteq s'_{j_2}, \dots, s_n \subseteq s'_{j_n}$.

Définition (Support d'une séquence). Soit $\mathcal{D} = \{s_1, s_2, \dots, s_n\}$ une base de séquences. On définit alors le *support* d'une séquence s dans \mathcal{D} comme l'ensemble des séquences de \mathcal{D} qui sont des sur-séquences de s , soit : $support(s, \mathcal{D}) = \{s' \mid s \sqsubseteq s', s' \in \mathcal{D}\}$. On écrira simplement $support(s)$ quand aucune confusion sur la base ne peut être faite. On dit qu'une séquence s est *fréquente* dans une base \mathcal{D} , si son support contient au moins $minSupp$ éléments. Selon le contexte, le support désigne soit cet ensemble d'éléments, soit sa cardinalité.

Problème (Extraction de motifs séquentiels fréquents). Étant donné un seuil de support minimal $minSupp \geq 1$ et une base de séquences \mathcal{D} , le problème consiste à trouver toutes les séquences s telles que $|support(s)| \geq minSupp$.

Exemple. La table 1 ci-contre présente une base $\mathcal{D} = \{s_1, s_2, s_3, s_4\}$ avec $\mathcal{I} = \{a, b, c, d, e, f\}$. Pour plus de lisibilité, nous simplifions la notation d'une séquence ainsi : $\langle \{a\}, \{a, b, c\}, \{a, c\}, \{d\}, \{c, f\} \rangle$ se note $\langle a\{abc\}\{ac\}d\{cf\} \rangle$. Dans cet exemple, $\langle abc \rangle$ est une sous-séquence de s_1 et s_4 mais pas de s_2 , ni de s_3 . Avec $minSupp = 3$, $\langle acc \rangle$ est fréquent, $\langle a\{bc\}a \rangle$ ne l'est pas.

id	description	classe
s_1	$\langle a\{abc\}\{ac\}d\{cf\} \rangle$	+
s_2	$\langle \{ad\}c\{bc\}\{ae\} \rangle$	+
s_3	$\langle \{ef\}\{ab\}\{df\}cb \rangle$	-
s_4	$\langle eg\{af\}cbc \rangle$	-

TAB. 1 – Une base de séquences

Considérons maintenant que l'on dispose d'une fonction *classe* : $\mathcal{D} \rightarrow \{+, -\}$, qui associe à chaque séquence $s \in \mathcal{D}$ sa classe positive ou négative¹, $classe(s) \in \{+, -\}$. On peut alors partitionner la base de séquences \mathcal{D} selon les classes (on note $\mathcal{D} = \mathcal{D}^+ \cup \mathcal{D}^-$) et caractériser le pouvoir discriminant d'un motif pour une classe à l'aide d'un taux de croissance (Dong et Li (1999); Chan et al. (2003)).

Définition (Taux de croissance). Soit une base de séquences $\mathcal{D} = \mathcal{D}^+ \cup \mathcal{D}^-$, le taux de croissance d'un motif séquentiel s depuis la base \mathcal{D}^x vers \mathcal{D}^y ($x \neq y$ et $x, y \in \{+, -\}$), est défini par

$$tauxCroissance(s, \mathcal{D}^x) = \frac{|support(s, \mathcal{D}^x)|}{|\mathcal{D}^x|} \times \frac{|\mathcal{D}^y|}{|support(s, \mathcal{D}^y)|}$$

De fait, plus $tauxCroissance(s, \mathcal{D}^x)$ est élevé, plus le motif s est caractéristique de la classe x . On peut alors restreindre la tâche d'extraction de motifs séquentiels fréquents pour n'extraire que ceux dont le taux de croissance pour une classe est supérieur à un seuil donné, on dira qu'il s'agit de motifs émergents. Un exemple issu de la table 1 est : $tauxCroissance(\langle cb \rangle, -) = \frac{2}{2} \times \frac{2}{1} = 2$, i.e. ici $\langle cb \rangle$ est deux fois plus présent dans la classe $-$ que dans la classe $+$.

1. On peut aussi considérer plusieurs classes comme par exemple dans (Chan et al., 2003).

3 Une méthode d'extraction de stratégies à partir de traces

Nous considérons le problème d'élicitation de stratégies à partir d'une base de traces d'interaction entre agents en concurrence pour un même et unique but, dans un environnement incertain et où les actions peuvent se produire simultanément en temps réel. Nous définissons alors formellement la notion de trace et motivons l'utilité des motifs séquentiels pour caractériser des stratégies. Pour quantifier la capacité d'un motif à discriminer des cas de victoires ou de défaites, nous montrons que la notion classique d'émergence définie à l'aide d'un taux de croissance n'est pas satisfaisante, et nous introduisons alors une nouvelle mesure.

Dans cet article, une trace dénote l'interaction entre deux agents en compétition où leurs actions sont généralement dépendantes, et un seul atteint l'objectif. Une trace est alors une série d'éléments observés, chacun représentant "un agent réalisant une action à un instant".

Définition (Élément observé). Soient A un ensemble d'actions, P un ensemble d'agents qui réalisent des actions, et $T \subseteq \mathbb{N}$. On définit $E \subseteq A \times P \times T$ comme l'ensemble des évènements où $e = (a, p, t) \in E$ est l'action a réalisée par l'agent p au temps t .

Définition (Trace d'interaction). On note $s = \langle e_1, \dots, e_n \rangle$ une trace d'interaction avec $e_i \in E, \forall i \in [1, n]$. Une trace implique exactement deux agents de P . Une collection de traces est notée \mathcal{D} , et la fonction $res : \mathcal{D} \rightarrow P$ associe à chaque trace l'agent qui a gagné.

Exemple. La Table 2 (gauche) présente une base de traces : $s = \langle (a_1, p_1, 12), (a_3, p_1, 25), (a_4, p_2, 68) \rangle$ est une interaction entre les agents p_1 et p_2 , et $res(s) = p_1$ est le gagnant. $(a_1, p_1, 12)$ se lit : "l'agent p_1 a réalisé l'action a_1 au temps 12".

Discrétisation des traces. Dans de nombreux cas, si l'intervalle de temps entre deux actions identiques est faible, on peut considérer ces dernières comme simultanées ou répétées et donc n'en garder qu'une seule. C'est le cas de notre application de jeu de stratégie en temps réel (Section 6). Se rapprochant d'une interaction "tour par tour" où les acteurs peuvent agir simultanément, nous utilisons une discrétisation par fenêtres disjointes de taille fixe w à partir du temps zéro : un temps $t \in T$ est alors remplacé par $\lceil \frac{t}{w} \rceil$. Cette discrétisation est illustrée par la Table 2 (droite) avec $w = 30$.

Trace originale $s \in \mathcal{D}$	$res(s)$	Trace s discrétisée et transformée
$\langle (a_1, p_1, 12), (a_3, p_1, 25), (a_4, p_2, 68) \rangle$	p_1	$\langle \{(a_1, +, 1)(a_2, +, 1)\}\{(a_4, -, 3)\} \rangle$
$\langle (a_1, p_2, 11), (a_2, p_2, 15), (a_1, p_2, 35), (a_4, p_4, 70) \rangle$	p_2	$\langle \{(a_1, +, 1)(a_2, +, 1)\}\{(a_1, +, 2)\}\{(a_4, -, 3)\} \rangle$
$\langle (a_1, p_3, 15), (a_2, p_2, 29), (a_4, p_4, 61), (a_4, p_4, 63) \rangle$	p_3	$\langle \{(a_1, +, 1)(a_2, +, 1)\}\{(a_4, -, 3)\} \rangle$
$\langle (a_1, p_6, 12), (a_2, p_6, 22), (a_4, p_7, 88), (a_4, p_7, 89) \rangle$	p_7	$\langle \{(a_1, -, 1)(a_2, -, 1)\}\{(a_4, +, 3)\} \rangle$

TAB. 2 – Une base de traces d'interaction : chaque trace $s \in \mathcal{D}$ a un gagnant et est transformée selon cette information avec une taille de fenêtre $w = 30$.

Transformation de traces. Pour éliciter des stratégies à partir de \mathcal{D} , nous cherchons des généralisations des séquences de la base qui apparaissent fréquemment. On se tourne naturellement vers l'extraction de motifs séquentiels fréquents. Cela vient avec plusieurs problèmes : (i) un motif séquentiel est une généralisation commune à plusieurs interactions entre deux agents identiques, ainsi on ne peut pas caractériser de stratégies utilisées en général par l'ensemble des agents de la base de traces et les motifs concernant les couples d'agents les plus représentés seront extraits ; (ii) on ne peut pas utiliser directement la classe d'une séquence pour calculer un taux de croissance reflétant la capacité victoire/défaite d'un motif séquentiel. Voyons ces

problèmes séparément. Dans le cas (i), une abstraction de l'information "agent" est nécessaire. Puisque l'on cherche à éliciter des stratégies gagnantes/perdantes, nous pouvons remplacer dans chaque élément observé l'agent par un élément de $\{+, -\}$ selon qu'il ait gagné (+) ou perdu (-). Ainsi, une séquence fréquente représente une série d'actions des deux agents qui a fréquemment amené au même résultat. Cette transformation est illustrée par la Table 2 (droite). Dans le cas (ii), la classe associée à une trace ne permet pas l'utilisation de motifs séquentiels émergents : c'est la victoire/défaite ($\{+, -\}$) qui devrait être utilisée comme classe, mais cette information est caractérisée par les événements observés et non par la trace. Nous montrons une manière de définir un taux de croissance dans ce cas.

Elicitation des stratégies. À partir de maintenant, \mathcal{D} dénote une base de séquences obtenue après discrétisation et transformation d'une base de traces comme expliqué ci-avant : l'ensemble des items \mathcal{I} (ou alphabet), à partir duquel les séquences sont définies, est composée d'éléments $e \in A \times R \times T$, avec $R = \{+, -\}$ et $e = (a, r, t)$ signifie qu'au temps t , une action a a été faite par l'agent victorieux si $r = +$ ou perdant si $r = -$. Alors, les motifs séquentiels fréquents extraits donnent les suites d'action les plus utilisées et chacune se comprend comme la combinaison de deux stratégies (celle de chaque joueur) où l'une apparaît gagnante et l'autre perdante. Cependant, on dira simplement d'un motif extrait qu'il représente une stratégie (après validation/interprétation par un expert).

Exemple. Dans la base illustrée par la Table 2 (gauche), avec $\min Supp = 3$, le motif séquentiel $s = \langle \{(a_1, +, 1)(a_2, +, 1)\}\{(a_4, -, 3)\} \rangle$ est fréquent ($|support(s)| = 3$).

Stratégies victorieuses, perdantes, et équilibrées. Nous proposons maintenant une mesure de type *taux de croissance*, qui reflète l'équilibre d'une stratégie. Intuitivement, cette mesure nous donne la proportion de victoire/défaite d'une stratégie. Nous introduisons tout d'abord la notion de *duale d'une séquence*, qui reflète les mêmes événements mais où l'issue est changée : l'agent qui réussit à atteindre le but dans une séquence donnée échoue dans la dite duale.

Définition (Événement dual, séquence duale). Soit $e = (a, r, t) \in A \times R \times T$ un événement qui dénote une action a au temps t et réalisée par l'agent victorieux si $r = +$ (perdant sinon). Le dual de e est noté $\tilde{e} = (a, R \setminus r, t)$. Le dual d'une séquence s est noté \tilde{s} : il s'agit de la séquence s où chaque événement (item) est remplacé par son dual.

Exemple. Soit $s = \langle \{(a_1, +, 1)(a_2, +, 1)\}\{(a_4, -, 3)\} \rangle$ un motif séquentiel fréquent. Il admet $\tilde{s} = \langle \{(a_1, -, 1)(a_2, -, 1)\}\{(a_4, +, 3)\} \rangle$ comme séquence duale.

Ainsi, un motif séquentiel et son dual représentent la même stratégie, mais la suite d'actions gagnante (reps. perdante) dans un motif devient perdante (resp. gagnante) dans son dual. Alors, il est direct de définir un taux de croissance pour un motif qui permet de juger l'équilibre d'une stratégie comme un ratio impliquant les supports du motif et de son dual.

Définition (Mesure d'équilibre). Soient un motif séquentiel fréquent s et sa séquence duale \tilde{s} , on définit la *mesure d'équilibre* de s comme : $balance(s) = \frac{|support_{\mathcal{D}}(s)|}{|support_{\mathcal{D}}(s)| + |support_{\mathcal{D}}(\tilde{s})|}$.

Exemple. Dans la Table 2, on trouve $balance(\langle \{(a_1, +, 1)(a_2, +, 1)\}\{(a_4, -, 3)\} \rangle) = \frac{3}{3+1} = 0.75$. Cela signifie que lorsque l'on rencontre une interaction où un joueur x réalise a_1 puis a_2 , et qu'un joueur y réalise ensuite a_4 , il y a 75% de cas où x gagne.

Il suit que $balance(s) \in]0; 1]$ et $balance(s) + balance(\tilde{s}) = 1$. Une valeur de 0,5 dénote un équilibre parfait, 0 ou 1 une situation extrême. Notons aussi le cas particulier suivant : toute séquence s symétrique (*ie.* chaque itemset de s contient un item et son dual) respecte

$balance(s) = 0,5$. Enfin, notons que pour deux séquences fréquentes de même support, les séquences duales correspondantes peuvent avoir des supports différents ce qui ne nous permet pas d'extraire les motifs fermés fréquents seulement (définis par Yan et al. (2003)). Par exemple, soient s_1 et s_2 deux motifs séquentiels fréquents tel que $support(s_1) = support(s_2)$ et $s_1 \sqsubset s_2$, on peut avoir $support(\tilde{s}_1) > 0$ et $support(\tilde{s}_2) = 0$, de fait $balance(s_1) < 1$ et $balance(s_2) = 1$. Enfin, il peut y avoir redondance dans la collection de motifs extraits.

Observation (Redondance dans la collection de motifs). Soient s et \tilde{s} deux séquences fréquentes pour un seuil $minSupp$ donné. On peut alors garder dans la collection soit s , soit \tilde{s} seulement. En effet, on a $|support_{\mathcal{D}}(\tilde{s})| = |support_{\mathcal{D}}(s)| \times \left(\frac{balance(\tilde{s})}{balance(s)}\right)$ et $balance(s) + balance(\tilde{s}) = 1$: on peut retrouver les valeurs support et équilibre de l'une à partir de l'autre.

4 Notes algorithmiques

Pour extraire des stratégies fréquentes et leur équilibre, nous pouvons utiliser tout algorithme d'extraction de motifs séquentiels fréquents et calculer pour chaque motif le support de sa séquence duale pour déterminer sa mesure d'équilibre. Cette technique triviale (appelée naïve en Section 5) ne passe pas à l'échelle. Nous utilisons l'arbre de motifs fréquents de *PrefixSpan* pour un calcul plus efficace de la mesure d'équilibre et éviter les motifs redondants.

L'algorithme PrefixSpan. Pour extraire les motifs fréquents, cet algorithme construit récursivement une nouvelle base de séquences de taille inférieure à chaque étape (base projetée). A chaque appel, on fournit une base de séquences en paramètre ainsi que le motif séquentiel fréquent en cours (la séquence vide lors du premier appel). On compte pour chaque item le nombre d'apparitions dans cette base de données fournie en paramètre. Si le nombre d'occurrences est supérieur au seuil fourni par l'utilisateur, alors un nouveau motif fréquent est créé en concaténant cet item avec le motif en cours d'analyse obtenu lors des appels récursifs précédents, et ce motif devient le motif séquentiel fréquent en cours pour les appels récursifs qui vont suivre. Au final, l'algorithme construit un arbre où chaque nœud correspond à un item dont découle la séquence obtenue par extension des nœuds précédents.

Exemple. Prenons l'exemple de la Table 2. Lors de la première itération, la base de données de séquences fournie est celle de la Table 2. Supposons que $minSupp = 3$. Pour tous les items, ou événements, de la base de données, on compte le nombre de séquences dans lesquelles ils apparaissent et si c'est supérieur à $minSupp$ on projette selon cet item et on effectue un appel récursif. Par exemple, l'évènement $(a_1, +, 1)$ apparaît dans 3 des 4 séquences de la base : on projette selon $(a_1, +, 1)$: on obtient la base projetée suivante $\{\{_ (a_2, +, 1)\}\{(a_4, -, 3)\}\}, \{\{_ (a_2, +, 1)\}\{(a_1, +, 2)\}\{(a_4, -, 3)\}\}, \{\{_ (a_2, +, 1)\}\{(a_4, -, 3)\}\}\}$. On effectue un appel récursif avec cette base projetée en paramètre ainsi que le motif fréquent $\langle\langle a_1, +, 1 \rangle\rangle$. Durant ce nouvel appel, on compte de nouveau le nombre d'occurrences de chacun des items présents. L'item $(a_2, +, 1)$ est présent dans les 3 séquences de la base projetée en cours : on projette selon $(a_2, +, 1)$: $\{\{\{(a_4, -, 3)\}\}, \{\{(a_1, +, 2)\}\{(a_4, -, 3)\}\}, \{\{(a_4, -, 3)\}\}\}$. On ajoute également le motif $\langle\langle (a_1, +, 1)(a_2, +, 1) \rangle\rangle$ à la liste des motifs séquentiels fréquents. On effectue un nouvel appel avec comme paramètres la nouvelle base projetée ainsi que le motif fréquent en cours $\langle\langle (a_1, +, 1)(a_2, +, 1) \rangle\rangle$. Lors de ce nouvel appel récursif, l'évènement $(a_1, +, 2)$ n'est présent que dans une seule séquence : ce n'est pas supérieur à $minSupp$, donc on ne projettera pas selon cet événement.

Calcul du support de la séquence duale. Le calcul du support de la séquence duale d'un motif est obligatoire pour déterminer la valeur de la mesure d'équilibre. *PrefixSpan* ne permet pas d'obtenir ce support de manière immédiate et une approche naïve n'est pas efficace. Nous proposons de le calculer en post-traitement mais en tirant bénéfice de la structure d'arbre de motifs produite par *PrefixSpan*. Pour cela, nous définissons tout d'abord $seq(i) = \{s \in \mathcal{D} | i \in s\}$ pour tout $i \in \mathcal{I}$ qui met en relation chaque item de l'alphabet avec l'ensemble des séquences qui contiennent cet item. La Figure 1 (gauche) donne un exemple de cette relation. Nous construisons la structure de données correspondante lors de la transformation de la collection de traces en base de séquences. Elle stocke au plus $|\mathcal{I}| \times |\mathcal{D}|$ éléments en mémoire (dans le pire des cas chaque item apparaît dans toutes les séquences). Nous pouvons noter aussi que chaque item apparaît au maximum une seule fois par séquence et que l'ensemble des items est muni d'un ordre total \leq tel que pour tout $(a_1, r_1, t_1), (a_2, r_2, t_2) \in \mathcal{I}$, $(a_1, r_1, t_1) \leq (a_2, r_2, t_2)$ ssi $t_1 \leq t_2$, $a_1 \leq a_2$, $r_1 \leq r_2$ (ordre sur les entiers naturels et ordres arbitraires sur l'ensemble des agents et actions, e.g. lexicographique). Ainsi, pour obtenir le support d'un motif séquentiel $\alpha = \langle A_1, A_2, \dots, A_n \rangle \diamond \langle b \rangle$ qui est une séquence obtenue par extension d'itemset ou de séquence, où A_i est un itemset et b un item, il suffit de faire l'intersection du support de $\langle A_1, A_2, \dots, A_n \rangle$ avec $Seq(b)$ qui est l'ensemble des séquences qui contiennent l'item b obtenu par la structure de données auxiliaire, puisque $b \geq i_n^m$ où i_n^m est le plus grand élément de A_n selon la relation d'ordre au sein d'un même itemset : $support_{\mathcal{D}}(\alpha) = support_{\mathcal{D}}(\langle A_1, A_2, \dots, A_n \rangle) \cap Seq(b)$.

Exemple. Afin d'illustrer le principe du calcul du support des séquences duales lors du post-traitement de *PrefixSpan*, prenons un extrait de l'arbre de motifs produit par *PrefixSpan* ainsi que la structure de données auxiliaire de la Figure 1 (gauche), obtenue à partir de l'exemple de la Table 2. Lors de l'énumération des résultats (en profondeur de gauche à droite), nous allons utiliser la propriété décrite dans le paragraphe précédent. Chaque nœud de l'arbre présenté dans la Figure 1 (à droite) correspond à un item, chaque fils d'un nœud signale une extension (d'itemset ou de séquence) du nœud parent. Le premier nœud traité par l'algorithme est le nœud 1. Comme ce nœud est situé au premier niveau de l'arbre (fils de la racine), pour calculer le support de la séquence duale du nœud 1, il suffit de récupérer via la structure de données auxiliaire le support de l'item dual de l'item associé au nœud 1. Ainsi, le support de la séquence duale du nœud 1 est $support_{\mathcal{D}}(2) = Seq(2) = \{4\}$, car l'item 2 est l'item dual de l'item 1. Le prochain nœud obtenu par l'énumération en profondeur de gauche à droite est le nœud 3. Le support de la séquence duale de la séquence correspondant au nœud 3 est : $support_{\mathcal{D}}(4) =$

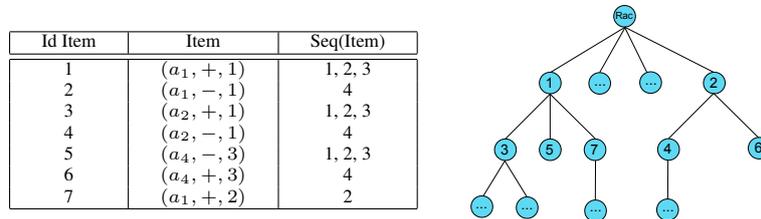


FIG. 1 – Structure de données $Seq(i), \forall i \in \mathcal{I}$ et l'extrait de l'arbre de motifs produit par *PrefixSpan* correspondant à la Table 2

$support_{\mathcal{D}}(2) \cap Seq(4) = \{4\}$ puisque le dual de l'item associé à 3 est l'item 4. Le nœud 5 est le prochain nœud de l'énumération, en considérant que l'on ait déjà traité les fils du nœud 3. Le support de sa séquence duale vaut $support_{\mathcal{D}}(6) = support_{\mathcal{D}}(2) \cap Seq(6) = \{4\}$.

Empêcher la redondance. Si s et \tilde{s} sont tous deux des motifs séquentiels fréquents, un seul doit être extrait (voir *Observation (Redondance dans la collection de motifs)*, Section 3). A la fin de l'exécution de *PrefixSpan*, on parcourt l'arbre des motifs fréquents ainsi. Un nœud est visité pour générer une séquence fréquente s , le nœud correspondant à sa séquence duale \tilde{s} est visité également (double parcours) et un "drapeau" y est levé (matérialisé par un booléen). Alors, un nœud où le drapeau est levé ne donne pas lieu à un motif fréquent en sortie.

5 Expérimentations

Nous expérimentons la capacité de notre approche à extraire des stratégies à partir de traces d'interactions autour du jeu vidéo de stratégie en temps réel StarCraft II (Blizzard Entertainment, 2010). Nous motivons d'abord ce choix dans le cadre émergent du jeu vidéo compétitif (sport-électronique, E-Sport) avant de montrer la faisabilité de l'extraction de stratégies et leur utilité pour les différents acteurs de l'E-Sport.

Le "Sport électronique", un champ d'application émergent. L'industrie du jeu vidéo a tiré grand profit de la digitalisation de notre société, générant des milliards d'euros par an et attirant des millions d'adeptes. Ce contexte a permis de révéler un phénomène ancien : le jeu vidéo compétitif. Le *sport électronique* est caractérisé par la pratique extrême des jeux vidéos, où des cyber-athlètes s'affrontent lors de compétitions internationales sur site ou sur le Web (avec des récompenses atteignant le million de dollar). Comme pour les sports classiques, ils sont entourés d'une équipe, d'entraîneurs et de sponsors. Les tournois sont diffusés par des commentateurs à travers des TV sociales suivies en grand nombre. Le lecteur curieux observera ce phénomène de société à travers le livre de Taylor (2012) et une étude quantitative de l'accès aux TV sociales dédiées par Kaytoue et al. (2012).

StarCraft II : un jeu de stratégie en temps-réel. Les éditeurs de jeux vidéos cherchent de plus en plus à proposer des licences où les joueurs professionnels peuvent exceller et les spectateurs se divertir (Cheung et Huang, 2011). L'une des licences phares de ce domaine est StarCraft II. Pendant le déroulement de la partie, les deux joueurs s'affrontent sur une carte de jeu (en vue aérienne), en contrôlant des bâtiments et des unités pour rassembler des ressources nécessaires à la construction de bâtiments et d'une armée afin de détruire la base adverse et ainsi gagner la partie². Nous proposons d'utiliser notre méthodologie pour discerner des stratégies gagnantes, perdantes ou équilibrées. L'intérêt pour les acteurs de l'E-Sport est multiple : (i) l'étude d'une sélection de parties d'un joueur particulier permet à son futur adversaire de préparer son match, et (ii) l'étude des stratégies les plus jouées et leur équilibre permet à l'éditeur du jeu d'étudier l'équilibrage et la diversité des stratégies jouées. Ce dernier point est capital pour l'éditeur faute de quoi le jeu perdrait tous ses adeptes. Il est une habitude pour la majorité des licences dans l'E-Sport d'observer les attitudes de joueurs et de modifier les règles lorsque des cas extrêmes ou gênants sont détectés.

Propriétés du jeu de stratégie en temps réel StarCraft II. Chaque joueur choisit une faction ("*Protoss*" (P), "*Terrans*" (T) ou "*Zergs*" (Z)). Il y a donc 6 possibilités de combinaisons

2. http://fr.wikipedia.org/wiki/Jeu_de_strat%C3%A9gie_en_temps_r%C3%A9el

de matchs possibles. Chaque combinaison implique des stratégies différentes dont les actions disponibles (ordonner la construction de bâtiments ou d'unités, le déplacement d'unités, ...) sont réalisées *en temps réel*. Chaque faction du jeu (P, T, Z) permet de construire des unités et des bâtiments différents avec des forces et des faiblesses variables suivant le principe du "pierre-feuille-ciseaux". Les joueurs évoluent dans un environnement *incertain* ou l'information est *imparfaite* : certaines intentions d'actions échouent ; un "brouillard de guerre" réduit le champ de vision des joueurs sur le jeu en dévoilant les bâtiment et les unités de l'adversaire seulement s'ils sont à portée de vue d'un bâtiment ou d'une unité du joueur.

Données. Une partie de StarCraft II est matérialisée par un *replay*, fichier où sont enregistrées toutes les actions des deux joueurs. Nous avons collecté 371267 *replays* à partir de différents sites web spécialisés, et avons retenu les 90678 replays impliquant des joueurs de haut niveau (maîtres et grands maîtres), seuls capables d'exécuter scrupuleusement une stratégie. Au final, 30 678 joueurs sont confrontés sur une période cumulée de 3 ans. La durée moyenne d'une partie est de 20 minutes où chaque joueur exécute environ 310 actions par minute.

Construction des séquences à partir des replays. Chaque type de rencontres (selon la faction de chaque joueur) implique des propriétés et stratégies différentes. Il y a 6 types de rencontres, nous dérivons donc autant de bases de séquences (Table 3). Pour caractériser les stratégies d'ouverture, seules les actions de type *ordre de construction d'un bâtiment* sont retenues³ : c'est le pilier d'une ouverture (bien que ce type d'ordre représente moins de 1% de toutes les actions). Ainsi, une séquence est une liste ordonnée d'évènements de type *construction d'un bâtiment*, élément d'une stratégie gagnante ou perdante pendant une fenêtre de temps (discrétisation avec $w = 30$ secondes par défaut). Ainsi, on construit pour chaque type de match une base de séquences \mathcal{D} définie sur l'alphabet $\mathcal{I} \subseteq A \times R, T$ où A est un ensemble de bâtiments et T dénote les fenêtre de temps. Le logiciel Sc2Gears⁴ est utilisé pour la génération de séquences à partir des replays bruts. Dans un second temps, nous modifions la définition de l'alphabet \mathcal{I} en ajoutant pour chaque item son indice d'apparition dans la séquence, i.e. $\mathcal{I} \subseteq \mathcal{A} \times \mathcal{T} \times \mathcal{R} \times \mathbb{N}$. Cela nous permet de caractériser une même action qui se produit plusieurs fois dans la même fenêtre et dans la même séquence, ce qui peut être intéressant selon le type de rencontres d'après l'expert (Table 3, partie droite).

\mathcal{D}	$\mathcal{I} \subseteq A \times T \times R$				$\mathcal{I} \subseteq A \times T \times R \times \mathbb{N}$			
	$ \mathcal{I} $	$ \mathcal{D} $	IS	I/IS	$ \mathcal{I} $	$ \mathcal{D} $	IS	I/IS
PvP	1,160	6,668	11.5	2.0	5,624	6,668	11.6	2.1
PvT	3,655	18,754	19.0	2.6	24,019	18,754	19.4	2.7
PvZ	3,748	22,784	19.6	2.7	27,913	22,784	20.1	2.8
TvT	2,201	7,457	20.7	2.8	15,572	7,457	21.2	2.9
TvZ	4,492	23,637	22.5	2.8	33,050	23,637	23.1	2.9
ZvZ	1,689	9,554	14.2	2.2	9,139	9,554	14.7	2.3

TAB. 3 – Caractéristiques des différentes bases de données de séquences. IS (resp. I/IS) représente le nombre moyen d'itemsets (resp. d'items par itemsets) en moyenne.

Résultats quantitatifs. Nous avons exécuté les expérimentations sur une machine avec 4 Go de RAM et processeur cadencé à 2,5 GHz. Nous avons adapté l'algorithme *PrefixSpan* de Pei et al. (2001) pour extraire les motifs séquentiels fréquents et effectuer les post-traitements pour calculer la mesure *balance* et supprimer les motifs redondants. Les résultats quantitatifs

3. Seuls les bâtiments qui permettent la construction d'unités ou de technologies

4. <https://sites.google.com/site/sc2gears/>

obtenus ont été réalisés sur le jeu de données PvP (les autres bases exhibent les mêmes comportements). A partir de maintenant, les seuils de support minimum sont donnés relatifs à la taille de la base considérée. La figure 2 (à gauche) présente le gain obtenu en post-traitant l'arbre de motifs fréquents comparé à l'approche naïve, ce qui permet d'utiliser des seuils $minSupp$ très bas. On y voit aussi la proportion de motifs redondants supprimés (entre 10% et 40%) sans ajouter de temps d'exécution. La Figure 2 (à droite) montre expérimentalement que le temps d'exécution double avec les post-traitements introduits. La figure 3 représente pour chaque motif la cardinalité de support et sa mesure d'équilibre. Notons son allure symétrique où la droite $y = 0,5$ représente les motifs équilibrés et que les motifs les plus fréquents sont plus équilibrés.

Découverte de stratégies d'ouverture. Les stratégies d'ouverture sont les plus connues et exécutées durant les 5 à 10 premières minutes d'une partie. Il est attendu des éditeurs du jeu qu'elles soient équilibrées, sinon le jeu n'aurait aucun attrait pour les joueurs et spectateurs. Étant donné le codage utilisé avec un numéro de fenêtre, on s'attend à ce que les ouvertures se retrouvent dans les motifs les plus fréquents, et qu'ils soient équilibrés. La Figure 3 montre empiriquement cela. Nous avons alors exploré avec un expert du jeu les motifs les plus fréquents. Par exemple, la base PvZ contient 591 motifs avec $minSupp = 5\%$. Les motifs les plus fréquents représentent des stratégies d'ouverture connues : $s = \{(Nexus, 5, g)\} \{(Gateway, 6, g)\} \{(Photon Cannon, 6, g)\}$ dénote la stratégie la plus utilisée par le joueur Protoss, peu importe la stratégie de son adversaire. Cette stratégie est équilibrée : $balance(s) = 0.52$.

Diversité de stratégies dans les rencontres miroirs. Il est plus délicat de parler d'équilibre lorsque les deux joueurs utilisent la même faction : ils ont à leur disposition les mêmes actions, à partir desquelles ils peuvent élaborer des stratégies différentes. Prenons par exemple la base PvP, en prenant en compte l'apparition de chaque item (Table 3 partie droite), avec $minSupp = 5\%$: 3418 motifs sont extraits. On y trouve la stratégie bien connue "4 Gates" par exemple à travers le motif $s = \{(Gateway, 3, reussite, 1)\} \{(Assimilator, 3, reussite, 1)\} \{(Cyb.Core, 4, reussite, 1)\} \{(Gateway, 7, reussite, 2)\} \{(Gateway, 7, reussite, 3)\} \{(Gateway, 7, reussite, 4)\}$ dont la mesure d'équilibre est $balance(s) = 0.59$. Ce résultat est surprenant de premier abord, mais reflète le côté efficace de cette stratégie : sa mise en œuvre est la plus facile et moins risquée que d'autres à exécuter. Depuis la sortie d'une mise à jour majeure du jeu (avec des bâtiments et unités supplémentaires), les stratégies utilisées lors des rencontres

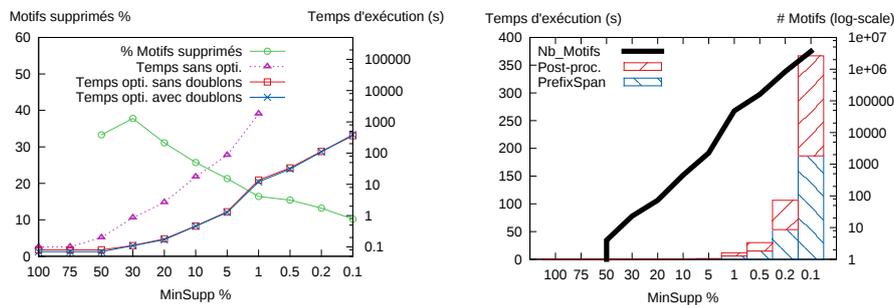


FIG. 2 – Temps d'exécution et nombre de motifs

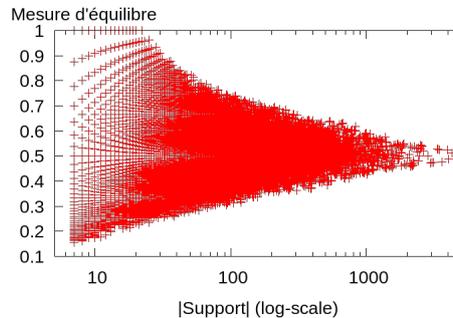


FIG. 3 – *Mesure d'équilibre et support absolu des motifs extraits.*

PvP se sont diversifiées et cette stratégie est moins utilisée d'après l'expert (nous n'avons pas les données pour le vérifier). La diversité est un aspect majeur pour l'éditeur.

Découverte de stratégies déséquilibrées. Les stratégies déséquilibrées doivent être détectées car l'équilibre est le point clé du sport électronique. Effectivement, un jeu déséquilibré n'a aucun intérêt. Afin de mettre en évidence une stratégie déséquilibrée, nous avons demandé à notre expert de nous donner un tel exemple connu. La stratégie appelée "Bunker Rush" est une stratégie pratiquée par les joueurs Terran contre des Zerg. Elle consiste à construire rapidement un *Bunker* dans la base adverse afin de perturber son économie et ainsi son développement. L'extraction dans la base TvZ avec $\text{min.Supp} = 1\%$ retourne 17990 motifs dont 602 mettaient en œuvre la construction d'un *Bunker* tôt dans la partie. Les 20 motifs les plus fréquents voient leur valeur d'équilibre supérieure à 0.6 ou inférieure à 0.4. Depuis, les développeurs ont changé des propriétés du jeu afin de limiter cette stratégie devenue peu jouée d'après l'expert.

6 Conclusion

Nous avons présenté une approche d'extraction de stratégies à partir de traces d'interaction entre deux agents en compétition et l'avons appliquée à des données du jeu de stratégie en temps réel StarCraft II, pour une validation à travers des résultats quantitatifs et qualitatifs. Nous la croyons d'intérêt dans le domaine d'application émergent qu'est le sport-électronique (mais aussi dans le sport classique). La méthode peut aussi être intégrée dans des outils de visualisation de traces de StarCraft II comme celui de Wender et al. (2013). Ces mêmes traces ont servi de support dans l'étude de Thompson et al. (2013) pour définir diverses métriques afin de différencier un novice d'un expert en sciences cognitives : les motifs extraits peuvent permettre de nouvelles métriques ("la capacité à suivre une stratégie"). La mesure d'équilibre définie peut se voir comme une mesure d'émergence (Dong et Li (1999)), mais sa définition et son calcul ne sont pas triviales à partir des traces brutes. Pousser le calcul de cette mesure lors de l'extraction des motifs fréquents ou top- k -fréquents (et non en post-traitement) fait partie des perspectives tout comme la construction d'un modèle prédictif de victoire au plus tôt à partir des motifs.

Références

- Chan, S., B. Kao, C. L. Yip, et M. Tang (2003). Mining emerging substrings. In *DASFAA*, pp. 119–126. IEEE Computer Society.
- Cheung, G. et J. Huang (2011). Starcraft from the stands : understanding the game spectator. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, New York, NY, USA, pp. 763–772. ACM.
- Dong, G. et J. Li (1999). Efficient mining of emerging patterns : discovering trends and differences. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '99, New York, NY, USA, pp. 43–52. ACM.
- Fürnkranz, J. (2001). Machines that learn to play games. Chapter Machine learning in games : a survey, pp. 11–59. Commack, NY, USA : Nova Science Publishers, Inc.
- Kaytoue, M., A. Silva, L. Cerf, W. Meira Jr., et C. Raïssi (2012). Watch me playing, i am a professional : a first study on video game live streaming. In *Proceedings of WWW'12 (Companion Volume)*, pp. 1181–1188. ACM.
- Lund, K. et A. Mille (2009). *Analyse de traces et personnalisation des environnements informatiques pour l'apprentissage humain*, Chapter Traces, traces d'interactions, traces d'apprentissages : définitions, modèles informatiques, structurations, traitements et usages , pp. 21–66.
- Pei, J., J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, et M. Hsu (2001). Prefixspan : Mining sequential patterns by prefix-projected growth. In D. Georgakopoulos et A. Buchmann (Eds.), *ICDE*, pp. 215–224. IEEE Computer Society.
- Taylor, T. L. (2012). *Raising the Stakes : E-Sports and the Professionalization of Computer Gaming*. MIT Press.
- Thompson, J. J., M. R. Blair, L. Chen, et A. J. Henrey (2013). Video game telemetry as a critical tool in the study of complex skill learning. *PLoS ONE* 8, e75129.
- Wender, S., A. Cordier, et I. Watson (2013). Building a trace-based system for real-time strategy game traces. In *Proceedings of EXPPORT : EXperience reuse : Provenance, Process-ORientation and Traces, ICCBR 2013, Saratoga Springs, NY, USA, 8-11 July 2013*.
- Yan, X., J. Han, et R. Afshar (2003). Clospan : Mining closed sequential patterns in large databases. In D. Barbará et C. Kamath (Eds.), *SDM*. SIAM.

Summary

To reach a goal, any agent competing against another work out strategies. When enough of such interaction traces are collected, it follows quite naturally to characterize strategies with frequent sequential patterns. In this article, we present a KDD process aiming at finding strategies and their balance. The method involves a particular representation of interaction traces as a sequence database and the introduction of a novel measure where unbalanced strategies are characterized as emerging patterns. Experiments are presented on real-world collections of interaction traces of a real time strategy game showing the efficiency of the method and its ability to discover actionable patterns for an emerging domain: electronic sports.